# IoS-based services, Platform Services, SLA and Models for the Internet of Services

Jorge Cardoso[1], Matthias Winkler[2], Konrad Voigt[2] and Henrike Berthold[2]

[1] University of Coimbra, Portugal
[2] SAP Research CEC, Chemnitzer Strasse 48, 01187 Dresden, Germany

**Abstract.** The Internet of Services (IoS) embraces new business and technological models that can radically change the way day-to-day services are provisioned and contracted. Within the IoS, services which were traditionally not created based on an established science and operated without a systematic approach, will be digitalized using proper methodologies, standards, and tools. The digital specification of services will enable their trading over the Internet using electronic marketplaces. This paper provides four main contributions to the IoS research: a definition and characterization for the concept of IoS-based service, the role and importance of platform services for the IoS, the challenges of managing SLA dependencies between IoS-based services in compositions, and a model-based approach for service engineering that can be used to design IoS-based services.

## 1 Introduction

Nowadays, all industrialized countries have become service-based economies in terms of the distribution of the people employed in the service sector [13]. While the first services were certainly delivered by humans to humans, the advances in computer systems over the past sixty years allowed computers to deliver services to humans. Information technologies (IT) have significantly contributed to the evolution of services. Over the years, each generation of innovation has created solutions that enable to automatically execute activities that were once done by human beings [10].

For example, Automated Teller Machines (ATM), introduced in the 70s, enabled banks to reduce costs by decreasing the need for tellers. Even with the expensive cost of IT in the late 70s and early 80s, the cost of automated processing with ATM was less than the expenditure of hiring and training a teller to carry out the same activity. In this case, a complex and specialized machine was developed to perform the same activity once executed by a human. As technology brought the automation to sophisticated machines, the number of workers required to execute many activities was gradually reduced. Nowadays, a broad spectrum of services is being replaced by automated machines. For example, undertaking a trip by train has traditionally required passengers to purchase a ticket from an office and show it for inspection when required by the train operator. As a response to technological development, automated dispensers have reduced the need to queue in order to purchase a ticket before a trip and, therefore, enable faster journeys.

The historical perspective and evolution of services has not only been confined to the use of machines to automate services. The emergence of the Internet, allied with the World Wide Web, has allowed a remote and generalized interaction between humans and computers. The technological developments in the late 90s have pushed the notion of *service* to *Web service* with the objective of supporting interoperable computer-to-computer interactions over a data network. This type of interaction required services to be autonomous and platform-independent, and needed services to be described, published, discovered and orchestrated using standard protocols for the purpose of building distributed solutions. The emphasis was on the definition of interfaces from a technical and programming perspective. The objective was on distributed systems communication, since Web services provide a technological solution to enable enterprise transaction systems, resource planning systems and customer management systems to be accessed programmatically through a digital network.

The IoS takes these developments one step further. So far, the use of services (i.e. Web services) has been restricted to IT professionals and to IT departments inside organizations. The Internet of Service targets to investigate and develop new theories, models, architectures and technologies to provide efficient and effective solutions that enable also non-professional users to create, trade and consume services. Furthermore, the notion of service is not limited to IT-based or technical services, but also to real world or day-to-day services. This larger spectrum of services immediately foresees a methodical study on how these services can be represented and modeled digitally.

This paper explores four important topics for the Internet of Services (these results are part of the output of the TEXO project which is part of the THESEUS program [3].) The first topic addresses the evolution of the concept of service, the economical value and the major characteristics of IoS-based services. These aspects are important in order to fully understand why the IoS and its services are fundamentally distinct from previous endeavors in the area of service provisioning. The second topic depicts a taxonomy for the various platform services that will be made available to providers and that will be used to design and operate IoS-based services. As the name indicates, platform services are provided by the underlying platform where IoS-based services are provisioned. The third topic tackles the support for managing SLA dependencies with a focus on IoS-based services that have a process composition in their backend. The violation of a particular SLA contract in a composition can affect related SLA and, therefore, dependencies and their impact need to be studied. Finally, the last topic covers the design of IoS-based services using a model-based approach. The design method proposed is part of the lifecycle of Service Engineering and relies on the integration of models that describe the various facets of a IoS-based service using efficient matching algorithms.

[3] http://theseus-programm.de/en-US/home/default.aspx

## 2 The Concept, Value and Characteristics of Services

Research in service science can often lead to confusion due to the simple fact that different communities use the same term to refer to conceptually distinct services. To avoid any ambiguity, this section will clarify key concepts associated with the term service upfront. The value that IoS-based services can bring to worldwide economies and the intrinsic characteristics of services when compared to products will also be reviewed.

### 2.1 The Concept of Service

Baida et al. [3] have identified that the terms *service*, *e-service* and *Web service* actually address related concepts from different domains such as computer science, information science and business science.

In computer science, the terms service, e-service and web service are generally used to identify an autonomous software component that is uniquely identified by a universal resource identifier (URI) and that can be accessed using standard Internet protocols, such as the simple object access protocol (SOAP) and the hypertext transfer protocol (HTTP), and languages such as the extensible markup language (XML). Hull et al. [6] have used the term e-service to describe the functionalities and characteristics of a Web service. They have defined an e-service as *a collection of network-resident software services accessible via standardized protocols, whose functionality can be automatically discovered and integrated into applications or composed to form more complex services.* In information science, services are a means of delivering value to customers by facilitating outcomes the customers want to achieve without the ownership of specific costs and risks [8]. Outcomes are created from the execution of tasks or activities under a set of constraints. In business science, a service is a set of intangible activities that generally take place in interactions between a provider and a consumer. The emphasis is not on standards, protocols or software, but on the study of how consumer experience with services can be evaluated and improved.

A deeper understanding of the terms associated to the concept of service is necessary in order to conceptually craft a common frame of reference. Such a shared understanding will help stakeholders involved in building enterprise wide solutions based on services for the IoS. Therefore, there is the need to identify the terms most often associated with the concept of service that have been introduced over time by the research community and by the industry. The four most relevant terms, illustrated in Figure 1, are real world services (day-to-day services), e-services, Web services and, more recently, IoS-based services. We introduce the term IoS-based service to differentiate a new type of service that will be represented with digital structures, specifications and standards to be traded on the Internet of Services.

**Real world service**. The term real world service (i.e., a day-to-day service or simply a service) is used to refer to any type of service that can be found in society. Because of their diversity and heterogeneity, real world services, have

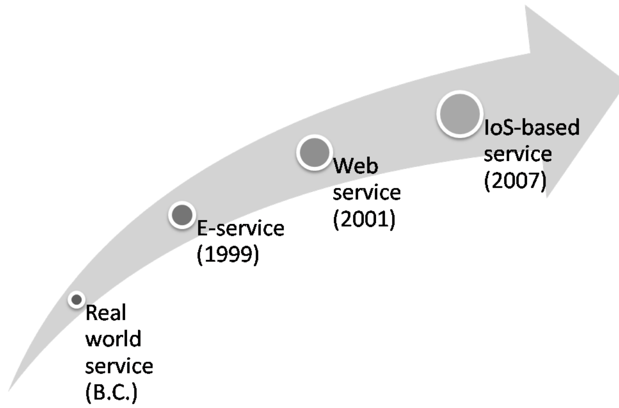


**Fig. 1.** Terms most frequently associated with the concept of service

traditionally been difficult to define. Kotler [7] defines a service as *any activity or benefit that one party can give to another that is essentially intangible and does not result in the ownership of anything*. Its production may or may not be tied to a physical product. For Payne [11], a service is an activity that has an element of intangibility associated with it and which involves the service provider's interaction either with the customers or with the property belonging to the customer.

**E-services**. E-services are services for which data networks, such as the Internet, are used as a channel that allow consumers to interact with remote services. Virtually any service can be transformed into an e-service if it can be invoked via a data network. E-services are independent of the specification language used to define its functionality, non-functional properties or interface. As with real world services, the definition of e-service is fairly broad. The main requirement is that the service must allow a remote invocation and interaction using a data network as a communication channel. The specification of a service from a business and technical perspective is not mandatory.

**Web services**. Web services allow software applications to easily communicate, independently of the underlying computing platform and language. The use of Web services is substantially less complex than the use of prior solutions for creating interoperable distributed systems. Heterogeneous, autonomous and distributed applications have been a vital field since computing shifted from jobs running on centralized mainframe computers to networked computers. Previous technologies that covered the same objectives as Web services included Remote Procedure Call (RPC), Common Object Request Broker Architecture (CORBA), Distributed Component Object Model (DCOM) and Java Remote Method Invocation (JRMI). These technologies had drawbacks that were considered significant when developing distributed application, such as incompatibility across vendors' implementations, and complexity and cost of solutions.

**IoS-based service**. The term "IoS-based service" is used to identify services provided through the Internet. Two main characteristics make IoS-based services distinct from previous services. On the one hand, this notion of service is not limited to IT-based services, but also to real world or day-to-day services. On the other hand, the stakeholders of such services, from the provisioning and consumption side, are not only IT professionals but also non-professional users. As a result, IoS-based services serve a dual purpose since they can be utilized directly by consumers, but they can also be invoked by technical systems to access business functionality which is provided remotely by business providers. An IoS-based service model defines a view on services that is provision-oriented and service-centric. An important feature of a digital service model is a separation of characteristics in terms of business, operational and technical perspectives.

Compared to previous approaches to support services – which were mainly implemented as pieces of software (e.g. Web services) – developing solutions for the IoS is more elaborate since real world services have very specific characteristics. While e-services and Web services are seen mainly as technological entities, the Internet of Services will take the representation of services one step further. IoS-based services will combine and correlate business, operational and IT aspects into service descriptions.

### 2.2 The Economical Value of Services

The intense competition of economies and the globalization of worldwide markets in conjunction with the generalization and expansion of IS and IT have opened up significant opportunities for the conception of new specialized services. Services are becoming quickly more productized. Providers are focusing on services for increased differentiation and creation of consumer value as a source of competitive advantage.

Recently, the concept of service has acquired a renewed importance since after several years of public debate, the European Parliament has approved the service directive [1]. This directive intends to enhance competition by removing restrictions on cross-border market access for services in Europe. The implications of this measure for businesses and the IT community are enormous since the service sector represents more than 70% of the Gross National Product and the directive can amplify the consumption of services in the European Union by 0.6% (37 billion Euros) [2]. Figure 2 illustrates the gross value added of services in Germany in 2005 provided by the Statistisches Bundesamt [4]. In Germany, the service sector represents 69.4% of the gross value added.

Services seem to be the new hub for most economies. Infrastructure services such as transportation and communication are fundamental building blocks

[4] http://www.destatis.de/

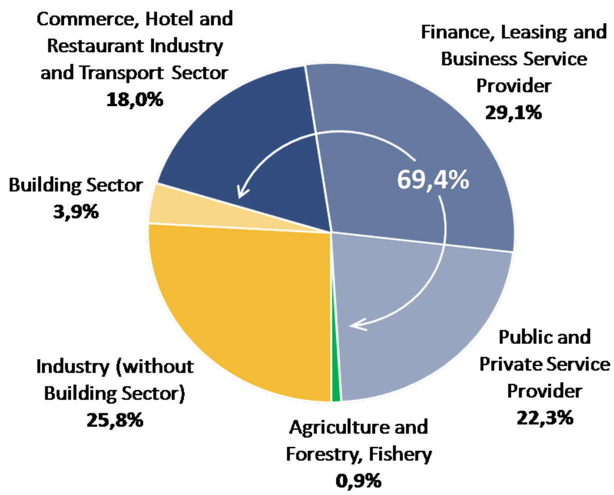


**Fig. 2.** Gross value added of services in Germany

which link to all other sectors. In most countries, one of the largest and most important providers of services is the government which operates in sectors such as water management, public safety and basic healthcare system.

Based on the economical value and importance of services, one question that immediately arises is how can the Internet provide a solution to create and enable a genuine market for the trade of cross-border services? Since the Internet is now an integral ingredient of the fabric of worldwide societies, economies and commerce, it can intuitively provide a fundamental infrastructure to enable the realization of the IoS.

### 2.3 Intrinsic Characteristics Services

Before proposing or building a solution for the IoS and for IoS-based services, it is fundamental to understand the nature of real world services since it is this type of services that will be digitalized and represented with proper models to enable their trading over the Internet. Real world services are often known to have one or more of the following characteristics: intangible, inseparable, immersive, bipolar, variable, ostensible, long-running, decoupled, perishable and qualitative.

Intangible (1). Services are intangible since they do not have a material existence. One can physically touch or view a product but most services are intangible. Nonetheless, it is often possible to see and evaluate the results that arise from the execution of a service. For eaxmple, it is not feasible to touch a legal advice or a doctor consultation. Therefore, it is difficult to create suitable specifications to model and to define attributes to objectively describe services. As such, research needs to be undertaken to determine which fundamental aspects

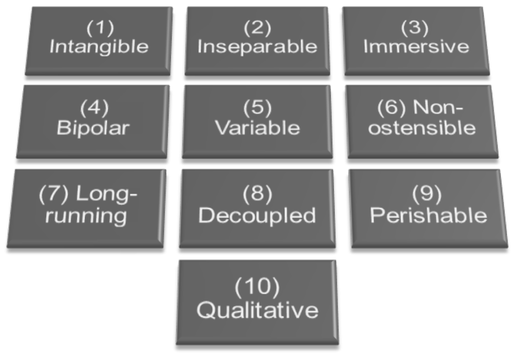


**Fig. 3.** Characteristics of services

and characteristics of real world services should be present in IoS-based service models and descriptions.

Inseparable (2). The provisioning and consumption of services occurs frequently in parallel. This implies that a rigorous match between supply and demand must be achieved. Otherwise, services are lost or consumers are queued and need to wait for service availability. This characteristic leads to a challenging research question: How can marketplaces of the IoS provide mechanisms to match between supply and demand efficiently?

Immersive (3). Services are often executed in collaboration, involving providers and consumers. This implies that in many cases it is difficult to determine the parties responsible for the degree of success or failure of a service. Therefore, when IoS-based services are executed, it is important to define service level agreements (SLA) that account for both parties. But in case of service provisioning failure, since many services are executed in collaboration, how can responsibilities be determined?

Bipolar or hybrid (4). Services are often executed by a blend of human and technological resources. While approaches to monitor purely technological resources are already available, solutions to monitor human involvement in services' execution and the complex relationship between the human and technological dimensions have not been studied in the context of highly distributed, autonomous and heterogeneous settings, such as the IoS. As a result, the following research question arises: How to create mechanisms that account for the monitoring of technological resources with the individual monitoring of human resources?

Variable (5). Products have a high degree of standardization, while services are very often tailor-made and are, therefore, heterogeneous. Organizations differentiate themselves in offering products and services, but the variations between similar products of different providers are less prominent than the variations between services. The quality and consistency of services is subject to a considerable variability since they are often delivered by humans. Human attitudes (such as behavior, cognitive skills, emotions, style, etc.) have a high variablity since they are difficult to control, manage and evaluate. For the IoS, this characteristic brings the need to devise new variability models for services, perhaps based on the variability models which have already been developed for products.

Ostensible ownership (6). The ownership between products and services is distinct. A stock can be called a financial product that the provider owns. A stock order may be placed by a consumer which might result in a transaction later on. When the transaction is completed, the ownership is transferred to the consumer. On the other hand, it is not possible to own a service. It possession is termed as an ostensible ownership. As a result, the IoS needs to enable providers to have the ostensible ownership of IoS-based services and enable them to remain in their control and management.

Long-running (7). Generally, the trading of products requires a low level of interaction between providers and consumers. For example, when a consumer buys a book at Amazon.com there is only one interaction point in time: the action of purchasing the book. In exceptional scenarios, the consumer may contact the provider in case of non-delivery of the product. On the other hand, services are often executed by a back-end business process which involves human interaction over time until the service is completed. For example, a service contracted to translate a book from German to English may run for several weeks and require a significant interaction between the translator and the writer. This brings humans, relationships between people, processes and activities to be an integral part of services. Therefore, IoS-based services need to account for the definition and representation of long-running business processes which include personal interactions (e.g., face-to-face or a telephone interaction) between providers and consumers.

Decoupled (8). A simplified lifecycle of a service includes generally five main phases: discovery, selection, invocation, execution and termination. In order to capture the full potential of services, consumers must have access to dynamic discovery mechanisms. Once a set of services is discovered, a selection is made and the selected service is invoked. Finally, the service is executed and terminates. These five phases can be carried out only with human involvement (humans add value in the form of labor, advice and skills), with a conjunction of humans and IT, or resorting purely on automated processing. Therefore, in the IoS, each phase is decoupled and may position itself anywhere in the spectrum of services executed solely by humans, on the one side, or purely automated on the other side. Here again, the representation of human and IT involvement needs to be equated when modeling IoS-based services.

Perishable (9). Since services have a propensity to be intangible, it is usually not possible to store them. As a consequence, services are perishable and unused capacity cannot be stored for future trade. For example, if a car is not sold today it can be sold tomorrow but spare seats on an airplane cannot be transferred to the next flight. Not being able to store services brings a challenge for electronic marketplaces since new management methods for service capacity are required.

Qualitative (10). In manufacturing industries, measures determining the quality of products are generally quantitative. On the other hand, the quality of a service is generally qualitative. The physical evidence or the tangible products that originate from service execution can provide indications that allow measuring a service quality. This characteristic is again a challenge for the IoS. How to identify which aspects of a service execution can be used to evaluate the quality of services quantitatively and qualitatively? Furthermore, the perceived service quality often results from consumers comparing the outcome of a service against what they expected to receive. Thus, how can consumers' expectations be properly captured and managed in the IoS?

All these intrinsic characteristics of services need to be explored and equated when real world services are digitalized into electronic models to be advertized in marketplaces and purchased remotely by consumers using the Internet.

## 3 Platform services

Platform services provide functionalities as a common infrastructure that can be used by IoS-based services in business value networks. We divide platform services into four groups: core business services, design support services, execution support services, and consumer support services. The elements of this classification are not orthogonal and one specific platform service can be classified under one or more types.

*Core business services* provide base functionalities required to complete a value-generating service to an IoS-based service. A value-generating service is the concrete service that generates a value for its consumer. An IoS-based service is a composite service that contains the value-generating service and core business services that are required to trade them. The set of core business services should contain services for payment, billing, security, and community rating. However, depending on the concrete service delivery platform, the set of these services can vary. The more core business services are available, the more business models can be supported. A core business service often provides a customized user interface that is integrated in the cockpit of service consumers. As a result, a consumer has to provide the required data for a core service to perform. For example, in a payment core service, the consumer must specify the kind of payment for the use of an IoS-based service.

*Execution support services* extend the functionality of the runtime environment. Their usage has to be specified at design time. Examples are adaptation services, monitoring services, and efficient data propagation services. An adaptation service automatically adapts an IoS-based service to changes in the runtime environment or consumer requirements without affecting the functional results. A monitoring service measures a number of runtime characteristics and can notify service consumers, service providers or platform providers about anomalies that occur at the technical and business level. An efficient data propagation service allows services that process large data volumes to exchange these data in an efficient way.

*Consumer support services* provide a functionality to retrieve services that fulfill the consumer's needs and requirements. Examples are services to search and select appropriate IoS-based services. Consumer support services can be accessed via the consumer cockpit . They use service description repositories for the search functionalities.

*Design support services* help the design of IoS-based services. They typically have a design component that supports business process modeling and a runtime component that runs as part of the overall process. Examples of design support services include message matching services and data integration services. The message matching service serves as an example of a platform service which is used to map complex message schemas between IoS-based services that form compositions. Therewith, it enables the composition of independently developed services offered on a service marketplace. The message matching service has two main components: the mapping component and the translation component. The mapping component is the design component. It automatically calculates a proposal for a mapping between two message schemas. A graphical user interface is provided for that component which presents a proposed mapping and supports its manual completion to a final mapping. The translation component is the runtime component. It uses the specified mapping to translate messages.

## 4 Managing Dependencies in IoS-based Service Compositions

One goal of the IoS vision is the creation of service compositions out of atomic IoS-based services provided by different service providers. The different atomic IoS-based services are composed in a way that they form more complex functionality, i.e. they are implicitly collaborating. These compositions are then offered to consumers via a marketplace. The provisioning of atomic as well as composite IoS-based services is regulated by service level agreements (SLA) which are negotiated between the service providers and consumers of the respective services.

An important challenge of this scenario is the management of such service compositions to ensure that the atomic services are working together in a proper way to achieve the overall goal of the composition. This management task is handled by the composite service provider who selects the atomic services and negotiates the SLAs with atomic service providers and composite service consumers. Managing service compositions is a challenging task due to the fact that the collaborating services have dependencies on each other. This leads to failure propagation. Also, the different SLAs need to be negotiated in a way which ensures proper collaboration between services. Changes to an SLA after the initial negotiation may require changes to other SLAs. In the following sections we outline the problem space of dependencies in service compositions and present our approach to managing these dependencies.

### 4.1 Introducing dependencies

A service dependency is a directed relation between services. It is expressed as a 1-to-n relationship where one service (dependant) depends on one or multiple services (antecedent). A service $S_1$ is dependent on a service $S_2$ if the provisioning of service $S_1$ is conditional to the provisioning of service $S_2$, e.g. if a property of service $S_1$ is affected by a property of $S_2$.

A service can be dependent on another service with regard to different aspects. A dependency occurs e.g. when one service provides something (e.g. data or goods) which is needed by another service to provide its functionality. A second example would be that the quality of service (QoS) of a composite service depends on the QoS of its atomic services. Dependencies occur either between atomic services in a process (horizontal dependencies) or between the composite service and atomic services (vertical dependencies) [15]. Horizontal dependencies mainly occur due to provider-consumer relationships or simultaneity constraints between services while vertical dependencies occur due to task-subtask relationships between the composite service and the atomic services [9]. Distinguishing the underlying causes of dependencies is important as they form the base for the dependency discovery strategy (see section 4.3).

### 4.2 Approach to managing dependencies

In order to manage service dependencies we developed an approach for capturing dependency information at design time in a dependency model and evaluating this model at runtime when problems occur or an SLA needs to be renegotiated. The approach follows a lifecycle of four steps: creation, validation, usage, retirement. During the first step (*creation*) the dependency information captured in a dependency model. Information about service dependencies is not available explicitly but rather implicitly in the process description and the related SLAs. To make it available explicitly for easier handling at runtime, the process description and SLA information are analyzed and dependency information is extracted and captured in a dependency model. The dependency model is then stored in a dependency repository. During the second step (*validation*) the dependency model is validated to check if the SLAs have been negotiated in a way that the services can successfully collaborate to achieve the composite service goals. During the third step (*usage*) the dependency model is evaluated at runtime in the context of occurring events such as SLA violations during monitoring or requests for SLA renegotiation. The goal of this evaluation is to determine effects of the current event on other services (i.e. if the SLA for service S1 is changed, which other services will be affected). Finally, during the fourth step (*retirement*) the dependency model is terminated once the composite service is terminated and the composite service SLA is expired. During this step the dependency model is removed from the dependency repository.

At the base of this approach there is a dependency model. A meta-model for capturing dependencies was developed for this purpose and a model editor for creating dependency model instances was implemented based on this meta-model [16]. While the model editor allows the full specification of dependency model instances, we also developed a model creation approach which partially automates this procedure.

### 4.3 Dependency model creation

The process of creating a dependency model is separated into two major steps. An initial model is generated automatically by an algorithm which analyses the process description and SLAs. In the second step the model can be refined by manual changes using a model editor. While the first step enables a more efficient creation process, the second step ensures that complex dependencies, which cannot be discovered automatically, can be included into the model. It also enables users to refine discovered dependencies.

The discovery algorithm takes the process description of the composite service and determines all valid paths from the start node to the end node. Next, the services within each path are checked for horizontal dependencies. The underlying assumption for this process is that services, which do not occur within a path do not have consumer-provider based dependencies. Synchronization constraints can occur also across paths, but they would have to be expressed explicitly since neither process description nor SLAs contain this information implicitly. Vertical dependencies are discovered by comparing the single services inside a path with the composite service. Dependencies regarding the QoS and price of services are not analyzed based on the created paths, but instead require a precise analysis of the process structure. QoS and price dependencies occur as 1-to-n relationships between the composite service and the atomic services. These dependencies are expressed as a function for calculating the respective composite value from the atomic values. The formula for composite value calculation is generated based on the process structure [4].

## 5 Models for the Internet of Services

The intrinsic complexity of IoS-based services requests for a new approach in Service Engineering (SE) and tools in developing such services [5]. Typically, services evolve in a common ecosystem in which organizations and IT provide value in form of services. SE provides methodologies to cope with the complexity of several business actors and their interaction. Furthermore, SE specifies tools for implementing and deploying services, covering both, IT and business perspectives.

Consequently, SE is a structured approach for creating a new service. It addresses two problems: 1) multiple stakeholders across different organizations and 2) different perspectives ranging from business to IT. To cope with these challenges we propose an integrated service engineering methodology and support by meta-model and model matching.

### 5.1 Integrated Service Engineering

For the development of IoS-based services we proposed the Integrated Service Engineering (ISE) methodology [5] and implemented it in the ISE workbench [12]. Thereby, we present a model-based approach; i. e. each aspect of a service is formally represented by a corresponding model. Figure 4 shows the ISE framework which is part of ISE methodology. Inspired by the Zachman framework and following the separation of concerns paradigm, it structures services into four main perspectives and five dimensions. These dimensions are: service description, workflow, data, people and rules. Each of these dimensions is divided into four perspectives (layers) of abstraction. These perspectives of the ISE methodology can be regarded as a phase in the development (model refinement) of services. Thus, the models which are assigned to each layer support the development from different viewpoints (i.e., scope, business, logical, and technical).

Additionally, models at different dimension but belonging to the same layer are bound to others in order to form the complete business service model at the respective level of abstraction. For all cells of the matrix, we have defined formal models which should be considered in the service development. Examples of models include UML, BPMN, BPEL, OWL, etc. Figure 4 presents the models selected and used in the ISE workbench.

Therefore, ISE acts as an integration platform for several models placed in cells of the framework. Throughout one dimension, models are created with respect to different views and refined until they conform to a technical specification. This leads to multiple representations of information on different layers of abstraction in the corresponding dimensions. Changes in one model have to be propagated (or at least detected) into related models holding overlapping information (depicted by arrows in Figure 4).

### 5.2 Support by Model matching

Multiple stakeholders and multiple perspectives result in several models designed in different ways. These models exhibit commonalities which need to be synchronized. This requires an integration of these models. The integration challenge is twofold: (1) one has to integrate the models by means of model transformation enabling an automatic synchronisation and (2) if a transformation is not available, one needs to identify commonalities. This covers use-cases such as duplicate

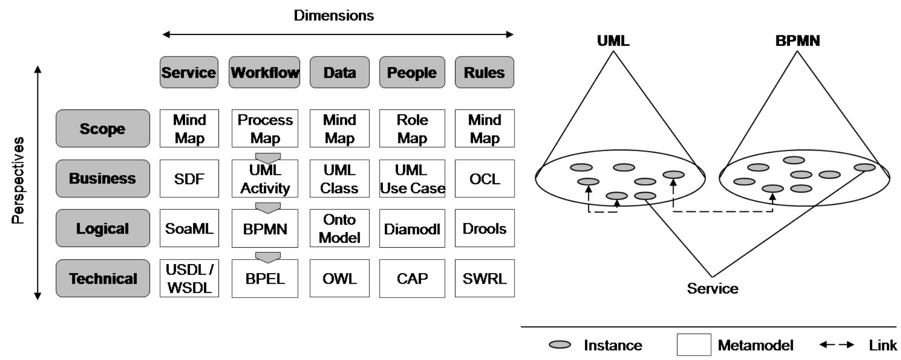


**Fig. 4.** The ISE models arranged in the corresponding matrix

detection as well as trace link creation. Thereby, having the trace link data available, common scenarios like change impact and orphaned elements analysis can be performed.

Figure 4 depicts a service, which is constituted of several models, each of them corresponding to a meta-model, i. e. a formalism representing a number of instances, e. g. a UML-class diagram or a BPMN process. These instances (models) have common features, since they represent the same services. For instance, a data object used in the BPMN has been modelled in a UML-class diagram. This information needs to be kept in sync in order to fulfill consistency. Nowadays, this is poorly supported by tools and if at all performed manually.

However, our approach has one major advantage; it is model-driven, which allows for a common formalism and therefore easy access to data in a closed world. To tackle the integration issue, we proposed to extend and adapt the aforementioned schema matching service, enabling it to consume meta-models [14]. We envision an extension allowing also a consumption of models, so correspondences can be discovered on meta-model *and* model level, thus performing meta-model *and* model matching in one system. This allows for a matching of BPMN and BPEL as well as BPMN and UML or any other meta-model. But we are not limited to these meta-models, but also support the reoccuring task of matching of concrete instances such as BPEL processes or Java classes. We name this approach *Layer Independent Matching*, since it is applicable to the meta and instance layer. Finally, one can match heterogeneous specifications, thus discover similarity (trace links) between different models like BPMN, BPEL, WSDL, USDL, etc as well as their concrete instances.

In model matching (instances) a bigger set of data is available compared to meta-model matching, so we feel that a stronger focus on structural heuristics is needed. Following that, we propose to apply graph edit distance algorithm taking advantage of planar graphs and using different clustering algorithms to cope with the increased dimension in size of models. For instance, a comparison between two concrete BPEL processes often contains more than 200 elements.

Assuming they are represented as formal models in a graph this can be extended (e.g. in Java classes) to more than 5000 nodes, comparing 5000 x 5000 nodes leads to 2.5 Mio nodes which requests for a clustering approach, thus reducing the dimensions of the problem to be matched.

## 6 Conclusions

In order for the Internet of Services to become a reality, numerous areas of research need to be (re)explored. From business science, contributions on new business models and pricing schema will be valuable. In the area of law and cyberlaw, new legal matters related to the provision and contracting aspects of IoS-based services supported by networked information devices and technologies will be required. From the area of social science, new community rating schema will be needed. The spectrum of research topics is substantial and sizable. In this paper we have centered our attention on four main topics: the notion and characteristics of IoS-based services, the characterization of platform services, the management of SLA contracts, and the design of complex IoS-based services. To correctly understand the notion of IoS-based services, an historical retrospective allied with a detailed identification of the specificities of day-to-day services that can be digitalized into the IoS are fundamental. The next topic presented platform services and introduced a taxonomy to better understand the type of platform services provided by marketplaces and provisioning platforms. Understanding the shared value-added contribution of an IoS-based service and the contribution of platform services is important to identify the focus of innovation and uniqueness. The third topic of study was the management of dependencies between services in compositions. We described an approach for the handling of dependencies at design and runtime. At its core it has a dependency model which is created by a semi-automatic approach of automatic discovery and additional modeling. Finally, the fourth topic described a structured and model-based approach to design and handle the intrinsic complexity of IoS-based services. Once individual models to describe a service are obtained, the challenge is to integrate the models using model matching and transformation. We presented a solution for supporting a semi-automatic matching of metamodels, models and instances using a Layer Independent Matching approach.

*Acknowledgements.* The TEXO project was funded by means of the German Federal Ministry of Economy and Technology under the promotional reference 01MQ07012. The authors take the responsibility for the contents. The information in this document is proprietary to the following Theseus Texo consortium members: SAP AG, empolis GmbH, intelligent views gmbh, ontoprise GmbH, Siemens AG, Fraunhofer Gesellschaft, FZI Forschungszentrum Informatik Karlsruhe, the German Research Center for Artificial Intelligence (DFKI GmbH), Technische Universität Darmstadt, Technische Universität Dresden, Technische Universität München and Universität Karlsruhe (TH). The information in this document is provided "as is", and no guarantee or warranty is given that the

information is fit for any particular purpose. The above referenced consortium members shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law.

## References


1. EU directive 2006/123/EC of the European parliament and of the council of 12 December 2006 on services in the internal market. Technical report, European Union, 2004.
2. Economic Assessment of the Barriers for the Internal Market for Services. Technical report, Copenhagen Economic, 2005.
3. Ziv Baida, Jaap Gordijn, Borys Omelayenko, and Hans Akkermans. A shared service terminology for online service provisioning. In Marijn Janssen, Henk G. Sol, and Rene W. Wagenaar, editors, *Proceedings of the Sixth International Conference on Electronic Commerce (ICEC04)*. ACM Press, 2004.
4. Jorge Cardoso, John Miller, Amit Sheth, and Jonathan Arnold. Quality of service for workflows and web service processes. *Journal of Web Semantics*, 1:281–308, 2004.
5. Jorge Cardoso, Konrad Voigt, and Matthias Winkler. Service engineering for the internet of services. In *Enterprise Information Systems*. Springer, 2008.
6. R. Hull, M. Benedikt, V. Christophides, and J. Su. E-services: a look behind the curtain. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 1–14. ACM Press, 2003.
7. Philip Kotler. *Marketing Professional Services*. Prentice Hall, Englewood Cliffs, 2002.
8. John Long. *ITIL Version 3 at a Glance: Information Quick Reference*. Springer Publishing Company, Incorporated, 2008.
9. Thomas W. Malone and Kevin Crowston. The interdisciplinary study of coordination. *ACM Computing Surveys*, 26(1):87–119, March 1994.
10. Richard R. Nelson. *Technology and global industry: Companies and nations in the world economy : Bruce R. Guile and Harvey Brooks (Eds.), (National Academy Press, Washington, D.C, 1987)*, volume 18. April 1989.
11. Adrian Payne. *Essence Services Marketing*. Pearson Education, Limited , 1993.
12. Gregor Scheithauer, Konrad Voigt, Veli Bicer, Matthias Heinrich, Anja Strunk, and Matthias Winkler. Integrated service engineering workbench: Service engineering for digital ecosystems. In *International ACM Conference on Management of Emergent Digital EcoSystems*, 2009.
13. Thomas F. Stafford and Carol S. Saunders. Moving towards chapter three. *e-Service Journal*, 3(1):3–5, Fall 2003.
14. Konrad Voigt. Towards combining model matchers for transformation development. In *Proceedings of 1st International Workshop on Future Trends of Model-Driven Development at ICEIS'09*, 2009.
15. Matthias Winkler and Alexander Schill. Towards dependency management in service compositions. In *Proceedings of the International Conference on e-Business*, pages 79 – 84, 2009.
16. Matthias Winkler, Christian Sell, Thomas Springer, and Alexander Schill. A dependency model for composite service management. To appear in Proceedings of the International Conference WWW/Internet 2009, November 2009.