An Analysis of Service Ontologies

Vikram Sorathia
University of Twente PO Box 217
7500 AE Enschede the Netherlands
v.s.sorathia@ewi.utwente.nl

Luís Ferreira Pires
University of Twente PO Box 217
7500 AE Enschede the Netherlands
l.ferreirapires@ewi.utwente.nl

Marten van Sinderen
University of Twente PO Box 217
7500 AE Enschede the Netherlands
m.j.vansinderen@ewi.utwente.nl

Abstract

Services are increasingly shaping the world's economic activity. Service provision and consumption have been profiting from advances in ICT, but the decentralization and heterogeneity of the involved service entities still pose engineering challenges. One of these challenges is to achieve semantic interoperability among these autonomous entities. Semantic web technology aims at addressing this challenge on a large scale, and has matured over the last years. This is evident from the various efforts reported in the literature in which service knowledge is represented in terms of ontologies developed either in individual research projects or in standardization bodies. This paper aims at analyzing the most relevant service ontologies available today for their suitability to cope with the service semantic interoperability challenge. We take the vision of the Internet of Services (IoS) as our motivation to identify the requirements for service ontologies. We adopt a formal approach to ontology design and evaluation in our analysis. We start by defining informal competency questions derived from a motivating scenario, and we identify relevant concepts and properties in service ontologies that match the formal ontological representation of these questions. We analyze the service ontologies with our concepts and questions, so that each ontology is positioned and evaluated according to its utility. The gaps we identify as the result of our analysis provide an indication of open challenges and future work.

Keywords: Services Science, Service Ontology, Knowledge Management, Quality of Service, Business Process.

Introduction

Services are dominating the economic activity across the globe already for some years (Chesbrough and Spohrer, 2006). The services industry involves entities that offer various kinds of consumer-specific services in different domains, like health, education, entertainment and so on. Developments in information and communication technology are also empowering services, by providing systems and environments that allow services to proliferate. Today, services can be consumed in many different scenarios, possibly involving multiple geographically scattered stakeholders and systems. However, this brings some challenges that follow from the heterogeneity of the entities involved in the service provision. Standardization efforts have solved some syntactic and system-level heterogeneity issues by defining standard protocols and data formats. Interoperability problems at semantic level are currently being addressed by semantic web technology, which is now starting to mature (Cardoso, 2007).

Semantic web technology makes use of ontologies to encode and reuse knowledge in various applications. This technology has emerged as an effective solution in heterogeneous collaboration environments. Many ontologies have been developed and are currently shared in order to provide a conceptualization of specific service aspects or domains. Some of these ontologies have been developed in rigorous academic exercises, while others have been established by a consensus-based community process. Reusable ontologies like Open cyc, SUMO, open EDI and LinkBase are quite popular nowadays (Cardoso et al., 2007). Among these, one can find ontologies defined specifically for the service domain (Hepp, 2008), targeted to solving service semantic interoperability issues.

The objective of this paper is to identify and analyze various ontologies proposed in the service domain for their utility to solve some of the challenges imposed by the geographical proliferation of services, especially the challenges related to semantic interoperability. A basic feature of an ontology is to provide a shared understanding of a conceptualization, so that the ontology can be reused without modification in different applications. However, many ontologies have been defined by different groups of people. These different ontologies may overlap but also contradict each other, while a set of ontologies for some domain may not cover all the required features of this domain, so that conceptual gaps may exist. This holds particularly for the service domain.

We follow a systematic method for analyzing service ontologies, by adopting the well-established formal approach to ontology design and evaluation described in (Uschold and Grüninger, 1996). Various service ontologies have been reported in the literature. We consider the service marketplace as identified in the emerging Internet of Services (IoS) (Cardoso et al., 2009) as our motivating scenario for selecting and analyzing ontologies. We identify competency questions and use them to determine the scope and coverage of each selected service ontology. The gaps in the coverage of the analyzed ontologies indicate the opportunities and areas for further work.

The rest of this paper is organized as follows. Section 'Background' discusses ontology engineering approaches and identifies some service ontologies available nowadays. Section 'Scenario' introduces the vision of a service marketplace as drawn in the Internet of Services (IoS). Section 'Review of Service Ontologies' introduces and discusses service ontologies according to the service marketplace vision and the questions we derive from this vision. Section 'Analysis' gives our analysis of the selected ontologies based on our service semantics classification schema. Section 'Results and Discussion' identifies conceptualization gaps and discusses the reusability of the currently available ontologies. Finally, Section 'Conclusions' reflects on the results of the paper and gives our conclusions.

DOI: 10.17705/1pais.02103

Background

In computer science, an ontology is normally defined as a formal specification of a shared conceptualization, where a conceptualization refers to state-of-affairs in the real world (Guarino, 1998). Accordingly, a service ontology should be a formal shared conceptualization of all aspects that are relevant for service provisioning. Since many different stakeholders are involved in service provisioning, capturing all these aspects can be quite challenging. However, if an accurate service ontology is obtained, it can be applied to solve many problems related to the heterogeneity that is inherent to service systems.

Due to the sheer amount of relevant service aspects that can be identified, it is neither feasible nor desirable to cover all these aspects in a single ontology. Therefore, in this paper we have assumed that a collection of (complementary) ontologies are applied in a modular and reusable way, so that together they cover all the relevant service aspects. These ontologies may also overlap in some aspects, which implies that they should be consistent whenever their overlap.

Alternatively, other conceptual modeling techniques could be used to represent service aspects instead of ontologies, like, for instance, conceptual graphs, UML class diagrams and object relational models. The biggest benefit of using ontologies is that they enable reasoning, so that assertions on a knowledge base can be processed at runtime. More recently, the semantic web community has developed ontology specification languages, like OWL, which allow ontologies to be coded as XML documents, facilitating their exchange, storage and processing. Efforts from the semantic web community also resulted in rich toolsets, most of them freely available, which are widely used by researchers and developers, fostering the usage of ontologies.

Ontology Engineering

Ontologies can be developed in many different ways. For example, taxonomies can be directly transformed into ontologies. A method for the automated transformation of product and service categories into an ontology is discussed in (Hepp, 2005). However, ontologies based on taxonomies have limited usefulness, as they do not capture the intricate interrelationships among the defined concepts (Dogac et al., 2002). These interrelationships and their implications may only be known to domain experts and closely associated stakeholders. Therefore, an ontology should be engineered with the help of experts and stakeholders from the target application domain (Åžensoy and Yolum, 2009).

A formal approach to ontology design and evaluation has been proposed by (Uschold and Grüninger, 1996). This well-established ontology engineering method prescribes that a motivating application scenario should be first identified, in which the proposed ontology is expected to be applied. Based on this scenario, a comprehensive list of so called informal competency questions should be defined. These questions must be answered by the ontology and are used to establish the terminology. A formal knowledge representation language is used to define the terminology, and a formal competency question is specified for each informal competency question defined before, based on the formal representation of the terminology. Formal axioms and definitions are given in order to complete the ontology. In addition, axioms and definitions should be justified in terms of theorems.

Formal languages are therefore necessary whenever an ontology is defined. The Web Ontology Language (OWL) is a standard XML-based language for representing ontologies (Bechhofer et al., 2004) that has been widely accepted and supported by the research community. OWL builds on the Resource Description Framework (RDF) (W3C, 2004b) and RDF Schema Language (W3C, 2004a). OWL is the most popular ontology description language nowadays, and is based on Description Logics, which is a family of logic-based knowledge representation formalisms (Baader, 2009).

Although a new version of OWL has been published recently (OWL 2), in this paper we consider the original OWL specification (W3C 2004) as reference. In this specification, OWL is described in terms of three sub-languages, namely OWL Lite, OWL DL and OWL Full. In this paper we normally refer to OWL DL, which imposes some limitations on the use of language constructs, but guarantees computational completeness and decidability.

We can characterize an ontology by considering the RDF and OWL elements used in its definition, determining in this way the language (features) used to define the ontology. The most basic representation is an attributive language (**AL**), which has features like atomic negation, concept intersection, universal restrictions and limited existential quantifications. Concepts in this language are defined using owl:Class, rdfs:subClassOf and owl:ObjectProperty language elements. The use of complex negation in an ontology representation language is indicated with symbol **C**, while **I** indicates the use of inverse properties, defined with owl:inverseOf. **H** indicates property hierarchy with terms defined using rdfs:subPropertyOf, **N** indicates number restrictions defined with owl:Cardinality, owl:minCardinality or owl:maxCardinality, **D** indicates use of data types, data values or data type properties defined with owl:DatatypeProperty, and **O** indicates nominals, enumerated classes or object value restrictions defined with owl:oneOf or owl:hasValue. We refrain from formalizing these features here, since the formalization of these features can already be found in the literature, like in, for example, (Horrocks and Sattler, 2001). Other features could be used to characterize ontologies, but we limit ourselves in this paper to the features often used in Description Logics, which are enough to cover the most popular languages used nowadays.

In general, ontologies with more features tend to have more expressivity and reasoning capabilities, while ontologies with a limited number of features tend to be more efficient to process. Therefore, when comparing ontologies based on their features we should not consider them as absolutely better or worse, since this depends on the context and tasks in which the ontologies are expected to be applied. An ontology designer should be aware of this trade-off between expressivity and processing efficiency when designing an ontology.

Service Ontologies

In the last years, considerable research effort has been spent on the definition of ontologies that cover different aspects of service provisioning. Some ontologies have been developed to determine the service (or product) category type of a given service (or product) instance, like industry classification schemes based on UNSPSC and NAICS. In addition to these schemes, the ecl@ss ontology is widely used across Europe and has an exhaustive list of categories. This ontology is employed to define offered business value and exchange of resources among various stakeholders. Another example is the Obelix (Ontology-Based Electronic Integration of Complex Products and Value Chains) service ontology (Akkermans et al., 2004), which addresses value aspects of the service industry. Ontologies are also employed to capture domain specific concepts and processes to support real-life business scenarios, like, e.g., in e-banking (Cobo et al., 2008).

In addition to real-world business services, ontologies are also widely used in computational services. Service descriptions can be semantically annotated using shared ontologies, thereby improving the discovery and selection process. Approaches for semantic annotation of service descriptions are discussed by (Sheth et al., 2008) and (Vitvar et al., 2007a). Accurate service discovery from large service pools (Bianchini et al., 2006) can be facilitated by using ontologies based on implicit properties.

A service description may include various functional and non-functional aspects of the service. Quality-of-Service (QoS) properties can be used as criteria for service selection. Ontologies that conceptually represent quality

https://aisel.aisnet.org/pajais/vol2/iss1/4
DOI: 10.17705/1pais.02103

aspects can be employed to enable quality-aware service discovery (Bleul and Zapf, 2007). From a consumer or a partner point-of-view, quality requirements need to be accurately specified for partner selection and to define Service Level Agreements (SLA). Ontologies can also be employed in QoS Requirement Specifications (Dobson *et al.*, 2005b).

From a system point-of-view, ontologies can be employed to achieve enterprise interoperability, in order to realize a semantically-enabled service-oriented architecture (Vitvar et al., 2007b). Inspired by the multi-disciplinary nature of services, a shared service terminology for business science, computer science and information science has been defined in (Baida et al., 2004).

Scenario

Since we follow the formal approach for ontology analysis of (Uschold and Grüninger 1996), we start with the identification of a motivating service application scenario. Below we give a concrete scenario in terms of a storyboard and after that we identify the elements of the service marketplace that should be covered by the service ontologies.

Scenario definition

Johan is the owner of a small hotel in Amsterdam. He started his hotel last year and was facing difficulties in manually handling customer identity verification, booking, billing and advertising, amongst other activities. He had problems in handling customers coming from all over the world, especially when these customers rescheduled their stay, asked for different means of payment, and demanded additional assistance, for instance, for renting a car or locating a tourist guide. He learned about the service marketplace that supports service providers and consumers in carrying out service activities, and, therefore, he decided to apply it to improve his service. He registered himself to the service marketplace as a hotel service provider and gave details on his offered facilities, booking conditions, rates, supported payment methods, accreditation and so on. With this new feature, he is getting more customers who use online portals to compare hotels based on various features and rates. He configured his reservation system to enable bookings from the marketplace, from travel agents and from self-service travel portals that do not individually assist customers. He started partnerships with providers that advertise and resell his services, and enable various payment gateways, customer reviews, and promotional package deals. In the service marketplace he discovered various service providers that can provide added-value to his service, like tourist guides, taxi services and car rental. These services can be combined with Johan's services whenever demanded by a customer. He is now able to get customer feedback and ratings to adjust his management strategy. Being part of service marketplace, he does not have to worry about customer identity verification and tracking, since his partner travel service providers, credit card companies, travel insurance companies and other providers in the marketplace use well established systems to validate customer identity. Since consumers can make necessary adjustments in travel plans, make the payments in advance, and specify preferences online at their ease, Johan is relieved from a considerable amount of manual effort that he had to invest. He now can focus on profitable partnerships and track customer reviews to promote his business.

This scenario illustrates that service systems are becoming increasingly complex, with stakeholders and systems that are not known *a priori*. A service provider can have multiple partners or enablers that provide specific services, so that this provider may also play the role of consumer at some point in time. In the scenario above this is illustrated by the interactions between Johan and the other service providers. Basic services can be used as component services in composite services. The same service can be offered by multiple providers, and it can also be consumed in various application scenarios. A complex business application may involve many atomic services, so that these services can be seen as offering fragments that are handled by autonomous and geographically dispersed entities. In order to realize meaningful service transactions, these entities must interact, process information, exchange resources and possibly perform actions that may have real world effects. Many transactions like these are increasingly being performed over the Internet, since available technology enables useful and accessible service offerings to be discovered and invoked. Authentication, payments, ratings and information services are common examples of these services.

In service systems, various entities form a cooperating network to offer, search, select, negotiate, operate, evaluate and regularize services. They utilize interoperable systems to interact and exchange resources, and their interactions may have real world effects. Hence, all these entities, their behavior and their interactions can be considered as a service ecosystem (Scheithauer et al., 2008) that supports complex service transactions, and may include service elements scattered across large geographical areas, each offering some specific functionality. In our scenario, Johan and his cooperating partners can be considered as parts of a service ecosystem.

The entities of a service ecosystem may fall into different business and governance domains, subject to different legal and community rules and requirements (Cardoso et al., 2009). Potential collaborators may adopt different approaches to negotiate, bill and charge, monitor and evaluate services. Each service entity can be engineered and deployed in different specific platforms, which may impose interoperability problems. These entities may have different notions of semantics for their terminology, which may hamper service selection, consumption and evaluation (Hepp, 2008). Since service experience depends on the expectations of each specific consumer, it is difficult to standardize service offers (Scheithauer et al., 2008). However, service capabilities should be identified, described and published in a public registry, allowing service offers to be discovered by potential consumers. Service capabilities can be encoded, published and discovered in many ways, so that the selection of a particular mechanism is a problem that has to be solved in the service ecosystem. Interoperability among service entities (people, information, hardware, software, environment and resources) should be guaranteed, otherwise various issues like functional mismatch, SLA mismatch and semantic mismatch can emerge (Wang and Xu, 2008).

Service marketplace

Since our intention is to realize an open service ecosystem, it is important to identify the context in which systems will operate. The Internet of Services (IoS) is a challenging context for research, and involves service engineering (Cardoso et al., 2009), services science (Ferrario and Guarino, 2009) and service computing (Schroth and Janner, 2007). IoS realizes a virtual marketplace for various service stakeholders to support their activities. Therefore, a basic requirement to realize this marketplace is that seamless service operations among multiple heterogeneous stakeholders are supported.

A service marketplace is an open service ecosystem in which multiple autonomous service providers can expose, advertise, manage, and perform service offerings. These providers can participate in complex services involving multiple partners. A single service operation may consist of a complex business workflow with possible real world effects. To support these conditions, service elements should be characterized, modeled and instantiated so that they can be processed manually or automatically during runtime. Therefore, a service marketplace should support the following features:

- Services are created, described, offered, consumed and monitored.
- Service events are tracked.
- Service regulations are implemented.
- User ratings and partner ratings are allowed.

DOI: 10.17705/1pais.02103

- Complex multi-dimensional criteria are applied to perform service search, rating and evaluation.

To enable the service marketplace vision necessary to support our motivational scenario, a comprehensive service ontology (or set of ontologies) should have the following characteristics:

- Unambiguous representation of service capabilities and features to support accurate discovery.
- Rules to realize complex business processes and workflows.
- Rules to automatically determine the quality, events and other characteristics of service instances from monitoring data.
- Transformation and resolution of terms from multiple representation schemes.
- Conceptual grounding to support interaction patterns for negotiation, agreement, fault handling and conflict resolutions.

Service Ontologies Review

According to the formal approach of ontology design and evaluation (Uschold and Grüninger 1996), an ontology is suitable for an application domain if it is able to provide answers to competency questions identified from a motivating application scenario. In our review we applied the scenario defined above and the service marketplace context to analyze the selected service ontologies.

Service ontologies can be evaluated based on their ability to answer the informal competency questions derived from a service usage scenario. We performed a comprehensive literature review, in which many ontology engineering efforts that address specific service aspects have been identified.

In our literature survey we started by searching for publications using keywords related to service ontologies. Our search strategy included publications from disciplines like computer science, business management and information systems. We considered journal papers, conference and workshop contributions, doctoral dissertations and standard specifications. Our selection has been mainly based on the number of citations of each particular reference. We selected 23 ontologies for our analysis.

To facilitate our analysis, we grouped the ontologies with similar objectives according to the aspects they capture as follows:

- Service concept: ontologies that cover the concept of service and its basic characteristics.
- Business: ontologies that cover the business aspects of services.
- Computing: ontologies that cover the computing (processing) aspects of services.
- Quality: ontologies that cover the qualitative aspects of service provisioning.
- Service types: ontologies that define classifications for service types.

We followed a systematic structure to analyze the ontologies identified in each group. Each ontology is reviewed below by stating its development objectives and the particular service aspects covered by the ontology. Important concepts and properties defined in the ontology are mentioned as examples. The expressivity of the ontology in terms of its language features is also indicated for some of the ontologies discussed in this section. To facilitate comparison, we list in a table the ontologies in each group, along with their development objectives and other features, such as the number of concepts and properties and the language features. The competency questions that apply to each group of ontologies are also given below. Appendix A provides an additional list of competency questions for each class of service semantics identified in Section ‘Analysis’.

Service Concept

We start our service ontologies review with ontologies that define the concept of service. Following the multi-disciplinary nature of services, Ferrario and Guarino defined the service concept from the services science point-of-view (Ferrario and Guarino, 2009). Based on a critical analysis of existing definitions of service in business management, computer science and related fields, a comprehensive definition of service is provided by using the concepts of *commitment*, *action*, *agent*, *triggering event*, along with dimensions of *time* and *location*. The proposal clearly distinguishes *service content* from *service commitment* and further characterizes a service process. This ontology identifies service roles like *consumer*, *trustee*, and *producer*, which can engage in various service activities like *discovery*, *activation*, *negotiation*, *monitoring* and *coordination*.

Among the basic service properties, the functional aspects of a service are fundamental for its potential consumption. A conceptualization that characterizes functional properties offered by a service is provided in (Oaks et al., 2003). Requirements related to the representation of functional properties of services are identified, and concepts like *action performed*, *inputs*, *preconditions* and *indirectly related objects* are proposed to fulfill these requirements. The proposal recommends that references to other ontologies and a classification of service capabilities in specific domains should be included. A conceptual metamodel for functional capabilities is provided using Object Role Modeling (ORM). This metamodel introduces service-related concepts like *Capability*, *Capability Parameter*, *Case Description*, *Signature*, and *Ontological Source*. Among these, the *Signature* of a capability represents specific input and output requirements of a service operation.

In addition to what an offered service is expected to do, we should also establish the qualitative (non-functional) properties of this service in order to allow fair evaluation and comparison with other service offerings. O'Sullivan provides a comprehensive conceptualization of non-functional service properties (O'Sullivan, 2006) to address this requirement. The *coverage* of the service is specified with help of the concepts *Temporal Entities* and *Locative Entities*. In order to establish the availability of a service, the concepts *Service Provision Availability* and *Service Request Availability* are introduced. In order to support real world business scenarios, the proposal also introduces non-functional service properties like *Obligation*, *Price*, *Payment*, *Reward Scheme*, *Discount*, *Penalty*, *Right*, *Trust, Quality* and *security*. Each of these properties is further defined in detailed conceptual models, represented using ORM.

An instance of service offering can be consumed in many ways. Wang and Xu identified various service elements and considered them as components (Wang and Xu, 2008). *People*, *information*, *resource* and *behavior* concepts are identified as service components that can be configured and utilized independently in order to offer a service. Each component uses and offers well-defined interfaces so that the interoperability required in diverse service consumption scenarios is guaranteed. In addition to basic service elements, the ontology introduces the concepts of *CapabilityMetrics, CapabilityRepresentation, ServiceValue, ServiceRisk, State* and *SLAParameter*. Interrelations among these concepts are defined in terms of additional properties. These components and their properties have been defined using OWL but unfortunately we could not find the corresponding *.owl* file.

The term service may have different meanings in different domains. Baida et

DOI: 10.17705/1pais.02103

al. introduced the terms *Service*, *Web Service* and *e-Service* to denote different notions of service from Business Science, Computer Science and Information Science, respectively (Baida et al., 2004). The authors argue that these three separate but closely related communities should reach a consensus on the basic service-related terms. They propose a terminology and an ontology to facilitate the shared and multi-disciplinary understanding of the key service-related terms.

Table 1 summarizes the five ontologies discussed above. We could not obtain the engineering artifacts (files, models, etc.) of these ontologies, so that an analysis of these ontologies in terms of complexity (number of concepts and properties) and expressivity (language features) has not been possible.

Table 1- Ontologies for the Service Concept

Ontology	Development goals
(Ferrario and Guarino, 2009)	Service concept
(Oaks et al., 2003)	Functional properties
(O'Sullivan, 2006)	Non-functional properties
(Baida et al., 2004)	Service multi-disciplinary definitions
(Wang and Xu, 2008)	Service elements as components

Concepts and properties represented in these ontologies allow the description of general service aspects like service offering, service availability, service coverage, cost, etc. and service properties in a given service scenario. Semantic queries on these ontologies should provide answers to the following competency questions:

CQ: What is a service?

CQ: What are the basic elements of a service?

CQ: What does a given service offer?

CQ: What are the components (e.g., people, information, resources) related to a given service instance?

CQ: What are non-functional properties of a service?

CQ: Who are the stakeholders of a given service?

Business Aspects

In a business environment, the provision of a service may involve resource intensive efforts of many related stakeholders. Furthermore, services can be restricted and governed by rules that ensure legitimate consumption, in order to avoid misuse and/or unintended service use. Stakeholders involved in a service offering may also be required to display some specific behavior during service provision. Therefore, business and organizational aspects of services should also be modeled in terms of concepts that cover organizational structures, roles, terms, conditions, agreement, partnerships and workflows.

A service offering may participate in multiple business consumption scenarios, so that the relation between a service offering and its potential business consumers and partners should be defined. This relation facilitates e-business and e-service provision, since it allows potential consumers and partners of a service offering to be found by querying an electronic catalog. The ecl@ss ontology (Hepp, 2006) aims at supporting e-business and defines a comprehensive set of product and service concepts along with their properties. This ontology helps manufacturers and service providers semantically annotate their offerings, enabling in this way the discovery of potential consumers for these offerings in manufacturing or service operations. This ontology is defined in OWL (file ecl@ss.owl). Concepts and properties in this ontology are represented using codes like, for example, *C_AKG697002* and *P_BAG548001*, respectively, and each concept and property is related to a human readable description defined with the rdfs:comment element. For example, the *C_AKG697002* concept is described as *Consulting service (training, further training)*. This ontology includes 76976 classes and 5525 properties. This ontology also defines data properties for product specification, like *operating volts*, *range*, *product dimensions*, etc., using the owl:DatatypeProperty element. Similarly, it also includes concepts and properties that cover various service industry operations. The ontology targets a vast range of products and services, organized in categories like food, packing material, chemical, service, medicine, etc. This ontology uses only basic language elements, and, therefore, its expressivity is **AL(D)**.

E-business applications require conceptualizations capable of expressing terms and conditions related to the availability, pricing, and other practical aspects of services. The developers of the ecl@ss ontology have acknowledged this requirement and defined the GoodRelations ontology (Hepp, 2008). This ontology describes common business terms in the form of web resources, legal entities, service offerings, prices, terms and conditions. This ontology has been developed by answering competency questions related to the location of service offers on the web, availability of services in spatial and temporal dimensions, eligibility of customers, payment options, delivery methods, and tax calculations. It also includes concepts like *ProductorServiceClass*, *QuantitativeValue* and property *hasWarrantyScope*,*hasUnitOfMeasurement* to match real world business requirements. The ontology also allows concepts to be related to a *MinValue* data type, so that the concept can have properties *hasMinValue* and *hasMaxValue*, thereby facilitating the specification of business terms. Following the formal approach for ontology engineering, in (Hepp, 2008) formal competency questions are introduced in the form of semantic queries. The ontology employs language elements to define property hierarchies and concept unions, hence its expressivity is **ALUH(D)**.

For some businesses an ontology-enabled approach can be extremely useful if concepts like goals of potential consumers, partners and enablers of services are represented. A multidimensional service ontology has been defined in (Orman, 2008) in which the structure, goal and target dimensions of a business can be represented. The ontology is illustrated with a vacation planning business scenario in which service entities like *flight, hotels, sights* etc. are defined, and the purpose of a potential service consumer is classified in a goal ontology as *sightseeing*, *family, shopping* and *relaxation*. Types of potential consumers are identified in the ontology by relating concepts properly. For example, a college student enrolled in a history department can be related to the goal of sightseeing historical venues, becoming a potential consumer of these sightseeing services. This ontology allows these relations to be defined and evaluated for particular persons and services.

The Obelix (Ontology-Based Electronic Integration of Complex Products and Value Chains) service ontology has been proposed based on the configurable nature of services (Akkermans et al., 2004). This ontology provides a conceptualization of the service *value* concept. The *Service element*, *service* *bundle* and *function* concepts are introduced to represent realistic service offerings and operations.

The concept of service bundle has been introduced to represent service compositions in different consumption scenarios. Service bundles are offered and consumed in business scenarios according to complex arrangements among various stakeholders. Therefore it should be possible to represent and reason about complex interactions and value exchanges among these bundles. The Serviguration ontology (Baida, 2006) addressed this issue by adding a configuration perspective to the work discussed in (Akkermans et al., 2004) and (Baida et al., 2004). This ontology covers the configuration of service components, by introducing concepts like service interface, input port and service link. This service ontology also defines the concepts of *service element*, *resource*, *service property*, *conditional output*, *pricing model* and *service port*. A configuration ontology defines the concepts of *component*, *resource*, *structural parameters*, *intrinsic constraint* and *port.* Intrinsic constraints, like *CoreEnhancing*, *CoreSupporting*, *OptionalBundle*, *Bundle* and *Substitute*, are introduced in order to allow the specification of complex offerings. This ontology is suitable to model complex service bundles offered in real world services. This ontology also facilitates the automatic identification of service bundles and service substitution scenarios.

Table 2 summarizes the five ontologies for business aspects discussed above.

In Table 2, information on the number of concepts and properties, and expressiveness has been omitted for some ontologies and indicated as N/A (not available). We have used N/A in our summary tables to indicate that either we could not find a formal encoding of an ontology or the ontology was encoded in a language unrelated to OWL.

The concepts and properties represented in these ontologies can be used to describe service bundles, functions, goals and value exchanges. These concepts enable knowledge representation of service scenarios with respect to these business aspects. Semantic queries on these ontologies should provide answers to the following competency questions:

CQ: What is the business value offered by a service?

CQ: Who are the partners of a given service instance?

CQ: What are the Service Level Agreement (SLA) parameters of a service?

CQ: Is a certain service an atomic offering or a service bundle?

Table 2 - Business Ontologies

Ontology	Concepts	Properties	DL Exp.	Focus
ecl@ss.owl	76976	5525	**AL(D)**	Industry classification with e-commerce
GoodRelations	30	58	**ALUH(D)**	e-commerce
(Orman, 2008)	N/A	N/A	N/A	Organizational aspects
OBELIX (Akkermans et al., 2004)	N/A	N/A	N/A	Bundling real-world services
Serviguration (Baida, 2006)	N/A	N/A	N/A	Configuration of service bundles
N/A: information not available				

Computational Aspects

Due to the increasing availability of ICT support, more business services are currently being offered as computational services. Similarly, an increasing number of common business functions to perform business transactions or handle information is being exposed as computational services. These systems are based on Service-Oriented Architecture (SOA) paradigm, in which business functions are encapsulated and accessed as (computational) services that may allow simple data sharing or may provide access to complex business transactions. Interoperable computational services accessed over a network are also known as web services.

Since many different and incompatible methods exist and could be used to access some computational function over the network, standardization bodies like W3C and OASIS have developed standards that help achieve interoperability among these service systems. SOAP, WSDL and UDDI have been defined as standards to invoke, describe and publish/find web services on the Internet, respectively. These standards are not really ontologies, but they provide standard vocabularies to grant access to computational services. A complex business process can be built by composing simpler component services, so that additional standards have been developed to describe and elicit business process as compositions of more elementary services. Additional business requirements have triggered the standardization process of some communication, transaction, security and trust aspects of web services. However, even with these standards it may be rather difficult to build complex business processes involving multiple separate (autonomous) entities due to semantic differences of these entities, i.e., simply because these entities may not mean the same concepts when referring to some term in a service description. Domain ontologies can be used to properly map business concepts onto computational concepts in order to achieve semantic interoperability, resulting in the so called semantic web services. Various approaches to realize semantic web services are discussed in (Sheth et al., 2008).

Web Service Modeling Ontology (WSMO) (Feier et al., 2005) is a development that provides a framework for the semantic description of goals and the functionality of Web Services. WSMO introduces the concepts of *WebService*, *goal*, *WSMO element, mediator*, *ontology* and *nonfunctional properties*. A mediator can be introduced between web services, goals or ontologies to cope with the differences between these elements in other to guarantee their interoperability. WSMO also defines the concepts of *function*, *instance*, *relation*, *value*, and *attribute* of web services, amongst others. WSMO addresses dynamic aspects by relating the concepts of *goal* and *interface* with the concepts of *Orchestration* and *Choreography*. WSMO ontologies are defined using the Web Service Modeling Language (WSML).

OWL-S (Martin et al., 2004) is an ontology that aims at providing semantic descriptions of services, particularly web services. OWL-S defines the concepts of service profile, service grounding and service model. A service profile further defines *process*, *service parameters*, *service category* and other related concepts. This part of the OWL-S ontology provides a conceptual model for the representation of processes (business workflows) by relating the concept of *process* to the concepts of *result*, *parameters, input, output,* and *participant*. A *simple* process can be realized with an *atomic* process, otherwise the process is a service composition and is defined as a *composite* process. Process workflows can be represented with constructs like *sequence*, *split*, *split-join*, *any order* and

DOI: 10.17705/1pais.02103

choice. OWL-S is defined in OWL and is represented in a set of *.owl* files. The expressivity of OWL-S is **ALCHOIN(D)**.

The Open Group (ToG) is engaged in the standardization of enterprise information architectures and governance-related issues and has proposed The Open Group Architecture Framework (ToGAF). ToG has addressed the semantic heterogeneity issues within SOA by proposing an ontology for the Service-Oriented Architecture (ToG, 2008). This ontology contains concepts like *Architecture Development Activity, Design Activity, Governance Activity, Implementation activity* and *service,* defined as sub-concepts of the *Activity* concept. This ontology also introduces the concepts of *Governance* and *GovernanceRegime* in order to cope with the requirements of service governance. The ontology defines a *hasComponent* property, with the sub-properties *hasDirectionActivity* and *hasInsfrastructure* so that computational aspects of a service element can be defined. The ontology consists of 50 concepts and 71 properties and uses various OWL language elements, so that its expressivity is **ALCHIN**.

OASIS proposed a reference ontology for semantic service-oriented architectures (OASIS, 2008). This ontology has been defined to support the specification of relationships among service elements. To cope with the static aspects of a service, the ontology defines the concepts of *service description*, *Goal description* and *capability description* and identifies mediation between them. Behavioral aspects are represented by the concept of *behavioral model* and the concepts of *orchestration* and *choreography*.

Some developments in the area of agent technology are also relevant for the semantic interoperability of services, since they target the automation of service operations like discovery, consumption and evaluation. The Web Service Agent Framework (WSAF) ontology (Maximilien and Singh, 2004) supports service discovery with the help of agents, by introducing concepts like *Service*, *ServiceAgent*, *AgentBehavior* and *ServiceDomain*. A service domain has sub-concepts *Computational*, *Business*, *Recreational* and *Government*. Since this ontology targets computational services implemented in an agent framework, it also describes interfaces and implementations with the *WsdlUri* concept defined with *implementation* and *interface* object properties. Quality is a major concern in agent architectures, so that the ontology also supports quality-related parameters. The expressivity of the WSAF ontology is **ALCIN(D)**.

Table 3 summarizes the five ontologies for the computational aspects of services discussed above.

Concepts and properties represented in these ontologies can be used to describe computational aspects of services, like description, discovery, components, interface, mediator, process, workflow, etc. Semantic queries on these ontologies should provide answers to the following competency questions:

CQ: Which goals are served by a certain service?

CQ: Which services are involved in the realization of a given business process?

CQ: Which activity is carried out for a given service artifact?

CQ: What computation is offered by a service?

CQ: What is the interface of a service?

CQ: What mediation systems are involved in the realization of a given service?

CQ: Under which governance regime does a given service operate?

Table 3 - Computational Ontologies

Ontology	Concepts	Properties	DL Exp.	Focus
ToG	50	71	**ALCHIN**	SOA interoperability
OASIS	N/A	N/A	N/A	SOA interoperability
OWL-S (*)	65	69	**ALCHOIN(D)**	Computational process
WSMO	N/A	N/A	N/A	Service mediation
WSAF	93	69	**ALCIN(D)**	Agent mediation

N/A: Information not available

* Numbers refer only to the process.owl ontology

Quality Aspects

Quality-of-service (QoS) properties are often used as criteria to select a service from a set of services with the same or similar functionality. QoS properties are non-functional properties of services, like, for example, performance, availability, reliability and costs, and can be measured or stated by the service providers or other entities on behalf of these providers. Potential service consumers may also describe the QoS of the services they are looking for when searching for appropriate service offers. There are many ways to represent service quality parameters, which can potentially hamper semantic interoperability. A possible solution to this problem is to represent QoS parameters in ontologies that not only address these possible representation schemes, but also conversions between them.

The QoSOnt ontology has been proposed in (Dobson et al., 2005a) to conceptualize quality aspects of services. This ontology introduces the *ServiceParameter* concept, with related concepts like *MeasureableAttribute*, *Metric* and *conversionRate*. The applicability of this ontology has been demonstrated with the SQRM (Service QoS Requirements Matcher) application, which supports web services discovery, differentiation and selection.

WS-QoSOnto (Tran, 2008) is another ontology that provides a comprehensive QoS model for web services. This ontology introduces the concepts of *Value Type*, *Impact Direction*, *QoS Dynamism* and *Valid Period*, in addition to the concepts defined in QoSOnt. It also introduces effect level, quality level, roles, constraints, QoS priority, QoS grouping and other similar concepts to support QoS monitoring and evaluation. This ontology also provides a taxonomy of quality properties that covers performance, availability, reliability and security aspects. The performance quality concept has sub-concepts *latency*, *throughout* and *response time*. Availability is further characterized by the sub-concepts of *MTTR*, *load balancing* and *Uptime*. Reliability has sub-concepts *Recoverable*, *consistency* and *messaging*. Security has sub-concepts *authentication*, *encryption* and *auditability*.

In case service instances are annotated with their QoS properties, these instances can be subject to automated selection based on quality criteria. Agent technology can be used to perform this automated task. An agent architecture that deals with the QoS of web services can benefit from an ontology, as demonstrated with the Web Service Agent Framework ontology (Maximilien and Singh, 2004). This ontology includes QoS concepts like *Quality*, *QAttribute, QMeasurement* and *QRelationship* in its

DOI: 10.17705/1pais.02103

QoS upper ontology. In its QoS middle ontology, it provides a detailed classification of *Quality*, with concepts like *Economic*, *Performance*, *Availability*, *Reliability*, *Security* and *Stability*. This ontology describes 93 concepts and 69 properties, and employs inverse properties, number restriction and data types, so that its expressivity is **ALCIN(D)**.

SL-Ontology (Bleul and Weise, 2005) represents service quality concepts at multiple levels of abstraction. At the lowest level of abstraction, the ontology defines the *service metric* concept, to enable the conversion of service parameters represented in different units of measure. A metric ontology defines concepts, units and data types that can be used by a service provider to specify the service quality metrics. The QoS ontology provides a conceptualization of service quality dimensions. Different service providers may use different identifiers to measure and define the same quality dimension, and the ontology defines mechanisms to solve this heterogeneity, by allowing the binding of individuals with external taxonomies. At the highest level of abstraction, the Service Level ontology defines the concepts of *service-level-offer*, *service-level-request* and *service-level-agreement*.

The ontologies discussed so far simply enumerate QoS parameters, whereas a QoS ontology should also provide conceptual definitions of QoS-related concepts. The Middle Quality Ontology (MQO) (Kim et al., 2007) addresses this problem and allows QoS parameters among existing ontologies to be modeled. MQO has been developed based on a motivational scenario and competency questions, and provides answers to questions related to QoS sampling plans, QoS measurements, tolerance, standard value, etc. It introduces concepts like *sample sizing, sampling plan, specification set, standard values,* and *Unit of measurement*, allowing not only the representation of QoS features, but the identification of concrete measures that are required to obtain QoS property values for a given service instance at any time. The concepts are organized in terms of a requirement ontology, a measurement ontology, a traceability ontology and a quality management system ontology. For example, the traceability ontology addresses the question ‘In which web services do problems arise?’ with the concepts of *tru* (traceable resource unit) and *primitive activity trace.*

Since many schemes for QoS are available, it has been necessary to standardize quality-related terms. To this end, the Foundation of Intelligent Physical Agent (FIPA) has developed the FIPA QoS ontology specification (FIPA, 2002). This ontology describes three types of QoS concepts, namely object, predicate and exception. Object descriptions include the concepts of *Rate Value*, *Time Value*, *Probability Value*, *Communication Channel*, etc. The predicate descriptions include concepts related to monitoring and time constraints. This ontology also includes a communicative act ontology that defines exception-related concepts like *not understood*, *refusal* and *failure*.

Table 4 summarizes the six ontologies that cover quality aspect of services discussed above.

The concepts and properties represented in these ontologies can be used to describe quality features of services, like quality parameters, measures, possible value ranges, etc. These concepts enable the knowledge representation of a given service scenario, allowing one to establish and assert the quality of a service instance. Semantic queries on these ontologies should provide answers to the following competency questions:

CQ: What types of QoS parameters are relevant for a given service type?

CQ: What are the QoS properties of a certain service instance?

CQ: What is the quality of a certain service?

CQ: What conversions are known for some given quality parameter?

CQ: What is the standard value of a quality parameter?

CQ: What is the agreed value of a service parameter?

CQ: What types of exceptions are possible for a service instance?

Table 4 -Service Quality Ontologies

Ontology	Concepts	Properties	DL Exp.	Focus
QoSOnt	N/A	N/A	N/A	Quality in service centric systems
WS-QoSOnt	N/A	N/A	N/A	Quality in Web Services
SL-Ontology	N/A	N/A	N/A	Service quality
FIPA	N/A	N/A	N/A	Quality standard
WSAF	93	69	**ALCIN(D)**	Agent mediation
MQO (Kim et al., 2007)	N/A	N/A	N/A	Conceptualization of quality parameters

N/A: information not available

Service Classification

In the service marketplace scenario it may be necessary to identify the category (application or business/industrial domain) to which a service belongs. A typical example is in case services from a certain domain have to comply with some specific policy or rules (e.g., all financial consulting services should be brought under a new taxation scheme). Therefore we need a categorization scheme to group all services belonging to a certain category. Some categorization schemes are available today, but the question is whether they are capable of covering all possible service categories in a service marketplace.

A service categorization forms a taxonomy, hence it is possible to derive an ontology from this categorization. For example, a methodology for deriving ontologies from existing product categorization standards is discussed in (Hepp, 2006). Many available product and services classifications like UNSPS, NAICS, eOTD, RosettaNet Technical Dictionary (RNTD) and eCl@ss are well-established schemes. Ontologies automatically derived from these categorizations can be evaluated based on specific metrics (Hepp et al., 2005) establishing their usability and relevance.

The North American Industry Classification System (NAICS) has been converted to an ontology according to this approach. The resulting NAICS.OWL ontology consists of 2341 concepts from the NAICS scheme (Mohr and Russell, 2002). This ontology is a simple industry classification, and describes a concepts hierarchy with expressivity **AL**. The UNSPSC.owl ontology was derived from the United Nations Standard Product and Service Code (UNSPSC) and describes 2548 concepts. ISIC.owl is another similar ontology based on a categorization and defines 538 Concepts. CPC.owl is yet another ontology derived from a categorization, and describes 3650 concepts with description logic expressivity **AL**. These ontologies are useful to assert whether a given product or service is an instance of a specific class defined in these standard schemes. Although these ontologies have limited expressivity and contain limited knowledge, they already provide useful information.

The eCl@ss.owl ontology (Hepp, 2006) copes with the limited expressivity of ontologies automatically derived from categorizations by providing a comprehensive reference in which 76976 concepts and 5525 properties are defined with expressivity **AL(D)**. eCl@ss.owl not only describes a product and service categorization, but also defines the basic properties of service offers and other related details. Therefore, this ontology has also been grouped with other business-related ontologies in subsection 'Business Aspects'.

Table 5 summarizes five ontologies proposed to cover the categorization of services according to their (industrial) domain.

Table 5 - Service industry classification ontology

Ontology	Concepts	Properties	DL Exp.	Focus
NAICS.owl	2341	0	**AL**	Industry classification
UNSPSC.owl	2548	0	**AL**	Industry classification
ISIC.owl	538	0	**AL**	Industry classification
CPC.owl	3650	0	**AL**	Industry classification
ecl@ss.owl	76976	5525	**AL(D)**	Industry classification with e-commerce

The concepts and properties defined in these ontologies can be used to determine the industrial domain of a given service, i.e. to determine whether this is an information service, logistics service, health service, etc. Semantic queries on these ontologies should provide answers to the following competency question:

CQ: Which standard service industry class a certain service belongs to?

Analysis

This section builds on the review of service ontologies discussed in section 'Service Ontologies Review'. In this section we identify gaps in the coverage of these ontologies and propose a classification schema for service semantics that aims at filling up these gaps. Finally we analyze the ontologies discussed before, indicating which areas of our classification schema are covered by which ontologies.

Conceptual gaps

The ontologies presented in section 'Service Ontologies Review' are capable of covering a wide range of aspects of service provisioning, but yet we identified some aspects in which these ontologies should be improved in order to enable the support of the service marketplace scenario that we are envisaging.

These ontologies fall short for what concerns the detailed characterization of service properties and consumption scenarios. These ontologies do not allow the characterization of legitimate service usage, and the representation of indirect value exchanges and relevant stakeholder's activities.

Issues related to business interactions, transactions, fault handling and market position should be handled in ontologies to allow the representation of market share, competition, customer behavior, service level agreements, negotiations, etc.

Service ontologies should also cover issues related to distributed trust, negotiation and user ratings. This would allow the representation of third-party ratings and open service platforms. Multi-disciplinary quality criteria should also be supported by service ontologies. This requires the representation of multi-disciplinary quality parameters and appropriate evaluation templates and would allow the modeling of comprehensive quality measures, transformations and multi-criteria quality analysis.

Finally, service ontologies should allow service offerings to be classified according to multiple categorization schemes. This requires a conceptual model that defines relationships (mappings) among these categorization schemes, which would allow a service instance categorized according to one scheme to be related to categories in other schemes.

Future work should focus on the development of ontologies (conceptual models) to bridge the gaps identified above.

Classification Schema for Service Semantics

In accordance with the formal ontology engineering approach that we have followed (Uschold and Grüninger, 1996), we derived informal competency questions from the service marketplace scenario. These questions provide guidelines to knowledge engineers, by identifying the semantic queries that the ontologies must resolve in order to successfully realize their development goals. In the particular case of the service marketplace, these questions should cover many different concerns, as it can be asserted in our review of relevant service-related ontologies. Keeping these questions together would result in a large and unmanageable list, so that we decided to apply separation of concerns and classify these competency questions in groups.

Competency questions for the service marketplace should address the concerns of the different stakeholders operating in different scenarios. For instance, service brokers, consumers, evaluators, auditors and engineers possibly have distinct viewpoints of the same services. These stakeholders determine, assert and utilize different service properties depending on their particular interests, authority, expertise and tool support. Questions that are useful for a stakeholder may not be useful to others. For example, system level aspects of the service may not be useful to partners, regulators or domain experts. Therefore, we group competency questions based on the focus of the service stakeholders.

Similarly, all identified conceptual gaps in the existing ontologies can also be related to some stakeholders to which they concern. For example, compliance to standards is a concern of a business stakeholder participating in a governance board. Hence, concepts related to compliance to standards are classified as governance semantics. Similarly, gaps related to market positioning and negotiations can be related to other business stakeholder activities and, therefore, they can be classified separately. In this way, we split all original groups further into more detailed groups. Table 6 indicates our grouping of service aspects with newly identified classes of service semantics.

The service semantic classes identified are briefly introduced below. Appendix A provides a set of informal competency questions for each class.

Service Domain

Domain aspects of services include the concept of service itself and related properties like service value, stakeholder, cost, benefit, etc. Establishment of service-related theories, definitions, best practices, and case studies are typical examples of activities related to the multi-disciplinary domain of services science (Spohrer et al., 2008). Theories for the description of service elements determine the service domain semantics.

DOI: 10.17705/1pais.02103

Table 6 - Classes of service semantics based on separation of concerns

Cluster	Issues and Conceptual gaps	Classification
Service Concept	Indirect value exchanges, Service elements identification	Service Domain
Business	Business Logic	Functional
	Market position, Stakeholder behavior	Situational
	Policy, Compliance	Governance
	Service organization structure, Contracts and agreements, Negotiations	Business Management & Organizational
Computation	Implementation logic, Execution environment	ICT Infrastructural
	Exceptions, fault, legitimate use, Third party rating	Transactional
Quality	Multi-criteria quality parameters	Non-Functional
Classification	Multi-classification scheme	Taxonomical

Functional

Functional aspects of services are concerned with the details of the functionality of services, which can be business functions, computations or value-creating actions. The starting point, end point, intermediate stages, resource requirements, human intervention, conditional selection, status and associated events are some of the issues related to the service functionality. These aspects can be considered as the functional semantics of services, and includes questions related to the logic and workflow that governs the service operation.

Situational

A service offering can be affected by the activities of various stakeholders other than the provider, like the behavior of some relevant external entities or situations in the application environment of the service. Legal entities, standardization bodies, competitors, consumers and domain experts are examples of entities that can indirectly affect the service. These aspects are captured in the situational semantics of services, and enable the handling of these activities and the interpretation of their impact on a given service instance.

Governance

Service governance aspects deal with identifying, setting, implementing and updating the policies related to services. Service governance is closely related to service lifecycle and service coverage, and is also relevant to establish the compliance with standards. Governance semantics should capture these aspects in order to infer the governance status of a given service instance.

Business Management and Organizational

Services can be offered as parts of a business process, which has a definite outcome in real world once it is executed. Depending upon the desired effect of this process, many roles like operators, enablers, partners, and managers may be needed. Therefore, organizational and business management aspects that are relevant for a service should also be addressed. These aspects can be considered as business management and organization semantics of services.

ICT Infrastructural

Some services may benefit from supporting systems whereas others are completely offered in a system environment. All computational services are realized, shared and managed in a system environment. E-services and many IT enabled services have different levels of IT support in service operation and management. Hence, it is necessary to specifically model system aspects that deal with various system-level properties of a given service instance. These aspects can be considered as the ICT infrastructural semantics of the service.

Transactional

Services can be parts of business transactions with specific outcomes. A service provider may simultaneously perform multiple transactions at a given point in time. Monitoring and tracking of transactions is therefore important to establish whether these transactions are performed successfully. These aspects can be considered as the transactional semantics of services. This aspect includes the tracking of special events, exceptions, faults and the violation of service level agreements.

Non-Functional

Non-functional aspects of services cover all properties related to performance, reliability, costs, payments, coverage, etc. These aspects can be considered as the non-functional semantics of services and address competency questions related to quality, coverage, terms, conditions and modes of value exchange.

Taxonomical

Since many services offer different values in the service marketplace, it is necessary to categorize these service offerings. Many viewpoints share this concern. Service categorization may enable the application of policy, tax schemes, or other special rules to services from some specific category. It also enables analysis and handling of service usage logs to identify specific trends. These aspects can be considered as the taxonomical semantics of services.

Coverage of available ontologies

The groups of service semantics introduced above indicate specific service dimensions can be considered for conceptualizations suitable for the service marketplace. Concepts in these dimensions can be identified, instantiated and processed by service stakeholders in different service life-cycle phases in an open service ecosystem. Yet, all these aspects need to be considered when developing a comprehensive service ontology (or set of ontologies).

Table 7 indicates the coverage of existing service ontologies with respect to identified service semantics. Each ontology has been discussed individually with its relation to identified informal competency questions. In Table 7, the ontologies are related with the groups of competency questions according to the concerns we identified before. Table 7 identifies the focus of each ontology development effort and shows that no ontology completely covers all the identified types of service semantics. In other words, none of the ontologies we considered in our survey covers every service aspects. Hence, in order to realize the service marketplace, the missing aspects must be included in future versions of the existing ontologies or in the new ontologies to come.

Results and Discussion

Our survey exposes the overwhelming number of service-related ontologies that can be found today. Table 7 reveals that the existing ontology efforts present clear gaps in some specific service aspects. Transactional aspects of services should be considered as a basic requirement of platforms to support an open service ecosystem. Pool of competitors, potential consumers, standards organizations and other relevant entities may

DOI: 10.17705/1pais.02103

continuously exhibit behavior that is should be tracked and adopted for the overall success of a service. Hence these situational aspects should be appropriately covered by service ontologies. The emerging multi-disciplinary field of services science also intends to establish new insights, aiming at improving the capabilities of services. Service ontologies must therefore be improved and extended with domain knowledge from the services science research community. However, some existing service ontologies provide a sufficiently strong foundation, by addressing other specific aspects of service that can be instrumental in realizing the service marketplace vision.

Figure 1 indicates how some of the ontologies discussed in this paper could be used together to form a foundation for addressing most of the relevant service aspects.

Table 7- Semantic Coverage of Service Ontologies

Ontology	Service Domain	Functional	Non-Functional	Taxonomical	Bus. Mgt .& Organization	ICT Infra.	Transactional	Situational	Governance
(Ferrario and Guarino, 2009)	Y		Y						
(Oaks et al., 2003)			Y						
(O'Sullivan, 2006)			Y						
(Baida et al., 2004)			Y						
(Wang and Xu, 2008)			Y			Y			
ecl@ss.owl (Hepp, 2006)			Y	Y	Y				
GoodRelations (Hepp, 2008)					Y	Y			
(Orman, 2008)					Y				
ToG (ToG, 2008)						Y			Y
OASIS (OASIS, 2008)						Y			
OWL-S (Martin et al., 2004)		Y				Y			
WSMO (Feier et al., 2005)		Y				Y			
WSAF (Maximilien and Singh, 2004)			Y			Y			
WSMO-lite (Vitvar et al., 2007a)						Y			
QoSOnt			Y						
WS-QoSOnt			Y						
SL-Ontology			Y						
FIPA (FIPA, 2002)			Y						
(Kim et al., 2007)			Y						
NAICS.owl				Y					
UNSPSC.owl				Y					
ISIC.owl				Y					
CPC.owl				Y					

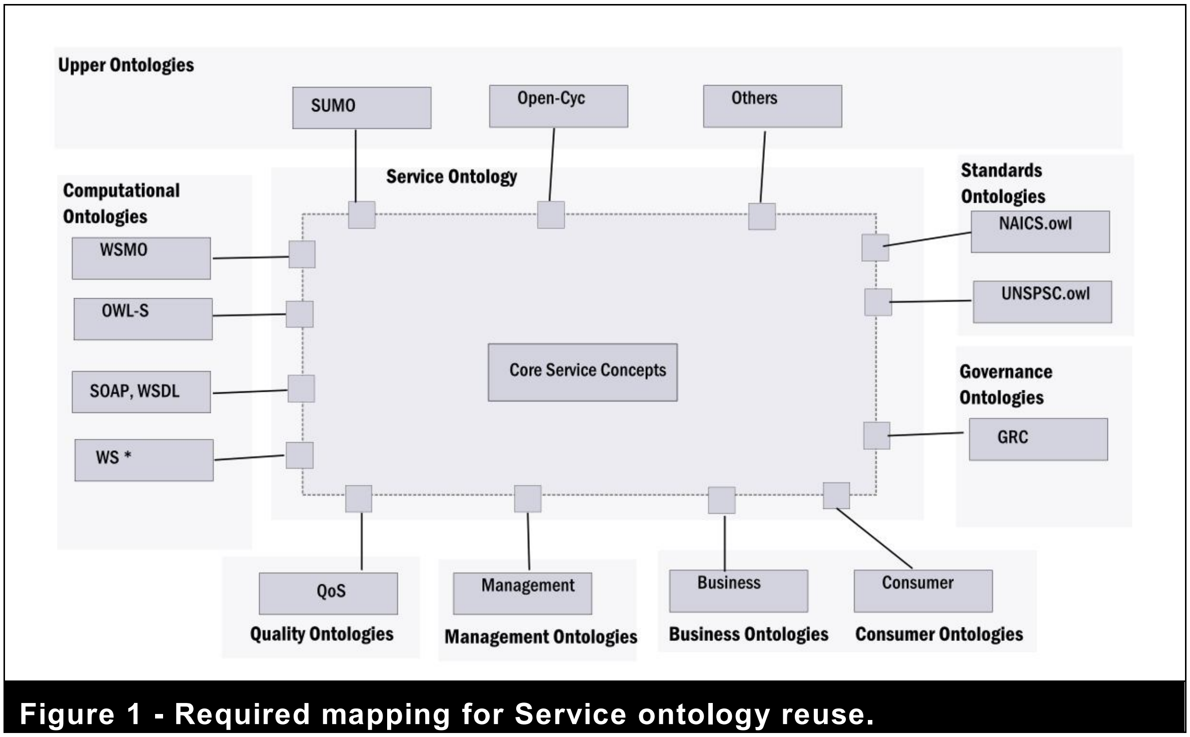

Figure 1 - Required mapping for Service ontology reuse.

Some of the discussed ontologies describe up to few thousands concepts in a single encoding. The expressivity of these ontologies ranges from relatively simple (**AL**) up to highly expressive (**ALCHOIN(D)**). The different number of concepts and expressivity can be a problem once these ontologies are used in combination. In addition to the service concepts, the general features related to physical properties, measurement schemes and abstract concepts like environment, events and situation are also useful in knowledge representation. These concepts are encoded and shared as upper ontologies (Cardoso et al., 2007). Some of these ontologies are based on and closely related to standardization processes and, therefore, can be directly utilized by all systems that comply with these standards. Some ontologies are derived as a result of comprehensive research work that can be reused.

Based on these facts, we conclude that many important service aspects are already represented in ontologies. Hence, a service ontology suitable for service ecosystems should provide the core conceptualization, thereby creating the conditions to fill the gaps identified in our analysis. For the conceptualization of the specific service aspects, mappings to existing ontologies can be provided. The repetition of scholarly effort can be avoided, allowing reuse and compliance to existing standards and terminologies. To enable the reuse of ontological representation, an appropriate ontology engineering strategy is required to solve issues related to updates, consistency checking and conflict resolution. A proper knowledge management infrastructure to support annotation, query resolution and inference should also be established.

Conclusions

Services are recognized for their important share in current economies, along with issues that emerge from their proliferation. Semantic web technology is an enabling technology capable of addressing some of these issues. Knowledge represented and shared in the form of ontologies is considered as

DOI: 10.17705/1pais.02103

an important development towards the solution of interoperability problems. This paper analyzed service ontologies developed from different perspectives, like business, management, information science, computation and standardization, in order to establish their suitability for addressing the issues identified in the service marketplace scenario. The analysis has also aimed to clearly determine the reuse potential of existing work, thereby avoiding unnecessary repetition. Wherever possible, we discussed the salient features of these ontologies, like their expressivity and conceptualization concerns. Service provisioning issues in a marketplace scenario revealed gaps in the conceptualization offered by the whole set of studied ontologies. From these gaps we identified topics for future work. The paper also gives a comprehensive literature review, providing insight in the state-of-the-art in service ontologies.

We followed a formal and well-established method to analyze existing ontologies. However, we defined competency questions and classes of service semantics based on our judgment, and we applied them to the service ontologies based on our understanding of these ontologies from their sources in the literature. Since we properly justify our choices and interpretations, we are convinced that our comprehensive survey and analysis is a solid contribution to this field.

The number of concepts and properties, and the features of ontologies we have shown in our survey are quantifiable and comparable, and give us measures of their complexity (expressiveness and size). However, these measures are not related to the needs of service analysts, architects and developers, and therefore they have limited practical implications. In contrast, the competency questions are strongly related to the practical needs of these practitioners, and by assessing whether ontologies are capable of answering these questions we can get insight in their coverage and suitability. We believe that these are the most valuable results of our analysis.

In this work we selected ontologies that have been explicitly defined as such. However, some tools or components like service registries or management software may be based on conceptual models that can be considered as ‘implicit ontologies’. An interesting extension of our work would be to identify and include these tools in the survey.

References

Akkermans, H., Baida, Z., Gordijn, J., Pena, N., Altuna, A. & Laresgoiti, I. (2004). “Value Webs: Using Ontologies to Bundle Real-World Services”, *IEEE Intelligent Systems,* 19 (4), pp. 57-66.

Åžensoy, M. & Yolum, P. n. (2009). “Evolving service semantics cooperatively: a consumer-driven approach”, *Autonomous Agents and Multi-Agent Systems,* 18(3), pp. 526-555.

Baader, F. (2009). ”Description Logics”, Vol. 5689, pp. 1-39: Springer-Verlag.

Baida, Z. (2006). ”Software-aided Service Bundling Intelligent Methods & Tools for Graphical Service Modeling”, *PhD Thesis*, Vrije Universiteit Amsterdam.

Baida, Z., Gordijn, J. & Omelayenko, B. (2004). “A shared service terminology for online service provisioning”, In *ICEC '04: Proceedings of the 6th international conference on*

Electronic commerce, pp. 1-10, New York, NY, USA: ACM.

Bechhofer, S., Harmelen, F. v., Hendler, J., Horrocks, I., McGuinness, D. L., Patel-Schneider, P. F. & Stein, L. A. (2004). “OWL Web Ontology Language Reference”, *W3C Recommendation*. World Wide Web Consortium (W3C). [Online] http://www.w3.org/TR/owl-ref/

Bianchini, D., Antonellis, V. D., Pernici, B. & Plebani, P. (2006). “Ontology-based methodology for e-service discovery”, *Information Systems,* 31 (4-5), pp. 361 - 380.

Bleul, S. & Weise, T. (2005). “An Ontology for Quality-Aware Service Discovery”, *First International Workshop on Engineering Service Compositions (WESC'05)*, pp. 35--42.

Bleul, S. & Zapf, M. (2007). “Ontology-Based Self-Organization in Service-Oriented Architectures”, In *2. Workshop Selbst-organisierende, adaptive, kontextsensitive verteilte Systeme (SAKS)* Bern, Swiss: VDE.

Cardoso, J. (2007). “The semantic web vision: Where are we?”, *IEEE Intelligent Systems,* 22 (5), pp. 84-88.

Cardoso, J., Lytras, M. D. & Hepp, M. (2007). “The Future Of The Semantic Web For Enterprises”, In *The Semantic Web: Real-World Applications from Industry*, Vol. 6, pp 3-15 (Eds J. Cardoso, M. Hepp and M. D. Lytras). Springer.

Cardoso, J., Voigt, K. & Winkler, M. (2009). “Service Engineering for The Internet of Services”, In *Enterprise Information Systems*: Springer.

Chesbrough, H. & Spohrer, J. (2006). “A research manifesto for services science”, *Commun. ACM,* 49 (7): pp. 35-40.

Cobo, J., Manuel L., Losada, S., Cicurel, L., Bas, J. e., Luis, Bellido, S. & Benjamins, V. R. (2008). “Ontology Management in e-Banking Applications”, In *Ontology Management*, Vol. 7, pp 229-244 (Eds M. Hepp, P. D. Leenheer, A. de Moor and Y. Sure). Springer.

Dobson, G., Lock, R. & Sommerville, I. (2005a). “QoSOnt: a QoS Ontology for Service-Centric Systems”, In *EUROMICRO '05: Proceedings of the 31st EUROMICRO Conference on Software Engineering and Advanced Applications*, pp 80--87 Washington, DC, USA: IEEE Computer Society.

Dobson, G., Lock, R. & Sommerville, I. (2005b). ”Quality of service requirements specification using an ontology”, In *Proceedings of International Workshop on Service-Oriented Computing: Consequences for Engineering Requirements (SOCCER'05)* Paris, 30th August 2005.

Dogac, A., Laleci, G., Kabak, Y. & Cingil, I. (2002). “ExploitingWeb Service Semantics: Taxonomies vs. Ontologies”, *The Computing Research Repository (CoRR)* cs.DB/0212051.

Feier, C., Arroyo, S., Cimpian, E., Domingue, J., Fensel, D., Kšnig-Ries, B., Lausen, H., Polleres, A. & Stollberg, M. (2005). “Web Service Modeling Ontology Primer,” *W3C Member Submission*, World Wide Web Consortium (W3C). [Online] http://www.w3.org/Submission/WSMO-primer/

Ferrario, R. & Guarino, N. (2009). “Towards an Ontological Foundation for Services Science,” In *Future Internet – FIS 2008*, pp 152-169. Springer Berlin / Heidelberg

FIPA (2002). “FIPA Quality of Service Ontology Specification”, *Standard* The Foundation of Intelligent Physical Agents (FIPA). [Online] http://www.fipa.org/specs/fipa00094/SC00094A.pdf

Guarino, N. (1998). “Formal Ontology and Information Systems”, in *Formal Ontology in Information Systems. Proceedings of FOIS’98*, Trento, Italy, 6-8 June 1998. Amsterdam, IOS Press, pp. 3-15.

Hepp, M. (2005). “A Methodology for Deriving OWL Ontologies from Products and Services Categorization Standards”. In *Proceedings of the 13th European Conference on Information Systems* (ECIS2005), May 26 - May 28, 2005, Regensburg, Germany, pp. 1-12.

Hepp, M. (2006). “Products and Services Ontologies: A Methodology for Deriving OWL Ontologies from Industrial Categorization Standards”, *Int. J. Semantic Web Inf. Sys.,* 2 (1), pp 72-99.

Hepp, M. (2008). “GoodRelations: An Ontology for Describing Products and Services Offers on the Web”, In *Proceedings of the 16th international conference on Knowledge Engineering: Practice and Patterns*, pp 329-346 Acitrezza, Italy: Springer-Verlag.

Hepp, M., Leukel, J. & Schmitz, V. (2005). “Content Metrics for Products and Services Categorization Standards”, In *EEE*, pp 740-745: IEEE Computer Society.

Horrocks, I. & Sattler, U. (2001). “Ontology reasoning in the SHOQ(D) description logic”, In *Proceedings of the 17th international joint conference on Artificial intelligence - Volume 1*, pp 199-204 Seattle, WA, USA: Morgan Kaufmann Publishers Inc.

Kim, H. M., Sengupta, A. & Evermann, J. (2007). “Web services ontologies for QoS and general quality evaluations”, *Int. J. Metadata Semant. Ontologies* 2(3): pp 195--200.

Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T., Sirin, E., Srinivasan, N. & Sycara, K. (2004). “OWL-S: Semantic Markup for Web Services”, *W3C Member Submission* World Wide Web Consortium (W3C). [Online] http://www.w3.org/Submission/OWL-S/

Maximilien, E. M. & Singh, M. P. (2004). “A Framework and Ontology for Dynamic Web Services Selection”, In *IEEE Internet Computin,* 8 (5), pp. 84-93.

Mohr, M. F. & Russell, A. S. (2002). “North American Product Classification System: Concepts and Process of Identifying Service Products”, In *17th Annual Meeting of the Voorburg Group on Service Statistics*.

O’Sullivan, J. (2006). “Towards a Precise Understanding of Service Properties”, *PhD Thesis,* Queensland University of Technology.

Oaks, P., ter Hofstede, A. H. M. & Edmond, D. (2003). “Capabilities: Describing What Services Can Do”, In *International Conference on Service Oriented Computing*, pp 1-16 Trento, Italy.

OASIS (2008). “Reference Ontology for Semantic Service Oriented Architectures Version 1.0. *Public Review Draft*, Organization for the Advancement of Structured Information Standards (OASIS). [Online] http://docs.oasis-open.org/semantic-ex/rosoa/v1.0/pr01/see-rosoa-v1.0-pr01.html

Orman, L. V. (2008). "Service Semantics, Structure and Design", *E-Service Journal* 6 (2), pp 58-77.

Scheithauer, G., Augustin, S. & Wirtz, G. (2008). "Describing Services for Service Ecosystems", In *ICSOC Workshops*, Vol. 5472, pp 242-255 (Eds G. Feuerlicht and W. Lamersdorf). Springer.

Schroth, C. & Janner, T. (2007). "Web 2.0 and soa: Converging concepts enabling the internet of services", *IT Professional,* 9(3), pp 36-41.

Sheth, A. P., Gomadam, K. & Ranabahu, A. (2008). "Semantics enhanced Services: METEOR-S, SAWSDL and SA-REST", *IEEE Data Eng. Bull.* 31(3): pp 8-12.

Spohrer, J., Anderson, L., Pass, N., Ager, T. & Gruhl, D. (2008). "Service Science". *Journal of Grid Computing,* 6 (3), pp. 313-324.

ToG (2008). "Service-Oriented Architecture Ontology", *Draft Technical Standard*, The Open Group. [Online] http://www.opengroup.org/projects/soa-ontology/uploads/40/16940/soa-ontology-200-draft.pdf .

Tran, V. X. (2008). "WS-QoSOnto: A QoS Ontology for Web Services", In *SOSE '08: Proceedings of the 2008 IEEE International Symposium on Service-Oriented System Engineering*, pp 233-238 Washington, DC, USA: IEEE Computer Society.

Uschold, M. & Grüninger, M. (1996). Ontologies: Principles, Methods and Applications. *Knowledge Engineering Review,* 11, pp. 93-136.

Vitvar, T., Kopecky, J., Zaremba, M. & Fensel, D. (2007a). "WSMO-Lite: Lightweight Semantic Descriptions for Services on the Web", In *Proceedings of the Fifth European Conference on Web Services*: IEEE Computer Society.

Vitvar, T., Zaremba, M., Moran, M., Zaremba, M. & Fensel, D. (2007b). "SESA: Emerging Technology for Service-Centric Environments", *IEEE Softw.,* 24 (6), pp 56-67.

W3C (2004a). "RDF Vocabulary Description Language 1.0: RDF Schema," *W3C Recommendation* World Wide Web Consortium (W3C). [Online] http://www.w3.org/TR/rdf-schema .

W3C (2004b). "Resource Description Framework (RDF): Concepts and Abstract Syntax" *W3C Recommendation* World Wide Web Consortium (W3C). [Online] http://www.w3.org/TR/rdf-concepts .

Wang, Z. & Xu, X. (2008). "Ontology-based Service Component Model for Interoperability of Service Systems," *Enterprise Interoperability III*, pp. 367-380.

DOI: 10.17705/1pais.02103

Appendix A. List of Competency Questions

Functional

CQ-F-1 Which business process does a given service instance follow?

CQ-F-2 Which business rules are applied in given service operation?

CQ-F-3 What are the decision points in a given service operation?

CQ-F-4 What are the input requirements for a given service operation?

CQ-F-5 What are interaction requirements for a given service operation?

CQ-F-6 Which are the value generating operations in a given service type?

Non-Functional

CQ-NF-1 What is the billing strategy *[Per Usage | Free | Indirect]* for a given service?

CQ-NF-2 What is the service experience type *[Resource | Access to Shared Resource | General Performance | Customized Performance | Personal Attention]* for a given service instance?

CQ-NF-3 What is the service provider type *[Human | Machine | Actor Network]* of a given service instance?

CQ-NF-4 What is the duration of the service experience *[On demand | Continuous]* for a given service type?

CQ-NF-5 What type of access is provided to the consumers? *[Personal | General] and [Competitive | Non-Competitive]*

CQ-NF-6 What is the service delivery point ownership *[Provider Owned | Consumer Owned | Shared]* in given service instance?

CQ-NF-7 What type of service contract is possible? *[By Choice | By law]*

CQ-NF-8 What quality dimensions are applicable for Service?

CQ-NF-9 What are measureable properties for a given service type?

CQ-NF-10 What is current service quality of a given service instance?

CQ-NF-11 What is the user rating of a service?

CQ-NF-12 Which properties of given service can be monitored?

CQ-NF-13 What are the payment options of a service?

CQ-NF-14 What are governance policies on which a service is based?

Classification

CQ-C-1*:* What is the industry classification of a given service?

CQ-C-2: What is the service type/name for a given service code?

CQ-C-3: What are the codes of other given services according to other classification schemes?

Business and Organization

CQ-B-1: What are the SLA properties of a given service type?

CQ-B-2: Who are the partners in a given service instance?

CQ-F-3 Which agreements are reached in a given service?

CQ-F-4 What are the terms that can be negotiated with the partners of a service?

CQ-F-5 What are the risk sharing agreements reached with service enablers?

System

CQ-SY-1: What delivery platform is used for a given service?

CQ-SY-2 What network addressed ports are used by a service?

CQ-SY-3 Which computations are carried out in a given instance of a service?

CQ-SY-4 What communication protocol is used by a service?

CQ-SY-5 Which instances are identified for backup?

CQ-SY-6 What is the current system load for a given service instance?

Transactional

CQ-TR-1: What is the instantaneous status of the service transaction?

CQ-TR-2: What is the number of ongoing transactions for a given service instance?

CQ-TR-3: What is the number of failed transactions for a given service instance?

CQ-TR-4: What are the events identified for a service transaction?

CQ-TR-5: What are the interpretations of a given service transaction?

CQ-TR-6: What are possible unfair uses of a service?

CQ-TR-7: What are the possible types of fault in a service transaction?

Situational

CQ-SI-1: What is the instantaneous competition of a service instance?

CQ-SI-2: What is the instantaneous demand for a service offering?

CQ-SI-3: What is the instantaneous market share of a service instance?

CQ-SI-4: What is the instantaneous economic environment of a service type?

CQ-SI-5: What is the instantaneous customer base of a service type?

Governance

CQ-G-1: What is the service engineering approach followed to develop a service?

CQ-G-2: What *[Standards | Rules | Regulations | Legislation]* is applicable to a given service type?

CQ-G-3: What is the current life-cycle phase of a given service instance?

CQ-G-4: Who is responsible for what in a given service instance?

CQ-G-5: Is a given service instance up-to-date *[Yes/No]*?

CQ-G-6: Is a given *[Quality | Coverage]* up-to-date?

CQ-G-7: Which policy is being applied to a given service?

CQ-G-8: Which legal requirements are relevant to a service?

Domain

CQ-G-1 Is a given activity a service?

CQ-G-2 Who are stakeholders for a given service instance?

CQ-G-3 Which academic disciplines are relevant for a given service type?

CQ-G-4 Which theories are being adopted to *[Engineer* | *Monitor* | *Evaluate]* a given service?

About the Authors

Vikram Sorathia is a post doctoral researcher at the University of Twente. He received his PhD degree from the Dhirubhai Ambani Institute of Information and Communication Technology (DA-IICT), India in 2009. His research interests are in the areas of services science, knowledge management, and software architecture frameworks.

Luís Ferreira Pires is an Associate Professor at the Faculty of Electrical Engineering, Mathematics and Computer Science of the University of Twente, Enschede, the Netherlands since 1994. He obtained his PhD degree in 1994 from the University of Twente and his MSc degree in 1989 from the 'Escola Politécnica da Universidade de São Paulo', São Paulo, Brazil. His research interests include design methodologies for distributed systems, architecture of distributed systems, modeling and specification techniques, middleware platforms and distributed applications. Lately he has been working on the application of model-driven architecture and semantic web technologies to the design of context-aware applications and services. He published more than 100 papers in international workshops, conferences and journals, and has contributed to the Program Committee of many conferences and workshops, particularly the IEEE EDOC Conference and the European Conference on Model Driven Architecture – Foundations and Applications (now called European Conference on Modeling Foundations and Applications). Amongst others, he contributed to the local organization of the ECMDA-FA conference in 2009 as Workshops Chair.

Marten J. van Sinderen is Associate Professor at the Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS) of the University of Twente. He is also research manager in the area of Service Architectures and Health Applications for the Centre for Telematics and Information Technology (CTIT), the ICT research institute of the University of Twente. His research interests include context-aware information systems, service-oriented architectures, model-driven design, and enterprise interoperability. He was leader of the Dutch Freeband A-MUSE project (BSIK 03025) on model-driven service design of context-aware mobile applications, and currently manages the Dutch GenCom U-Care project (IGC0816) on user-tailorable healthcare services for the home environment. He contributed to several European projects, including MODA-TEL (Model Driven Architecture for Telecommunications Systems Development and Operations, IST 37785), AMIGO (Ambient Intelligence for the Networked Home Environment, IST 004182), and SPICE (Service Platform for Innovative Communication Environment, IST 027617). He is a member of the Editorial Board of the Enterprise Information Systems journal, published by Taylor & Francis. He is also member of the Managerial Board of IFIP WG5.8 on Enterprise Interoperability.

https://aisel.aisnet.org/pajais/vol2/iss1/4
DOI: 10.17705/1pais.02103