Chapter 13

Context-Aware Semantic Web Service Discovery through Metric-based Situation Representations

Stefan Dietze, John Domingue, Michael Mrissa, Alessio Gugliotta

Abstract Semantic Web Services (SWS) enable the automatic discovery of distributed Web services based on comprehensive semantic representations. However, although SWS technology supports the automatic allocation of Web services for a given well-defined task, it does not entail their discovery according to a given situational context. Whereas tasks are highly dependent on the situational context in which they occur, SWS technology does not explicitly encourage the representation of domain situations. Moreover, describing the complex notion of a specific situation in all its facets is a costly task and may never reach sufficient semantic expressiveness. Particularly, following the symbolic SWS approach leads to ambiguity issues and does not entail semantic meaningfulness. Apart from that, not any real-world situation completely equals another, but has to be matched to a finite set of semantically defined parameter descriptions to enable context-adaptability. To overcome these issues, we propose Conceptual Situation Spaces (CSS) which are aligned to established SWS standards. CSS enable the description of situation characteristics as members in geometrical vector spaces following the idea of Conceptual Spaces. Semantic similarity between situations is calculated in terms of their Euclidean distance within a CSS. Extending merely symbolic SWS descriptions with context information through CSS enables similarity-based matchmaking between real-world situation characteristics and predefined resource representations as part of SWS descriptions. To prove its feasibility, we apply our approach to the E-Learning and E-Business domains and provide a

0-8493-0052-5/00/$0.00+$.50

proof-of-concept prototype.

13.1 Introduction

Context-aware discovery and invocation of *Web services* is highly desired across a wide variety of application domains and subject to intensive research throughout the last decade (9, 35, 39). According to Dey's well-known definition (7), we define *context* as the entire set of surrounding characteristics that characterize an entity relevant to the interaction between a user and an application (including the user and the application themselves). Each individual *situation* represents a specific state of the world, and more precisely, a particular state of actual context. A *situation description* defines the context in a particular situation, and is described by a combination of *situation parameters*, each representing a particular situation characteristic. Following this definition, context-adaptation can be defined as the ability to adapt to distinct possible situations.

Semantic Web Services (SWS) technology (16) supports the automatic discovery of distributed Web services for a given task based on comprehensive semantic descriptions. Concretely, a SWS is a Web service with a description that contains explicit semantic information, and SWS technology consists of tools and technologies that enable getting the benefits from these semantically-explicit service descriptions. First results of SWS research are available, in terms of reference ontologies - e.g. OWL-S (27) and WSMO (1) - as well as tools and comprehensive frameworks (e.g. DIP project[1] results).

However, whereas SWS technology supports the allocation of appropriate resources based on semantic representations, it does not entail the discovery of appropriate SWS representations for a given situation, i.e. the actual context. Even though tasks, as semantically described through SWS representations, are highly dependent on the situation in which they occur, current SWS technology does not explicitly encourage the representation of domain situations related to task representations. Furthermore, describing the complex notion of a specific situational context in all its facets is a costly task and may never reach sufficient semantic expressiveness. The symbolic approach - describing symbols by using other symbols without a grounding in the real world - of established SWS and Semantic Web representation standards in general, such as RDF[2], OWL[3], OWL-S (27), or WSMO (1) leads to ambiguity issues and does not entail semantic meaningfulness, since meaning requires both the definition

[1]DIP Project: `http://dip.semanticweb.org`

[2]`http://www.w3.org/RDF/`

[3]`http://www.w3.org/TR/owl2-primer/`

of a terminology in terms of a logical structure (using symbols) and grounding of symbols to a cognitive or perceptual level (5, 23). Moreover, whereas not any situation or situation parameter completely equals another, the number of predefined semantic representations of situations and situation parameters within a SWS description is finite. Consequently, to enable context-adaptive resource discovery, a potential infinite set of (real-world) situation characteristics has to be matched to a finite set of semantically defined situation parameter descriptions. Therefore, rather fuzzy classification and matchmaking techniques are required to classify a real-world situation based on a limited set of predefined parameter descriptions to support the discovery of the most appropriate SWS representation within a given situation context.

Conceptual Spaces (CS), introduced by Gärdenfors (19, 18), follow a theory for describing entities in terms of their natural characteristics similar to natural human cognition in order to avoid the symbol grounding issue. CS enable representation of objects as vector spaces within a geometrical space which is defined through a set of quality dimensions. For instance, a particular color may be defined as point described by vectors measuring the quality dimensions hue, saturation, and brightness. Describing instances as vector spaces where each vector follows a specific metric enables the automatic calculation of their semantic similarity, in terms of their Euclidean distance, in contrast to the costly representation of such knowledge through symbolic SW representations. Even though several criticisms have to be taken into account when utilizing CS (Section 15.7) they are considered to be a viable option for knowledge representation.

In this paper, we propose *Conceptual Situation Spaces (CSS)* which utilize CS to represent situational contexts. CSS are mapped to standardized SWS representations to enable, first, context-aware discovery of appropriate SWS descriptions, and finally, automatic discovery and invocation of appropriate Web services to achieve a given task within a particular situation. Extending merely symbolic SWS descriptions with context information on a conceptual level through CSS enables fuzzy and similarity-based matchmaking between real-world situation characteristics and predefined SWS representations. Whereas similarity between situation parameters, as described within a CSS, is indicated by the Euclidean distance between them, real-world situation parameters are classified along predefined prototypical parameters which are implicit elements of a SWS description. Whereas current SWS technology addresses the issue of allocating resources for a given task, our approach supports the discovery of SWS representations within a given situational context. Consequently, the expressiveness of current SWS standards is extended through CSS in order to enable fuzzy matchmaking mechanisms when allocating resources for a given situation.

To prove the feasibility of our approach two proof-of-concept prototypes are provided. The first prototype relates to the domain of E-Learning and uses CSS to describe learning styles, following the Felder-Silverman Learning Style theory (15), as particular learning situation parameter. The second prototype illustrates a CSS application to the E-Business domain, and uses CSS to describe business actors' requirements according to current situational contexts.

The paper is organized as follows. The following Section 13.2 provides background information on SWS and the discovery problem, and gives an overview on related works in the field. Section 13.3 introduces our approach of Conceptual Situation Spaces which are aligned to current SWS representations. Section 13.4 illustrates the application of CSS to the E-Learning domain and introduces a Conceptual Learning Situation Space, particularly, a CSS subspace representing learning styles. Utilizing CSS, we introduce our approach to similarity-based classification of a given situation based on distance calculation at runtime in Section 13.5. Section 13.6 shows an application of CSS in the E-Business examples, and details how CSS helps discovering SWS depending on business actors' requirements at runtime, before giving some insight on the use of CSS for semantic mediation of data in business processes. Finally, we conclude our work in Section 15.7 and provide an outlook to future research.

13.2 Background and Motivation

In this section, we provide some background information and motivate our approach by reporting on the current state of the art in Semantic Web Services discovery.

13.2.1 Semantic Web Services (SWS) and Context-dependent SWS Mediation

SWS technology aims at the automatic discovery, orchestration and invocation of distributed services for a given user goal on the basis of comprehensive semantic descriptions. SWS are supported through representation standards such as WSMO (1) and OWL-S (27). In this paper, we particularly refer to the Web Service Modelling Ontology (WSMO), a well established SWS reference ontology and framework. WSMO is currently supported through several software tools and runtime environments, such as the Internet Reasoning Service IRS-III (2) and WSMX (22). The conceptual model of WSMO defines the following four main entities:

- **Domain ontologies** not only support Web service related knowledge representation but semantic knowledge representation in general. They provide the foundation for describing domains semantically, and they are used by the three other WSMO entities.

- **Goals** define the tasks that a service requester expects a Web service to fulfill. In this sense they express the requester's intention.
- **Web service descriptions** represent the functional behavior of an existing deployed Web service. They also outline how Web services communicate (choreography) and how they are composed (orchestration). In this paper, a SWS represents the semantic description of a particular Web service and is synoymous with the term SWS description.
- **Mediators** handle data and process interoperability issues that arise when handling heterogeneous systems.

A SWS description (either the description of the Web service or the description of the service request) is formally represented within a particular ontology that complies with a certain SWS reference model such as OWL-S (27) or WSMO (1). By adopting a common formalisation of an ontology (11, 12), we define a populated *service ontology* O – as utilised by a particular SWS representation – as a tuple:

$$O = \{C, I, P, R, A\} \subset SWS$$

With C being a set of n *concepts* where each concept C_i is described through $l(i)$ *concept properties pc*, i.e.:

$$PC_i = \left\{ (pc_{i1}, pc_{i2}, \ldots, pc_{l(i)}) \,| pc_{ix} \in C_i \right\}.$$

I represents all m *instances* where each instance I_{ij} represents a particular instance of a concept C_j and consists of *l(i) instantiated properties pi* instantiating the concept properties of C_j:

$$PI_{ij} = \left\{ (pi_{ij1}, pi_{ij2}, \ldots, pi_{l(i)}) \,| pi_{ijx} \in I_{ij} \right\}.$$

Hence, the properties P of an ontology O represent the union of all concept properties PC and instantiated properties PI of O:

$$P = \{(PC_1, PC_2, \ldots, PC_n) \cup (PI_1, PI_2, \ldots, PI_m)\}$$

Given these definitions, we would like to point out that properties here exclusively refer to so-called data type properties. Hence, we define properties as being distinctive to relations R. The latter describe relations between concepts and instances. In addition, A represents a set of *axioms* which define constraints on the other introduced notions. Since certain parts of a SWS ontology describe certain aspects of the Web service (request), such as its capability *Cap*, interface *If* or non-functional properties *Nfp* (4), a SWS ontology can be perceived as a conjunction of ontological subsets:

$$Cap \cup If \cup Nfp = O \subset SWS$$

The semantic capability description, as central element of a SWS description, consists of further subsets, describing the assumptions As, effects *Ef*, preconditions *Pre* and postconditions *Post*. However, given the lack of a clear distinction between assumption/effect and pre-/postcondition, we prefer the exclusive usage of assumptions/effects:

$$As \cup Ef = Cap \subset O \subset SWS$$

Given that a SWS ontology by its very nature always captures the semantics of a service from a specific perspective, it represents a specific context in which the annotated service is meant to be used. Hence, even when explicitly representing information about the Web service context, the nature of ontologies - being symbolic representations of conceptualisations from a specific viewpoint - leads to highly heterogeneous SWS descriptions.

SWS mediation aims at addressing heterogeneities among distinct SWS to support all stages that occur at SWS runtime, namely *discovery, orchestration* and *invocation*. In contrast to (4, 32), we classify the mediation problem into (i) *semantic level* and (ii) *data level mediation* (Fig. 13.1).

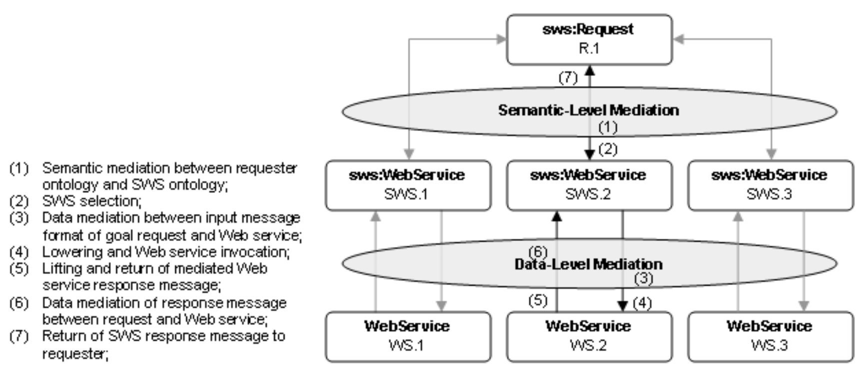

Figure 13.1: Semantic level and data level mediation as part of SWS discovery, orchestration and invocation

Whereas (i) refers to the resolution of heterogeneities between concurrent semantic representations of services and service contexts – the actual SWS representations – (ii) refers to the mediation between mismatches related to the Web service implementations themselves, i.e. related to the structure, value or format of I/O messages. Hence, semantic level mediation primarily supports the discovery stage, whereas data level mediation occurs during orchestration and invocation. As shown in Fig. 13.1, semantic level mediation occurs before a particular SWS is selected and aims at aligning distinct semantic vocabularies, for instance, the ones used by a SWS requester and a

SWS provider. Please note that, for the sake of simplification, Fig. 13.1 just depicts mediation between a SWS request and multiple SWS, while leaving aside mediation between different SWS or between different requests.

13.2.2 Semantic Web Services Discovery - An Ontology Mapping Problem

In this chapter, we exclusively address semantic level mediation between distinct context-representations, which is perceived to be a fundamental requirement for context-adaptive SWS discovery, and hence, to further exploit SWS approaches on a Web scale. In order to better understand the needs of semantic level mediation, it is necessary to understand the requirements of the SWS discovery task to which semantic level mediation is supposed to contribute. In order to identify whether a particular SWS S_1 is potentially relevant for a given request S_2 - also representing a particular context - a SWS broker has to compare the capabilities of S_1 and S_2 ,i.e. it has to identify whether the following holds true:

$$As_2 \subset As_1 \cup Ef_2 \subset Ef_1$$

However, in order to compare distinct contextual annotations of available SWS which each utilize a distinct vocabulary, these vocabularies have to be aligned. For instance, to compare whether an assumption expression of one particular S_1 is the same as of another S_2, where I_i represents a particular instance, matchmaking engines have to perform two steps: (a) identification of relationships between concepts/instances involved in distinct SWS representations; (b) evaluation whether the semantics of the logical expressions used by each SWS match each other, i.e. represent the same fact (i.e. capability).

Whereas current SWS execution environments exclusively focus on (b), semantic level mediation also requires mediation between different SWS context ontologies, as in (a), and can be perceived as a particular instantiation of the *ontology mapping problem* (40). With respect to (3), we define ontology mapping as the creation of structure-preserving relations between multiple ontologies. I.e. the goal is, to establish formal relations between a set of knowledge entities E_1 from an ontology O_1 - used to represent a particular SWS S_1 - and entities E_2 which represent the same or a similar semantic meaning in a distinct ontology O_2 (11, 12) which is used to represent an additional S_2. The term set of entities here refers to the union of all concepts C, instances I, relations R and axioms A defined in a particular SWS ontology. In that, semantic mediation strongly relies on identifying *semantic similarities* between entities across different SWS ontologies. Hence, the identification of similarities is a necessary requirement to solve the heterogeneity problem for multiple SWS representations (33, 40). However, in this respect, the following issues have to be taken into account:

1. Symbolic SWS and context representations lack grounding to concep-

tual level: similarity-detection across distinct SWS descriptions requires semantic meaningfulness which inherently describes semantic similarity between represented entities. However, the symbolic approach - i.e. describing symbols by using other symbols, without a grounding in the real world - of established SWS representation standards, leads to ambiguity issues and does not fully entail semantic meaningfulness, since meaning requires both the definition of a terminology in terms of a logical structure (using symbols) and grounding of symbols to a conceptual level (5).

2. Lack of automated similarity-detection methodologies: Describing the complex notion of specific SWS contexts in all their facets is a costly task and may never reach sufficient semantic expressiveness due to the above. While contextual representations across distinct SWS representations - even those representing the same real-world entities - hardly equal each other, semantic similarity is not an implicit notion within SWS representations. But manually or semi-automatically defining similarity relationships is costly. Moreover, such relationships are hard to maintain in the longer term.

Given the lack of inherent similarity representation, current approaches to *ontology mapping* could be applied to facilitate context-aware SWS discovery. These approaches aim at semi-automatic similarity detection across ontologies mostly based on identifying linguistic commonalities and/or structural similarities between entities of distinct ontologies (3, 31). Work following a combination of such approaches in the field of ontology mapping is reported in (31, 13, 20, 28). However, it can be stated, that such approaches require manual intervention, are costly and error-prone, and hence, similarity-computation remains as central challenge. In our vision, instead of semi-automatically formalising individual mappings, we rely on methodologies to automatically compute or implicitly represent similarities across distinct SWS representations, which are better suited to facilitate SWS mediation.

13.2.3 Spatial Approaches to Knowledge Representation

Distinct streams of research approach the automated computation of similarities through spatially oriented knowledge representations. *Conceptual Spaces (CS)* follow a theory of describing entities at the conceptual level in terms of their quality characteristics similar to natural human cognition in order to bridge between the neural and the symbolic world. (19) proposes the representation of concepts as multidimensional geometrical *Vector Spaces* which are defined through sets of quality dimensions. Instances are supposed to be represented as vectors, i.e. particular points in a space CS. For instance, a particular color may be defined as point described by vectors measuring the quality dimensions hue, saturation, and brightness. Describing instances as points within vector spaces where each vector follows a specific metric enables the automatic calculation of their semantic similarity by means of distance metrics such as the Euclidean, Taxicab or Manhattan distance (26) or the Minkowsky Metric (37). Hence, in contrast to the costly formalization of such knowledge through symbolic representations, semantic similarity is implicit information carried within a CS representation which is perceived as the major contribution of the CS theory. *Soft Ontologies (SO)* (25) follow a similar approach by representing a knowledge domain D through a multidimensional *ontospace* A, which is described by its so-called *ontodimensions*. An item I, i.e. an instance, is represented by scaling each dimension to express its impact, presence or probability in the case of I. In that, a SO can be perceived as a CS where dimensions are measured exclusively on a ratio-scale.

However, although CS and SO aim at solving SW(S)-related issues, several issues still have to be taken into account. For instance, similarity computation within CS requires the description of concepts through quantifiable metrics even in case of rather qualitative characteristics. Moreover, CS as well as SO do not provide any notion to represent any arbitrary relations (36), such as *part-of* relations which usually are represented within symbolic knowledge models such as SWS representations. In this regard, it is even more obstructive that the scope of a dimension is not definable, i.e. a dimension always applies to the entire CS/SO (36).

13.3 Conceptual Situation Spaces for Semantic Web Services

In this section, we describe the formalisms developed to backup our approach to context-aware SWS discovery, which is based on describing situational contexts as members within a domain-specific *Conceptual Situation Space (CSS)* which are incorporated into SWS descriptions.

13.3.1 Approach: Grounding SWS Contexts in Conceptual Situation Spaces

CSS enable the description of a particular context within a particular situation as a member of a dedicated CS which are used to describe SWS capabilities. CSS enable the implicit representation of semantic similarities across heterogeneous SWS context representations provided by distinct agents. Hence, refining heterogeneous SWS context descriptions into a set of shared CSS supports similarity-based mediation at the semantic level and consequently facilitates context-aware SWS discovery. Whereas CSS allow the representation of semantic similarity as an implicit notion, it can be argued,

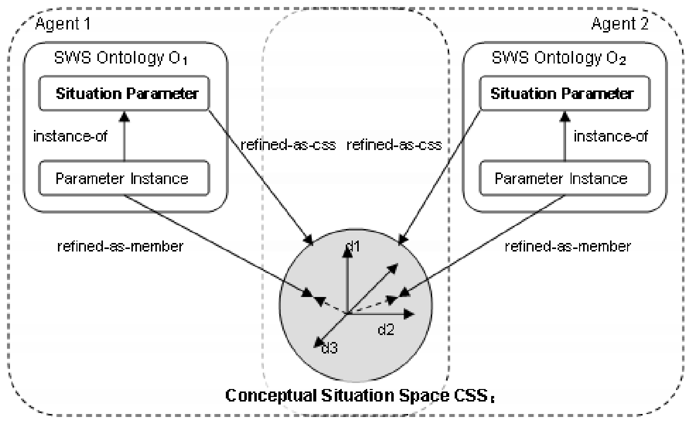

Figure 13.2: Representing distinct situation parameters - being part of heterogeneous SWS representations - through shared CSS

that representing an entire SWS context through a coherent CSS might not be feasible, particularly when attempting to maintain the meaningfulness of the spatial distance as a similarity measure.

Therefore, we claim that CSS are a particularly promising model when being applied to individual situation parameter concepts - as part of SWS descriptions - instead of representing an entire SWS ontology in a single CSS. In that, we would like to highlight that we consider the representation of a set of n situation parameters S of a SWS ontology O through a set of n CSS 13.2. Hence, instances of parameters are represented as members in the respective CSS. While still taking advantage from implicit similarity information within a CSS, our hybrid approach - combining SWS descriptions with multiple CSS - allows to overcome CS-related issues by maintaining the advantages of ontology-based SWS context representations. Please note that our approach relies on the agreement on a common set of CSS for a given set of distinct SWS ontologies O_1 and O_2, instead of a common agreement on set of shared ontologies. Hence, while in the latter case two agents have to agree on a common ontology at the concept and instance level, our approach only requires agreement at the concept level, since instance similarity becomes an implicit notion. Moreover, we assume that the agreement on ontologies at the concept level (Fig. 13.2) becomes an increasingly widespread case, due to, on the one hand, increasing use of upper-level ontologies such as DOLCE (17), SUMO[4] or OpenCyc[5] which support a certain degree of commonality between distinct ontologies. On the other hand, SWS ontologies often are provided

[4]`http://www.ontologyportal.org/`
[5]`http://www.opencyc.org/`

within closed environments, for instance, virtual organizations, where a common agreement to a certain extent is ensured. In such cases, the derivation of a set of common CSS is particularly applicable and straightforward.

13.3.2 A Formalisation for Conceptual Situation Spaces

Our approach formalizes the notion of CSS via a dedicated *Conceptual Situation Space Ontology (CSSO)*, based on OCML (29) and aligned to SWS in order to enable SWS description. Since both metamodels, WSMO as well as CSS, are represented based on the OCML representation language, the alignment was accomplished by defining relations between concepts of both ontologies. The design, formalization and alignment processes have been detailed in previous work. To make this paper self-contained and for a good understanding, we provide the necessary results obtained from this previous work in the following. For additional details, we refer the reader to (10). Hence, a CSS is defined as a vector space with weighted dimensions:

$$C^n = \{(p_1c_1, p_2c_2, \ldots, p_nc_n)|c_i \in C, p_i \in P\}$$

where c_i being the quality dimensions of C and P the set of real numbers. The prominence value p is attached to each dimension in order to reflect the impact of a specific quality dimension on the entire CSS. We enable dimensions to be detailed further in terms of subspaces. Hence, a dimension within one space may be defined through another conceptual space by using further dimensions (34, 10).

Semantic similarity between two members of a space is perceived as a function of the Euclidean distance between the points representing each of the members. Hence, given a CSS definition C and two members represented by two vector sets V and U, defined by vectors $v_0, v_1, \ldots, v_n$ and $u_1, u_2, \ldots, u_n$ within C, the distance between V and U can be calculated as:

$$|d(u,v)|^2 = \sum_{i=1}^{n}(z(u_i) - z(v_i))^2$$

where $z(u_i)$ is the so-called Z-transformation or standardization (6) from u_i. Z-transformation facilitates the standardization of distinct measurement scales which are utilized by different quality dimensions in order to enable the calculation of distances in a multi-dimensional and multi-metric space. The z-score of a particular observation u_i in a dataset is calculated as follows:

$$z(u_i) = \left(\frac{u_i - \bar{u}}{s_u}\right)$$

where $\overline{u}$ is the mean of a dataset U and s_u is the standard deviation from U. Considering prominence values p_i for each quality dimension i, the Euclidean distance $d(u,v)$ indicating the semantic similarity between two members described by vector sets V and U can be calculated as follows:

$$d(u,v) = \sqrt{\sum_{i=1}^{n} p_i \left(\left(\frac{u_i - \bar{u}}{s_u} \right) - \left(\frac{v_i - \bar{v}}{s_v} \right) \right)^2}$$

13.4 A Conceptual Learning Situation Space

In (8) we introduce a general-purpose procedure for refining arbitrary ontologies, such as SWS, through CS-based representations such as a CSS. However, in this section we would like to illustrate CSS through a particular example from the e-Learning domain, a *Conceptual Learning Situation Space*, which also demonstrates the applicability of our metamodel even to rather qualitative parameters. As described in (10) a learning situation is defined by parameters such as the technical environment used by a learner, his/her competency profile or the current learning objective. This Section focuses exemplarily on the representation of one parameter through a CSS subspace, which is of particular interest within the E-Learning domain: the learning style of a learner. A learning style is defined as an individual set of skills and preferences on how a person perceives, gathers, and processes learning materials (24). Whereas each individual has his/her distinct learning style, it affects the learning process (15) and consequently has to be perceived as an important parameter describing a learning situation. To describe a learning style, we refer to the Felder-Silverman Learning Style Theory (FSLST) (15) approach to describe learning styles within computer-aided educational environments (14), where a learning style is described with four quality dimensions (15) defined here with 4 quality dimensions l_i that hold metric scale, datatype, value range and prominence values in a CSS L, as presented in Table 13.1:

	Quality Dimension	Metric Scale	Data-type	Range	Prominence
l_1	Active-Reflective	Interval	Integer	-11..+11	1.5
l_2	Sensing-Intuitive	Interval	Integer	-11..+11	1
l_3	Visual-Verbal	Interval	Integer	-11..+11	1.5
l_4	Global-Sequential	Interval	Integer	-11..+11	1

As depicted in Table 13.1, each quality dimension is ranked on an interval scale with a value range being integers between -11 and +11. This particular measurement scale was defined with respect to an established assessment method, the Index of Learning Styles (ILS) questionnaire defined by Felder and Soloman (14), aimed at identifying and rating the particular learning style of an individual. The authors would like to highlight, that prominence values have been assigned which rank the first (l_1) and the third dimension

(l_3) higher than the other two, since these have a higher impact with respect to the purpose of the learning situation, which is focused on the aim to deliver appropriate learning material to the learner.

In order to classify an individual learning style, we define prototypical members in the FSLST-based vector space L. To identify appropriate prototypes, we utilized existing knowledge about typical correlations between the FSLST dimensions, as identified throughout research studies such as (21, 38). Details on the construction methodology of these prototypes are given in (10). This resulted in the following 5 prototypical members and their characteristic vectors shown in Table 13.2.

Prototype	Act/Ref	Sen/Int	Vis/Ver	Seq/Glo
P1: Active-Visual	-11	-11	-11	+11
P2: Reflective	+11	-11	-11	0
P3: Sensing-Seq.	-11	-11	-11	-11
P4: Intuitive-Glob.	-11	+11	-11	+11
P5: Verbal	-11	+11	+11	+11

13.5 Fuzzy SWS Goal Discovery and Achievement at Runtime

To prove the feasibility of our approach, a proof-of-concept prototype application[6] was provided, which utilizes the CSS metamodel and ontology framework introduced in Sections 13.3 and 13.4 to implement a use case from the E-Learning domain.

13.5.1 Runtime Reasoning Support for CSS and SWS

In order to describe situations within the domain of E-Learning, a CSS specific for the domain of E-Learning was provided which is able to represent domain-specific situations described by concepts defined within a particular WSMO domain ontology. Linking each situation parameter, defined within a WSMO SWS description, to a particular CSS, and defining prototypical instances within each CSS enables the automatic classification of situation parameters in terms of their similarity with a set of prototypical parameters.

[6] The application is utilized within the EU FP6 project LUISA (`http://www.luisa-project.eu/www/`)

Fig. 13.3 depicts the architecture used to support reasoning on CSS and SWS in distinct domain settings through a Semantic Execution Environment (SEE) which is in our case implemented through IRS-III (Section 13.5).

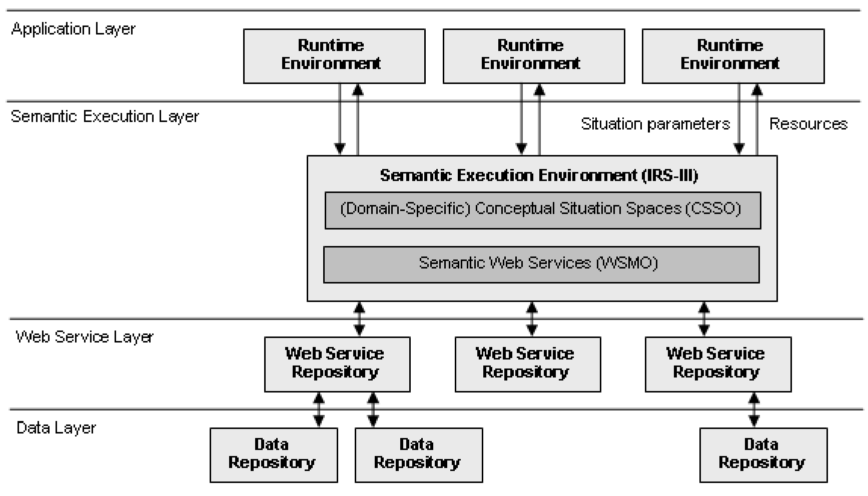

Figure 13.3: Architecture to support runtime reasoning on CSS and SWS models

SEE utilizes a semantic representation of the CSS metamodel (CSSO), which is derived for specific domains, and of the SWS metamodel based on WSMO. Both are represented utilizing the OCML representation language (29). IRS-III dynamically classifies a given situation based on the CSSO and provides resources - represented based on WSMO - which suit a specific runtime situation. Distinct runtime environments can serve as user interfaces to enable users to interact with SEE and to provide knowledge about the current real-world situation. Given a set of real-world situation parameters, their semantic distance to predefined prototypical situation parameters, defined within a domain-specific CSS, is calculated to enable classification of a set of real-world situation parameters. The SEE finally discovers and orchestrates appropriate Web services which show the capabilities to suit the given situation.

13.5.2 SWS Goal Discovery based on Context Classification

In order to reach situation awareness, the application automatically detects semantic similarity of specific situation parameters with a set of predefined prototypical parameters to enable the allocation of context-appropriate resources through the SEE. Referring to a CSS subspace L described previously, given a particular member U in L, its semantic similarity with each of the prototypical members is indicated by their Euclidean distance. Since we utilize a CSS described by dimensions which each use the same metric scale (ordinal scale), the distance between two members U and V is calculated disregarding a Z-transformation (Section 13.3) for each vector.

The calculation of Euclidean distances is accomplished by a standard Web service which is exposed as SWS and is invoked through IRS-III at runtime. Given a particular CSS description, a member (representing a specific parameter instance) as well as a set of prototypical member descriptions (representing prototypical parameter instances), the Web service calculates similarities at runtime in order to classify a given situation parameter. For instance, a particular situation description includes a learner profile indicating a learning style parameter which is defined by a member U in the specific CSS subspace to describe learning styles following FSLST with the following vectors:

$$U = \{(u_1 = -5, u_2 = -5, u_3 = -9, u_4 = 3)\,|u_i \in L\}$$

Learning styles such as the one above, could be assigned to individual learners by utilizing the ILS Questionnaire as assessment method. Calculating the distances between U and each of the prototypes described in Table 13.2 of Section 13.4 led to the following results:

Prototype	**Euclidean Distance**
P1: Active-Visual	12.649110640673518
P2: Reflective	20.85665361461421
P3: Sensing-Sequential	17.08800749063506
P4: Intuitive-Global	19.493588689617926
P5: Verbal	31.20897306865447

As depicted in Table 13.3, the lowest Euclidean distance between U and the prototypical learning styles applies to $P1$, indicating a rather active and visual learning style described as in Table 13.2 of Section 13.4.

Classified contexts are utilized to discover the most appropriate SWS goal representation for a given context, by utilizing the alignment of CSS and SWS (Section 13.3). Given a specific situation description, IRS-III first identifies SWS goal representations (*wsmo:Goal*) which suit the given situation and finally selects and orchestrates SWS which are appropriate to suit the given runtime situation. For instance, in the proposed use case, distinct SWS goal representations are available, each retrieving content which addresses a distinct learning style (Fig. 13.4).

Given the similarity-based classification of a set of real-world parameters - e.g. learning styles - a SWS goal representation which assumes matching

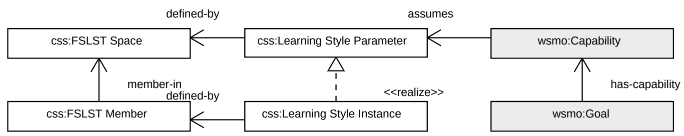

Figure 13.4: SWS Goals assuming learning styles described as members in a CSS

prototypical parameter instances is selected and achieved through IRS-III (Fig. 13.3, Section 13.4). Finally, IRS-III utilizes the SWS goal capability to identify a SWS which suits the given goal. For instance, given a classified learning style together with classifications of all further situation parameters, a SWS goal representation which assumes matching prototypical situation parameter instances is selected and achieved at runtime. Consequently, following the alignment of CSS with established SWS frameworks, context-aware SWS applications are enabled which automatically discover not only Web services for a given task but also SWS goal descriptions for a given situation.

13.6 Applying CSS to the E-Business Domain

As demonstrated in the previous section, refining SWS through CSS helps describing and reasoning about SWS capabilities, which in turn enhances context-aware selection/matchmaking of SWS. In this section, we demonstrate the relevance of CSS with respect to the E-Business application domain. We detail how CSS dimensions are designed and utilized, according to the requirements of involved business actors. In addition, we discuss in the end of this section how CSS could be used in conjunction with a context model presented in (30), in order to enhance semantic mediation of data between Web services. Hence, this section demonstrates the independence of CSS with respect to applications domains and underlying semantic models, and offers some insights on foreseeable use of CSS for other purposes such as semantic mediation of data.

13.6.1 The Ordering Example

In contrast to the previous example from the E-Learning domain, in this section we develop another use case of CSS application, illustrated with a typical E-Business scenario. Let us assume a UK-based goods reseller who aims at ordering manufactured goods from a Japanese producer via a broker that selects the best producer WS according to its client's requirements. For the purpose of this example, we identify the requirements of the reseller with the following set of criteria:

- shipping delay, measured in days (comprised between 0 and ∞),
- type of packaging, set to fragile, normal or secured and identified with the values 1, 2 and 3 in the CSS dimension,
- quantity of items per delivery (comprised between 0 and ∞),
- payment security level, set to low (no encryption), medium (weak encryption method) or high (strong encryption such as SSL) and identified with 1, 2 or 3,
- authentication security level, set to 0 (no authentication) or 1 (password-based authentication) or 2 (certificate-based authentication).

For the purpose of SWS discovery, we evaluate the quality offered by each producer SWS with respect to current reseller's requirements represented as a CSS member. Indeed, the selected producer SWS is "only" the closest match to the reseller's requirements and may not completely satisfy them. In order to give priority to some dimensions over others, weighted prominences were setup on quality dimensions. Table 13.4 summarizes the different dimensions composing the CSS and proposes prominences that are useful to the selection task.

	Quality Dimension	Metric Scale	Data-type	Range	Prominence
l_1	shipping delay	Interval	Integer	$+1..+\infty$	1.5
l_2	type of packaging	Interval	Integer	$+1..+3$	1
l_3	number of items/delivery	Interval	Integer	$+1..+\infty$	0.5
l_4	payment security level	Interval	Integer	$+1..+3$	2
l_5	authentication security level	Interval	Integer	$+0..+2$	1.5

The values of these requirements may often change as they depend on external (environmental) conditions related to business/economical issues. Therefore, in order to select the best producer, the realization of SWS discovery is relevant at runtime via a broker. The SWS discovery and ordering process involves simple interactions between business actors. These interactions are modeled as a business process, as described in Fig. 13.5.

The reseller sends a SWS request by means of a WSMO goal to the broker (step 1). This request includes a set of parameters that describe the aforementioned requirements. Then, the broker searches for SWS that provide the functionality (step 2). It sends back either (step 3a) a positive answer containing access information to the best WS found, or (step 3b) a negative answer (no SWS found). In the latter case, the business process either goes back to step 1 or ends with a failure notification (step 4b). If the answer was positive, the actual order is sent to the broker, that acts as a proxy (step 4a) and the transaction is performed (step 5), before ending the business process with a successful notification (step 6). Interactions between Web services are typically specified in a WS-BPEL[7] file, as a sequence of WSMO goal invocations which are resolved by the broker (i.e. IRS-III) [8]. For the sake of brevity we do not detail the technical aspects related to the business process that is sketched in Fig. 13.5.

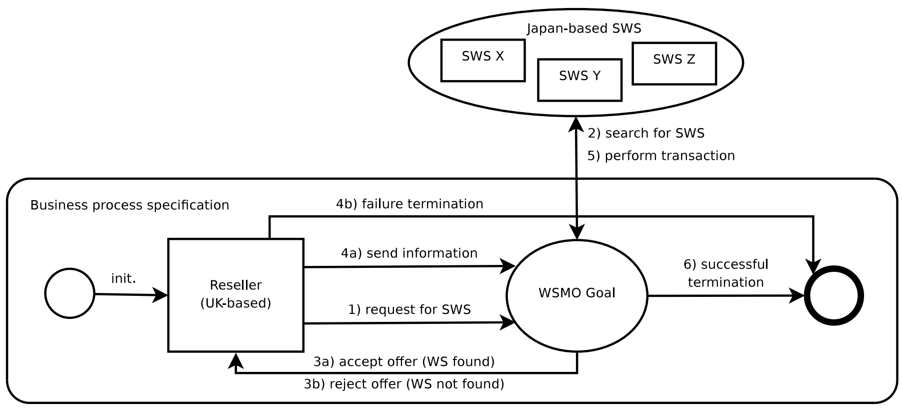

Figure 13.5: Overview of the ordering business process

13.6.2 Overview of SWS Discovery with CSS

In the following, we provide details about the SWS discovery process to be achieved in step 2) of Fig. 13.5, according to the aforementioned reseller's requirements, which are described as CSS members (each requirement is expressed as a vector valueing one of the dimensions). Each producer SWS is also represented as a CSS member (a set positions on vectors) and matched with the requirements of the reseller. The discovery process takes place in a similar way to the one described in Section 13.4. In order to illustrate the selection process, we provide a sample of the evaluation process on the basis of our developed example. The CSS is described as illustrated in Table 13.4. The reseller's requirements are described with the following CSS member:

$$V_{Optimal} = \{(l_1 = 1, l_2 = 3, l_3 = 20, l_4 = 2, l_5 = 2)\,|l_i \in L\}$$

[7]www.oasis-open.org/committees/wsbpel/

[8]See the SUPER project for additional details `http://www.ip-super.org/`.

where L is the CSS subspace defined with Table 13.4. For the purpose of the demonstration, we described the providers X, Y and Z shown in Fig. 13.5. They each present specific characteristics that are described in CSS as follows:

$$V_{Prov_X} = \{(l_1 = 3, l_2 = 1, l_3 = 20, l_4 = 1, l_5 = 0)\,|l_i \in L\}$$
$$V_{Prov_Y} = \{(l_1 = 2, l_2 = 2, l_3 = 15, l_4 = 2, l_5 = 1)\,|l_i \in L\}$$
$$V_{Prov_Z} = \{(l_1 = 3, l_2 = 3, l_3 = 5, l_4 = 2, l_5 = 2)\,|l_i \in L\}$$

With this information, it is possible for the broker to calculate the distance between the CSS representation of the reseller's requirements and each SWS. In our example, we obtain the following result:

Provider	**Euclidean Distance**
Provider X	4.242640687119285
Provider Y	4.06201920231798
Provider Z	10.88577052853862

Table 13.5 shows the respective distance from each SWS to the given requirements. In our case, *providerY* is the best SWS for these requirements and is then selected by the broker.

Concretely, the broker makes use of a Web service that calculates the Euclidean distance between members of a CSS according to a given base member (here $V_{optimal}$). The response from the Web service returns the respective distances of the members and lastly, a SmallestDistance tag indicates the member that happens be the most appropriate for the requirements of the request.

13.6.3 An Insight on Context-based Data Mediation with CSS

In (30), a context model is proposed to enable mediation of data exchanged between Web services. According to Section 13.2.1, where semantic-level and data-level mediation are introduced, we provide in this section some insight on how CSS can be used with this context model to offer novel possibilities for data-level mediation.

At the step 4a and 5 of our E-Business example, certain data (i.e. input/output data) is exchanged between the reseller and the best producer selected by the broker. Indeed, the data exchanged is interpreted according to a certain context[9], which is related to the environment of the business actors. In step 4a, the reseller sends a price proposal as part of the order. At this point, data-level mediation is needed at runtime in order to preserve the semantic

[9]Here, context refers to any piece of information required in order to correctly interpret the data.

consistency of the data exchanged between services. The "price" information has to be converted from a specific British context (in our example, GBP as a currency, included VAT and a scalefactor of 1) to a specific Japanese context (in our example, JPY as a currency, excluded VAT, and a scalefactor of 1000) to reach semantic correctness in the business process.

In the following we give some insight on how to use CSS as a support to enhance mediation of the data exchanged in a business process, on the basis of the context model developed in (30). We first remind the reader with the basics of the context model, before showing how its integration with CSS could be possible and to which extent this integration could ease the mediation task.

A Reminder on the Context Model

In (30), a context model dedicated to semantic, data-level mediation has been developed. This model helps making explicit the underlying semantic assumptions related to Web service input and output data. It has for objective to provide mediators with the required semantic information to reason about and mediate between alternative representations (i.e. following different contexts). This model builds around the notion of semantic object, which consists of a concept c from a domain ontology, an XML type t that grounds the data to a particular physical representation, a value v that represents the value of the data and is instanciated at runtime, and a set C of semantic attributes called context, that is made of other semantic objects. Context is organized as a tree structure that details the semantic information required for a correct data interpretation.

This model has been tested with the representation of monetary values (i.e. prices). The representation of prices depends on a tree of contextual attributes (i.e. date, date format, country, VAT rate, etc.) that play a role in the interpretation of a price instance and form together its "context". As an example, a simple "price" semantic objet p of value 15 expressed in Euro with VAT included is described as follows:

```
p = {ns:price, xsd:float, 15, {
    currency = Euro,
    VATincluded = true {VATRate = 19,6%},
    scaleFactor = 1}}
```

We show in the following how a semantic object in a specific context can also be described as a specific CSS member.

Additional Features of CSS

Context is built-up as a tree in order to refine the semantic description of context attributes, which complies with the definition of CSS that allows a dimension to be further refined via sets of conceptual subspaces. However, the use of CSS implies a couple of enhancements to the context model. The following features of CSS are relevant to mediation, and added as extensions to the original context model:

- the notion of **quality dimension**: involves describing context attributes as points on a scale, which is particularly useful for mediation purposes, as detailed hereafter.
- the notion of **prominence**: enhances mediation by allowing to setup specific prominence levels that play a role during the mediation process. For example, Web services may agree on a minimum prominence level, under which a quality dimension is considered as negligible for the mediation purpose.

An Insight on CSS Mediation with the Ordering Example

As demonstrated previously, in order to preserve the semantic consistency of our illustrative business process, we need to make explicit the different contextual aspects related to prices.

In our example, data-level mediation consists in applying various tranformation operations to the price value transmitted from the reseller to the producer, in order to converting context dimensions from their original values to the targeted values. Price currencies correspondences can be described by associating GBP, USD, EUR or JPY to values on the "currency" scale, corresponding to the actual relative values of these currencies with respect to each other. For instance, when USD is worth 1,2 Euro, then the USD is placed on 1 and Euro on 1,2 on the currency scale. Accordingly, scalefactors are placed on 1 and 1000. The VAT boolean is set to 1 if a VAT applies to the price, and the VAT rate is expressed as a float. In our example, we show that a price of value 15 expressed in GBP with a scale factor of 1 and VAT included. This context information is represented as a CSS member over the dimensions expressed previously:

$$Context_1 = (currency = GBP, sf = 1, VATincluded = true, (VATrate = UKrate))$$

The target context can also be expressed as a CSS member as follows:

$$Context_2 = (currency = JPY, sf = 1000, VATincluded = false, (VATrate = JPYrate))$$

Therefore, the data-level mediation task can be reduced to converting the "price" value according to the operations to be applied to the CSS dimensions for $Context_1$ to become equal to $Context_2$. In our example, in order to change the sf value from 1 to 1000 we need to multiply the price by 1000. In order to change the currency we need to convert the price from GBP to JPY according to their values on the CSS dimensions, and similarly for the VAT rates.

We have shown in this section that CSS could be utilized to simplify data-level mediation. It replaces complex conversion rules stored in a knowledge repository involved in the original context-based mediation work (30) with simple value conversions, according to the positions of context elements over the CSS dimensions. Also, the use of CSS adds the notion of prominence that is missing in the existing context model, and further refines the mediation strategy by explicitly stating the relative importance of contexts elements with respect to the mediation purpose.

13.7 Conclusions

In this paper, we proposed an approach to support fuzzy, similarity-based matchmaking between real-world context characteristics and predefined SWS capability descriptions by incorporating semantic context information on a conceptual level into symbolic SWS representations utilizing a metamodel for Conceptual Situation Spaces (CSS). By utilizing the CSS and its alignment to SWS technology, the most appropriate resources, whether data or services, for a given situation are identified based on the semantic similarity, calculated in terms of the Euclidean distance, between a given real-world situation and predefined resource descriptions as part of SWS capability representations. Consequently, by aligning CSS to established SWS frameworks, the expressiveness of symbolic SWS standards is extended with vector-based context information to enable fuzzy context-aware discovery of services and resources at runtime. Whereas current SWS frameworks such as WSMO and OWL-S address the allocation of distributed services for a given (semantically) well-described task, the CSS approach particularly addresses the similarity-based discovery of the most appropriate SWS task representation for a given context. To prove the feasibility of our approach, two proof-of-concept prototype applications were presented. Whereas the first one applies the CSS metamodel to enable context-adaptive resource discovery in the domain of E-Learning, the second one applies CSS to an E-Business scenario.

However, although our approach aims at solving Semantic Web (Services)-related issues such as the symbol grounding problem, several criticisms still have to be taken into account when applying CSS. While defining situations, respectively instances within a given CSS appears to be a straightforward process of assigning specific values to each quality dimension of a CSS, the definition of the CS itself is not trivial and strongly dependent on individual perspectives and subjective appraisals. Whereas the semantics of an object are grounded to metrics in geometrical vector spaces within a CS, the quality dimensions itself are subject to ones perspective and interpretation, which may lead to ambiguity issues. With regard to this, the approach of CSS does not appear to fully solve the symbol grounding issue but to shift it from the process of describing context instances to the definition of a CSS. Indeed, distinct semantic interpretations and vector-based groundings of each dimension may be applied by different individuals. Apart from that, whereas the size and resolution of a CS is indefinite, defining a reasonable CSS for a specific purpose or domain may become a challenging task. Nevertheless, distance calculation as major contribution of CSS, relies on the fact, that entities are described in the same geometrical space.

Consequently, CS-based approaches such as CSS may be perceived as step forward but do not fully solve the issues related to symbolic Semantic Web (Services)-based knowledge representations. Hence, future work has to deal with the aforementioned issues. For instance, we foresee to enable adjustment of prominence values to quality dimensions of a specific CSS to be accomplished by a user him/herself, in order to most appropriately suit his/her specific priorities and preferences regarding the resource allocation process, since the prioritization of dimensions is a highly individual and subjective process. In addition, it is intended to apply different distance metrics in order to evaluate the most appropriate similarity measure within CSS. Nevertheless, further research will be concerned with the application of our approach to further domain-specific situation settings.

References

[1] S. Arroyo and M. Stollberg. WSMO Primer. WSMO Deliverable D3.1, DERI Working Draft. Technical report, WSMO, 2004. http://www.wsmo.org/2004/d3/d3.1/.

[2] L. Cabral, J. B. Domingue, S. Galizia, A. Gugliotta, B. Norton, V. Tanasescu, and C. Pedrinaci. Irs-iii: A broker for semantic web services based applications. In *Proceeding of the 5th International Semantic Web Conference (ISWC2006)*, 2006.

[3] N. Choi, I.-Y. Song, and H. Han. A survey on ontology mapping. *SIGMOD Rec.*, 35(3):34–41, September 2006.

[4] E. Cimpian, A. Mocan, and M. Stollberg. Mediation enabled semantic web services usage. In *1st Asian Semantic Web Conference (ASWC2006), September 2006*, page 2006, 2006.

[5] A. Cregan. Symbol grounding for the semantic web. In E. Simperl, J. Diederich, and G. Schreiber, editors, *ESWC*, volume 275 of *CEUR Workshop Proceedings*, pages 429–442. CEUR-WS.org, 2007.

[6] J. Devore and R. Peck. Statistics: the exploration and analysis of data, 1999.

[7] A. K. Dey. Understanding and using context. *Personal and Ubiquitous Computing*, 5:4–7, 2001.

[8] S. Dietze and J. Domingue. Exploiting conceptual spaces for ontology integration. In *Proceedings of Workshop on Data Integration through Semantic Technology (DIST2008) @ 3rd Asian Semantic Web Conference 2008*, Bangkok, Thailand, 2007.

[9] S. Dietze, A. Gugliotta, and J. Domingue. A semantic web services-based infrastructure for context-adaptive process support. In *Proceedings of IEEE 2007 International Conference on Web Services (ICWS)*, 2007.

[10] S. Dietze, A. Gugliotta, and J. Domingue. Towards context-aware semantic web service discovery through conceptual situation spaces. In *CSSSIA '08: Proceedings of the 2008 international workshop on Context enabled source and service selection, integration and adaptation*, pages 1–8, New York, NY, USA, 2008. ACM.

[11] M. Ehrig and S. Staab. Qom - quick ontology mapping. pages 683–697. Springer, 2004.

0-8493-0052-5/00/$0.00+$.50

[12] M. Ehrig and Y. Sure. Ontology mapping - an integrated approach. pages 76–91. Springer Verlag, 2004.

[13] J. Euzenat, P. GuÃl'gan, and Valtchev. P.: Ola in the oaei 2005 alignment contest. In *Proceedings of the K-CAP Workshop on Integrating Ontologies*, pages 61–71, 2005.

[14] R. Felder and J. Spurlin. Index of learning styles questionnaire, 1997.

[15] R. M. Felder and L. K. Silverman. Learning and teaching styles in engineering education, 1988.

[16] D. Fensel, H. Lausen, A. Polleres, J. de Bruijn, M. Stollberg, D. Roman, and J. Domingue. Enabling semantic web services - the web service modelling ontology, 2006.

[17] A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, and L. Schneider. Sweetening ontologies with dolce. pages 166–181. Springer, 2002.

[18] P. Gärdenfors. *How to Make the Semantic Web More Semantic*, pages 17–34.

[19] P. Gärdenfors. *Conceptual Spaces: The Geometry of Thought*. The MIT Press, March 2000.

[20] F. Giunchiglia, P. Shvaiko, and M. Yatskevich. S-match: an algorithm and an implementation of semantic matching. In *In Proceedings of ESWS*, pages 61–75, 2004.

[21] S. Graf, S. R. Viola, Kinshuk, and T. Leo. Representative characteristics of felder-silverman learning styles: An empirical model. In *Proceedings of the IADIS International Conference on Cognition and Exploratory Learning in Digital Age, Barcelona, Spain, 2006*. IADIS Press, 2006.

[22] A. Haller, E. Cimpian, A. Mocan, E. Oren, and C. Bussler. Wsmx - a semantic service-oriented architecture. In *ICWS*, pages 321–328. IEEE Computer Society, 2005.

[23] S. Harnad. The symbol grounding problem. *CoRR*, cs.AI/9906002, 1999.

[24] C. Johnson and C. Orwig. What is learning style?, 1998.

[25] M. Kaipainen, P. Normak, K. Niglas, J. Kippar, and M. Laanpere. Soft ontologies, spatial representations and multi-perspective explorability. *Expert Systems*, 25(5):474–483, November 2008.

[26] E. Krause. *Taxicab Geometry*. Dover, 1987.

[27] D. L. Martin, M. Paolucci, S. A. McIlraith, M. H. Burstein, D. V. McDermott, D. L. McGuinness, B. Parsia, T. R. Payne, M. Sabou, M. Solanki, N. Srinivasan, and K. P. Sycara. Bringing Semantics to Web Services: The OWL-S Approach. In J. Cardoso and A. P. Sheth, editors, *SWSWPC*, volume 3387 of *Lecture Notes in Computer Science*, pages 26–42. Springer, 2004.

[28] P. Mitra, N. F. Noy, and A. R. Jaiswal. Ontology mapping discovery with uncertainty. In *in Fourth International Conference on the Semantic Web (ISWC-2005*, pages 537–547, 2005.

[29] E. Motta. An overview of the ocml modelling language. In *8th Workshop on Methods and Languages*, 1998.

[30] M. Mrissa, C. Ghedira, D. Benslimane, and Z. Maamar. A context model for semantic mediation in web services composition. In D. W. Embley, A. Olivé, and S. Ram, editors, *ER*, volume 4215 of *Lecture Notes in Computer Science*, pages 12–25. Springer, 2006.

[31] N. F. Noy and M. A. Musen. The prompt suite: Interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, (59):2003, 2003.

[32] M. Paolucci, N. Srinivasan, and K. Sycara. Expressing wsmo mediators in owls. In *Proceedings of the workshop on Semantic Web Services: Preparing to Meet the World of Business Applications held at the 3rd International Semantic Web Conference (ISWC 2004*, 2004.

[33] Y. Qu, W. Hu, and G. Cheng. Constructing virtual documents for ontology matching. In *In Proceedings of the 15th International World Wide Web Conference*, pages 23–31. ACM Press, 2006.

[34] M. Raubal. Formalizing conceptual spaces. In V. A. Vieu and L., editors, *Proc. of the Third International Conference on Formal Ontology in Information Systems*, Frontiers in Artificial Intelligence and Applications, pages 153–164. IOS Press, Amsterdam, NL, 2004.

[35] A. Schmidt and C. Winterhalter. C.: User context aware delivery of e-learning material: Approach and architecture. *Journal of Universal Computer Science (JUCS*, (10):28–36, 2004.

[36] A. Schwering. Hybrid model for semantic similarity measurement. In R. Meersman, Z. Tari, M. S. Hacid, J. Mylopoulos, B. Pernici, ö. Babaoglu, H. A. Jacobsen, J. P. Loyall, M. Kifer, and S. Spaccapietra, editors, *OTM Conferences (2)*, volume 3761, pages 1449–1465. Springer, 2005.

[37] P. Suppes, D. H. Krantz, R. Luce, and A. Tversky. *Foundations of Measurement*, volume 2: Geometrical, threshold and probabilistic representations. Academic Press, New York, 1989.

[38] S. R. Viola and S. Graf. Investigating relationships within the index of learning styles: A data-driven approach. *International Journal of Interactive Technology and Smart Education*, 4(1):7–18, 2007.

[39] H. w. Gellersen, A. Schmidt, and M. Beigl. Multi-sensor context-awareness in mobile devices and smart artifacts. *Mob. Netw. Appl*, 7:341–351, 2002.

[40] Z. Wu, K. Gomadam, A. Ranabahu, A. Sheth, and J. Miller. Automatic composition of semantic web services using process mediation. In *Proceedings of the 9th Intl. Conf. on Enterprise Information Systems ICES 2007*, 2007.