Coordination in Service Value Networks

A Mechanism Design Approach

Zur Erlangung des akademischen Grades eines
Doktors der Wirtschaftswissenschaften

(Dr. rer. pol.)

von der Fakultät für
Wirtschaftswissenschaften
der Universität Karlsruhe (TH)

genehmigte

Dissertation

von

Dipl.-Inform.Wirt Benjamin Sebastian Blau

Tag der mündlichen Prüfung: 31.07.2009

Referent: Prof. Dr. Christof Weinhardt

Koreferent: Prof. Dr. Rudi Studer

Prüfer: Prof. Dr. Oliver Stein

2009 Karlsruhe

Abstract

The fundamental paradigm shift from traditional value chains to agile service value networks (SVN) implies new economic and organizational challenges. In service value networks, a multitude of participants co-create complex services that create added value for customers by providing highly specialized service components and by leveraging lightweight paradigms such as RESTful architectures and mashup technologies. Addressing the challenge of coordinating distributed activities in order to achieve a desired outcome, auctions have proven to perform quite well in situations where intangible and heterogeneous economic entities are traded [Smi89, LR00].

Nevertheless, traditional approaches in the area of multidimensional combinatorial auctions [BK05, Sch07] are not quite suitable to enable the trade of composite services. A flawless service execution and therefore the requester's valuation highly depends on the accurate sequence of the functional parts of the composition, meaning that in contrary to service bundles, composite services only generate value through a valid order of their components. From a technical perspective, service composition research [ZBD$^+$03] traditionally assumes complete information about QoS characteristics and prices and does not account for self-interested service owners that intent to maximize their utility and therefore behave strategically.

Addressing these challenges, in the work at hand, the complex service auction (CSA) is developed following a mechanism design approach. The auction mechanism facilitates the allocation of multidimensional service offers within service value networks, enables service level enforcement and determines prices for complex services. The mechanism and the bidding language support various types of QoS characteristics and their individual aggregation by incorporating semantic information. Compliant with state of the art standards such as WS-Coordination, a possible implementation of the complex service auction in distributed environments is presented and a computational tractable algorithm to solve the winner determination problem is introduced.

Leveraging analytical and numerical research methods, the mechanism's properties are evaluated comprehensively. It is analytically shown that the social choice

implemented by the complex service auction is incentive compatible with respect to all dimensions of the service offer (quality and price), i.e. although service providers act strategic, it is a weakly dominant strategy to report their multidimensional type truthfully to the auctioneer. Counteracting the absence of budget balance, a payment scheme is presented which is robust to manipulation and at the same time incentivizes service providers to increase their services' degree of interoperability which is shown by means of an agent-based simulation. To leverage synergies and to reduce costs, it is beneficial for service providers under certain circumstances to offer bundled services. Depending on how service providers are situated within a service value network, bundling and unbundling strategies are analyzed following a simulation approach.

Acknowledgements

This work would not have been possible without the guidance and support of many people. I would like to thank my advisor Professor Dr. Christof Weinhardt for giving me the great opportunity to do this work and for his constant support and innovative ideas. He granted me the freedom and the help necessary and encouraged me during in times.

Additionally, I would like to thank my co-advisor Professor Dr. Rudi Studer for his guidance and fruitful discussions that improved and enriched especially the technical elements of my work. Thanks also to the other members of the committee, Professor Dr. Oliver Stein and Professor Dr. Stefan Tai who in particular sensitized me to additional technical aspects to round up this work.

I would like to thank the outstanding team of the research group on Information and Market Engineering at the Institute of Information Systems and Management (IISM) and the colleagues of the Karlsruhe Service Research Institute (KSRI). Their inspiration and valuable comments significantly improved my work and helped me to solve initially "unsolvable" problems. I would also like to thank Professor Dr. Dirk Neumann for his support in the early stage of this research and his seminal ideas. In particular I am grateful to my friends Tobias Conte and Jochen Stößer for proof reading major parts of this work and especially for providing me with critical and constructive questions and comments.

Above all, I am indebted to my parents, Thomas Blau and Heide Blau, to my sister Alexandra Blau, and to my fiancée Katharina Gofron. This work would not have been possible without their constant support and their caring encouragement.

Benjamin Blau

Contents

List of Figures

List of Tables

List of Abbreviations

ACID	Atomicity, Consistency, Isolation, Durability
B2B	Business-to-Business
BN	Business Network
BPEL	Business Process Execution Language
CRM	Customer Relationship Management
CTF	Compatibility Transfer Function
FIN	Finance
FOL	First-Order Logic
FTP	File Transfer Protocol
GXL	Graph eXchange Language
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
ICT	Information and Communication Technology
IT	Information Technology
JSON	JavaScript Object Notation
QoS	Quality of Service
RDF	Resource Description Framework
REST	Representational State Transfer
RPC	Remote Procedure Call
RSS	Rich Site Summary
SaaS	Software-as-a-Service
SBN	Smart Business Network
SCM	Supply Chain Management
SLA	Service Level Agreement
SMTP	Simple Mail Transfer Protocol
SOA	Service-oriented Architecture
SOAP	Simple Object Access Protocol
SROM	Service Request and Order Management
SVN	Service Value Network
SVNP	Service Value Network Planner
UDDI	Universal Description, Discovery, and Integration
UML	Unified Modeling Language
URI	Uniform Resource Identifier
VCG	Vickrey-Clarke-Groves
VO	Virtual Organization
W3C	World Wide Web Consortium

WADL .... Web Application Description Language
WSDL .... Web Service Description Language
XML .... eXtensible Markup Language

Part I

Foundations

Chapter 1

Introduction

The principle of utility neither requires nor admits of any other regulator than itself.

[Ben38]

This chapter firstly motivates the work at hand in Section 1.1 and elaborates arguments that support the necessity and relevance of the addressed research questions. Section 1.2 describes the research outline and the research questions underlying this work. Based on the construction of the research outline, Section 1.3 briefly introduces the main structure followed by an illustration of the research development with respect to publications and presentations of different parts of this work.

1.1 Motivation

Businesses are undergoing a paradigm shift from developing and distributing goods to providing services as their core business [VL04]. As the focus on service customization increases in order to provide tailored-solutions to customers, companies gain competitive advantage through the provision of highly specialized services [VL04, LVO07]. In recent years the service sector has become a rapidly growing sector in world economies. In Brazil, Russia, Japan, and Germany, services account for 50 percent of the labor force and 75 percent of the labor force in the United Kingdom and the United States [OEC05]. The Bureau of Economic Analysis (BEA) reported that in the United States, the private service-producing sector continued to lead overall GDP growth in 2006, increasing by 4.2 percent, whereas growth in the private goods-producing sector decreased down to 0.8 percent [BEA08].

A renaissance of HTTP appreciation through e.g. the RESTful architectural style [Fie00, RR07] drives simplicity of service descriptions and interfaces and enables service consumers to participate in the so called *programmable Web*. A primer example for this trend is Amazon's Simple Storage Service (S3)[1] that is fully accessible and manageable through basic HTTP methods following a RESTful architectural style[2]. Programmatic access to services with lightweight APIs can be used by consumers without in-depth technical knowledge. In January 2008, Amazon announced that the Amazon Web Services[3] consume more bandwidth than the entire global network of Amazon.com retail sites [Ama08]. This reflects the shift from the production and consumption of statically presented information to ″living“ information services. Knowledge and information is more and more intensively shared by building situational services (e.g. service mashups, intelligent document mashups, situational applications) instead of statically predefined information goods (e.g. blog posts, information on static Web sites). Driven by simplicity and easy-of-use, this trend also implies a strong involvement of the service consumer in the production process of services. The process of consuming and contributing to service artifacts is no longer separable which results in a new role called the *service prosumer* who co-creates value proactively [TW06]. As the provision and consumption of services blurs, the number of co-created services increases rapidly.

Due to growing modularization and simplicity, services are composable in a plug-and-play fashion [VvHPP05, ZBD$^+$03] in order to be rearranged into value-added complex services. The process of composing and rearranging existing and newly created service components enables agile innovation processes [BC00]. All these trends foster a rapid growth of so called service value networks. Service value networks are constituted by loosely-coupled formations of companies that provide modularized services while concentrating on their core competencies. These Web-enabled services expose standardized interfaces and foster an ad-hoc composition in order to jointly generate added value for customers in an on-demand fashion.

Service composition enabled through modularization and simplicity leverages the power of business in the long tail [And06]. Flexible combining customized service components increases variety and individuality which leverages the power of mass-customization [DSBF01]. Traditionally, most of the individual demand for specialized services could not be satisfied by off-the-shelf solutions. By enabling the opportunity to co-create solutions and building nearly unlimited versions through innovating and recomposing loosely-coupled services into value-added complex services, demand is nearly generated by customers themselves.

Nevertheless, current leading service providers traditionally offer their services charging static prices (e.g. pay-per-use or flat fees). However, such static pricing

[1] `http://aws.amazon.com/s3/`

[2] A detailed introduction to the Amazon S3 architecture and the programmatic management can be found in [RR07]

[3] `http://aws.amazon.com/`

models do not reflect the agility and distributed nature of service value networks and situational applications from an economic perspective. Multiple distributed self-interested providers that contribute to a value-added complex service have different preferences for different outcomes which are private information. Static pricing schemes ignore such preferences and additional information that is inherent in the market. Although service providers like Amazon start to incorporate economies of scale in their pricing models [BBT09] these pricing schemes are still static and are not capable of balancing supply and demand. A primer example for dynamic pricing models in the context of electronic services is Google's AdWords[4] and Yahoo*!* Search Marketing[5]. Google for example provides a *generalized second price auction* to allocate and price keywords and corresponding search rankings [EOS07, Var09]. In the first quarter of 2009, 67 percent of Google's revenues are realized by the AdWords campaign and further 30 percent through the complementary AdSense program reflecting Google's partner network[6]. In total, Google's revenue is predominantly generated (97 percent) through its advertisement programs that are based on an *auction pricing model* [EOS07].

Auctions have proven to perform quite well in situations where intangible and heterogenous entities are traded [Smi89]. Furthermore, valuations are hard to determine for single and especially value-added complex services as the value of the service's outcome highly depends on the customer's preferences for which current pricing models do not account. Auctions are predestinated to *aggregate information* from distributed parties which results in an aggregated valuation [PS00, Jac03]. Without prior knowledge about the valuations of each participant, auctions can provide suitable incentives to make truth-revelation an equilibrium strategy and therefore automatically *aggregate necessary information* from self-interested participants to determine adequate prices for complex services.

1.2 Research Outline

The overall question underlying this work is how an adequate auction mechanism can be designed which enables the trade of complex (composite) services in distributed environments such as service value networks. A suitable mechanism must satisfy economic and applicability requirements and must at the same time be theoretically sound. A well-known result from Market Engineering states that there is no such thing as an omnipotent mechanism that is suitable and applicable in any domain and any setting [WHN03]. Thus, a mechanism design for the allocation and

[4] `http://adwords.google.com/`
[5] `http://searchmarketing.yahoo.com/`
[6] `http://investor.google.com/releases/2009Q1_google_earnings.html`

pricing of complex services depends on economic and technical characteristics of typical service offers in service value networks (e.g. utility and elementary services with different QoS characteristics), different requesters' preferences for various QoS characteristics of complex services [ZBD$^+$03] and the overall goals of the mechanism designer (e.g. revenue vs. welfare maximization) [Rot02, Neu04]. Addressing these challenges and satisfying detailed requirements derived from an environmental analysis, the work at hand extends the body of research on mechanisms for trading combinatorial entities with special focus on sequential compositions of service components in service value networks.

The first research question deals with the properties of service value networks and complex services which embody the final outcome that is provisioned to service requesters. As an initial step, this question lays the groundwork for the design of an adequate mechanism that enables the trade of service compositions in service value networks. Hence, the first research question is stated as follows:

Research Question 1 ≺ENVIRONMENTAL ANALYSIS≻. *What are the characteristics of service value networks and complex services, and what are resulting economic and applicability requirements upon a mechanism to coordinate value creation?*

The question is addressed by *(i)* defining *traditional services*, *e-service*, *software services* and *Web services* and analyzing their key characteristics, *(ii)* providing a clear understanding of *service value networks* by defining their characteristics, their structure, and their components and filling the lack of definitions in current related literature *(iii)* analyzing the concept of a *complex services* as a final outcome created by a service value network through the realization of a sequence of modularized service offers. Finally, based on these results, economic and applicability requirements upon an adequate mechanism for coordinating value creation in service value networks are derived. In summary, the environmental analysis and resulting requirement analysis serve as a starting point for the further development of the work at hand.

Targeting the core contribution of this work, the second research question addresses the challenge of how to design an adequate multidimensional and scalable auction mechanism which enables the allocation and pricing of complex services in service value networks.

Research Question 2 $\prec$MECHANISM DESIGN$\succ$. *How can a scalable, multidimensional auction mechanism for allocating and pricing of complex services in service value networks be designed that limits strategic behavior of service providers?*

The question is addressed by *(i)* providing an abstract *model of service value networks* that captures the key characteristics and components in a comprehensive manner, *(ii)* designing a *bidding language* that enables the specification of multi-dimensional service offers and service requests, *(iii)* specifying a *scoring function* to capture the service requester's preferences for different QoS characteristics and prices of complex services and *(iv)* designing an *auction mechanism* – the Complex Service Auction (CSA) – consisting of an allocation and transfer function that implements an allocative efficient, individual rational and incentive compatible social choice with respect to all dimensions of the providers' bids. Focusing on a computational tractable implementation of the auction mechanism, *(v)* an *algorithm* is presented that solves the winner determination problem in polynomial time regarding the number of service offers and feasible service compositions.

While traditional service composition approaches assume complete information about the service components and their providers [ZBD$^+$03], service value networks are characterized by self-interested service providers that try to maximize their individual utility. Pursuing individual goals, service providers act strategically and have private information about their preferences for different outcomes [NR01, Par01] (e.g. information about true valuations and QoS characteristics of their services is private an cannot be assumed to be truthfully reported). Bridging this information gap, the approach of mechanism design targets the implementation of incentives (e.g. by means of an auction mechanism) that make truth-revelation a dominant strategy equilibrium and consequently allows for computing a system-wide solution. Nevertheless, traditional combinatorial auctions [BK05, Sch07] and especially corresponding bidding languages are not quite suitable to enable the trade of complex services. A flawless service execution and the requester's valuation for the outcome highly depends on the accurate sequence of the functional parts of the composition, meaning that in contrary to service bundles, complex services only generate value through a valid order of their components.

In order to enable the mechanism's application to the domain of service value networks and the coordination of distributed service activities, the following research question states the challenges regarding necessary applicability extensions to be addressed by this work:

Research Question 3 $\prec$APPLICABILITY EXTENSIONS$\succ$. *How can an auction mechanism be extended to support complex QoS characteristics and service level enforcement? How can the pricing scheme be modified in order to achieve budget balance and incentivize interoperability endeavors of service providers?*

Providing highly specialized services, providers shift from price to quality competition [Pap08]. Addressing the long tail of business, service providers tend to offer various customized versions of their services at different QoS levels in order to satisfy varying idiosyncratic demands. Consequently, a mechanism must account for complex QoS characteristics, that on the one hand are expressed by service providers and on the other hand are incorporated in the requester's preferences. The challenge is to provide a common conceptualization of quality attributes and enable their description, aggregation and enforcement from an economic and technical perspective. Addressing this question, the auction mechanism is extended in order to *support complex QoS characteristics* by means of rule-based semantic concepts and a toolbox of adequate aggregation operations. Furthermore, the mechanism is extended by a a compensation function which incorporates ex-post information about each services' performance in order to impose penalties if necessary. The compensation function is designed to implement a truth-telling equilibrium with respect to all dimensions of service providers' bids, i.e. truthful reporting of QoS attributes is a weakly dominant strategy for all service providers.

It is well-known in mechanism design research that based on strong theoretic results certain combinations of economic desiderata are impossible to achieve at the same time [GL78, Wal80, HW90, MS83]. There exist interdependencies between the properties of a mechanism and implemented social choice. Thus, mechanism design goals often result in a trade-off between different properties. Budget balance is an important property for a mechanism in order to be sustainable in the long-run as continuous external subsidization is neither reasonable nor profitable for e.g. a platform provider. Addressing the second part of Research Question 3, an extended transfer function – *the Interoperability Transfer Function (ITF)* – is developed which *restores budget balance* by sacrificing incentive compatibility to a certain extent and at the same time *incentivizes service providers to increase their services' degree of interoperability*, i.e. to increase the capability of their offered services to communicate and function with other services within the service value network.

The challenge of how a mechanism's properties can be evaluated by means of analytical and numerical methodologies is stated in the following research question:

Research Question 4 $\prec$EVALUATION$\succ$. *How can an auction mechanism be analytically and numerically evaluated regarding its economic properties as well as cooperation and bundling strategies of service providers?*

Research Question 4 is firstly addressed by an analytical evaluation of the mechanism's properties which shows that the complex service auction implements a social choice that is *allocative efficient* and *incentive compatible* with respect to all dimensions of service providers' bids, i.e. truth-revelation of private QoS attributes and valuations of offered services is an equilibrium in dominant strategies. Furthermore it is analytically shown that there exist ex-ante agreements between service providers about a *form of cooperation* to reduce internal costs that are mutually beneficial.

By means of simulation-based analysis, the extended budget-balanced transfer function is evaluated with respect to the robustness against bid manipulation, i.e. to what degree it is beneficial for service providers to deviate from their true valuation. Results show that even in settings with a low level of competition *strategic behavior of service providers is tremendously limited* as a deviation from a truth-telling strategy is not significantly beneficial even in small service value networks. The incentive for service providers to increase their services' degree of interoperability is numerically evaluated by means of an agent-based simulation. Compared to an equal transfer function which distributes available surplus equally among allocated service providers, it is shown that the ITF extension implements incentives to *foster a higher overall degree of interoperability* in settings with a low level of competition. Thus, the ITF extension supports service value networks in an early stage of development as a high degree of interoperability increases the multitude of feasible complex service instances that can be offered to customers. An increase of variety and interoperability leverages network externalities [SV99, FK07, LM94, KS85] and attracts customers which in turn attracts more service providers to participate in the complex service auction.

Broadening the strategic scope of service providers that participate in the complex service auction, it might be beneficial from a provider perspective – dependent on how they are situated within the service value network– to offer their services as a bundle together with matching service providers. This question is addressed by means of an agent-based simulation. It is evaluated if it is beneficial to offer bundled services which decreases flexibility but leverages synergy effects and reduces costs or if it is beneficial to offer single highly specialized services that are more flexibly composable into various complex service instances. In summary, there two main strategies analyzed: *(i)* Competing in quality through differentiation and flexibility and *(ii)* competing in price through bundling synergies and cost reduction. Results show that in general service providers that own services within the service value network which are highly competitive, i.e. they are likely to be allocated, act best by following an unbundling strategy. In contrary, for service providers with less competitive service offers it is beneficial to form bundled service offers while leveraging synergy effects. Nevertheless, this strategic recommendation only holds in settings with a low level of competition.

1.3 Structure

The outline of this work is structured accordingly as depicted in Figure 1.1.

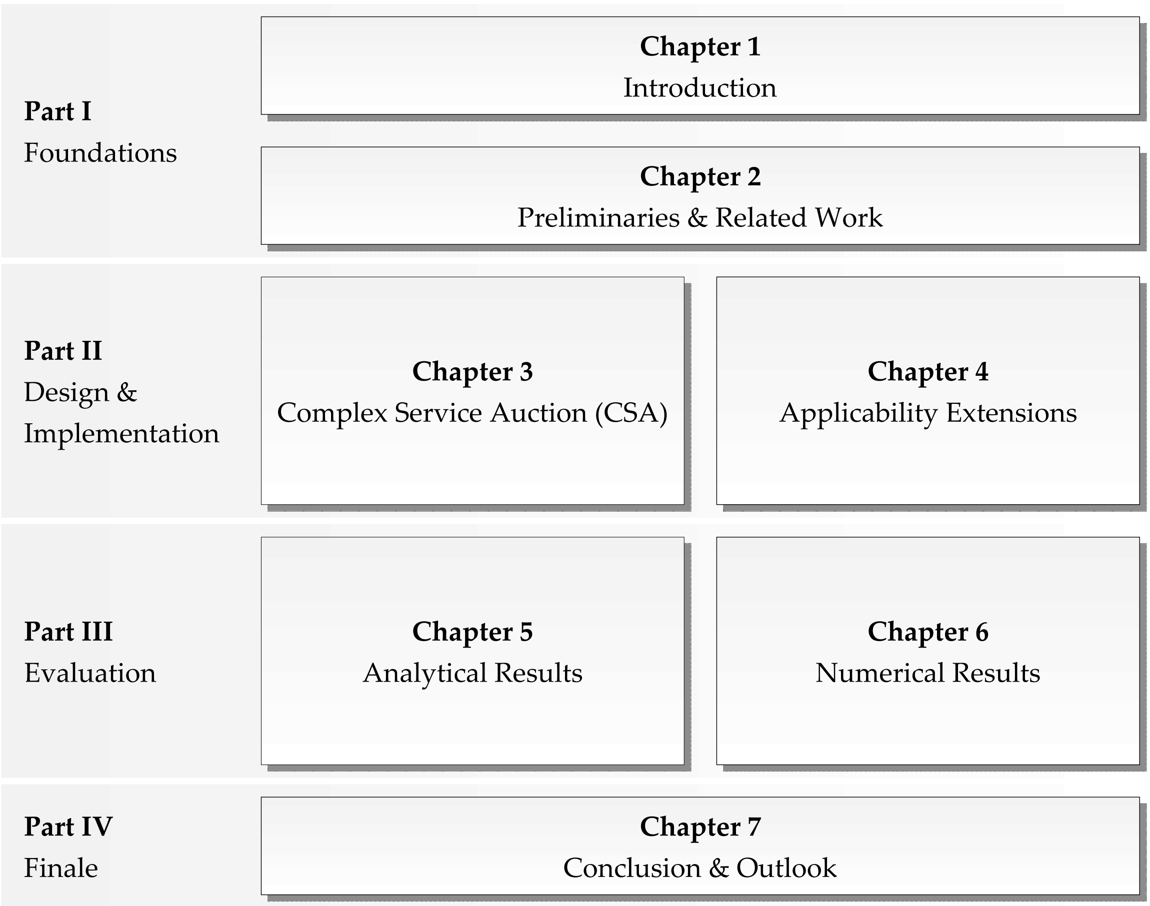


Figure 1.1: Structure of this work.

Chapter 2 introduces technologies, concepts and methods, which are fundamental for the work at hand. First, the concepts and key characteristics of different kind of services are discussed and corresponding definitions are outlined. Then service enabler technologies and paradigms such as service-oriented architectures, service value networks, and situational applications are introduced in detail. Bridging the gap between a more technical to an economic perspective, the idea of service markets is introduced and motivated in the context of complex services and service value networks. The discussion is followed by the description of the discipline of market engineering, which provides a structured approach for designing, implementing, and evaluating market mechanisms in different domains such as the service sector. The approach of mechanism design underlying the work at hand is introduced as well as important impossibility and possibility results. Summarizing the preliminaries, economic and applicability requirements upon a suitable mechanism for trading complex services in service value networks are discussed The requirement analysis is followed by a detailed description of related approaches in that particular research area with respect to stated requirements and identified shortcomings. Chapter 2 concludes with a brief description of research methods, which are used to analyze the research questions throughout this work.

Introducing the core model and mechanism implementation of the complex service auction as well as corresponding applicability extensions, Chapters 3 and 4 embody the central part of this work. Based on the design part, Chapters 5 and 6 analyze properties of the complex service auction mechanism following analytical and numerical research methods. For the convenience of the reader, each chapter entails detailed related work regarding the specific research question addressed additionally to the previously outlined approaches, which are closely related to the work at hand.

Finally, Chapter 7 summarizes the key contributions of this work, outlines complementary research and points out further challenges to be addressed in the future.

1.4 Publications & Research Development

Excerpts of this thesis have been published in European and international academic conferences and as journal articles. This section provides a brief overview regarding what parts have been presented, discussed and refined in the context of which research community. This section furthermore illustrates how the work at hand has been developed focusing on its steps of refinement and extension.

Laying the groundwork for this work at hand in Chapter 2, an analysis about characteristics of traditional and e-services as well as corresponding service definitions have been published in the Proceedings of the 18th International World Wide Web Conference (WWW 2009) [MB09]. The service decomposition model and the conceptual framework for categorizing different service artifacts have been presented at the Multikonferenz Wirtschaftsinformatik [BS08] and a revised version at the Joint Conference of the INFORMS Section on Group Decision and Negotiation, the EURO Working Group on Decision and Negotiation Support, and the EURO Working Group on Decision Support Systems [BBS08].

Basic ideas and concepts about situational Web applications introduced in the preliminaries have been published in the Proceedings of the 2nd Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009, WWW 2009 pre-conference workshop) [BLH09]. A first position paper about service value networks, their differentiation from related concepts, characteristics, components, and an abstract model has been presented at the 11th IEEE Conference on Commerce and Enterprise Computing (CEC 2009) [BKCvD09].

With respect to Chapter 3, first versions of the auction mechanism and the idea of applying path auctions to composition problems have been published in the 10th IEEE Joint Conference on E-Commerce Technology (CEC 2008) and Enterprise Computing, E-Commerce and E-Services (EEE 2008) [BLNW08]. A further refined version of the model including first simulation-based evaluations have been presented at the 16th European Conference on Information Systems (ECIS 2008) [BNWM08]. The next step of revision and extension of the complex service auction has been published in the Proceedings of the 9th International Conference on Business Informatics [CvD09].

The comprehensive model of the complex service auction as introduced in the work at hand including a complete analytical analysis of the mechanism's properties with respect to allocation efficiency and incentive compatibility as outlined in Chapter 5 has been presented at the the 17th European Conference on Information Systems (ECIS 2009) [BCM09] and published in the Journal of Business and Information Systems Engineering, Special Issue Internet of Services (forthcoming) [BvDC$^+$09].

A simulation-based evaluation of service providers' bundling and unbundling strategies participating in the complex service auction as introduced in Chapter 6 has been submitted to the Journal Electronic Commerce Research and Applications, Special Issue on Emerging Economic, Strategic and Technical Issues in Online Auctions and Electronic Market Mechanisms [BvDCW09].

As outlined in Chapter 7, complementary and future research with respect to implementing mechanisms that – in contrary to traditional mechanism design goals – provide innovative incentives to support service value networks in their early stage of growth have been presented at the 15th Americas Conference on Information Systems (AMCIS 2009) [CBSvD09].

Chapter 2

Preliminaries & Related Work

In contrast to a good, a service is not an entity that can exist independently of its producer or consumer and therefore should not be treated as if it were some special kind of good, namely an 'immaterial' one.

[Hil99]

The goal of this chapter is to give a thorough introduction into technical and economic foundations, which are essential for the remainder of this thesis. The work at hand focuses on the design and evaluation of an auction mechanism to coordinate value generation among distributed parties. The mechanism design provides means for the feasible and efficient allocation and pricing of composite services in service value networks.

This chapter firstly discusses the differentiation between tangible and intangible goods and the central concept of a *service*. Based on these results, a *service decomposition model* is presented that provides a conceptualization scheme for different classes of services and highlights the concept of a *complex service*. Following these definitions and classifications, the paradigm of a *service-oriented architecture* is introduced, which embodies the key principles leading to enabler technologies for service-centric electronic networks. Technical foundations cover the concept of *Web services*, emerging technologies with a focus on lightweight protocols, puristic architectural styles and slim message formats as well as *quality of service* aspects and their legal manifestation in *service level agreements*. As coordination plays a central role in distributed environments with self-interested parties such as the Web, frameworks and specifications in the Web service context are introduced that provide means for realizing *coordination mechanisms* from a technical perspective.

As the work at hand focuses on not only distributed but also networked service environments, the emergence of *service value networks* as a novel form of inter-organizational interaction and value generation is described and a model for capturing essential characteristics is provided. Service value networks allow for the realization of short-living complex services that fulfil customers' needs on a individual basis. Hence, such *situational applications* and *service mashups* are briefly introduced.

Following this introduction of service concepts, definitions and technologies, the need for *auction mechanisms* in these environments is discussed. Since this work targets on providing a comprehensive design and evaluation of a suitable service coordination mechanism from a technical and an economic perspective, this chapter introduces the idea of *algorithmic mechanism design* and the interdisciplinary approach inherent in this emerging discipline. In the context of coordinating distributed and self-interested participants, central economic and computational desiderata, prominent mechanisms, and important impossibility results are outlined.

Finally, the research methods underlying this work are briefly introduced. This chapter introduces related work and state of the art that is broadly related to the research questions at hand. Adjacent literature, a clear differentiation and a detailed discussion is provided in the remainder of this thesis.

2.1 Service Concepts, Definitions, and Technologies

The whole concept of distributed (service-oriented) computing can be viewed as simply a global network of cooperating business objects.

[YP00]

The goal of this section is to provide a thorough introduction to the service concept itself, conceptual classification models, related paradigms and technology, and emerging service-centric environments.

Section 2.1.1 describes the differences between tangible and intangible goods and the concept of a service by elaborating specific properties that allow for a more or less strict differentiation. Based on this analysis, the service concept is defined and its main characteristics are presented in detail. Concretizing the service concept by restricting its production and consumption channels to primarily electronic networks, the concept of an e-service is described and its implications on the general characteristics of a service are argued.

These foundations lay the groundwork for a service decomposition model as illustrated in Section 2.1.2, which serves as a conceptual classification scheme for different types of services with respect to their granularity and level of abstraction.

Besides utility and elementary services, complex services – as a special type of service – are introduced in detail as they embody a central concept for the work at hand.

Section 2.1.3 is concerned with the paradigm of a service-oriented architecture and its key principles which can be seen as the foundation for enabler-technology such as Web services. Service-oriented architectures allow for the agile production and consumption of distributed services in electronic networks such as the Web, that is, they enable value generation from a technical perspective. Value, created by a service is mainly dominated by intangible elements that are experienced during its performance, which therefore highly depends on the service's quality. Hence, the main quality aspects that together constitute quality of service (QoS) are argued and how a legal foundation is constituted by service level agreements. Distributed service activities that foster value generation and produce an overall quality that is provisioned to the consumer must be coordinated by suitable mechanisms. By introducing a standardized framework that specifies how coordination can be realized in the context of Web services, this challenge is initially addressed from a technical perspective.

Designing suitable mechanisms to coordinate value generation through complex services requires a deep understanding of emerging forms of organization of distributed service activities. Therefore, Section 2.1.4 presents the concept of a service value network, its characteristics, the various roles involved and how they are organized in order to jointly create value for potential service requesters. The overall objectives underlying this value generation process are individually specified by the services requester and consequently change frequently. This leads directly to the concept of situational applications and service mashups which is elaborated from a technical and an economic perspective in the remainder of Section 2.1.4.

2.1.1 Tangibles, Intangibles, and Services

The differentiation between the terms good, intangible good, tangible good and service is ambiguous and not exhaustive in the literature. Nevertheless a fundamental understanding of the concepts at hand is inevitable to derive requirements and implications in the context of service value networks, value generation and their coordination.

2.1.1.1 Tangible and Intangible Goods

A good is an economic entity with a defined ownership. The ownership is defined by means of a legal right that allows the owner to use the good exclusively and to prevent others from doing so. According to [Hil99] there are two main characteristics of a good observable: *(i)* The existence of a good is independent of the existence of its owner, meaning that a good's identity is retained over time. *(ii)* Ownership rights can be transferred from one economic entity to another, which implies that goods are tradable. The owner of a good derives some economic benefit from it (in contrary to a *bad* that decreases the utility of its owner). A more rigorous differentiation between goods and services appears in the context of production. The production process of goods involves inputs and outputs that are entirely owned by the producer of the good. A good may be inventoried, sold or traded, consumed or disposed after production as separated activities. The fact that production and use are distinct activities is important from an economic perspective as it allows for the transfer and exchange of goods even multiple times.

Although most of the goods are material, economic entities exist that expose all key characteristics of a good but are immaterial. According to [Hil99], "these consist of intangible entities originally produced as outputs of persons, enterprises, engaged in creative or innovative activities of a literary, scientific, engineering, artistic or entertainment nature." Although these *information goods* are immaterial they are goods because ownership can be defined and transferred from one economic unit to another. The main value for the consumer is derived from the information itself. They are also intangible because they expose no physical dimensions (except from the medium the information is stored on, which is not the economic entity at hand). The production process itself is mostly very costly and time consuming, whereas the reproduction or copying of information goods is cheap. The value of information goods generally increases through sharing and use [SV99, BBL99][1].

2.1.1.2 Services

Analogues to the fact that attributes, properties and characteristics of a service are rather fuzzy, the concept of a service itself is hardly definable especially in a consistent way across different application areas. Complementary to a short definition, this section defines the service concept and differentiates it from adjacent concepts such as goods and products through the identification of its main characteristics and their implications.

In general a service is some kind of *activity* or *performance*. The result of such an activity is the *change of condition* of some person or good. This change of state is based on an agreement of the economic unit owning the good and the one providing the service [Hil77, Gad92].

[1]Note that this fact is not universally true. E.g. the value of private information about shares of a company decreases through sharing.

Definition 2.1 [SERVICE]. *A service is an activity which an economic unit A (service provider) performs for another economic unit B (service consumer) that results in a change of state or condition of an economic unit C whereas The output of that activity cannot circulate in the economy independently of economic unit C.*[2]

Services expose a set of unique characteristics that have strong implications from an economic perspective and allow a more or less consistent differentiation from traditional goods or products. In order to analyze key characteristics of services, it is important to differentiate the relevant phases of a service's lifecycle as depicted in Figure 2.1.

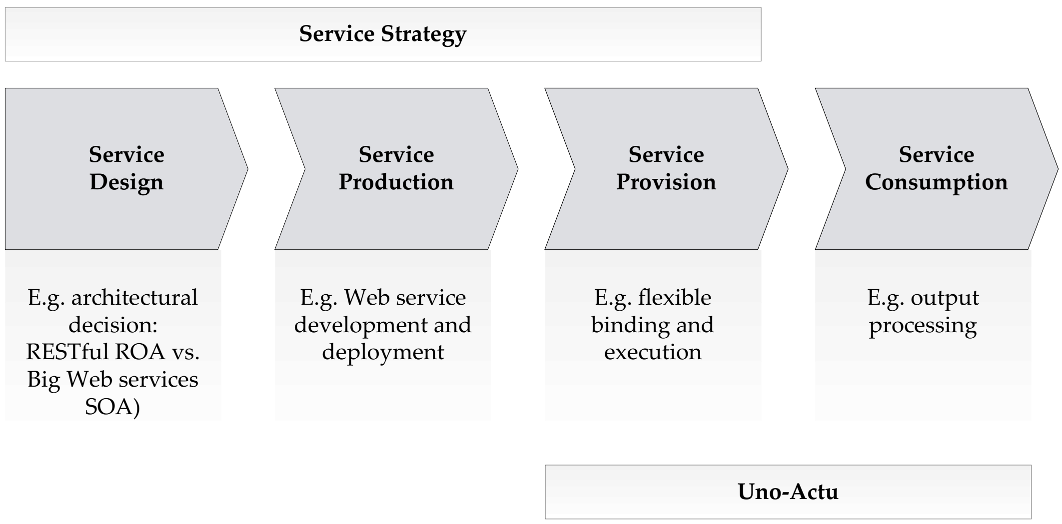


Figure 2.1: Service lifecycle. Elements are partly derived from ITIL V3[3]

The overall lifecycle is determined and evaluated based on a global strategy, i.e. the *service strategy*, that defines requirements and goals of the service portfolio. Based on initial requirements, the *service design* phase lays the groundwork while dealing with conceptual decisions regarding a service's design (e.g. is the room service available all the time? Which architectural design to choose for implementing a Web service?). Based on the initial design, the service itself is developed in the *service production* phase and all necessary resources for the service provisioning are prepared (e.g. a Web service is implemented using the Ruby programming language, a hotel room is cleaned and the mini bar is refilled). According to the central service characteristic, the uno-actu principle, which is explained in detail in the remainder of this section, *service provision* and *service consumption* occur simultaneously, i.e. they coincide in time under the presence of a producer and consumer. It is important to strictly differentiate between service production and provision, as the latter is the central phase for the following analysis of service key characteristics.

[2]This definition is based on [Hil77, Gad00]

[3]`http://www.itil-officialsite.com/`

In literature it has been argued that intangibility is the main characteristic to differentiate goods from services [Rat66, ZVB96]. Especially in the marketing area, intangibility has been identified as the most difficult aspect of services to deal with when it comes to the evaluation of service value creation as well as quality control and assurance [Lev81, LW01]. Focusing on economic properties and their implications for the coordination of value creation, intangibility is not the only fundamental characteristic to differentiate goods from services. The following list of the key service characteristics serves as a basis to derive requirements for adequate market mechanisms to coordinate value generation through services.

C 2.1 [UNO-ACTU]. *Service provision and consumption are not separable and coincide in time.*

In contrary to goods where the production, use and ownership can be separated from the economic entity itself, a service cannot be treated independently from its producer or consumer. "Services involve *relationships* between producers and consumers" [Hil99]. This implies that the process of production and consumption cannot be separated, meaning that there is no producer without a consumer and the other way around (e.g. a barber can only cut hair if the customer is present at the same time, which implies that there is no hair cutting activity possible without the barber or the customer being present). This principle is also called *uno-actu* and states that production coincides with consumption. *Uno-actu is the central and most important key characteristic of services*. Hence, it is fundamental to distinguish services from goods and it causally implicates most of the following service characteristics.

C 2.2 [NOT STORABLE]. *Services cannot be inventoried or produced on stock.*

The main value generated by the consumption of services comes from an action or performance. Service are *ephemeral* – *transitory* and *perishable* – which implies that they cannot be stored or produced on stock. It is not possible to produce services in advance in order to meet fluctuating demand. It is of great importance to distinguish between the actual performance that leads to an immediate change in state and its effect on reality. The activity itself on the one hand cannot be produced on stock as it is intangible and perishable. The person or good that is affected by this activity on the other hand can mostly be preserved over time [Gad00] (e.g. the actual deed of cutting hair cannot be produced on stock, whereas the change of condition – the physical cut hair – can be inventoried and exists over time). It has been argued by [Sta79] that the possibility to store and transport an economic entity is the main distinguishing element of services. Considering energy as an economic entity, this argumentation does not hold or must at least be relaxed, which questions its suitability for a strict differentiation.

C 2.3 [Co-creation]. *Services are generally co-created by their consumers.*

According to Definition 2.1, services are deeds or actions that change the condition of another economic unit. This economic unit – often referred to as external factor – is mostly brought in by the consumer. The consumer proactively influences the service activity and might therefore influence its result and quality. The degree of customer participation and co-production in the context of different service categories is analyzed in [BFHZ97]. Depending on the type of service *(i)* customer presence might be required during service delivery, *(ii)* customer inputs might be required for the actual service creation or *(iii)* customer inputs are completely mandatory. Co-production is argued to be the main characteristic to differentiate services from goods [Fuc68]. However, recent production strategies of traditional goods heavily integrate customers in the production process – often referred to as mass customization [PMS04] – which shows that co-production does not appear to be a suitable service characteristic in order to strictly distinguish services from goods.

C 2.4 [Intangible value creation]. *Value creation through services is characterized by intangible elements.*

Some services include physical elements in the process of value creation (i.e. spare parts during a repair process). However, the most value is created in the form of intangible, immaterial elements. The consumer of a service *experiences* the performance or activity, which embodies the main portion of created value [LW01]. Services create value when service consumers benefit from experiencing a service without a transfer of ownership (e.g. booking a hotel room). Due to this fact, the assessment of quality and its assurance is a critical issue in the context of services as an experience or an intangible result is hard to measure and strongly depends on the economic unit to which it is provided. A continuous spectrum from tangible-dominant to intangible-dominant to differentiate between goods and services is suggested in [Sho85].

C 2.5 [Fuzzy inputs and outputs]. *Service inputs and outputs are fuzzy and tend to vary more widely.*

Implied by the previous characteristic, it is hardly possible to control quality aspects of a service in a way that outcomes are predictable and constant over time [GW97]. Services are produced and consumed coincidentally and the value that is created during this process varies widely due to the lack of control instruments and various facets of service experience. This issue is even more intensified by another phenomenon that is specific to services. The quality of a service might depend on the ″quality″ or effort of the service consumer (e.g. in teaching or consulting) [Gri92]. Due to the fact that the quality or effort of a service consumer is not under the control of the provider and tends to vary from individual to individual, the final outcome of a service activity is fuzzy and varies more widely.

2.1.1.3 E-Services

With the rise of information and communication technology and the rapid growth of the Web, the environment for service development, production, provision and consumption has changed completely. In this context the concept of *e-services* emerged. The term e-service stands for a special form of "service that is provided over electronic networks" [RK02]. The e-service paradigm [RK03] is based on a broader view than the concepts of software services or IT services[4].

Definition 2.2 [E-SERVICE]. *An e-service or electronic service is a service provided over electronic networks.*[5]

Based on the implications of these novel environments that foster the e-service paradigm it is necessary to recall the service characteristics introduced in Section 2.1.1.2. As an e-service is a specific type of service, its characteristics are quite similar the characteristics of a general service. Nevertheless they have to be revised and adapted according to the conditions of the changed surroundings.

C 2.1 (UNO-ACTU) In the context of e-services, the roles "service producer" and "service consumer" are not strictly definable according to a traditional perspective. In most cases, the consumer of such a service is also an e-service or another automated electronic entity (e.g. search agents, spiders and robots). The role of the service producer is analogously hard to specify as e-services are developed and ready for execution via electronic networks, meaning that – under the assumption that there are no capacity constraints imposed by e.g. the network's bandwidth – these services can be performed anytime in a distributed manner to multiple consumers. Hence, dependent of how the provision and the actual consumption is defined in the context of e-services,

[4]"A Service provided to one or more Customers by an IT Service Provider. An IT Service is based on the use of Information Technology and supports the Customer's Business Processes. An IT Service is made up from a combination of people, Processes and technology and should be defined in a Service Level Agreement." [RH07]

[5]Based on the definition in [RK02]

this fact blurs the definition of the uno-actu principle which states that service producer and service consumer are contemporaneously involved in the performance of a service. Although the principle still holds in the e-service context, its relevance and implications on service provision and consumption have to be relaxed dependent of how provision and consumption are definable and separable.

C 2.2 (NOT STORABLE) E-services can be developed and stored to be ready for execution. Although the physical storage of the program code that determines the behavior of the service is possible, the actual execution, which is the value generating element of the service, can obviously not be performed on stock. This also implies a fluctuating supply as capacity constraints in the form of bandwidth or computing power limit the ability to satisfy peaks in demand. Resource-focused capacity constraints can partly be overcome by the use of computer grids or cloud computing environments that allow for the flexible scaling of computing power and storage.

C 2.3 (CO-CREATION) In order to perform a service, the consumer mostly has to provide additional information that is either transformed by the service or used to scope and customize the service execution according to the needs of the consumer. Although the service consumer does not bring in a physical economic entity that is a central part of the service activity, the consumer still influences and co-produces the final outcome of an e-service by providing necessary additional information or data. Thus, co-production is still a central element of service provision and consumption in the context of e-services.

C 2.4 (INTANGIBLE VALUE CREATION) Value that is created through the execution of an e-service is idiosyncratic and highly depends on the preferences of the service consumer. Although, the experience of a service performance in an electronic environment also depends on expectations, needs and preferences of the service consumer, e-services partly allow for an objective measurement of service quality, which highly correlates with the value generated. The proportion of value-determining aspects of a service outcome that can objectively be measured increases in the context of e-services, which leads to an increase of uncertainty about the value generated through a service activity.

C 2.5 (FUZZY INPUTS AND OUTPUTS) A great advantage of e-services is the possibility to describe their main functionality and capabilities in a standardized manner, which simplifies their usage and management. Inputs and outputs of e-services can be specified using standardized description languages that are common knowledge to service producers and service consumers. Thus, standardization and common sense about specifications reduce uncertainty about inputs and outputs in the context of e-services. Nevertheless, also in the context of electronic networks service, inputs and outputs highly depend on the state of the environment they 'live' in. E.g. capacity constraints, network failures and unreliable transportation influence the service outcome and its quality which increases uncertainty and unpredictability. Another factor that has an impact on the output generated by the service is the consumer's information that is either transformed or used to scope the service execution. Fuzzyness of service inputs and outputs can be reduced by means of standardized service description but is still an issue in the context of e-services.

Summarizing described key characteristics, Table 2.1 shows an overview over differentiation criteria of tangibles, intangibles, services, and e-services that have been discussed in this section.

Table 2.1: Differentiation criteria of tangibles, intangibles, services, and e-services. (● = fully satisfied, ◐ = partly satisfied, ○ = not satisfied, NA = not applicable)

Criterion	Tangibles	Intangibles	Services	E-Services
Ownership rights definable and transferable	●	●	○	○
Immaterial	○	●	●	●
Costly initial production	●	●	●	●
Costly reproduction	●	○	○	○
Sharing increases value	○	◐	NA	NA
Uno-actu	○	○	●	●
Not storable	○	○	●	◐
Co-creation	◐	◐	●	◐
Intangible value creation	○	●	●	●
Fuzzy inputs and outputs	NA	NA	◐	○

2.1.2 Service Decomposition Model

This section gives a thorough classification of groups of services that share common characteristics from a technical and economic perspective as depicted in Figure 2.2. The *Service Decomposition Model* is based on the classification in [BS08] and the extension in [BBS08]. The model distinguishes three different service layers grouping *Utility Services*, *Elementary Services* and *Complex Services*.

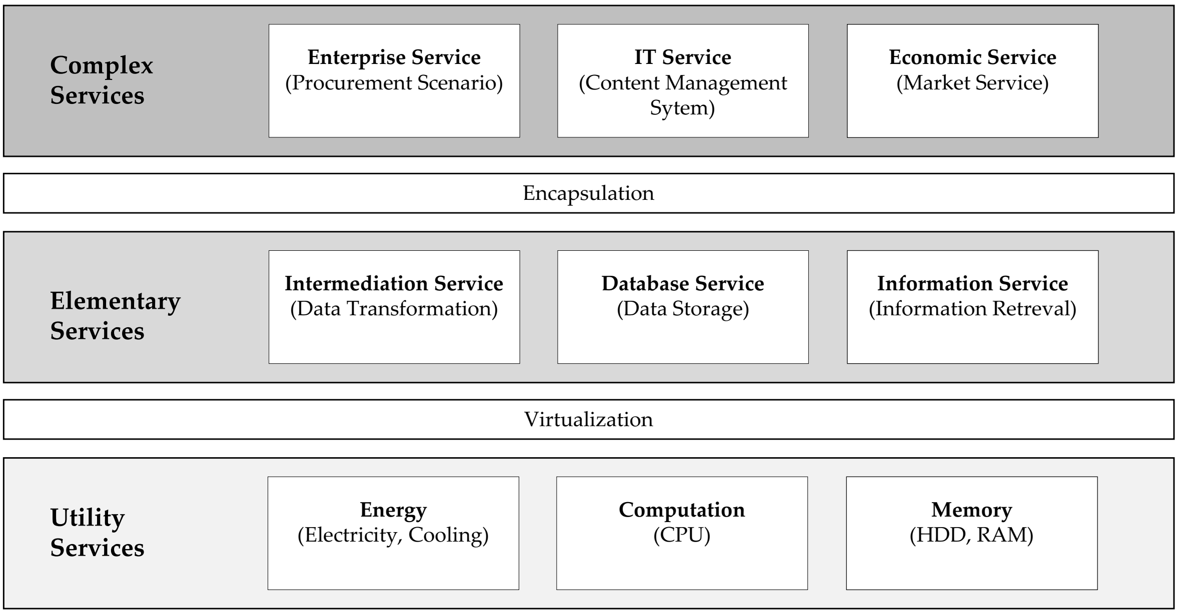


Figure 2.2: Service decomposition model [BBS08].

2.1.2.1 Utility Services

Utility services reflect a vision where services can be accessed dynamically in analogy to electricity and water: "Utility computing is the on-demand delivery of infrastructure, applications, and business processes in a security-rich, shared, scalable, and standards-based computer environment over the Internet for a fee. Customers will tap into IT resources – and pay for them – as easily as they now get their electricity or water." [Rap04]. Utilities are characterized by necessity, reliability, ease of use, fluctuating utilization patterns, and economies of scale. In [Rap04], base pricing in utility computing on metering usage (also coined "pay-what-you-use" or "pay-as-you-go") is suggested, as is the case with classic utilities such as water, telephone and Internet access. With the fast rise of energy prices, the meaning of utility services is even extended back to the roots where the name originally came from: Basic computing services in hosting centers need to be managed explicitly taking into account energy consumption as a relevant optimization criterion [CAT+01]. "Heterogeneous server clusters can be made more efficient by conserving power and energy while exploiting information from the service level, such as request priorities established by service level agreements" [BR04]. Even temperature aware computing solutions for data centers are proposed [MSS+08].

2.1.2.2 Elementary Services

Elementary services virtualize the utility services layer and encapsulate underlying functionality. They provide rather basic functionality such as data format converting services, storage services, or pure information services that retrieve information from designated sources. Although the type and behavior of these services are mostly standardized, they have multiple attributes with varying characteristics. For instance, storage services may differ according to their capacity, access time and data throughput. These varying characteristics of the same type of service, as well as the service itself can be described by means of standardized description languages. The input and output semantics of these so-called elementary services are well-accepted and interpretable. Examples might be database services and data format transformation services. Services in this layer are required for several different higher-level applications and, as a consequence, are utilized by a multitude of different users. Similar to utility services, the provided quality of service for the same type of service may vary. For instance, a set of data format transformation services may vary from their offered response time; however, it is assumed that these characteristics can also be described in a standardized form.

2.1.2.3 Complex Services

While elementary services provide simple functions such as credit checking and authorization, inventory status checking, or weather reporting, complex services may appropriately unify disparate business functionality to provide a whole range of automated processes such as insurance brokering, travel planning, insurance liability services or package tracking [PD04]. A complex service is composed of multiple service components (which are either elementary or complex themselves), often requiring an interaction or conversation between the user and services, so that the user can make decisions [MSZ01]. According to [Pap08], a complex service can be defined as follows:

Definition 2.3 [Complex Service]. *Complex (or composite) services typically involve the assembly and invocation of many pre-existing services possibly found in diverse enterprises to complete a multi-step business interaction.*

Complex services combine the functionality and capabilities of modularized service components (which themselves can be utility, elementary or complex services) by sequential composition in order to generate added value. To illustrate the idea of complex services this section provides exemplary business cases from the enterprise sector which are based on current market information.

Example 2.1 [Complex Service: Payment Processing]. *Consider a manager of a mid-size company that distributes flowers over the Internet. As payment processing is not a core competency of the company, the board decides on the integration of third-party services* *into existing business processes in order to decrease the costs of operation and maintenance. Figure 2.3 shows the overall business scenario and in detail the payment processing complex service that is intended to be replaced by a third-party service from external providers.*

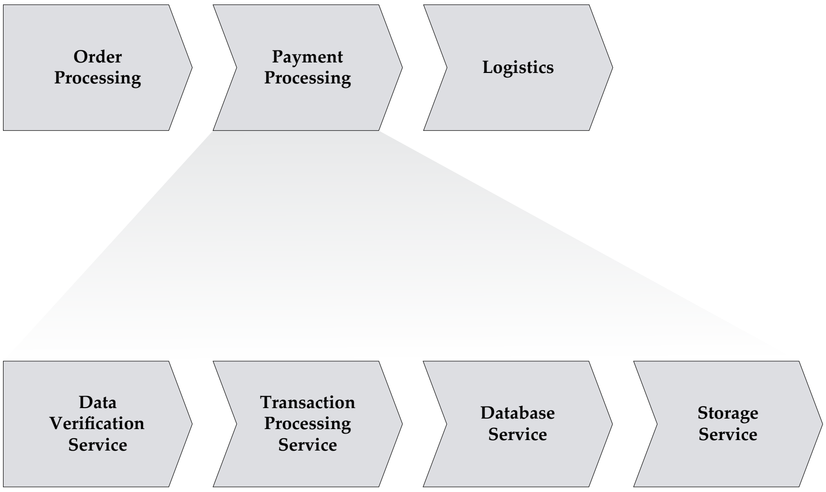


Figure 2.3: Business scenario integrating a payment processing service.

Focusing on the payment processing complex service and necessary components, the diagram in Figure 5.1 sketches an excerpt of the service components of an exemplary complex service that provides payment processing functionality.

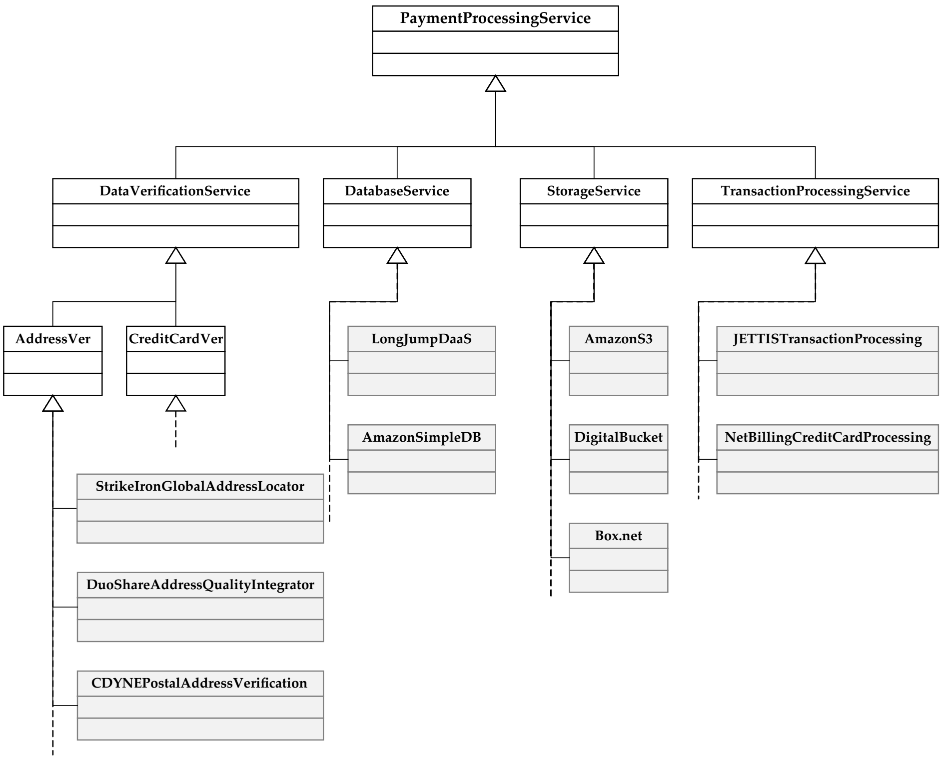


Figure 2.4: Payment processing service (static view).

The PaymentProcessingService *facilitates service components from Strike Iron*[6]*, Duo Share*[7] *and CDYNE*[8] *to verify the customer's address and credit card information. Customer data is stored and managed using a* StorageService *and a* DataBaseService *from third-parties. Sample services from decentralized storage providers are Amazon S3*[9]*, Digital Bucket*[10] *and Box.net*[11]*. Services for organizing and managing customer data are Amazon Simple DB*[12] *and Long Jump DaaS*[13]*. The actual execution of the financial transaction through the* TransactionProcessingService *is provided by JETTIS Transaction Processing*[14] *and Net Billing Credit Card Processing*[15]*.*

The process behavior of the payment processing complex service is depicted in Figure 2.5. Customer data is validated in the first step. After validation the actual transaction takes place and the customer's credit card account is charged by a transaction processing service. The change in state must be updated in the internal database of the company. A database service updates corresponding customer data that is stored using a decentralized storage service.

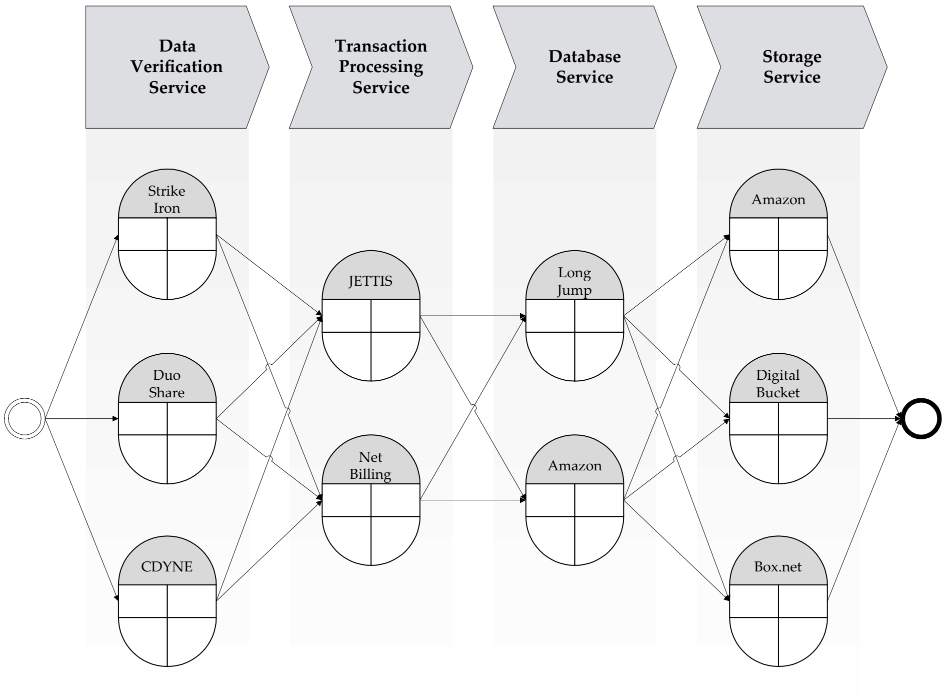


Figure 2.5: Payment processing service (dynamic view).

[6] `http://strikeiron.com/`
[7] `http://duoshare.com/`
[8] `http://cdyne.com/`
[9] `http://aws.amazon.com/s3/`
[10] `http://digitalbucket.net/`
[11] `http://box.net/`
[12] `http://aws.amazon.com/simpledb/`
[13] `http://longjump.com/daas/`
[14] `http://jettis.com/`
[15] `http://netbilling.com/`

For each step of the complex service there is a potential pool of suitable candidates to fulfill required business transaction. The result of each transaction is passed to the successor service. In order to successfully instantiate the complex service the overall transaction requires a service candidate from each pool.

Example 2.1 shows that core service competencies can be leveraged by procuring complex services from third party providers to close competency gaps in business processes. The granularity of complex services ranges from services that are parts of a business process to services that cover whole business scenarios as illustrated in the following example.

Example 2.2. *To further illustrate the idea of a complex service a business scenario which is actually delivered to customers as part of SAP's BusinessByDesign*[16] *is introduced exemplarily. The scenario consists of modular service components that can be provided by decentralized service providers. The integration scenario "Service Request and Order Management" (cp. Figure 2.6) describes operational processes in a customer service based on service requests, service orders and service confirmations. From an end-to-end perspective the scenario includes the integration into related applications such as logistics planning and execution, invoicing and payment, as well as financial accounting.*

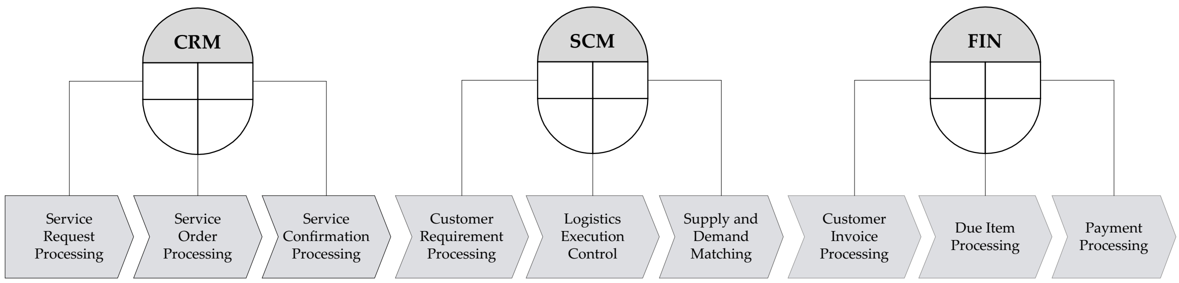


Figure 2.6: Business scenario "Service Request and Order Management" (SROM).

The complex service is formed by decentralized service providers that contribute to the achievement of an overall goal. In the presented scenario this goal is the flawless execution of a business scenario in order to provide defined functionality to the customer. Many service providers offer differentiated and specialized services covering various types of functionality within the complex service. They provide service components regarding customer relationship management (CRM), supply chain management (SCM) and finance (FIN). In this scenario the functionality of each component can be modularized and therefore performed by different software-as-a-service (SaaS) providers as depicted in Table 2.2.

The rapid growth of the number of on-demand service providers shows the high degree of innovation and market penetration as a result of service modularization. Service providers offer specialized services and concentrate on their core competencies. Each service provider

[16] `http://www.sap.com/solutions/sme/businessbydesign/`

Table 2.2: SaaS providers for CRM, SCM and FIN components of the business scenario SROM.

CRM	SCM	FIN
Salesforce `http://salesforce.com/`	GXS `http://gxs.com/`	Cashview `http://cashview.com/`
Rightnow `http://rightnow.com/`	7Hills `http://7hillsbiz.com/`	Opsource `http://opsource.net/`
Oracle `http://oracle.com/crmondemand/`	Intacct `http://intacct.com/`	
SAP `http://www.sap.com/solutions/sme/businessbydesign/`		

is responsible for a certain part of the overall functionality, which consequently spreads the risk of an erroneous business process over all contributing service providers. Furthermore, they partly grant access to their own resources thus supporting the realization of the overall business scenario.

2.1.3 Service-Oriented Architectures

This section introduces fundamentals and basic concepts of service-oriented architectures with a focus on technologies and definitions that serve as a basis for the remainder of this thesis. In Section 2.1.3.1, service-oriented architecture as a paradigm for organizing distributed services that are under the control of different domains is introduced. The section provides a definition of the service-oriented architecture concept and introduces its key principles. The concept of Web services as the most prominent example of a technology that leverages the strength of service-oriented architectures is presented in Section 2.1.3.2. The section guides through the Web service technology stack and state-of-the-art specifications and standards. It is well-known that the main value generated by a service activity is determined by its quality characteristics and their manifestation at run-time. Hence, Section 2.1.3.3 introduces the concept of quality of service (QoS), relevant factors in the context of Web services and how QoS guarantees can be formulated in contracts, i.e. service level agreements. Contracts defining QoS aspects provide the legal basis for the market-based trade of services as a special form of coordination. Thus, technologies and concepts for the coordination of Web services are introduced in Section 2.1.3.4 that provide means for organizing dependencies among distributed service activities that have to be governed to achieve an overall outcome.

2.1.3.1 Basic Concepts

Service-oriented architectures (SOAs) have gained a lot of momentum over the last years. SOA is a paradigm to organize distributed capabilities possibly under the control of different domains. The paradigm itself and its concrete implementations are fundamental for the development, production, innovation and provision of services via electronic channels. Technology that is based on the SOA principle can be seen as the enabler technology for service-oriented computing. Definitions of service-oriented architectures and related concepts are based on the OASIS Reference Model for Service Oriented Architectures [MLM$^+$06].

The main goal of service-oriented architectures is the composition of complex applications out of loosely-coupled service components that provide specific well-defined functionality. Service components are designed to live independently of the application they are part of and are therefore reusable and recomposable in different application contexts [Ley03]. In order to illustrate the idea of the flexible composition of loosely-coupled service components, the concept of a service and its interaction with central roles in the context of service-oriented architectures have to be elaborated in detail.

Relevant services in the context of service-oriented architectures are a subset of e-services as defined in Section 2.1.1.3. These types of electronic services are called *software services*. Software services are self-describing software components that provide certain capabilities through a programmatic interface via electronic networks such as the Internet. A *service interface* publishes the service's signature describing input and output parameters as well as message types. The objectives of a service are defined through its *capabilities*, which are acts or performances that solve problems of an economic unit. They state the conceptual purpose and expected result of the service by using terms or concepts defined in an application-specific taxonomy [PG03]. Narrowing down Definition 2.1, capabilities are provided through a software service by a *service provider* and consumed by a *service requester* in order to fulfill certain needs. Software services expose three major properties that are essential for the SOA paradigm:

- The programmatic interface of the service is platform-independent.
- The service can be dynamically located and invoked.
- The service maintains its own state (self-contained).

By means of a well-defined platform independent interface, the service can be consumed from anywhere, on any operating system and in any programming language. The service can be discovered by means of a look-up mechanism facilitating a service registry. In any state of its lifestyle the service manages its own state independently. Compromising this information the definition of software services is the following:

Definition 2.4 [SOFTWARE SERVICE]. *A software service is a self-describing, self-contained mechanism that enables the access to certain capabilities of an encapsulated software component via an electronic network by means of a well-defined platform-independent programmatic interface. A software service is an open component that can be dynamically located, bound and invoked.*

The definition at hand is more restrictive then Definition 2.2 because it requires the existence of a well-defined platform-independent programmatic interface[17]. An example of a software service would be a credit card verification service accessible over the Internet that verifies credit cards at a central authority based on the card number provided through the service's interface. In contrary a Web blog might not be considered to be a software service according to Definition 2.4 as it does not expose a well-defined programmatic interface in the narrow sense.

In the context of service-oriented architectures there are three primary operations to manage the interaction between the provider and requester roles as depicted in Figure 2.7. These are the *publication* of the service descriptions at a service registry by the service provider, *finding* of the service descriptions, *binding* and *execution* of the services based on their description by the service requester [Pap08].

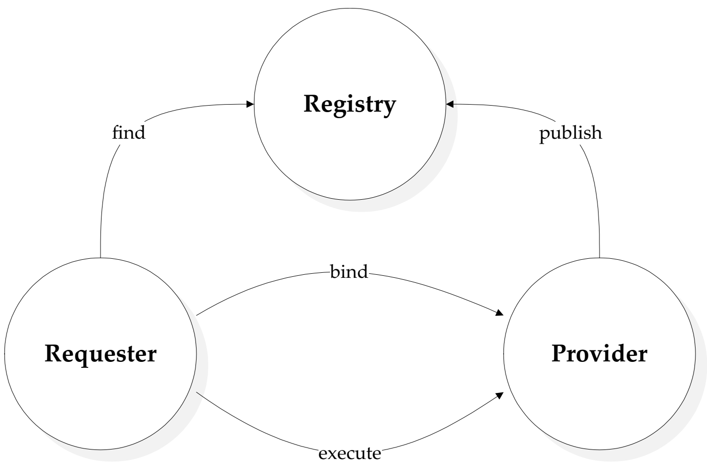


Figure 2.7: Roles and primary operations in service-oriented architectures.

Publishing a service at a service registry mainly consists of two steps. The first step is to describe the service at hand, that is, a *description* of its interface and usage conditions. The second step is the actual *registration* of the service in order to

[17]For the reader's convenience, the terms *software service* and *service* are from now on used interchangeably.

facilitate discovery and reusability by service requesters. The finding of a service involves two steps as well: The first step is to create a description in the form of a *query* that defines criteria and search terms concretizing the service that is needed by the service requester. The second step, is the *selection* of the set of services retrieved from the discovery agency. Criteria defined in the query consist of the type of service that is needed, quality aspects and other technical as well as non-functional service characteristics. The query is executed against the data set stored in the service registry and a subset of services that meet the criterions in the search query are retrieved. In the second step the service requester has to chose from the set of discovered services either statically at design-time or automatically bound at run-time. Binding and invocation are the most important operations in service-oriented architectures. Once a service is chosen either statically or dynamically, the service requester and the service provider agree to a well-defined and unambiguous *contract* that describes the service at hand and corresponding service level agreements. The invocation can either be performed *directly* by the service requester using the technical service description from the registry or via a *mediation* through the registry.

Having defined services, related concepts, roles and primary operations in the context of service-oriented architectures, the paradigm itself, its main goals and its key principles can be defined

Definition 2.5 [SERVICE-ORIENTED ARCHITECTURE]. *A service-oriented architecture is an architectural design paradigm to structure, utilize and compose distributed interoperable software services that are under the control of decentralized ownership domains in order to realize distributed applications.*

In order to achieve defined purposes the SOA paradigm relies on the following key principles.

Loose-coupling The term *coupling* refers to the degree of dependency between two systems. Therefore, loosely-coupled services can interact more freely as they do not need to know the location, behavior, implementation or any other details of communication partners. Systems that are designed in a loosely-coupled manner are mostly based on asynchronous or event-driven interaction schemes instead of synchronous communication [Pap08]. A loosely-coupled design allows for the flexible restructuring of processes and application logic without having to touch the internal structure of the services involved as they live independently within a service-oriented architecture [Bur04].

Interoperability A main benefit of service-oriented architectures is the heterogeneity of services that can be integrated in a distributed system. This diversity and continuous evolution of services during their lifecycle implies a high complexity to enable a seamless communication between services without manual adaption, i.e *interoperability*. The high degree of standardized formalisms and protocols in service-oriented architectures are key concepts to achieve the desired interoperability of distributed services.

Reusability As services in a service-oriented architecture are self-contained, loosely-coupled and not bound to a concrete system, they can be reused in different application contexts. Due to reusability, the number of redundant components in a service-oriented architecture is generally much lower compared to traditional systems. This results in a lower effort for change management and maintenance in service-oriented architectures.

Discoverability In order to reuse services in a service-oriented architecture, a potential consumer or developer must be able to find the service that matches the specified requirements. Discoverability is mostly realized by a service repository that entails services including their description to enable their search and usage. The process of service discovery can either be performed manually by consumers or automatically by the system.

The key principles of service-oriented architectures are pursued and enabled by the architectural design through the encapsulation of *infrastructure*, *application logic*, *services* and *business processes* in a transparent manner. Figure 2.8 schematically illustrates the architectural layers of a SOA as well as their interactions.

The *infrastructure layer* comprises physical resources providing computing power, storage, memory and bandwidth. Encapsulation and flexible resource provisioning is achieved by the adoption of virtualization technologies that allow for the dynamic instantiation and migration of virtual resource environments independent from their physical hosting location [BDF+03]. Virtualization is an important step towards a service enablement of physical resources, which fosters a service-oriented management of hardware units.

Above the virtualized infrastructure is the *application logic layer*, which entails applications and application systems that provide the actual functionality in the form of software components. These systems are a mixture of up-to-date application systems and old legacy systems. Applications in the application logic layer are enhanced by service definitions to enable encapsulation and abstraction in order to be manageable in a service-oriented context.

The application logic layer is abstracted by services in the *service layer*. They encapsulate functionality in a self-describing, self-contained and loosely-coupled manner and provide access through well-defined interfaces. The *service bus* is the main

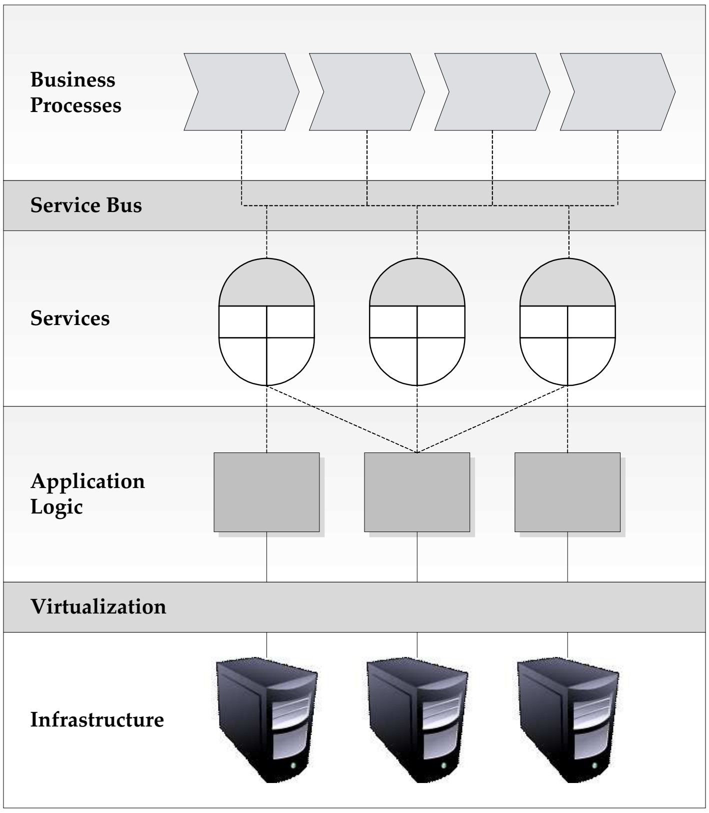


Figure 2.8: SOA layers.

component of a service-oriented architecture. It functions as the connecting element between the set of services providing loosely-coupled functionality and business processes reflecting organizational criterions and real-world business procedures. The service bus enables the retrieval, provision and binding of services [Ley03] while supporting standards to facilitate distributed communication and message exchange between services.

2.1.3.2 Web Services

Over the last decade the Web has evolved from a content- or document-oriented environment to a service-centric environment. This is due to the rise of the concept of Web services. The term Web service in general does not per se imply a concrete form of realization. Web services are a way to expose functionality in a standardized manner that is accessible over the Web in order to realize complex distributed applications. The use of standard Web technology reduces heterogeneity and enables the reuse and integration of distributed functionality independent of platforms and programming models. In contrary to traditional inter-company middleware that is centrally organized and controlled by a single company, the Web service paradigm allows for the integration of globally distributed services across organizational boundaries.

A huge body of work has been done defining Web services. The most prominent definitions range from a very generic perspective to a strict and language-oriented view. Nevertheless, only focusing on the aspect that Web services are applications that are accessible over the Web to other applications [ABC+02] is certainly not practical. In contrary, the notion of the World Wide Web consortium (W3C) [AGB+04] is much stricter as it limits Web services to those services that expose interfaces that are described using the eXtensible Markup Language (XML) [BPSM+06]. The W3C defines a Web service as "[...] a software system identified by a URI [BLFM98], whose public interfaces and bindings are defined and described using XML. Its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using XML based messages conveyed by Internet protocols." This definition excludes Web services that exchange messages in a more lightweight manner facilitating formatting standards that in contrary to XML reduce payload. In order to include these types of Web services the definition by W3C has to be relaxed regarding language limitations.

Definition 2.6 [WEB SERVICE]. *A Web service is a software service identified by a URI [BLFM98] that exposes a public interfaces, based on Internet standards. A Web service can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using Internet standard based messages conveyed by Internet protocols.*

Conceptually Web services can be divided in two main categories depending on the architectural style used for their realization, i.e. *RESTful Web services* [18] and *Big Web services* . [PZL08].

Recently, RESTful Web services have increased attention not only because of their usage in the context of Web 2.0[19], service mashups and situational applications, but also because of the presumed simplicity and their lightweight character.

RESTful Web services are based on an architectural style that is used for realizing distributed hypermedia information systems (e.g. the Web). Messages are transported via the HTTP protocol without the need for an envelope on-top such as SOAP that generates extra XML payload. RESTful Web services expose unique document processing interfaces. The signature consists of the *scoping information* specified by a URI (e.g. "`/reports/open-bugs/`") and *method information* specified in the HTTP header (e.g. `GET`, `HEAD`, `PUT`, `DELETE`). Due to the strict and exclusive use of standardized HTTP methods valuable properties are retained, i.e. *safety* and *idempotence*. Safety refers to the property that – assuming a correct implementa-

[18] The term Representational State Transfer (REST) was firstly introduced in [Fie00]

[19] cp. `http://programmableweb.org/apis/`

tion of a RESTful Web service– the execution of HTTP methods `GET` and `HEAD` does not change the state of the corresponding service. Idempotence is a property of an operation that states that the result of an operation is independent of the number of executions[20]. It is important that HTTP methods such as `PUT` and `DELETE` are idempotent operations due to the unreliable nature of the Web and the uncertainty of a successful method execution. Therefore, it is possible to invoke the same method multiple times without having to care about the implications of the repeated calls. Furthermore, RESTful Web services are *addressable*, *connected* and *stateless* meaning that they can be uniquely identified, they mostly point to other services that make sense in a certain context, and any information that is necessary to understand a message is enclosed in the HTTP message.

Up to now the lightweight nature of RESTful Web services and the lack of expensive service descriptions have been regarded as feature of the approach especially in the context of service mashups and situational applications. However, as applications become more complex and the number of services grows, the lack of a service description becomes increasingly problematic (see also discussion in [PZL08, Pau08]). Therefore, first approaches for annotating RESTful Web services have been proposed. Similar to the approach used in SAWSDL [FL07] for WSDL-based services, SA-REST [SGL07, LGS07] can be used to attach model reference annotations to HTML using RDFa [AB08]. It can thus be used to annotate RESTful Web services.

Recently, many service providers claim to offer RESTful Web services but mostly violate important properties that are outlined in this section [RR07]. Prominent examples of service providers that offer correctly implemented RESTful Web services are Amazon and Yahoo!. Amazon offers storage capacity through its Simple Storage Service (S3)[21] that is fully accessible and manageable in the manner of REST. Most of Yahoo!'s Web services[22] are also available as RESTful Web services.[23]

To pursue SOA principles such as interoperability and platform independence, Web service technology is based on standardized Internet protocols and description languages to allow for the interoperable automation of distributed applications without the need for human intervention. Thus, Web services are not built in a monolithic manner but rather founded on a stack of complementary standards encapsulating several functional layers as illustrated in Figure 2.9.

[20] e.g. the function $f(x) = 1 \cdot x$ is idempotent as $f(f(x)) = f(x)$ and in general $f \circ \cdots \circ f = f$

[21] `http://aws.amazon.com/s3/`

[22] `http://developer.yahoo.com/`

[23] Note that also most static Web sites are accessible and manageable as RESTful Web services [RR07].

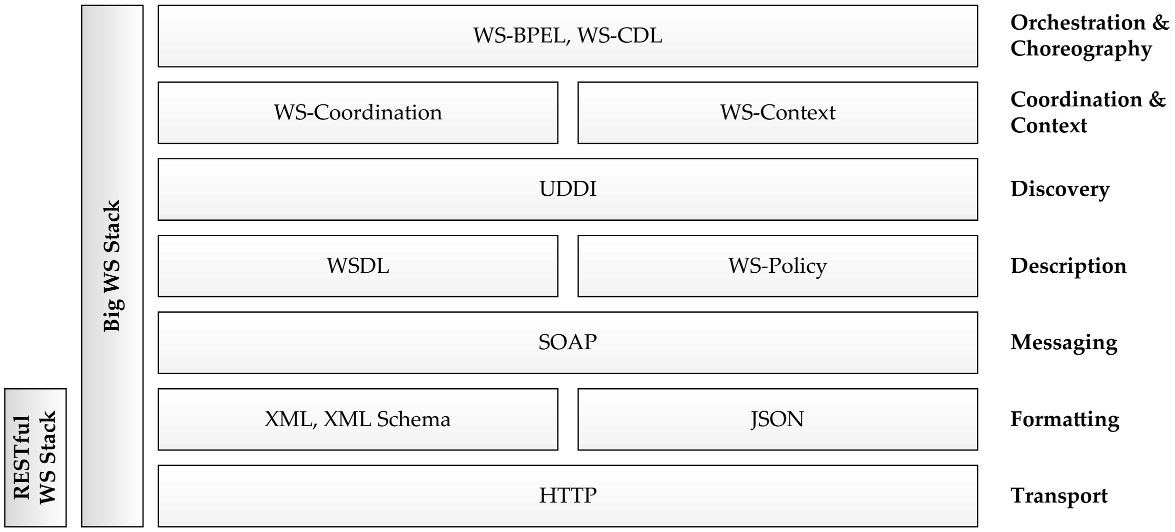


Figure 2.9: Web service technology stack.

Due to this design principle, new standards in the context of Web services emerge quickly as they are developed on-top of existing functionality[24].

Transport

Web services facilitate basic Internet infrastructure technology such as the Hypertext Transfer Protocol (HTTP) [FGM+99], the Simple Mail Transfer Protocol (SMTP) or the File Transfer Protocol (FTP). The HTTP protocol enables transportation, ensures almost universal reach and support and is the most prominent transport protocol used by Web servers and browsers. It allows for the stateless interoperability of distributed, collaborative information systems. In order to enable the unique addressing for transportation, resources on the Web are identified using a Unique Resource Identifier (URI) [BLFM98].

Formatting

Messages that are exchanged via the transport layer are structured based on formatting standards. The most prominent example that is widely used is the eXtensible Markup Language (XML) [BPSM+06] but there are also lightweight formats mainly pushed through Web 2.0 technology such as the JavaScript Object Notation (JSON) [Cro06].

[24]The interested reader is referred to `http://www.innoq.com/soa/ws-standards/poster/` for a comprehensive overview of state-of-the-art Web service standards.

Messaging

Message exchange in distributed environments such as the Web have to be organized using standardized specifications. Specifications for the exchange of messages are developed on top of the transport layer and protocols such as HTTP, SMTP or FTP and function as an envelope that defines how messages should be exchanged between communication partners. A well-established framework for Web service information exchange is the Simple Object Access Protocol (SOAP) [BEK$^+$00]. SOAP is a further development of XML-RPC [Win99]. It is a network protocol that enables the XML-based message exchange between distributed software systems in the manner of a Remote Procedure Call (RPC) architectural style. It specifies how messages should be structured, formatted and interpreted independent of semantics and application-specific information. SOAP messaging can be enhanced by complementary Web service standards such as WS-Security [NKMHB06] to allow for integrity and confidentiality of information exchange procedures.

Description

The publish-find-bind-execute paradigm as illustrated in Figure 2.7 allows service providers to publish services at a central registry, that can then be discovered, bound and executed by service requesters. In order to enable such roles, operations and interactions in a service-oriented architecture, Web services need to be described in a consistent manner. Thus, only if a service requester is able to gather all necessary information on a service's interface and the type and structure of the messages being exchanged, services can be assembled and composed into value-added complex services that expose business functionality. Service description reduces the need for a common understanding and custom programming and is a key driver of loosely-coupling in service-oriented architectures. It is a machine-understandable description of a service's structure, operational characteristics and non-functional properties [Pap08].

The Web service Description Language (WSDL) [CCMW01] is widely used especially for the description of SOAP-enabled Web services. Generally, WSDL describes what a service does, that is, the operations the service provides, where it is located, and how to invoke it. WSDL is based on XML consisting of an *abstract part* and a *concrete part*. A service's *interface* consisting of operations and corresponding *data types* of input and output messages are specified in the abstract part by means of a *port type*. The concrete part binds the abstract port type to a message encoding protocol and adds a concrete *end point* address to each port type.

Although the Web is mainly based on HTTP as the transport protocol, WSDL and SOAP hardly use the features of HTTP at all (e.g. SOAP only uses HTTP response codes "200" and "500"). Nevertheless, it is also possible to leverage the power of HTTP by facilitating all features originally provided by HTTP 1.1 in order to describe Web services. Exemplary, the Web Application Description Language (WADL) [Had06] describes resources or services that respond to HTTP's uniform interface by grouping their operations into a single end point.

Discovery

The full potential of reusable loosely-coupled Web services can only be utilized if there exist mechanisms that enable service providers to publish information on the capabilities of their service offers and how to access and use them. Service requesters should be able to discover adequate services that match their requirements and the necessary information to bind and invoke them. Service discovery is the process of querying a service registry and retrieving published Web service descriptions that specify the Web service's properties, its capabilities and how to properly interact with it. The discovery process can be differentiated in two basic types, *static* and *dynamic* discovery [GSB+02]. Static discovery queries a registry and receives necessary information at design-time while dynamic discovery proceeds these steps during run-time. After having retrieved a set of Web services that match the query criteria, the service requester has to select a service to be invoked.

The Universal Description, Discovery, and Integration (UDDI) [CHvRR04] is a framework representing a central registry to publish and discover Web services in a global and open manner. Information provided by a UDDI registry is threefold. *White pages* provide contact information on companies that publish their services in a UDDI registry. *Yellow pages* provide the classification of information based on standardized industry taxonomies. *Green pages* accommodate service requesters with necessary technical information regarding exposed Web services.

Coordination & Context

In distributed environments with decentralized service providers, the coordination of transactions is a fundamental concept in order to govern interactions of participants to achieve a desired outcome. A detailed introduction to the WS-Coordination specification [NRFJ07] is provided in Section 2.1.3.4.

Orchestration & Choreography

Generating value from a business perspective is achieved by loosely-coupled Web services that are composed into complex applications as the main objective of the SOA paradigm. There are essentially two types of service composition as depicted in Figure 2.10 that have to be differentiated.

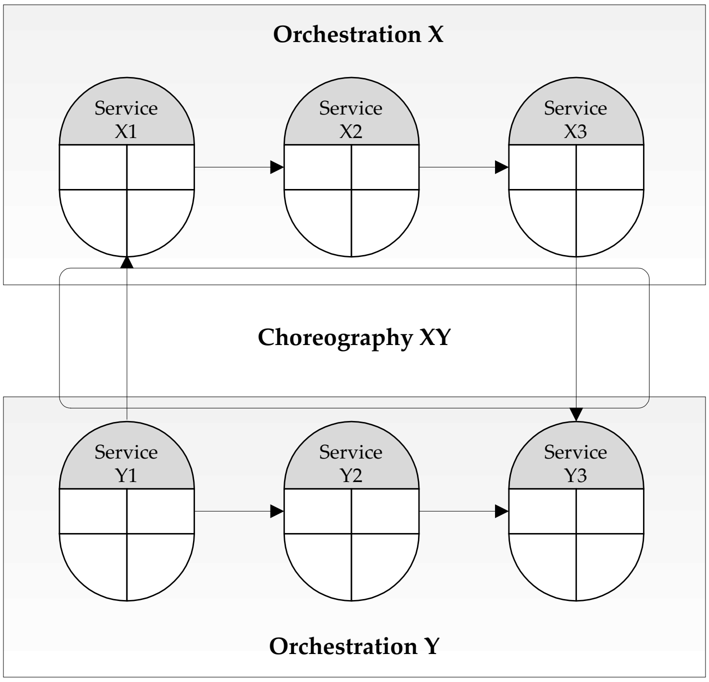


Figure 2.10: Service orchestration versus service choreography.

Service orchestration completely describes the composition procedure of internal or external services controlled by a central element. Each service that is part of an orchestration has a limited scope that restricts its decision radius. Activities that run internally within a service component are transparent and hidden to other services. A specification of a service orchestration describes service components, conditional dependencies and alternatives within a composition.

Service choreography is the description of a protocol that defines rules for the interaction between service components and their function within the composition. There is no central element to control and assure a correct behavior of each service component and the composition itself. A service choreography focuses on the exchange of messages between services components and the definition of necessary protocols.

In short the difference between service orchestration and choreography can be narrowed down as follows:

Orchestration defines procedure, choreography defines protocol.

From a business perspective the goal of a service-oriented architecture is to provide the architectural design that enables a flexible customization of business processes in order to align IT and business. As business processes are volatile and change frequently, service-oriented architectures allow for an ad-hoc adaption of business processes according to situational needs and changing market requirements. The final process flow is instantiated at run-time, which enables just-in-time reflection of real-world business processes in a way that IT aligns with business and not vice versa.

Web service standards such as SOAP, WSDL and UDDI provide means for the realization of relatively simple Web services that fulfill limited tasks by providing adequate functionality. Extending the vision of a loosely-coupled service-oriented architecture that overcomes physical boundaries and enables an inter- and intra-organizational integration of business functionality requires standardized formalisms to describe Web service orchestration into business processes and their choreography in a seamless manner. A *Web service business process* describes how operations are composed out of a set of potential Web services, how they interact, share information and what partners are involved in order to create the required business value.

The Web Service Business Process Execution Language (WS-BPEL) [AAA+07] provides a standardized description language for specifying business processes composed of operations that are exposed by WSDL-based Web services. Hence, WS-BPEL supports service composition models, recursive composition, separation of composability of concerns, stateful conversation and lifecycle management, and recoverability properties [WCL+05]. WS-BPEL mainly contains five sections, i.e. the *message flow*, the *control flow*, the *data flow*, the *process orchestration*, and the *fault and exception handling* section as illustrated in Listing 2.1.

The selection of services for composition and for the definition of relationships among services revolves around the notion of *partner links*. WS-BPEL maintains the state of the process and control data which is stored in variables analogous to *variables* in programming languages which are specified by names and types. Partner links describe a pair of roles which exchange messages and port types that the service playing these roles has to implement. Enabling the mapping of messages to composition instances, *correlation sets* can be defined that describe how to correlate messages with concrete instances.

The component model of WS-BPEL consists of *basic* and *structured activities*. Structured activities define the actual orchestration whereas basic activities spec-

```
<process name="paymentProcessing" ...>

  <partnerLinks> ... </partnerLinks>

  <variables> ... </variables>

  <correlationSets> ... </correlationSets>

  <!- Activities -->

  <faultHandlers> ... </faultHandlers>

  <compensationHandlers> ... </compensationHandlers>

  <eventHandlers> ... </eventHandlers>

</process>
```

Listing 2.1: WS-BPEL Structure

ify the components itself and correspond to the invocation of a WSDL operation. As basic activities, WS-BPEL provides invoke activities, that invoke operations, as well as receive and reply activities which correspond to the receipt of a client's message and to the reply in response to an operation invoked. Structured activities however are capable of defining more sophisticated process logic by combining other activities (basic and structured). Constructs of structured activities are sequences, switches, picks, whiles and flows.

Providing means for exception handling, *fault handlers* define how certain exceptions should be managed. fault handlers specify a catch element which defines the fault it manages and the corresponding activity that is triggered in case an exception occurs. Combining exception handling and transactional techniques, *compensation handlers* define the logic required to undo the execution of activities as a compensation. In contrary to the try-catch-approach, event handlers continuously monitor certain events and define activities to be triggered in case that particular event occurs.

2.1.3.3 Quality of Service (QoS)

The value generated by a service is mainly embodied through intangible elements exposed at execution (cp. service characteristic C 2.4). Therefore, a service consumer expects a service to function reliably and to deliver a consistent outcome at a variety of levels, i.e. quality of service (QoS). To provision, control and assure QoS it requires not only for focus on functional properties of a service but also on non-functional aspects. The context of a service also influences its quality, which is experienced by the consumer, e.g. the partner network that comes with a service, its reputation in certain communities or advertisement campaigns promoting the service. From an economic perspective, QoS is the most important characteristic that differentiates service offerings and leverages market advantage, as price competition is tough due to low variable costs of service provisioning. Thus, QoS is the key criterion to keep the business side competitive as it has serious implications on the provider and consumer side [Pap08]. The provision of services with a defined QoS over electronic networks such as the Web is challenging due to issues like infrastructure problems, unpredictable reliability, low performance of Web protocols and many more. In addition, the distributed nature of Web service environments and their high degree of complexity requires a comprehensive description of Web service quality characteristics, both functional and non-functional. The main aspects of QoS in a Web service context, which are partly derived from [MN02, ZBD+03, LNZ04, CSM+04, Pap08] are as follows:

Availability Service *availability* is the likelihood of absence of downtime, i.e. the probability that a service is available for invocation. Small values indicate an unpredictability of the service to be accessible at a certain point of time. This probability can be estimated by incorporating historical data on a service's downtime. The ration of observed average downtime and total time of potential availability results in an estimated probability of unavailability for the future, whereas the probability of the complementary event reflects an estimated probability of availability of a service.

Reliability Service *reliability* refers to the characteristic to function correctly and consistently, i.e to produce the desired outcome or result. This is usually expressed in transaction failures over a defined period of time. It can be be measured using historical data of previous invocations and a corresponding successful delivery.

Scalability The ability to service requests independently of volume is referred to as the *scalability* of a service. Scalability is important in periods with high peaks of demand with uncertain occurrence and hardly predictable patterns.

Performance The service quality aspect *performance* consists of two parts, *throughput* and *latency*. A service's throughput refers to the number of requests that can be served at a defined time period. Latency of a service is the time between sending a request and receiving the outcome or result. This means that high throughput and low latency characterize a service with a high degree of performance.

Security As Web services are usually provided over the Internet, security is an important issue for service providers and consumers. Especially in order to represent long-lived mission-critical business transactions that involve private business information, Web services must fulfill serious security requirements such as access control (authentication, authorization), confidentiality, and integrity of information.

Reputation The reputation of a service is a measurement of its trustworthiness. The value creation of a service is mostly dominated by intangible elements and is therefore subjective to the individual that experiences a service's outcome. As the sum of individual experiences is a suitable indicator for service quality, reputation is an important aspect that takes consumers' experiences and opinions into account[25].

An agreement between service provider and service consumer about the QoS to be delivered must be founded on a legal basis, i.e by specifying a service level agreement. A service level agreement is a contract that defines mutual understandings and expectations of a service between service provider and service consumer [JMS02]. It defines service characteristics and the quality to be delivered by the provider and monetary penalties in case of non-performance. Such a contract represents a guarantee for the service consumer, which assures the delivery of the defined quality or an adequate charge-back mechanism.

Depending on the frequency by which a service level agreement can be redefined and adapted according to changed requirements or conditions, two types of service level agreements can be differentiated, *static* and *dynamic* service level agreements. Static service level agreements generally remain unchanged for a long period of time or multiple service time intervals. The quality of situational and short-termed Web services is covered by dynamic service level agreements that change from period to period. This type of service level agreement is inevitable in highly dynamic environments where Web services are composed and provisioned on-demand and roles of service provider and consumer change quickly.

2.1.3.4 Web Service Coordination

Environments in which distributed units provide functionality in a loosely-coupled manner (according to the SOA paradigm) require some sort of process or set of rules to align activities in order to generate a desired outcome, i.e. they require *coordination*. The objective of coordination is to make a set of entities – either by providing incentives or establishing constraints upon them – pursue a common goal, e.g. producing a defined outcome.

[25] A star ranking mechanism is a possible solution to capture consumers' valuations for a service. An example can be found at `http://aws.amazon.com/`.

Definition 2.7 [COORDINATION]. *Coordination is managing the dependencies of activities.*[26]

Coordination can be formalized by designing adequate *mechanisms*, i.e sets of rules that govern the interaction between the various entities. Coordination is the key instrument to organize multiple activities especially in distributed environments. In the context of Web services two specifications provide frameworks to implement coordination scenarios, WS-Coordination [NRFJ07] and WS-CF [CNLP05]. This section focuses on WS-Coordination as it is a finalized standard in contrary to WS-CF, which is still a public review draft. A detailed comparison of WS-Coordination and WS-CF can be found in [LW03] and [Kra05]. WS-Coordination is based on concepts and roles that are represented by Web services. *Initiator*, *coordinator* and *participants* communicate using a common *context* that glues their interaction to the coordinated activity. The framework allows for different coordination protocols to be plugged in to coordinate domain-specific work between clients, services and participants. Work is defined as activities performed by one or more distributed parties. Examples for specific transaction protocols are WS-AtomicTransaction [NRLW07] and WS-Business Activity [NRFL07]. WS-AtomicTransaction specifies a rudimentary ACID[27] transaction protocol focusing on ad-hoc short-term transactions in a general manner. In contrast, WS-BusinessActivity defines transactions with relaxed ACID properties with the purpose to coordinate long-term business transactions.

The process of coordination and the roles involved according to the WS-Coordination specification are depicted in Figure 2.11. The sequence diagram illustrates the main phases *activation*, *registration*, *invitation* and *communication*.

Activation The WS-Coordination framework exposes an *activation service* that is responsible for the creation of specific coordinator instances with concrete *protocols* and associated *context*. To start a coordination process, the initiator sends a `CreateCoordinationContext` message to the endpoint of the activation service in an asynchronous manner. The coordinator either replies with a `CreateCoordinationContextResponse` message or an error message. A `CreateCoordinationContext` message has the following structure:
The `CoordinationType` points to a uniform resource identifier that specifies the type of coordination to be used in the coordination process (e.g. WS-AT, WS-BA). `wsu:Expires` is an optional argument that defines a time-out value for the corresponding coordination context. The semantic of this argument

[26]The definition of coordination is based on [MC94] and is consistent with literature from organization theory [Gal73]

[27]ACID stands for *atomicity*, *consistency*, *isolation* and *durability*, which are properties that guarantee a reliable transaction.

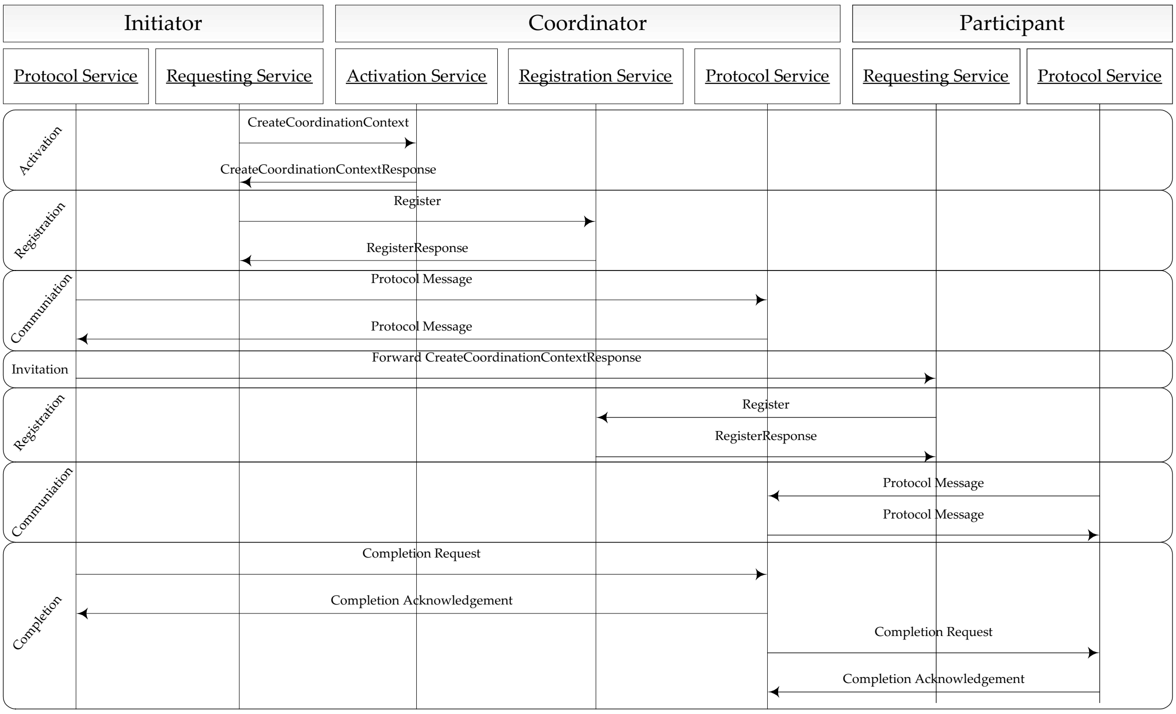


Figure 2.11: WS-Coordination sequence diagram.

```
<CreateCoordinationContext ...>
   <CoordinationType> ... </CoordinationType>
   <wsu:Expires> ... </wsu:Expires>
   <CurrentContext> ... </CurrentContext>
   ...
</CreateCoordinationContext>
```

Listing 2.2: Structure of a `CreateCoordinationContext` Message

depends on the coordination type used. The `CurrentContext` argument is also optional and can be used to hand over an existing context (activity import). In this case, the coordinator participates at the running activity instead of creating a new context.

In case the activation is successful, the coordinator replies asynchronously with a `CreateCoordinationContextResponse` message that is structured as follows:
The `CoordinationContext` consists of a unique `Identifier` that guar-

```
<CreateCoordinationContextResponse ...>
   <CoordinationContext>
      <Identifier> ... </Identifier>
      <CoordinationType> ... </CoordinationType>
      <RegistrationService> ... </RegistrationService>
   </CoordinationContext>
</CreateCoordinationContextResponse>
```

Listing 2.3: Structure of a `CreateCoordinationContextResponse` Message

antees a well-defined mapping from message to activity. The argument

`CoordinationType` defines the type of coordination. The actual endpoint reference to the *registration service* exposed by the coordinator is specified using WS-Addressing [BCC+04] in the `RegistrationService` section. The registration service is responsible for handling registration requests from participants that intent to participate in the activity.

Registration Once a coordinator has been activated by the activation service, a registration service is exposed that allows for participants to register for being part of the activity and to send – if this is supported by the coordination protocol – and receive protocol messages. Via the `CoordinationContextRespond` message, the initiator receives and endpoint reference to the registration service. By sending a `Register` message to this uniform resource identifier, the initiator's registration is confirmed by the coordinator with a `RegisterRespond` message. The `RegisterRespond` message contains and endpoint reference to the *protocol service* of the coordinator that is responsible for managing the communication between participating roles. A `Register` message is structured as follows:

The `ProtocolIdentifier` argument specifies the coordination protocol that

```
<Register ...>
   <ProtocolIdentifier> ... </ProtocolIdentifier>
   <ParticipantProtocolService> ... </ParticipantProtocolService>
   ...
</Register>
```

Listing 2.4: Structure of a `Register` Message

is supported by the chosen coordination type of the coordination context. An endpoint reference to the protocol service of the initiator is defined in the `ParticipantProtocolService` section as the destination for further communication. In case of a successful registration, the coordinator sends a `RegisterRespond` message to the initiator that is structured as follows:

The registration response message contains the endpoint reference to the pro-

```
<RegisterResponse ...>
   <CoordinationProtocolService> ... </CoordinationProtocolService>
   ...
</RegisterRepsonse>
```

Listing 2.5: Structure of a `RegisterResond` Message

tocol service of the coordinator in the `CoordinationProtocolService` section.

Invitation Recall, the `CreateCoordinationContextResponse` message contains the endpoint reference to the registration service of the coordinator and can therefore be used as an invitation or call for participation. By forwarding the message to potential participants they obtain the possibility to register for the activity at hand. Although the initiator normally invites further participants, one can think of multiple scenarios with different roles to be the inviting party in the process. The coordinator can step into the role of pushing the invitation process using a UDDI registry to find suitable participants. It is also possible to reverse the roles in such a lookup scenario, meaning that potential participants are proactively searching for suitable coordination services. Potential participants could also subscribe to a notification service – analogue to the observer design pattern – using the WS-Notification [GNC+04] specification in order to automatically be informed if an adequate coordination service is available.

Communication Initiator and participants share common knowledge about the endpoint reference of the coordinator's protocol service. Depending on the coordination type and the activity that is realized by the coordination process, initiator and participants use the protocol service of the coordinator to exchange messages in an asynchronous manner. The registration phase also provides the coordinator with the necessary address information about the active parties to be able to respond to incoming messages.

Completion Termination of the coordination process is usually initiated by the initiator. The initiator sends a completion request message to the coordinator that acknowledges the request by a completion acknowledge message. The coordinator informs all registered participants by sending a completion request message. A confirmation of each registered participant is then responded as a completion acknowledge message back to coordinator.

Example 2.3 [WS-Coordination compliant reverse auction]. *To illustrate the specification of a coordination model according to the WS-Coordination framework, an auction mechanism is introduced as a special type of coordination, i.e a single item sealed bid reverse auction. There is one* buyer *that intents to procure a single good or service from multiple* sellers*. The auction conduction including the type of messages to be exchanged between the participants is specified by* auction rules *which are controlled and enforced by an* auctioneer*. The mapping between roles and entities in a reverse auction and a coordination model is depicted in Figure 2.12.*

The buyer starts the auction by announcing a request *for the desired good or service. The auctioneer receives sealed* offer *bids from the sellers by a public deadline. After the deadline the winner determination is performed by the auctioneer, the good or service is transferred and the winning seller receives its payment.*

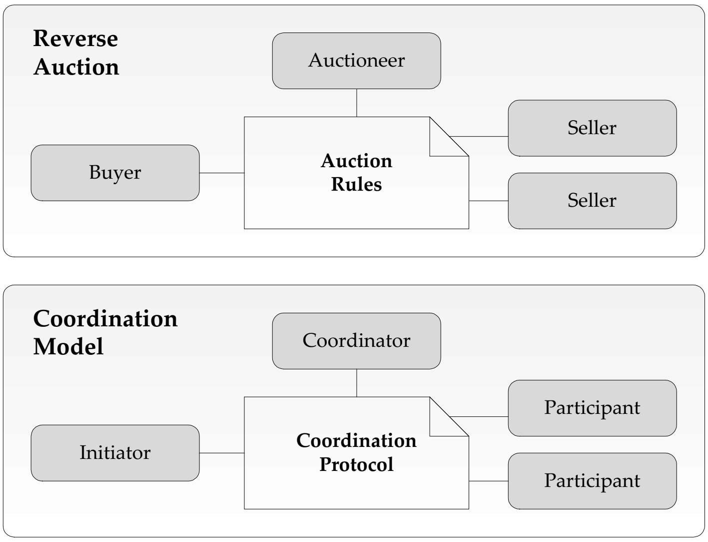


Figure 2.12: Mapping of a reverse auction to a coordination model.

Based on the WS-Coordination framework, the buyer is represented by the initiator and the sellers are instances of the participant role. The auctioneer as the coordinator is responsible for the coordination protocol, that is, the set of auction rules. The initiator starts the activation phase and receives a coordination context from the coordinator. The invitation phase is generally done by the initiator according to [NRFJ07]. Nevertheless this not practicable for the reverse auction scenario as the buyer is not necessarily responsible for the discovery and selection of potential sellers. As the WS-Coordination framework provides a generic coordination model independent of a domain-specific application logic, a tailored invitation process can be implemented on-top in order to shift responsibilities.

2.1.4 Service Value Networks and Situational Applications

Complete industries are moving from integration to *specialization*. Hierarchically organized firms that started to cooperate in firmly-coupled strategic networks with stable inter-organizational ties recently explore the benefits of exploiting more loosely-coupled configurations of legally independent firms. In theory, complex products or services can be produced by a single vertically integrated company. However, doing so, the company cannot focus on its core competencies since it has to cover the whole spectrum of the value chain. Also, it has to burden all the risks in a complex, changing and uncertain environment by itself.

2.1.4.1 Networks as a Type of Governance Form

As a consequence, *business networks* (BNs) have been proposed as the superior governance form for today's highly dynamic and complex business world [MS86]. Business networks evolve from a pool of potential horizontal as well as vertical business partnerships. In this respect they differ both from *strategic alliances*, comprising only horizontal business partners, and *supply chains*, denoting purely vertical relationships. The advantages of business networks compared to more traditional governance forms are manifold:

- Insurance against uncertainty in demand and supply.
- Balancing adaptability to highly complex tasks while maintaining control.
- Protection of business knowledge through modularization.
- Market-based forces as coordination mechanism to ensure efficiency.

A bulk of managerial and academic literature deals with variations of such business networks, whose complete characterization would be far beyond the scope of this section. In this section, *Service Value Networks (SVNs)* as a special type of business networks are identified and the differences to related organizational forms, which are to described in the following are described.

Virtual Organizations (VOs) are temporary networks of independent enterprises that bring in complementary competencies and resources for mutual benefit [DM93]. Virtual organizations stress the complementarity of firms' core competencies in the value creation process and the temporary nature of the interaction. However, virtual organizations often suffer from trust related problems and are therefore usually constituted among firms in a closed pool of known network partners.

Smart Business Networks (SBNs) are one way beyond the virtual organization framework and particularly stress the *smart* use of information and communication technology (ICT) as a facilitator to network interaction. Smartness is thereby a relative term, which refers to effectiveness and a comparative advantage through the use of ICT. Moreover, ICT is also seen as an enabler of network agility, i.e. the network's ability to "rapidly pick, plug, and play" business processes [vHV07]. Furthermore, nodes in a smart business network need to meet specific requirements in order to be ready to contribute to ad-hoc joint value creation. This modularity of potential network members allows not only for spontaneous network orchestration, but also for better protection of a firm's core competencies as compared to virtual organizations. Trust problems are thus not as severe and the smart business networks may therefore recruit members from a more open pool of potential partners. The instantiations of smart business networks are also more short-lived than those of virtual organizations. However, like in virtual organizations, the network pool itself is sustainable over time.

Business Webs are defined as "customer-centric, hetrarchical organizational forms that consisting of legally independent but economically interdependent specialized firms that co-opetitively contribute modules to a product system based on a value-enabling platform under the presence of network externalities which are supported by extensive usage of information and communication technologies." [Ste04]. Business Webs stress the internet as the primary channel for business communications [TLT00]. Moreover, the so-called "shaper-adapter configuration" is an important assumption: A shaper (i.e. a focal company or nucleus) controls the central element in a business web, while adapters (i.e. context providers) add complementary elements. A closely related field of research considers *Business Ecosystems* whose quintessence is each participant's ultimate connection to the fate of the network as a whole [IL04].

In this context, service value networks are a special type of smart business networks with features of business webs. They exhibit the crucial features of smart business networks, such as the smart use of ICT, agility, ad-hoc value creation and sustainability of the network pool. With respect to business webs, service value networks share the feature of being enabled through ubiquitously available ICT, foremost the Internet. However, service value networks are distinct to business webs because they do not follow the shaper-adapter paradigm and are rather constituted by market-based composition from an open pool of network partners.

2.1.4.2 Service Value Networks

Companies tend to engage in networked value creation, which allows participants to focus on their strengths. Partners in such ecosystem-like environments can leverage the know-how and capital assets of partners, at the same time spreading risk and sharing investment cost. Focusing on core competencies does not put constraints on the company or limit its reach. In contrary, by re-aggregating with partners, a network of companies can broaden its range of customer attraction. Especially in complex and highly dynamic industries, forming such open networks is more than an attractive strategic alternative. Service value networks bring together mutually networked, permanently changing, legally independent actors in customer centric, mostly heterarchical organizational forms in order to *create joint value for customers*. Specialized firms co-opetitively contribute modules to an overall value proposition under the presence of network externalities.

There is still only few research in the context of service value networks, especially regarding attempts to provide a definition. Service value networks are constituted by loosely-coupled formations of companies that provide modularized services while concentrating on their core competencies. These Web-enabled services expose standardized interfaces and foster an ad-hoc composition in order to jointly generate added value for customers in an on-demand fashion. This argumentation leads to the following definition:

Definition 2.8 [SERVICE VALUE NETWORK]. *Service value networks are goal-oriented business networks, which provide business value through the agile and market-based composition of complex services from a steady, but open pool of complementary as well as substitutive standardized service modules by the use of ubiquitously accessible information technology.*

To foster a fundamental understanding of the service value network concept, Figure 2.13 depicts the main components and their interdependencies in a simplified manner.

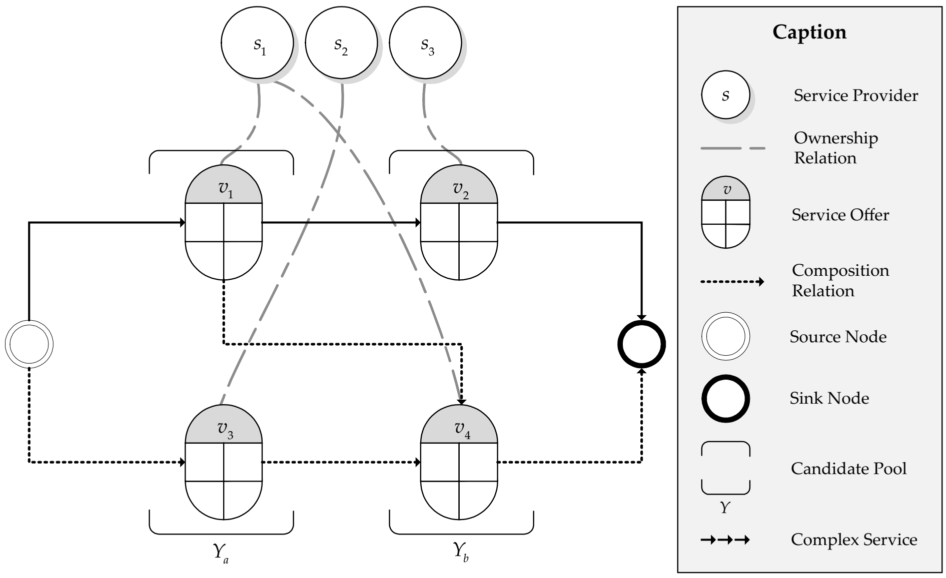


Figure 2.13: Service value network model.

A service value network consists of a set of *service providers* ($s \in S$) that supply a portfolio of *service offers* ($v \in V$) that provide specified functionality. Each service provider can own one or multiple service offers, indicated by an *ownership relation*. The example in Figure 2.13 shows a service value network with four service offers (v_1,v_2,v_3,v_4) that are owned by three service providers (s_1,s_2,s_3). Service offers that are substitutes – which provide roughly similar functionality – are clustered in *candidate pools* ($Y \in \mathcal{Y}$). A candidate pool is a set of potential service offers that are substitutes and can therefore be replaced on-demand. Service offers that are compatible, this is, they are interoperable regarding their interfaces and input and output capabilities, expose a directed *composition relation*. Service offers – clustered into candidate pools – and their connections form a graph-like structure that is directed and a-cyclic starting from a source node and ending at a sink node. Each feasible connected set of service offers within this graph is called a *path* and represents a possible instantiation of a *complex service* consisting of functionality from each candidate pool. According to the example in Figure 2.13, a complex service can be instantiated either by a composition of v_1 and v_2 or v_1 and v_4 or v_3 and v_4.

Service Providers and Service Offers The number of service providers offering various types of utility, elementary and complex services in ecosystem-like environments is constantly increasing.

Exemplarily, Amazon offers utility services based on their infrastructure as a computing and a storage service called Elastic Compute Cloud (EC2)[28] and Simple Storage Service (S3)[29] that are accessible and manageable through simple highly standardized interfaces based on REST and WSDL. In most cases, such cloud computing infrastructures are organized in a cluster-like structure facilitating virtualization technologies. Nevertheless, there are service providers that focus on offering computing on-demand through a server Grid such as the Sun Grid Computing Utility[30]. Among providing pure utility services, providers such as RightScale[31] often enrich their offerings through value-added elementary services for managing the underlying hardware (i.e. scaling, migration) that are accessible via Web front-ends.

Service providers such as StrikeIron[32] offer a comprehensive portfolio of elementary and complex Web services that provide functionality in the context of communications, customer relationship management (CRM), data enhancement, e-commerce, finance, and marketing. Especially in the financial sector, companies (e.g. Xignite[33]) sell Web services providing financial information such as real-time stock quotes, options, historical data, commodity prices, mutual funds, currency rates, and financial market indices.

Nevertheless, not only rather simple, but also complex services supporting multi-step business processes are offered modularized in an on-demand fashion. For instance, providers like salesforce.com[34] or Netsuite[35] successfully entered the business software ecosystem with their entirely Web-based on-demand customer relationship management (CRM) suites. Components offered within these suites can be dynamically composed to customized complex services. AppExchange[36], the service marketplace offered by salesforce.com,

[28] `http://aws.amazon.com/ec2/`
[29] `http://aws.amazon.com/s3/`
[30] `http://www.network.com/`
[31] `http://www.rightscale.com/`
[32] `http://www.strikeiron.com/`
[33] `http://www.xignite.com/`
[34] `http://www.salesforce.com/`
[35] `http://www.netsuite.com/`
[36] `http://www.salesforce.com/appexchange/`

offers a range of pre-integrated complementary services provided by third-party vendors grouped around the core service Salesforce CRM.

Service Requester The open and dynamic character of service value networks enables customers to request customized complex services from whatever service value network they prefer that satisfy their needs and match market requirements. Service requesters creatively create their complex services by composing adequate service components from multiple candidate pools in a plug-and-play fashion in order to receive added value. By concentrating on their core competencies, companies are not forced to provide solutions covering the whole range of a business process but they are able to complement their service portfolio by requesting complex services from service value networks (cp. Example 2.1).

Candidate Pool The structure of service value networks, characterized by their participants and their interrelations, is not static and predefined but formed on-demand in a short term, goal-oriented fashion. The formation process requires a steady pool of distributed and loosely-coupled service providers that offer predefined functionality through modularized services to be ready on call. In order to participate in service value networks, i.e. participate in candidate pools to be ready for service provision, service providers must register at a central registry and satisfy a set of minimum requirements such as interoperability through well-defined interfaces based on Internet standards. The process of registration can be activated by switching initiators, meaning that also an operator of a central registry might query and proactively invite suitable service providers to join a candidate pool. The open character of service value networks allows any service provider to potentially participate in value creation as long as minimum requirements are met.

Candidate pools group service offers of multiple service providers by functionality and capabilities exposed. Service offers covering the same spectrum of functionality (e.g. login/ID services such as OpenID[37] and Google Accounts[38,39]) are categorized in identical candidate pools. These services are replaceable and represent service substitutes form an economic perspective. The actual formation process occurs when a concrete service request is addressed to the loosely formation of service providers. Based on the required functionality and capabilities described by the request, feasible candidate pools are iteratively arranged in a way that they together contain the potential to jointly

[37] http://openid.net/

[38] https://google.com/accounts/

[39] Note that the Google Accounts service is not an adequate candidate to participate in an service value network in a strict sense, as it is proprietarily bound to Google services and does not expose a well-defined interface to be accessed in an open manner.

generate desired value. A coordination mechanism is required to chose a single service offer from each candidate pool based on a set of rules in order to efficiently instantiate the requested complex service to be provided to the service requester.

Complex Service The final outcome that is produced by a service value network is realized through a sequence of modularized service offers from a set of iteratively arranged candidate pools (cp. Figure 2.13), that is, a complex service. This final outcome is the added value generated for the service requester. The concept of a complex service, its characteristics and the way it is composed is explained in detail in Section 2.1.2.3.

Coordination Mechanism In environments with distributed, self-interested entities that jointly contribute to an overall goal, mechanisms are needed that coordinate procedures from multiple parties with possibly colliding objectives. Service value networks are a prominent example of such complex environments and their success therefore highly depends on adequate and efficient coordination mechanisms. As already mentioned in Section 2.1.3.4, *coordination is managing the dependencies of activities*. It is obvious that there exist various facettes of coordination forms that have to be chosen according to the characteristics and requirements imposed by the type of environment. The continuum of coordination ranges from market-based approaches to hierarchical control and dictatorships [Tho91, MC94]. Market-based approaches manage the activities of distributed, self-interested entities only indirectly by institutionalizing a rule set that incentivizes market participants to act in a desired manner in order to achieve an overall goal. Actors and dependencies of their activities are managed 'invisible' and 'unseen' driven by rational behavior of utility-maximizing economic entities and incentivized by rules to perform a social choice and compensate the entities' efforts. Nevertheless, there are situations in which this 'liberal' form of coordination results in inefficient outcomes. In this case, the economic entities need to be consciously organized in hierarchical forms to streamline activities in an efficient manner.

The problem of efficiently choosing adequate service offers from candidate pools to instantiate a complex service that meets the requirements imposed by the service requester is a traditional problem of coordination. Service providers are self-interested, act rational and therefore try to maximize their utility without accounting for a system-wide solution (e.g. a solution that maximizes welfare). Thus, the design of adequate coordination mechanisms is crucial to the efficiency and success of a service value network.

Example 2.4 [SVN Realizing a CRM Complex Service]. *This example shows the formation of a service value network that is ready to instantiate a complex service based on the requirements imposed by a service request. A service requester requires a complex service that scans calendar entries within the upcoming week with regard to future meetings within a company. Based on the the meetings' descriptions, the complex service queries soft skills of all meeting participants by browsing their profiles in social communities. Gathered information is then updated in a CRM data base that is stored by on-demand storage infrastructure (Figure 2.14).*

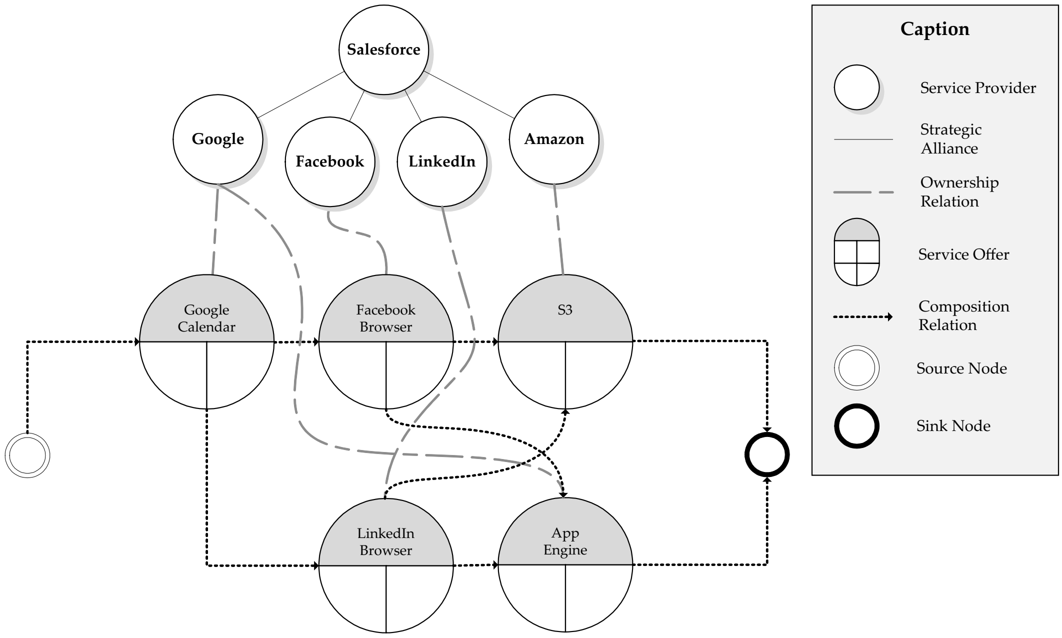


Figure 2.14: Example of a service value network realizing a CRM complex service.

A set of service providers participates in the service value network by offering services grouped in candidate pools. Google offers its Google Calendar service[40] *and Google App Engine*[41] *which provides a scalable infrastructure for service development and storage. The social community platforms Facebook*[42] *and LingedIn*[43] *provide services to browser profiles of registered users. Amazon offers flexible storage capabilities through its Simple Storage Service (S3)*[44]*. As depicted in Figure 2.14 the requested complex service can be realized in four different versions by selecting feasible service combinations (e.g. Google Calendar, LinkedIn Browser and Amazon S3).*

This example shows that service value networks foster the ad-hoc creation of short-living complex services that fulfill individual needs of a variety of consumers.

[40] `http://google.com/calendar`
[41] `http://code.google.com/appengine/`
[42] `http://facebook.com/`
[43] `http://linkedin.com/`
[44] `http://aws.amazon.com/s3/`

This type of complex service is also called service mashup or situational application. The following section introduces fundamentals of situational applications and service mashups, explains their role within service-ecosystems, and introduces key principles they are based on.

2.1.4.3 Situational Applications and Service Mashups

Competitive forces in today's markets result in the fact that dealing with change is a necessity for companies. This needs to be exploited and enabled by achieving flexibility in the organization and IT infrastructure [Eva91, GS06, AB91]. Flexibility is mainly concerned with the quick development of new applications to support changing business processes. In the past, IT departments have fallen short to satisfy the demand for new applications. Typically, situational applications that are needed only for a limited time span never made it into realization in favor of strategically important applications as part of the development backlog. Nowadays, most of the efforts of the IT departments are devoted to maintenance leaving many application wishes unfulfilled. With the advent of Web 2.0 technologies and the renaissance of HTTP appreciation, the possibilities to build "good enough" applications have greatly increased and traditional roles of service provider and service consumer blur.

A so-called *service mashup* is an application or Web site that aggregates content such as data feeds, applications, widgets, or gadgets from different sources [Mer06]. The number of publicly available mashups is dramatically increasing and can be checked at programmableweb.org[45]. While the first mashups were dedicated to small consumer mashups, where simple data (e.g. RSS feeds [BDBD^{+}00]) is integrated in the Web browser, mashup technology promises to integrate enterprise applications. In fact, mashups can be considered to provide solutions for the long tail of applications [And06].

As depicted in Figure 2.15, standard applications (such as ERP modules) are standardized, but need customization. This mass market exhibits only small degrees of customization but enjoys demand by many customers, i.e. *volume business*. Software companies have been exploiting these market segments. However, there is also a long tail of applications, which require highly specialized features – accordingly, this highly specialized software cannot be offered to many customers in scalable manner. It is thus not astonishing that these segments around the long tail have so far not been exploited. Summarizing, the long tail of applications is very fat in a sense that the demand for customized and quality differentiated software is immense, i.e. *value*

[45] `http://programmableweb.org/`

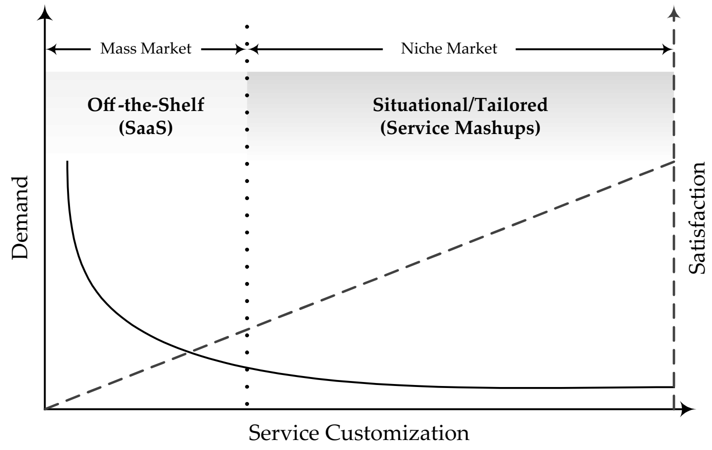


Figure 2.15: Situational applications address the long tail of business.

business. Due to the diversified demand there are numerous, hitherto unexploited niche markets, where the project set-up costs exceed the benefit.

With the technology of mashups, it is now possible to exploit the long tail as customization becomes cheaper through the aggregation of small services. Big and RESTful Web services encapsulate functionality and put it behind clearly defined interfaces based on SOAP, WSDL and HTTP respectively. Typically, it is distinguished between consumer, data and enterprise mashups. In fact, consumer mashups combine data elements from different sources and hides them behind a simple GUI (e.g. TuneGlue being an interactive visualization of the music artists available at Last.fm[46] which is linked with Amazon customer data). Data mashups combine data streams from different sources into one single data feed with one dedicated user interface attached to it. Enterprise mashups integrate data and other services (e.g. infrastructure services) from internal and external sources creating composite Web applications. Because of the simplicity in setting up composite applications, mashup technologies are expected to evolve significantly. Fierce competition and the corresponding needs for applications coerce companies into imperatives of the modern service-oriented economy that opens up the long tail of strong differentiation of their service offerings, and customer-centricity in the creation of services.

Service mashups also allow end-users to create customized applications by combining content, presentation functionality and business logic from heterogeneous sources using lightweight Web technologies. Through the extensive reuse of existing resources and simple programming models mashups facilitate the ad-hoc development of highly situation-specific applications which are often used for a short time only. Mashups therefore support the long tail of business, which cannot be served by traditional off-the-shelf software. Situational applications embody the next step in service-oriented computing and their ease of use heralds the next generation of

[46] `http://lastfm.com/`

flexibly recombined services. The following principles encompass the key innovation of situational applications:

Principle 2.1 [SIMPLIFICATION AND STANDARDIZATION]. *Service mashups and the way they are developed is a prominent result of a clear trend towards the simplification and standardization. Even complex services are increasingly exposed in the manner of puristic service descriptions and interfaces. As explained in Section 2.1.3.2, RESTful architectural styles leverage the power of the highly standardized and interoperable HTTP protocol. HTTP methods (e.g.* `GET`, `DELETE` *and* `CREATE`*) are used to build the most elementary signatures encapsulating scalable functionality in a distributed fashion. Unlike heavy-weight RPC-style architectures with high XML payload and complex programming-language-like interfaces, RESTful Web services are founded on unified interfaces based on HTTP methods and scoping information encoded in the service's URI.*

Principle 2.2 [LIGHTWEIGHT COMPOSITION AND FLEXIBLE BINDING]. *Puristic Web APIs such as REST and other lightweight approaches to Web service protocols and messages formats (e.g. JSON) enable ad-hoc composition and flexible binding of replaceable services [Jhi06]. Situational applications mostly focus on simple data manipulation and can therefore be piped sequentially. Well-defined building blocks as components of these sequences can be composed, decomposed and rearranged dynamically and enable demand-driven customization and satisfaction of individual consumer needs. A high degree of standardization regarding service interfaces allows for the specification of reusable service blueprints that define a skeleton of service mashups. Service components within these blueprints can be bound and instantiated at run-time as they are replaceable and puristic in nature.*

Principle 2.3 [MASS COLLABORATION AND CUSTOMIZATION]. *The central principle of a continuous development of situational applications is collaboration and customization [Mul06]. Participants are part of a mass co-production process that blurs the border between creation and consumption. Users contribute their individual knowledge about the existence, capabilities and compatibilities of feasible service components to service mashup models. A high degree of customization and self-selection continuously generates new demand and satisfies niche markets in the long tail [And06].*

Principle 2.4 [PERPETUAL BETA]. *The development of service mashups is comparable to agile software development and extreme programming [Mul06]. Multiple users continuously create and re-engineer service compositions using components that are mostly under the control of distributed owners. Service mashups are living applications that never reach a final state. They are created and improved through a trial-and-error-process that involves many participants manipulating models according to their needs and mostly self-interest.*

The following example illustrates the idea behind service mashups and how key principles are realized in the context of consumer mashups.

Example 2.5. *As an example consider a user Anna who wants to blog links about horseback riding on Iceland. The link list should be updated automatically as new articles about this topic are published on the Web. Since manual creation of the link list is therefore not possible, Anna decides to quickly create a tiny mashup for gathering, tagging and displaying the links.*

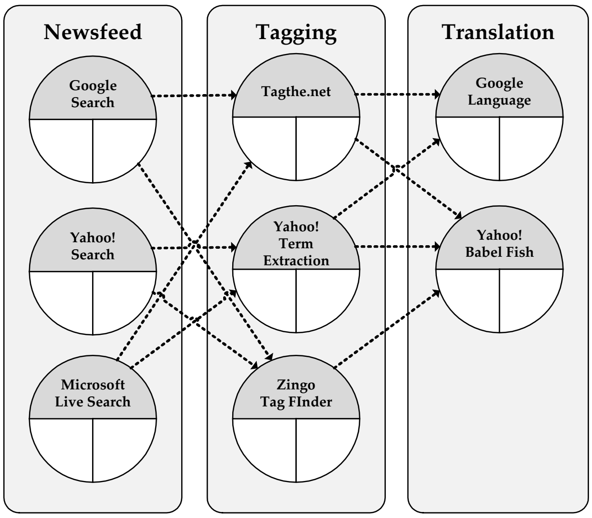


Figure 2.16: Blueprint of a translation and tagging service mashup.

As depicted in Figure 2.16, the mashup requires a newsfeed, tagging and translation service. Newsfeed services take the desired topic as input and return relevant news articles. In the following, relevant tags have to be determined for these articles. As Anna would like to keep her blog consistent in German, a service is required to translate the foreign language tags.

2.2 Markets in a Service World

The community is a fictitious body, composed of the individual persons who are considered as constituting as it were its members. The interest of the community then is, what? – the sum of the interests of the several members who compose it?

[Ben38]

This section elaborates the idea, necessity and applicability of markets in service-dominated environments which are constantly evolving in almost any field of society. Providing a first insight and a general motivation to the topic, Section 2.2.1 provides a thorough line of argument answering the question why auctions should be applied in the context of complex services and how they can serve to coordinate distributed activities to enable a flawless composition. The argumentation builds upon the general service characteristics as introduced in Section 2.1.1.2 and proclaims the need for auction-based dynamic allocation and pricing of service components generating added value through the composition of complex services.

Laying the groundwork for the design of mechanisms, Section 2.2.3 introduces the approach of mechanism design, elaborates economic objectives that are desirable when implementing a social choice, and briefly introduces prominent mechanisms along with a set of impossibility theorems. Bringing mechanism design in the context of service value networks and information systems design, the idea behind algorithmic mechanism design is motivated.

As the process of designing market-mechanisms for a specific domain is complex and involves many steps and multiple factors, Section 2.2.2 introduces the concept of an electronic market and provides a market engineering process as a structured approach for the discipline of market engineering. Each phase within the market engineering process is iteratively mapped on the structure of the work at hand.

The Section 2.2.4 concludes with a detailed analysis of economic and applicability requirements, an auction mechanism has to meet to support dynamic allocation and pricing of complex services in networked environments such as service value networks. Based on the requirements analysis, related work is presented and evaluated illustrating the research gap which is filled by this thesis.

2.2.1 Why Auctions for Complex Services?

In general, an adequate approach for allocation and pricing of complex services has to account for *service characteristics* as introduced in Section 2.1.1.2. As stated by [Smi89] "auctions flourish in situations in which the convential ways of establishing price and ownership are inadequate". Smith concretizes the argumentation by briefly pointing out the main characteristics of such situations which are predestinated for the application of auctions by focusing on the roles and items involved: "*costs cannot be established*, [...], there is something *special* or *unusual* about the item, *ownership is in question*, *different persons assert special claims*, [...]." Although this statement is rather fuzzy, the characterization of the type of 'item' which price is best established by the application of an auction mechanism opens up an analogy to the service concept. Recall, in Section 2.1.1.2 services are characterized by the coincide of production and consumption (uno-actu), they cannot be inventoried, value creation is dominated by intangible elements, consumer co-production and fuzzy inputs and outputs.

Smith points out that auctions are preferable in situation where costs cannot be established. From an microeconomic perspective such costs refer to internal costs that are *private information* to the one producing the item, i.e. the producer's individual valuation for the item. In the context of services, this argument also holds for the consumer side. According to the service characteristic C 2.4, value that is generated for the service consumer is mostly dominated by intangible elements and therefore hard to determine. An objective measurement of quality which might be an indicator for the consumer's valuation is also hardly applicable due to a service's fuzzy inputs and outputs according to characteristic C 2.5. The *complexity of value elicitation* and the problem of establishing adequate prices even increases in scenarios with joint value creation through service compositions (e.g. in service value networks where complex services are produced). Analogue to Smith's argumentation, such problems can be addressed by the design of a suitable auction mechanism that induces incentives for service providers to report their private valuations truthfully. Auctions haven proven to be the ideal instrument to *aggregate information* from distributed parties which results in an aggregated valuation [PS00, Jac03]. Without prior knowledge about the valuations of each participant, auctions can provide suitable incentives to make truth revelation an equilibrium strategy and therefore automatically *aggregate necessary information* from self-interested participants to determine adequate prices for complex services.

Another criterion that is crucial to establishing a suitable approach for allocation and pricing according to [Smi89] is if the item subject to trade exposes special or unusual characteristics. The uno-actu principle (C 2.1) implies that in the context of services there cannot be a producer without at a consumer as *production and consumption coincides in time*. This service characteristic has fundamental implications on coordination aspects as service cannot be inventoried in order to balance demand and supply. Following the same direction, Lucking-Reiley enriches this argumentation by adding an economic perspective which explicitly focuses on the trade of services by stating that "[...] in the future we may see much more auctioning of services [...]. Services are particularly attractive for auctions because they are in *relatively fixed supply* – unlike durable goods, *one cannot store surpluses or draw down inventory* in order to meet fluctuating demand." [LR00]. Market mechanisms such as auctions are preferable in situations with a *fast changing demand and supply ratio* as dynamic pricing smoothes high amplitudes. This property is crucial to success of *efficient allocation* and pricing especially when perishable services are traded [Eso01].

The rapid growth of information and communication technology has tremendously decreased transaction costs for service provision and consumption. Computing power and storage raises exponentially while prices drop anti-proportionally for hardware as illustrated by Moore's Law. This development directly leads to a tough *price competition* for service providers. In order to stay competitive, service providers have to *differentiate their service offers with respect to quality* (not price) [Dev98, MV98, DLP03, LSW01, BP91]. *Quality is the main value-determining factor* in the context of services as service consumers experience a service activity mainly based on the quality provided. Quality is idiosyncratic to the individual and often determined by various factors and the interplay of multiple service components that are part of a service composition. Hence, it is unbearable for service consumers to reason about all feasible combinations of single services and the resulting quality provided by the service composition in order to meet their requirements. Therefore an auction mechanism is needed which accounts for *different preferences of service requesters* defined for a *variety of quality characteristics* that are determined by each component that is part of feasible complex service instances (cp. Section 2.1.2.3). Especially in the context of situational complex services provided by distributed parties in service value networks, a QoS-sensitive auction mechanism allows for the provision and pricing of highly customized short-term solutions to various types of customers leveraging the nature and benefits of situational applications and service mashups (cp. Section 2.1.4.3). As a consequence, service providers in service value networks are able to *address the long tail of business* by satisfying a great amount of individual service requests [And06]. In these environments, it is assumed that service offers are under the control of distributed self-interested owners. In the absence of central control, non-performance or complete drop-outs of service components maybe rare but inevitable. Auction mechanisms that are *computational feasible* allow for reallocation and price adaption during run-time enabling *dynamic failovers* in unreliable environments [FKNT02].

2.2.2 Electronic Markets and Market Engineering

Coordination of transactions requires an adequate form of organization and coordination mechanism. From an economic theory perspective, two extreme forms have to be distinguished: *markets* and *hierarchies*. Markets coordinate transactions by means of a rule set which constraints the way transactions may take place. The coordination itself results from a balance between demand and supply and consequently determines dynamic prices, quantities, quality and so forth. In the past, markets have been used in environments with relatively simple products with respect to attributes and quality and low specificity (e.g. commodity goods) due to high coordination costs for message exchange and matching of demand and supply (cp. Figure 2.17). In the absence of modern information and communication technology, complex products or services are costly to coordinate (e.g. complex descriptions require complex bidding languages and messages as well as highly sophisticated matching algorithms) [MS84]. Traditionally, in scenarios with complex products, hierarchies have proven to perform quite well due to a higher degree of planning and control, which results in lower coordination costs (less messages have to be exchanged and no complex matching is required). A detailed analysis of trade-offs between markets and hierarchies with respect to transaction and coordination costs can be found in [Wil79, Mal85, MS84, Mal87].

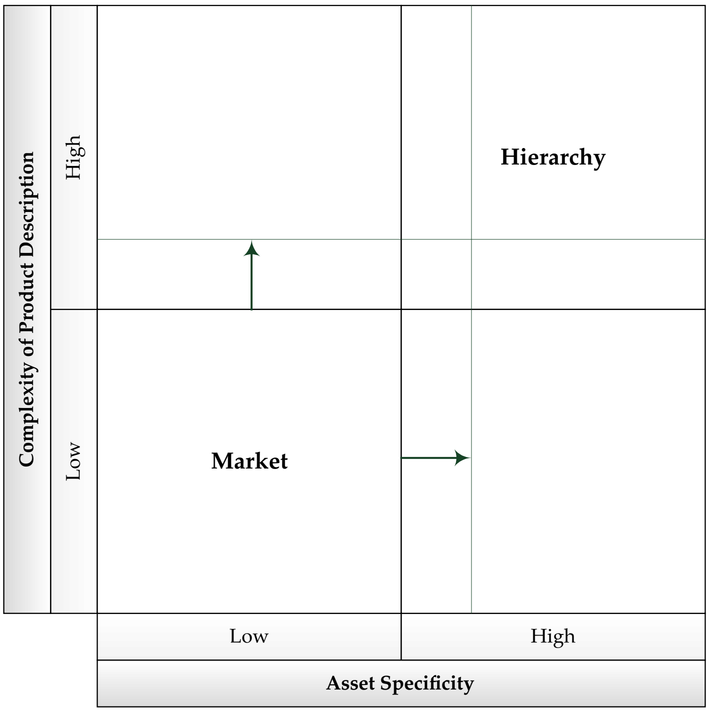


Figure 2.17: Characteristics of products and services affect forms of organization [MYB87].

However, this argumentation does not hold under the presence of modern information and communication technology and powerful dynamic infrastructures built upon the principles of the SOA paradigm. Due to more efficient and sophisticated information and communication infrastructures, market-based coordination in electronic environments can be realized [MYB87]. Therefore the following definition of an *electronic market* can be concluded:

Definition 2.9 [ELECTRONIC MARKET]. *An electronic market is an institutions built upon information and communication technology that establishes a market-based coordination of transactions by enabling the ubiquitous trade of products and services between multiple distributed participants.*

Designing market mechanisms in electronic environments is a complex process that requires knowledge and expertise in the area of economics and computer science. Interdependencies between economic desiderata such as allocation efficiency (cp. Section 2.2.3) and technical applicability requirements such as computational tractability have to be identified and feasible trade-offs have to be analyzed in order to achieve desired goals [WNH06]. Different aspects from technical and economic viewpoints often lead to colliding objectives that have to be resolved through relaxation of requirements and objectives or designing suitable trade-offs between conflicting goals. Relying on existing market mechanisms originally designed for other environments may often lead to poor market performance and inefficient outcomes [Lai05].

Hence, the process of designing markets for a specific domain must be well-structured and based on a solid engineering methodology. The market engineering process according to [Smi82, Neu04, WNH06] is structured as depicted in Figure 2.18. It mainly consists of four stages: *Environmental analysis*, *design and implementation*, *testing*, and *introduction*. Each stage is briefly introduced in the remainder of this section.

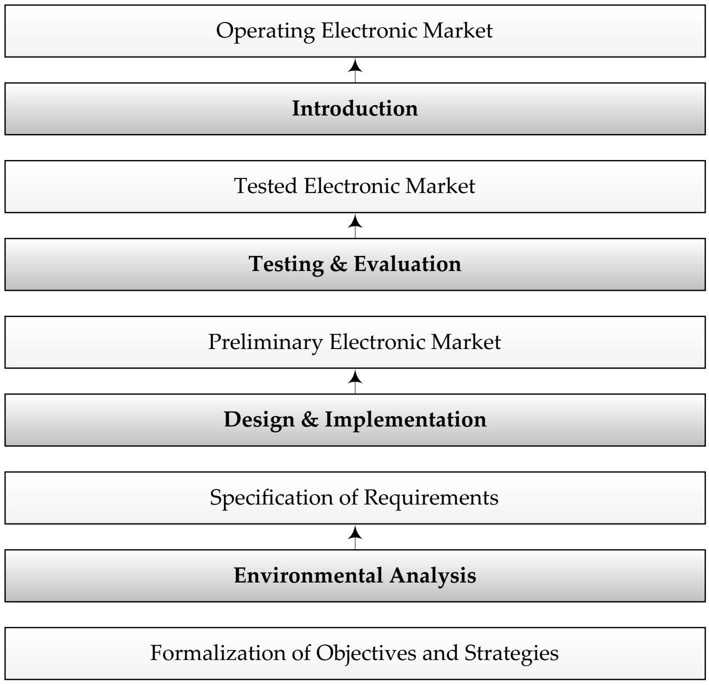


Figure 2.18: Stages of the market engineering process [Neu04].

2.2.2.1 Environmental Analysis

The *environmental analysis* is the first phase of the market engineering process and comprehends the phases *environmental definition* and *requirement analysis*.

The environmental definition targets the gathering of necessary information that allows for an efficient market design. This information covers the characteristics and types of objects that are subject to trade, possible market participants, their objectives and possible strategies as well as information about intermediaries in the market as analyzed in Chapter 2. Based on this information, potential market segments are identified and evaluated comparatively.

Hence, this analysis serves as a basis for deriving requirements and desiderata for the design phase, i.e. the requirement analysis. A thorough environmental analysis is fundamental to the success of an efficient market design. The results of the environmental analysis of this work are outlined in Section 2.2.4.

2.2.2.2 Design and Implementation

Having derived desiderata and requirements for a domain-specific market design, the next stage covers the *conceptual design phase* as the central element of the market engineering process. Analogously to the design of systems and architectures in the computer science domain, markets are meaningfully composed out of modularized elements in order to achieve a desired market performance and outcome. The conceptual design constitutes a set of institutional rules in an abstract manner independent of a concrete implementation (analogue to a platform- and programming-model-independent design of a software artifact e.g. in UML [OMG07]). The conceptual design of this work that comprehends the design of a bidding language to express service offers and requests as well as a mechanism design with additional extensions is introduced in Section 3 using an implementation-independent mathematical formalization.

The conceptual design lays the groundwork for the actual implementation of the market into an information system. This phase is distinguished in the *embodiment phase* and the *implementation phase*. In the embodiment phase, the conceptual design is refined, concretized and extended where required into a more specific market scheme but still remains implementation-independent. This phase of the market engineering process is realized in the work at hand in Chapter 4.

The condensed market scheme is subsequently modeled into a formal process model describing the domain-specific market to be prototypically realized. Section 3.5 introduces the process model for the auction conduction which serves as procedural blueprint for the subsequent implementation phase.

Finally, in the implementation phase, the prototypical implementation of the market design is realized based on the results of the previous phases. A prototypical implementation of the work at hand is introduced and briefly described in Section 3.6.

2.2.2.3 Testing and Evaluation

Having completed the conceptual design phase, the embodiment phase and the implementation phase, the created artifacts are tested and evaluated with respect to the specified desiderata and requirements in the environmental analysis. In the *evalu-*

ation phase, both, technical and applicability requirements (e.g. support for service compositions) as well as economic requirements (e.g. incentive compatibility) are evaluated and verified in this phase.

Depending on the aspect subject to evaluation, adequate methods and approaches have to be chosen and selected based on their applicability. Exemplary, the economic desideratum, which states that the mechanism shall implement a social choice function that is weakly budget-balanced can be theoretically evaluated using mathematical proofs. Strategic behavior of market participants with respect to bundling strategies might be too complex to be theoretically investigated but requires an agent-based simulation approach to evaluate such aspects. The evaluation phase of the work at hand is therefore divided into an analytical evaluation part in Chapter 5 and an numerical evaluation part in Chapter 6.

Based on the obtained information out of the testing and evaluation phase about the satisfaction of requirements by the market design and the achievement of desired outcomes, a final refinement takes place to complete the market for operative introduction.

2.2.2.4 Introduction

The introduction phase constitutes the final phase of the market engineering process. In this phase, the evaluated and refined electronic market is introduced and initiates its operation cycle.

2.2.3 Mechanism Design

Mechanism design is a subfield of game theory that pursues the idea of designing institutions that determine decisions as a function of the information that is known by the individuals in the economy in order to achieve a desired outcome [Mye88]. Mechanisms serve as a unifying conceptual structure, which allows for analyzing and comparing economic institutions with respect to their properties and suitability in order to foster certain outcomes. Analogue to traditional game theory, mechanism design assumes individuals in an economy to be rational-acting and self-interested, meaning they pursue individual utility maximization. According to [Par01] the mechanism design problem can be defined as follows:

Definition 2.10 [Mechanism Design]. *The mechanism design problem is to implement an optimal system-wide solution (social choice) to a decentralized optimization problem with self-interested agents with private information about their preferences for different outcomes.*

2.2.3.1 Social Choice

The main goal of mechanism design is to provide mechanisms that implement a *social choice*. A social choice function is an aggregation of the preferences of multiple participants into a single joint decision [NRTV07]. In environments with decentralized, rationally-acting agents that have private information about their preferences for different outcomes, the implementation of a social choice function is necessary to achieve an overall goal due to the absence of complete information.

Given the agent's *type* $\theta_i \in \Theta_i$ with $i \in \mathcal{I}$, the preferences for different outcomes $\rho \in \mathcal{R}$ result in the agent's utility $u_i(\rho,\theta_i)$. A *social choice function* selects – given the agents' types – the optimal outcome ρ^*.

Definition 2.11 [SOCIAL CHOICE FUNCTION]. *A social choice function $\omega : \Theta_1 \times \cdots \times \Theta_I \rightarrow \mathcal{R}$ selects an optimal outcome $\omega(\theta) = \rho^*$ with $\rho^* \in \mathcal{R}$ given the agent's types $\theta = (\theta_1,\ldots,\theta_I)$. The outcome ρ is decomposable into a choice $\omega_o(\theta) \in \Omega_o$ and payments made by each agent $\omega_{t_i}(\theta) \in \Omega_t$.* [47]

The outcome of a social choice function is a system-wide solution that can not be solved directly as the agent's types are private information to the agents. Thus, an adequate mechanism is needed that defines a set of game theoretic rules to implement the solution to the social choice function accounting for rational and selfish behavior of the agents. The behavior of agents is game theoretically defined by means of strategies. A *strategy* describes a complete and contingent plan that defines the actions an agent will select in every possible state of a game [Gib92, Par01]. A strategy $\psi_i(\theta_i)$ of an agent i is defined as $\psi_i(\theta_i) \in \Psi_i$ where θ_i denotes the type of agent i and S_i all possible strategies depending on its type. Based on the concept of a social choice function and agents' behavior by means of their strategies, a mechanism is defined as follows:

Definition 2.12 [MECHANISM]. *A mechanism $\mathcal{M} = (\Psi_1,\ldots,\Psi_I,m(\cdot))$ defines an outcome rule $m(\cdot)$ that maps strategies $\Psi_1,\ldots,\Psi_I$ of agents $1,\ldots,I$ to an outcome $\rho \in \mathcal{R}$ such that $m : \Psi_1 \times,\ldots,\times\Psi_I \rightarrow \mathcal{R}$. The outcome rule $m(o(\cdot),t(\cdot))$ consists of a choice or allocation rule $o(\cdot)$ and a payment or transfer rule $t(\cdot)$ that determines the monetary transfer to the agents.* [47]

Hence, a mechanism determines the agents' strategy space and defines a certain *outcome* given the chosen strategies. The outcome defines an *allocation* (e.g. agent s_r

[47] Decomposition into a choice and a payment component is only feasible under the assumption of quasi-linear preferences which is common in game theory.

gets service v from agent s_p) and the monetary exchange – the *transfer* – between agents (e.g. agent s_r has to transfer an amount x to agent s_p).

Recall that the goal of mechanism design is to implement an optimal system-wide solution (social choice) to a decentralized optimization problem even though the participants are self-interested and have private information about their preferences for different outcomes. As agents are assumed to act rational and therefore to maximize their individually utility, a solution in such a scenario must be a state where no agent gains by changing its own chosen strategy unilaterally, i.e. an *equilibrium* in game theoretic terms. The goal of a mechanism is to *implement* a social choice function, that is, a mechanism constitutes an equilibrium that yields the same outcome as the optimal solution to the social choice function for all possible agent preferences.

Definition 2.13 [MECHANISM IMPLEMENTATION]. *A social choice function $\omega(\theta)$ with outcome $\rho^* \in \mathcal{R}$ is implemented by a mechanism $\mathcal{M} = (\Psi_1,\ldots,\Psi_I,m(\cdot))$ if $m(\psi_1^*(\theta_1),\ldots,\psi_I^*(\theta_I)) = \rho^*$ with $(\psi_1^*,\ldots,\psi_I^*) \in \Psi_1\times,\ldots,\times\Psi_I$ and $(\theta_1,\ldots,\theta_I) \in \Theta_1\times,\ldots,\times\Theta_I$ where strategy profile $(\psi_1^*,\ldots,\psi_I^*)$ is an equilibrium strategy given mechanism $\mathcal{M}$.*

One can distinguish between direct and indirect mechanisms. In a direct mechanism, agents submit their messages once to the mechanism and the outcome is computed subsequently. In an indirect mechanism, agents may submit several messages to the mechanism an receive feedback which is incorporated by the agents. The focus of the work at hand is restricted to direct mechanisms. A direct-revelation mechanism is defined as follows:

Definition 2.14 [DIRECT-REVELATION MECHANISM]. *A direct-revelation mechanism restricts the strategy set for all agents $i \in I$ to strategies where agent i reports the type $\acute{\theta}_i = \psi_i(\theta_i)$ based on its actual preferences θ_i.*

The relation between a mechanism, its implementation and the achievement of the same outcome as a social choice function depicted in Figure 2.19, which is based on the illustration in [Rei77].

In distributed environments with self-interested agents, a system-wide solution to a social choice problem is not solvable directly as rational-acting agents cannot be assumed to reveal their private information e.g. for the sake of welfare. The agents' primary objective is to maximize their individual utility, which mostly collides with a truth-telling strategy. In the absence of complete information regarding agents' preferences for different outcomes, a mechanism $\mathcal{M}$ must be designed that imple-

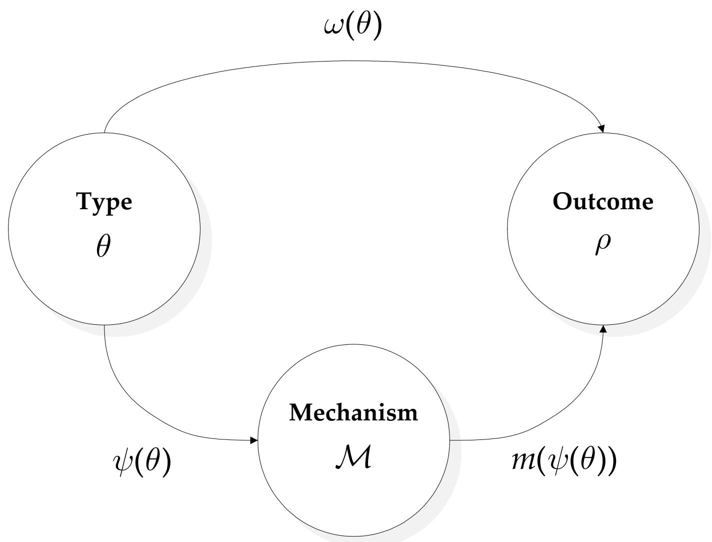


Figure 2.19: Triangle relation of mechanism implementation and social choice [Rei77].

ments a desired social choice function by means of a rule set that specifies how to allocate and how to transfer payments. The mechanism implementation induces incentives that constitute an equilibrium strategy profile which yields the same outcome as the social choice function such that $m(\psi(\theta)) = \omega(\theta)$.

2.2.3.2 Properties of Social Choice and Mechanism Implementations

The objective of mechanism design is to implement a social choice function in equilibrium strategies that yields desired properties. Such properties are often referred to as mechanism properties. Nevertheless mechanisms do not directly expose these properties but they implement social choice functions that do. For the reader's convenience properties of social choice are also referred to as mechanism properties in the remainder of this thesis. For an extended introduction to mechanism and social choice properties, the interested reader is referred to [Par01].

Desideratum 2.1 [Allocative Efficiency]. *A social choice function $\omega(\theta) = (\omega_o(\theta), \omega_t(\theta))$ is allocative efficient if it maximizes the total utility over all agents. Let $\omega_o^*(\theta) \in \Omega_o$ be an allocative efficient choice, then no alternative choice $\acute{\omega}_o(\theta) \in \Omega_o$ yields a higher utility for all agents such that:*

$$\sum_{i \in \mathcal{I}} u_i(\omega_o^*(\theta), \theta_i) \geq \sum_{i \in \mathcal{I}} u_i(\acute{\omega}_o(\theta), \theta_i), \quad \forall \acute{\omega}_o(\theta) \in \Omega_o \tag{2.1 (AE)}$$

Desideratum 2.2 [(Dominant Strategy) Incentive Compatibility]. *A mechanism $\mathcal{M}$ is incentive compatible if agents report truthful information about their preferences in equilibrium. A mechanism $\mathcal{M}$ is* strategy-proof *or* dominant-strategy incentive-compatible *if each agent i's best response to any strategy of the other agents* *is revealing its true type, i.e. reporting true information about the preferences is a dominant strategy in equilibrium. In other words there is no incentive for agents to announce untruthful information about their preferences in order to increase their individual utility. Let* $\psi_i^*(\theta_i) = \theta_i$ *be the truth-revelation strategy for agent i. For a strategy-proof mechanism* $\mathcal{M}$ *it is required that*

$$u_i(m(\psi_i^*(\theta_i), \psi_{-i}(\theta_{-i})), \theta_i) \geq u_i(m(\acute{\psi}_i(\theta_i), \psi_{-i}(\theta_{-i})), \theta_i), \quad \forall \acute{\psi}_i \in \Psi_i \setminus \{\psi_i^*\}, \quad \forall \psi_{-i} \in \Psi_{-i}, \quad \forall i \in \mathcal{I} \tag{2.2}$$

which means that the truth-revelation strategy is a dominant strategy for all agents. Furthermore it is required that the strategy profile

$$\psi^* = (\psi_1^*(\theta_1), \ldots, \psi_I^*(\theta_I)) \qquad \text{(DSIC)} \tag{2.3}$$

is an equilibrium given mechanism $\mathcal{M}$.

Desideratum 2.3 [Individual Rationality]. *A mechanism* $\mathcal{M}$ *is individual rational if it implements a social choice function* $\omega(\theta) = (\omega_o(\theta), \omega_t(\theta)) = \rho$ *that guarantees that agents are not worse-off by participating. Let* $u_i(\rho, \theta_i)$ *be the utility of agent i in case of participation and* $\bar{u}_i(\theta_i)$ *the utility of its outside option, i.e. its utility if agent i does not participate.*

$$u_i(\rho, \theta_i) \geq \bar{u}_i(\theta_i), \quad \forall i \in \mathcal{I} \qquad \text{(IR)} \tag{2.4}$$

Assuming a mechanism where an agent can withdraw once it knows the outcome ex-post is individual rational if participation makes the agent not worse-off compared to the outside option of not participating for all possible agent types in the system. In mechanisms where agents are not able to observe the outcome, meaning the decision to participate has to be done ex-ante, the concept of interim individual rationality *is introduced, which is a weaker property from an* ex-ante *perspective.*

$$E(u_i(\rho, \theta_i)) \geq E(\bar{u}_i(\theta_i)), \quad \forall i \in \mathcal{I} \qquad \text{(IIR)} \tag{2.5}$$

The expected utility $E(u_i(\rho, \theta_i))$ *for agent i from participation is not worse then its expected utility* $E(\bar{u}_i(\theta_i))$ *from not participating.*

Desideratum 2.4 [Budget Balance]. *A social choice function* $\omega(\theta) = (\omega_o(\theta), \omega_t(\theta))$ *is strong budget-balanced if all payments made by the agents are distributed among all agents. This means that there are no outside payments necessary to realize transfers ac-*

cording to the outcome of the social choice function.

$$\sum_{i \in \mathcal{I}} \omega t_i(\theta) = 0 \qquad (2.6) \quad \textit{(BB)}$$

There are no net transfers neither into the system nor out of the system. A weaker version of budget balance is if there are transfers out of the system but not into the system, i.e. weak budget balance.

$$\sum_{i \in \mathcal{I}} \omega t_i(\theta) \geq 0 \qquad (2.7) \quad \textit{(WBB)}$$

Although all of these valuable properties of social choice and mechanism implementations are desired from an economical perspective, they cannot be achieved at the same time due to impossibilities, which are presented in detail in Section 2.2.3.4.

2.2.3.3 Possibility Results

Maybe the most important possibility result in mechanism design is the revelation principle as it implies that it is sufficient to restrict to direct incentive compatible mechanisms. The principle is defined as follows:

Definition 2.15 [Revelation Principle]. *Any mechanism $\mathcal{M}$ that implements a social choice function $\omega(\cdot)$ in dominant strategies*[48] *can also be implemented by an* incentive compatible direct-revelation mechanism *that implements the same social choice function $\omega(\cdot)$ in dominant strategies.*

The intuition behind the revelation principle can be illustrated as follows: Assuming the agents' strategy profile $\psi^* = (\psi_1^*, \ldots, \psi_I^*)$ in equilibrium in a mechanism $\mathcal{M}$ leads to an outcome $\rho(\psi^*)$. Now, the behavior of the agents is simulated by a mechanism $\acute{\mathcal{M}}$ called a *simulator* which computes the optimal strategies of the agents based on their reported preferences. Hence, for each agent $i \in I$ it is a dominant strategy to report its type truthfully to the mechanism $\acute{\mathcal{M}}$. Consequently the simulator $\acute{\mathcal{M}}$ implements the same social choice function as $\mathcal{M}$.

To illustrate the idea of the revelation principle the following example presents a general mechanism and an equivalent incentive compatible direct-revelation mechanism that leads to the same outcome. The example is a slightly changed variant of an example in [Mye88] with an extensive analysis.

[48]Note that the first version of the revelation principle in [Gib73] is restricted to mechanisms that implement a social choice function in *dominant strategies*. In [Mye82] the principle is extended to the general case for all equilibrium concepts e.g. Bayesian-Nash equilibria.

Example 2.6 [INCENTIVE COMPATIBLE DIRECT-REVELATION MECHANISM]. *Consider a game where two agents i and $-i$ have private valuations v_i and v_{-i} for a good g. Both agents separately put amounts b_i and b_{-i} in two different envelops. The agent that reports the higher amount gets the good and the other one gets both envelopes. Presented game is symmetric and therefore both agents try to maximize the same expected utility. Without loss of generality, agent i's expected utility is analyzed.*

$$E_i(\cdot) = P(b_i > b_{-i})\,[v_i - b_i] + P(b_i < b_{-i})\,[b_i + b_{-i}] \tag{2.8}$$

Two cases must be considered:

1. *Getting the good g yields a higher utility for agent i then getting both envelopes such that*

$$\begin{aligned} (v_i - b_i) &> (b_i + b_{-i}) && (2.9)\\ v_i - 2b_i &> b_{-i} && (2.10) \end{aligned}$$

   Consequently agent i wants to maximize the probability of winning the good. $P(b_i > b_{-i})$ is maximized by reporting an amount $b_i = v_i - 2b_i$ which leads to the strategy of reporting an amount $b_i = \frac{1}{3}v_i$.

2. *Getting the good g yields a lower utility for agent i then getting both envelopes such that*

$$\begin{aligned} (v_i - b_i) &< (b_i + b_{-i}) && (2.11)\\ v_i - 2b_i &< b_{-i} && (2.12) \end{aligned}$$

   Consequently agent i wants to maximize the probability of getting both envelopes and loosing the good. $P(b_i < b_{-i})$ is maximized by reporting an amount $b_i = v_i - 2b_i$ which leads to strategy of reporting an amount $b_i = \frac{1}{3}v_i$.

The strategy of announcing an amount $b_i^* = \frac{1}{3}v_i$ is the best response of agent i not knowing agent $-i$'s strategy. As the game is symmetric this argumentation also holds for agent $-i$. Consequently, the strategy $b^* = \frac{1}{3}v$ constitutes an equilibrium.

Without loss of generality let agent i be the agent that wins the good g such that $b_i > b_{-i}$. Thus, the outcome of the game based on the agents' equilibrium strategies evolves as follows:

$$\begin{aligned} u_i(\cdot) &= \frac{2}{3}v_i && (2.13)\\ u_{-i}(\cdot) &= \frac{1}{3}v_{-i} + \frac{1}{3}v_i && (2.14) \end{aligned}$$

According to the revelation principle (Definition 2.15) an equivalent incentive compatible direct-revelation mechanism can be designed that yields the same outcome:

The mechanism allocates the good g to the agent that reports the higher amount and charges one-third of that amount. The other agent that does not receive the good gets one-third of both reported amounts. Analogously to the previous game, the expected utility of agent i is analyzed.

$$E_i(\cdot) = P(b_i > b_{-i})\left[v_i - \frac{1}{3}b_i\right] + P(b_i < b_{-i})\left[\frac{1}{3}b_i + \frac{1}{3}b_{-i}\right] \tag{2.15}$$

Two cases must be considered:

1. *Getting the good g yields a higher utility for agent i then getting one-third of both reported amounts such that*

$$(v_i - \frac{1}{3}b_i) > (\frac{1}{3}b_i + \frac{1}{3}b_{-i}) \tag{2.16}$$
$$3v_i - 2b_i > b_{-i} \tag{2.17}$$

   Consequently agent i wants to maximize the probability of winning the good. $P(b_i > b_{-i})$ *is maximized by reporting an amount* $b_i = 3v_i - 2b_i$ *which leads to the truth-telling strategy* $b_i = v_i$.

2. *Getting the good g yields a lower utility for agent i then getting one-third of both reported amounts such that*

$$(v_i - \frac{1}{3}b_i) < (\frac{1}{3}b_i + \frac{1}{3}b_{-i}) \tag{2.18}$$
$$3v_i - 2b_i < b_{-i} \tag{2.19}$$

   Consequently agent i wants to maximize the probability of getting both envelopes and loosing the good. $P(b_i < b_{-i})$ *is maximized by reporting an amount* $b_i = 3v_i - 2b_i$ *which also leads to the truth-telling strategy* $b_i = v_i$.

Without loss of generality let agent i be the agent that wins the good g such that $b_i > b_{-i}$*. Thus, the outcome of the game based on the agents' equilibrium truth-telling strategies evolves as follows:*

$$u_i(\cdot) = \frac{2}{3}v_i \tag{2.20}$$
$$u_{-i}(\cdot) = \frac{1}{3}v_{-i} + \frac{1}{3}v_i \tag{2.21}$$

The example at hand illustrates the idea of the revelation principle by showing that there exists a direct-revelation mechanism that yields the same outcome as the general mechanism in a truth-telling equilibrium, i.e its incentive compatible. Note that the example demonstrates the application of the more general revelation principle according to [Mye82] that extends results in [Gib73] – that restrict the revelation principle to dominant strategy equilibria – to the general case for multiple equilibrium concepts e.g. Bayesian-Nash equilibria.

Summarizing, with the results of the revelation principle, impossibility results can be proven over the space of direct-revelation mechanisms, and possibility results can be constructed over the space of direct-revelation mechanisms.

Maybe the most prominent family of direct-revelation mechanisms are the Vickrey-Clarke-Groves (VCG) mechanisms [Vic61], [Cla71] and [Gro73]. VGC mechanisms belong to the class of Groves mechanisms and are *individual rational*, *allocatively-efficient* and *strategy-proof* direct-revelation mechanisms. For a detailed analysis of the family of VCG mechanisms and their properties, the interested reader should refer to [Par01].

2.2.3.4 Impossibility Results

Despite of possibility results such as the revelation principle, there are important impossibility results that have strong limitations to design goals that can be simultaneously pursued. In fact, it is impossible to achieve certain combinations of design desiderata as outlined in the previous section. Among the most prominent are the following theorems:

Theorem 2.1 [HURWICZ (GREEN-LAFFONT) IMPOSSIBILITY THEOREM]. *There is no double-sided mechanism that is at the same time allocative efficient, budget-balanced, and truthful in settings with quasi-linear preferences [GL78, Wal80, HW90].*

The Theorem 2.1 restricts its proposition and applicability to dominant-strategy equilibria, whereas the following theorem by Myerson and Satterthwaite makes a more generic statement:

Theorem 2.2 [MYERSON-SATTERTHWAITE IMPOSSIBILITY THEOREM]. *There is no double-sided mechanism that is at the same time allocative efficient, budget-balanced, Bayesian-Nash incentive compatible, and (interim) individually rationality, even in settings with quasi-linear preferences [MS83].*

Theorem 2.2 extends the former theorem also to situations where reporting ones true type is a Bayesian-Nash equilibrium where participants intent to maximize their expected utility instead of their ex-post utility. By extending their proposition, Myerson and Satterthwaite add the condition that the mechanism must be individual rational.

In summary, the Myerson-Satterthwaite Impossibility Theorem implies that at most two desiderata out of allocation efficiency, individual rationality, and budget balance can be achieved when designing truthful mechanisms in settings where agents are assumed to have quasi-linear preferences.

2.2.3.5 Algorithmic Mechanism Design

Algorithmic mechanism design – firstly introduced by [NR01] – broadens the economic focus by considering problems that are inherent in the mechanism design problem from a computer science and algorithmic perspective such as complexity and computational feasibility of computing an optimal system-wide solution. Internet protocols for example are designed under the implicit assumption that each participant within the system behaves according to a deterministic procedure or program. Nevertheless, this assumption does not hold in environments such as the Web as participants and owner of computer systems and applications are self-interested and act according to their individual objectives.

Many challenges in computer science involve selfish behavior of decentralized participants and thus, require adequate mechanisms to implement an efficient solution such us internet routing, scheduling and task allocation, resource allocation, and service composition [NRTV07]. In such scenarios, agents cannot be assumed to follow a deterministic algorithm but try to maximize their own utility which might collude with other objectives and a system-wide solution.

Especially the coordination of service composition requires a mechanism design that accounts for selfish behavior of distributed service providers by implementing the right incentives to jointly achieve a common goal that serves the objectives and well-being of the overall system. Despite of such economic challenges, this scenario puts further technical requirements upon a potential mechanism design in order to be applicable for the coordination of composite service creation. Hence, this broadens the view of mechanism design regarding the field of algorithms and information systems design [DJP03].

2.2.4 Environmental Analysis and Related Work

This section outlines requirements upon a mechanism in order to be applicable in the context of coordination in service value networks from an economic and technical perspective (Section 2.2.4.1). Based on the requirement analysis, Section 2.2.4.2

introduces and describes related work and critically examines their shortcomings in the context of stated requirements and the approach at hand.

2.2.4.1 Requirements

There is a number of requirements a mechanism must and partly should satisfy in order to be applicable in the context of service composition in service value networks from an economic and technical perspective. Requirements upon a mechanism are basically dividable into *economic requirements* and *applicability requirements*. Economic requirements are explained in detail in Section 2.2.3.5 and are therefore only outlined briefly at this point:

Requirement 1 [ALLOCATIVE EFFICIENCY]. *A mechanism is said to be allocative efficient if it always determines the outcome that maximizes the overall utility across all participants (consumer and provider surplus), i.e. it always maximizes the system's welfare (cp. Desideratum 2.1).*

Requirement 2 [INCENTIVE COMPATIBILITY]. *A mechanism is said to be (dominant strategy) incentive compatible or truthful if the truth-telling strategy is an equilibrium in weakly dominant strategies (cp. Desideratum 2.2).*

Incentive compatibility is an important requirement as it functions a precondition for the allocative efficiency requirement. In distributed environments incentive compatibility enables the transition from incomplete (private) information to the situation in which participants reveal their true types voluntarily. This reported information is necessary for a welfare-maximizing solution to be always computable as stated in Requirement 1. Furthermore, truthfulness tremendously reduces the complexity of the strategy space of participants. Under the presence of a weakly dominant strategy there is no need to reason about the other participants' preferences.

Requirement 3 [INDIVIDUAL RATIONALITY]. *A mechanism implements a social choice that is said to provide the property of individual rationality if agents cannot suffer a loss in utility from participating in the mechanism, i.e. the option to participate in the mechanism is not worth than the outside option.*

Requirement 4 [BUDGET BALANCE]. *A mechanism is said to be (weakly) budget-balanced if its transfers do not require external subsidization by outside payments, i.e. the requester's willingness to pay covers payments transferred to providers (cp. Desideratum 2.4).*

Budget balance and individual rationality are crucial for a sustainable implementation of a mechanism with respect to the underlying business model. If budget balance is not met, the mechanism must continuously be subsidized by outside payments which is not feasible from the strategic perspective of e.g. a service platform provider. Additionally if individual rationality is not me by the mechanism, agents will not voluntarily participate in the mechanism as they face the risk of being worse off compared to their outside option.

For a mechanism in order to be applicable in the context of complex services in service value networks from a technical and domain-specific perspective, the following requirements have to be met:

Requirement 5 [COMPUTATIONAL TRACTABILITY]. *A mechanism is said to be computational tractable if it computes an allocation and corresponding prices in polynomial runtime in the size of its inputs, i.e. e.g. the number of service offers and their feasible compositions into a complex service.*

Computational tractability is important for mechanisms that need to perform in online systems, i.e. they need to compute an allocation and prices at runtime within a feasible time frame. Especially in the context of service value networks, the number of feasible paths through the network – that is, the number of feasible complex service instances – increases rapidly (exponentially) as the number of service providers and candidate pools increases[49].

Requirement 6 [SERVICE COMPOSITION SUPPORT]. *Service compositions, in contrary to service bundles, only generate value for the requester in the right order of their components. Thus, a mechanism in a broader sense is said to support service composition if its bidding language and allocation function accounts for the well-defined sequence of service components in order to form a feasible complex or composite service.*

Support for service composition is a rare requirement in the context of combinatorial mechanisms. Although most approaches in this area provide rich bidding languages, they only support bundles in an economic sense which ignores the order of the entities the bundle consists of[50].

[49]Based on the service value network model in Section 2.1.4, the number of feasible paths depends on the number of candidate pools and service offers per candidate pool. Assuming an equal number of service offers per pool, the number of paths is $\left(\frac{|V\setminus\{v_s,v_f\}|}{K}\right)^K$, with K denotes the number of candidate pools.

[50]E.g. its not possible to express a preference like $(A,B) \succ (B,A)$

Requirement 7 [QoS-Sensitivity]. *A mechanism in a broader sense is said to be QoS-sensitive if it accounts for complex QoS characteristics by providing an adequate bidding language and allocation function that is implemented by a corresponding allocation algorithm.*

Requirement 8 [Service Level Enforcement]. *A mechanism in a broader sense is said to provide service level enforcement if it incorporates information about the fulfillment of QoS aspects. Based on this information, the mechanism's transfer function provides means for rewarding or penalizing agents.*

Requirements 6 and 8 together are important to provide a sustainable support for the coordination and trade of complex services as it enables differentiation in quality and a trustworthy environment for service contracts.

2.2.4.2 Related Work

This section outlines research approaches that are closely related to the work at hand and highlights research gaps and shortcomings that are addressed and partly solved by this approach.

A double-sided market mechanism for trading Grid resources is presented in [Sto09]. The computation of the allocation is based on a greedy heuristic which is scalable and performs well also in large-scale settings while minimizing efficiency loses compared to an optimal solution that is computational intractable. In the work, two pricing schemes are presented. The first, a proportional critical value pricing scheme that successfully limits strategic behavior of market participants on the expense of computational costs. The second pricing scheme, k-pricing is highly scalable while sacrificing incentive compatibility to a certain degree. Nevertheless, only low-level resource-oriented services (cp. the bottom layer in the service decomposition model in Section 2.1.2) are tradable as the mechanism and the bidding language do not support compositions of services, complex QoS characteristics and their enforcement.

Allowing the trade of service bundles, MACE (Multi-Attribute Combinatorial Exchange [Sch07]) and the Bellagio System [ACSV04] provide mechanism for the trade of infrastructure resources. Resource service are specified by rudimentary quality attributes and can be requested and provisioned as bundled services. Although the trade of service bundles is supported, their is no support for service compositions as the bidding language is only capable of capturing bundle specifications independent of the sequence of entailed service components. Furthermore, preferences for service attributes can only be specified by means of rudimentary operations such as AND, OR, and XOR whereas only simple quality attributes such as response time are supported. From an economic perspective, neither mechanism implements truthfulness with respect to resource prices which allows for strategic behavior of participants that is only partly limited by the pricing scheme. From a technical perspective, the winner determination problem in both mechanisms is computational intractable which does not allow for their application in large-scale online settings.

In [LS06], the MACE exchange is extended by means of semantic concepts and technologies. A combinatorial double auction is presented that is continuously cleared. Corresponding bidding language supports the trade of service bundles but is not capable of capturing information about sequential compositions. Services are specified by means of semantically describable quality attributes which allows for highly differentiated service offers with respect to their QoS characteristics. Nevertheless, from an economic perspective, the auction mechanism does not provide incentives for truth-revelation of private valuations and QoS attributes of traded services. Furthermore, in settings which require the timely allocation of services, the auction mechanism is not applicable as it exposes exponential run-time behavior.

Focusing on mechanisms for allocation and pricing of service compositions that expose a well-defined control sequence, a combinatorial auction for QoS-aware dynamic web services composition is proposed in [MNM$^+$07]. Their composition model heavily relies on the work in [ZBD$^+$03] where feasible service compositions are predefined based on a statechart graph. Based on this model, a QoS-sensitive combinatorial auction mechanism is proposed which allocates the composition of services which yields the highest quality level based on the requesters preferences subject to budget constraints which results in a computational intractable problem. From an economic perspective, the mechanism's design does not implement incentives for truth-revelation of QoS attributes and private valuations. The mechanism neither verifies the services' performance ex-post nor incorporates penalties at the current state of the work.

In summary, as comprised in Table 2.3, a lot of work has been done with respect to designing suitable mechanisms for allocation and pricing of services in different levels of granularity (utility, elementary and complex services). Nevertheless, there still exist various research gaps especially in the context of incorporating feasibility of service compositions in the allocation problem as well as QoS-sensitivity and adequate ex-post verification mechanisms to impose penalties for non-performance.

Table 2.3: Requirements satisfaction degree of related approaches (● = fully satisfied, ◐ = partly satisfied, ○ = not satisfied).

Approach	*(R 1)* **Allocative Efficiency**	*(R 2)* **Incentive Compatibility**	*(R 3)* **Individual Rationality**	*(R 4)* **Budget Balance**	*(R 5)* **Computational Tractability**	*(R 6)* **Service Composition Support**	*(R 7)* **QoS-Sensitivity**	*(R 8)* **Service Level Enforcement**
	Economic Requirements					*Applicability Requirements*		
Stößer 2009	○	◐	●	●	●	○	○	○
Schnizler 2007	○	○	●	●	○	○	◐	○
Lamparter et al. 2006	○	○	●	●	○	○	●	○
Mohabey et al. 2007	○	○	●	●	○	●	●	○
This Work	●	●	●	○	●	●	●	●
This Work (extended)	◐	◐	●	●	●	●	●	●

2.3 Research Methods

The primary goal of the work at hand is not to analyze existing mechanisms but to design novel mechanisms that expose desired properties and induce desired behavior of participants in a particular domain. As pointed out in [Rot02], an "engineering approach" is required for designing suitable market mechanisms. This work is founded on the approach of mechanism design [Mye88, NR01] which is introduced in detail in Section 2.2.3.5. In order to evaluate the properties and the behavior of participants in the developed auction mechanism, the complex service auction, this work heavily relies on two methodologies: *theoretical analysis* and *simulations* which are briefly introduced in the remainder of this section.

2.3.1 Theoretical Analysis

To study the main properties of the auction mechanism, concepts and methods from *game theory* are employed. This implies to make strong assumption about the market participants with respect to the information about other participants and the utility functions [MCWG95]. There exist multiple solution concepts in game theory such as Nash equilibria and dominant strategy equilibria. Theoretical analysis provides strong results. Nevertheless, in order to apply analytical game theoretic evaluations, models usually rely on strong assumptions that do not necessarily reflect real world settings.

2.3.2 Simulations

Evaluating certain mechanism properties or behavior of participants in settings with a multitude of variable factors, a theoretical analysis is not applicable in most of the cases due to the high complexity of the system. As a remedy, numerical simulations provide a useful tool to analyze particular properties of a mechanism by means of randomly generated problem sets, i.e. the variable factors are randomly generated for multiple simulation runs. Numerical simulations can provide insights into the general problem structure, performance aspects of the algorithm that solves the winner determination problem, mechanism properties and strategic behavior of participants.

Focusing on more complex settings with participants that face large strategy spaces which precludes theoretical solutions, the methodology of agent-based simulations has proven to be promising [Bon02]. Strategic behavior is simulated by means of collections of computerized agents that implement the ability to learn their surroundings and the space of feasible solutions. In contrary to a traditional game theoretic analysis, agent-based simulations provide means for the evaluation of rare strategies which are more complex and occur in special domains. Nevertheless, it is crucial to design reasonable strategies and learning behavior and incorporate them into software agents. However, a lot of work has been done in the area of agent-based simulations and a whole set of different strategies has been shown to work well in many settings [Phe08].

Part II

Design & Implementation

Chapter 3

Complex Service Auction (CSA)

I believe that in the future we may see much more auctioning of services [...]. Services are particularly attractive for auctions because they are in relatively fixed supply – unlike durable goods, one cannot store surpluses or draw down inventory in order to meet fluctuating demand.

[LR00]

The fundamental paradigm shift from vertical integration to horizontal specialization and the coherent transformation of traditional value chains to highly dynamic value networks is predominantly observable in the service sector. At the same time, customers' demand for sophisticated, customized services has considerably been rising in recent years. Open standards and service-oriented architectures have emerged as important building blocks for innovative service value networks tying together the competencies of specialized contributors. Thus, by modularization, complex services are increasingly able to be composed in a "plug-and-play"-manner [VvHPP05]. This novel form of value creation in loosely-coupled service ecosystems is unique from a coordination and incentive engineering perspective as it exposes cooperative and non-cooperative aspects. Participants in such service value networks are both, self-interested – i.e. they try to maximize their individual utility – but also fully bound to the success of the whole system.

It is a well-known result from Market Engineering (cp. Section 2.2.2) that there is no general mechanism that fits any possible setting [WHN03]. An adequate mechanism depends amongst others on the properties of the trading objects – which are service components and complex services in the work at hand – and the goals of the designer (e.g. welfare vs. revenue maximization). Having analyzed the characteristics of services in general in Section 2.1.1.2, and special aspects of software services in Section 2.1.3 as well as their composition into complex services in service value networks in Section 2.1.2 and 2.1.4, the set of requirements and desiderata from a technical and an economic perspective upon a suitable mechanism were outlined in Section 2.2.4.

Section 3 focuses on the design of an auction mechanism – the Complex Service Auction (CSA) – that enables based on service offers and requests the allocation of multidimensional service components which are sequentially composed into feasible complex service instances. An abstract model is introduced that comprehends a bidding language to describe information objects that are exchanged during the auction process. Additionally the model provides means to formalize service value networks in a graph-based structure. The mechanism itself is capable of allocating service components and determining dynamic prices and corresponding QoS characteristics of complex services. Furthermore, in Chapter 4 extensions to the complex service auction are developed in order to meet the applicability requirements such as QoS-sensitivity and service level enforcement and to achieve budget balance.

For the remainder of this section it is useful to refer to the design framework for market mechanisms depicted in Figure 3.1. Analogue to the structure of this section, there are three fundamental components in the design of a market mechanism [DVVfMSiES03]: the *bidding language* (cp. Section 3.2), that provides means for formalizing information objects and all their necessary parts for the requester and the provider side that are exchanged during the conduction of e.g. the complex service auction; the *allocation function* (cp. Section 3.3.1) which determines which trading object(s) are allocated to which participant(s); and the *transfer function* (cp. Section 3.3.2) that determines based on the allocation the monetary transfers that have to be realized among the participants. Focusing on the realization of a mechanism implementation, the concrete *allocation algorithm* that computes the allocation function is a central design issue. In this context, design desiderata such as computational tractability and allocative efficiency strongly depend on the design of the allocation algorithm. Counteracting complexity, heuristic algorithms might restore computational tractability by sacrificing optimality to a certain extent [Sto09]. In contrary, exact algorithms enable the computation of an allocative efficient outcome (assuming incentive compatibility) but might result in exponential run-time [Sch07].

Based on the impossibility results as described in Section 2.2.3.4, there is an inherent trade-off between design desiderata (cp. Section 2.2.4.1) that has to be considered when constructing the mechanism's components.

For the reader's convenience, the formal notation that is used throughout this section, is outlined in Section A.1 in tabular form.

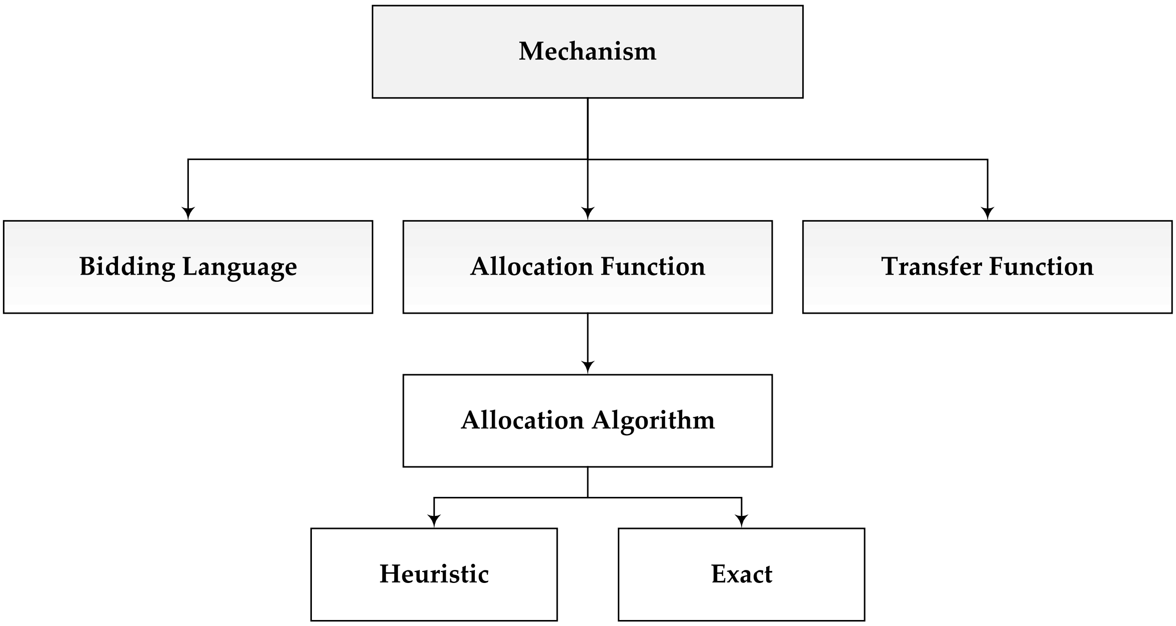


Figure 3.1: Framework for the design of mechanisms.

3.1 Service Value Network Model

Recall that Section 2.1.4 is concerned with an initial description of service value networks, their main characteristics and the various roles involved in value creation. In addition to this first outline, this section focuses on providing a mathematical model of a service value network that captures the presented aspects in a comprehensive technical manner.

A service value network is described by means of a simplified statechart model [HN96] and is aligned with the representation in [ZBD+03] as depicted in Figure 3.2. Statecharts have proven to be the preferred choice for specifying process models as they expose well-defined semantics and they provide flow constructs offered by prominent process modeling languages (e.g. WS-BPEL) and therefore allow for simple serialization in standardized formalisms.

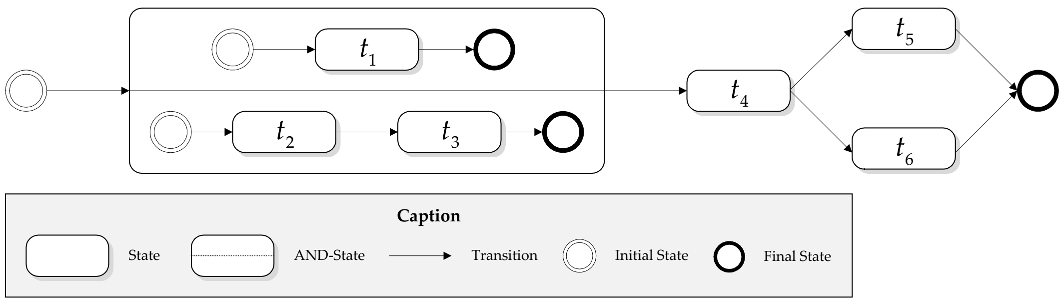


Figure 3.2: Statechart formalization [HN96, ZBD+03].

Hence, a service value network is represented by a k-partite, directed and acyclic graph $G = (V, E)$. Each partition $Y_1, \ldots, Y_K$ of the graph represents a *candidate pool* that entails service offers that provide the same (business) functionality. The set of N nodes $V = \{v_1, \ldots, v_N\}$ represents the set of *service offers*[1] with u, v, i, j being arbitrary service offers. There are two designated nodes v_s and v_f that stand for source and sink in the network and are not part of any partition $\mathcal{Y} = (Y_1, \ldots, Y_K)$, hence $V = Y_1 \cup \cdots \cup Y_K \cup \{v_s, v_f\}$. Services are offered by a set of Q *service providers* $S = \{s_1, \ldots, s_Q\}$ with s being an arbitrary service provider. The *ownership information* $\sigma : S \to \mathcal{P}(V \setminus \{v_s, v_f\})$ that reveals which service provider owns which services within the network is public knowledge[2]. The set of edges $E = \{e_{ij} | i, j \in V\}$ denotes technically feasible service composition such that e_{ij} represents an interoperable connection of service $i \in V$ with service $j \in V$[3]. If two services are not interoperable at all, they are not connected within the network.

A *service configuration* A_j of service offer $j \in V$ is fully characterized by a vector of attributes $A_j = (a_j^1, \ldots, a_j^L)$ where a_j^l is an attribute value of attribute type $l \in \mathcal{L}$ of service offer j's configuration. Attribute types can be either functional attribute types or non-functional attribute types (e.g. availability or privacy). A service's configuration represents the quality level provided and differentiates its offering from other services. According to [Lam07], a service configuration can be defined as follows:

Definition 3.1 [SERVICE CONFIGURATION]. *A service configuration A_j of a service $j \in V$ selects a value a_j^l for each attribute type $l \in \mathcal{L}$ of a service and thereby unambiguously defines all relevant service characteristics. The choice of configuration might affect the functional and non-functional aspects of a service and is a major determinant of the price.*

Furthermore let c_{ij} denote the *internal variable costs* that the service provider that owns service j has to bear for that service being interoperable with service i and for the execution of service j as a successor of service i. The representation of a detailed cost structure of service providers is intentionally omitted which serves a better understanding and does not restrict the generalization of the model. It is assumed that the representation of internal variable costs reflects the service providers' valuations for their service offers being executed in different composition-related contexts.

Example 3.1 [CONTEXT-DEPENDENT COST STRUCTURES]. *In order to illustrate the idea of context-dependent cost structures of service providers refer to Figure 2.1. For simplification, the complex service is reduced to the first two business transactions,* data verification *and the* transaction processing. *Figure 3.3 shows the service value network with service offers and corresponding costs dependent on the preceeding service. Data verification*

[1] For the reader's convenience the terms *service offer*, *service* and *node* are used interchangeably

[2] The reverse ownership information $\sigma^{-1} : V \setminus \{v_s, v_f\} \to S$ maps service offers to single service providers that own that particular service

[3] For the reader's convenience the notion e_{ij} is equivalent to $e_{v_i v_j}$ representing an interoperable connection of service $i \in V$ with service $j \in V$.

can be performed by either Strike Iron (s_A) and its service offer A or CYDNE (s_B) offering service B. The execution of the actual monetary transaction is done by Net Billing (s_C) offering service C.

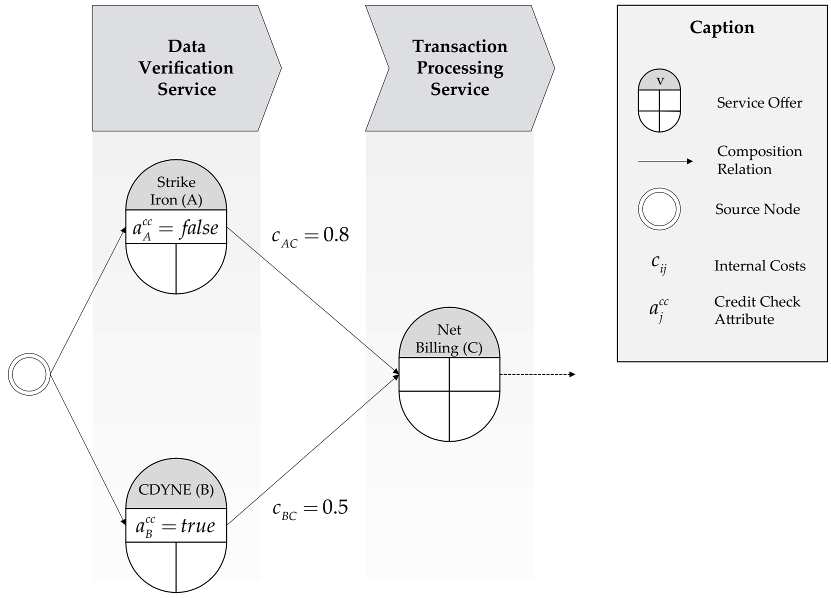


Figure 3.3: Context-dependent cost structures of service providers.

A mandatory step of the overall payment processing service is the credit assessment. As a precondition, a transaction processing service has to check if the customer is credit worthy in order to charge the corresponding account. The credit assessment has to be performed at a central authority (e.g. Equifax, Experian or Trans Union) and generates variable costs each time it is executed. In the concrete scenario, Net Billing has to bear higher costs of 0.8 *in case it is executed as a successor of the Sales Force data verification service as it does not provide a credit check in advance. In contrary, the service offered by CYDNE is capable of performing a credit check, which results in lower internal costs for Net Billing of* 0.5.

As already illustrated in Section 2.1.2.3 and Section 2.1.4, the instantiation of a complex service is represented by a path from source to sink within the service value network. Let F denote the set of all feasible paths from source to sink. Every $f \in F$ with $f \subset E$ represents a possible instantiation of the complex service[4].

Definition 3.2 [SERVICE VALUE NETWORK MODEL]. *A service value network model is an acyclic, k-partite and directed graph such that*

$$G = (V, E) \tag{3.1}$$

[4]Focusing on the presence or absence of a particular service $i \in V$, F_{-i} represents the set of all feasible paths from source to sink in the reduced graph G_{-i} without node i and without all its incoming and outgoing edges. In contrary, let F_i be the subset of all feasible paths from source to sink that explicitly entail node i.

with the set of nodes V representing service offers and the set of edges E that denotes technically feasible service compositions. G contains two designated nodes v_s and v_f representing source and sink such that every feasible path $f \in F$ connecting both nodes is a possible instantiation of the complex service.

For illustration purpose, Figure 3.4 shows the model of a service value network with service offers $V = \{v_1, \ldots, v_4\} \cup \{v_s, v_f\}$ and service providers $S = \{s_1, \ldots, s_3\}$. Every feasible path $f \in F$ connecting source node v_s and sink node v_f represents a possible realization of the overall complex service.

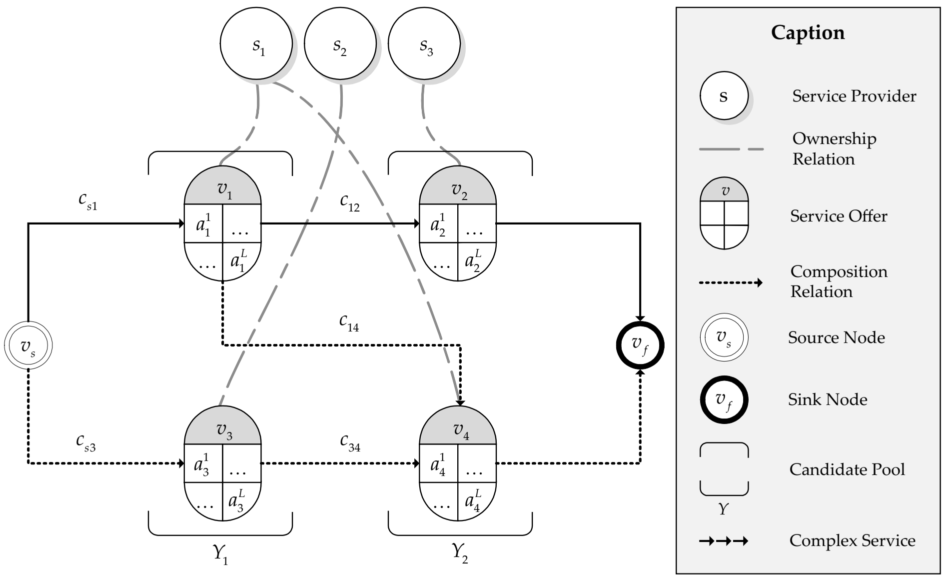


Figure 3.4: Service value network model.

3.2 Bidding Language

As a formalization of information objects which are exchanged during auction conduction a bidding language is introduced that is based on bidding languages for products with multiple attributes as discussed in [EWL06]. The formalization is aligned to multiattribute auction theory as presented in [PK02, RL05] and assures compliance with the WS-Agreement specification [ACD$^+$04] in order to enable realization in decentralized environments such as the Web.

3.2.1 Scoring Function

A complex service – represented by a path f – is characterized by a configuration $\mathcal{A}_f$. The importance of certain attributes and prices of a requested complex service is idiosyncratic and depends on the preferences of the requester. The requesters' preferences are represented by a scoring function $\mathcal{S}$ of the form:

$$\mathcal{S}(\mathcal{A}_f) = \left(\sum_{l=1}^{L} \lambda_l \|\mathcal{A}_f^l\| \right) \tag{3.2}$$

The scoring function $\mathcal{S}$ represents the requesters' preferences for a configuration $\mathcal{A}_f$ of the complex service represented by f analog to the definition of scoring rules in [AC08]. It maps the configuration of a complex service to a value representing the requester's score such that $\mathcal{S} : \mathcal{A} \rightarrow [0;1]^5$. The scoring function is determined by a vector of weights $\Lambda = (\lambda_1, \ldots, \lambda_L)$ with $\sum_{l=1}^{L} \lambda_l = 1$ that defines the requester's preferences of each attribute type $l \in \mathcal{L}$. The configuration $\mathcal{A}_f$ of the complex service f is constituted by the aggregation of all attribute values of contributing services with incoming edges on the path f such that

$$\mathcal{A}_f = (\mathcal{A}_f^1, \ldots, \mathcal{A}_f^L) \text{ with } \mathcal{A}_f^l = \bigoplus_{e_{ij} \in f} a_j^l \tag{3.3}$$

The aggregation operation $\bigoplus$ for attribute values depends on their type (e.g. the attribute type encryption is aggregated using a Boolean AND operator whereas response time is aggregated by a sum operator). Table 3.1 shows different types of aggregation functions for sample multiple attribute types.

Table 3.1: Aggregation operations for different attribute types.

Attribute Type	**Aggregation**
$l \in \mathcal{L}$	$\bigoplus_{e_{ij} \in f \mid j \neq v_f} a_j^l$
Response Time (rt)	$\sum_{e_{ij} \in f \mid j \neq v_f} a_j^{rt}$
Encryption Type (et)	$\bigwedge_{e_{ij} \in f \mid j \neq v_f} a_j^{et}$
Error Rate (er)	$\max_{e_{ij} \in f \mid j \neq v_f} a_j^{er}$
Throughput (tp)	$\min_{e_{ij} \in f \mid j \neq v_f} a_j^{tp}$
Probability of Default (pd)	$1 - \prod_{e_{ij} \in f \mid j \neq v_f} (1 - a_j^{pd})$

[5] Note that the scoring function is only capable of expressing soft policies and no goal policies (cp. [Lam07]). Nevertheless, in Section 4.3 an extension is introduced which enables the specification of more complex QoS characteristics and corresponding goal policies.

The list of aggregation operations in Table 3.1 only shows a rather trivial subset of possible and practical aggregation operations for different quality aspects of services and is not exhaustive. The bidding language also supports rich semantic approaches towards more complex aggregation operations in order to deal with various non-functional service attributes. For example, services are capable of different types of encryption algorithms and a requester prefers asymmetric ciphers, semantic subsumption can be used to evaluate if a complex service fulfils the requester's requirements and therefore to determine the score. Bidding, aggregation and management of complex QoS aspects within the CSA is presented in detail in Section 4.3.

To assure comparability of attribute values from different attribute types and to express requesters' preferences more sophisticated, the aggregated attribute values are normalized on an interval [0;1] using preference functions with lower (bottom) and upper (top) boundaries. Boundaries are defined by a vector $\Gamma = ((\gamma_B^1, \gamma_T^1), \ldots, (\gamma_B^L, \gamma_T^L))$ for each attribute type l with $\gamma_B^l \neq \gamma_T^l \; \forall l \in \mathcal{L}$. γ_B^l represents the attribute value boundary that results in a zero utility for the requester with respect to attribute type l (bottom boundary). γ_T^l denotes the attribute value boundary for type $l \in \mathcal{L}$ that just leads to a maximum utility of 1 for the requester (top boundary). The mapping of attribute values is specified by the following piecewise defined function.

$$\|\mathcal{A}_f^l\| = \begin{cases} g^l(\mathcal{A}_f^l) & \text{,if } \gamma_T^l > \gamma_B^l \wedge \gamma_B^l < \mathcal{A}_f^l < \gamma_T^l \\ 1 & \text{,if } \gamma_T^l > \gamma_B^l \wedge \mathcal{A}_f^l \geq \gamma_T^l \\ 0 & \text{,if } \gamma_T^l > \gamma_B^l \wedge \mathcal{A}_f^l \leq \gamma_B^l \\ h^l(\mathcal{A}_f^l) & \text{,if } \gamma_T^l < \gamma_B^l \wedge \gamma_T^l < \mathcal{A}_f^l < \gamma_B^l \\ 1 & \text{,if } \gamma_T^l < \gamma_B^l \wedge \mathcal{A}_f^l \leq \gamma_T^l \\ 0 & \text{,if } \gamma_T^l < \gamma_B^l \wedge \mathcal{A}_f^l \geq \gamma_B^l \end{cases} \tag{3.4}$$

The function $g : \mathcal{A} \to [0;1]$ is a monotonically increasing utility function such that g^l represents the requesters' utility function for attribute type l. An increasing utility function g^l indicates that the requesters utility increases with higher values of attribute type l. Attribute types such as response time result in a loss of utility the higher the attribute value. The preference for these types of attributes is expressed by a monotonically decreasing utility function such that $h : \mathcal{A} \to [0;1]$.

Example 3.2 [Scoring Function Computation]. *This example illustrates how different attribute types are aggregated along a path of composed service offers in service value networks. It furthermore shows how the requester's weights and boundaries for different* *attribute types are used to compute the requesters individual score for feasible service compositions constituting complex service instances.*

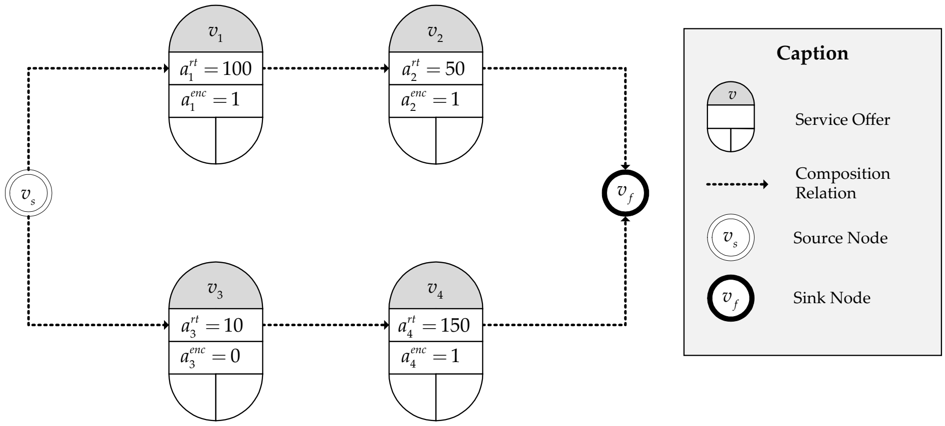


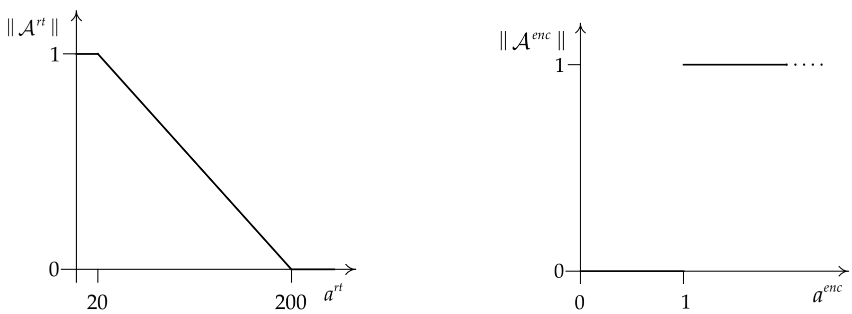
Figure 3.5: Service value network with service offers and corresponding configurations.

As depicted in Figure 3.5 the service value network contains four service offerings unambiguously specified by attribute values for the types response time (rt) and encryption (enc). Each feasible path $f^a = \{e_{s1}, e_{12}, e_{2f}\}$ *and* $f^b = \{e_{s3}, e_{34}, e_{4f}\}$ *from source to sink represents a possible instantiation of the complex service. Attribute values for the complex service are computed using suitable aggregation operations according to Table 3.1. Hence, the upper path has a response time of* $\mathcal{A}^{rt}_{f^a} = 150$ *and an encryption level* $\mathcal{A}^{enc}_{f^a} = 1$*. Analogue for the lower path:* $\mathcal{A}^{rt}_{f^b} = 160$ *and* $\mathcal{A}^{enc}_{f^b} = 0$*.*

In this example, the requester's reported vector of boundaries is $\Gamma = ((200, 20), (0, 1))$*. For simplicity it is assumed that its utility functions for each attribute type are linear such that*

$$h^{rt}(\mathcal{A}^{rt}_f) = \frac{200 - \mathcal{A}^{rt}_f}{200 - 20} \text{ and } g^{enc}(\mathcal{A}^{enc}_f) = \mathcal{A}^{enc}_f$$

According to the piecewise defined normalization function (cp. Equation (3.4)*), the requester's utility for different types of attributes and their values is illustrated in Figure 3.6.*



(a) Requester Utility for Different Levels of Response Time

(b) Requester Utility for Different Levels of Encryption

Figure 3.6: Requester utility for different attribute types.

Normalization of the attribute values according to Equation (3.4) *leads to the following values for each feasible complex service instance:*

$$\|\mathcal{A}^{rt}_{f^a}\| = 0.28, \|\mathcal{A}^{enc}_{f^a}\| = 1, \|\mathcal{A}^{rt}_{f^b}\| = 0.22, \|\mathcal{A}^{enc}_{f^b}\| = 0$$

In the example at hand it is assumed that response time is more important to the service requester then encryption, which leads to the vector of weights $\Lambda = (0.7, 0.3)$. *According to Equation* (3.2) *the requesters final score for each complex service instance computes as follows:*

$$\begin{aligned}\mathcal{S}(\mathcal{A}_{f^a}) &= 0.7 \cdot 0.28 + 0.3 \cdot 1 = 0.496 \\ \mathcal{S}(\mathcal{A}_{f^b}) &= 0.7 \cdot 0.22 + 0.3 \cdot 0 = 0.154\end{aligned}$$

Based on the requester's preferences (specified by the vector of boundaries), the utility functions and the vector of weights for different attribute types, the complex service f^a *yields a higher individual score, i.e. it is preferable for the service requester.*

3.2.2 Service Requests

Having defined how the score for certain outcomes is computed based on the requester's preferences, a specification of the willingness to pay is introduced that determines the rate of substitution between score and price. Let $\mathcal{T}_f = \sum_{s \in S} t^s$ represent the sum of all monetary transfers to service providers, i.e. the overall price of the complex service denoted by f. Hence, the requester's utility gained from purchasing a complex service specified by a path f with a configuration $\mathcal{A}_f$ evolves as follows:

$$\mathcal{U}^R_f(\alpha, \Lambda, \Gamma, \mathcal{A}_f, \mathcal{T}_f) = \alpha \mathcal{S}(\mathcal{A}_f) - \mathcal{T}_f \tag{3.5}$$

The factor α represents the requester's willingness to pay for a "perfect" configuration $\mathcal{A}_f$ with score $\mathcal{S}(\mathcal{A}_f) = 1$ based on reported preferences. In other words α defines the individual substitution rate between quality and price such that the requester is indifferent between an increase of 1 score unit and α monetary units. Incorporating that information, a service request for a multidimensional complex service is defined as follows:

Definition 3.3 [Multidimensional Service Request]. *A multidimensional service request for a complex service is a vector of the form:*

$$R := (\mathcal{Y}, \alpha, \Lambda, \Gamma) \tag{3.6}$$

such that $\mathcal{Y} = (Y_1, \ldots, Y_K)$ represents all candidate pools with the service value network, i.e. necessary information for each service provider about preceeding service offers[6]*. The maximum willingness to pay for a configuration that yields a score of 1 is denoted by α. The set of weights Λ represents the requesters' preferences for different attribute types $l \in \mathcal{L}$. Γ denotes the set of lower and upper boundaries for each attribute type.*

Example 3.3 [Multidimensional Service Request]. *Recalling Example 3.2, a multidimensional service request of a requester with a willingness to pay of $\alpha = 100$ is denoted by*

$$R = (\{v_1, v_3\}, \{v_2, v_4\}, 100, (0.7, 0.3), ((200, 20), (0, 1)))$$

For realization in a distributed environment such as the Web, compliance with interoperable and standardized exchange formats such as the WS-Agreement specification [ACD+04] is preferable. As the representation of α, Λ and Γ is straightforward, the information about the service value network topology requires an intermediate XML-based serialization such as the Graph eXchange Language (GXL) [Win02].

3.2.3 Service Offers

Having specified the bidding language for requesters we define a notation for the provider side. A multidimensional service offer consists of an announced service configuration A_j and a corresponding price p_{ij} that a service provider wants to charge for the service j being invoked depending on the predecessor service i. An offer bid $b_{ij} = (A_j, p_{ij})$ is a service offer for invocation of service j as a successor of service i. A service provider s announces a matrix of bids $B^s \in \mathcal{B}$ for all incoming edges to every service it owns:

Definition 3.4 [Multidimensional Service Offer]. *A multidimensional service offer is a matrix of bids of the form:*

$$B^s := \begin{cases} b_{ij} = (A_j, p_{ij}), & i \in \tau(j), j \in \sigma(s) \\ (\bar{A}_j, -\infty), & \textit{otherwise} \end{cases} \tag{3.7}$$

with $\tau(v)$ denotes the set of all predecessor services to service v with $\tau : V \to V$ and $\sigma(s)$ the set of all services owned by service provider s. $\bar{A}_j$ is an arbitrary service configuration.

[6] Note that there are no preceeding service offers for services v with $v \in Y_1$.

Example 3.4 [MULTIDIMENSIONAL SERVICE OFFER]. *Recall, the computation of a scoring function is illustrated in Example 3.2. This example is extended with respect to internal costs that occur on the provider side for the invocation of a service offer in a certain context. Figure 3.7 shows the extended service value network.*

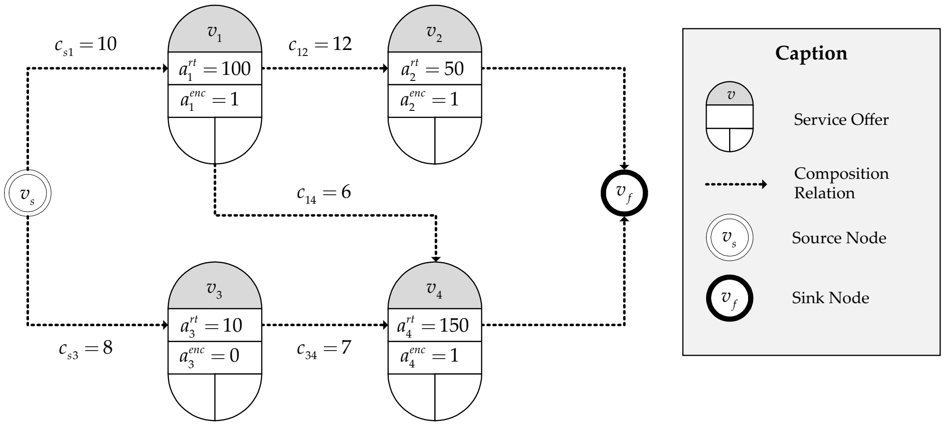


Figure 3.7: Service value network with service offers and internal costs.

It is assumed that service offers v_1 and v_4 are owned by a service provider s_1 and service offers v_2 and v_3 are owned by another service provider s_2. Therefore, the ownership information $\sigma(s_1) = \{v_1, v_4\}$ and $\sigma(s_2) = \{v_2, v_3\}$ is public knowledge. For simplicity, it is further assumed that service providers follow a truth-telling strategy, that is, they report their multidimensional types truthfully. According to Definition 3.4 the service offer bid matrixes for service providers s_1 and s_2 evolve as follows:

$$B^{s_1} = \begin{pmatrix} -\infty & ((100,1),10) & -\infty & -\infty & -\infty & -\infty \\ -\infty & -\infty & -\infty & -\infty & ((150,1),6) & -\infty \\ -\infty & -\infty & -\infty & -\infty & -\infty & -\infty \\ -\infty & -\infty & -\infty & -\infty & ((150,1),7) & -\infty \\ -\infty & -\infty & -\infty & -\infty & -\infty & -\infty \\ -\infty & -\infty & -\infty & -\infty & -\infty & -\infty \end{pmatrix}$$

$$B^{s_2} = \begin{pmatrix} -\infty & -\infty & -\infty & ((10,0),8) & -\infty & -\infty \\ -\infty & -\infty & ((50,1),12) & -\infty & -\infty & -\infty \\ -\infty & -\infty & -\infty & -\infty & -\infty & -\infty \\ -\infty & -\infty & -\infty & -\infty & -\infty & -\infty \\ -\infty & -\infty & -\infty & -\infty & -\infty & -\infty \\ -\infty & -\infty & -\infty & -\infty & -\infty & -\infty \end{pmatrix}$$

3.3 Mechanism Implementation

To design a procurement auction for complex services we follow the approach of *algorithmic mechanism design* as introduced in [NR01]. The discipline of mechanism design forms a subset of game theory that focuses on solving social choice problems from an engineering perspective accounting for technical constraints and preconditions. The central objective is to maximize the system's welfare by allocating adequate service offers from a set of decentralized, self-interested and rationally acting service providers. All service providers have private information about their internal costs and the quality of their services representing the providers' multidimensional types. The challenge is to design a mechanism $m = (o, t)$ consisting of an allocation function o and a transfer function t that incentivizes service providers to report their types truthfully to the auctioneer with respect to all dimensions of all their service offerings. Such truthful information is necessary in order to achieve the system-wide solution as desired. The allocation outcome of such a mechanism yields the same solution as the overall problem based on the same social choice in a fictive setting with complete information about the agents' types.

The auctioneer has to solve the problem of allocating a path f^* from source to sink connecting selected service offers within the network G that yields the highest welfare as the sum of all utilities (consumer and provider surpluses). The main challenge in such a setting is that types are private information to service providers. Therefore the auctioneer is not capable of solving the welfare maximization problem directly but instead has to implement adequate incentives to make truth-telling a dominant strategy equilibrium.

3.3.1 Allocation

Let $\mathcal{U}_f$ denote the overall utility of path f based on the reported types. Let further $\mathcal{P}_f$ be the sum of all price bids for allocated service offers on the path f such that $\mathcal{P}_f = \sum_{e_{ij} \in f} p_{ij}$. The allocation function $o : \mathcal{B} \to F$ maps the service providers' bids $B \in \mathcal{B}$ – their reported types – to a feasible path from source to sink $f^* \in F$[7] such that:

$$o(B) := \operatorname*{argmax}_{f \in F} \mathcal{U}_f = \operatorname*{argmax}_{f \in F} \left(\alpha \mathcal{S}(\mathcal{A}_f) - \mathcal{P}_f \right) \tag{3.8}$$

[7]For the sake of simplicity, the expression "allocated service offer" means that this service offer has an incoming edge that is entailed in the allocated set of edges f^*. Analogously, the expression "allocated service provider" means that a service provider owns at least one "allocated service offer".

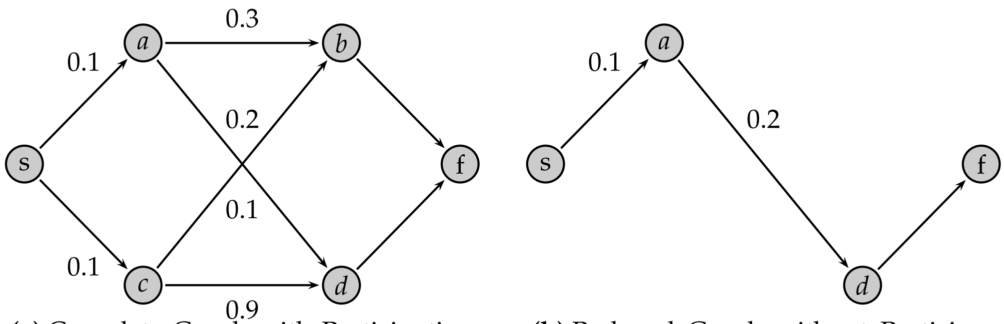
Having defined an allocation function to perform a desired social choice that selects a set of edges within G that determine the instance of the complex service, a function that specifies monetary transfers to service providers has to be designed. Let $\mathcal{U}^*$[8] denote the overall utility of the allocated path meaning the utility of the path f^*, which maximizes the overall utility. Furthermore, let $\mathcal{U}^*_{-s}$ denote the overall utility of a path f^*_{-s} that yields the maximum welfare in a reduced graph G_{-s} without every service owned by service provider s and without incoming and outgoing edges of these service offers, i.e. the complex service instance that maximizes welfare in an service value network without service provider s's participation.

Definition 3.5 [Critical Value]. *The* critical value $\Delta t^{crit,s}$ *of a service provider s represents its contribution to the system as the difference between the overall utility $\mathcal{U}^*$ in the complete graph and the overall utility in the reduced graph $\mathcal{U}^*_{-s}$ without service offers owned by service provider s and incoming and outgoing edges of these services such that*

$$\Delta t^{crit,s} = \mathcal{U}^* - \mathcal{U}^*_{-s} \tag{3.9}$$

The following example shows the computation of service provider s's contribution to the system.

Example 3.5 [Critical Value and Individual Contribution]. *The service value network in Figure 3.8a consists of four service offers a, b, c and d and source and sink nodes s and f. Service provider s_1 owns two services b and c such that $\sigma(s_1) = \{b, c\}$. For simplicity there are no quality attributes of service offers, which implies one dimensional types of service providers.*



(a) Complete Graph with Participation of Service Provider z

(b) Reduced Graph without Participation of Service Provider z

Figure 3.8: Critical value and individual contribution.

Values on the edges within the graph denote price bids of service providers for all incoming edges of service offers they own. Focusing on service provider s_1, there are bids $b_{ab} = 0.3$, $b_{cb} = 0.2$ and $b_{sc} = 0.1$. Assuming a service requester's willingness to pay of

[8]For the reader's convenience, the notion $\mathcal{U}^*$ is short for $\mathcal{U}_{o(B)}$ which denotes the overall utility of the path f^* allocated by $o(B)$ based on service providers' bids.

α the path $f^* = \{e_{sc}, e_{cb}, e_{cf}\}$ is allocated by $o(B)$ as it yields the highest overall utility of $\mathcal{U}^* = \alpha - 0.2$, which represents the highest welfare.

In order to determine service provider s_1's critical value $\Delta t^{crit,s_1}$ – i.e. s_1's utility contribution to the system – the overall utility $\mathcal{U}^*_{-s_1}$ in the reduced graph depicted in Figure 3.8b without s_1's participation is computed. In the absence of service provider s_1's service offers b and c only a single path from source to sink remains. Hence, the path $f^*_{-s_1} = \{e_{sa}, e_{ad}, e_{df}\}$ is allocated and represents the only feasible complex service instance which results in an overall utility of $\mathcal{U}^*_{-s_1} = \alpha - 0.3$.

Consequently the critical value evolves as $\Delta t^{crit,s_1} = 0.1$, which represents service provider s_1's contribution the overall system.

3.3.2 Transfer

Every service provider s receives a monetary transfer t^s for all services s owns that are allocated by $o(B)$. Analogue to the idea of a second-price auction, a monetary compensation $t^s - \sum_{e_{ij}|e_{ij}\in o, j\in\sigma(s), i\in\tau(j)} p_{ij}$ for service provider s that owns service offers $j \in \sigma(s)$ corresponds to the monetary equivalent of the utility gap between the allocated path and the allocated path in the reduced graph without s and all its incoming and outgoing edges, i.e the critical value of service provider s. In other words the additional payment $t^s - \sum_{e_{ij}|e_{ij}\in o, j\in\sigma(s), i\in\tau(j)} p_{ij} \geq 0$ is a monetary equivalent to the utility service provider s contributes to the overall utility of the system. Thus, the transfer t^s represents the price that service provider s could have charged without loosing its participation in the winning allocation:

$$
\begin{aligned}
\mathcal{U}^* - \mathcal{U}^*_{-s} &= t^s - \sum_{e_{ij}|e_{ij}\in o, j\in\sigma(s), i\in\tau(j)} p_{ij} \\
t^s &= \sum_{e_{ij}|e_{ij}\in o, j\in\sigma(s), i\in\tau(j)} p_{ij} + (\mathcal{U}^* - \mathcal{U}^*_{-s}) \\
t^s &= \sum_{e_{ij}|e_{ij}\in o, j\in\sigma(s), i\in\tau(j)} p_{ij} + \Delta t^{crit,s}
\end{aligned}
$$

Consequently, the transfer function t^s for service provider s is defined as

$$
t^s := \begin{cases} \sum_{i\in\tau(j)} \sum_{j\in\sigma(s)} p_{ij} + (\mathcal{U}^* - \mathcal{U}^*_{-s}), & \text{if } e_{ij} \in o \\ 0, & \text{otherwise} \end{cases} \tag{3.10}
$$

The transfer function belongs to the class of VCG-based payment schemes which implements valuable mechanism properties that are extensively analyzed in Chapter 5.

Costs c^s that service provider s has to bear for performing offered and allocated services result accordingly:

$$c^s := \begin{cases} \sum_{i \in \tau(j)} \sum_{j \in \sigma(s)} c_{ij}, & \text{if } e_{ij} \in o \\ 0, & \text{otherwise} \end{cases} \tag{3.11}$$

3.3.3 Summary

The goal of the mechanism implementation is to incentivize service providers to report their types truthfully to the auctioneer. This fosters a system-wide solution in a decentralized environment that maximizes welfare among all participants although they are assumed to act self-interested. The properties of the implemented social choice are extensively analyzed in Chapter 5.

Summarizing the presented mechanism implementation for the complex service auction, Figure 3.9 depicts the mechanism implementation triangle underlaying the complex service auction.

3.4 Related Work

Recently, an enormous body of work has been done that blurs the border between game theory and computer science [Pap01]. Especially the discipline of mechanism design that focuses on the problem to coordinate self-interested participants in pursuing an overall goal are introduced by [NR01]. The authors design suitable mechanisms to standard optimization problems in the area of task scheduling and routing. In incentive compatible mechanisms agents are incentivized to choose the strategy of revealing their true type. Incentive compatible mechanisms such as the celebrated Vickrey-Clarke-Groves (VCG) mechanism are firstly introduced and extensively investigated by [Vic61, Cla71, Gro73, GL78].

Most of the research has been done with respect to truth-telling of one-dimensional types. The field of designing incentive compatible mechanisms, that induce truth-telling of multidimensional properties of goods or services, still lacks deeper research. A thorough analysis and investigation in the area of multidimensional optimal auctions and the design of optimal scoring rules has been done by [CIoWM93, Bra97, AC05]. An investigation of the winner determination problem in

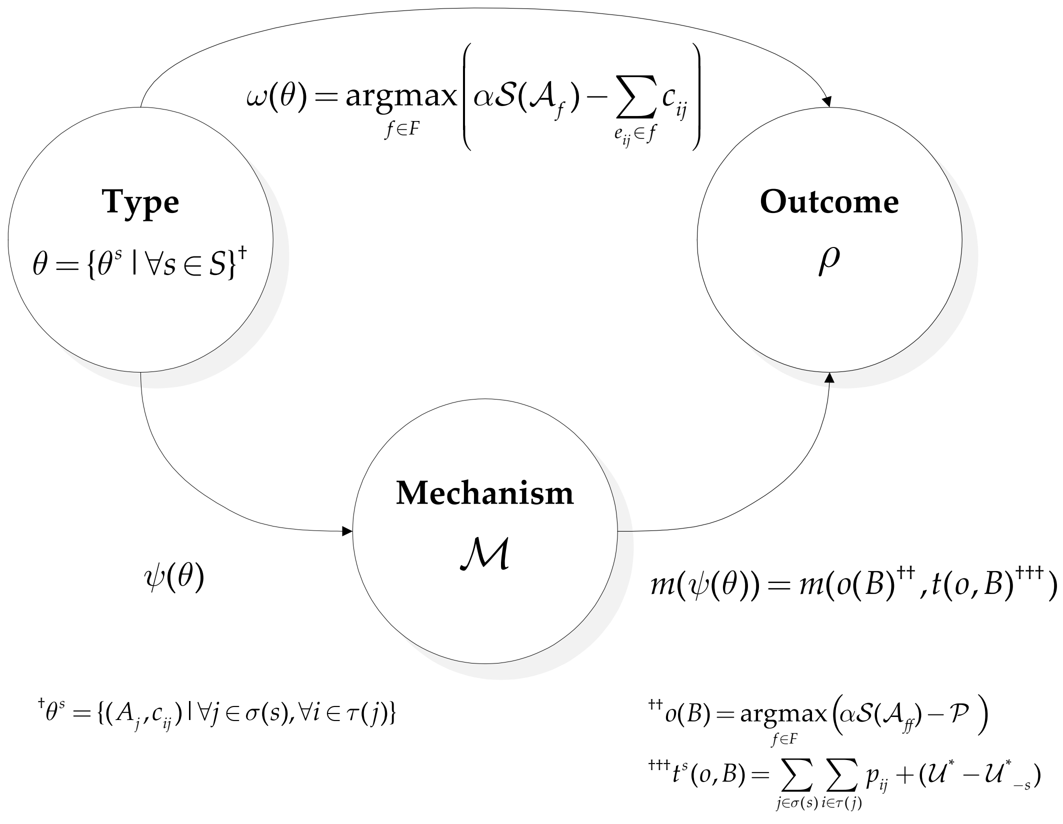


Figure 3.9: Triangle relation of the CSA mechanism implementation and social choice.

configurable multiattribute auctions from an operational research perspective without accounting for mechanism design aspects such as incentive compatibility has been done in [BK05]. In [PK02, PK05], iterative multiattribute procurement auctions are introduced while focusing on mechanism design issues and on solving the multiattribute allocation problem. Preferences for multidimensional goods and multidimensional types in scoring auctions are extensively investigated in [AC08] and extended to combinatorial auctions in [MPW08]. Nevertheless their work does not consider compositions and sequences of services as well as their technical feasible interrelations in order to coordinate value generation. All of these approaches assume bundles of goods in scenarios where the sequence and order does not matter and therefore cannot be applied to composite services that only fulfil their objectives in the right sequence of composition.

Nevertheless, combinatorial auctions yield major drawbacks regarding computational feasibility that result from an NP-hard complexity. Computational feasibility implies a trade-off between optimality and valuable mechanism properties such as incentive compatibility. Several authors propose approximate solutions for incentive compatible mechanisms to overcome issues of computational complexity [aN08, NR07, Ron01, RL05]. Path auctions as a subset of combinatorial auctions reduce complexity through predefining all feasible service combinations in an underlying graph topology and are investigated by [FRS06, HS01, AT07]. In their work, path auctions are utilized for pricing and routing in networks of resources such as computation or electricity. Application-related issues of auctions to optimal routing are examined by [FCSS05, MT07]. All of these approaches deal with the utility services layer according to the service classification by [BS08, BBS08] and hence do not cover the problems related to elementary services and complex services.

3.5 Auction Process Model & Architecture

The auction conduction is divided in two main phases: a *solicitation phase* and the actual *auction phase* as depicted in Figure 3.10.

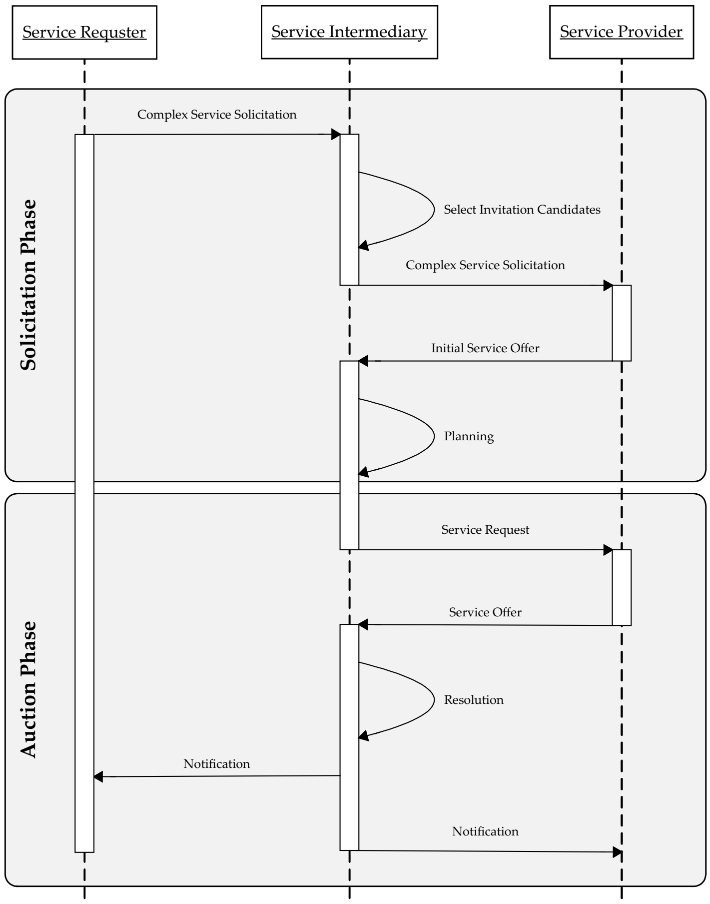


Figure 3.10: Process model of the CSA.

The solicitation phase serves as an initial screening phase regarding the service request and potential service provider candidates to be invited to participate in the auction. The service requester sends a *complex service solicitation* to the service intermediary which initiates the coordination process. The complex service solicitation specifies required modularized functionality which determines the candidate pools that are sequentially involved in the production of the complex service requested.

Based on this information, the service intermediary reasons about potential service providers to be invited to participate in the auction phase. There are different forms of finding and defining suitable participants. The service intermediary can step into the role of pushing the invitation process using e.g. a registry to find adequate service providers. It is also possible to reverse the roles in such a lookup scenario, meaning that potential participants are proactively searching for suitable coordination services provided by a service intermediary. Potential participants could also subscribe to a notification service – analogue to the observer design pattern – in order to automatically be informed if an adequate auction service is available.

Having defined the set of potential service providers to participate in the auction phase, the service intermediary sends out the complex service solicitation and additional information as an invitation to the candidates. This information enables service providers to register their service offerings to be part of the service value network and to be considered in the auction phase by sending *initial service offers*.

Combining the information about the complex service solicitation and the initial service offers from service providers, the service intermediary *plans* the topology of the service value network and proceeds its virtual formation (cp. Section 2.1.4 and Section 3.1). This step concludes the solicitation phase and lays the basis to the actual auction phase.

The auction phase embodies the central coordination process to allocate and price complex services. Messages and information objects exchanged during the auction conduction are fully specified according to the bidding language in Section 3.2. The topology information about the service value network as well as the requester's preferences and willingness to pay is sent as a *service request* (cp. Section 3.2.2) to registered service providers. Having received the requester's information, the service providers privately submit their *service offers* – as specified in Section 3.2.3 – to the service intermediary. Having collected necessary information from requester and provider side, the service intermediary *resolves* the auction by computing the winner determination and resulting monetary transfers. The auction process concludes with notifications about the final outcome and corresponding transfers sent to the service requester and the service providers.

Providing an architectural overview, Figure 3.11 shows service providers that intent to participate in the auction, their service offers which are realized in a lightweight manner and necessary big Web services that enable the overall coordination of the auction process.

The CSA platform as the central coordination unit communicates with potential participants via a *coordinator service* implemented as a Web service with a WSDL interface. Analogously, each service provider exposes a *participant service* for the

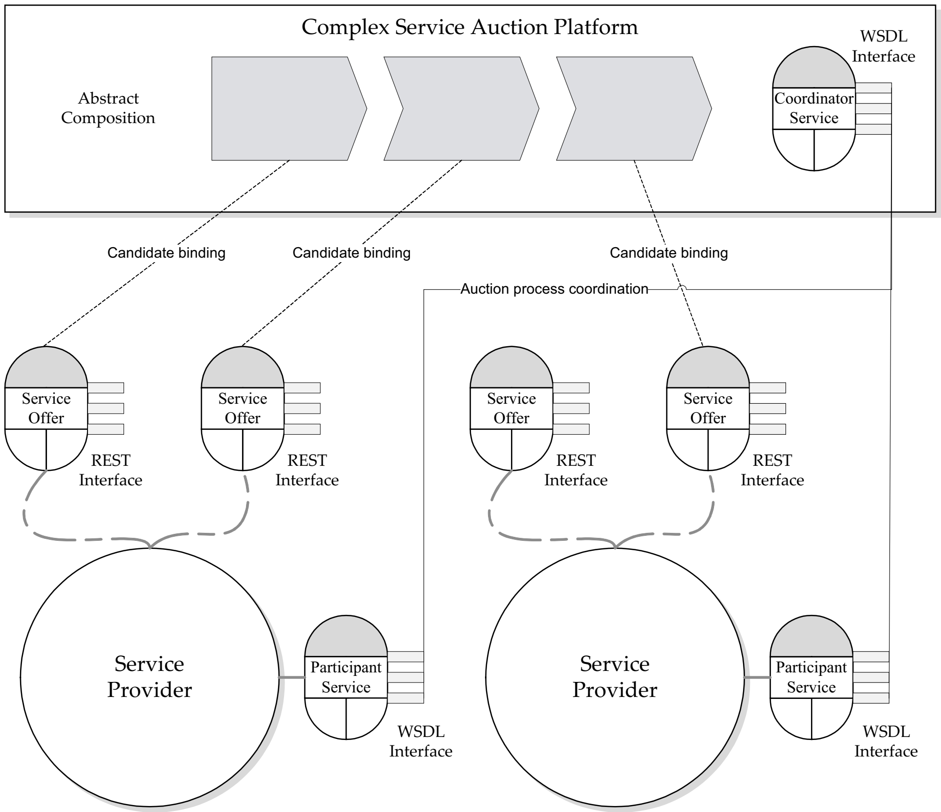


Figure 3.11: Architectural overview of the CSA.

message exchange with the coordinator. After the coordination phase is completed, concrete candidate service instances are bound to each step in the abstract composition in a lightweight manner leveraging the simplicity of REST/HTTP interfaces. The final composition embodies the outcome of the coordination process in the form of a concrete complex service instance.

3.6 Realization & Implementation

This section provides an in-depth analysis of the *ComputeAllocation* algorithm which performs the winner determination in the complex service auction. Special challenges that result from aggregation operations such as min and max as well as Boolean operations which are used in the context of semantic QoS extensions (cp. Section 4.3) are outlined and adequate remedies are discussed. The procedure of the algorithm is illustrated stepwise by means of an extensive example. Furthermore, this section introduces a prototypical implementation of a service value network planner tool and an agent-based simulation tool to analyze the complex service auction.

From an algorithmic mechanism design perspective computational feasibility according to Requirement 5 is a central desideratum in order to implement the mechanism in an online system which requires on-the-fly computation at run-time.

It is well-known that solving the winner determination problem in general combinatorial auctions is $\mathcal{NP}$-complete. Focusing on finding efficient computational approaches, several algorithms have been proposed to solve the winner determination problem [PS98, RPH98, SSGL05].

The solution to the allocation problem in (3.8) can be compute in polynomial time using well-known graph algorithms to determine the “shortest” path within a network such as the Dijkstra algorithm [Dij59].

According to the payment scheme in (3.11) the allocation must be computed twice for each allocated service offer – based on the graph with the service offerings of the service provider receiving the payment and without its participation. In the second case the graph can be preprocessed and reduced by all service offerings owned by the service provider that receives the payment. After the reduction the allocation can be computed accordingly which yields the same time complexity.

Nevertheless, the extension of the complex service auction with respect to complex QoS aggregation using also aggregation operations that require complete information about predecessors’ attribute values – memory-dependent attribute types (e.g. cp. Section 4.3) – such as min, max and Boolean operations may result in suboptimal solutions using the traditional Dijkstra algorithm. Analogue to the problem of negative edge weights which is well-known in literature [Dij59], memory-dependent operations may result in non-monotone utility characteristics. Such behavior conflicts with the main procedure of the Dijkstra algorithm, that is, it truncates a sub-path which is directly dominated by another sub-path that intersects it at the point of intersection. Considering an attribute type *encryption* which is aggregated by a Boolean `AND` operation according to Table 3.1. A sub-path f_s^1 dominates another sub-path f_s^2 as it yields a higher utility which results from an aggregated value for encryption of `TRUE`. In case both sub-paths intersect at a certain node, the Dijkstra algorithm only considers f_s^1 and drops f_s^2 as f_s^1 yields a higher overall utility so far. Nevertheless, this might be error prone if the subsequent service offer does not support encryption which leads to an aggregated encryption value for f_s^1 of `FALSE`. Hence, the former decision of dropping f_s^2 might have been incorrect since now both sub-paths are not encrypted and f_s^2 might dominate f_s^1 in price.

To overcome illustrated shortcomings of the Dijkstra algorithm, Algorithm 3.1 accounts for attribute types which are aggregated by memory-dependent operations always yielding an optimal solution.

Algorithm 3.1 ComputeAllocation

Require: V, E, B
1: $Q \leftarrow \mathit{getNodesPoolWise}(V)$
2: **for all** $u \in Q$ **do**
3: $\quad \mathit{states}[u] \leftarrow \mathit{getNonMonotoneStates}(u)$
4: $\quad$ **for all** $w \in \mathit{states}[u]$ **do**
5: $\quad\quad \mathit{utility}[u][w] \leftarrow -\infty$
6: $\quad\quad \mathit{path}[u][w] \leftarrow \emptyset$
7: **while** $\mathit{getNextNode}(Q) \neq \mathit{null}$ **do**
8: $\quad u \leftarrow \mathit{getNextNode}(Q)$
9: $\quad \mathit{removeNode}(u, Q)$
10: $\quad$ **for all** $v \in \mathit{getSuccesors}(u, E)$ **do**
11: $\quad\quad$ **for all** $w \in \mathit{states}[u]$ **do**
12: $\quad\quad\quad \bar{w} \leftarrow \mathit{computeState}(w, e_{uv}, B)$
13: $\quad\quad\quad \mathit{altUtility} \leftarrow \mathit{computeUtility}(\mathit{path}[u][w] \cup \{e_{uv}\}, B)$
14: $\quad\quad\quad$ **if** $\mathit{altUtility} > \mathit{utility}[v][\bar{w}]$ **then**
15: $\quad\quad\quad\quad \mathit{utility}[v][\bar{w}] \leftarrow \mathit{altUtility}$
16: $\quad\quad\quad\quad \mathit{path}[v][\bar{w}] \leftarrow \mathit{path}[u][w] \cup \{e_{uv}\}$
17: $w^* \leftarrow \text{argmax}_{w \in \mathit{states}[v_f]} (\mathit{utility}[v_f][w])$
18: **return** $\mathit{path}[v_f][w^*]$

In order to describe the procedure of the *ComputeAllocation* algorithm and its complexity, Algorithm 3.1 is divided into 3 parts, namely the *initialization phase* (lines 1-6), the *main phase* (lines 7-16) and the *consolidation phase* (lines 17-18).

Initialization phase In the initialization phase, required variables are initialized and set to their starting values. In contrary to the traditional Dijkstra algorithm, the *ComputeAllocation* algorithm visits every node within the graph which is equal to the worst-case behavior of a Dijkstra search. Therefore the node queue Q entails all nodes $u \in V$ ordered by the sequence of the candidate pools in the network such that *getNodesPoolWise(V)* $= (u_1^1, \ldots, u_{|Y_1|}^1, \ldots, u_1^K, \ldots, u_{|Y_K|}^K)$[9] with $\{u_1^1, \ldots, u_{|Y_1|}^1\} = Y_1$ and $\{u_1^K, \ldots, u_{|Y_K|}^K\} = Y_K$. The function $\mathit{getNonMonotoneStates}(u)$ retrieves all possible combinations of memory-dependent attribute values of service offer u. Exemplary, if service offer u is only characterized by an encryption attribute type with boolean values, hence $\mathit{getNonMonotoneStates}(u) = \{\texttt{TRUE}, \texttt{FALSE}\}$. Let the set W entail all possible states after aggregation, then the time complexity of the initialization phase is $\mathcal{O}(|V| \cdot |W|)$.

Main phase In the main phase, the algorithm iterates over all nodes in Q and removes each node after processing until there is no entry left in the queue. Each successor v of the current node u is evaluated for all states of u. The

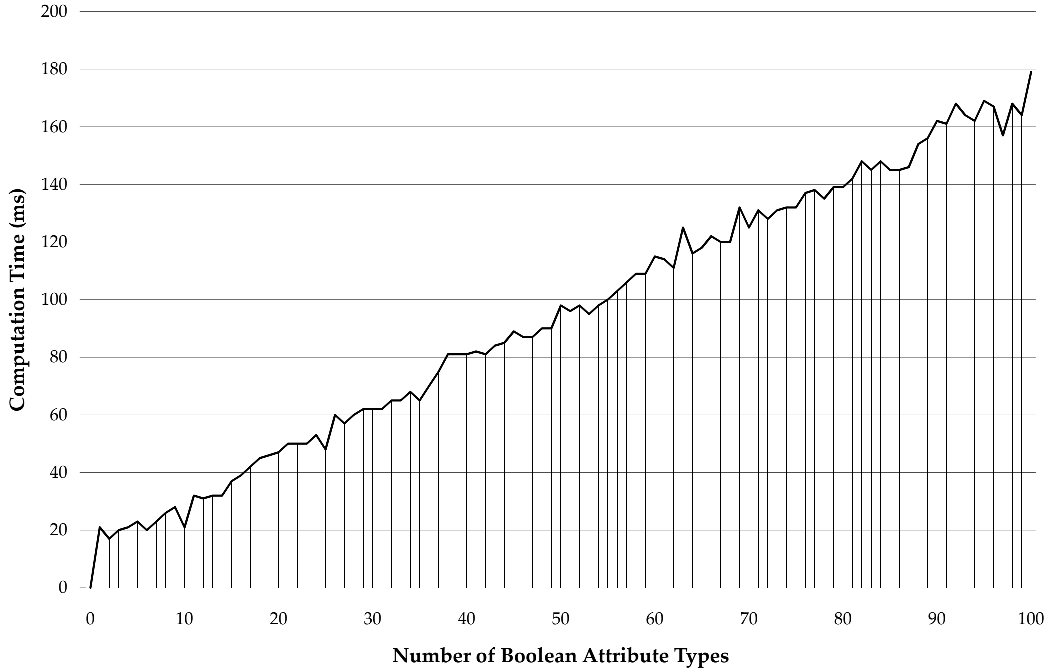
[9] The order within each candidate pool is not important.

utility of the sub-path including v is computed based on the overall utility $\mathcal{U}_f$ introduced in Section 3.3.1. These alternatives are compared to the current utility entry for node v and updated in case of improvement. The variables *utility* and *path* store for each node u and each state the highest utility and the corresponding path respectively. Traversing all successors of every node in Q, the *ComputeAllocation* algorithm processes every edge in the main phase and compares every state of each node. This leads to a time complexity of the main phase of $\mathcal{O}(|E| \cdot |W|)$.

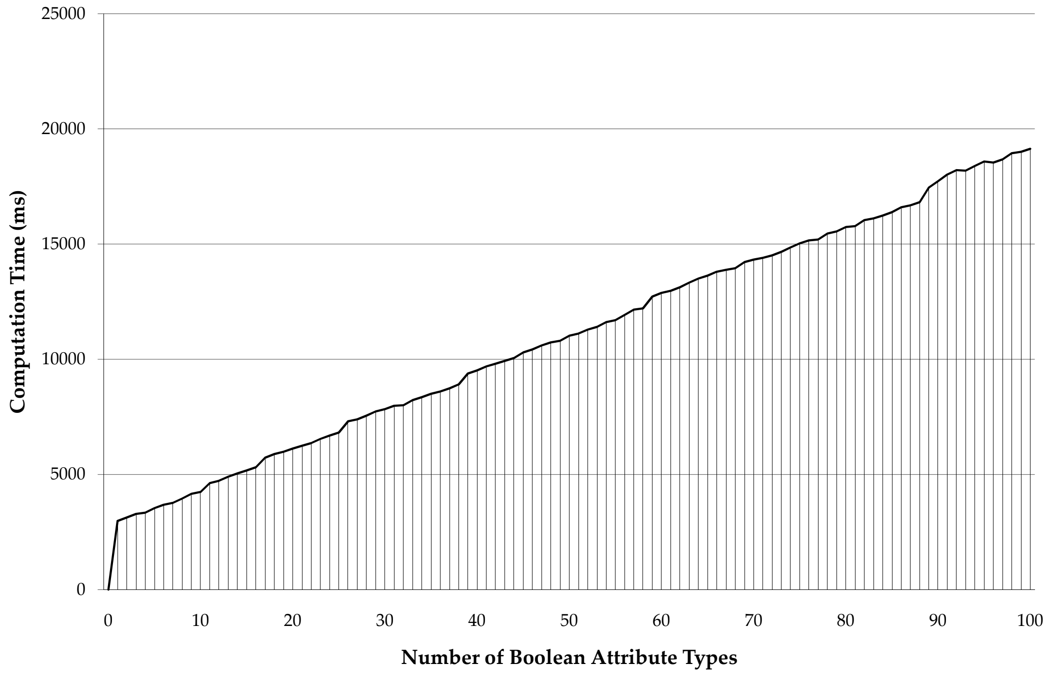
Consolidation phase After the main part has terminated once Q is empty, i.e. all nodes have been processed, the consolidation phase evaluates the results. The path from source to sink is analyzed and the state s^* that maximizes the overall utility is determined. Based on this state the final allocation $path[v_f][s^*]$ is returned. Implemented as a linear search, the consolidation phase yields a time complexity of $\mathcal{O}(|W|)$.

The time complexity of the *ComputeAllocation* algorithm consisting of the initialization phase, the main phase and the consolidation phase evolves as $\mathcal{O}(|V| \cdot |W| + |E| \cdot |W| + |W|)$. Assuming a worst case number of edges with respect to the number of nodes $|E|$ can be substituted by $(\frac{|V|-2}{2})^2 + (|V| - 2)$. This leads to an overall complexity of $\mathcal{O}(|W| \cdot |V|^2)$. The time complexity regarding the number of service offers and connecting edges, the number of paths respectively, is polynomial which means that the algorithms run-time is robust with respect to a changing number of participants and feasible complex service instances. In contrary to the $\mathcal{NP}$-complete complexity in general combinatorial auctions this is a valuable achievement that enables the conduction of the complex service auction in online systems.

Nevertheless, with respect to the number of memory-dependent attribute types and the number of their discrete values, the computational complexity is exponential (e.g. assuming N Boolean attribute types, $|W| = 2^N$). From a domain-specific perspective, the impact of this theoretical result is rather weak, as the number of states that have to be iterated by the algorithm decreases rapidly in the average case. Figure 3.12 illustrates the run-time performance of the *ComputeAllocation* algorithm in a scenario with 100 service offers in 10 candidate pools (cp. Figure 3.12a) and 1000 service offers in 100 candidate pools (cp. Figure 3.12b). The service value network is assumed to be fully connected which means that each service offer has the maximum number of incoming edges which results in the maximum number of feasible paths from source to sink. The algorithm's performance is evaluated dependent on the number of memory-dependent attribute types. Attribute types are assumed to be Boolean and their values are uniformly distributed for each service offer. Although the theoretical worst case analysis of the computational complexity is exponential with respect to the number N of memory-dependent attribute types ($\mathcal{O}(2^N)$), the average case with boolean attribute types results in a linear increasing computation time. The *ComputeAllocation* algorithm quickly solves the winner determination problem even for huge instances and satisfies Requirement 5 (computational tractability).



(a) Performance analysis with 100 service offers in 10 candidate pools.



(b) Performance analysis with 1000 service offers in 10 candidate pools.

Figure 3.12: Performance analysis of the *ComputeAllocation* algorithm.

Example 3.6 [ALLOCATION COMPUTATION WITH MEMORY-DEPENDENT QOS]. *This example illustrates the procedure of the* ComputeAllocation *algorithm in a stepwise manner based on the service value network as depicted in Figure 3.13.*

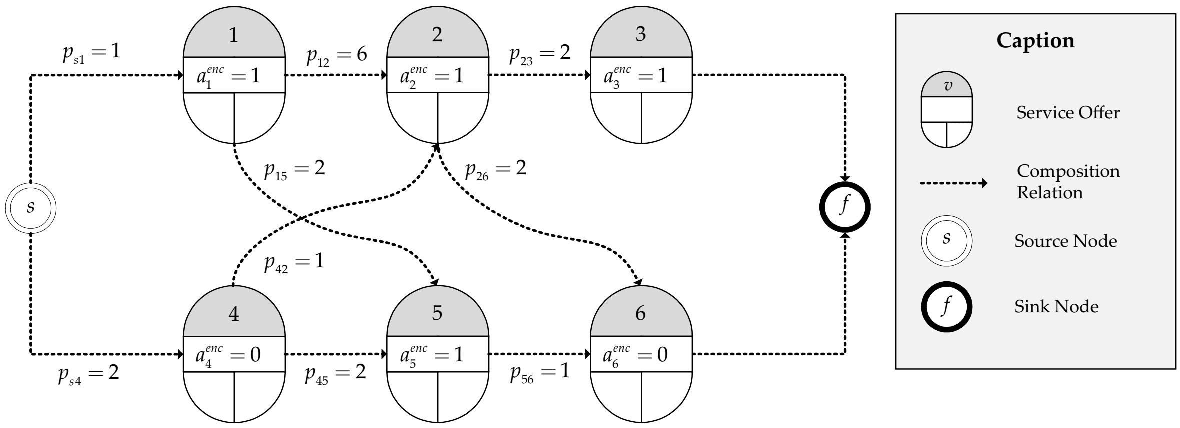


Figure 3.13: Service value network with service offers exposing memory-dependent attribute types.

The service value network consists of 6 service offers $V = \{1,2,3,4,5,6\} \cup \{s,f\}$. Each service offer u is unambiguously configured through a boolean attribute value a_u^{enc} for the attribute type encryption whereas $1 \equiv$ `TRUE` and $0 \equiv$ `FALSE`. Values on incoming edges p_{ij} represent price bids of service providers. It is assumed that the service requester's willingness to pay $\alpha\mathcal{S}(\mathcal{A}_f)$ for a complex service depending on its QoS characteristics $\mathcal{A}_f$ evolves as

$$\alpha\mathcal{S}(\mathcal{A}_f) = \begin{cases} 15, & \text{if } \mathcal{A}_f = 1 \\ 12, & \text{if } \mathcal{A}_f = 0 \end{cases}$$

Table 3.2 illustrates the algorithm's procedure to find an optimal allocation based on the allocation function in Section 3.3.1 accounting for the memory-dependent attribute type encryption representing the QoS of service offers.

In the last step when node f is processed, the optimal path given a not encrypted complex service results as $f^*_{FALSE} = \{e_{s1}, e_{15}, e_{56}, e_{6f}\}$ and yields an overall utility of $\mathcal{U}_{f^*_{FALSE}} = 8$. Given a encrypted complex service, the optimal allocation is $f^*_{TRUE} = \{e_{s1}, e_{12}, e_{23}, e_{3f}\}$ with an overall utility of $\mathcal{U}_{f^*_{TRUE}} = 6$. Thus, the state $s^* =$ `FALSE` yields an optimal path $f^* = \{e_{s1}, e_{15}, e_{56}, e_{6f}\}$ that maximizes the system's overall utility $\mathcal{U}^* = 8$.

Table 3.2: Allocation computation stepwise procedure example.

Node			**-**	**s**	**1**	**4**	**2**	**5**	**3**	**6**	**f**
Q			$\{s,1,4,2,5,3,6,f\}$	$\{1,4,2,5,3,6,f\}$	$\{4,2,5,3,6,f\}$	$\{2,5,3,6,f\}$	$\{5,3,6,f\}$	$\{3,6,f\}$	$\{6,f\}$	$\{f\}$	$\emptyset$
utility *path*	s	`TRUE`	**15** $\emptyset$	**15** $\emptyset$	**15** $\emptyset$	**15** $\emptyset$	**15** $\emptyset$	**15** $\emptyset$	**15** $\emptyset$	**15** $\emptyset$	**15** $\emptyset$
utility *path*	s	`FALSE`	**12** $\emptyset$	**12** $\emptyset$	**12** $\emptyset$	**12** $\emptyset$	**12** $\emptyset$	**12** $\emptyset$	**12** $\emptyset$	**12** $\emptyset$	**12** $\emptyset$
utility *path*	1	`TRUE`	$-\infty$ $\emptyset$	**14** $\{e_{s1}\}$	**14** $\{e_{s1}\}$	**14** $\{e_{s1}\}$	**14** $\{e_{s1}\}$	**14** $\{e_{s1}\}$	**14** $\{e_{s1}\}$	**14** $\{e_{s1}\}$	**14** $\{e_{s1}\}$
utility *path*	1	`FALSE`	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$
utility *path*	2	`TRUE`	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	**8** $\{e_{s1},e_{12}\}$	**8** $\{e_{s1},e_{12}\}$	**8** $\{e_{s1},e_{12}\}$	**8** $\{e_{s1},e_{12}\}$	**8** $\{e_{s1},e_{12}\}$	**8** $\{e_{s1},e_{12}\}$	**8** $\{e_{s1},e_{12}\}$
utility *path*	2	`FALSE`	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	**9** $\{e_{s4},e_{42}\}$	**9** $\{e_{s4},e_{42}\}$	**9** $\{e_{s4},e_{42}\}$	**9** $\{e_{s4},e_{42}\}$	**9** $\{e_{s4},e_{42}\}$	**9** $\{e_{s4},e_{42}\}$
utility *path*	3	`TRUE`	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	**6** $\{e_{s1},e_{12},e_{23}\}$	**6** $\{e_{s1},e_{12},e_{23}\}$	**6** $\{e_{s1},e_{12},e_{23}\}$	**6** $\{e_{s1},e_{12},e_{23}\}$	**6** $\{e_{s1},e_{12},e_{23}\}$
utility *path*	3	`FALSE`	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	**7** $\{e_{s4},e_{42},e_{23}\}$	**7** $\{e_{s4},e_{42},e_{23}\}$	**7** $\{e_{s4},e_{42},e_{23}\}$	**7** $\{e_{s4},e_{42},e_{23}\}$	**7** $\{e_{s4},e_{42},e_{23}\}$
utility *path*	4	`TRUE`	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$
utility *path*	4	`FALSE`	$-\infty$ $\emptyset$	**10** $\{e_{s4}\}$	**10** $\{e_{s4}\}$	**10** $\{e_{s4}\}$	**10** $\{e_{s4}\}$	**10** $\{e_{s4}\}$	**10** $\{e_{s4}\}$	**10** $\{e_{s4}\}$	**10** $\{e_{s4}\}$
utility *path*	5	`TRUE`	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	**12** $\{e_{s1},e_{15}\}$	**12** $\{e_{s1},e_{15}\}$	**12** $\{e_{s1},e_{15}\}$	**12** $\{e_{s1},e_{15}\}$	**12** $\{e_{s1},e_{15}\}$	**12** $\{e_{s1},e_{15}\}$	**12** $\{e_{s1},e_{15}\}$
utility *path*	5	`FALSE`	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	**8** $\{e_{s4},e_{45}\}$	**8** $\{e_{s4},e_{45}\}$	**8** $\{e_{s4},e_{45}\}$	**8** $\{e_{s4},e_{45}\}$	**8** $\{e_{s4},e_{45}\}$	**8** $\{e_{s4},e_{45}\}$
utility *path*	6	`TRUE`	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\{e_{s1},e_{12},e_{26}\}$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$
utility *path*	6	`FALSE`	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	**7** $\{e_{s4},e_{42},e_{26}\}$	**8** $\{e_{s1},e_{15},e_{56}\}$	**8** $\{e_{s1},e_{15},e_{56}\}$	**8** $\{e_{s1},e_{15},e_{56}\}$	**8** $\{e_{s1},e_{15},e_{56}\}$
utility *path*	f	`TRUE`	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	**6** $\{e_{s1},e_{12},e_{23},e_{3f}\}$	**6** $\{e_{s1},e_{12},e_{23},e_{3f}\}$	**6** $\{e_{s1},e_{12},e_{23},e_{3f}\}$
utility *path*	f	`FALSE`	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	$-\infty$ $\emptyset$	**7** $\{e_{s4},e_{42},e_{23},e_{3f}\}$	**8** $\{e_{s1},e_{15},e_{56},e_{6f}\}$	**8** $\{e_{s1},e_{15},e_{56},e_{6f}\}$

Chapter 4

Applicability Extensions

The management of QoS metrics directly impacts the success of organizations participating in e-commerce.

[CSM+04]

This section introduces design extensions to the complex service auction to enable the applicability in service value networks in order to coordinate distributed activities in creating and provisioning complex services to customers. A compensation transfer function is introduced in Section 5.1.2. The auction conduction is divided in a declaration phase and an execution phase in order to incorporate ex-post information on provided QoS levels (monitoring information) into the monetary transfers which are distributed among participating service providers. Counteracting the absence of budget balance, Section 4.2 introduces the budget-balanced interoperability transfer function (ITF). By sacrificing incentive compatibility to a certain degree, the design of the payment scheme incentivizes service providers to increase their services' degree of interoperability. Properties of the ITF are analyzed in detail in Section 6.2. As quality aspects are gaining importance especially in the context of services, Section 4.3 introduces and rule-based extension to the complex service auction which allows for the description and evaluation of complex QoS characteristics and their incorporation in the allocation and pricing component of the basic mechanism.

4.1 Verification and Service Level Enforcement

In Section 2.1.3.3, the expressiveness of the complex service auction with respect to complex QoS characteristics and their management has been introduced in detail. From a computer science perspective, protocols and algorithms for distributed environments such as the Internet have been designed under the implicit assumption that participants report their information (e.g. the QoS of their service offers) truthfully. This assumption only holds for predefined algorithms and processes that produce a deterministic outcome but not in the context of self-interested service providers that constantly seek to maximize their individual utility while participating in distributed systems.

This section provides an extension for the complex service auction that enhances the transfer function (cp. Section 2.2.3.5) by a compensation function, which on the one hand punishes service providers for untruthful announcements about the QoS of their service offers and on the other hand compensates service requesters for the utility loss they incur due to resulting non-performance.

4.1.1 Related Work

The assumption that service providers only announce attribute values that they actually perform during execution is not realistic [NRTV07]. The basic assumption in traditional mechanism design theory is that agents can follow any of their strategies no matter what their type is[1]. Nevertheless, especially in algorithmic mechanism design, settings are observed in which computer systems can gain *extra information* about the agents and their behavior that can be used in the mechanism. According to [NR01] the mechanism implementation can be divided into two phases: a *declaration phase* and an *execution phase*.

Declaration phase In the declaration phase the service requester and the service providers announce requests and offers according to the bidding language introduced in Section 3.2. The declaration phase predominantly collects information objects exchanged according to the coordination protocol. These information objects represent agents' types which are directly reported to the coordinator. This information which is explicitly announced by the agent, is the only information available to the coordinator at this point of time.

Execution phase Based on the information gathered in the declaration phase, the coordinator allocates a subset of service offers that together form the desired complex service instance. In the execution phase the service offers that have been allocated by the mechanism embody the complex service instance, which is executed sequentially. During this phase the actual realized output of each participant can be observed by the coordinator using monitoring techniques [SMS+02, PBB+04]. Required monitoring tasks can also be outsourced by the

[1] Nevertheless it is obvious that the agents' strategy space is limited due to technological and physical restrictions

coordinator in order to leverage external core competencies [Men02]. Such a scenario enables the coordinator to observe the agents' types with respect to reported QoS attributes and control the actual outcome of offered services. Consequently, payments to allocated agents are transferred *after execution* in order to incorporate ex-post information about the services' performances.

The utilization of the extra information about the agents that can be observed ex-post in the execution phase enables the design of a penalty for deviating from the announced attributes. That is an equivalent monetary penalty component which enhances the transfer function in order to implement a threat based on a punishment for lying about the offered QoS.

4.1.2 Compensation

Let a_j^l be the *announced attribute value* for attribute type l of service j's configuration. Furthermore let $\tilde{a}_j^l$ be the *verified attribute value* for attribute type l realized by service j and monitored during execution. Analogously, A_j and $\tilde{A}_j$ denote announced and verified configurations of service j. Distinguishing between announced and verified attribute values, the overall utility may also differ. Recall that $\mathcal{U}^*$ denotes the *ex-ante overall utility* of the allocated path f^* based on the information available in the declaration phase. Furthermore, $\tilde{\mathcal{U}}^{*s}$ denotes the *ex-post overall utility* that results from the complex service instance formed by allocated service offers on a path f^* and based on the verified attribute values $\tilde{a}_j^1, \ldots, \tilde{a}_j^l$ of all service offers $j \in \sigma(s)$. According to the *Compensation-and-Bonus mechanism* introduced in [NR01] a compensation function $\Delta t^{comp,s}$ is constructed as follows:

$$\Delta t^{comp,s} := (\mathcal{U}^* - \tilde{\mathcal{U}}^{*s}) \tag{4.1}$$

The compensation function represents the *overall utility gap* that results from the utility difference based on the announced attribute values and the verified ones measured after execution. In other words $\Delta t^{comp,s}$ is the utility loss the whole system incurs because of service provider s's untruthful announcement(s). The monetary equivalent to this utility gap represents the penalty payment the untruthful service provider has to bear for deviating from the announced attribute values. This "*negative consequence*" can be interpreted as a contractual penalty for not realizing specified service level agreements[2] as defined in [SB04]. Based on the design of the compensation function the transfer function is extended as follows:

$$t^s := \begin{cases} \sum_{j \in \sigma(s)} \sum_{i \in \tau(j)} p_{ij} + \Delta t^{crit,s} - \Delta t^{comp,s}, & \text{if } e_{ij} \in o \\ 0, & \text{otherwise} \end{cases} \tag{4.2}$$

Example 4.1 [Service Level Verification and Enforcement]. *This example illustrates the effect of untruthful announcements about QoS characteristics on the whole system and the service requester. It further demonstrates how the compensation function counteracts such behavior through imposing a penalty on the causer, which represents the utility loss regarding the whole system while compensating the service requester and retaining the previous level of overall utility.*

Figure 4.1 shows a service value network with four service offers $V = \{1,2,3,4\} \cup \{s,f\}$. *For simplicity it is assumed that each service provider owns a single service offer within the network such that* $\sigma(s_1) = \{1\}, \tau(s_2) = \{2\}, \sigma(s_3) = \{3\}$ *and* $\sigma(s_4) = \{4\}$. *There are two feasible paths from source to sink representing a complex service instance* $f^1 = \{e_{s1}, e_{12}, e_{2f}\}$ *and* $f^2 = \{e_{s3}, e_{34}, e_{4f}\}$. *Each service configuration is characterized by a single attribute value* a^{er} *of the attribute type error rate*[3] *which is aggregated according to Table 3.1. A value for error rate represents the average percentage of failures during execution. Values on incoming edges* p_{ij} *represent price bids of service providers for the corresponding service offer.*

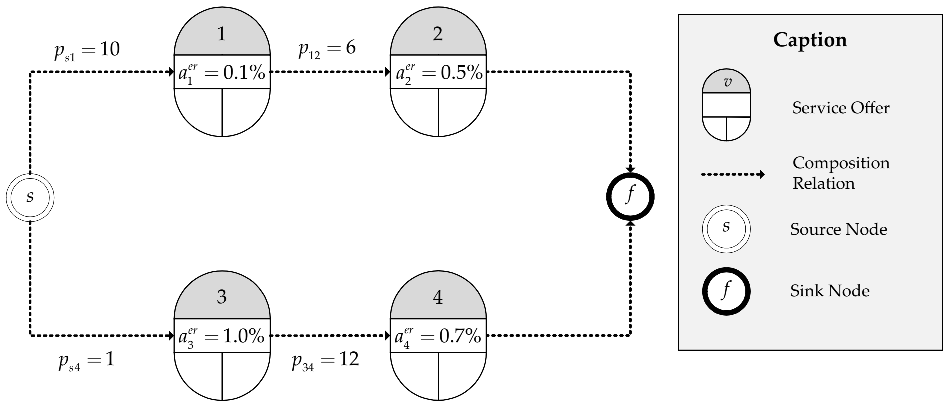


Figure 4.1: Service value network with service offers characterized by error rate quality attributes.

[2] For the design of the verification payment scheme a risk-neutral service requester is assumed. In real-world scenarios a rather risk averse design of SLAs is observable, *overcompensating* service requesters in case of non-performance of service providers.

[3] Error rate describes the ratio of occurred number of failed operations during execution compared to the total number of operations executed by the service.

The analysis of the example scenario is divided into the declaration phase and the execution phase:

Declaration phase (ex-ante) *Service providers announce prices and configurations of the service offers they own (cp. Figure 4.1). The service requester announces a lower boundary* $\gamma_B^{er} = 0.02$ *and an upper boundary* $\gamma_T^{er} = 0$ *which means that an error rate equal or greater than* 2% *yields a utility of* 0 *and an error rate equal to* 0% *results in maximum utility of* 1. *The service requester's willingness to pay for a complex service with score* 1 *is reported as* $\alpha = 50$. *Assuming a linear utility characteristic with respect to error rates between the boundaries, the requester's score for a complex service depending on its QoS evolves as follows:*

$$\mathcal{S}(\mathcal{A}_f) = \|\mathcal{A}_f^{er}\| = \begin{cases} \frac{0.02-\mathcal{A}_f^{er}}{0.02}, & \text{if } 0 < \mathcal{A}_f^{er} < 0.02 \\ 1, & \text{if } \mathcal{A}_f^{er} = 0 \\ 0, & \text{if } \mathcal{A}_f^{er} \geq 0.02 \end{cases}$$

This leads to the following scores for paths f^1 *and* f^2*:*

$$\begin{aligned} \mathcal{S}(\mathcal{A}_{f^1}) &= \frac{0.02 - \max\{0.001, 0.005\}}{0.02} = 0.75 \\ \mathcal{S}(\mathcal{A}_{f^2}) &= \frac{0.02 - \max\{0.01, 0.007\}}{0.02} = 0.5 \end{aligned}$$

The overall utility caused by each allocation consequently is $\mathcal{U}_{f^1} = 50 \cdot 0.75 - 16 = 21.5$ *and* $\mathcal{U}_{f^2} = 50 \cdot 0.5 - 13 = 12$. *As* $\mathcal{U}_{f^1} > \mathcal{U}_{f^2}$ *the upper path is allocated by* $o(B)$. *If transfers would be given in the declaration phase, service provider* s_1 *received* $t^{s_1}_{\textit{ex-ante}} = 10 + (21.5 - 12) = 19.5$ *and service provider* s_2 *received* $t^{s_2}_{\textit{ex-ante}} = 6 + (21.5 - 12) = 15.5$. *This would lead to a service requester's utility of* $\mathcal{U}^R_{\textit{ex-ante}} = 50 \cdot 0.75 - (19.5 + 15.5) = 2.5$.

Execution phase (ex-post) *After the completion of the declaration phase and the final allocation based on the reported types, the complex service instance is executed and the performance of each service component is verified using a monitoring service. The quality announced by service provider* s_1 *for the service offer* 1 *can be confirmed. In contrary, service component* 2 *produces a marginal failure during execution which increases the announced error rate from* 0.5% *to* 0.6%. *The compensation function regarding service offer* 2 *evolves as:*

$$\Delta t^{comp,s_2} = (\mathcal{U}^* - \tilde{\mathcal{U}}^{*s_2})$$

$$= \quad 21.5 - \left(50 \times \frac{0.02 - \max\{0.001, 0.006\}}{0.02} - 16\right) = 2.5$$

Hence, the monetary equivalent to the utility loss caused by service provider s_2 *is* 2.5. *According to the extended transfer function (Equation 4.2), the ex-post transfer for service provider* s_2 *including the penalty is* $t^{s_2}_{ex\text{-}post} = 10 + (21.5 - 12) - 2.5 = 13$. *The decrease in transfer represents the monetary compensation for the loss in quality which compensates the service requester. The service requester's utility is equal to the ex-ante situation as* $\mathcal{U}^R_{ex\text{-}post} = 50 \times 0.7 - (19.5 + 13) = 2.5 = \mathcal{U}^R_{ex\text{-}ante}$.

The service level enforcement extension to the complex service auction satisfies Requirement 8. Incentives provided by the mechanism's extension are central to implement favorable properties with respect to the service providers' multidimensional bids and their services' true QoS characteristics. Such properties are analyzed in detail in Section 5.1.2.

4.2 Achieving Budget Balance

Recall that the mechanism implementation of the complex service auction as introduced in Section 3 consists of a transfer function that pays each service provider z that owns allocated service offers the corresponding price bid and the *critical value* $\Delta t^{crit,z}$ in addition. The critical value represents a monetary equivalent to the provider's utility contribution to the whole system such that $\Delta t^{crit,z} = \mathcal{U}^* - \mathcal{U}^*_{-z}$. Price bids of each service offer that is allocated by the mechanism plus the corresponding critical value has to be payed by the service requester to the service providers. A provider's critical value compensates the individual contribution to the system which depends on the contributions of the other participants. Hence, the payments, the service requester has to distribute among service providers depend on multiple factors (e.g. the network topology). In case the payments exceed the requester's willingness to pay in the complex service auction, the budget balance (cp. Requirement 4) cannot be achieved by the mechanism.

Example 4.2 [ACHIEVING BUDGET BALANCE]. *This example illustrates a non-budget-balanced outcome of the complex service auction. Figure 4.2 shows a service value network with service offers* $V = \{1, 2, 3, 4, 5, 6\} \cup \{s, f\}$. *For simplicity it is assumed that each service provider* $s_1, \ldots, s_6$ *only owns a single service within the network such that* $\sigma(s_i) = \{i\}$ *with* $i = 1, \ldots, 6$. *Furthermore it is assumed that the requester's willingness to pay is* $\alpha = 12$.

The mechanism allocates the path $f^* = \{e_{s1}, e_{14}, e_{4f}\}$ *as it yields the highest overall utility of* $\mathcal{U}_{f^*} = 12 - (2 + 2) = 8$. *According to the transfer function, each service provider*

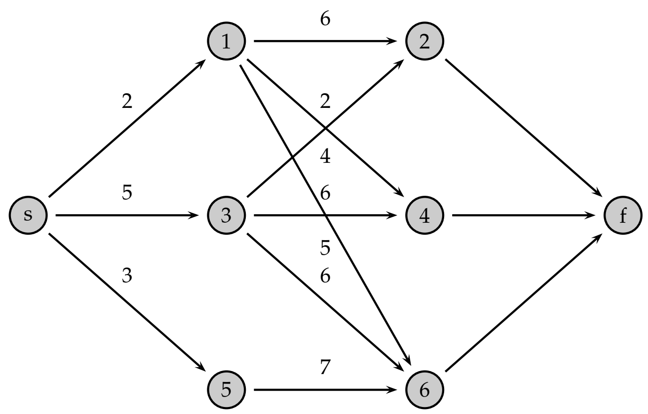


Figure 4.2: Non-budget-balanced outcome of the CSA.

that owns allocated service offers receives a payment consisting of the corresponding price bid and the critical value such that $t_1 = 2 + (8-3) = 7$ *and* $t_4 = 2 + (8-4) = 6$*. The sum of transfers which are distributed among the service providers exceeds the service requesters willingness to pay as* $\mathcal{U}^R = 12 - (7+6) = -1$*. Thus, an amount of* 1 *unit has to be externally subsidized in order to obtain the efficient allocation maximizing welfare.*

This section introduces an extension to the complex service auction that restores the desideratum of budget balance (cp. Requirement 4) by sacrificing truthfulness to a certain degree. The extension is based on the design of a transfer function – the Interoperability Transfer Function (ITF) – that limits overpayments to satisfy budget balance constraints (cp. Section 2.2.3.5). The ITF implements incentives for increasing services' interoperability with adjacent offers to foster the growth of agile service value networks with an increased level of feasible complex service instantiations.

4.2.1 Related Work

In VCG-based mechanisms, the transfers are indeterministic and can be arbitrarily high [AT07]. These so called *overpayments* or a mechanism's *frugality* is a central characteristic of a mechanism implementation, which is extensively analyzed in mechanism design research especially in the context of graph-based implementations [ESS04, AT07, Tal03, KK05]. A frugality ratio that measures the payments in a truthful mechanism compared to a non-truthful implementation is a ratio that "characterizes the cost of insisting on truthfulness" [KK05]. Approaches to predict overpayments that occur in truthful graph-based mechanisms have been developed in [KN04] in the context of random graphs and in [KN05] for large-scale networks.

Addressing this shortcoming of VCG-based mechanisms, an approximately efficient and budget-balanced solution to overpayment issues in VCG-based combinatorial auctions is introduced in [PKE01] while focusing on solving linear problems subject to budget balance that yield approximate incentive compatible solutions. Another approach to counteract the loss of budget balance by sacrificing efficiency is introduced in [AT07] in the context of path auctions. In their work they replace the efficient allocation function by a class of "minimum functions" that yield lower overpayments in certain scenarios. Nevertheless they show that it is always possible to construct worse case scenarios in which minimum functions perform as bad as the efficient variant.

4.2.2 Interoperability Transfer

Let T denote the sum of all incoming edges to service offers $V \setminus \{v_f\}$. Furthermore let τ_i be the number of incoming edges to service offer i such that $\sum_{i \in V \setminus \{v_f\}} \tau_i = T$. The ratio $r_i = \frac{\tau_i}{\mathrm{T}}$ denotes the incoming-edge-ratio for each node. Recall, e_{ui} represents an interoperable connection of service $i \in V$ with service $u \in V$, meaning that service i is capable of interpreting service u's output, i.e. service i is interoperable with service u. Thus, the more incoming edges to a service offer, the higher its feasible interoperability with its predecessor services. Hence, the incoming-edge-ratio r_i represents the degree of interoperability of service i with its predecessor services in comparison to all other services. Focusing on all service offers owned by a service provider s, the ratio $r^s = \frac{\sum_{i \in \sigma(s)} \tau_i}{\mathrm{T}}$ denotes the incoming-edge-ratio of service provider s.

Let $\Delta t^{crit,s}$ denote the critical value of service provider s. The idea to construct a transfer function that accounts for budget balance constraints is based on the work in [PKE01] and focuses on choosing adequate discounts Δ^s for each service provider $s \in S$ instead of paying every allocated service provider the critical value. The decision on how to choose adequate discounts is formulated as a general optimization problem subject to budget balance constraints.

$$L_\tau(\Delta, \Delta t^{crit,s}) = \sum_{s \in S} r^s (\Delta t^{crit,s} - \Delta^s) \tag{4.3}$$

L_τ represents the weighted distance function that measures the distance between the service providers' critical values and computed discounts with respect to the incoming-edge-ratio. The goal is to distribute the surplus $\mathfrak{S}^* = \alpha \mathcal{S}(\mathcal{A}_{f^*}) - \mathcal{P}_{f^*}$ in a way that it minimizes the distance function L_τ. In other words, the goal is to transfer discounts Δ^s to service providers, which together minimize the overall weighted distance $\sum_{s \in S} r^s (\Delta t^{crit,s} - \Delta^s)$ and do not exceed the surplus $\mathfrak{S}^*$. Minimizing the distance function L_τ subject to budget balance, individual rationality and the critical values as upper boundaries leads to the following special optimization problem:

$$\min_{\Delta} \sum_{s \in S} r^s (\Delta t^{crit,s} - \Delta^s) \tag{4.4}$$

$$\text{s.t.} \sum_{s \in S} \Delta^s \leq \mathfrak{S}^* \tag{BB}$$

$$\Delta^s \leq \Delta t^{crit,s}, \forall s \in S \tag{CV}$$

$$\Delta^s \geq 0, \forall s \in S \tag{IR}$$

The Lagrangian problem consequently follows such that

$$z(\lambda) = \min_{\Delta} \sum_{s \in S} r^s (\Delta t^{crit,s} - \Delta^s) + \lambda (\sum_{s \in S} \Delta^s - \mathfrak{S}^*)$$
$$\text{s.t. } 0 \leq \Delta^s \leq \Delta t^{crit,s}, \forall s \in S$$

The problem decomposes into smaller problems for each s.

$$\min_{\Delta^s} (r^s \Delta t^{crit,s}) - \Delta^s (\lambda - r^s)$$
$$\text{s.t. } 0 \leq \Delta^s \leq \Delta t^{crit,s}, \forall s \in S$$

If the coefficient $(\lambda - r^s)$ is negative, the expression is minimized by setting Δ^s to the maximum value that does not violate the side condition which is $\Delta^{*s} = \Delta t^{crit,s}$. If the term $(\lambda - r^s)$ is positive, the whole expression is minimized by $\Delta^{*s} = 0$. If $(\lambda - r^s) = 0$, Δ^{*s} is set to a value $\tilde{\Delta}^s$ which is defined in the remainder of this section. Consequently the optimization problem implies finding a optimal threshold parameter C_τ for λ such that

$$\Delta^{*s}(C_\tau) = \begin{cases} \Delta t^{crit,s}, & \text{if } C_\tau < r^s \\ \tilde{\Delta}^s, & \text{if } C_\tau = r^s \\ 0, & \text{otherwise} \end{cases} \tag{4.5}$$

Based on the optimal solution Δ^*, the complete interoperability transfer function evolves accordingly:

$$t^{ITF,s} := \begin{cases} \sum_{i\in\tau(j)}\sum_{j\in\sigma(s)} p_{ij} + \Delta t^{crit,s}, & \text{if } e_{ij} \in o, C_\tau < r^s \\ \sum_{i\in\tau(j)}\sum_{j\in\sigma(s)} p_{ij} + \tilde{\Delta}^s, & \text{if } e_{ij} \in o, C_\tau = r^s \\ \sum_{i\in\tau(j)}\sum_{j\in\sigma(s)} p_{ij}, & \text{if } e_{ij} \in o, C_\tau > r^s \\ 0, & \text{otherwise} \end{cases} \tag{4.6}$$

Service providers that have an incoming-edge ratio which equals the threshold ($C_\tau = r^s$) and own service offers with allocated incoming edges, receive a part of their critical value which depends on the number of service providers with $C_\tau < r^s$, corresponding critical values and the number of service providers with $C_\tau = r^s$. The value $\tilde{\Delta}^s$ is defined as follows:

$$\tilde{\Delta}^s := \frac{\mathfrak{S}^* - \sum\limits_{s\in S|C_\tau<r^s} \Delta t^{crit,s}}{\sum\limits_{s\in S|C_\tau=r^s} 1} \tag{4.7}$$

4.2.3 Finding the Optimal Threshold Parameter

The threshold C_τ divides allocated service providers into two groups where one gets a discount of $\Delta t^{crit,s}$ and the other 0. Let k denote the threshold index such that if C_τ falls into the interval k such that $C_\tau \in [r_{\tau_{k+1}}, r_{\tau_k})$ service providers $1,\dots k$ (ordered increasingly based on their critical values) get their critical value while service providers $k+1,\dots,I$ get no discount. Putting the solution $\Delta^{*s}(C_\tau)$ in the Lagrangian problem $z(C_\tau)$ leads to

$$z(C_\tau,k) = \sum_{i=k+1}^{I} (r^i \Delta t^{crit,i}) + C_\tau \left(\sum_{i=1}^{k} \Delta t^{crit,i} - \mathfrak{S}^* \right) \tag{4.8}$$

The optimum is attained at

$$C_\tau^* = r_{k^*+1}, \sum_{i=1}^{k^*} \Delta t^{crit,i} \leq \mathfrak{S}^* \wedge \sum_{i=1}^{k^*+1} \Delta t^{crit,i} > \mathfrak{S}^* \tag{4.9}$$

Example 4.3 [Achieving Budget Balance (Continued)]. *Recalling Example 4.2, this continuation illustrates how budget balance can be retained by implementing the interoperability transfer function. In order to determine an optimal threshold parameter* C_τ*, each service provider that owns allocated service offers is decreasingly ordered by its incoming-edge-ratio* r^s*. The number of possible edges within* G *is denoted by* $\mathrm{T} = 10$*. Con-*

sequently, the incoming-edge-ratio r for service providers that own allocated service offers evolves as $r^{s_1} = \frac{\sum_{i \in \sigma(s_1)} \tau_i}{\mathrm{T}} = \frac{1}{10}$ *and* $r^{s_4} = \frac{2}{10}$*. The vector of the ordered incoming-edge ratios is* $(\frac{2}{15}, \frac{1}{10})$*. Equation (4.9) is satisfied by* $C^*_\tau = \frac{1}{10}$ *with* $k^* = 2$ *which is the solution that satisfies the conditions* $\sum_{i=1}^{k^*} \Delta t^{crit,i} \leq \mathfrak{S}^* \wedge \sum_{i=1}^{k^*+1} \Delta t^{crit,i} > \mathfrak{S}^*$*. The value* $\tilde{\Delta}$ *for service provider* s_1 *is* $\tilde{\Delta}^{s_1} = \frac{8-4}{1} = 4$*. Payments for allocated service offers evolve accordingly such that* $t^{ITF,s_1} = 2 + 4 = 6$ *and* $t^{ITF,s_4} = 2 + 4 = 6$*. As* $\mathcal{U}^R = 12 - (6 + 6) = 0$*, the outcome of the extended complex service auction is budget-balanced and does not have to be subsidized externally. It is important to notice that the interoperability transfer function rewards service provider* s_4 *for the high degree of interoperability – i.e. the incoming-edge-ratio* r^{s_4} *– which increases the variety of feasible complex service compositions.*

4.2.4 Summary

In summary, the ITF extension as a novel budget-balanced payment scheme which satisfies Requirement 4 implements incentives for service providers to increase their services' degree of interoperability which is shown in Section 6.2.2.

It is important to note that the incentives provided by the ITF are twofold: First, the ITF limits strategic behavior of service providers which is shown in Section 6.1. Second, the ITF rewards interoperability endeavors. Depending on the design goals the payment scheme can be adjusted in order to calibrate both effects. Introducing a calibration weight $\beta^{ITF} \in [0;1]$ and a threshold term $\tilde{r}^s := \beta^{ITF} r^s + (1 - \beta^{ITF}) \frac{t^{crit,s}}{\sum_{s \in S} \Delta t^{crit,s}}$ an adjustable interoperability transfer function evolves as follows:

$$t^{ITF,s} := \begin{cases} \sum_{i \in \tau(j)} \sum_{j \in \sigma(s)} p_{ij} + \Delta t^{crit,s}, & \text{if } e_{ij} \in o, \tilde{C}_\tau < \tilde{r}^s \\ \sum_{i \in \tau(j)} \sum_{j \in \sigma(s)} p_{ij} + \tilde{\Delta}^s, & \text{if } e_{ij} \in o, \tilde{C}_\tau = \tilde{r}^s \\ \sum_{i \in \tau(j)} \sum_{j \in \sigma(s)} p_{ij}, & \text{if } e_{ij} \in o, \tilde{C}_\tau \geq \tilde{r}^s \\ 0, & \text{otherwise} \end{cases} \tag{4.10}$$

The computation of the optimal threshold parameter $\tilde{C}_\tau$ is done analogously to the procedure described in Section 4.2.3 accounting for $\tilde{r}^s$ instead of r^s. Thus, β^{ITF} adjusts the transfer function with respect to both incentives. Higher values for β^{ITF} result in stronger incentives for interoperability endeavors whereas lower values provide stronger incentives to reduce strategic behavior.

With respect to the service level enforcement extension, the ITF can easily be combined with the compensation function as introduced in Section 4.1. Service providers that pass the threshold receive their critical value minus their compensation value. Note that in this case the computation of the optimal threshold parameter has to be adjusted accordingly to assure budget balance.

4.3 Managing Service Quality

Recall that with the tremendous decrease of costs for the provision of highly scalable services, service providers shift from price to quality competition. QoS is the key criterion to keep the business competitive as it has serious implications on the provider and consumer side [Pap08]. Thus, an efficient management of highly complex QoS characteristics is inevitable for service-oriented value creation in service value networks. In Section 3.2, the basic concept of QoS aggregation and evaluation has been described based on rather simple QoS attributes such as response time, which are characterized by well-defined metrics to measure corresponding values.

In order to determine the overall score for a provider based on the scoring function, the attribute values of the complex service have to be computed. The type of operation for aggregating attribute value highly depends on the attribute type. Basic quality of service attributes such as response time for example can be aggregated with a sum operator. Table 3.1 shows different types of aggregation functions for multiple attribute types exemplarily. For example, the overall throughput of a complex service that consists of multiple service components is determined by the lowest throughput rate within the allocation and can therefore be computed using a minimum operator.

Nevertheless, only considering basic quality of service attributes is not sufficient for dealing with complex non-functional service characteristics that express rich semantic information. The auction mechanism must be capable of aggregating a broad range of descriptive service attributes that express multiple quality aspects (e.g. the physical hosting location of a service and additional semantic information about the environment, a service's usage policies or ownership rights) . This section focuses on providing the conceptual foundations for a seamless management of more sophisticated QoS characteristics, which require a semantic understanding of their context and interrelations in order to measure and evaluate their particular occurrences.

To represent semantic knowledge about service quality attributes in an interoperable manner, ontologies are used to describe a conceptualization of service characteristics and properties. The following definition is predominantly used in the semantic Web community [SBF98].

Definition 4.1 [ONTOLOGY]. *An ontology is a formal explicit specification of a shared conceptualization of a domain of interest.*

In order to enable automatic processing and interpretation of explicit knowledge representations, adequate and machine-interpretable formalisms are used, which are explained in the following section.

4.3.1 Knowledge Representation Formalisms

As a formalism to represent an ontology framework the Web Ontology Language (OWL) is used. OWL is an ontology language standardized by the World Wide Web Consortium (W3C) [MvH04] and is based on the description logic (DL) formalism [BCM$^+$07]. Due to its close connection to DL it facilitates logical inferencing and allows to derive conclusions from an ontology that have not been stated explicitly. As a brief introduction a review of some of the modeling constructs of OWL using its DL-syntax is outlined here. The main elements of OWL are *individuals*, *properties* that relate individuals to each other and *classes* that group together individuals, which share some common characteristics. Classes as well as properties can be put into subsumption hierarchies. Furthermore, OWL allows for describing classes in terms of complex *class constructors* that pose restrictions on the properties of a class. For example, the statement $BigCity \sqsubseteq \exists\ isConnectedTo.Highway$ describes the class of big cities, which are connected to some Highway. Subclass relationship can be expressed by a statement like $BigCity \sqsubseteq InterestingCity$, saying that any big city is also interesting.

For the reader's convenience, ontologies are illustrated in UML notation where UML classes correspond to OWL concepts, UML associations to object properties, UML inheritance to sub-concept relations, UML dependencies to OWL class instantiations and UML attributes to OWL datatype properties [BVEL04].

To enable rule-like knowledge representation which is not supported by the modeling primitives based on OWL-DL, the Semantic Web Rule Language (SWRL) [HPSB$^+$04] allows to extend OWL with Horn-like rules according to first-order semantics. Additionally, SWRL provides an XML-based formalization, which enables automatic processing of rule-based knowledge as an extension to the OWL semantics. Furthermore SWRL allows for the implementation of algorithmic calculations such as mathematic operations and string comparison.

4.3.2 Semantic QoS Management

To foster a comprehensive management of QoS characteristics, the complex service auction is extended using concepts from Semantic Web research. Providing a broad contextual knowledge about attribute types, their conceptualization and relations to other concepts in a machine-readable and interoperable manner, ontologies are used to capture relevant semantic information. Based on this knowledge, individual attribute types can be expressed using a rule language formalism. The following example demonstrates the expressiveness of a semantic approach towards the description of QoS characteristics and the expression of individual requirements of requesters.

Example 4.4 [CSA WITH SEMANTIC QOS MANAGEMENT]. *For the reader's convenience, the scenario is reduced to a minimal setting that is sufficient to illustrate the strength of semantic service description and attribute aggregation. Figure 4.3 shows a service value network with four service offers* 1,2,3 *and* 4 *and three feasible paths from source to sink:* $f^1 = \{e_{s1}, e_{12}, e_{2f}\}$, $f^2 = \{e_{s1}, e_{14}, e_{4f}\}$ *and* $f^3 = \{e_{s3}, e_{34}, e_{4f}\}$.

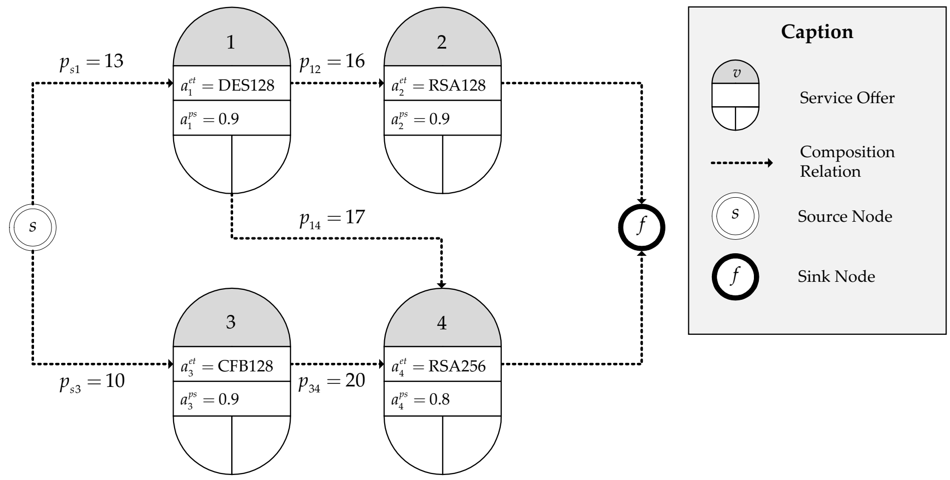


Figure 4.3: Service value network with semantic QoS characteristics.

For simplicity it is assumed that each service provider owns only a single service such that $\sigma(s_1) = \{1\}$, $\sigma(s_2) = \{2\}$, $\sigma(s_3) = \{3\}$ *and* $\sigma(s_4) = \{4\}$. *Price values* p_{ij} *on the edges represent price bids announced by service providers. Each service configuration* A_j *consists of attribute values for* encryption type a_j^{et} *and* probability of success a_j^{ps}. *The attribute values in Figure 4.3 are assumed to be announced by each service provider additionally to the corresponding price bid such that* $b_{ij} = (A_j p_{ij})$. *Attribute values are aggregated according to the aggregation operations in Table 3.1. Attribute values for* encryption type *are derived from the concepts in the security algorithm ontology as illustrated in Figure 4.4.*

The security encryption ontology provides a brief conceptualization of encryption types an their hierarchical classification in symmetric and asymmetric cipher methods. Symmet-

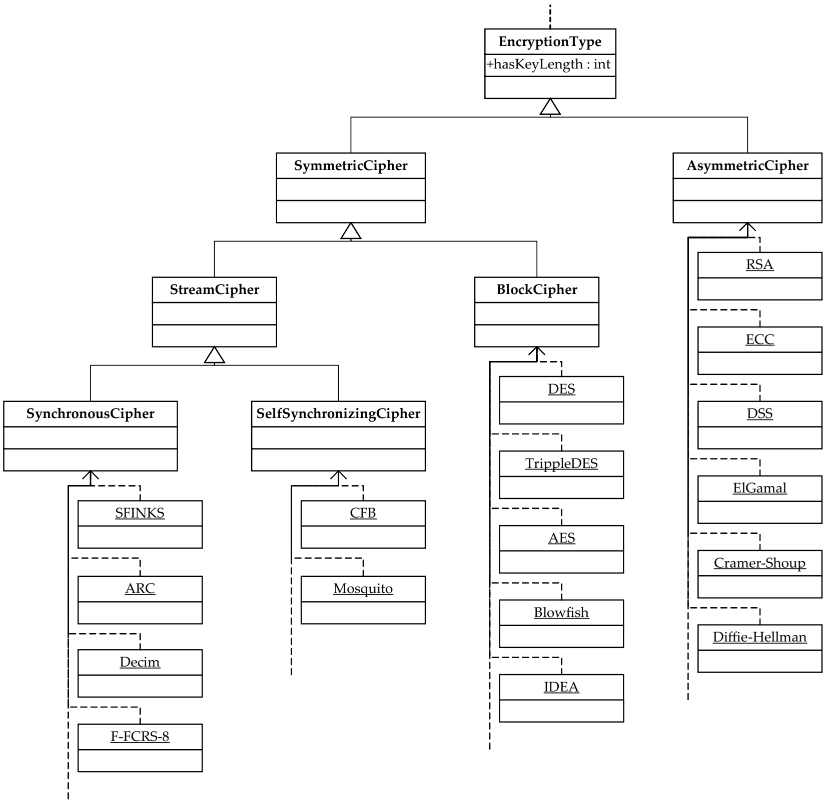


Figure 4.4: Security encryption ontology.

ric cipher methods are further divided into synchronous and self-synchronizing stream ciphers and block cipher methods. Based on this semantic information about different encryption types, the requester is capable of designing an individual attribute type which incorporates the preferred encryption configuration. The following rules are implementation-independently formulated in First-Order Logic (FOL) syntax.

$$\text{(R1)} \qquad a^{ie} \longleftarrow \mathit{EncryptionType}(a^{et}), \mathit{BlockCipher}(a^{et}), \mathit{hasKeyLength}(a^{et}, k), \mathit{isGreaterOrEqual}(k, 128)$$

$$\text{(R2)} \qquad a^{ie} \longleftarrow \mathit{EncryptionType}(a^{et}), \mathit{AsymmetricCipher}(a^{et}), \mathit{hasKeyLength}(a^{et}, k), \mathit{isGreaterOrEqual}(k, 256)$$

In this example the requester specifies an attribute type $ie \in \mathcal{L}$ representing individual encryption. *This attribute type is defined by Rule* (R1) *and Rule* (R2). *If a single rule fires, the boolean attribute value a^{ie} is set to true, meaning that the service offer satisfies the individual encryption requirements expressed by the requester.*

Assuming a requester's maximum willingness to pay for a complex service with a score of 1 is $\alpha = 100$ and preferences for attribute types individual encryption *and* probability of success *are* $\lambda_{ie} = 0.2$ *and* $\lambda_{ps} = 0.8$*, the overall utility of each feasible path evolves as follows*

$$\begin{aligned}
\mathcal{U}_{f^1} &= 100 \times (0.2 \times (1 \wedge 0) + 0.8 \times (0.9 \times 0.7)) - (13 + 16) = 21.4 \\
\mathcal{U}_{f^2} &= 100 \times (0.2 \times (1 \wedge 1) + 0.8 \times (0.9 \times 0.8)) - (13 + 17) = 47.6 \\
\mathcal{U}_{f^3} &= 100 \times (0.2 \times (0 \wedge 1) + 0.8 \times (0.9 \times 0.8)) - (10 + 20) = 27.6
\end{aligned}$$

As the complex service instance f^2 *yields the highest overall utility, service offers* 1 *and* 4 *via edges* e_{s1},e_{14} *and* e_{4f} *are allocated by* $o(B)$*. Thus, service providers* s_1 *and* s_2 *receive a transfer according to the transfer function in Equation* (3.10) *based on their critical value.*

$$\begin{aligned}
t^{s_1} &= t_1^{s_1} = 13 + (47.6 - 27.6) = 33 \\
t^{s_4} &= t_4^{s_4} = 17 + (47.6 - 21.4) = 43.2
\end{aligned}$$

Consequently the service requester's utility evolves as

$$\mathcal{U}^R = 100 \times (0.2 \times (1 \wedge 1) + 0.8 \times (0.9 \times 0.8)) - (33 + 43.2) = 1.4$$

In summary, the integration of rule-based semantic description techniques allows for the specification, aggregation and management of highly complex QoS characteristics which satisfies Requirement 7.

Part III

Evaluation

Chapter 5

Analytical Results

[...] the set of incentive-compatible direct-revelation mechanisms has simple mathematical properties that often make it easy to characterize, because can be defined by a set of linear inequalities.

[Mye88]

This chapter thoroughly analyzes the economic properties of the complex service auction and their extensions as introduced in Chapter 3. Section 5.1 analytically shows that the complex service auction with the service level enforcement extension implements a strategyproof social choice, i.e. reporting ones true multidimensional type is an equilibrium in weakly dominant strategies. Focusing on cooperative behavior of adjacent service providers in service value networks, Section 5.2 studies the effect of interface customization and implicit cost reductions for preceeding or succeeding services within service value networks.

5.1 Incentive Compatibility & Individual Rationality

Recalling Section 2.2.4, incentive compatibility is a valuable property to be achieved in mechanism design. In decentralized environments such as service value networks with self-interested participants that have private information about their preferences for different outcomes, solving a global optimization problem fully depends on how participants can be incentivized to report their private information to the auctioneer in a truthful manner. This information is needed to compute e.g. an allocative efficient outcome in such a setting. Hence, incentive compatibility can be seen as a necessary precondition in order to achieve a welfare maximizing outcome in scenarios with incomplete information.

Another major beneficial result that derives from truthfulness is that it tremendously simplifies the strategy space of participants as they do not have to reason about strategies of other participants. Thus, incentive compatibility reduces the participants' strategy space and simplifies their decision problem to a single weakly dominant strategy maximizing their individual utility.

The remainder of this section analytically shows that in the basic complex service auction (without the compensation function extension), bidding ones true *valuations* for all offered services is a weakly dominant strategy for all participating service providers (Section 5.1.1). Based on these results, Section 5.1.2 shows that in the complex service auction with the service level enforcement extension (cp. Section 4.1), bidding true valuations *and* true QoS characteristics for all offered services is a weakly dominant strategy for all participating service providers which satisfies Requirement 2. Based on the results regarding truthfulness it is briefly shown that service providers always end up with a payoff equal to or greater than zero which satisfies individual rationality as stated in Requirement 3.

5.1.1 One-Dimensional Bids in the Basic CSA

This section is concerned with strategic behavior in the basic complex service auction, i.e. the basic mechanism implementation without the compensation function extension which enables service level enforcement. The following analytical evaluation of the mechanism implementation with respect to service providers' bidding strategy considers *price bids* only in the first place. Thus, the providers' strategy space is reduced to announcing prices for each incoming edge of each service offer they own.

First, Corollary 5.1 shows that once a service provider is allocated – that is, the service provider owns service offers that have at least one incoming edge which is allocated by the mechanism – its payoff is independent of its bidding strategy. This means that once a service provider is allocated it is indifferent between any alternative bidding strategy within its strategy space.

Consequently, the only event that service providers can actively influence by their bidding strategy is whether they are allocated by the mechanism or not. Based on the results of Corollary 5.1, Theorem 5.1 considerers the cases in which service providers intent to be allocated and derives the optimal bidding strategy: Service providers act best (or at least equally good) by following a truth-telling strategy, i.e. reporting their true valuations – which are assumed to be reflected by corresponding internal costs – for each service offer is a weakly dominant strategy for all service providers that participate in the complex service auction.

Corollary 5.1. *For each service provider $s \in S$ that participates in the complex service auction, the transfer t^s is independent of its price bid. More precisely this means that for each service offer $j \in V$ owned by $s \in S$ with an incoming edge which is allocated by o such that $e_{ij} \in o$ with $j \in \sigma(s)$ and $i \in \tau(j)$, service provider s's payoff is independent of its price bid p_{ij}.*

Proof 5.1 [COROLLARY 5.1]. *Let F_{-s} denotes the set of all feasible paths from source to sink in the reduced graph G_{-s} without every service offer owned by service provider s and corresponding incoming and outgoing edges. Let further f^* denote the path which is allocated by o. Let $\mathcal{U}^*$ be the utility of path f^*. Let $\mathcal{U}^*_{-s}$ be the utility of path f^*_{-s} in the reduced graph G_{-s}. Let $\tilde{E}^s$ denote the set of edges with $\tilde{E}^s = \{e_{ij} | e_{ij} \in o, j \in \sigma(s), i \in \tau(j)\}$. Distinguishing two possible cases, service provider s's payoff π^s evolves as follows.*

1. *$\tilde{E}^s = \emptyset$. Service provider s is not allocated. More precisely, none of the incoming edges of service offers owned by service provider s are allocated by o.*
   It follows directly that in this case $\pi^s = 0$ independent of s's price bid.

2. *$\tilde{E}^s \neq \emptyset$. Service provider s is allocated. More precisely, at least one of the incoming edges of service offers owned by service provider s is allocated by o.*

$$\begin{aligned}
\pi^s &= t^s - c^s \\
\pi^s &= \sum_{\tilde{E}^s} p_{ij} + (\mathcal{U}^* - \mathcal{U}^*_{-s}) - \sum_{\tilde{E}^s} c_{ij} \\
\pi^s &= \sum_{\tilde{E}^s} p_{ij} + \alpha S(\mathcal{A}_{f^*}) - \sum_{e_{ij} \in o} p_{ij} - \mathcal{U}^*_{-s} - \sum_{\tilde{E}^s} c_{ij} \\
\pi^s &= \alpha S(\mathcal{A}_{f^*}) - \sum_{e_{ij} | e_{ij} \in o, e_{ij} \notin \tilde{E}^s} p_{ij} - \mathcal{U}^*_{-s} - \sum_{\tilde{E}^s} c_{ij} \qquad (5.1)
\end{aligned}$$

This shows that for each service offer j owned by s that has an incoming edge e_{ij} which is allocated by o – otherwise s does not receive a transfer – the corresponding profit is independent of s's price bid p_{ij}. □

Theorem 5.1. *For each service provider $s \in S$ that participates in the complex service auction, the price bidding strategy $p_{ij} = c_{ij}$ (truth-telling) $\forall i \in \tau(j), \forall j \in \sigma(s)$ is a weakly dominant strategy.*

Proof 5.1 [THEOREM 5.1]. *Corollary 5.1 shows that the transfer t^s for each service provider $s \in S$ is independent of the price bid. Consequently, the only event that s can proactively influence by its bidding strategy is whether its service offers are allocated by o or not. Let $\tilde{E}^s = \{e_{ij} | e_{ij} \in o, j \in \sigma(s), i \in \tau(j)\}$ denote the set of incoming edges of service offers owned by service provider s that are allocated by o. Service provider s wants incoming edges of service offers that s owns to be allocated by o ($\tilde{E}^s \neq \emptyset$) iff $\pi^s > 0$. Hence, service* *provider s wants the following equivalence*[1] *to be fulfilled through an adequate choice of its price bid.*

$$
\begin{array}{llllll}
 & \tilde{E}^s \neq \emptyset & \iff & \mathcal{U}^* > \mathcal{U}^*_{-s} & \iff & \pi^s > 0 \\
(5.2) & & & \mathcal{U}^* - \mathcal{U}^*_{-s} > 0 & \iff & \sum_{\tilde{E}^s} (p_{ij} - c_{ij}) + (\mathcal{U}^* - \mathcal{U}^*_{-s}) > 0
\end{array}
$$

Equation (5.2) *holds for* $p_{ij} = c_{ij}\ \forall j \in \sigma(s), i \in \tau(j)$. *According to Corollary 5.1, if* $\tilde{E}^s \neq \emptyset$, *s is indifferent between any other solution that satisfies Equation* (5.2) *which means that reporting true internal costs is a weakly dominant price bidding strategy for service provider s.* □

5.1.2 Multidimensional Bids in the Extended CSA

The analytical evaluation of service providers' bidding strategies in this section is conducted analogously to the one-dimensional case. Nevertheless, the following evaluation accounts for the complete strategy space of service providers, i.e. service providers announce multidimensional bids consisting of a *price and QoS component* for each incoming edge of every service offer they own within the service value network. The analysis is based on the complex service auction mechanism with the compensation function extension (cp. Section 4.1) which implements a service level enforcement component.

Laying the groundwork for Theorem 5.2, Corollary 5.2 shows that once a service provider is allocated, its payoff is independent of its announced price and corresponding attribute values which characterize guaranteed QoS. This means that once a service provider is allocated it is indifferent between any alternative bidding strategy within its strategy space with respect to all dimensions of its bid.

However, the service providers' bid (price and attribute values) influences the chance of being allocated by the mechanism. Based on the results of Corollary 5.2, Theorem 5.2 considerers the cases in which service providers intent to be allocated and derives the optimal bidding strategy. Theorem 5.2 shows that service providers act best (or at least equally good) by reporting their true multidimensional type, i.e. reporting their true valuations and guaranteed QoS for each service offer regarding its predecessor is a weakly dominant strategy for all service providers that participate in the extended complex service auction.

Corollary 5.2. *For each service provider* $s \in S$ *that participates in the complex service auction with the compensation function extension (cp. Section 4.1), the transfer* t^s *is indepen-*

[1] Two statements are equivalent as denoted by $\iff$ if and only if both statements yield the same outcome for every possible interpretation.

dent of all dimensions of s's bids (configuration and price). This means that for each service offer $j \in V$ owned by $s \in S$ that has an incoming edge which is allocated by o such that $e_{ij} \in o$ with $j \in \sigma(s)$ and $i \in \tau(j)$, service provider s's payoff is independent of all dimensions of its bid $b_{ij} = (A_j, p_{ij})$.

Proof 5.2 [COROLLARY 5.2]. *Let F_{-s} denote the set of all feasible paths from source to sink in the reduced graph G_{-s} without every service offer owned by service provider s and corresponding incoming and outgoing edges. Let further f^* denote the path which is allocated by o. Let $\mathcal{U}^*$ be the utility of path f^*. Let $\mathcal{U}^*_{-s}$ be the utility of path f^*_{-s} in the reduced graph G_{-s}. Let $\tilde{\mathcal{U}}^{*s}$ denote the overall utility of the allocated path f^* computed based on the verified attribute values $\tilde{a}^1_j, \ldots, \tilde{a}^L_j$ of the verified configurations $\tilde{A}_j$ of all service offers $j \in \sigma(s)$. Let $\tilde{E}^s$ denote the set of edges with $\tilde{E}^s = \{e_{ij} | e_{ij} \in o, j \in \sigma(s), i \in \tau(j)\}$. Distinguishing two possible cases, service provider s's payoff π^s evolves as follows.*

1. *$\tilde{E}^s = \emptyset$. Service provider s is not allocated. More precisely, none of the incoming edges of service offers owned by service provider s are allocated by o.*
   It follows directly that in this case $\pi^s = 0$ independent of s's price bid.

2. *$\tilde{E}^s \neq \emptyset$. Service provider s is allocated. More precisely, at least one of the incoming edges of service offers owned by service provider s is allocated by o.*

$$\begin{aligned}
\pi^s &= t^s - c^s \\
\pi^s &= \sum_{\tilde{E}^s} p_{ij} + (\mathcal{U}^* - \mathcal{U}^*_{-s}) - t^{comp,s} - \sum_{\tilde{E}^s} c_{ij} \\
\pi^s &= \sum_{\tilde{E}^s} p_{ij} + (\mathcal{U}^* - \mathcal{U}^*_{-s}) - (\mathcal{U}^* - \tilde{\mathcal{U}}^{*s}) - \sum_{\tilde{E}^s} c_{ij} \\
\pi^s &= \sum_{\tilde{E}^s} p_{ij} + (\tilde{\mathcal{U}}^{*s} - \mathcal{U}^*_{-s}) - \sum_{\tilde{E}^s} c_{ij} \\
\pi^s &= \alpha S(\tilde{\mathcal{A}}^s_{f^*}) - \sum_{e_{ij} | e_{ij} \in o, e_{ij} \notin \tilde{E}^s} p_{ij} - \mathcal{U}^*_{-s} - \sum_{\tilde{E}^s} c_{ij} \qquad (5.3)
\end{aligned}$$

Equation (5.3) *shows that for each service offer $j \in V$ owned by $s \in S$ that has an incoming edge which is allocated by o such that $e_{ij} \in o$ with $j \in \sigma(s)$ and $i \in \tau(j)$, service provider s's payoff is independent of all dimensions of its bid $b_{ij} = (A_j, p_{ij})$.* □

Theorem 5.2. *For each service provider $s \in S$ that participates in the complex service auction with the compensation function extension (cp. Section 4.1), the bidding strategy $b_{ij} = (\tilde{A}_j, c_{ij})$ with $\tilde{A}_j = (\tilde{a}^1_j, \ldots, \tilde{a}^L_j)$ – truth telling with respect to all dimensions of the bid – $\forall i \in \tau(j), \forall j \in \sigma(s)$ is a weakly dominant strategy.*

Proof 5.2 [THEOREM 5.2]. *Let F_{-s} denote the set of all feasible paths from source to sink in the reduced graph G_{-s} without every service offer owned by service provider s and corresponding incoming and outgoing edges. Let further f^* denote the path which is allocated* *by o. Let $\mathcal{U}^*$ be the utility of path f^*. Let $\mathcal{U}^*_{-s}$ be the utility of path f^*_{-s} in the reduced graph G_{-s}. Let $\tilde{\mathcal{U}}^{*s}$ denote the overall utility of the allocated path f^* computed based on the verified attribute values $\tilde{a}^1_j, \dots, \tilde{a}^L_j$ of the verified configurations $\tilde{A}_j$ of all service offers $j \in \sigma(s)$. Let $\tilde{E}^s$ denote the set of edges with $\tilde{E}^s = \{e_{ij} | e_{ij} \in o, j \in \sigma(s), i \in \tau(j)\}$. Corollary 5.2 shows that the transfer t^s for each service provider $s \in S$ is independent of all dimensions of its bid. In other words, s's bid does not have an impact on its transfer t^s and its payoff π^s respectively. Nevertheless, the bidding strategy influences service provider s's chance of being allocated by o. Thus, s wants to be allocated iff $\pi^s > 0$.*

$$
\begin{aligned}
\tilde{E}^s \neq \emptyset \iff \mathcal{U}^* > \mathcal{U}^*_{-s} &\iff \pi^s > 0 \\
\mathcal{U}^* > \mathcal{U}^*_{-s} &\iff \sum_{\tilde{E}^s} p_{ij} + (\tilde{\mathcal{U}}^{*s} - \mathcal{U}^*_{-s}) - \sum_{\tilde{E}^s} c_{ij} > 0 \\
\mathcal{U}^* > \mathcal{U}^*_{-s} &\iff \sum_{\tilde{E}^s} p_{ij} + \tilde{\mathcal{U}}^{*s} > \sum_{\tilde{E}^s} c_{ij} + \mathcal{U}^*_{-s}
\end{aligned}
\tag{5.4}
$$

Equation (5.4) holds for $p_{ij} = c_{ij}$ and $\mathcal{U}^* = \tilde{\mathcal{U}}^{*s}$. According to Corollary 5.2, if $\tilde{E}^s \neq \emptyset$, s is indifferent between any other solution that satisfies Equation (5.4) which means that reporting attribute values $a^1_j, \dots, a^l_j$ truthfully meaning that the announced values equal the verified ones in the execution phase such that $a^l_j = \tilde{a}^l_j \; \forall l \in L, \forall j \in \sigma(s)$ and consequently $\mathcal{U}^* = \tilde{\mathcal{U}}^{*s}$ is a weakly dominant strategy. □

The analytical proof in Section A.2 evaluates service providers' bidding strategies from the perspective of the providers' *expected* payoff which they intent to maximize. Analogue to the previous result, it turns out that there exists a single bidding strategy that maximizes service providers' expected payoff.

5.1.3 Results & Implications

Theorem 5.2 shows that service providers act best (or at least as good as any other alternative) by reporting their services' configurations and internal costs truthfully which is a valuable mechanism property as it enables the computation of an optimal welfare maximizing outcome although the scenario is predominated by incomplete information. This property assures that although all service providers act self-interested and therefore try to maximize their profit, their dominant strategy maximizes the system's welfare and the requester receives a technically feasible instantiation of the desired complex service at a guaranteed service level[2]. The presence of a single beneficial strategy tremendously lowers strategic complexity for service providers and fosters a trustful requester-provider-relationship. The results

[2] Despite of service level agreement violations caused by events which are not under the control of service providers.

at hand show that the extended complex service auction satisfies Requirement 2. It is straightforward to see that with the results of Theorem 5.2, participating service providers always end up with a payoff equal to or greater than zero which satisfies individual rationality as stated in Requirement 3. In other words, service providers have an incentive to participate in the complex service auction without running into the risk of being worth of than their outside option. Furthermore, it follows directly form Corollary A.1 that Requirement 1 is satisfied through the social choice implemented by the complex service auction.

It is well-known in literature that incentive compatibility in VCG-based mechanisms may fail in repeated games [BS00]. Assuming that participants are able to gather historic information about previous outcomes, deviation from truth-telling might be beneficial in certain situations and the theoretical results from this section might not hold. However, in service value networks through a high degree of alteration with respect to changing service providers, variable costs and network topologies is observable. As outlined in Section 2.1.4, the complex service auction is designed for scenarios with fast changing participants that together foster value creation which satisfies situational needs. Thus, each auction setting is different from the preceding one which makes learning from past situations impossible and each game can therefore be treated as a one-shot game. For a simulation-based analysis of collusion behavior in the complex service auction, the interested reader is referred to [CvD09].

5.2 Cooperation within the Value Chain

This section studies a special form of cooperation in the context of the complex service auction in service value networks. Traditionally in social network research, the creation of links connecting players requires a cooperative process such that both participants have to agree to a connection. Removing links, however, is a non-cooperative act as it can be done unilaterally by a single player within the network. In the context of service value networks where service components' input and outputs are plugged together realizing a value-added complex service, service providers have the strategic opportunity to customize their service offers in a way that they are interoperable with predecessor services. This form of establishing a feasible connection to another component within the network is – in contrary to traditional social network theory – unilateral and non-cooperative. Predecessor services cannot control which successor service creates a connection by postprocessing its output.

5.2.1 Related Work

In [JW96] the evolution of social and economic networks where self-interested individuals form or sever links is analyzed. In [JW02] network formation is founded upon players' individual improvements resulting from changes in the network topology. Traditionally, breaking relationships can be done unilaterally while the formation of links requires consent from both players [JW96]. In [BG00], however, links can be formed by individual decision under certain circumstances. This is also the case in service value networks since service providers cannot influence which other services process their outputs.

5.2.2 A Model of Cooperation

In a service value network with four service offers a, b, y, z are two particular service offers $y \in V$ and $z \in V$ that are owned by two different service providers $s_y \in S$ and $s_z \in S$. Based on the topology of the Service Value Network y is the predecessor of z connected by an edge e_{yz}. Costs that service provider s_z has to bear for its service z being executed as a successor of service y are denoted by c_{yz}.

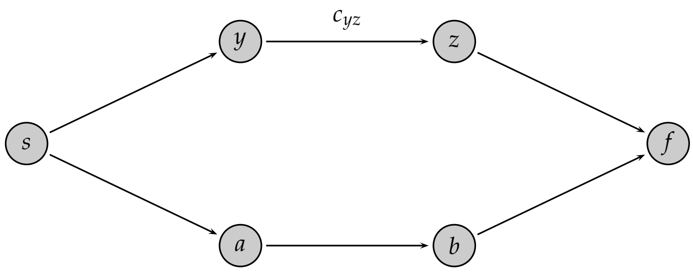


Figure 5.1: Cost dependency between service provider s_y and s_z.

Furthermore it is assumed that service provider s_y has the strategic opportunity to invest an amount I in order to customize its service offer y in a way that costs c_{yz} of service provider s_z are reduced from c^H_{yz} to c^L_{yz} with $c^H_{yz} > c^L_{yz}$. As s_y is familiar with its internal processes and properties of its service offer y, proportionate investment costs I are less then the effect of cost reduction for s_z such that $I < c^H_{yz} - c^L_{yz}$. Focusing on one-shot games, incorporating total fix costs for service customization in order to reduce variable costs caused by the preceeding service is not reasonable. Therefore I constitutes *proportionate* investment costs as a fraction of the total fix costs for a particular auction conduction. The assumption is that these proportionate investment costs are less than the reduction in variable costs caused by the preceeding service.

Corollary 5.3 [Cooperation within the Value Chain]. *Given two service providers s_y and s_z that own service offers y and z with y being the predecessor service of z. Furthermore let Θ_{yz} be an enforceable ex-ante agreement that states that iff services y*

and z are allocated such that $e_{yz} \in f^*$ then service provider s_y is committed to invest I in order to reduce costs c_{yz} from c^H_{yz} to c^L_{yz}. Committing to an agreement Θ_{yz} is an equilibrium in weakly dominant strategies if $I \leq c^H_{yz} - c^L_{yz}$.

Proof 5.3 [COROLLARY 5.3]. *Let $\mathcal{U}^{*H}(e_{yz})$ be the overall utility of the path allocated by o that entails edge e_{yz} and costs c^H_{yz}. Analogously let $\mathcal{U}^{*L}(e_{yz})$ be the overall utility of the path allocated by o that entails edge e_{yz} and costs c^L_{yz}. Let further $\mathcal{U}^*_{-s_y}$ be the overall utility of the path allocated by o in the reduced graph without node y and all its incoming and outgoing edges. Service offer i is an arbitrary predecessor of y.*

The expected payoff of service provider s_y under the assumption that there is no agreement Θ_{yz} evolves as follows

$$E^{s_y} = P(\mathcal{U}^{*H}(e_{yz}) > \mathcal{U}^*_{-s_y}) \left[p_{iy} + (\mathcal{U}^{*H} - \mathcal{U}^*_{-s_y}) - \Delta t^{comp,s_y} - c_{iy} \right]$$

With the results of Theorem 5.2 that each service provider reports its type truthfully the equation can be simplified to

$$E^{s_y} = P(\mathcal{U}^{*H}(e_{yz}) > \mathcal{U}^*_{-s_y}) \left[\mathcal{U}^{*H} - \mathcal{U}^*_{-s_y} \right]$$

Analogously for service provider s_z

$$E^{s_z} = P(\mathcal{U}^{*H}(e_{yz}) > \mathcal{U}^*_{-s_z}) \left[\mathcal{U}^{*H} - \mathcal{U}^*_{-s_z} \right]$$

Assuming that s_y and s_z commit to the agreement Θ_{yz} expected payoffs evolve as follows

$$E^{s_y}_{\Theta_{yz}} = P(\mathcal{U}^{*L}(e_{yz}) > \mathcal{U}^*_{-s_y}) \left[\mathcal{U}^{*L} - \mathcal{U}^*_{-s_y} - I \right] \tag{5.5}$$

$$E^{s_z}_{\Theta_{yz}} = P(\mathcal{U}^{*L}(e_{yz}) > \mathcal{U}^*_{-s_z}) \left[\mathcal{U}^{*L} - \mathcal{U}^*_{-s_z} \right] \tag{5.6}$$

In order to be an equilibrium in weakly dominant strategies, the commitments θ_y and θ_z to agreement Θ_{yz} must be a weakly dominant strategy for service provider s_y and s_z. The strategy space of each service provider and corresponding expected payoffs are illustrated as a normal form game in Table 5.1.

The strategy θ is a weakly dominant strategy for each player if $E^{s_y}_{\Theta_{yz}} \geq E^{s_y}$ and $E^{s_z}_{\Theta_{yz}} \geq E^{s_z}$.

Based on the assumption that $c^H_{yz} > c^L_{yz}$ and the quasi-linearity of $\mathcal{U}$ it follows that $\mathcal{U}^{*H}(e_{yz}) < \mathcal{U}^{*L}(e_{yz})$. Consequently the probability of service offer y being allocated by o increases if s_y follows strategy θ_y such that $P(\mathcal{U}^{*H}(e_{yz}) > \mathcal{U}^*_{-s_y}) < P(\mathcal{U}^{*L}(e_{yz}) > \mathcal{U}^*_{-s_y})$. If investment costs I for service provider y are lower (or at least equal) compared to the cost reduction $c^H_{yz} - c^L_{yz}$ for service provider z it can be derived that $\mathcal{U}^{*H} - I \leq \mathcal{U}^{*L}$. Finally it can be concluded that $E^{s_y}_{\Theta_{yz}} \geq E^{s_y}$.

Table 5.1: Cooperation decision as a normal form game. θ denotes an ex-ante commitment to an agreement Θ whereas $\bar{\theta}$ states the decision not to commit to an agreement Θ.

y,z	θ	$\bar{\theta}$
θ	$E^{s_y}_{\Theta_{yz}}, E^{s_z}_{\Theta_{yz}}$	E^{s_y}, E^{s_z}
$\bar{\theta}$	E^{s_y}, E^{s_z}	E^{s_y}, E^{s_z}

As the service provider s_z can only benefit from a cost reduction the same argumentation leads to $P(\mathcal{U}^{*H}(e_{yz}) > \mathcal{U}^*_{-s_z}) < P(\mathcal{U}^{*L}(e_{yz}) > \mathcal{U}^*_{-s_z})$, $\mathcal{U}^{*H} < \mathcal{U}^{*L}$ and directly to $E^{s_z}_{\Theta_{yz}} > E^{s_z}$. □

Example 5.1 [Cooperation within the Value Chain]. *To illustrate Corollary 5.3 and its implications for cooperative behavior in service value networks, Example 2.1 is consulted. For the reader's convenience the complex service is reduced to the first two business transactions, data verification and transaction processing. Figure 5.2 shows the service value network with service offers and corresponding costs. Each feasible path from s to f represents a possible instantiation of the payment processing complex service. Data verification can be performed by either StrikeIron (s_y) and its service offer y or Duo Share (s_a) offering service a. The execution of the actual monetary transaction can be done by JETTIS (s_z) offering service z or service b offered by Net Billing (s_b).*

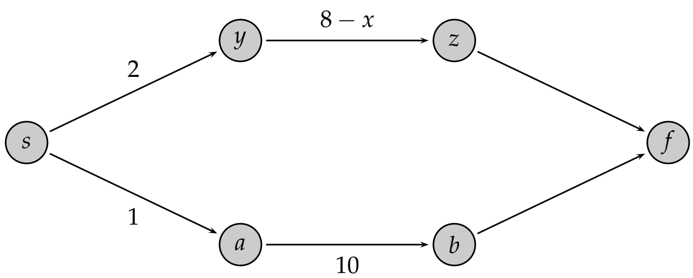


Figure 5.2: Cooperation within the value chain of a payment processing complex service.

A mandatory step for a transaction processing service is the credit assessment. As a precondition, a transaction processing service has to check if the customer is credit worthy in order to charge the corresponding account. The credit assessment has to be performed at a central authority and generates variable costs each time the transaction processing service is executed. The predecessor service that verifies the customer's data has to consult the same central authority to assure the correctness of processed data.

The provider of the data verification service has the strategic opportunity to customize its internal process in a way that a credit assessment is done on the fly which is cheaper than doing it in the second transaction. In other words if service provider s_y agrees to bear *proportionate investment costs of* $I \in \mathbb{R}^+$ *with* $I \leq x$ *to customize its internal process in order to enable credit assessment in case of* e_{yz} *being allocated, service provider* s_z *can reduce its costs by* $x \in \mathbb{R}^+$.

To analyze the effect of such an agreement Θ_{yz} *according to Corollary 5.3 two cases are considered:*

1. *There is no conclusion to agreement* Θ_{yz} *such that* $x = 0$

   The top path f^T *consisting of service offer y and z such that* $f^T = \{e_{sy}, e_{yz}, e_{zf}\}$ *generates a welfare of* $\mathcal{U}_{f^T} = \alpha - 10$ *whereas the bottom path* $f^B = \{e_{sa}, e_{ab}, e_{bf}\}$ *generates a welfare of* $\mathcal{U}_{f^B} = \alpha - 11$. *Consequently service offers y and z are allocated by o such that* $f^* = \{e_{sy}, e_{yz}, e_{zf}\}$. *Each service provider that owns a service that is allocated receives its transfer. Service provider* s_y *is payed* $t^{s_y} = 2 + (11 - 10) = 3$ *and* s_z *gets* $t^{s_z} = 8 + (11 - 10) = 9$. *This leads to a payoff for provider* s_y *of* $\pi^{s_y} = 1$ *and for service provider* s_z *of* $\pi^{s_z} = 1$. *The requester's utility evolves as* $\mathcal{U}^R = \alpha - 12$.

2. *Both parties agree on* Θ_{yz} *such that costs for* s_z *are reduced by x*

   In this case the top path f^T *consisting of service offer y and z such that* $f^T = \{e_{sy}, e_{yz}, e_{zf}\}$ *generates a welfare of* $\mathcal{U}_{f^T} = \alpha - 10 + x$ *whereas the bottom path* $f^B = \{e_{sa}, e_{ab}, e_{bf}\}$ *generates a welfare of* $\mathcal{U}_{f^B} = \alpha - 11$. *Analogue to the first case, service offers y and z are allocated by o such that* $f^* = \{e_{sy}, e_{yz}, e_{zf}\}$. *Service provider* s_y *is payed* $t^{s_y} = 2 + (11 - 10 + x) = 3 + x$ *and* s_z *gets* $t^{s_z} = 8 - x + (11 - 10 + x) = 9$. *This leads to a payoff for provider* s_y *of* $\pi^{s_y} = 1 + x$ *and for service provider* s_z *of* $\pi^{s_z} = 1$. *The requester's utility evolves as* $\mathcal{U}^R = \alpha - 12 - x$.

   The example shows that it is beneficial (or at least equally good) for adjacent service providers to commit to an agreement according to Corollary 5.3 as $\pi^{s_y}_{case\ 1} = 1 \leq \pi^{s_y}_{case\ 2} = 1 + x - I$ *and* $\pi^{s_z}_{case\ 1} = 1 \leq \pi^{s_z}_{case\ 2} = 1$.

Chapter 6

Numerical Results

In economic applications the analytical apparatus [...] diminishes the economic content of the models.

[KV98]

This chapter analyzes properties of the complex service auction and their extensions as well as strategic behavior of service providers by means of a simulation-based evaluation. In Section 6.1, the interoperability transfer function (ITF) is analyzed with respect to manipulation attempts of service providers that deviate from their truth-telling strategy. The question is answered to what degree bid manipulation is beneficial for service providers given different levels of competition in service value networks. Based on these results, Section 6.2 evaluates the incentives provided by the ITF which fosters interoperability endeavors of service providers, i.e. the ITF provides incentives for service providers to customize their services' interfaces to increase interoperability with adjacent service components. Focusing on bundling and unbundling strategies of service providers, Section 6.3 analyzes strategic behavior by means of an agent-based simulation. Based on these results strategic recommendations for service providers are derived depending on how they are situated within service value networks.

6.1 Manipulation Robustness of the ITF Extension

This section considerers strategic behavior of service providers participating in the complex service auction with the interoperability transfer function (ITF). Recall, in the basic complex service auction, allocated service providers are payed their price bid plus their critical value compensating their contribution to the whole system. This critical value is designed to implement a dominant strategy equilibrium in which every service provider reports its multidimensional type truthfully to the auctioneer according to Theorem 5.2.

Nevertheless, incentive compatibility comes at the price of losing budget balance, i.e. the sum of service providers' transfers may exceed the service requester's willingness to pay which results in a negative budget that has to be subsidized externally. As a possible remedy to retain budget balance, the ITF extending the basic complex service auction was introduced in Section 4.2. The ITF distributes the available surplus – the difference between the service requester's willingness to pay and the sum of providers' transfers – in a way that additionally to their bid, allocated providers are payed their critical value in the priority of their degree of interoperability subject to budget balance. It is obvious that in order to recover budget balance, incentive compatibility has to be sacrificed to a certain degree. Incurring this trade-off, the set of possibly beneficial bidding strategies of service providers increases and from a pure analytical perspective Theorem 5.2 does not hold under the presence of the ITF extension. Although the primary goal from an incentive engineering perspective of the ITF is to reward interoperability endeavors, the design of the ITF gives a good indication that bid manipulation is only beneficial to a certain level which strongly depends on the level of competition [Jac92, RP76, Hur72].

This section analyzes strategic behavior of service providers in the complex service auction with the ITF extension following a simulation-based approach (cp. Section 2.3.2).

6.1.1 Simulation Model

To analyze the manipulation robustness of the complex service auction with the ITF extension, a simulation is conducted as follows. A random service value network topology is created with density 1.0 (complete graph) and – depending on the degree of competition – with a predefined number of service offers and candidate pools. For simplicity and without loss of generality it is assumed that each service provider owns only a single service offer within the service value network. The competition rate results from the number of alternative complex service instances (number of feasible paths) without the participation of a single service provider. The number of feasible paths depends on the number of service offers within the network, the number of candidate pools and the density of the graph, i.e. the ratio between the number of edges and the number of all possible edges in the graph. The ratio between the number of service offers and the number of candidate pools is also responsible for the number of possible service compositions.

Each problem set is characterized by a random network topology with random costs c_{ij} assigned to each incoming edge of service offers drawn from $U(0,1.0)$. Furthermore, the requester's willingness to pay α is analogously drawn from $U(0, \frac{1}{2}K)$[1] with K being the number of candidate pools.

For each problem set, a random service offer's incoming edge e_{ij} is randomly drawn. The bid price p_{ij} is manipulated stepwise from 50% to 150% in steps of 10% of the truth-telling price $p_{ij} = c_{ij}$. For each manipulation rate the auction is conducted and the service provider's utilities for the deviation and the truth-telling strategies are computed based on the critical value transfer function and the ITF. Figure 6.1 depicts the stepwise procedure of the simulation.

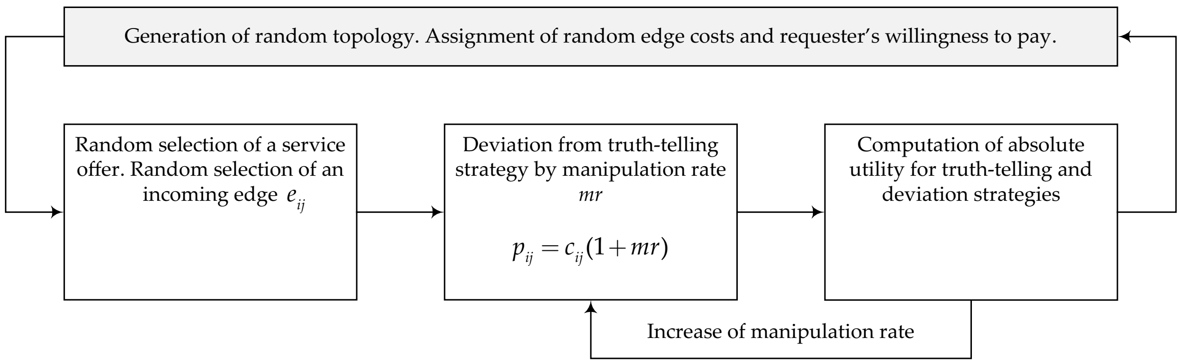


Figure 6.1: Simulation model for the evaluation of manipulation robustness using the ITF.

As the number of variable parameters and their interdependencies are high, heavy statistical noise is likely to be generated. To counteract the high volatility of the simulation model, a large number of problem sets of 5000 is evaluated for each degree of manipulation and the mean results are reported. In order to identify the degree of manipulation for which a deviation from the truth-telling strategy is beneficial for service providers, the statistical significance is tested using a one-tailed matched-pairs t-test analyzing the alternative hypothesis that service providers benefit from manipulation, that is, the mean difference in utility is greater than zero. The large size of analyzed problem sets for each observation assures robustness of the t-test to violations of the normality assumption [SB92, BS99, Ram80].

6.1.2 Results

Participating in the complex service auction with the ITF extension, service providers' strategies and corresponding outcomes are illustrated in Figure 6.2. The decision tree evaluates possible bidding strategies in comparison to a truth-telling strategy. Focusing on a single service provider, two fundamental cases must be considered in order to evaluate the result of different strategies:

[1] $\frac{1}{2}K$ denotes the mean price of a complex service in a network with K candidate pools and internal costs of service providers drawn from $U(0, 1.0)$ under the presence of truth-revelation.

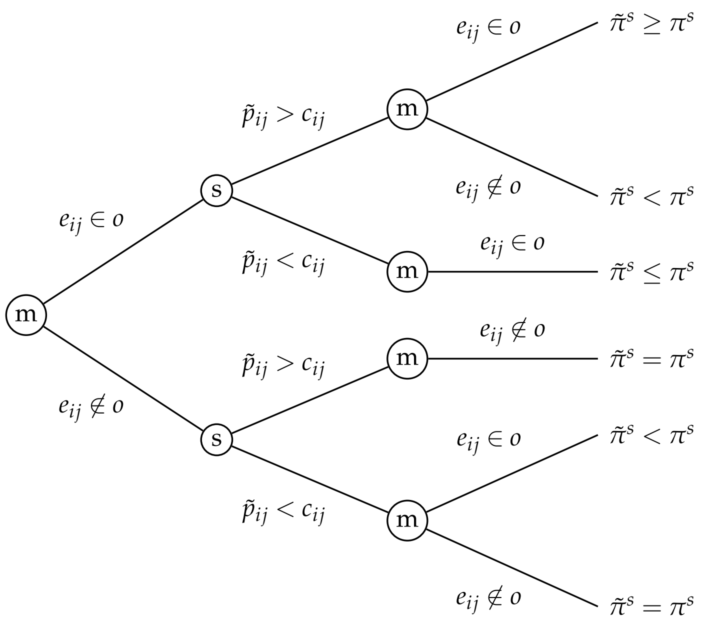


Figure 6.2: Decision tree of service providers.

1. Having followed a truth-telling strategy, the service provider s would have been allocated by o.

   In this case, *overstating* the true valuation by announcing a price $\tilde{p}_{ij} > c_{ij}$ leads to a payoff $\tilde{\pi}^s \geq \pi^s$ if the service providers stays allocated and to a payoff $\tilde{\pi}^s < \pi^s$ if it is dropped out of the allocation. The monotonicity of the allocation function assures that the service provider still gets allocated by understating the true valuation such that $\tilde{p}_{ij} < c_{ij}$ which leads to a payoff $\tilde{\pi}^s \leq \pi^s$.

2. Having followed a truth-telling strategy, the service provider s would not have been allocated by o.

   In this case, by *overstating* the true valuation announcing a price $\tilde{p}_{ij} > c_{ij}$, the service provider is not allocated due to monotonicity of the allocation function which leads to a payoff $\tilde{\pi}^s = \pi^s$. *Understating* the true valuation results in a payoff $\tilde{\pi}^s < \pi^s$ if the service provider gets allocated and to a payoff $\tilde{\pi}^s = \pi^s$ if it is not allocated.

The effect of a bid manipulation strategy of service providers is highly dependent on the level of competition in the service value network as this increases the risk of dropping out of the allocation by overstating ones true valuation. As market size increases, participants become price takers and strategic considerations converge towards a truth-telling strategy [Jac92, RP76, Hur72]. In the complex service auction, the level of competition results from the number of alternative paths in the absence of a single service provider. Therefore a good indication for the level of competition can be derived from the number of feasible paths in the network[2]. The lower the level of competition, the higher the benefit for service providers that deviate from their truth-telling strategy.

Table 6.1 shows the utility of a singe manipulating service provider in a low competition setting with 12 service offers in 4 candidate pools. Understating one's true valuation results in a negative utility gain compared to a truth-telling strategy. However, service providers that overstate their true valuation significantly benefit from a deviation up to 100% of their true valuation.

Table 6.1: Utility for a single manipulating service provider with 12 service offers in 4 candidate pools. *abs* denotes the mean absolute utility and *rel* the ratio of means of the utility with manipulation and the utility following a truth-telling strategy. *sd* is the standard deviation of the mean absolute utility. * denotes significance at the level of $p = 0.1$, ** at $p = 0.05$, and *** at $p = 0.01$.

	Critical Value Transfer			**Interoperability Transfer**		
Manipulation Rate	abs	rel	sd	abs	rel	sd
-50%	0.0423	0.5865	0.0793	-0.0209	-0.6871	0.1022
-40%	0.0562	0.7789	0.0506	-0.0009	-0.0308	0.0714
-30%	0.0631	0.8741	0.0334	0.0113	0.3645	0.0478
-20%	0.0693	0.9603	0.0136	0.0194	0.6763	0.0264
-10%	0.0715	0.9904	0.0050	0.0250	0.8795	0.0144
0%	0.0722	1.0000	0.0000	0.0302	1.0000	0.0000
10%	0.0715	0.9906	0.0050	0.0317	1.0688***	0.0125
20%	0.0705	0.9771	0.0097	0.0327	1.0968***	0.0199
30%	0.0703	0.9738	0.0102	0.0393	1.1380***	0.0283
40%	0.0696	0.9638	0.0137	0.0384	1.1776***	0.0355
50%	0.0673	0.9320	0.0261	0.0379	1.1774***	0.0435
60%	0.0640	0.8870	0.0383	0.0384	1.1016***	0.0445
70%	0.0627	0.8691	0.0424	0.0377	1.0866***	0.0486
80%	0.0603	0.8354	0.0508	0.0355	1.0535***	0.0449
90%	0.0596	0.8251	0.0521	0.0362	1.0233*	0.0475

[2]Based on the service value network model in Section 2.1.4, the number of feasible paths depends on the number of candidate pools and service offers per candidate pool. Assuming an equal number of service offers per pool, the number of paths is $\left(\frac{|V\setminus\{v_s,v_f\}|}{K}\right)^K$, where K denotes the number of candidate pools.

Table 6.1: Utility for a single manipulating service provider with 12 service offers in 4 candidate pools. *abs* denotes the mean absolute utility and *rel* the ratio of means of the utility with manipulation and the utility following a truth-telling strategy. *sd* is the standard deviation of the mean absolute utility. * denotes significance at the level of $p = 0.1$, ** at $p = 0.05$, and *** at $p = 0.01$.

	Critical Value Transfer			**Interoperability Transfer**		
Manipulation Rate	abs	rel	sd	abs	rel	sd
100%	0.0591	0.8181	0.0533	0.0351	1.0581***	0.0508
110%	0.0578	0.8006	0.0560	0.0378	1.0091	0.0537
120%	0.0554	0.7670	0.0632	0.0354	0.9652	0.0524
130%	0.0550	0.7613	0.0639	0.0314	0.9824	0.0543
140%	0.0534	0.7395	0.0672	0.0317	0.9529	0.0576
150%	0.0526	0.7285	0.0685	0.0344	0.9557	0.0581

A marginal increase in the level of competition decreases the number of beneficial manipulation strategies. Table 6.2 shows the simulation results in a setting with 16 service offers in 4 candidate pools. The utility of a single manipulating service provider is analyzed with respect to its manipulation rate. In this settings, deviation from truth-telling is only significantly beneficial – at a level of $p = 0.05$ – up to a manipulation rate of 60%. It is also noticeable that the mean utility gains of manipulation strategies compared to a truth-telling strategy are smaller and less favorable in comparison to the previous setting.

Table 6.2: Utility for a single manipulating service provider with 16 service offers in 4 candidate pools. *abs* denotes the mean absolute utility and *rel* the ratio of means of the utility with manipulation and the utility following a truth-telling strategy. *sd* is the standard deviation of the mean absolute utility. * denotes significance at the level of $p = 0.1$, ** at $p = 0.05$, and *** at $p = 0.01$.

	Critical Value Transfer			**Interoperability Transfer**		
Manipulation Rate	abs	rel	sd	abs	rel	sd
-50%	0.0171	0.4002	0.0757	-0.0081	-0.3140	0.0845
-40%	0.0300	0.7035	0.0465	0.0072	0.2799	0.0546
-30%	0.0383	0.8983	0.0217	0.0158	0.6344	0.0315
-20%	0.0413	0.9687	0.0095	0.0209	0.8354	0.0176
-10%	0.0424	0.9954	0.0027	0.0234	0.9331	0.0083
0%	0.0426	1.0000	0.0000	0.0248	1.0000	0.0000
10%	0.0425	0.9980	0.0013	0.0263	1.0453***	0.0070
20%	0.0420	0.9858	0.0055	0.0274	1.0659***	0.0131

Table 6.2: Utility for a single manipulating service provider with 16 service offers in 4 candidate pools. *abs* denotes the mean absolute utility and *rel* the ratio of means of the utility with manipulation and the utility following a truth-telling strategy. *sd* is the standard deviation of the mean absolute utility. * denotes significance at the level of $p = 0.1$, ** at $p = 0.05$, and *** at $p = 0.01$.

	Critical Value Transfer			**Interoperability Transfer**		
Manipulation Rate	abs	rel	sd	abs	rel	sd
30%	0.0403	0.9466	0.0144	0.0276	1.0334***	0.0213
40%	0.0402	0.9434	0.0149	0.0283	1.0562***	0.0246
50%	0.0394	0.9244	0.0180	0.0271	1.0570***	0.0282
60%	0.0382	0.8974	0.0227	0.0281	1.0256*	0.0309
70%	0.0373	0.8757	0.0261	0.0267	1.0170	0.0325
80%	0.0359	0.8418	0.0315	0.0268	0.9777	0.0376
90%	0.0352	0.8259	0.0339	0.0268	0.9607	0.0391
100%	0.0348	0.8168	0.0348	0.0276	0.9411	0.0395
110%	0.0329	0.7724	0.0414	0.0254	0.8877	0.0383
120%	0.0320	0.7504	0.0437	0.0245	0.8816	0.0412
130%	0.0314	0.7376	0.0463	0.0240	0.8616	0.0420
140%	0.0305	0.7153	0.0487	0.0246	0.8350	0.0444
150%	0.0299	0.7012	0.0506	0.0234	0.8274	0.0440

In the setting with 20 service offers in 4 candidate pools as shown in Table 6.3, service providers do not significantly gain from deviation of more than 20%. Although, the complex service auction with the ITF extension is not incentive compatible in a strict theoretical sense, in settings with relatively low competition (e.g. 28 service offers in 4 candidate pools), service providers cannot significantly benefit from deviation from reporting their true valuation as shown in Table 6.4, i.e. the truth-telling strategy is a best (or equally good) strategy compared to any manipulation strategy.

Table 6.3: Utility for a single manipulating service provider with 20 service offers in 4 candidate pools. *abs* denotes the mean absolute utility and *rel* the ratio of means of the utility with manipulation and the utility following a truth-telling strategy. *sd* is the standard deviation of the mean absolute utility. * denotes significance at the level of $p = 0.1$, ** at $p = 0.05$, and *** at $p = 0.01$.

	Critical Value Transfer			**Interoperability Transfer**		
Manipulation Rate	abs	rel	sd	abs	rel	sd
-50%	0.0025	0.1122	0.0630	-0.0111	-0.7315	0.0741

Table 6.3: Utility for a single manipulating service provider with 20 service offers in 4 candidate pools. *abs* denotes the mean absolute utility and *rel* the ratio of means of the utility with manipulation and the utility following a truth-telling strategy. *sd* is the standard deviation of the mean absolute utility. * denotes significance at the level of $p = 0.1$, ** at $p = 0.05$, and *** at $p = 0.01$.

	Critical Value Transfer			**Interoperability Transfer**		
Manipulation Rate	abs	rel	sd	abs	rel	sd
-40%	0.0107	0.4870	0.0425	0.0003	0.0187	0.0495
-30%	0.0173	0.7854	0.0231	0.0090	0.5533	0.0292
-20%	0.0208	0.9444	0.0089	0.0137	0.8251	0.0146
-10%	0.0219	0.9916	0.0020	0.0150	0.9434	0.0063
0%	0.0220	1.0000	0.0000	0.0167	1.0000	0.0000
10%	0.0219	0.9920	0.0017	0.0169	1.0298***	0.0059
20%	0.0215	0.9748	0.0051	0.0168	1.0227***	0.0086
30%	0.0205	0.9300	0.0108	0.0157	0.9929	0.0111
40%	0.0195	0.8849	0.0156	0.0150	0.9266	0.0143
50%	0.0191	0.8662	0.0169	0.0149	0.9129	0.0163
60%	0.0189	0.8562	0.0176	0.0150	0.8881	0.0166
70%	0.0185	0.8387	0.0197	0.0148	0.8794	0.0187
80%	0.0183	0.8324	0.0201	0.0153	0.8847	0.0201
90%	0.0182	0.8246	0.0207	0.0149	0.8776	0.0218
100%	0.0179	0.8125	0.0217	0.0149	0.8526	0.0220
110%	0.0176	0.7988	0.0235	0.0148	0.8480	0.0234
120%	0.0174	0.7888	0.0243	0.0154	0.8303	0.0266
130%	0.0168	0.7602	0.0270	0.0139	0.7904	0.0270
140%	0.0165	0.7474	0.0285	0.0139	0.7947	0.0293
150%	0.0163	0.7397	0.0293	0.0139	0.7869	0.0279

Table 6.4: Utility for a single manipulating service provider with 28 service offers in 4 candidate pools. *abs* denotes the mean absolute utility and *rel* the ratio of means of the utility with manipulation and the utility following a truth-telling strategy. *sd* is the standard deviation of the mean absolute utility. * denotes significance at the level of $p = 0.1$, ** at $p = 0.05$, and *** at $p = 0.01$.

	Critical Value Transfer			**Interoperability Transfer**		
Manipulation Rate	abs	rel	sd	abs	rel	sd
-50%	0.0000	0.0005	0.0501	-0.0048	-0.4739	0.0540
-40%	0.0081	0.6271	0.0247	0.0037	0.3617	0.0305
-30%	0.0103	0.8014	0.0152	0.0069	0.6498	0.0191
-20%	0.0119	0.9275	0.0070	0.0090	0.8521	0.0100
-10%	0.0127	0.9908	0.0014	0.0097	0.9500	0.0042
0%	0.0129	1.0000	0.0000	0.0101	1.0000	0.0000
10%	0.0127	0.9873	0.0018	0.0108	1.0044	0.0029
20%	0.0122	0.9489	0.0058	0.0101	0.9681	0.0063
30%	0.0120	0.9315	0.0069	0.0107	0.9546	0.0080
40%	0.0119	0.9240	0.0072	0.0099	0.9526	0.0084
50%	0.0116	0.9059	0.0088	0.0098	0.9350	0.0103
60%	0.0113	0.8799	0.0110	0.0099	0.9054	0.0123
70%	0.0109	0.8455	0.0133	0.0098	0.8773	0.0141
80%	0.0106	0.8232	0.0146	0.0094	0.8464	0.0144
90%	0.0104	0.8083	0.0154	0.0092	0.8546	0.0163
100%	0.0099	0.7667	0.0181	0.0087	0.7969	0.0187
110%	0.0099	0.7667	0.0181	0.0088	0.8045	0.0183
120%	0.0095	0.7410	0.0199	0.0087	0.7596	0.0212
130%	0.0093	0.7208	0.0216	0.0081	0.7390	0.0229
140%	0.0091	0.7089	0.0223	0.0083	0.7360	0.0228
150%	0.0089	0.6937	0.0231	0.0082	0.7289	0.0224

Providing an overview over multiple settings with different levels of competition, Figure 6.3 illustrates the relative utility gain following a manipulation strategy compared to truth-telling.

More detailed results of the simulation-based analysis with respect to different competition scenarios can be found in Section A.4.

Figure 6.3: Utility for a single manipulating service provider in different competition scenarios. ITF_$|\tilde{V}|$_K denotes the setting with $|\tilde{V}|$ service offers in K candidate pools, where $|\tilde{V}| = V \setminus \{v_s, v_f\}$.

6.1.3 Implications

In Section 6.1, strategic behavior of service providers has been analyzed in the complex service auction with the interoperability transfer in comparison to the complex service auction with the critical value transfer.

As shown analytically in Section 5.1, the complex service auction with critical value transfer implements a truth-telling equilibrium in weakly dominant strategies, i.e. service providers cannot benefit from misreporting their true valuation. This is a valuable property for a mechanism and the implemented social choice as it assures truthful behavior of all participants which allows for an efficient allocation that maximizes welfare among service providers and the service requester. It furthermore reduces the strategy space of beneficial strategies to a single weakly dominant strategy independent of the strategies of all other participants. This implies that service providers do not have to reason about the behavior of other participants in the complex service auction.

Incentive compatibility comes at the price of budget balance. As a remedy for this shortcoming, the ITF has been introduced in Section 4.2. The ITF sacrifices incentive compatibility and efficiency to a certain degree in order to retain budget balance. The ITF furthermore rewards service providers that offer highly interoperable services within the service value network, which increases the number of feasible service compositions that can be offered to the requester. Thus, the ITF implements incentives to increase a services' interoperability and therefore fosters the growth of vital and more agile service value networks. This property is analyzed in detail in Section 6.2.

Using the complex service auction with the critical value transfer as a benchmark, the robustness of the complex service auction with the ITF extension has been analyzed with respect to bid manipulation (deviation from the truth-telling strategy). The simulation-based results show that in scenarios with a low level of competition, implementing the ITF extension opens up strategic behavior to a certain degree. Service providers can significantly benefit from misreporting their true valuation. Nevertheless, in settings with a slightly higher level of competition (e.g. 20 service offers in 4 candidate pools), the set of beneficial manipulation strategies is decreased tremendously. Although the complex service auction with the ITF extension is not incentive compatible in a strict analytical sense, service providers cannot significantly benefit from misreporting their true valuation in settings with a still relatively low level of competition (e.g. cp. results in Table A.5 in a setting with 28 service providers in 4 candidate pools).

As the attraction of service value networks underlays network externalities, the value that service requesters gain from initiating a complex service auction highly depends on the number of participating service providers and the number of feasible complex service instances that can be provided through the network. Hence, especially in an early growing stage of a service value network, it might be desirable for platform providers to implement a mechanism that rewards service providers for offering multiple services with a high degree of interoperability, such as the complex service auction with the ITF extension does. Especially in settings with a low level of competition, critical values of service providers can be relatively high and unpredictable for the platform provider. Hence, a budget-balanced variant might be favorable in such an early stage as well. Reaching a critical mass of participants the network's inherent competition increases and critical values of service providers tremendously decrease. Assuring complete truthful behavior of service provider, the complex service auction with the critical value transfer might be beneficial for both service providers and the service requester. Service providers do not have to reason about the other participants' behavior and the service requester trustfully receives a tailored complex service instance. This variant always assures a welfare maximizing solution accounting for the providers' and the requester's side.

6.2 Incentivizing Interoperability Endeavors

The interoperability transfer function (ITF) is designed as a remedy to overcome the lack of budget balance in the complex service auction. The design goal of the ITF is on the one hand to reduce strategic behavior of service providers with respect to beneficial deviation from the truth-telling strategy as evaluated in Section 6.1. On the other hand the design of the ITF targets to incentivize service providers to increase their services' degree of interoperability, i.e. to increase the capability of their offered services to communicate and function with other services within the service value network. A higher degree of interoperability increases the potential of a service value network to satisfy different customers' needs and to provide a huge variety of feasible complex service instances to requesters. Increasing customers' choice leads to a rapid growth of demand and addresses the long tail of business [And06](cp. Section 2.1.4.3). These implications are especially important for service value networks in their early stage of development as it attracts various customers which leads to a growth of rich candidate pools by attracting service providers to participate in value creation (the effect of network externalities is well-known in literature [SV99, FK07, LM94, KS85]).

To study the effect of the ITF on the network's degree of interoperability, the work at hand follows the research method of an agent-based simulation as outlined in Section 2.3.2. As a suitable benchmark to evaluate incentives implemented by the ITF, an Equal Transfer Function (ETF) is consulted that distributes the system's surplus equally among all allocated service providers [PKE01][3]. The ETF represents a neutral payment scheme as it equally distributes the same surplus as the ITF. The goal of this evaluation is to analyze if and to what degree increasing the interoperability degree of service offers within a service value network is beneficial for service providers in the complex service auction with the ITF compared to the complex service auction with the ETF. This leads to the following hypotheses:

Hypothesis 6.1. *The overall interoperability degree of a service value network increases by establishing the ITF compared to the ETF.*

Hypothesis 6.2. *The interoperability degree of allocated service offers increases using the ITF compared to the ETF.*

Hypothesis 6.3. *The interoperability degree of non-allocated service offers increases using the ITF compared to the ETF.*

[3] The equal transfer function that serves as a benchmark is similar to the k-pricing scheme in [Sto09, Sch07] with parameter selection $k = 1$

6.2.1 Simulation Model

According to the design of the ITF, allocated service providers can gain by increasing their degree of interoperability as this increases their chance of receiving their critical value as a discount in addition. Nevertheless, in the complex service auction with the ETF it might also be beneficial to increase one's degree of interoperability. Focusing on non-allocated service offers, by building additional connections to predecessor services proactively, service providers face the opportunity to change the network's topology and augment the chance of being allocated. It is unclear which effect dominates in settings with different levels of competition and different proportionate investment costs.

Each service provider is assumed to have a set of strategies representing the degree of its service's interoperability that the service provider intents to realize depending on how it is situated within the network[4]. This means that depending on the number of predecessor services, service providers can decide on how many edges to predecessor services they want to establish. Recall, an edge between two adjacent services denotes the capability of interpreting each others inputs and outputs, i.e. both services are interoperable and therefore can be iteratively combined within a complex service instance.

Each agent's[5] strategy space is determined by all feasible degrees of interoperability (*ID*) of its service offer represented by its number of incoming edges. E.g. if a service offer has 4 predecessor services within the service value network and the initial number of incoming edges is 2, the service provider's strategy space is $\{2,3,4\}$.

For each extra edge built additionally to the initial number of incoming edges the service provider is charged proportionate investment costs (*IVC*) no matter if the service is being allocated or not[6]. Proportionate investment costs are calculated as a fraction of the internal costs for executing the particular service depending on the predecessor service. It is assumed that internal costs for context-dependent execution reflect the degree of similarity of both services' interfaces (e.g. low internal costs indicate a high degree of interfaces' compatibility). Hence, investment costs for reprogramming a service's interface in order to work seamlessly with another service component behave accordingly.

[4]For simplicity it is assumed that each service provider owns only a single service within the network

[5]In the context of the agent-based simulation, the terms *service provider* and *agent* are used interchangeably.

[6]It is important to note that the complex service auction is conducted as a one-shot game which has to be considered when evaluating specific properties. Therefore, accounting for full investment costs that are necessary to reprogram a service's interface in order to enable interoperability with certain other services results in prohibitively high costs which hinders a feasible one-shot game analysis.

Analogue to Section 6.1.1, each problem set is characterized by a random network topology with random costs c_{ij} assigned to each incoming edge of service offers drawn from $U(0,1.0)$. Furthermore, the requester's willingness to pay α is analogously drawn from $U(0,\frac{1}{2}K)$ with K being the number of candidate pools.

The evaluation is conducted by means of an agent-based simulation based on a simple form of a Q-Learning model [WD92]. In contrary to more sophisticated variants of Q-learning models, the simulation model at hand only considers a single state which reduces the parameter complexity and therefore simplifies the calibration of the simulation. Simplifying the simulation model reduces the number of assumptions which allows for a better generalization of results.

Each agent maintains a fitness table which keeps track of the "successfulness" of each action such that $\mathfrak{f}^r_{ik}$ represents the fitness of agent i for action k in simulation round r. The fitness for each chosen action is updated based upon the resulting "reward" (represented by the agent's utility u^r_{ik}). Balancing past and present experiences, the learning parameter $\beta \in [0;1]$ determines to which degree past and present feedback is incorporated into the fitness update. Thus, the fitness update evolves as follows:

$$\mathfrak{f}^r_{ik} = \beta \mathfrak{f}^{r-1}_{ik} + (1-\beta) u^r_{ik} \tag{6.1}$$

Each action is selected based on a *softmax* selection method [SB99], i.e. each action is randomly chosen based on the probability P^r_{ik} that results from the action's fitness relative to the sum of all actions' fitness such that

$$P^r_{ik} = \frac{\mathfrak{f}^r_{ik}}{\sum_k \mathfrak{f}^r_{ik}} \tag{6.2}$$

The simulation is conducted as depicted in Figure 6.4. The simulation process is divided into an *exploration phase* and a *simultaneous exploitation phase*.

Exploration Phase In this phase each agent explores the solution space in a constant environment where only a single agent learns simultaneously. Starting based on an initial fitness table with equal probabilities for every action, each agent adapts its individual best action given the other agents do not change their decisions. The exploration phase is conducted 100 rounds [7] for each agent $i \in V \setminus \{v_s, v_f\}$.

[7] The number of required rounds in order to achieve a convergence of the fitness values for each action has been analyzed by means of a sensitivity analysis.

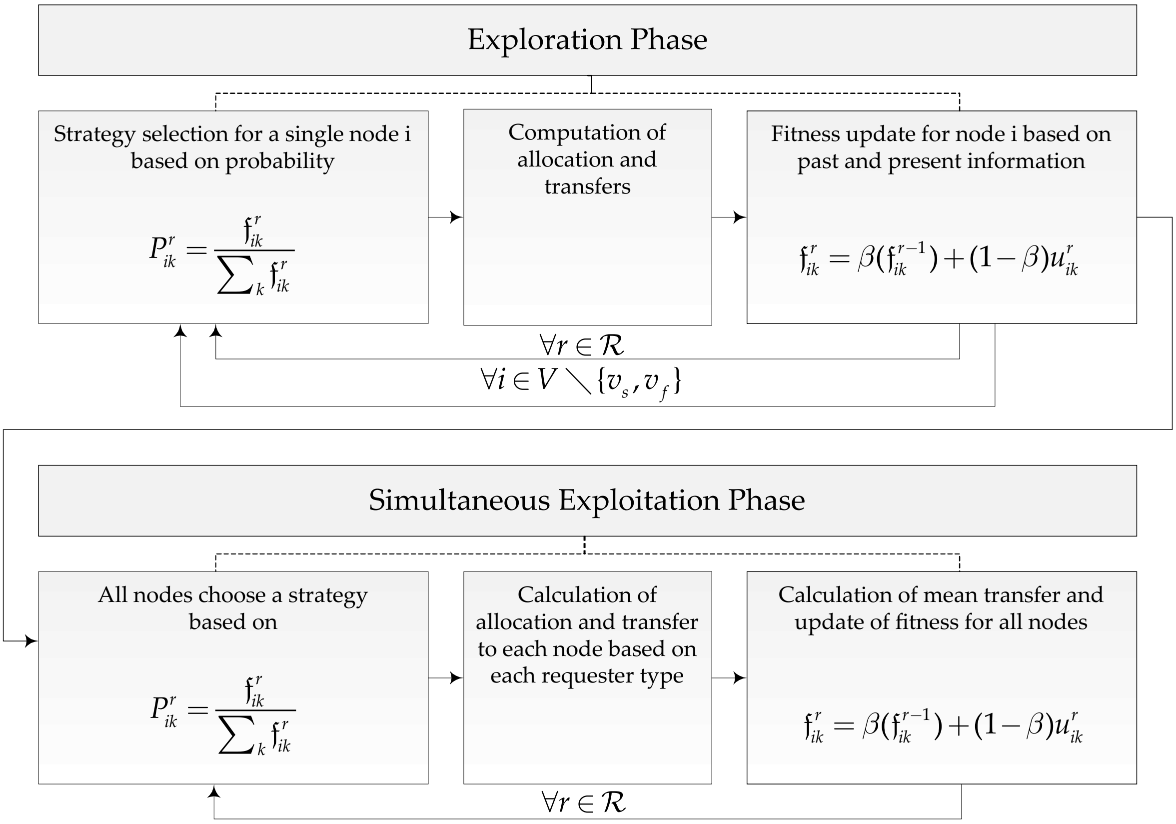


Figure 6.4: Simulation model for the evaluation of interoperability incentives using the ITF.

Simultaneous Exploitation Phase In order to determine the most promising action for each agent dependent on the decision taken by every other agent, in the exploitation phase every agent learns its best action simultaneously based on the experiences gained from the exploration phase. The simultaneous exploration phase is conducted 100 rounds. [7]

As the number of observations is relatively high ($N = 50$) and the data is normally distributed which has been tested by means of a Kolmogorov-Smirnov test, stated hypothesis are tested using a one-tailed matched-pairs t-test. With respect to the overall network, allocated, and non-allocated service offers, the alternative hypothesis that the interoperability degree of a service value network increases by establishing the ITF compared to the ETF is analyzed, i.e. the mean difference in interoperability degrees is greater than zero.

6.2.2 Results

Recall, the complex service auction with the interoperability transfer function (ITF) is designed to incentivize service providers to increase their services' degree of interoperability. In order to evaluate this property, the ITF is benchmarked against an equal transfer function (ETF) which distributes the system's surplus among all allocated service providers equally.

Table 6.5 and Figure 6.5 show a comparison of the ITF and the ETF with respect to resulting interoperability degrees (*ID*) at different levels of proportionate investment costs (*IVC*) for 20 service offers in 4 candidate pools.

Table 6.5: Interoperability degrees (*ID*) at different levels of proportionate investment cost (*IVC*) for 20 service offers in 4 candidate pools. *ID* denotes the overall interoperability degree, *ID_A* denotes the interoperability degree of all allocated service offers, and *ID_NA* denotes the interoperability degree of all non-allocated service offers. * denotes significance at the level of $p = 0.1$, ** at $p = 0.05$, and *** at $p = 0.01$.

	Critical Value Transfer			**Interoperability Transfer**		
IVC	ID	ID_A	ID_NA	ID	ID_A	ID_NA
0%	0.6665	0.7571	0.6438	0.6766***	0.7711***	0.6530***
10%	0.4595	0.6025	0.4238	0.4891***	0.6710***	0.4436***
20%	0.3676	0.4811	0.3392	0.3963***	0.5780***	0.3509***
30%	0.3343	0.4201	0.3129	0.3544***	0.4934***	0.3196***
40%	0.3199	0.3838	0.3040	0.3347***	0.4474***	0.3065***
50%	0.3201	0.3831	0.3043	0.3321***	0.4394***	0.3053*
60%	0.3147	0.3576	0.3039	0.3218***	0.3899***	0.3048**
70%	0.3118	0.3355	0.3059	0.3164***	0.3616***	0.3051*
80%	0.3145	0.3612	0.3029	0.3196***	0.3854***	0.3032
90%	0.3097	0.3407	0.3019	0.3133***	0.3616***	0.3013
100%	0.3111	0.3617	0.2985	0.3137***	0.3772***	0.2979
110%	0.3101	0.3542	0.2990	0.3113***	0.3614***	0.2988
120%	0.3150	0.3789	0.2990	0.3159***	0.3841***	0.2989
130%	0.3084	0.3749	0.2918	0.3110***	0.3877***	0.2918
140%	0.3114	0.3504	0.3017	0.3122***	0.3537***	0.3018
150%	0.3091	0.3431	0.3006	0.3101***	0.3479**	0.3007
160%	0.3101	0.3407	0.3025	0.3111**	0.3469**	0.3022
170%	0.3076	0.3416	0.2991	0.3080*	0.3437*	0.2991
180%	0.3115	0.3563	0.3003	0.3076*	0.3505	0.2969
190%	0.3126	0.3539	0.3022	0.3126	0.3541	0.3022

Table 6.5: Interoperability degrees (*ID*) at different levels of proportionate investment cost (*IVC*) for 20 service offers in 4 candidate pools. *ID* denotes the overall interoperability degree, *ID_A* denotes the interoperability degree of all allocated service offers, and *ID_NA* denotes the interoperability degree of all non-allocated service offers. * denotes significance at the level of $p = 0.1$, ** at $p = 0.05$, and *** at $p = 0.01$.

	Critical Value Transfer			**Interoperability Transfer**		
IVC	ID	ID_A	ID_NA	ID	ID_A	ID_NA
200%	0.3098	0.3598	0.2973	0.3101	0.3613	0.2973

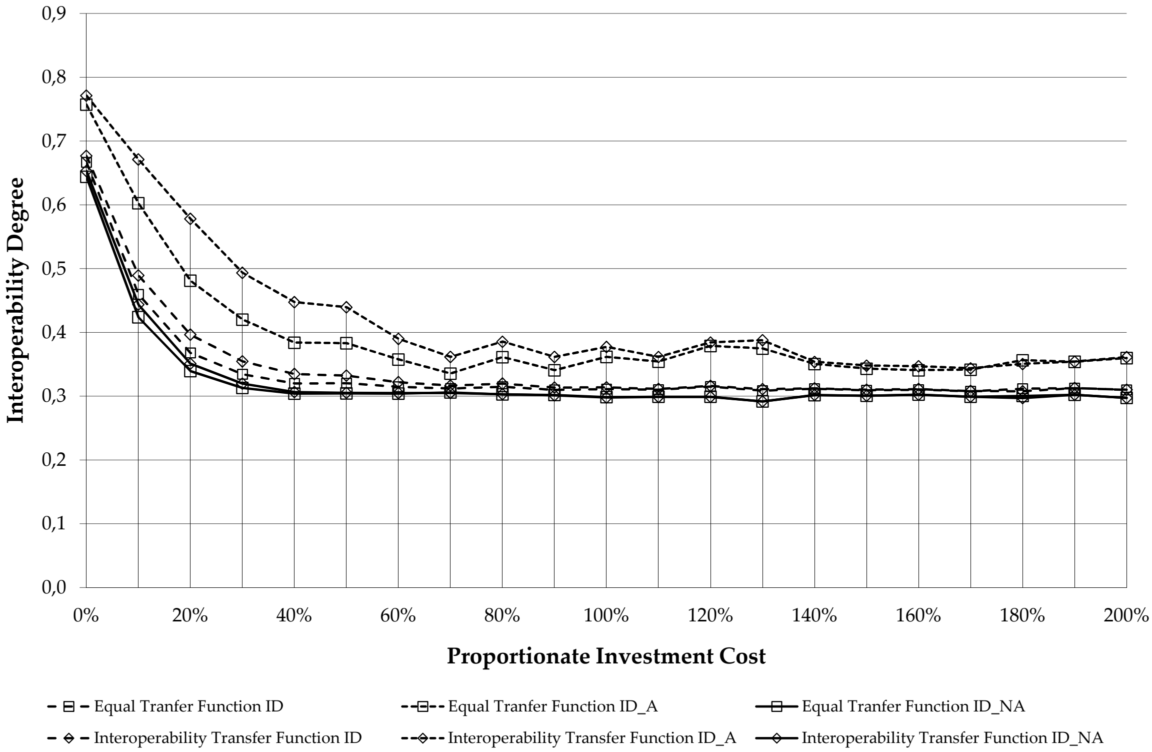


Figure 6.5: Interoperability degrees (*ID*) at different levels of proportionate investment Cost (*IVC*) for 20 service offers in 4 candidate pools. *ID* denotes the overall interoperability degree, *ID_A* denotes the interoperability degree of all allocated service offers, and *ID_NA* denotes the interoperability degree of all non-allocated service offers.

In general, it is observable that an increase of proportionate investment costs results in a decrease of interoperability degrees with respect to both transfer functions. Investment costs are obviously a disincentive for increasing ones services' degree of interoperability and therefore counteract the incentive schema provided by the ITF. Despite of the primary incentives provided by the transfer function, service providers might also have an incentive to increase their degree of interoperability independent of the design of the transfer function as establishing more relations to other services allows for proactively changing the initial topology of the service value network. By doing so, service providers face the opportunity to be better situated within the network and increase the likelihood of being allocated. Thus, proportionate investment costs also disincentivize interoperability endeavors under the presence of a "neutral" transfer function such as the ETF which results in a decrease of interoperability degrees with respect to both transfer functions.

Furthermore *the degree of interoperability is higher for allocated service offers than for non-allocated services offers.* The reason for this phenomenon is based on the fact that service offers that are initially more interoperable with other services face a higher likelihood of being allocated than service offers with a low degree of interoperability. Hence, independent of the design of the transfer function, allocated services yield a higher degree of interoperability than non-allocated services. Nevertheless the difference in interoperability between allocated and non-allocated services decreases as proportionate investment costs increase. Due to the fact that investment costs are a disincentive for being interoperable, each service's interoperability degree is pushed down towards the initial density of the service value network.

In the setting with 20 service offers in 4 candidate pools (cp. Table 6.5), Hypothesis 6.1 is supported significantly until a level of proportionate investment costs of 180%. Distinguishing between allocated and non-allocated service offers, Hypothesis 6.2 is supported until 170% investment costs and Hypothesis 6.3 is significantly supported up to 70% proportionate investment costs. The difference in the levels of investment costs until each hypothesis is supported bases on two effects. First, *allocated services are primarily incentivized by the construction of the ITF* whereas non-allocated services only benefit from a higher degree of interoperability if they are allocated in the changed topology. Hence, *for service providers that own non-allocated services, the effect of the implemented incentive is compensated earlier by the disincentive provided through the investment costs*. The second effect for the different support levels of each hypothesis is based on the fact that there are more discrete degrees of interoperability for the overall network than for a subset of service offers. This means that as allocated service offers are rare, if a single service's degree of interoperability decreases, the overall degree of interoperability for all allocated services drops rapidly.

Looking at different levels of competition in the service value network, Table 6.6 shows a comparison of the ITF and the ETF with respect to resulting interoperability degrees at different levels of proportionate investment costs for 32 service offers in 4 candidate pools.

Table 6.6: Interoperability degrees (*ID*) at different levels of proportionate investment cost (*IVC*) for 32 service offers in 4 candidate pools. *ID* denotes the overall interoperability degree, *ID_A* denotes the interoperability degree of all allocated service offers, and *ID_NA* denotes the interoperability degree of all non-allocated service offers. * denotes significance at the level of $p = 0.1$, ** at $p = 0.05$, and *** at $p = 0.01$.

	Critical Value Transfer			**Interoperability Transfer**		
IVC	ID	ID_A	ID_NA	ID	ID_A	ID_NA
0%	0.6118	0.7298	0.5949	0.6189***	0.7369***	0.6020***
50%	0.2026	0.2474	0.1962	0.2051***	0.2642***	0.1966*
100%	0.2015	0.2453	0.1952	0.2017***	0.2472**	0.1952*
150%	0.2016	0.2427	0.1957	0.2016*	0.2433*	0.1957
200%	0.2004	0.2369	0.1952	0.2004	0.2369	0.1952

Compared to the previous setting, the overall incentive provided by the ITF to increase interoperability is weakened. At a level of 150% proportionate investment costs, Hypothesis 6.1 and 6.2 are only supported at a level of $p = 0.1$ whereas Hypothesis 6.3 is not supported at all. A higher level of competition decreases critical values of service providers. Thus, increasing ones degree of interoperability to obtain ones critical value is less favorable in highly competitive settings.

6.2.3 Implications

In Section 6.2 the interoperability transfer function (ITF) is analyzed with respect to its design to incentivize service providers to increase their services' degree of interoperability. The evaluation is conducted by means of an agent-based simulation comparing the complex service auction with the ITF extension and the ITF with an equal transfer function (ETF) that distributes the available surplus equally among service providers that own allocated service offers within the service value network.

Summarizing the results in Section 6.2.2, the ITF extension incentivizes service providers – those which own allocated (cp. Hypothesis 6.2) and non-allocated (cp. Hypothesis 6.3) service offers – to increase their services' degree of interoperability as stated by Hypothesis 6.1. That is, the design of the ITF implements incentives to undertake endeavors to customize service interfaces which enables communication and data transfer with multiple adjacent service components. Of course, proportionate investment costs that service providers have to bear for this customization process function as a disincentive counteracting interoperability endeavors. In general, in service value networks with a low level of competition and only few interrelated service offers, the ITF extension appears to be a promising approach to foster the growth of service value networks' variety in an early stage of development and to increase the multitude of feasible complex service instances that can be offered to customers. An increase of variety and interoperability leverages network externalities [SV99, FK07, LM94, KS85] and attracts customers which in turn attracts more service providers to participate in the complex service auction.

6.3 Bundling Strategies of Service Providers

Recall, in Section 5.1 it has been shown that under the assumption of rationality, service providers act best (or at least equally good) by revealing their true multi-dimensional type which reduces their bidding strategy space to a single strategy. Broadening service providers' strategic horizon, it might be beneficial under certain circumstances to form coalitions and offer services in a bundled fashion. This section focuses on strategies of service providers with focus on opportunities to form bundled offers with other providers depending on how they are situated within service value networks.

Since a service provider's offer is only successful if one of its edges is allocated, service providers tend to find strategies to improve their situation. Two options are mainly distinguished, *unbundling* vs. *bundling*. Service providers can decide on either offering services on their own with a certain degree of interoperability to preceeding offers. Such a strategy is referred to as *unbundling strategy*. On the other hand service providers can also provide bundled services together with service providers that own services in adjacent candidate pools (either preceeding or succeeding), i.e. two service providers from different candidate pools combine their offers to a single service which aggregates both service characteristics. It is furthermore assumed that a combined service offer results in lower internal costs due to synergy effects that can be leverage through bundled offers. This strategy is referred to as *bundling strategy*. There are mainly two contrary effects and it is unclear which effect dominates in what setting.

Competing in quality through differentiation and flexibility It is certainly just reasonable to follow an unbundling strategy if a provider's service offers expose significantly lower prices (due to lower internal costs) or significantly better QoS characteristics than competing offers. Additionally, unbundled services offer more differentiated and specialized functionality which increases their flexible integration into different complex services, and thus, increase the number of possible combinations with other services from other candidate pools.

Competing in price through cost reduction On the other hand, it might be advantageous for service providers to cut costs through forming bundled offers collaboratively, i.e. combining their service offers to a service bundle which offers the functionality of both services in an integrated manner. In that case internal costs of the bundled services are likely to be lower compared to the sum of internal costs of two single offers. In the case of bundling, an aggregation of attribute values defining the service's configuration is done according to aggregation operations in Table 3.1. Nevertheless, bundling service offers results in a reduction of the degree of interoperability, i.e. a merge of service offers prunes incoming edges to preceeding services which decreases the number of complex service instances the bundled offer is part of.

It is unclear which strategy is beneficial for service providers with respect to how their service offers are situated within the service value network. Even for service offers that are competitive in price and attractive for the service requesters – i.e. they are allocated solely – forming a bundled offer with a less competitive service offer may be mutually beneficial for both partners. The following example illustrates the phenomenon where a bundling strategy is mutually beneficial for an allocated and a non-allocated service provider at the same time even though there is no reduction of internal costs due to bundling synergies assumed:

Example 6.1 [Beneficial Bundling Strategy]. *Figure 6.6 depicts the service value network from an initial ex-ante perspective. Without loss of generality it is assumed that service providers only announce price bids (no QoS) and each service provider only owns a single service offer within the service value network. Consequently there are four service providers* s_y, s_z, s_a, s_b *that own service offers* y, z, a, b*. Numbers on incoming edges to each node represent price bids placed by service providers*[8]*.*

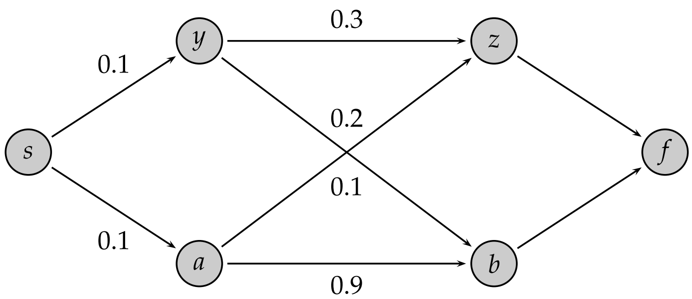


Figure 6.6: Beneficial bundling strategy for allocated and non-allocated service providers (ex-ante case).

According to the CSA mechanism, the path $f^* = \{e_{sa}, e_{az}, e_{zf}\}$ *is allocated as it yields the overall lowest price of* 0.2 *and therefore maximizes welfare. The "second-best" path*

[8]Note that according to Theorem 5.2 it is a dominant strategy equilibrium in the CSA that service providers report their valuations truthfully, that is, they announce their internal costs.

$f^2 = \{e_{sy}, e_{yb}, e_{zf}\}$ yields an overall price of 0.3. According to the CSA's transfer function, payments are given to allocated service providers such that $t^{s_a} = 0.1 + (0.3 - 0.2) = 0.2$ and $t^{s_z} = 0.1 + (0.3 - 0.2) = 0.2$.

Focusing on the ex-post case depicted in Figure 6.7, service providers s_y and s_z have agreed on offering their service offers y and z as a bundle yz. As it is assumed that it is not possible to realize a cost reduction following a bundling strategy, internal costs for offering the single services add up to 0.4 for service offer yz.

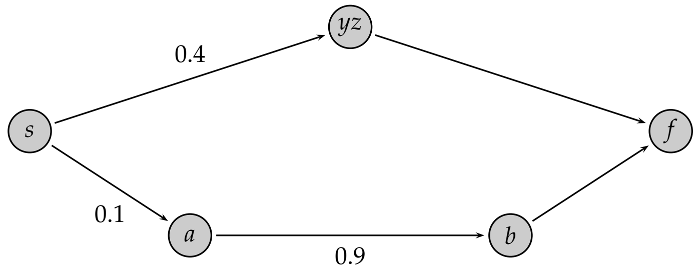


Figure 6.7: Beneficial bundling strategy for allocated and not allocated service providers (ex-post case).

According to the CSA mechanism, the path $f^* = \{e_{syz}, e_{zf}\}$ is allocated which results in a price of 0.4 whereas the other path $f^2 = \{e_{sa}, e_{ab}, e_{bf}\}$ yields a price of 1.0. It is assumed that service providers s_y and s_z divide their payoff according to their contribution to the alliance which means the ratio of their internal costs determines their share. Consequently payments to service providers evolve es follows: $t^{s_y} = \frac{3}{4}(0.4 + (1.0 - 0.4)) = 0.75$ and $t^{s_z} = \frac{1}{4}(0.4 + (1.0 - 0.4)) = 0.25$.

The example at hand shows that although if there is no cost reduction due to synergy effects when following a bundling strategy it might be beneficial for allocated and non-allocated service providers to jointly offer a bundled solution. In this scenario the effect of reducing the network's density (meaning cutting edges by merging service offerings) also affects the number of feasible complex service instances and the composition outcome.

Both fundamental strategies imply advantageous and disadvantageous effects and it is unclear which effect dominates: lower costs to increase the likelihood of being part of the allocation by offering bundled services at a lower price but at the same time a decrease in interoperability which reduces the number of possible service combinations that entail the bundled offer, and thus, reducing the likelihood to be part of the allocation. In contrary an unbundling strategy increase differentiation and specialization but disables opportunities to realize synergy effects. It is proposed that the question whether or not bundling or unbundling is the better strategy to follow depends on the service provider's individual strategic strength. Thus, it is distinguished in service providers that are part of the allocation and those which are not. The following hypotheses are derived:

Hypothesis 6.4. *Service offers which are not allocated have a higher likelihood of being allocated by choosing a bundling strategy instead of an unbundling strategy.*

Hypothesis 6.5. *For service offers which are not allocated, a bundling strategy leads to a higher expected payoff than an unbundling strategy.*

Hypothesis 6.6. *Allocated service offers have a higher likelihood of staying allocated by following an unbundling strategy instead of a bundling strategy.*

Hypothesis 6.7. *For service offers that are allocated, an unbundling strategy leads to a higher payoff than following a bundling strategy.*

The terms *likelihood* or *probability* and *expected payoff* are used with respect to the limited set of observations. Therefore the likelihood or probability of an event refers to the relative frequency of the occurrences of that particular event. Analogously, the term expected payoff refers to the relative frequency times the mean payment observed.

6.3.1 Simulation Model

The stated hypotheses are studied following a simulation approach. The problem is modeled as an n-person game in which each node represents a service offer. Without loss of generality it is assumed that service providers only own a single service offer within the network. Each service offer is characterized by an attribute value for the types *encryption* and *response time*. Dependent on the network topology each service provider faces the decision of choosing an action k which is either to offer a service on its own, i.e. an *unbundling strategy* which is denoted by $k = \mathfrak{u}$, or to form a bundled offer with one of its successors, i.e. a *bundling strategy* which is denoted by $k = \mathfrak{b}$. Thus, in each simulation round $r \in \mathcal{R}$ each node $i \in V \setminus \{v_s, v_f\}$ has to choose an action $k \in \mathcal{K}_i$. The service provider's utility u_{ik} resulting from the action chosen is dependent on the topology of the network, the service requester's scoring function, and all other service offers within the network including their quality and price. For each topology all these factors are stochastic. As such, the node's action decision does not solely control the payoff. Thus, the decision problem of the nodes is comparable to an *n-armed bandit problem*. Since reinforcement learning has proven to cope with such a model-free situation, a simple form of a reinforcement learning algorithm is applied in the present approach. Each node i assigns a fitness value $\mathfrak{f}^r_{ik}$ to each possible action $k \in \mathcal{K}_i$. The fitness of the chosen action k is updated at the end of the period according to the update rule with the learning rate $\beta \in [0;1]$.:

$$\mathfrak{f}_{ik}^{r} = \beta \mathfrak{f}_{ik}^{r-1} + (1 - \beta) u_{ik}^{r} \tag{6.3}$$

Actions are chosen according to a probability choice rule based on each fitness propensity.

$$P_{ik}^{r} = \frac{\mathfrak{f}_{ik}^{r}}{\sum_k \mathfrak{f}_{ik}^{r}} \tag{6.4}$$

The action's propensity is calculated as its fitness weighted by the sum of all fitness values corresponding to the node's actions.

Analogue to the simulation model in Section 6.2.1, the conduction of the simulation is divided in two phases: an *exploration phase* and a *simultaneous exploitation phase*. Figure 6.8 displays the simulation phases and the steps of each phase. Each phase consists of a certain number of rounds $r \in \mathcal{R}$. Each round in the single exploration phase consists of 3 steps. In the first step a single node i chooses an action k with propensity P_{ik}^{r} out of its action set. In the second step, the allocation is computed as well as the mean payoffs for all allocated nodes based on all requester types (different requester types are explained in detail in Section 6.3.2). It is important to notice that, depending on the requesters' scoring functions, allocated service offers and corresponding payoffs differ. In the third step, the fitness value of the chosen action is updated based on the mean payoff computed based on all requester types.

After having trained all nodes, the simultaneous exploitation phase starts in order to evaluate settings with simultaneous decisions. Analogue to the exploration phase, each round of the simultaneous exploitation phase runs through three steps. In the first step, all nodes simultaneously choose a strategy based on P_{ik}. Note, that in the training phase it is just one node choosing the strategy. Only if bilateral bundling decisions match, service offers are merged to a single node forming a bundled offer. The allocation and the mean payoffs based on all requester types are computed in the second step. Each service provider is assigned a numerical value indicating its market power within the service value network. In case two service offers are merged to a bundled offer which is allocated, resulting payoff is distributed based on the market power ratio of both service providers. The last step is again to update the fitness values of all nodes based on the mean payoff.

The data of the simultaneous exploitation phase is analyzed with respect to every possible event that may occur during the conduction of the complex service auction. Table 6.7 shows each possible event that is analyzed with respect to its relative

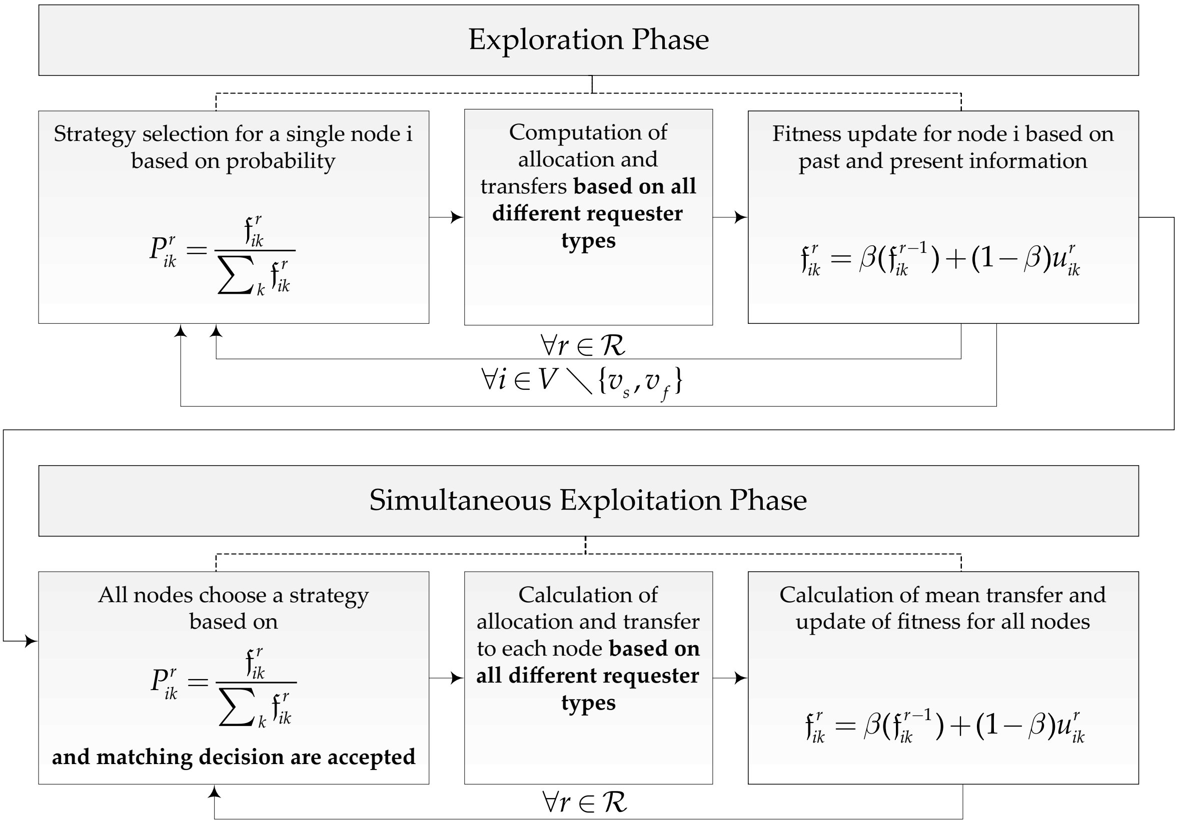


Figure 6.8: Simulation model for the evaluation of bundling and unbundling strategies of service providers.

frequency of occurrence (which can be interpreted as the likelihood of the event's realization) and its expected payoff, i.e. the corresponding mean payoffs received times the event's likelihood of occurrence.

Table 6.7: Analyzed events for the evaluation of bundling and unbundling strategies of service providers with respect to their relative frequency of occurrence and the corresponding expected payoffs. The set $\tilde{E}^s$ denotes the set of edges with $\tilde{E}^s = \{e_{ij} | e_{ij} \in o, j \in \sigma(s), i \in \tau(j)\}$, i.e. the set of allocated edges that belong to service provider s's service offers.

Metric	$\tilde{E}_t$	$\tilde{E}_{t+1}$	$k = \mathfrak{b}$	$k = \mathfrak{u}$
Relative	$\tilde{E}_t \neq \emptyset$	$\tilde{E}_{t+1} \neq \emptyset$	$P(\tilde{E}_{t+1} \neq \emptyset \| k = \mathfrak{b}, \tilde{E}_t \neq \emptyset)$	$P(\tilde{E}_{t+1} \neq \emptyset \| k = \mathfrak{u}, \tilde{E}_t \neq \emptyset)$
		$\tilde{E}_{t+1} = \emptyset$	$P(\tilde{E}_{t+1} = \emptyset \| k = \mathfrak{b}, \tilde{E}_t \neq \emptyset)$	$P(\tilde{E}_{t+1} = \emptyset \| k = \mathfrak{u}, \tilde{E}_t \neq \emptyset)$
Frequency	$\tilde{E}_t = \emptyset$	$\tilde{E}_{t+1} \neq \emptyset$	$P(\tilde{E}_{t+1} \neq \emptyset \| k = \mathfrak{b}, \tilde{E}_t = \emptyset)$	$P(\tilde{E}_{t+1} \neq \emptyset \| k = \mathfrak{u}, \tilde{E}_t = \emptyset)$
		$\tilde{E}_{t+1} = \emptyset$	$P(\tilde{E}_{t+1} = \emptyset \| k = \mathfrak{b}, \tilde{E}_t = \emptyset)$	$P(\tilde{E}_{t+1} = \emptyset \| k = \mathfrak{u}, \tilde{E}_t = \emptyset)$
Expected	$\tilde{E}_t \neq \emptyset$	$\tilde{E}_{t+1} \neq \emptyset$	$E(\tilde{E}_{t+1} \neq \emptyset \| k = \mathfrak{b}, \tilde{E}_t \neq \emptyset)$	$E(\tilde{E}_{t+1} \neq \emptyset \| k = \mathfrak{u}, \tilde{E}_t \neq \emptyset)$
		$\tilde{E}_{t+1} = \emptyset$	$E(\tilde{E}_{t+1} = \emptyset \| k = \mathfrak{b}, \tilde{E}_t \neq \emptyset)$	$E(\tilde{E}_{t+1} = \emptyset \| k = \mathfrak{u}, \tilde{E}_t \neq \emptyset)$
Payoff	$\tilde{E}_t = \emptyset$	$\tilde{E}_{t+1} \neq \emptyset$	$E(\tilde{E}_{t+1} \neq \emptyset \| k = \mathfrak{b}, \tilde{E}_t = \emptyset)$	$E(\tilde{E}_{t+1} \neq \emptyset \| k = \mathfrak{u}, \tilde{E}_t = \emptyset)$
		$\tilde{E}_{t+1} = \emptyset$	$E(\tilde{E}_{t+1} = \emptyset \| k = \mathfrak{b}, \tilde{E}_t = \emptyset)$	$E(\tilde{E}_{t+1} = \emptyset \| k = \mathfrak{u}, \tilde{E}_t = \emptyset)$

The stated hypothesis are tested using a Wilcoxon signed-rank test as the number of observations is relatively small ($N = 30$) and the data is not normally distributed which was tested by means of a Kolmogorov-Smirnov test. The data is based on the mean relative frequencies of each event and corresponding expected payoffs over all service providers.

6.3.2 Simulation Settings

As introduced in Section 6.3 there are two fundamental strategic alternatives service providers have to face: Focusing on differentiation and the provision of flexible service offers that are of highly specialized by following an *unbundling strategy* or focusing on cost reduction due to synergy effects in order to compete in price by following a *bundling strategy*.

To evaluate the success of both strategies and how advantageous and disadvantageous effects of both strategies dominate under which conditions, five different representative types of services requesters are simulated that have different preferences over different QoS attributes and prices of the complex service. Each of these five standard subjects represents a homogenous group of requesters[9].

Table 6.8: Simulation settings for the evaluation of bundling and unbundling strategies of service providers.

Parameter		**Value**
Exploration phase	# rounds	500
Exploitation phase	# rounds	500
Learning rate β		0.1
Service offers	#	varied
	Response time (a_j^{rt})	$\in U(0,1.0)$
	Encryption (a_j^{et})	$\in \{0,1\}$
	Costs (c_{ij})	$\in U(0,1.0)$
	Market power mp	$\in U(0,1.0)$
Service requesters	#	5
	α	$\frac{1}{2}K$
	Type A	$\lambda_{rt} = 0.3, \lambda_{et} = 0.7$
	Type B	$\lambda_{rt} = 0.4, \lambda_{et} = 0.6$
	Type C	$\lambda_{rt} = 0.5, \lambda_{et} = 0.5$
	Type D	$\lambda_{rt} = 0.6, \lambda_{et} = 0.4$
	Type E	$\lambda_{rt} = 0.7, \lambda_{et} = 0.3$

As the results are dependent on the level of competition, multiple scenarios with different numbers of service offers and candidate pools are evaluated. Each scenario has been evaluated with 30 different problems sets, i.e. 30 randomly generated

[9] An alternative approach is the simulation of service requesters with randomly chosen preferences. Nevertheless, this results in heavy statistical noise and hinders the convergence of service providers' fitness in an appropriate number of exploration and exploitation rounds.

topologies based on the parameters outlines in Table 6.8. The exploration phase as well as the simultaneous exploitation phase are conducted 500 times[10].

Each service offer is characterized by attribute values for the types response time and encryption. Attribute values for the type response time are uniformly distributed over the interval [0,0.1]. Encryption values are also randomly chosen and can be either `FALSE` or `TRUE` indicated by 0 and 1. Internal costs of service offers on each incoming edge are drawn from a uniform distribution over the interval [0,0.1].

6.3.3 Results & Implications

For the assessment two different situations for a service provider's service offer are distinguished: it either is part of the allocation or it is not for the case that the service is solely offered. In both cases, the service provider can decide on the $\mathfrak{u}$ or the $\mathfrak{b}$ strategy which can result in either allocation or non allocation. As such, there are eight possible results. The probability of ending up in either of these states is the conditional probability of the described preconditions. These conditional probabilities are derived through the mean relative frequencies (over all service providers) of each event within the simulation. Table 6.7 displays the possible states, the conditional probabilities of these states as well as the expected payoff in these states.

As the number of effects is manifold, the analysis of protruding observations, their interpretation, and implications are structured as follows:

- Analysis within a single competition and cost reduction scenario
- Analysis across different levels of cost reduction and competition
- Bird's eye analysis regarding the overall provider surplus

Analysis within a single competition and cost reduction scenario – Focusing on a single competition and cost reduction scenario, Table 6.9 shows the results in a setting with 20 service offers in 4 candidate pools with no cost reduction due to synergy effects.

The results show that service offers which are not allocated have a significantly higher likelihood of being allocated by choosing a bundling strategy instead of an unbundling strategy which supports Hypothesis 6.4. Also with respect to expected payoffs, for service offers which are not allocated, a bundling strategy leads to a significantly higher expected payoff than an unbundling strategy which supports Hypothesis 6.5. The fact, that these service offers are not allocated initially indicates

[10] A sensitivity analysis has shown that after 500 rounds with a learning rate of $\beta = 0.1$, which avoids stagnation in local optima, the agents' fitness converges to a single best action.

Table 6.9: Evaluation of bundling and unbundling strategies of service providers with 20 service offers in 4 candidate pools and 0% cost reduction due to synergy effects. Relative frequency of possible events and corresponding expected payoffs of service providers are analyzed in $t+1$ for bundling and unbundling strategies depending on the allocation in t. * denotes significance at the level of $p = 0.1$, ** at $p = 0.05$, and *** at $p = 0.01$. Only tested results that correspond to stated hypothesis are indicated.

Metric	$\tilde{E}_t$	$\tilde{E}_{t+1}$	$k = \mathfrak{b}$	$k = \mathfrak{u}$
Relative Frequency	$\tilde{E}_t \neq \emptyset$	$\tilde{E}_{t+1} \neq \emptyset$	0.4707	0.7269***
		$\tilde{E}_{t+1} = \emptyset$	0.5293	0.2730
	$\tilde{E}_t = \emptyset$	$\tilde{E}_{t+1} \neq \emptyset$	0.1904***	0.0355
		$\tilde{E}_{t+1} = \emptyset$	0.8095	0.9645
Expected Payoff	$\tilde{E}_t \neq \emptyset$	$\tilde{E}_{t+1} \neq \emptyset$	0.2834	0.4013***
		$\tilde{E}_{t+1} = \emptyset$	0.0000	0.0000
	$\tilde{E}_t = \emptyset$	$\tilde{E}_{t+1} \neq \emptyset$	0.1009***	0.0193
		$\tilde{E}_{t+1} = \emptyset$	0.0000	0.0000

that they are either not pricewise competitive or that their QoS characteristics are not sufficiently valuable for the service requesters (or both). Thus, by combining their offers with more attractive components – although bearing the loss of interoperability as edges to adjacent service offers are pruned – less competitive service providers increase their chance of being allocated and manage to increase their payoff at the same time (cp. Hypothesis 6.4 and 6.5).

Service providers that are initially capable of competing successfully within the service value networks, i.e. their unbundled service offers are pricewise attractive and expose valuable characteristics for the requesters, have a higher chance of staying allocated by following an unbundling strategy instead of a bundling strategy. Thus, Hypothesis 6.6 is supported. Also with respect to the expected payoff, an unbundling strategy is beneficial for allocated service providers and outperforms a bundling strategy significantly which supports Hypothesis 6.7.

Summarizing the results, Figure 6.9 shows the corresponding decision tree for service providers participating in the complex service auction with respect to bundling and unbundling strategies in a setting with a low level of competition and no cost reduction due to bundling synergies.

Analysis across different levels of cost reduction and competition – On average, the results show that *cost reduction due to synergy effects realized through a bundling strategy increase the likelihood of being allocated in more competitive scenarios*. This effect is not observable in a setting with 20 service offers in 4 candidate pools as the relatively low level of competition requires a tremendous cost reduction to outperform other substitute service offers (cp. Table 6.9 and Table 6.10).

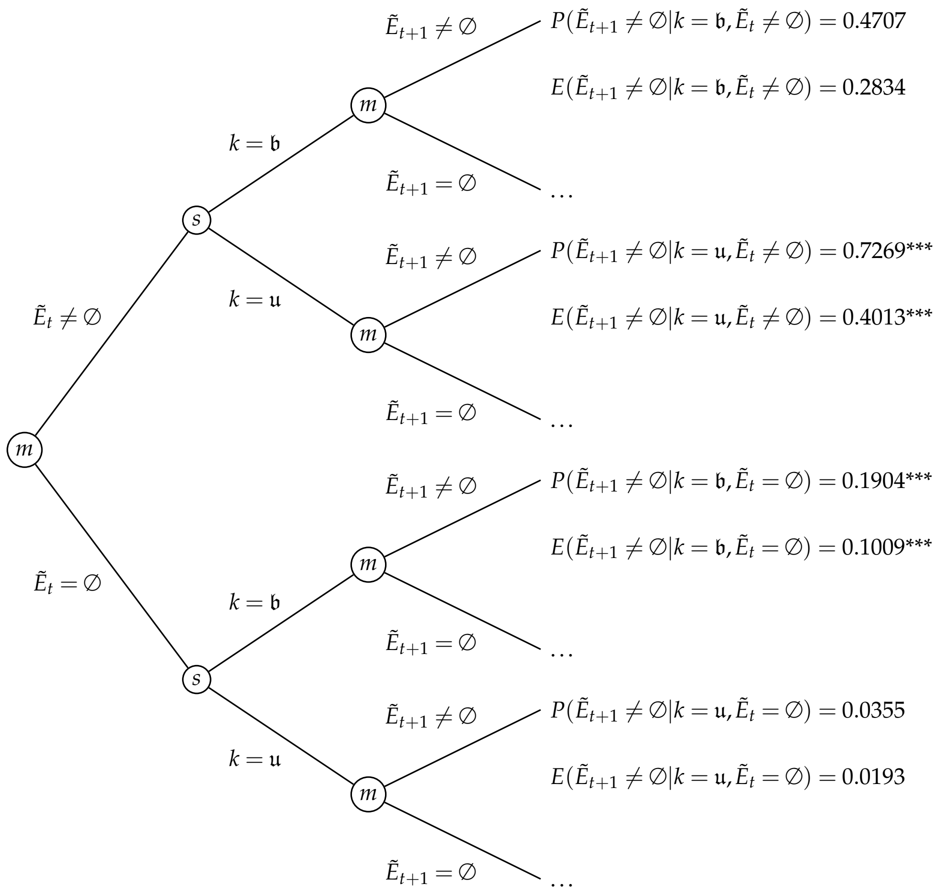


Figure 6.9: Relative frequencies and expected payoffs of bundling and unbundling strategies with 20 service offers in 4 candidate pools and no cost reduction due to synergy effects. Nodes indicated by m denote a decision triggered by the mechanism and s a decision by the service provider.

Table 6.10: Evaluation of bundling and unbundling strategies of service providers with 20 service offers in 4 candidate pools and 50% cost reduction due to synergy effects. Relative frequency of possible events and corresponding expected payoffs of service providers are analyzed in $t+1$ for bundling and unbundling strategies depending on the allocation in t. * denotes significance at the level of $p = 0.1$, ** at $p = 0.05$, and *** at $p = 0.01$. Only tested results that correspond to stated hypothesis are indicated.

Metric	$\tilde{E}_t$	$\tilde{E}_{t+1}$	$k = \mathfrak{b}$	$k = \mathfrak{u}$
Relative	$\tilde{E}_t \neq \emptyset$	$\tilde{E}_{t+1} \neq \emptyset$	0.5035	0.7068***
		$\tilde{E}_{t+1} = \emptyset$	0.4965	0.2931
Frequency	$\tilde{E}_t = \emptyset$	$\tilde{E}_{t+1} \neq \emptyset$	0.1851***	0.0328
		$\tilde{E}_{t+1} = \emptyset$	0.8148	0.9672
Expected	$\tilde{E}_t \neq \emptyset$	$\tilde{E}_{t+1} \neq \emptyset$	0.2519	0.3940***
		$\tilde{E}_{t+1} = \emptyset$	0.0000	0.0000
Payoff	$\tilde{E}_t = \emptyset$	$\tilde{E}_{t+1} \neq \emptyset$	0.0698***	0.0157
		$\tilde{E}_{t+1} = \emptyset$	0.0000	0.0000

In other words, the spread between dominant and dominated service providers is larger in settings with a low level of competition which makes efforts to increase a service offer's attractiveness harder than in high competition settings. In settings with an increased level of competition (e.g. 28 service offers in 4 candidate pools) the effect is significantly observable as a cost reduction of 50% is sufficient to make previously dominated service providers pricewise attractive for the requesters as bundled offers. For a comparison of the results, Table 6.11 shows a setting with an increased level of competition and no cost reduction whereas Table 6.12 shows results assuming a 50% cost reduction for a bundling strategy.

Table 6.11: Evaluation of bundling and unbundling strategies of service providers with 28 service offers in 4 candidate pools and 0% cost reduction due to synergy effects. Relative frequency of possible events and corresponding expected payoffs of service providers are analyzed in $t+1$ for bundling and unbundling strategies depending on the allocation in t. * denotes significance at the level of $p = 0.1$, ** at $p = 0.05$, and *** at $p = 0.01$. Only tested results that correspond to stated hypothesis are indicated.

Metric	$\tilde{E}_t$	$\tilde{E}_{t+1}$	$k = \mathfrak{b}$	$k = \mathfrak{u}$
Relative	$\tilde{E}_t \neq \emptyset$	$\tilde{E}_{t+1} \neq \emptyset$	0.3947	0.9398***
		$\tilde{E}_{t+1} = \emptyset$	0.6053	0.0601
Frequency	$\tilde{E}_t = \emptyset$	$\tilde{E}_{t+1} \neq \emptyset$	0.0502**	0.0129
		$\tilde{E}_{t+1} = \emptyset$	0.9497	0.9871
Expected	$\tilde{E}_t \neq \emptyset$	$\tilde{E}_{t+1} \neq \emptyset$	0.1553	0.4248***
		$\tilde{E}_{t+1} = \emptyset$	0.0000	0.0000
Payoff	$\tilde{E}_t = \emptyset$	$\tilde{E}_{t+1} \neq \emptyset$	0.0199	0.0052
		$\tilde{E}_{t+1} = \emptyset$	0.0000	0.0000

Table 6.12: Evaluation of bundling and unbundling strategies of service providers with 28 service offers in 4 candidate pools and 50% cost reduction due to synergy effects. Relative frequency of possible events and corresponding expected payoffs of service providers are analyzed in $t+1$ for bundling and unbundling strategies depending on the allocation in t. * denotes significance at the level of $p = 0.1$, ** at $p = 0.05$, and *** at $p = 0.01$. Only tested results that correspond to stated hypothesis are indicated.

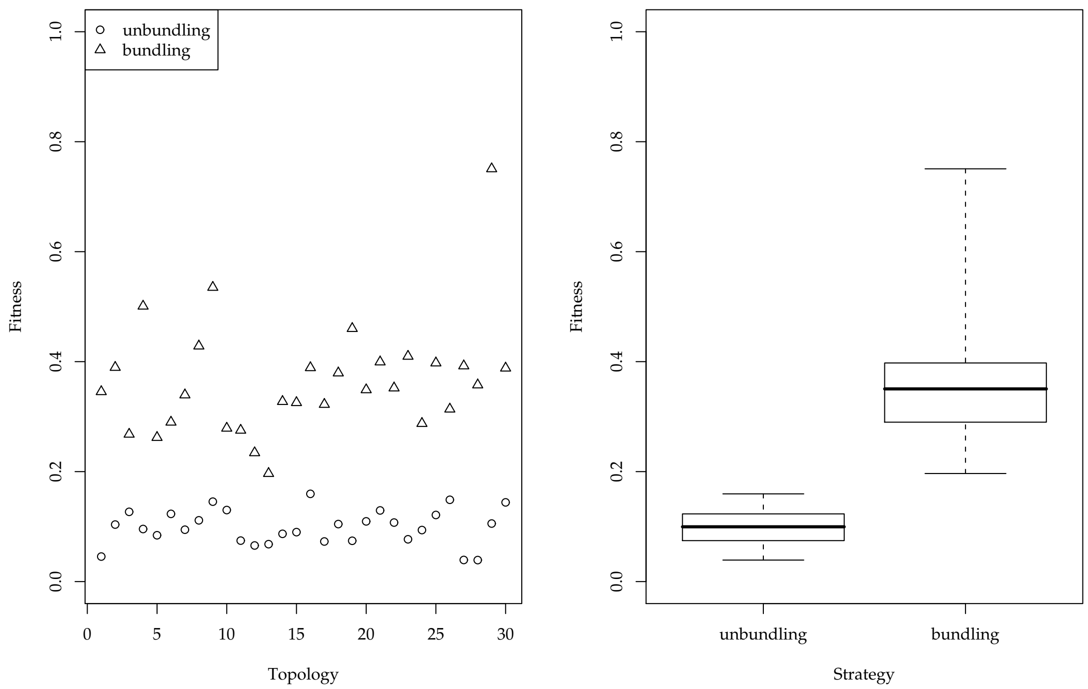
Metric	$\tilde{E}_t$	$\tilde{E}_{t+1}$	$k = \mathfrak{b}$	$k = \mathfrak{u}$
Relative	$\tilde{E}_t \neq \emptyset$	$\tilde{E}_{t+1} \neq \emptyset$	0.4396	0.9275***
		$\tilde{E}_{t+1} = \emptyset$	0.5604	0.0725
Frequency	$\tilde{E}_t = \emptyset$	$\tilde{E}_{t+1} \neq \emptyset$	0.1127***	0.0128
		$\tilde{E}_{t+1} = \emptyset$	0.8872	0.9872
Expected	$\tilde{E}_t \neq \emptyset$	$\tilde{E}_{t+1} \neq \emptyset$	0.1274	0.4556***
		$\tilde{E}_{t+1} = \emptyset$	0.0000	0.0000
Payoff	$\tilde{E}_t = \emptyset$	$\tilde{E}_{t+1} \neq \emptyset$	0.0509***	0.0040
		$\tilde{E}_{t+1} = \emptyset$	0.0000	0.0000

As shown in Theorem 5.2 it is a weakly dominant strategy for service providers to bid truthfully which implies that reducing costs results in a reduced price which service providers charge for their offerings. Nevertheless, Corollary 5.2 shows that in case of being part of the allocation, the service providers' payoff is independent of their bids which means that in contrary to an increased likelihood to become allocated, a cost reduction does not influence the agents payoff.

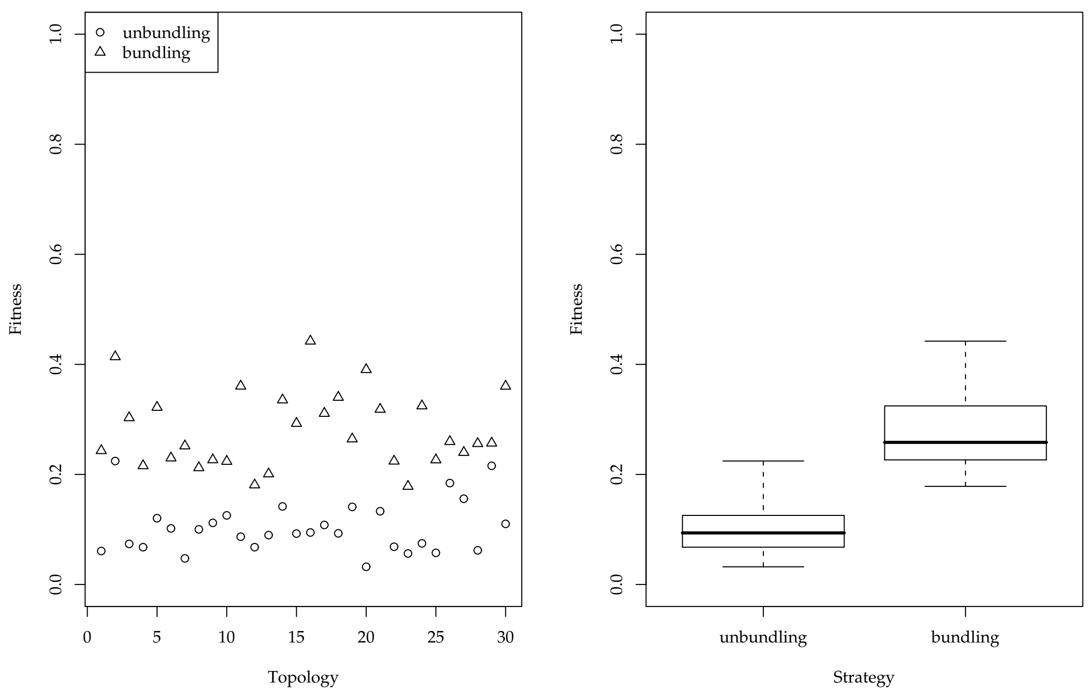
In contrary to e.g. a setting with 20 service offers in 4 candidate pools and no cost reduction, *Hypothesis 6.5 is not supported in settings with a high level of competition and no cost reduction* as illustrated in Table 6.11. With an increase of the number of service offers, interrelations and feasible complex services, a bundling strategy results in a tremendous loss of interoperability. The more preceeding and succeeding service offers and the higher the number of interrelations between services, the higher the loss of interoperability incurred through a merge of single offers within a service value network. In the setting with 28 service offers in 4 candidate pools and no cost reduction for bundled services, the likelihood to get allocated is still higher when following a bundling strategy (supported at a significance level of $p = 0.05$). Nevertheless, the expected payoff that results from that strategy is not significantly better than for the case of unbundling. Thus, *in case the service providers' services are not allocated solely given a high level of competition and given there are no synergy effects that reduce costs for bundled offers, they are indifferent between a bundling and an unbundling strategy.* As a result of the higher level of competition, critical values for service providers are generally lower and especially in the case of bundling, both service providers have to share their payoff according to their market power which again decreases payments in case of getting allocated.

Bird's eye analysis regarding the overall provider surplus – Recall, in the simulation model, service providers maintain a fitness table for each bundling and unbundling strategy. Fitness values indicate the "successfulness" of feasible strategies based on the payoff received when choosing a particular strategy (e.g. higher fitness values indicate beneficial strategies). Thus, fitness values for each strategy are closely related to the payments gained as a feedback to the actions triggered by service providers. Mean fitness values over all service providers for each problem set are depicted in Figure 6.10 and Figure 6.11 in scenarios with different levels of competition and different levels of cost reduction.

In general, bundling strategies seem to outperform unbundling strategies regarding their fitness values. Nevertheless, this is only true for the collectivity of service providers. It is important to notice that there are less allocated service offers than non-allocated services and service providers that own services within each group valuate each strategy differently. As already shown, following an unbundling strategy is in general not beneficial for providers that offer less competitive services which is true for the majority of participants. Hence, fitness values for an

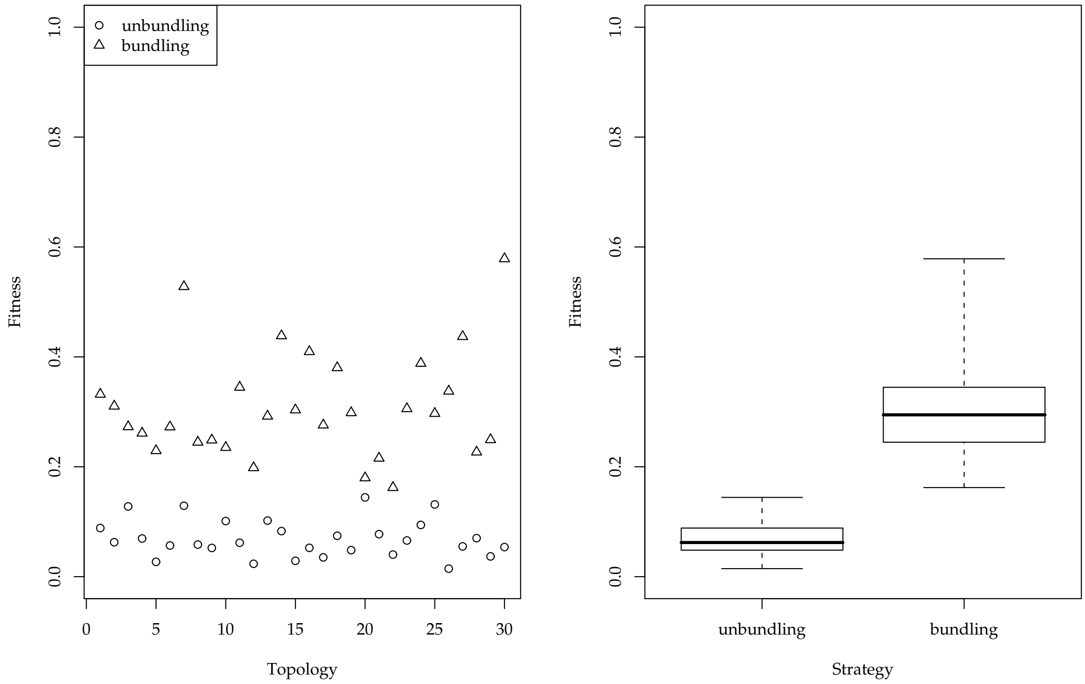


(a) No cost reduction due to bundling synergies with 20 service offers in 4 candidate pools.

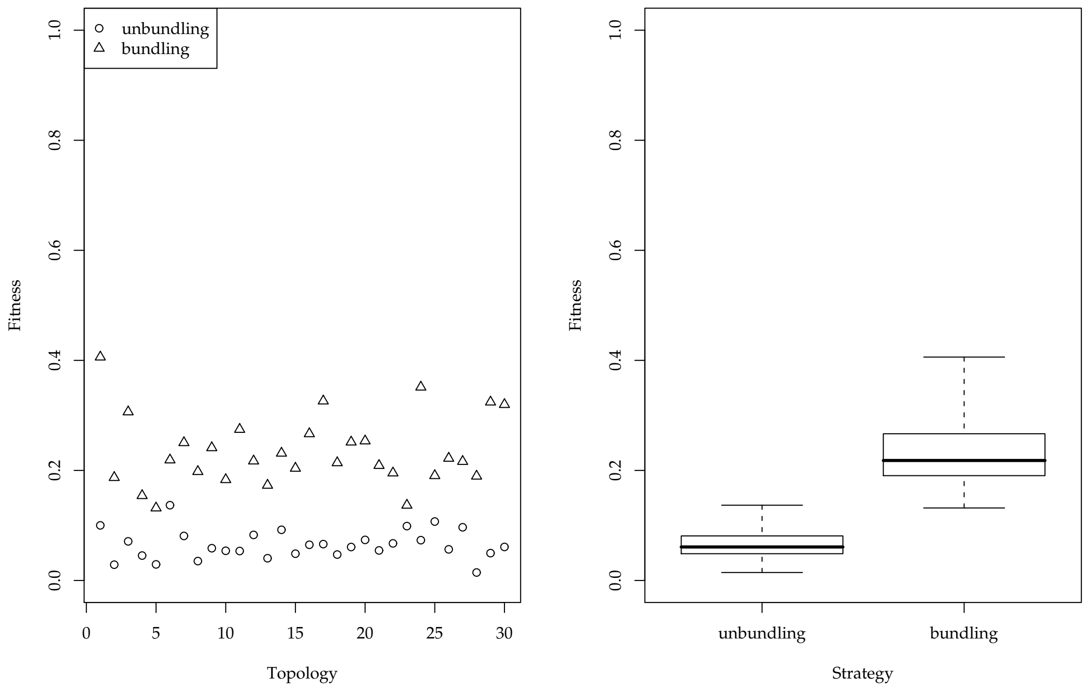


(b) 50% cost reduction due to bundling synergies with 20 service offers in 4 candidate pools.

Figure 6.10: Strategy fitness in different cost reduction scenarios with 20 service offers in 4 candidate pools.



(a) 0% cost reduction due to bundling synergies with 28 service offers in 4 candidate pools.



(b) 50% cost reduction due to bundling synergies with 28 service offers in 4 candidate pools.

Figure 6.11: Strategy fitness in different cost reduction scenarios with 28 service offers in 4 candidate pools.

unbundling strategy for service providers that offer less competitive services are close to zero which in turn strongly decreases the mean fitness for that strategy.

A fundamental effect is observable when comparing scenarios with no cost reduction due to missing synergies as illustrated in Figure 6.10a and with large synergy effects as depicted in Figure 6.10b. The higher the synergy effects realized through bundled offers, the lower the mean fitness value for that strategy. Recall, fitness value are closely related to the payments gained by following a particular strategy. Thus, a decrease in the mean fitness value for the bundling strategy reflects the fact that service providers receive lower payments when realizing synergy effects. Synergy effects reduce costs for service provision. A reduction of costs is directly reflected in the bid prices as shown in Theorem 5.2. Consequently, by simultaneously realizing synergy effects and reducing costs, service providers run into a stronger price competition which is constantly decreasing their payoffs. Looking at service providers as a collectivity, realizing synergy effects by offering bundled solutions decreases the overall provider surplus.

6.3.4 Strategic Recommendations

Based on the results described in Section 6.3.3, the following coarse-grained strategic recommendations regarding single service offers and bundled forms are derived.

For less competitive service offers, a bundling strategy leads to a significantly higher expected payoff than an unbundling strategy and increases the likelihood of being allocated if synergy effects can be realized. Less competitive means that these service offers are either not pricewise competitive or that their QoS characteristics are not sufficiently valuable for the service requesters (or both). Thus, by combining their offers with more attractive components – although bearing the loss of interoperability as edges to adjacent service offers are pruned – less competitive service providers increase their chance of being allocated and manage to increase their payoff at the same time.

Service providers that are initially capable of competing successfully within the service value network have a higher chance of staying allocated and also face a higher expected payoff by following an unbundling strategy instead of a bundling strategy even though synergy effects lie idle. In this case, a loss of interoperability through the merge with another service offer even if compensated by a reduction of costs is not advantageous as it increases the risk of being less favorable from a requester perspective.

Part IV

Finale

Chapter 7

Conclusion & Outlook

This explosion of large-scale e-commerce poses new computational challenges that stem from the need to understand incentives. Because individuals and organizations that own and operate networked computers and systems are autonomous, they will generally act to maximize their own self-interest – a notion that is absent from traditional algorithm design.

[FPP09]

Concluding the work at hand, this chapter points out the key contributions in Section 7.1 followed by an elaboration of open questions and future research directions that are closely related to this work in Section 7.2. Section 7.3 briefly outlines research streams and future challenges that complement the topics addressed in the work at hand.

7.1 Contribution

The key objective of this work is to design a mechanism that enables the coordination of value generation in service value networks which requires that it is on the one hand theoretically sound and on the other hand applicable in the context of electronic services and their composition. It is a well-known result from algorithmic or computational mechanism design [NR01, DJP03] and market engineering [WHN03, Neu04] that these theoretical and practical goals are oftentimes conflicting which requires reasonable solutions regarding these trade-offs to satisfy the requirements upon a suitable mechanism in a certain domain. Addressing these challenges and satisfying detailed requirements derived from a thorough environmental analysis, the work at hand extends the body of research on mechanisms for trading combinatorial entities in distributed environments with special focus on sequential compositions of service components in service value networks. The fact that service compositions only generate value for requesters that expose a feasible order of their service components imposes novel challenges on an adequate coordination mechanism.

A thorough mechanism design requires an in-depth understanding of the economic and technical environment, i.e. the trading objects, the market participants, and the characteristics of the surrounding environment. Hence, the intention of the following research question is to lay the groundwork for the design, implementation and evaluation of an adequate mechanism that enables the trade of composite services in service value networks.

Research Question 1 $\prec$ENVIRONMENTAL ANALYSIS$\succ$. *What are the characteristics of service value networks and complex services, and what are resulting economic and applicability requirements upon a mechanism to coordinate value creation?*

Addressing this question, characteristics and definition of tangibles, intangibles and services are developed and discussed in Section 2.1.1. This discussion is followed by an analysis of different types of services categorized by a service decomposition model in Section 2.1.2. Especially *complex services* constituting the final outcome of the value creation process in service value networks through the realization of a sequence of modularized service offers is in the focus of this analysis. The concept of *traditional services*, *e-services*, *software services*, *Web services* and related technical concepts such as service-oriented architectures are analyzed and their key characteristics are outlined in Section 2.1.3. Based on these results, a clear understanding of *service value networks* is provided in Section 2.1.4 by defining their characteristics, their structure, and their components, and by filling the lack of definitions in current related literature. The discussion about service value networks which embody the trading environment subject to the work at hand is followed by an analysis of economic and applicability requirements upon an adequate mechanism for coordinating value creation in service value networks in Section 2.2.4.1. Based on these requirements, current approaches which are closely related to this work are analyzed and existing research gaps are identified in Section 2.2.4.2. In summary, the environmental analysis and resulting requirement analysis serves as a starting point for further research.

Research Question 2 focuses on the core contribution: The development of an adequate multidimensional and scalable auction mechanism to coordinate value creation in service value networks.

Research Question 2 $\prec$MECHANISM DESIGN$\succ$. *How can a scalable, multidimensional auction mechanism for allocating and pricing of complex services in service value networks be designed that limits strategic behavior of service providers?*

The question is addressed by the development of an abstract *model of service value networks* that captures the key characteristics and components in a comprehensive manner in Section 3.1. As part of the mechanism, a *bidding language* is provided that enables the specification of multidimensional service offers and service requests in Section 3.2. To allow for the expression of the service requester's preferences for different QoS characteristics and prices of complex services, the specification of a *scoring function* is developed. Finally, the core mechanism – *the Complex Service Auction (CSA)* – consisting of an allocation and transfer function which implements valuable properties that are analyzed in detail in the evaluation part, is introduced in Section 3.3. A process model and an adequate architecture of the CSA from a technical perspective are presented in Section 3.5. Focusing on a computational tractable implementation of the auction mechanism, an *algorithm* is presented in Section 3.6 that solves the winner determination problem in polynomial time regarding the number of service offers and feasible service compositions.

Focusing on the applicability of the proposed auction model in real-world scenarios such as a Web-based intermediation service, Research Question 3 states additional requirements and addresses the challenge of developing necessary extensions to the core mechanism in order to be applicable in practical settings.

Research Question 3 $\prec$APPLICABILITY EXTENSIONS$\succ$. *How can an auction mechanism be extended to support complex QoS characteristics and service level enforcement? How can the pricing scheme be modified in order to achieve budget balance and incentivize interoperability endeavors of service providers?*

In order to provide trust and assurance of service quality, service level enforcement is an inevitable applicability aspect. In Section 4.1, the mechanism is enriched by a compensation function which incorporates ex-post information about each service's performance in order to impose penalties if necessary. The compensation function provides valuable economic properties which are analyzed in detail in the evaluation part. Addressing the challenge of supporting complex QoS characteristics, a common conceptualization of quality attributes and their description, aggregation and enforcement from an economic and technical perspective is provided. The auction mechanism is extended in order to *support complex QoS characteristics* by means of rule-based semantic concepts and a toolbox of adequate aggregation operations in Section 4.3.

Another central requirement upon a mechanism from an economic perspective is budget balance which is an important property for a mechanism in order to be sustainable in the long-run as a continuous external subsidization is neither reasonable nor profitable for e.g. a platform provider and its business model. It is well-known from impossibility results in mechanism design that the achievement of certain combinations of economic desiderata is not possible. Addressing the second part of Research Question 3, an extended transfer function – *the Interoperability Transfer Function (ITF)* – is developed in Section 6.2 which *restores budget balance* by sacrificing incentive compatibility to a certain extent and at the same time *incentivizes service providers to increase their services' degree of interoperability*, i.e. to increase the capability of their offered services to communicate and function with other services within the service value network which is shown addressing Question 4.

Research Question 4 $\prec$EVALUATION$\succ$. *How can an auction mechanism be analytically and numerically evaluated regarding its economic properties as well as cooperation and bundling strategies of service providers?*

Focusing on central economic properties of a mechanism and the implemented social choice function, Research Question 4 is firstly addressed in Chapter 5 by an analytical evaluation which shows that the complex service auction implements a social choice function that is *incentive compatible* and *individual rational* for service providers (Section 5.1). The mechanism is strategyproof with respect to all dimensions of service providers' bids, i.e. the truthful announcement of private information on QoS attributes and valuations of offered services is an equilibrium in dominant strategies. Consequently, if the service requester announces its accurate preferences for different outcomes, the social choice is *allocative efficient* as it is shown in Section A.3. Based on a *model of cooperation* provided in Section 5.2, it is further shown that there exist mutually beneficial ex-ante agreements between service providers that face the opportunity to customize their service offers in order to reduce internal costs.

Following a numerical research method in Chapter 6, the extended budget-balanced transfer function ITF is firstly evaluated with respect to its robustness against misreporting of service providers by means of simulation-based analysis in Section 6.1. The question is more precisely: To what degree is it beneficial for service providers to deviate from their true valuation? Results show that even in settings with a low level of competition *strategic behavior of service providers is tremendously limited* as a deviation from a truth-telling strategy is not significantly beneficial. Despite of the incentives that limit service providers' strategic behavior, the ITF rewards service providers to increase their services' degree of interoperability. This property is elaborated in detail in Section 6.2 by means of an agent-based simulation. Compared to an equal transfer function which distributes available surplus equally among allocated service providers, it is shown that the ITF extension implements incentives to *foster a higher overall degree of interoperability* in settings with a low level of competition and up to a certain level of proportionate investment costs for customization.

Focusing on cooperation models in the form of offering bundled services, the question arises whether it is beneficial to offer bundled services which decreases flexibility but leverages synergy effects or if it is beneficial to offer single highly specialized services that are more flexibly composable into various complex service instances. By means of an agent-based simulation with reinforcement learning, this question is addressed in Section 6.3. More precisely there are two main strategies analyzed: Competing in quality through differentiation and flexibility and competing in price through bundling synergies as cost reduction. Results show that in general service providers that own services within the service value network which are highly competitive, i.e. they are likely to be allocated, act best by following an unbundling strategy. In contrary, for service providers with less competitive service offers it is beneficial to form bundled service offers while leveraging synergy effects.

7.2 Open Questions

Based on the above mentioned results, there is a number of possible future research directions and open questions which are briefly addressed in the remainder of this section.

Allocation computation in the context of sophisticated control logic

The allocation function of the complex service auction computes the "shortest" path in graphs and is therefore only capable of allocating rudimentary flow logic in the form of sequential compositions whereas e.g. AND-states have to be split up in separate statecharts and different auction processes. Such an approach is sufficient for the allocation of more granular service components that are iteratively composed into a complex service.

However, more sophisticated flow logic increases the complexity of finding feasible allocations that embody a flawless instantiation of a complex service from a technical perspective. This leads directly to the questions of *how more complex control logic (e.g. AND-states, loops, branches, conditional flows) can be covered by an allocation function?* However, a more complex allocation problem that results from a more powerful control logic of complex services directly leads to an increase of computational complexity with respect to solving the winner determination problem while assuring feasible solutions from a technical perspective. This hinders the satisfaction of Requirement 5 which stresses the importance of computational tractable algorithms to solve the winner determination problem in polynomial time for the application in online systems. Addressing this challenge, heuristics might be a reasonable approach to solve the allocation problem in the context of complex services that expose highly sophisticated control logic. Nevertheless, in the absence of an optimal solution, the central Requirement 1 of allocative efficiency is not fully satisfied depending on the degree of optimality of the heuristic allocation algorithm. In case the mechanism is designed to foster an incentive compatible social choice, a suboptimal solution of the winner determination problem becomes critical from an economic perspective. The heuristic has to satisfy certain properties such as monotonicity – i.e. an allocated participant in the complex service auction cannot drop out of the allocation by decreasing its bid price – in order to retain truthfulness [MN08, NS06].

Allocation and pricing of people services

Hybrid complex services that involve electronic and human activities impose new challenges from an economic and organizational perspective. So far, micro-task markets such as Amazon's Mechanical Turk[1] provide a platform to leverage the power of human intelligence – the so called *crowdsourcing* – for highly specialized tasks such as image recognition. A pool of human individuals encapsulated by well-defined interfaces can be integrated in hybrid processes. A seamless integration of human work force in automated compositions of multiple services opens up further research questions to be addressed in the future. *How can people services sufficiently be described and integrated into service value networks and the coordination of value creation?* The challenges that arise from the service characteristic C 2.5 describing the fuzzyness of input and output parameters and capabilities are partly addressed by the high degree of standardization and specified description languages (e.g. WSDL, WS-BPEL) which are common sense. Nevertheless, in the context of people services, these challenges arise anew as human work force is hardly parameterizable and the scope, capabilities and quality of the output

[1] `http://mturk.com/`

vary widely. Thus, incorporating human activities in automated processes requires well-specified task descriptions [KCS08]. As inputs and outputs have to be carefully described the issue of quality assurance becomes even more crucial. The question arises of *how these activities can be monitored in order to compute compensation transfers and apply service level enforcement mechanisms.*

Allocation and pricing of highly complex application services

As introduced in Section 2.1.4.3, a trend towards simplification is observable that enables an agile composition of highly specialized services that expose puristic interfaces and descriptions e.g. as in RESTful architectures based on the CRUD paradigm[2]. Nevertheless as outlined in Section 2.1.2.3, complex services consist of service components that can themselves be a utility, elementary or complex service (analogue to the recursive specification in WS-BPEL). As the granularity of service components decreases, the complexity of their interfaces and necessary descriptions grows which implies new challenges for the mechanism. As a result of the increased interface complexity and the semantic of input and output values, the computational complexity of the algorithm that solves the respective winner determination problem augments as well. This conflicts with the requirement of computation tractability which is inevitable for a mechanism in order to be realized in online systems. Furthermore, investment costs for the customization of service offers' interfaces fostering a higher degree of interoperability rise which results in more static and less multifaceted service value networks. More complex service descriptions and interfaces also impact the elicitation and expression of preferences for different QoS levels. Service requesters have to specify their preferences for different outcomes regarding the complex service's attributes which leads to the question of *how service consumers can be supported by tools and concepts to enable the elicitation and expression of preferences for complex multidimensional QoS characteristics.*

Multi-layered markets for utility and complex services

Service components that are traded in e.g. the complex service auction require low level resource services (utility services) to enable their deployment and assure scalability during run-time. Focusing on the infrastructure layer, it is also reasonable to trade utility services themselves independent from mechanisms to allocate and price complex services. Nevertheless, utility services expose different characteristics and therefore impose different requirements upon suitable market mechanisms [Neu04]. There are several market mechanisms for the trade of utility services proposed in literature [Sto09, Sch07]. Combining the trade of utility and complex services as depicted in Figure 7.1, the question arises of *how a multi-layered market can*

[2] CRUD stands for the persistent functions create, read, update, and delete.

be designed in order to enable a seamless allocation and pricing of complex services and corresponding utility service which are required by the layer above.

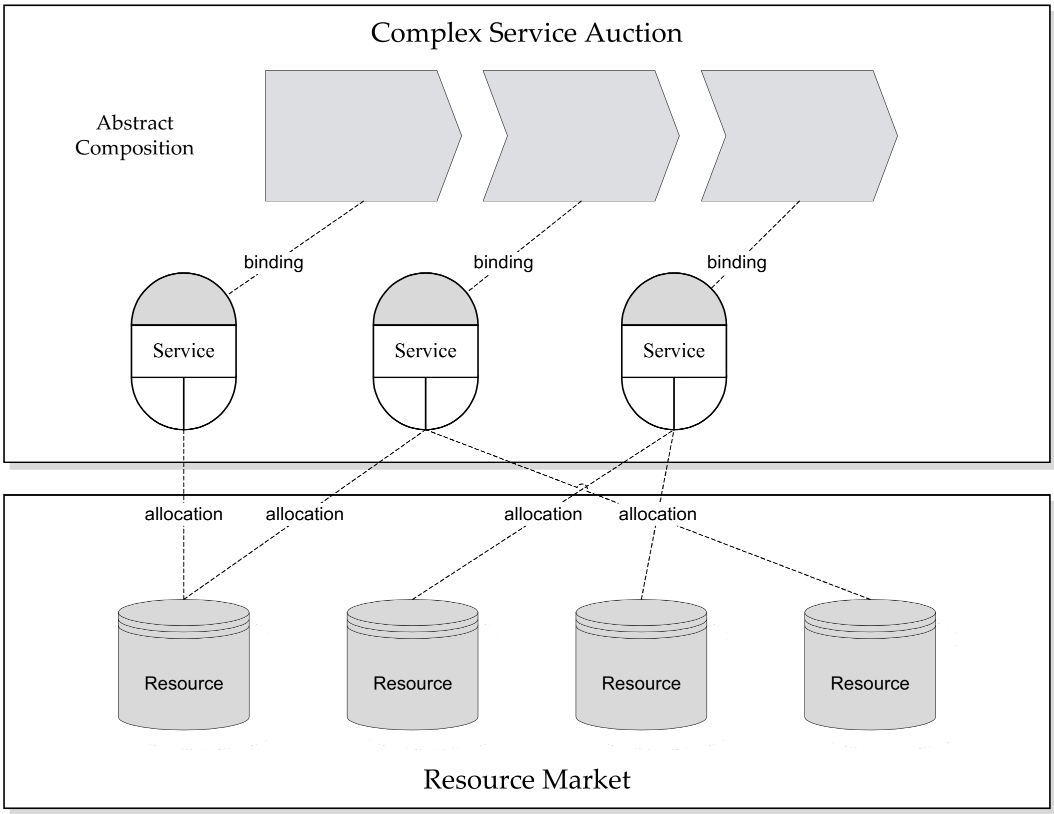


Figure 7.1: Multi-layered market for complex services and resources.

7.3 Complementary Research

Besides research directions closely related to the work at hand as illustrated in Section 7.2, this section points out research questions which are partly complementary to this work and therefore possibly enrich certain aspects.

Alternative design goals and business models for platform providers

The design of the complex service auction mechanisms implements a social choice that is allocative efficient, i.e. it maximizes welfare. Although this is a commonly desired design goal that has valuable implications for all participants, there are alternative design desiderata that are favorable for certain stake holders. From the perspective of a platform provider that offers an intermediation service to e.g. a service value network, a revenue maximizing social choice is certainly beneficial compared to an optimal solution from a utilitarian point of view if e.g. the intermediary receives a fraction of the each service provider's revenue. Research that deals with auction formats which are designed to maximize the revenue for e.g. the seller of an economic entity is well-known in literature as *optimal auction design* [Mye81]. Focusing on procurement scenarios where price and quality matters, optimal buying mechanisms that intent to maximize the buyer's expected payoff are evaluated in [CIoWM93, AC05]. Looking at optimal auction designs and revenue models for platform providers, the question of *how to design a successful business model for providers of intermediation services* arises. The structure of "traditional" business model types might not be sufficient in order to address the requirements that result from highly agile and distributed environments such as service value networks [MWL$^+$06]. Recall that a mechanism in order to be sustainable in the long-run must satisfy the economic design desideratum of budget balance (cp. Desideratum 2.4) in order to avoid the need for external subsidization as well as the desideratum of individual rationality (cp. Desideratum 2.3) to provide incentives to participate in the market. In this regard, revenue models for platform providers that stipulate for charging participation fees may violate individual rationality and (strong) budget balance. However, in certain cases it might be reasonable for a e.g. a public institution to subsidize an efficient market. Nevertheless, such implications of the revenue model on economic properties of a mechanism implementation must be carefully analyzed and considered when constructing and structuring novel business models.

Preference elicitation

It is a typical assumption in game theory and especially mechanism design research that market participants know their true valuations. However, elicitation of preferences especially in multidimensional settings (e.g. preferences for different QoS levels of multiple service attributes and their semantics) embodies a complex task for service providers and requesters. In combinatorial settings (cp. the complex service auction), participants must be capable of expressing preferences for different combinations of e.g. service components. This is a crucial task as it implicitly requires the comparison of a large set of alternative combinations. Although preference elicitation embodies a prerequisite of any market-based approach, research in this area is still in its infancy [SNP$^+$05]. For instance, prominent approaches for the elicitation of preferences – e.g. in the context of services – are conjoint analysis [GR71, LT64] and analytical hierarchical processing [Saa80, Saa08]. A major shortcoming of these approaches is that they become infeasible in settings with large sets of attributes which are common in e.g. service markets.

Automated bidding

Having suitably determined the true valuations for the trading object, a bidding strategy must be developed in order to successfully participate in the market. With preference elicitation as a prerequisite, developing such a bidding strategy and efficiently communicating it to the market is another complex task to be solved by participants. In order to support users in evaluating and expressing a beneficial bidding strategy, tools for automated bidding are a promising approach to overcome complexity and effort [MMW06, Tes01]. Another advantage of facilitating tools to interact with markets is that there is no need to constantly monitor market activities and incorporate information in the bidding strategy as this information can be processed and interpreted by automatic bidding agents. Although these tools can simplify market interaction, participants want to keep control over their strategy and resulting actions. Hence, hybrid models are more practical as they still hide complexity and simplify the trading process but also allow for a manual interaction triggered by the user which might also be necessary for legal reasons. Another success factor of automatic trading agents is the parameter selection and their customization for the application in different market mechanisms that impose different requirements upon beneficial strategies. Addressing these challenges, strategies for bidding agents are developed that successfully perform in multiple settings and market mechanisms [Bor09].

Reputation mechanisms

Another class of mechanisms that enable coordination of distributed activities in a broader sense are reputation mechanisms. Using feedback information, reputation mechanisms aim at building trust in environments with self-interested participants [BKO02]. Reputation mechanisms aggregate trading histories of e.g. service providers and requesters and compute a metric which indicates the trustworthiness of market participants. This information can be incorporated in the allocation and pricing procedure providing additional characteristics of the trading parties. For example, the lower the reputation of a service provider, the less likely is the allocation of services offered by this service provider. Although it is well-known in literature that reputation mechanisms have proven to perform well in distributed systems in the absence of a central instance such as in peer-to-peer networks [WV03], it is an interesting question of how such reputation components can be designed and realized additionally to a central market mechanism. Challenges that arise in this context are e.g. how to make truthful revelation of reputation information an optimal strategy market participants [JF03]. For a detailed survey on state-of-the-art trust and reputation systems for service provision via electronic networks, the interested reader is referred to [JIB07].

7.4 Final Remarks

Services become a central component of value creation in today's society. Novel technical, economic, and organizational challenges arise from their unique nature as services' provision and consumption coincide in time [Hil77]. Recognizing and understanding the importance of an efficient design, production, and provision of services under the presence of their special characteristics is inevitable for individuals and the society to compete in today's global economy. Especially rapid service innovation driven by the power of modularity that is inherent in the concept of services [BC00] embodies the success factor in service-centric environments. However, when composing distributed service activities, the question of an efficient form of coordination comes to light and turns out to be fundamental to govern distributed value creation. As complex services are living artifacts that generally exist under the ownership of different economic entities which are self-interested in nature, system-wide goals are hard to achieve as they mostly collide with individual objectives and are therefore not intrinsically pursued [Par01].

The approach of mechanism design [Hur73, Mye88] – and the revelation principle [Gib73, Mye82] as the central possibility result – considers economic problems in situations where individuals' private information and actions are hard to monitor. The main objective is to design mechanisms that provide incentives for individuals to "share information and exert efforts" [Mye88] which implements a social choice that constitutes a system-wide solution. Hence, although individuals (e.g. service owners) seek to maximize their utility based on their private information about their preferences for different outcomes, they inevitably contribute to the achievement of a global goal.

Following the approach of mechanism design, this work provided an auction mechanism which enables the trade of composite services in service value networks. The mechanism constitutes an equilibrium in which truth-revelation of private multidimensional types is a weakly dominant strategy for all service providers and implements a social choice that maximizes the utility across all participants. The mechanism exposes valuable properties as it is not beneficial for individuals to lie about their private information, neither on their services' QoS characteristics nor on corresponding private valuations. Furthermore, participation is voluntary and beneficial for service providers and the mechanism results in an allocation which is optimal and constitutes a system-wide welfare maximizing solution.

The work at hand shows that mechanism design in combination with technical, computational, and applicability considerations is a promising approach to efficiently govern distributed service activities in agile and fast changing environments such as service value networks. However, open questions and complementary research directions constitute further challenges that need to be mastered in an integrated manner in order to leverage the power of algorithmic mechanism design and to move the results at hand from theory to practice, to innovation.

Appendix A

Appendix

A.1 Formal Notation

Table A.1: Notation of abstract model and mechanism implementation.

Notation	Meaning
$G = (V, E)$	Service Value Network
$V \setminus \{v_s, v_f\} = \{v_1, \ldots, v_N\}$	N Service offers/services/nodes with $i, j \in V$ are arbitrary services
$v_s, v_f \in V$	Source and sink node
$E = \{e_{ij} \mid i, j \in V\}$	Technical feasible combinations of services
$f \in F$	Feasible path from source to sink that is an instantiation of a complex service f
$S = \{s_1, \ldots, s_Q\}$	Q Service providers
$\sigma : S \to V$	Ownership function
$A_j = \{a_j^1, \ldots, a_j^L\}$	Configuration of service j with a_j^l is the attribute value of type $l \in L$
c_{ij}	Interoperability costs of service j as a successor of service i
$\mathcal{A}_f = (\mathcal{A}_f^1, \ldots, \mathcal{A}_f^L)$	Configuration of complex service f with $\mathcal{A}_f^l$ is the attribute value of type $l \in L$
$\mathcal{S} : \mathcal{A} \to [0;1]$	Scoring function of service requester
$\Lambda = (\lambda_1, \ldots, \lambda_L)$	Preference structure of service requester with λ_l is the weight for attribute type $l \in L$
$\Gamma = (\gamma_B^1, \gamma_T^1, \ldots, \gamma_B^L, \gamma_T^L)$	Preference boundaries of service requester with γ_B^l is the lower and γ_T^l is the upper boundary for attribute type $l \in L$

Table A.1: Notation of abstract model and mechanism implementation.

Notation	Meaning
α	Willingness to pay of service requester for a complex service f with $\mathcal{S}(\mathcal{A}_f) = 1$

A.2 Incentive Compatibility

Proof A.1 [THEOREM 5.2]. [1] *Let F_{-s} denotes the set of all feasible paths from source to sink in the reduced graph G_{-s} without every service offer owned by service provider s and corresponding incoming and outgoing edges. Let further f^* denote the path which is allocated by o. Let $\mathcal{U}^*$ be the utility of path f^*. Let $\mathcal{U}^*_{-s}$ be the utility of path f^*_{-s} in the reduced graph G_{-s}. Let $\tilde{\mathcal{U}}^{*s}$ denote the overall utility of the allocated path f^* computed based on the verified attribute values $\tilde{a}^1_j, \ldots, \tilde{a}^L_j$ of the verified configurations $\tilde{A}_j$ of all service offers $j \in \sigma(s)$. Let $\tilde{E}^s$ denote the set of edges with $\tilde{E}^s = \{e_{ij} | e_{ij} \in o, j \in \sigma(s), i \in \tau(j)\}$. Service provider s wants to maximize its expected payoff:*

$$
\begin{aligned}
E(\pi^s) &= P(\mathcal{U}^* > \mathcal{U}^*_{-s}) \left[\sum_{\tilde{E}^s} p_{ij} + (\mathcal{U}^* - \mathcal{U}^*_{-s}) - \Delta t^{comp,s} - \sum_{\tilde{E}^s} c_{ij} \right] \\
E(\pi^s) &= P(\mathcal{U}^* > \mathcal{U}^*_{-s}) \left[\sum_{\tilde{E}^s} p_{ij} + ((\mathcal{U}^* - \mathcal{U}^*_{-s}) - (\mathcal{U}^* - \tilde{\mathcal{U}}^{*s})) - \sum_{\tilde{E}^s} c_{ij} \right] \\
E(\pi^s) &= P(\mathcal{U}^* > \mathcal{U}^*_{-s}) \left[\sum_{\tilde{E}^s} p_{ij} + (\tilde{\mathcal{U}}^{*s} - \mathcal{U}^*_{-s}) - \sum_{\tilde{E}^s} c_{ij} \right]
\end{aligned}
$$

This leads to two possible cases:

1. *If s's payoff π^s is positive, it wants to maximize the probability of being allocated which leads to the problem statement*

$$
\max_{p_{ij}, A_j | j \in \sigma(s), i \in \tau(j)} P(\mathcal{U}^* > \mathcal{U}^*_{-s})
$$

$$
st. \left[\sum_{\tilde{E}^s} p_{ij} + (\tilde{\mathcal{U}}^{*s} - \mathcal{U}^*_{-s}) - \sum_{\tilde{E}^s} c_{ij} \right] > 0
$$

From the side condition it follows directly that $\sum_{\tilde{E}^s} p_{ij} + \tilde{\mathcal{U}}^{*s} - \sum_{\tilde{E}^s} c_{ij} > \mathcal{U}^*_{-s}$. Hence, $P(\cdot)$ is maximized by setting $p_{ij} = c_{ij}$ and $A_j = \tilde{A}_j$, $\forall j \in \sigma(s), i \in \tau(j)$ as this leads to $\mathcal{U}^* = \tilde{\mathcal{U}}^{*s}$ and finally to $P(\cdot) = 1$.

[1] This proof is based on the argumentation in [MMV94]

2. *If s's payoff π^s is negative, it wants to minimize the probability of being allocated which leads to the problem statement*

$$\min_{p_{ij},A_j|j\in\sigma(s),i\in\tau(j)} P(\mathcal{U}^* > \mathcal{U}^*_{-s})$$
$$st. \left[\sum_{\tilde{E}^s} p_{ij} + (\tilde{\mathcal{U}}^{*s} - \mathcal{U}^*_{-s}) - \sum_{\tilde{E}^s} c_{ij}\right] < 0$$

Symmetrically to the first case, it follows directly from the side condition that $\sum_{\tilde{E}^s} p_{ij} + \tilde{\mathcal{U}}^{*s} - \sum_{\tilde{E}^s} c_{ij} < \mathcal{U}^*_{-s}$. Hence, $P(\cdot)$ is minimized by setting $p_{ij} = c_{ij}$ and $A_j = \tilde{A}_j$, $\forall j \in \sigma(s), i \in \tau(j)$ as this leads to $\mathcal{U}^* = \tilde{\mathcal{U}}^{*s}$ and finally to $P(\cdot) = 0$.

In any case one solution that maximizes the expected payoff $E(\pi^s)$ of service provider s is $p_{ij} = c_{ij}$ and $A_j = \tilde{A}_j$, $\forall j \in \sigma(s), i \in \tau(j)$. This solution is the truth-telling strategy as s reveals its true multidimensional type. Although truth-telling is not the only solution, service provider s does not benefit from deviation as its strategy does not influence its payoff as shown in Corollary 5.2 which makes truth-telling with respect to the multidimensional types of service providers (configuration and price) a weakly dominant strategy. □

A.3 Allocative Efficiency

This section briefly shows that under the assumption of the absence of strategic behavior of the service requester, the complex service auction always leads to a welfare maximizing outcome:

Corollary A.1 [Welfare Maximization]. *The allocation function according to* (3.8) $\operatorname{argmax}_{f\in F} \left(\alpha\mathcal{S}(\mathcal{A}_f) - \mathcal{P}_f\right)$ *is efficient as it maximizes the system's welfare with α representing the requester's maximal willingness to pay, $\mathcal{S}(\mathcal{A}_f)$ its score for the configuration of the complex service f and $\mathcal{P}_f$ the sum of all price bids of service providers that own service offers that have incoming edges on the path f.*

Proof A.1 [Corollary A.1]. *Let $\mathcal{U}^R = \alpha\mathcal{S}(\mathcal{A}_f) - \mathcal{T}_f$ denote the service requester's utility with α represents the requester's maximal willingness to pay, $\mathcal{S}(\mathcal{A}_f)$ the requester's score for the configuration of the complex service f and $\mathcal{T}_f$ the sum of all transfer payments to allocated providers according to* (4.2). *Furthermore let $\mathcal{U}^s = t^s - c^s$ be the utility of service provider $s \in S$. The system's welfare W_f based on an allocated path f is the sum of consumer (requester) and providers' surplus such that*

$$\begin{aligned} W_f &= \mathcal{U}^R + \sum_{s\in S} \mathcal{U}^s \\ W_f &= \alpha\mathcal{S}(\mathcal{A}_f) - \mathcal{T}_f + \sum_{s\in S} (t^s - c^s) \end{aligned}$$

$$
\begin{aligned}
W_f &= \alpha\mathcal{S}(\mathcal{A}_f) - \mathcal{T}_f + \mathcal{T}_f - \sum_{s\in S} c^s \\
W_f &= \alpha\mathcal{S}(\mathcal{A}_f) - \sum_{s\in S} c^s
\end{aligned}
$$

Based on the results of Theorem 5.2 truth-telling with respect to configuration and price is a weakly dominant strategy for all service providers so it can be directly concluded that

$$W_{f^*} = \alpha\mathcal{S}(\tilde{\mathcal{A}}_{f^*}) - \mathcal{P}_{f^*}$$

□

A.4 Manipulation Robustness

Table A.2: Utility for a single manipulating service provider with 12 service offers in 4 candidate pools. *abs* denotes the mean absolute utility and *rel* the ratio of means of the utility with manipulation and the utility following a truth-telling strategy. *sd* is the standard deviation of the mean absolute utility. * denotes significance at the level of $p = 0.1$, ** at $p = 0.05$, and *** at $p = 0.01$.

	Critical Value Transfer			Interoperability Transfer		
Manipulation Rate	abs	rel	sd	abs	rel	sd
-50%	0.0423	0.5865	0.0793	-0.0209	-0.6871	0.1022
-45%	0.0506	0.7007	0.0634	-0.0113	-0.3802	0.0860
-40%	0.0562	0.7789	0.0506	-0.0009	-0.0308	0.0714
-35%	0.0604	0.8359	0.0413	0.0055	0.1809	0.0596
-30%	0.0631	0.8741	0.0334	0.0113	0.3645	0.0478
-25%	0.0656	0.9092	0.0275	0.0158	0.5254	0.0394
-20%	0.0693	0.9603	0.0136	0.0194	0.6763	0.0264
-15%	0.0702	0.9724	0.0103	0.0235	0.7919	0.0196
-10%	0.0715	0.9904	0.0050	0.0250	0.8795	0.0144
-5%	0.0721	0.9981	0.0015	0.0291	0.9477	0.0066
0%	0.0722	1.0000	0.0000	0.0302	1.0000	0.0000
5%	0.0721	0.9982	0.0012	0.0326	1.0378***	0.0075
10%	0.0715	0.9906	0.0050	0.0317	1.0688***	0.0125
15%	0.0711	0.9847	0.0074	0.0302	1.1036***	0.0148
20%	0.0705	0.9771	0.0097	0.0327	1.0968***	0.0199
25%	0.0704	0.9750	0.0100	0.0365	1.1194***	0.0238

Table A.2: Utility for a single manipulating service provider with 12 service offers in 4 candidate pools. *abs* denotes the mean absolute utility and *rel* the ratio of means of the utility with manipulation and the utility following a truth-telling strategy. *sd* is the standard deviation of the mean absolute utility. * denotes significance at the level of $p = 0.1$, ** at $p = 0.05$, and *** at $p = 0.01$.

	Critical Value Transfer			**Interoperability Transfer**		
Manipulation Rate	abs	rel	sd	abs	rel	sd
30%	0.0703	0.9738	0.0102	0.0393	1.1380***	0.0283
35%	0.0702	0.9721	0.0109	0.0397	1.1700***	0.0328
40%	0.0696	0.9638	0.0137	0.0384	1.1776***	0.0355
45%	0.0690	0.9554	0.0184	0.0422	1.1672***	0.0402
50%	0.0673	0.9320	0.0261	0.0379	1.1774***	0.0435
55%	0.0664	0.9201	0.0304	0.0383	1.1507***	0.0455
60%	0.0640	0.8870	0.0383	0.0384	1.1016***	0.0445
65%	0.0636	0.8806	0.0388	0.0390	1.0768***	0.0480
70%	0.0627	0.8691	0.0424	0.0377	1.0866***	0.0486
75%	0.0605	0.8381	0.0504	0.0364	1.0366**	0.0438
80%	0.0603	0.8354	0.0508	0.0355	1.0535***	0.0449
85%	0.0602	0.8335	0.0511	0.0365	1.0537***	0.0470
90%	0.0596	0.8251	0.0521	0.0362	1.0233*	0.0475
95%	0.0592	0.8206	0.0529	0.0366	1.0422***	0.0489
100%	0.0591	0.8181	0.0533	0.0351	1.0581***	0.0508
105%	0.0580	0.8039	0.0557	0.0362	1.0204	0.0534
110%	0.0578	0.8006	0.0560	0.0378	1.0091	0.0537
115%	0.0566	0.7838	0.0605	0.0352	1.0146	0.0518
120%	0.0554	0.7670	0.0632	0.0354	0.9652	0.0524
125%	0.0552	0.7641	0.0634	0.0366	0.9901	0.0549
130%	0.0550	0.7613	0.0639	0.0314	0.9824	0.0543
135%	0.0540	0.7484	0.0660	0.0349	0.9504	0.0548
140%	0.0534	0.7395	0.0672	0.0317	0.9529	0.0576
145%	0.0534	0.7395	0.0672	0.0371	0.9328	0.0566
150%	0.0526	0.7285	0.0685	0.0344	0.9557	0.0581

Table A.3: Utility for a single manipulating service provider with 16 service offers in 4 candidate pools. *abs* denotes the mean absolute utility and *rel* the ratio of means of the utility with manipulation and the utility following a truth-telling strategy. *sd* is the standard deviation of the mean absolute utility. * denotes significance at the level of $p = 0.1$, ** at $p = 0.05$, and *** at $p = 0.01$.

	Critical Value Transfer			**Interoperability Transfer**		
Manipulation Rate	abs	rel	sd	abs	rel	sd
-50%	0.0171	0.4002	0.0757	-0.0081	-0.3140	0.0845
-45%	0.0247	0.5793	0.0597	0.0020	0.0757	0.0678
-40%	0.0300	0.7035	0.0465	0.0072	0.2799	0.0546
-35%	0.0340	0.7977	0.0361	0.0107	0.4300	0.0439
-30%	0.0383	0.8983	0.0217	0.0158	0.6344	0.0315
-25%	0.0397	0.9310	0.0163	0.0181	0.7444	0.0234
-20%	0.0413	0.9687	0.0095	0.0209	0.8354	0.0176
-15%	0.0418	0.9814	0.0067	0.0247	0.9011	0.0138
-10%	0.0424	0.9954	0.0027	0.0234	0.9331	0.0083
-5%	0.0426	0.9988	0.0010	0.0252	0.9748	0.0044
0%	0.0426	1.0000	0.0000	0.0248	1.0000	0.0000
5%	0.0425	0.9981	0.0012	0.0265	1.0175***	0.0046
10%	0.0425	0.9980	0.0013	0.0263	1.0453***	0.0070
15%	0.0423	0.9927	0.0035	0.0273	1.0557***	0.0102
20%	0.0420	0.9858	0.0055	0.0274	1.0659***	0.0131
25%	0.0415	0.9744	0.0082	0.0277	1.0570***	0.0157
30%	0.0403	0.9466	0.0144	0.0276	1.0334***	0.0213
35%	0.0402	0.9444	0.0148	0.0266	1.0529***	0.0228
40%	0.0402	0.9434	0.0149	0.0283	1.0562***	0.0246
45%	0.0399	0.9361	0.0162	0.0291	1.0416***	0.0259
50%	0.0394	0.9244	0.0180	0.0271	1.0570***	0.0282
55%	0.0387	0.9079	0.0212	0.0272	1.0326**	0.0304
60%	0.0382	0.8974	0.0227	0.0281	1.0256*	0.0309
65%	0.0377	0.8839	0.0252	0.0272	1.0037	0.0307
70%	0.0373	0.8757	0.0261	0.0267	1.0170	0.0325
75%	0.0367	0.8623	0.0288	0.0277	0.9994	0.0331

Table A.3: Utility for a single manipulating service provider with 16 service offers in 4 candidate pools. *abs* denotes the mean absolute utility and *rel* the ratio of means of the utility with manipulation and the utility following a truth-telling strategy. *sd* is the standard deviation of the mean absolute utility. * denotes significance at the level of $p = 0.1$, ** at $p = 0.05$, and *** at $p = 0.01$.

	Critical Value Transfer			**Interoperability Transfer**		
Manipulation Rate	abs	rel	sd	abs	rel	sd
80%	0.0359	0.8418	0.0315	0.0268	0.9777	0.0376
85%	0.0355	0.8333	0.0330	0.0262	0.9778	0.0366
90%	0.0352	0.8259	0.0339	0.0268	0.9607	0.0391
95%	0.0350	0.8204	0.0344	0.0274	0.9673	0.0372
100%	0.0348	0.8168	0.0348	0.0276	0.9411	0.0395
105%	0.0335	0.7854	0.0405	0.0266	0.9083	0.0372
110%	0.0329	0.7724	0.0414	0.0254	0.8877	0.0383
115%	0.0324	0.7599	0.0430	0.0239	0.8655	0.0404
120%	0.0320	0.7504	0.0437	0.0245	0.8816	0.0412
125%	0.0314	0.7376	0.0463	0.0237	0.8639	0.0403
130%	0.0314	0.7376	0.0463	0.0240	0.8616	0.0420
135%	0.0306	0.7191	0.0485	0.0238	0.8278	0.0443
140%	0.0305	0.7153	0.0487	0.0246	0.8350	0.0444
145%	0.0305	0.7153	0.0487	0.0245	0.8290	0.0434
150%	0.0299	0.7012	0.0506	0.0234	0.8274	0.0440

Table A.4: Utility for a single manipulating service provider with 20 service offers in 4 candidate pools. *abs* denotes the mean absolute utility and *rel* the ratio of means of the utility with manipulation and the utility following a truth-telling strategy. *sd* is the standard deviation of the mean absolute utility. * denotes significance at the level of $p = 0.1$, ** at $p = 0.05$, and *** at $p = 0.01$.

	Critical Value Transfer			**Interoperability Transfer**		
Manipulation Rate	abs	rel	sd	abs	rel	sd
-50%	0.0025	0.1122	0.0630	-0.0111	-0.7315	0.0741
-45%	0.0075	0.3412	0.0502	-0.0032	-0.1944	0.0588
-40%	0.0107	0.4870	0.0425	0.0003	0.0187	0.0495
-35%	0.0147	0.6651	0.0316	0.0065	0.3905	0.0373
-30%	0.0173	0.7854	0.0231	0.0090	0.5533	0.0292

Table A.4: Utility for a single manipulating service provider with 20 service offers in 4 candidate pools. *abs* denotes the mean absolute utility and *rel* the ratio of means of the utility with manipulation and the utility following a truth-telling strategy. *sd* is the standard deviation of the mean absolute utility. * denotes significance at the level of $p = 0.1$, ** at $p = 0.05$, and *** at $p = 0.01$.

	Critical Value Transfer			**Interoperability Transfer**		
Manipulation Rate	abs	rel	sd	abs	rel	sd
-25%	0.0194	0.8822	0.0155	0.0129	0.7391	0.0208
-20%	0.0208	0.9444	0.0089	0.0137	0.8251	0.0146
-15%	0.0212	0.9621	0.0068	0.0135	0.8736	0.0102
-10%	0.0219	0.9916	0.0020	0.0150	0.9434	0.0063
-5%	0.0220	0.9958	0.0011	0.0161	0.9756	0.0031
0%	0.0220	1.0000	0.0000	0.0167	1.0000	0.0000
5%	0.0220	0.9965	0.0009	0.0156	1.0155***	0.0027
10%	0.0219	0.9920	0.0017	0.0169	1.0298***	0.0059
15%	0.0217	0.9855	0.0032	0.0160	1.0339***	0.0074
20%	0.0215	0.9748	0.0051	0.0168	1.0227***	0.0086
25%	0.0210	0.9543	0.0079	0.0168	0.9996	0.0107
30%	0.0205	0.9300	0.0108	0.0157	0.9929	0.0111
35%	0.0199	0.9050	0.0135	0.0152	0.9629	0.0131
40%	0.0195	0.8849	0.0156	0.0150	0.9266	0.0143
45%	0.0192	0.8691	0.0167	0.0151	0.9063	0.0156
50%	0.0191	0.8662	0.0169	0.0149	0.9129	0.0163
55%	0.0190	0.8604	0.0173	0.0152	0.9012	0.0168
60%	0.0189	0.8562	0.0176	0.0150	0.8881	0.0166
65%	0.0188	0.8536	0.0177	0.0150	0.9143	0.0185
70%	0.0185	0.8387	0.0197	0.0148	0.8794	0.0187
75%	0.0184	0.8350	0.0200	0.0152	0.8847	0.0211
80%	0.0183	0.8324	0.0201	0.0153	0.8847	0.0201
85%	0.0183	0.8295	0.0204	0.0152	0.8771	0.0207
90%	0.0182	0.8246	0.0207	0.0149	0.8776	0.0218
95%	0.0181	0.8198	0.0211	0.0143	0.8751	0.0231
100%	0.0179	0.8125	0.0217	0.0149	0.8526	0.0220

Table A.4: Utility for a single manipulating service provider with 20 service offers in 4 candidate pools. *abs* denotes the mean absolute utility and *rel* the ratio of means of the utility with manipulation and the utility following a truth-telling strategy. *sd* is the standard deviation of the mean absolute utility. * denotes significance at the level of $p = 0.1$, ** at $p = 0.05$, and *** at $p = 0.01$.

	Critical Value Transfer			**Interoperability Transfer**		
Manipulation Rate	abs	rel	sd	abs	rel	sd
105%	0.0178	0.8075	0.0222	0.0147	0.8461	0.0224
110%	0.0176	0.7988	0.0235	0.0148	0.8480	0.0234
115%	0.0175	0.7925	0.0241	0.0143	0.8359	0.0254
120%	0.0174	0.7888	0.0243	0.0154	0.8303	0.0266
125%	0.0173	0.7856	0.0245	0.0146	0.8280	0.0238
130%	0.0168	0.7602	0.0270	0.0139	0.7904	0.0270
135%	0.0165	0.7487	0.0284	0.0136	0.7826	0.0286
140%	0.0165	0.7474	0.0285	0.0139	0.7947	0.0293
145%	0.0165	0.7474	0.0285	0.0141	0.7801	0.0291
150%	0.0163	0.7397	0.0293	0.0139	0.7869	0.0279

Table A.5: Utility for a single manipulating service provider with 28 service offers in 4 candidate pools. *abs* denotes the mean absolute utility and *rel* the ratio of means of the utility with manipulation and the utility following a truth-telling strategy. *sd* is the standard deviation of the mean absolute utility. * denotes significance at the level of $p = 0.1$, ** at $p = 0.05$, and *** at $p = 0.01$.

	Critical Value Transfer			**Interoperability Transfer**		
Manipulation Rate	abs	rel	sd	abs	rel	sd
-50%	0.0000	0.0005	0.0501	-0.0048	-0.4739	0.0540
-45%	0.0046	0.3551	0.0371	0.0005	0.0468	0.0411
-40%	0.0081	0.6271	0.0247	0.0037	0.3617	0.0305
-35%	0.0091	0.7086	0.0208	0.0054	0.5255	0.0243
-30%	0.0103	0.8014	0.0152	0.0069	0.6498	0.0191
-25%	0.0113	0.8765	0.0112	0.0076	0.7570	0.0142
-20%	0.0119	0.9275	0.0070	0.0090	0.8521	0.0100
-15%	0.0124	0.9681	0.0042	0.0095	0.9224	0.0066
-10%	0.0127	0.9908	0.0014	0.0097	0.9500	0.0042
-5%	0.0128	0.9972	0.0007	0.0106	0.9837	0.0023

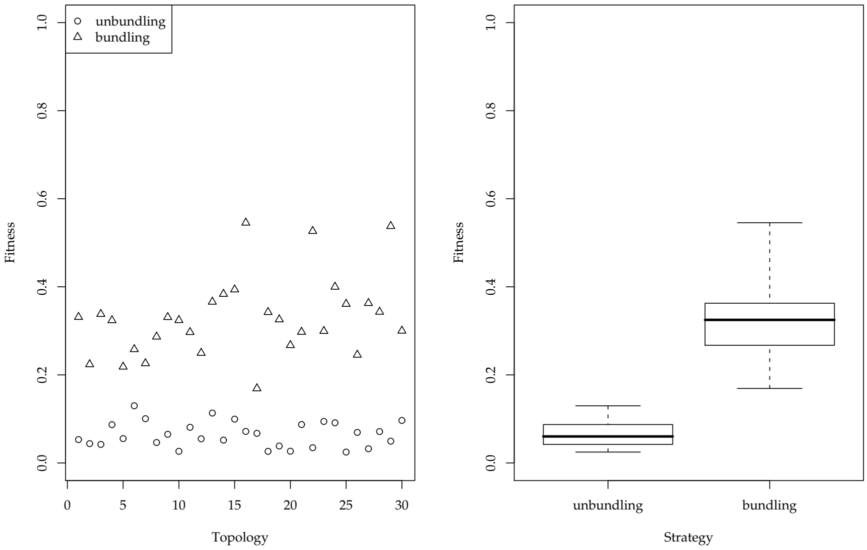
Table A.5: Utility for a single manipulating service provider with 28 service offers in 4 candidate pools. *abs* denotes the mean absolute utility and *rel* the ratio of means of the utility with manipulation and the utility following a truth-telling strategy. *sd* is the standard deviation of the mean absolute utility. * denotes significance at the level of $p = 0.1$, ** at $p = 0.05$, and *** at $p = 0.01$.

	Critical Value Transfer			**Interoperability Transfer**		
Manipulation Rate	abs	rel	sd	abs	rel	sd
0%	0.0129	1.0000	0.0000	0.0101	1.0000	0.0000
5%	0.0128	0.9959	0.0009	0.0106	1.0080***	0.0019
10%	0.0127	0.9873	0.0018	0.0108	1.0044	0.0029
15%	0.0124	0.9625	0.0047	0.0104	0.9845	0.0058
20%	0.0122	0.9489	0.0058	0.0101	0.9681	0.0063
25%	0.0121	0.9393	0.0064	0.0101	0.9587	0.0071
30%	0.0120	0.9315	0.0069	0.0107	0.9546	0.0080
35%	0.0119	0.9268	0.0071	0.0106	0.9563	0.0080
40%	0.0119	0.9240	0.0072	0.0099	0.9526	0.0084
45%	0.0117	0.9133	0.0082	0.0098	0.9396	0.0093
50%	0.0116	0.9059	0.0088	0.0098	0.9350	0.0103
55%	0.0116	0.9022	0.0090	0.0098	0.9432	0.0100
60%	0.0113	0.8799	0.0110	0.0099	0.9054	0.0123
65%	0.0111	0.8628	0.0122	0.0095	0.8963	0.0137
70%	0.0109	0.8455	0.0133	0.0098	0.8773	0.0141
75%	0.0107	0.8294	0.0142	0.0095	0.8635	0.0145
80%	0.0106	0.8232	0.0146	0.0094	0.8464	0.0144
85%	0.0104	0.8115	0.0152	0.0094	0.8522	0.0164
90%	0.0104	0.8083	0.0154	0.0092	0.8546	0.0163
95%	0.0101	0.7858	0.0169	0.0091	0.8210	0.0167
100%	0.0099	0.7667	0.0181	0.0087	0.7969	0.0187
105%	0.0099	0.7667	0.0181	0.0091	0.8050	0.0190
110%	0.0099	0.7667	0.0181	0.0088	0.8045	0.0183
115%	0.0097	0.7556	0.0190	0.0090	0.7827	0.0190
120%	0.0095	0.7410	0.0199	0.0087	0.7596	0.0212
125%	0.0095	0.7360	0.0201	0.0086	0.7604	0.0202

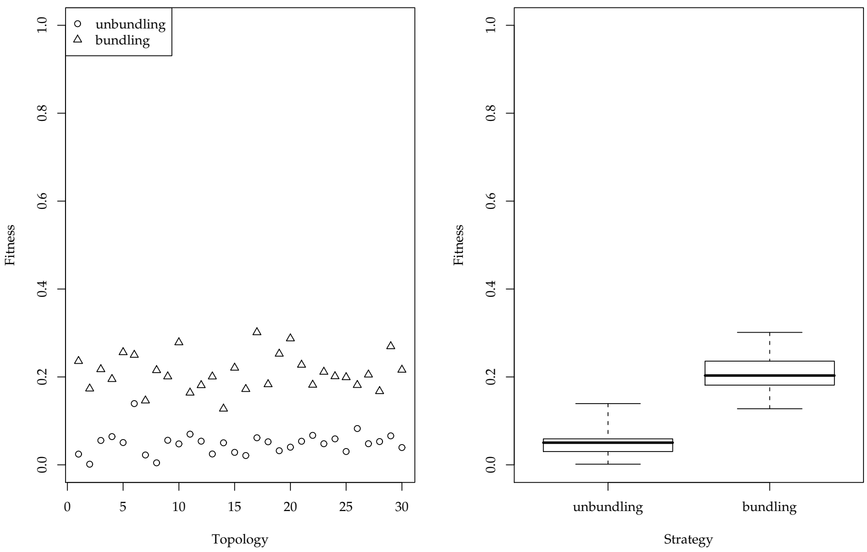
Table A.5: Utility for a single manipulating service provider with 28 service offers in 4 candidate pools. *abs* denotes the mean absolute utility and *rel* the ratio of means of the utility with manipulation and the utility following a truth-telling strategy. *sd* is the standard deviation of the mean absolute utility. * denotes significance at the level of $p = 0.1$, ** at $p = 0.05$, and *** at $p = 0.01$.

	Critical Value Transfer			**Interoperability Transfer**		
Manipulation Rate	abs	rel	sd	abs	rel	sd
130%	0.0093	0.7208	0.0216	0.0081	0.7390	0.0229
135%	0.0093	0.7208	0.0216	0.0086	0.7696	0.0220
140%	0.0091	0.7089	0.0223	0.0083	0.7360	0.0228
145%	0.0090	0.7031	0.0226	0.0081	0.7336	0.0232
150%	0.0089	0.6937	0.0231	0.0082	0.7289	0.0224

A.5 Bundling Strategies



(a) 0% cost reduction due to bundling synergies with 32 service offers in 4 candidate pools.



(b) 50% cost reduction due to bundling synergies with 32 service offers in 4 candidate pools.

Figure A.1: Strategy fitness in different cost reduction scenarios with 32 service offers in 4 candidate pools.

References

[AAA+07] Alexandre Alves, Assaf Arkin, Sid Askary, Charlton Barreto, Ben Bloch, Francisco Curbera, Mark Ford, Yaron Goland, Alejandro Guízar, Neelakantan Kartha, Canyang Kevin Liu, Rania Khalaf, Dieter König, Mike Marin, Vinkesh Mehta, Satish Thatte, Danny van der Rijn, Prasad Yendluri, and Alex Yiu. Web Service Business Process Execution Language (WS-BPEL). Technical report, OASIS, 4 2007. http://docs.oasis-open.org/wsbpel/.

[AB91] B.R. Allen and A.C. Boynton. Information Architecture: In Search of Efficient Flexibility. *MIS Quarterly*, 15(4):435–445, 1991.

[AB08] Ben Adida and Mark Birbeck. Resource Description Framework - in - attributes. Technical report, W3C, 10 2008. http://www.w3.org/TR/xhtml-rdfa-primer/.

[ABC+02] Eric Armstrong, Stephanie Bodoff, Debbie Carson, Maydene Fisher, Dale Green, and Kim Haase. *The Java Web Services Tutorial*. Addison-Wesley, 2002.

[AC05] J. Asker and E. Cantillon. Optimal Procurement When Both Price and Quality Matter. Technical report, 2005.

[AC08] J. Asker and E. Cantillon. Properties of Scoring Auctions. *The RAND Journal of Economics*, 39(1):69–85, 2008.

[ACD+04] A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, J. Pruyne, J. Rofrano, S. Tuecke, and M. Xu. Web Services Agreement Specification (WS-Agreement). In *Global Grid Forum*, 2004.

[ACSV04] A. AuYoung, B.N. Chun, A.C. Snoeren, and A. Vahdat. Resource Allocation in Federated Distributed Computing Infrastructures. In *Proceedings of the 1st Workshop on Operating System and Architectural Support for the On-demand IT InfraStructure*, 2004.

[AGB$^+$04] Daniel Austin, W. W. Grainger, Abbie Barbir, Christopher Ferris, and Sharad Garg. Web Services Architecture Requirements. Technical report, W3C, 2 2004. `http://www.w3.org/TR/wsa-reqs/`.

[Ama08] Amazon. Amazon Web Services Blog. Technical report, Amazon, 5 2008. `http://aws.typepad.com/aws/2008/05/lots-of-bits.html`.

[aN08] Ahuva Mu alem and Noam Nisan. Truthful Approximation Mechanisms for Restricted Combinatorial Auctions. *Games and Economic Behavior*, 2008.

[And06] C. Anderson. *The Long Tail: How Endless Choice is Creating Unlimited Demand*. Random House Business Books, 2006.

[AT07] Aaron Archer and Eva Tardos. Frugal Path Mechanisms. *ACM Transactions on Algorithms*, 3(1):3, 2007.

[BBL99] Y. Bakos, E. Brynjolfsson, and D. Lichtman. Shared Information Goods. *The Journal of Law and Economics*, 42(1):117–156, 1999.

[BBS08] B. Blau, C. Block, and J. Stösser. How to trade Electronic Services? – Current Status and Open Questions. In *Proceedings of the Joint Conference of the INFORMS section on Group Decision and Negotiation, the EURO Working Group on Decision and Negotiation Support, and the EURO Working Group on Decision Support Systems*, 2008.

[BBT09] James Broberg, Rajkumar Buyya, and Zahir Tari. MetaCDN: Harnessing Storage Clouds for High Performance Content Delivery. *Journal of Network and Computer Applications*, In Press, Corrected Proof, 2009.

[BC00] C.Y. Baldwin and K.B. Clark. *Design Rules: Volume 1: The Power of Modularity*. Mit Press Cambridge, MA, 2000.

[BCC$^+$04] Don Box, Erik Christensen, Francisco Curbera, Donald Ferguson, Jeffrey Frey, Marc Hadley, Chris Kaler, David Langworthy, Frank Leymann, Brad Lovering, Steve Lucco, Steve Millet, Nirmal Mukhi, Mark Nottingham, David Orchard, John Shewchuk, Eugene Sindambiwe, Tony Storey, Sanjiva Weerawarana, and Steve Winkler. Web Services Addressing (WS-Addressing). Technical report, W3C, 8 2004. `http://www.w3.org/Submission/ws-addressing/`.

[BCM$^+$07] F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, and P.F. Patel-Schneider. *The Description Logic Handbook*. Cambridge University Press New York, NY, USA, 2007.

[BCM09] B. Blau, T. Conte, and T. Meinl. Coordinating Service Composition. In *Proceedings of the 17th European Conference on Information Systems*, 2009.

[BDBD$^+$00] Gabe Beged-Dov, Dan Brickley, Rael Dornfest, Ian Davis, Leigh Dodds, Jonathan Eisenzopf, David Galbraith, R.V. Guha, Ken MacLeod, Eric Miller, Aaron Swartz, and Eric van der Vlist. RDF Site Summary (RSS) 1.0. Technical report, 2000. http://purl.org/rss/1.0/spec/.

[BDF$^+$03] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. Xen and the Art of Virtualization. *ACM SIGOPS Operating Systems Review*, 37(5):164–177, 2003.

[BEA08] BEA. Revised Statistics of Gross Domestic Product by Industry, 2004-2006. Technical report, BEA (Bureau of Economic Analysis), 2008.

[BEK$^+$00] Don Box, David Ehnebuske, Gopal Kakivaya, Andrew Layman, Noah Mendelsohn, Henrik Frystyk Nielsen, Satish Thatte, and Dave Winer. Web Services Architecture Requirements. Technical report, W3C, 5 2000. http://www.w3.org/TR/soap/.

[Ben38] J. Bentham. An Introduction to the Principles of Morals and Legislation. *The Works of Jeremy Bentham*, 43, 1838.

[BFHZ97] M.J. Bitner, W.T. Faranda, A.R. Hubbert, and V.A. Zeithaml. Customer Contributions and Roles in Service Delivery. *International Journal of Service Industry Management*, 8(3):193–205, 1997.

[BG00] V. Bala and S. Goyal. A Noncooperative Model of Network Formation. *Econometrica*, pages 1181–1229, 2000.

[BK05] M. Bichler and J. Kalagnanam. Configurable Offers and Winner Determination in Multi-Attribute Auctions. *European Journal of Operational Research*, 160(2):380–394, 2005.

[BKCvD09] B. Blau, J. Krämer, T. Conte, and C. van Dinther. Service Value Networks. In *Proceedings of the 11th IEEE Conference on Commerce and Enterprise Computing (CEC 2009)*, 2009.

[BKO02] G. Bolton, E. Katok, and A. Ockenfels. How Effective are Online Reputation Mechanisms. *Discussion Papers on Strategic Interaction*, 25:2002–25, 2002.

[BLFM98] T. Berners-Lee, R. Fielding, and L. Masinter. RFC2396: Uniform Resource Identifiers (URI): Generic Syntax. *RFC Editor United States*, 1998.

[BLH09] B. Blau, S. Lamparter, and S. Haak. remash*!* - Blueprints for RESTful Situational Web Applications. In *Proceedings of the 2nd Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009)*, 2009.

[BLNW08] B. Blau, S. Lamparter, D. Neumann, and C. Weinhardt. Planning and pricing of service mashups. In *10th IEEE Joint Conference on E-Commerce Technology (CEC 2008) and Enterprise Computing, E-Commerce and E-Services (EEE 2008), 21-24 July 2008, Washington, D.C., USA*, 2008.

[BNWM08] B. Blau, D. Neumann, C. Weinhardt, and W. Michalk. Provisioning of service mashup topologies. In *Proceedings of the 16th European Conference on Information Systems, ECIS 2008*, 2008.

[Bon02] E. Bonabeau. Agent-Based Modeling: Methods And Techniques for Simulating Human Systems. In *National Academy of Sciences*, volume 99, pages 7280–7287. National Acad Sciences, 2002.

[Bor09] Nikolay Borissov. Q-Strategy: Automated Bidding and Convergence in Computational Markets. In *21st Innovative Applications of Artificial Intelligence (IAAI) Conference collocated with IJCAI*, July 2009.

[BP91] L.L. Berry and A. Parasuraman. *Marketing Services: Competing Through Quality*. Free Press, 1991.

[BPSM$^+$06] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, and François Yergeau. Extensible Markup Language (XML). Technical report, W3C, 8 2006. `http://www.w3.org/XML/`.

[BR04] R. Bianchini and R. Rajamony. Power and Energy Management for Server Systems. *Computer*, 37(11):68–76, 2004.

[Bra97] F. Branco. The Design of Multidimensional Auctions. *RAND Journal of Economics*, 28(1):63–81, 1997.

[BS99] P.D. Bridge and S.S. Sawilowsky. Increasing PhysiciansŠ Awareness of the Impact of Statistics on Research Outcomes Comparative Power of the T-Test and Wilcoxon Rank-Sum Test in Small Samples

Applied Research. *Journal of Clinical Epidemiology*, 52(3):229–235, 1999.

[BS00] K. Binmore and J. Swierzbinski. Treasury Auctions: Uniform or Discriminatory? *Review of Economic Design*, 5(4):387–410, 2000.

[BS08] B. Blau and B. Schnizler. Description languages and mechanisms for trading service objects in grid markets. In Martin Bichler, Thomas Hess, Helmut Krcmar, Ulrike Lechner, Florian Matthes, Arnold Picot, Benjamin Speitkamp, and Petra Wolf, editors, *Multikonferenz Wirtschaftsinformatik, MKWI 2008, München, 26.2.2008 - 28.2.2008, Proceedings*. GITO-Verlag 2008 Berlin, 2 2008.

[Bur04] M. Burner. Service Orientation and Its Role in Your Connected Systems Strategy. *Microsoft White Paper, July*, 2004.

[BvDC+09] Benjamin Blau, Clemens van Dinther, Tobias Conte, Yongchun Xu, and Christof Weinhardt. How to Coordinate Value Generation in Service Networks? – A Mechanism Design Approach. *(forthcoming), Journal of Business and Information Systems Engineering (Wirtschaftsinformatik), Special Issue Internet of Services*, 2009.

[BvDCW09] Benjamin Blau, Clemens van Dinther, Tobias Conte, and Christof Weinhardt. A Multidimensional Procurement Auction for Trading Composite Services. *Electronic Commerce Research and Applications, Special Issue on Emerging Economic, Strategic and Technical Issues in Online Auctions and Electronic Market Mechanisms (submitted)*, 2009.

[BVEL04] S. Brockmans, R. Volz, A. Eberhart, and P. Loffler. Visual Modeling of OWL DL Ontologies Using UML. *Lecture Notes in Computer Science*, pages 198–213, 2004.

[CAT+01] Jeffrey S. Chase, Darrell C. Anderson, Prachi N. Thakar, Amin M. Vahdat, and Ronald P. Doyle. Managing Energy and Server Resources in Hosting Centers. *SIGOPS Oper. Syst. Rev.*, 35(5):103–116, 2001.

[CBSvD09] T. Conte, B. Blau, G. Satzger, and C. van Dinther. Enabling service networks through contribution-based value distribution. In *Proceedings of the 15th Americas Conference on Information Systems*, 2009.

[CCMW01] Erik Christensen, Francisco Curbera, Greg Meredith, and Sanjiva Weerawarana. Web Service Description Language (WSDL) 1.1. Technical report, W3C, 3 2001. `http://www.w3.org/TR/wsdl/`.

[CHvRR04] Luc Clement, Andrew Hately, Claus von Riegen, and Tony Rogers. Universal Description, Discovery, and Integration (UDDI). Technical report, OASIS, 10 2004. `https://http://uddi.org/pubs/`.

[CIoWM93] Y.K. Che, Social Systems Research Institute, and University of Wisconsin-Madison. Design Competition Through Multidimensional Auctions. *RAND Journal of Economics*, 24:668–668, 1993.

[Cla71] E.H. Clarke. Multipart Pricing of Public Goods. *Public Choice*, 11(1):17–33, 1971.

[CNLP05] Martin Chapter, Eric Newcomer, Mark Little, and Greg Pavlik. Web Services Coordination Framework (WS-CF). Technical report, OASIS, Public Review Draft, 10 2005. `http://www.oasis-open.org/committees/ws-caf/`.

[Cro06] D. Crockford. JSON: The Fat-Free Alternative To XML. In *Proceedings of XML*, 2006.

[CSM$^+$04] J. Cardoso, A. Sheth, J. Miller, J. Arnold, and K. Kochut. Quality of Service for Workflows and Web Service Processes. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(3):281–308, 2004.

[CvD09] T. Conte C. van Dinther, B. Blau. Strategic Behavior in Service Networks under Price and Service Level Competition. In *Proceedings of the 9th International Conference on Business Informatics*, 2009.

[Dev98] J.F. Devlin. Adding Value to Service Offerings: The Case of UK Retail Financial Services. *European Journal of Marketing*, 32(11):1091–1109, 1998.

[Dij59] EW Dijkstra. A Note on Two Problems in Connexion With Graphs. *Numerische Mathematik*, 1(1):269–271, 1959.

[DJP03] RK Dash, NR Jennings, and DC Parkes. Computational-Mechanism Design: A Call to Arms. *IEEE Intelligent Systems*, 18(6):40–47, 2003.

[DLP03] A. Dan, H. Ludwig, and G. Pacifici. Web Service Differentiation with Service Level Agreements. *White Paper, IBM Corporation*, 3 2003.

[DM93] W.H. Davidow and M.S. Malone. *The Virtual Corporation: Structuring and Revitalizing The Corporation for the 21st Century*. HarperBusiness, 1993.

[DSBF01] G. Da Silveira, D. Borenstein, and F.S. Fogliatto. Mass Customization: Literature Review and Research Directions. *International Journal of Production Economics*, 72(1):1–13, 2001.

[DVVfMSiES03] S. De Vries, R.V. Vohra, Center for Mathematical Studies in Economics, and Management Science. Combinatorial Auctions: A Survey. *INFORMS Journal on Computing*, 15(3):284–309, 2003.

[EOS07] B. Edelman, M. Ostrovsky, and M. Schwarz. Internet Advertising and the Generalized Second-Price Auction: Selling Billions of Dollars Worth of Keywords. *American Economic Review*, 97(1):242–259, 2007.

[Eso01] M. Eso. An Iterative Online Auction for Airline Seats. *IMA Volumes In Mathematics And Its Applications*, 127:45–58, 2001.

[ESS04] E. Elkind, A. Sahai, and K. Steiglitz. Frugality in Path Auctions. In *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 701–709. Society for Industrial and Applied Mathematics Philadelphia, PA, USA, 2004.

[Eva91] J.S. Evans. Strategic Flexibility for High Technology Manoeuvres: A Conceptual Framework. *Journal of Management Studies*, 28(1):69–89, 1991.

[EWL06] Yagil Engel, Michael P. Wellman, and Kevin M. Lochner. Bid Expressiveness and Clearing Algorithms in Multiattribute Double Auctions. In *Proceedings of the 7th ACM Conference on Electronic Commerce*, pages 110–119. ACM, 2006.

[FCSS05] Michael Feldman, John Chuang, Ion Stoica, and Scott Shenker. Hidden-Action in Multi-Hop Routing. In *Proceedings of the 6th ACM Conference on Electronic commerce*, pages 117–126. ACM, 2005.

[FGM$^+$99] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. RFC2616: Hypertext Transfer Protocol–HTTP/1.1. *RFC Editor United States*, 1999.

[Fie00] Roy Thomas Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, 2000.

[FK07] J. Farrell and P. Klemperer. Coordination and Lock-In: Competition with Switching Costs and Network Effects. *Handbook of Industrial Organization*, page 1967, 2007.

[FKNT02] I. Foster, C. Kesselman, J.M. Nick, and S. Tuecke. Grid Services for Distributed System Integration. *COMPUTER*, pages 37–46, 2002.

[FL07] Joel Farrell and Holger Lausen. Semantic Annotations for WSDL and XML Schema. Technical report, W3C, 8 2007. `http://www.w3.org/TR/sawsdl/`.

[FPP09] Joan Feigenbaum, David C. Parkes, and David M. Pennock. Computational Challenges in E-commerce. *Communications of the ACM*, 52(1):70–74, 2009.

[FRS06] Joan Feigenbaum, Vijay Ramachandran, and Michael Schapira. Incentive-Compatible Interdomain Routing. In *Proceedings of the 7th ACM Conference on Electronic Commerce*, pages 130–139, 2006.

[Fuc68] V.R. Fuchs. *The Service Economy*. Natl Bureau of Economic Res, 1968.

[Gad92] J. Gadrey. L'économie des Services. 1992.

[Gad00] J. Gadrey. The Characterization of Goods and Services: An Alternative Approach. *Review of Income and Wealth*, 46(3):369–387, 2000.

[Gal73] J.R. Galbraith. *Designing Complex Organizations*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1973.

[Gib73] Allan Gibbard. Manipulation of Voting Schemes: A General Result. *Econometrica*, 41(4):587–601, July 1973.

[Gib92] R. Gibbons. *Game Theory for Applied Economists*. Princeton University Press Princeton, 1992.

[GL78] Jerry R. Green and Jean-Jacques Laffont. *Incentives in Public Decision – Making, Studies in Public Economics*. North–Holland Publishing Company, Boston, 1978.

[GNC$^+$04] Steve Graham, Peter Niblett, Dave Chappell, Amy Lewis, Nataraj Nagaratnam, Jay Parikh, Sanjay Patil, Shivajee Samdarshi, Igor Sedukhin, David Snelling, Steve Tuecke, William Vambenepe, and Bill Weihl. Web Services Notification (WS-Notification). Technical report, OASIS, 5 2004. `http://www.oasis-open.org/committees/wsn/`.

[GR71] P.E. Green and V.R. Rao. Conjoint Measurement for Quantifying Judgmental Data. *Journal of Marketing Research*, pages 355–363, 1971.

[Gri92] Z. Griliches. Output Measurement in the Service Sectors, Studies in Income and Wealth. 56, 1992.

[Gro73] Theodore Groves. Incentives in Teams. *Econometrica*, 41(4):617–631, 1973.

[GS06] J. Gebauer and F. Schober. Information System Flexibility and the Cost Efficiency of Business Processes. *Journal of the Association for Information Systems*, 7(3):122–147, 2006.

[GSB$^+$02] S. Graham, S. Simeonov, T. Boubez, D. Davis, G. Daniels, Y. Nakamura, and R. Neyama. *Building Web services with Java*. Sams, 2002.

[GW97] F. Gallouj and O. Weinstein. Innovation in Services. *Research Policy*, 26(4-5):537–556, 1997.

[Had06] Marc J. Hadley. Web Application Description Language (WADL). Technical report, Sun Microsystems Inc., 11 2006. https://wadl.dev.java.net/.

[Hil77] T.P. Hill. On Goods and Services. *Review of Income and Wealth*, 23(4):315–338, 1977.

[Hil99] T.P. Hill. Tangibles, Intangibles and Services: A New Taxonomy for the Classification of Output. *Canadian Journal of Economics*, 32:426–446, 1999.

[HN96] D. Harel and A. Naamad. The STATEMATE Semantics of Statecharts. *ACM Transactions on Software Engineering and Methodology*, 5(4):293–333, 1996.

[HPSB$^+$04] Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, and Mike Dean. Semantic Web Rule Language (SWRL). Technical report, W3C, 5 2004. http://www.w3.org/Submission/SWRL/.

[HS01] J. Hershberger and S. Suri. Vickrey Prices and Shortest Paths: What Is an Edge Worth? In *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*, pages 252–259, 2001.

[Hur72] L. Hurwicz. On Informationally Decentralized Systems/Decision And Organization. *Radner, R., CB McGuire. In Honor of J. Marschak*, 1972.

[Hur73] L. Hurwicz. The Design of Mechanisms for Resource Allocation. *American Economic Review*, 63(2):1–30, 1973.

[HW90] L. Hurwicz and M. Walker. On the Generic Nonoptimality of Dominant-Strategy Allocation Mechanisms: A General Theorem that Includes Pure Exchange Economies. *Econometrica: Journal of the Econometric Society*, pages 683–704, 1990.

[IL04] M. Iansiti and R. Levien. Strategy as Ecology. *Harvard Business Review*, 82(3):68–81, 2004.

[Jac92] M.O. Jackson. Incentive Compatibility and Competitive Allocations. *Economics Letters*, 40:299–302, 1992.

[Jac03] M.O. Jackson. Efficiency and Information Aggregation in Auctions With Costly Information. *Review of Economic Design*, 8(2):121, 2003.

[JF03] R. Jurca and B. Faltings. An Incentive Compatible Reputation Mechanism. In *Proceedings of the IEEE International Conference on E-Commerce*, pages 285–292, 2003.

[Jhi06] A. Jhingran. Enterprise Information Mashups: Integrating Information, Simply. In *Proceedings of the 32nd International Conference on Very Large Data Bases*, pages 3–4. VLDB Endowment, 2006.

[JIB07] A. Jøsang, R. Ismail, and C. Boyd. A Survey of Trust and Reputation Systems for Online Service Provision. *Decision Support Systems*, 43(2):618–644, 2007.

[JMS02] L. Jin, V. Machiraju, and A. Sahai. Analysis on Service Level Agreement of Web Services. *HP*, 6 2002.

[JW96] M.O. Jackson and A. Wolinsky. A Strategic Model of Social and Economic Networks. *Journal of economic Theory*, 71(1):44–74, 1996.

[JW02] M.O. Jackson and A. Watts. The Evolution of Social and Economic Networks. *Journal of Economic Theory*, 106(2):265–295, 2002.

[KCS08] A. Kittur, E.H. Chi, and B. Suh. Crowdsourcing User Studies with Mechanical Turk. 2008.

[KK05] AR Karlin and D. Kempe. Beyond VCG: Frugality of Truthful Mechanisms. In *Foundations of Computer Science, 2005. FOCS 2005. 46th Annual IEEE Symposium on*, pages 615–624, 2005.

[KN04] D. Karger and E. Nikolova. VCG Overpayment in Random Graphs. In *DIMACS Workshop on Computational Issues in Auction Design*, 2004.

[KN05] D. Karger and E. Nikolova. Brief Announcement: On the Expected Overpayment of VCG Mechanisms in Large Networks. In *Proceedings of the twenty-fourth annual ACM symposium on Principles of distributed computing*, pages 126–126. ACM New York, NY, USA, 2005.

[Kra05] B. Kratz. Protocols For Long Running Business Transactions. Technical Report 17, Infolab Technical Report Series, 2005.

[KS85] M.L. Katz and C. Shapiro. Network Externalities, Competition, and Compatibility. *The American Economic Review*, pages 424–440, 1985.

[KV98] S. Kochugovindan and N.J. Vriend. Is the Study of Complex Adaptive Systems Going to Solve the Mystery of Adam Smith's Invisible Hand? *Independent Review*, 3:53–66, 1998.

[Lai05] K. Lai. Markets are Dead, Long Live Markets. *ACM SIGecom Exchanges*, 5(4):1–10, 2005.

[Lam07] Steffen Lamparter. *Policy-Based Contracting in Semantic Web Service Markets*. PhD thesis, Universität Karlsruhe (TH), 2007.

[Lev81] T. Levitt. Marketing Intangible Products and Product Intangibles. *Cornell Hotel and Restaurant Administration Quarterly*, 22(2):37, 1981.

[Ley03] F. Leymann. Web Services: Distributed Applications without Limits. *Business, Technology and Web*, 2003.

[LGS07] Jon Lathem, Karthik Gomadam, and Amit P. Sheth. SA-REST and (S)mashups: Adding Semantics to RESTful Services. In *ICSC '07: Proceedings of the International Conference on Semantic Computing*, pages 469–476, Washington, DC, USA, 2007. IEEE Computer Society.

[LM94] SJ Liebowitz and S.E. Margolis. Network Externality: An Uncommon Tragedy. *The Journal of Economic Perspectives*, pages 133–150, 1994.

[LNZ04] Yutu Liu, Anne H. Ngu, and Liang Z. Zeng. QoS Computation and Policing in Dynamic Web Service Selection. In *Proceedings of the 13th international World Wide Web conference on Alternate Track Papers & Posters*, pages 66–73, New York, NY, USA, 2004. ACM.

[LR00] D. Lucking-Reiley. Auctions on the Internet: What's Being Auctioned, and How? *Journal of Industrial Economics*, 48(3):227–252, 2000.

[LS06] S. Lamparter and B. Schnizler. Trading Services in Ontology-Driven Markets. In *Proceedings of the 2006 ACM symposium on Applied computing*, pages 1679–1683. ACM New York, NY, USA, 2006.

[LSW01] Z. Liu, M.S. Squillante, and J.L. Wolf. On Maximizing Service-Level-Agreement Profits. In *Proceedings of the 3rd ACM conference on Electronic Commerce*, pages 213–223. ACM New York, NY, USA, 2001.

[LT64] R.D. Luce and J.W. Tukey. Simultaneous Conjoint Measurement: A New Type of Fundamental Measurement. *Journal of Mathematical Psychology*, 1(1):1–27, 1964.

[LVO07] R.F. Lusch, S.L. Vargo, and M. OŠBrien. Competing Through Service: Insights From Service-Dominant Logic. *Journal of Retailing*, 83(1):5–18, 2007.

[LW01] C.H. Lovelock and J. Wirtz. *Services Marketing: People, Technology, Strategy*. Prentice Hall, 2001.

[LW03] M. Little and J. Webber. Introducing WS-CAF – More Than Just Transactions. *Web Services Journal*, 3(12):52–55, 2003.

[Mal85] T.W. Malone. Organizational Structure and Information Technology: Elements of a Formal Theory. 1985.

[Mal87] Thomas W. Malone. Modeling Coordination in Organizations and Markets. *Management Science*, 33(10):1317–1332, 1987.

[MB09] T. Meinl and B. Blau. Web Service Derivatives. In *Proceedings of the 18thInternational World Wide Web Conference (WWW2009)*, Madrid, Spain, 4 2009.

[MC94] Thomas W. Malone and Kevin Crowston. The Interdisciplinary Study of Coordination. *ACM Comput. Surv.*, 26(1):87–119, 1994.

[MCWG95] A. Mas-Colell, M.D. Whinston, and J.R. Green. *Microeconomic Theory*. Oxford University Press New York, 1995.

[Men02] DA Menasce. QoS Issues in Web services. *IEEE Internet Computing*, 6(6):72–75, 2002.

[Mer06] D. Merrill. Mashups: The New Breed of Web App – An Introduction to Mashups. Technical report, IBM, 8 2006. `http://www.ibm.com/developerworks/xml/library/x-mashups.html`.

[MLM+06] C. Matthew MacKenzie, Ken Laskey, Francis McCabe, Peter F Brown, and Rebekah Metz. Reference Model for Service Oriented Architecture 1.0. Technical report, OASIS, 10 2006.

[MMV94] J.K. MacKie-Mason and H.R. Varian. Generalized Vickrey Auctions. *Technology report. University of Michigan, July*, 1994.

[MMW06] J.K. MacKie-Mason and M.P. Wellman. Automated Markets and Trading Agents. *Ann Arbor*, 1001:48109–1092, 2006.

[MN02] A. Mani and A. Nagarajan. Understanding quality of service for Web services. *IBM developerWorks*, 1 2002.

[MN08] A. Mu'Alem and N. Nisan. Truthful Approximation Mechanisms for Restricted Combinatorial Auctions. *Games and Economic Behavior*, 64(2):612–631, 2008.

[MNM+07] M. Mohabey, Y. Narahari, S. Mallick, P. Suresh, and SV Subrahmanya. A Combinatorial Procurement Auction for QoS-Aware Web Services Composition. In *IEEE International Conference on Automation Science and Engineering, 2007. CASE 2007*, pages 716–721, 2007.

[MPW08] R. Müller, A. Perea, and S. Wolf. Combinatorial Scoring Auctions. Technical report, 2008.

[MS83] R. Myerson and M. Satterthwaite. Efficient Mechanisms for Bilateral Exchange. *Journal of Economic Theory*, 28:265–281, 1983.

[MS84] T.W. Malone and S.A. Smith. Tradeoffs in Designing Organizations: Implications for New Forms of Human Organizations and Computer Systems. 1984.

[MS86] R.E. Miles and C.C. Snow. Organizations: New Concepts for New Forms. *California Management Review*, 28(3):62–74, 1986.

[MSS+08] Justin Moore, Ratnesh Sharma, Rocky Shih, Jeff Chase, Chandrakant Patel, and Parthasarathy Ranganathan. Going beyond CPUs: The Potential of Temperature-Aware Solutions for the Data Center. Whitepaper, Hewlett Packard Labs, January 2008.

[MSZ01] S.A. McIlraith, T.C. Son, and H. Zeng. Semantic Web Services. *IEEE Intelligent Systems*, pages 46–53, 2001.

[MT07] P. Maille and B. Tuffin. Why VVG Auctions Can Hardly be Applied to the Pricing of Inter-Domain and Ad Hoc Networks. In *3rd EuroNGI Conference on Next Generation Internet Networks*, pages 36–39, 2007.

[Mul06] A. Mulholland. The End of Business as Usual: Service-Oriented Business Transformation. *Lecture Notes in Computer Science*, 4294:540, 2006.

[MV98] P. Matthyssens and K. Vandenbempt. Creating Competitive Advantage in Industrial Services. *Journal Of Business and Industrial Marketing*, 13:339–355, 1998.

[MvH04] Deborah L. McGuinness and Frank van Harmelen. Web Ontology Language (OWL). Technical report, W3C, 2 2004. http://www.w3.org/2004/OWL/.

[MWL+06] T.W. Malone, P. Weill, R.K. Lai, V.T. D'Urso, G. Herman, T.G. Apel, S. Woerner, and I. Author. Do Some Business Models Perform Better than Others? Technical report, 2006.

[MYB87] Thomas W. Malone, Joanne Yates, and Robert I. Benjamin. Electronic Markets and Electronic Hierarchies. *Communications of the ACM*, 30(6):484–497, 1987.

[Mye81] R.B. Myerson. Optimal Auction Design. *Mathematics of operations research*, pages 58–73, 1981.

[Mye82] Roger B. Myerson. Optimal Coordination Mechanisms in Generalized Principal-Agent Problems. *Journal of Mathematical Economics*, 10(1):67–81, June 1982.

[Mye88] R.B. Myerson. Mechanism Design. 1988.

[Neu04] Dirk Georg Neumann. *Market Engineering – A Structured Design Process for Electronic Markets*. PhD thesis, Universität Karlsruhe (TH), 2004.

[NKMHB06] Anthony Nadalin, Chris Kaler, Ronald Monzillo, and Phillip Hallam-Baker. Web Services Security: SOAP Message Security 1.1 (WS-Security). Technical report, OASIS, 2 2006. http://docs.oasis-open.org/wss/v1.1/.

[NR01] N. Nisan and A. Ronen. Algorithmic Mechanism Design. *Games and Economic Behavior*, 35(1-2):166–196, 2001.

[NR07] N. Nisan and A. Ronen. Computationally Feasible VCG Mechanisms. *Journal of Artificial Intelligence Research*, 29:19–47, 2007.

[NRFJ07] Eric Newcomer, Ian Robinson, Max Feingold, and Ram Jeyaraman. Web Services Coordination (WS-Coordination). Technical report, OASIS, 7 2007. `http://docs.oasis-open.org/ws-tx/wscoor/`.

[NRFL07] Eric Newcomer, Ian Robinson, Tom Freund, and Mark Little. Web Services Business Activity (WS-BusinessActivity). Technical report, OASIS, 7 2007. `http://docs.oasis-open.org/ws-tx/wsba/`.

[NRLW07] Eric Newcomer, Ian Robinson, Mark Little, and Andrew Wilkinson. Web Services Atomic Transaction (WS-AtomicTransaction). Technical report, OASIS, 7 2007. `http://docs.oasis-open.org/ws-tx/wsat/`.

[NRTV07] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, 2007.

[NS06] N. Nisan and A. Sen. Weak Monotonicity Characterizes Deterministic Dominant-Strategy Implementation. *Econometrica*, pages 1109–1132, 2006.

[OEC05] OECD. Science, Technology and Industry Scoreboard 2005 – Towards a Knowledge-Based Economy. Technical report, OECD, 2005.

[OMG07] OMG. The Unified Modeling Language (UML) 2.1.2. Technical report, Object Management Group (OMG), 4 2007. `http://www.omg.org/spec/UML/2.1.2/`.

[Pap01] C. Papadimitriou. Algorithms, games, and the internet. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 749–753. ACM New York, NY, USA, 2001.

[Pap08] P. Papazoglou. *Web Services: Principles and Technologies*. Prentice Hall, 2008.

[Par01] D.C. Parkes. *Iterative Combinatorial Auctions: Achieving Economic and Computational Efficiency*. PhD thesis, University of Pennsylvania, 2001.

[Pau08] C. Pautasso. BPEL for REST. In *Proceedings of the 6th International Conference on Business Process Management (BPM 2008), Milan, Italy*. Springer, September 2008.

[PBB+04] M. Pistore, F. Barbon, P. Bertoli, D. Shaparau, and P. Traverso. Planning and Monitoring Web service Composition. *Lecture Notes in Computer Science*, pages 106–115, 2004.

[PD04] M.P. Papazoglou and J. Dubray. A Survey of Web Service Technologies. Technical report, University of Tronto, Department of Information and Communication Technology, 6 2004.

[PG03] M.P. Papazoglou and D. Georgakopoulos. Service-Oriented Computing. *Communications of the ACM*, 46(10):25–28, 2003.

[Phe08] S.G. Phelps. *Evolutionary Mechanism Design*. PhD thesis, University of Liverpool, 2008.

[PK02] D. Parkes and J. Kalagnanam. Iterative Multiattribute Vickrey Auctions. Technical report, Harvard University, 2002.

[PK05] D.C. Parkes and J. Kalagnanam. Models for Iterative Multiattribute Procurement Auctions. *Management Science*, 51(3):435–451, 2005.

[PKE01] D.C. Parkes, J. Kalagnanam, and M. Eso. Achieving Budget-Balance with Vickrey-Based Payment Schemes in Combinatorial Exchanges. Technical report, IBM Research, 2001.

[PMS04] F.T. Piller, K. Moeslein, and C.M. Stotko. Does Mass Customization Pay? An Economic Approach to Evaluate Customer Integration. *Production Planning & Control*, 15(4):435–444, 2004.

[PS98] C.H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Dover Publications, 1998.

[PS00] W. Pesendorfer and J.M. Swinkels. Efficiency and Information Aggregation in Auctions. *American Economic Review*, 90(3):499–525, 2000.

[PZL08] C. Pautasso, O. Zimmermann, and F. Leymann. RESTful Web Services vs. Big Web Services: Making the Right Architectural Decision. ACM New York, NY, USA, 2008.

[Ram80] P.H. Ramsey. Choosing the Most Powerful Pairwise Multiple Comparison Procedure in Multivariate Analysis of Variance. *Journal of Applied Psychology*, 65(3,317-326), 1980.

[Rap04] M.A. Rappa. The Utility Business Model and the Future of Computing Services. *IBM Systems Journal*, 43(1):32–42, 2004.

[Rat66] J.M. Rathmell. What is meant by services? *Journal of Marketing*, 30(4):32–36, 1966.

[Rei77] Stanley Reiter. Information and Performance in the (New) Welfare Economics. *The American Economic Review*, 67(1):226–234, 1977.

[RH07] Stuart Rance and Ashley Hanna. Glossary of Terms and Definitions. Technical report, ITIL IT Service Management, 2007.

[RK02] R.T. Rust and PK Kannan. *E-Service: New Directions in Theory and Practice*. ME Sharpe, 2002.

[RK03] R.T. Rust and PK Kannan. E-service: A New Paradigm for Business in the Electronic Environment. *Communications of the ACM*, 46(6):36–42, 2003.

[RL05] A. Ronen and D. Lehmann. Nearly Optimal Multi-Attribute Auctions. In *Proceedings of the 6th ACM conference on Electronic commerce*, pages 279–285. ACM Press New York, NY, USA, 2005.

[Ron01] Amir Ronen. On Approximating Optimal Auctions. In *Proceedings of the 3rd ACM Conference on Electronic Commerce*, pages 11–17. ACM, 2001.

[Rot02] A.E. Roth. The Economist as Engineer: Game Theory, Experimentation, and Computation as Tools for Design Economics. *Econometrica*, pages 1341–1378, 2002.

[RP76] D.J. Roberts and A. Postlewaite. The Incentives for Price-Taking Behavior in Large Exchange Economies. *Econometrica: Journal of the Econometric Society*, pages 115–127, 1976.

[RPH98] M.H. Rothkopf, A. Pekeč, and R.M. Harstad. Computationally Manageable Combinational Auctions. *Management Science*, pages 1131–1147, 1998.

[RR07] L. Richardson and S. Ruby. *RESTful Web Services*. O'Reilly, 2007.

[Saa80] T.L. Saaty. *The Analytical Hierarchy Process*. McGraw-Hill, New York, 1980.

[Saa08] T.L. Saaty. Decision Making with the Analytic Hierarchy Process. *International Journal of Services Sciences*, 1(1):83–98, 2008.

[SB92] SS Sawilowsky and RC Blair. A More Realistic Look at the Robustness and Type II Error Properties of the T Test to Departures from Population Normality. *Psychological Bulletin*, 111(2):352–360, 1992.

[SB99] RS Sutton and AG Barto. Reinforcement Learning. *Journal of Cognitive Neuroscience*, 11(1):126–134, 1999.

[SB04] M. Salle and C. Bartolini. Management by Contract. *Network Operations and Management Symposium, 2004. NOMS 2004. IEEE/IFIP*, 1, 2004.

[SBF98] R. Studer, V.R. Benjamins, and D. Fensel. Knowledge Engineering: Principles and Methods. *Data & Knowledge Engineering*, 25(1-2):161–197, 1998.

[Sch07] B. Schnizler. *Resource allocation in the Grid. A Market Engineering Approach*. PhD thesis, Universität Karlsruhe (TH), 2007.

[SGL07] Amit P. Sheth, Karthik Gomadam, and Jon Lathem. SA-REST: Semantically Interoperable and Easier-to-Use Services and Mashups. *IEEE Internet Computing*, 11(6):91–94, 2007.

[Sho85] G.L. Shostack. Planning the Service Encounter. *The Service Encounter, Lexington Books, Lexington, MA*, pages 243–54, 1985.

[Smi82] V.L. Smith. Microeconomic Systems as an Experimental Science. *The American Economic Review*, pages 923–955, 1982.

[Smi89] C.W. Smith. *Auctions: The Social Construction of Value*. University of California Press, 1989.

[SMS$^+$02] A. Sahai, V. Machiraju, M. Sayal, A. Van Moorsel, F. Casati, and L.J. Jin. Automated SLA Monitoring for Web services. *Lecture Notes in Computer Science*, pages 28–41, 2002.

[SNP$^+$05] J. Shneidman, C. Ng, D.C. Parkes, A. AuYoung, A.C. Snoeren, A. Vahdat, and B. Chun. Why Markets Could (But DonŠt Currently) Solve Resource Allocation Problems in Systems. In *Proceedings of the 10th Conference on Hot Topics in Operating Systems*, pages 7–7, 2005.

[SSGL05] T. Sandholm, S. Suri, A. Gilpin, and D. Levine. CABOB: A Fast Optimal Algorithm for Winner Determination in Combinatorial Auctions. *Management Science*, 51(3):374–390, 2005.

[Sta79] T.M. Stanback. *Understanding the Service Economy: Employment, Productivity, Location*. Johns Hopkins Univserity Press, 1979.

[Ste04] F. Steiner. *Formation and Early Growth of Business Webs: Modular Product Systems in Network Markets*. Physica-Verlag Heidelberg, 2004.

[Sto09] Jochen Stoesser. *Market-Based Scheduling in Distributed Computing Systems*. PhD thesis, Universität Karlsruhe (TH), 2009.

[SV99] C. Shapiro and H.R. Varian. *Information Rules*. Harvard Business School Press Boston, Mass, 1999.

[Tal03] K. Talwar. The Price of Truth: Frugality in Truthful Mechanisms. *Lecture Notes in Computer Science*, pages 608–619, 2003.

[Tes01] L. Tesfatsion. Introduction to The Special Issue on Agent-Based Computational Economics. *Journal of Economic Dynamics and Control*, 25(3-4):281–293, 2001.

[Tho91] G. Thompson. *Markets, Hierarchies and Networks: The Coordination of Social Life*. Sage, 1991.

[TLT00] D. Tapscott, A. Lowy, and D. Ticoll. *Digital Capital: Harnessing the Power of Business Webs*. Harvard Business School Press, 2000.

[TW06] D. Tapscott and A.D. Williams. *Wikinomics: How Mass Collaboration Changes Everything*. Portfolio, 2006.

[Var09] H.R. Varian. Online Ad Auctions. *American Economic Review*, 2009.

[vHV07] E. van Heck and P. Vervest. Smart Business Networks: How the Network Wins. *Communications of the ACM*, 50(6):29–37, 2007.

[Vic61] William Vickrey. Counterspeculation, Auctions, and Competitive Sealed Tenders. *The Journal of Finance*, 16(1):8–37, 1961.

[VL04] S.L. Vargo and R.F. Lusch. Evolving to a New Dominant Logic for Marketing. *Journal of Marketing*, 68(1):1–17, 2004.

[VvHPP05] P. Vervest, E. van Heck, K. Preiss, and L.F. Pau. *Smart Business Networks*. Springer, 2005.

[Wal80] M. Walker. On the Nonexistence of a Dominant Strategy Mechanism for Making Optimal Public Decisions. *Econometrica: Journal of the Econometric Society*, pages 1521–1540, 1980.

[WCL+05] S. Weerawarana, F. Curbera, F. Leymann, T. Storey, and D.F. Ferguson. *Web Services Platform Architecture: SOAP, WSDL, WS-Policy,* *WS-Addressing, WS-BPEL, WS-Reliable Messaging and More*. Prentice Hall PTR Upper Saddle River, 2005.

[WD92] C.J.C.H. Watkins and P. Dayan. Q-Learning. *Machine learning*, 8(3):279–292, 1992.

[WHN03] C. Weinhardt, C. Holtmann, and D. Neumann. Market Engineering. *Wirtschaftsinformatik*, 45(6):635–640, 2003.

[Wil79] O.E. Williamson. Transaction-Cost Economics: The Governance of Contractual Relations. *The journal of Law and Economics*, 22(2):233, 1979.

[Win99] Dave Winer. Extensible Markup Language Remote Procedure Call (XML-RPC). Technical report, 7 1999. `http://www.xmlrpc.com/spec/`.

[Win02] A. Winter. Exchanging Graphs with GXL. *Lecture Notes in Computer Science*, pages 485–500, 2002.

[WNH06] C. Weinhardt, D. Neumann, and C. Holtmann. Computer-Aided Market Engineering. *Communications of the ACM*, 2006.

[WV03] Y. Wang and J. Vassileva. Trust and Reputation Model in Peer-to-Peer Networks. In *Proceedings of the 3rd International Conference on Peer-to-Peer Computing*, pages 150–157, 2003.

[YP00] J. Yang and M.P. Papazoglou. Interoperation Support for Electronic Business. *Communications of the ACM*, 43(6):39–47, 2000.

[ZBD+03] Liangzhao Zeng, Boualem Benatallah, Marlon Dumas, Jayant Kalagnanam, and Quan Z. Sheng. Quality Driven Web Services Composition. In *Proceedings of the 12th international conference on World Wide Web*, pages 411–421, New York, NY, USA, 2003. ACM.

[ZVB96] A. Zeithaml Valarie and M.J. Bitner. *Services Marketing*. 1996.