

# Conceptual Situation Spaces for Semantic Situation-Driven Processes

Stefan Dietze, Alessio Gugliotta, John Domingue  
Knowledge Media Institute,  
Open University,  
MK7 6AA, Milton Keynes, UK  
{s.dietze, a.gugliotta, j.b.domingue}@open.ac.uk

**Abstract.** Context-awareness is a highly desired feature across several application domains. Semantic Web Services (SWS) technologies address context-adaptation by enabling the automatic discovery of distributed Web services for a given task based on comprehensive semantic representations. Whereas SWS technology supports the allocation of resources based on semantics, it does not entail the discovery of appropriate SWS representations for a given situation. Describing the complex notion of a situation in all its facets through symbolic SWS representation facilities is a costly task which may never lead to semantic completeness and introduces ambiguity issues. Moreover, even though not any real-world situation completely equals another, it has to be matched to a finite set of parameter descriptions within SWS representations to enable context-adaptability. To overcome these issues, we propose Conceptual Situation Spaces (CSS) to facilitate the description of situation characteristics as members in geometrical vector spaces following the idea of Conceptual Spaces. CSS enable fuzzy similarity-based matchmaking between real-world situation characteristics and predefined situation descriptions. Following our vision, the latter are part of semantic Situation-Driven Process (SDP) descriptions, which define a composition of SWS Goals suitable to support the course of an evolving situation. Particularly, we refer to the WSMO approach for SWS. Consequently, our approach extends the expressiveness of WSMO by enabling the automatic discovery, composition and execution of achievable goals for a given situation. To prove the feasibility, we apply our approach to the domain of eLearning and provide a proof-of-concept prototype.

**Keywords:** Semantic Web, Conceptual Spaces, Semantic Web Services, WSMO.

## 1 Introduction

*Context* is a highly important aspect in information systems (IS) and has been subject to intensive research across a wide variety of application domains throughout the last decade [5] [14] [26]. Context-adaptation can be defined as the ability of IS to adapt their behavior to multiple possible situations. A *situation* is a complex combination of features; i.e. its parameters. Usually, situations evolve throughout the course of a *process*. The latter, therefore, can be best perceived as a sequence of intermediate situations, leading from an initial to a final situation.

To consider the relation between situations and processes, we introduce the notion of *Situation-Driven Processes (SDP)*, which consider a process context from two perspectives: the user and the system perspective. Whereas the user perspective is concerned with user situations and goals throughout the course of a process, the system perspective takes into account the resources - data and services - which are required to support each user goal in a given situation.

*Semantic Web Services (SWS)* [9] address the system perspective of a process since they enable the automatic discovery and selection of distributed resources - services and data exposed via Web services - for a particular goal. Current results of SWS research are available in terms of reference ontologies, such as OWL-S [22] and WSMO [28], as well as comprehensive frameworks (see DIP project<sup>1</sup> results). However, the definition and discovery of the most appropriate goal for a given process situation - i. e. the user perspective of a process - remains a challenging task. Current SWS approaches do neither explicitly consider the notion of a situation nor do they facilitate the grounding of purely symbolic SWS representations to a conceptual level to fully support semantic meaningfulness [3][21]. Therefore, we claim that fuzzy matchmaking methodologies are crucially required, to match a possibly infinite number of (real-world) situation characteristics to a finite set of predefined parameter instance representations as part of SDP and consequently SWS descriptions.

*Conceptual Spaces (CS)*, as introduced by Gärdenfors [10] [11], follow a theory of describing entities at the conceptual level in terms of their natural characteristics, similar to natural human cognition to avoid the symbol grounding issue. CS enable the representation of objects as vector spaces within a geometrical space which is defined by 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 description of such knowledge through symbolic representations. Even though several criticisms have to be taken into account when utilizing CS (Section 6), they are considered to be a viable option for knowledge representation.

To enable the use of SWS technology as part of SDP, we propose *Conceptual Situation Spaces (CSS)*, which are mapped to the established SWS framework WSMO and enable the discovery of appropriate SWS representations capable to achieve a given task within a particular situation. Extending merely symbolic SWS descriptions based on WSMO with context information at a conceptual level through CSS enables fuzzy, 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. Consequently, the expressiveness of SWS facilities, respectively WSMO, is extended through CSS in order to enable fuzzy matchmaking mechanisms when allocating resources for a given situation.

---

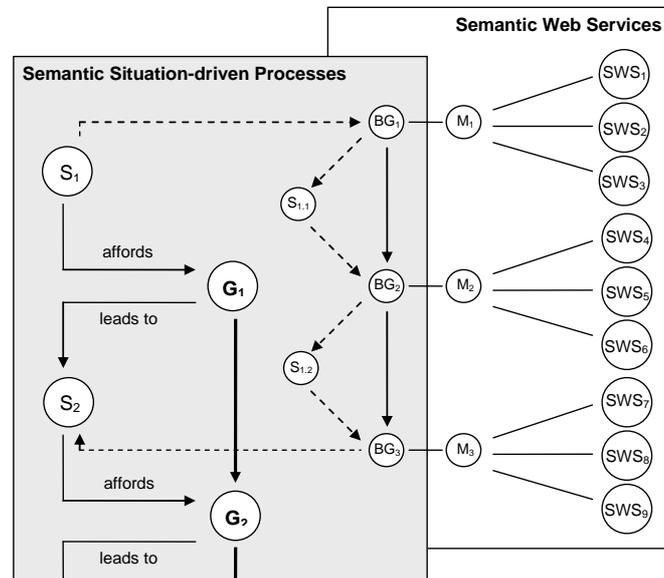
<sup>1</sup> DIP Project: <http://dip.semanticweb.org>

Since a situation always occurs within a specific domain setting, and can be described based on domain-context specific entities only, the CSS metamodel considers domain-specific derivations. In this paper, we exemplarily refer to the e-Learning domain. To prove the feasibility of our approach, we provide a proof-of-concept prototype that uses CSS to describe learning styles, following the Felder-Silverman Learning Style theory [6], as particular learning situation parameter utilized within comprehensive SDP models.

The remainder of the paper is organized as follows: Section 2 introduces our vision of SDP as motivations for CSS, which are introduced in Section 3. Section 4 illustrates the application of CSS to the eLearning domain as utilized within a prototype application which is explained in Section 5. Finally, we discuss and conclude our work in Section 6.

## 2 Motivation: Situation-Driven Processes

Following our vision, a SDP consists of *Situations* ( $S$ ) and *Goals* ( $G$ ), where a Goal represents a particular activity within a process from a user perspective and links two situations leading from an initial situation to a desired situation. Each Goal is supported by a set of *Brokered Goals* ( $BG$ ) which are achievable and brokered via a SWS Execution Environment. Since we refer to WSMO as SWS reference implementation,  $BG$  are derived WSMO Goals (Figure1).



**Fig. 1.** Utilizing SWS to support Situation-Driven Processes

$BG$  and  $SWS$  support the system perspective of a process, since they are linked (via mediators) to semantic descriptions of available Web services that, once discovered and selected, will provide the appropriate resources needed to progress a situation.

For instance, one BG could be aimed at providing required information out of specific databases whereas another aims at computing a specific calculation, such as the current stock of a specific article. Note that the achievement of BG subsequently modifies the actual situation until the desired final situation is reached (Figure 1).

Utilizing the OCML knowledge modelling language [20], the SDP metamodel has been formalized into an ontology (*SDPO*), which is aligned to an established and well known foundational ontology: the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) [13] and, in particular, its module Descriptions and Situations (D&S) [12]. WSMO is adopted to enable the description of SWS, as well as data resources in terms of its main elements – goals, Web services, ontologies, and mediators. As a result, the SDP metamodel extends the expressiveness of WSMO representation facilities by incorporating WSMO Goals into meaningful situation-based process context descriptions. It is important to note that process situations are highly dependent on the domain and nature of a process, since each domain emphasizes different situation parameters. Therefore, we foresee multiple domain-specific derivations of the introduced meta-model.

However, to fully enable situation-aware discovery of resources through SDP and SWS, the following shortcomings have to be considered:

- I1. *Lack of explicit notion of context*: current SWS technology does not entirely specify how to represent domain contexts. For example, WSMO [28] addresses the idea of context: Goal and web service represent the user and provider local views, respectively; the domain ontologies define the terminologies used in each view; and the mediators are the semantic bridges among such distinct views. However, WSMO does not specify what a context description should define and how the context elements should be used.
- I2. *Symbolic Semantic Web representations lack grounding to conceptual level*: the symbolic approach, i.e. describing symbols by using other symbols, without a grounding in the real world, of established SWS, and Semantic Web representation standards in general, leads to ambiguity issues and does not 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 [3][21].
- I3. *Lack of fuzzy matchmaking methodologies*: Describing the complex notion of a specific situation in all its facets is a costly task and may never reach semantic completeness. Whereas not any situation and situation parameter completely equals another, the number of (predefined) semantic representations of situations and situation parameters is finite. Therefore, a possibly infinite set of given (real-world) situation characteristics has to be matched to a finite set of predefined parameter instance representations which are described within an IS. Consequently, fuzzy classification and matchmaking techniques are required to classify a real-world situation based on a limited set of predefined parameter descriptions.

### 3 Conceptual Situation Spaces

To address I1 – I3 introduced in Section 2, we propose *Conceptual Situation Spaces* (CSS) applying CS to represent situations as part of SDP.

#### 3.1. CSS Metamodel

CSS enable the description of a particular situation as a member of a dedicated CS. Referring to [11],[17], [24], we define a CSS (*css:Conceptual Situation Space* in Figure 2) as a vector space:

$$C^n = \{(c_1, c_2, \dots, c_n) | c_i \in C\}$$

with  $c_i$  being the quality dimensions (*css:Quality Dimension*) of  $C$ . Please note, that we do not differentiate between domains, as sets of integral dimensions [11], but 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 [24]. In such a case, the particular quality dimension  $c_j$  is described by a set of further quality dimensions with

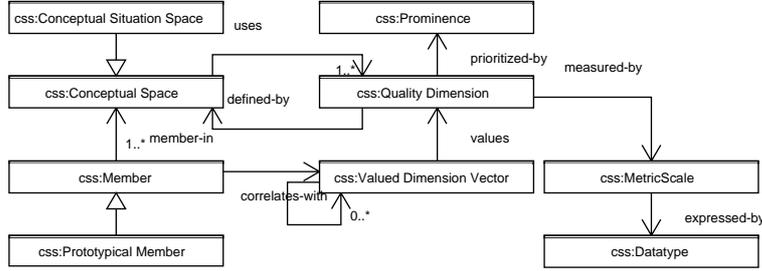
$$c_j = D^n = \{(d_1, d_2, \dots, d_n) | d_k \in D\}.$$

In this way, a CSS may be composed of several subspaces and consequently, the description granularity of a specific situation can be refined gradually. This aspect of CSS corresponds to the approach Dolce D&S [12], utilized within SDPO, to gradually refine a particular description by using parameters where each parameter can be described by an additional description. To reflect the impact of a specific quality dimension, we consider a prominence value  $p$  (*css:Prominence*) for each dimension. Therefore, a conceptual space is defined by

$$C^n = \{(p_1 c_1, p_2 c_2, \dots, p_n c_n) | c_i \in C, p_i \in P\}$$

where  $P$  is the set of real numbers. However, the usage context, respectively the domain, of a particular CSS strongly influences the ranking of its quality dimensions. For instance, within a learning situation the competencies of a particular learner may be more important whereas in a business situation, the costs of a particular task may be weighted higher. This clearly supports our position of describing distinct CSS explicitly for specific domains only.

Particular members (*css:Member*) in the CSS are described through a set of valued dimension vectors (*css:Valued Dimension Vectors*). Moreover, referring to [11] we consider prototypes which represent specific prototypical members (*css:Prototypical Member*) within a particular space. Prototypical members are utilised to categorize a specific CSS member as they enable the classification of any arbitrary member  $m$  within the same space, by simply calculating the Euclidean distances between  $m$  and all prototypical members to identify the closest neighbours of  $m$ . For instance, given a CS to describe apples based on their shape, taste and colour, a green apple with a strong and fruity taste may be close to a prototypical member representing the typical characteristics of the Granny Smith species. Figure 2 depicts the CSS metamodel.



**Fig. 2.** The CSS metamodel.

The metamodel introduced above has been formalized into a *Conceptual Situation Space Ontology (CSSO)*, utilizing OCML [20]. In particular, each of the depicted entities is represented as a concept within CSSO whereas associations are reflected as their properties in most cases. The correlation relationship between several quality dimensions indicates whether two dimensions are correlated or not. For instance, when describing an apple the quality dimension describing its sugar content may be correlated with the taste dimension. Information about correlation is expressed within the CSSO through axioms related to a specific quality dimension instance.

Semantic similarity between two members of a space can be perceived as a function of the Euclidean distance between the points representing each of the members. Applying a formalization of CS proposed in [24] to our definition of a CSS, we formalize the Euclidean distance between two members in a CSS as follows. Given a CSS definition  $C$  and two members represented by two vector sets  $V$  and  $U$ , defined by vectors  $v_0, v_1, \dots, v_n$  and  $u_1, u_2, \dots, u_n$  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 [4][24] 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 to be calculated as follows:

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

where  $\bar{u}$  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:

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

### 3.2. Aligning CSS and SWS

CSS are aligned to WSMO to support the automatic discovery of the most appropriate goal representation for a specific process situation. Figure 3 depicts the main relationships between CSS and WSMO.

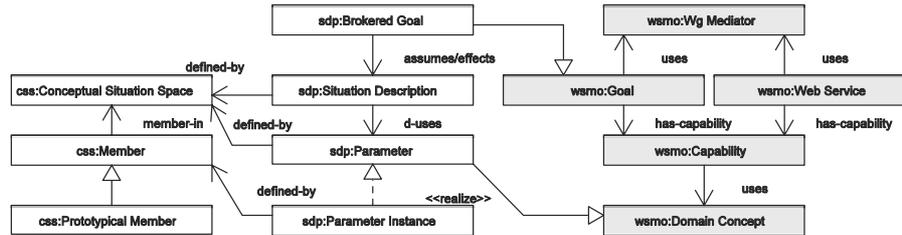


Fig. 3. Alignment of CSS and WSMO

Grey colored concepts in Figure 3 represent concepts of WSMO [28]. A goal description (*wsmo:Goal*, *sdp:Brokered Goal*) utilizes particular domain concepts (*wsmo:Domain Concept*, *sdp:Parameter*) to semantically describe its capabilities, i.e. its assumptions, effects, preconditions and postconditions. A WSMO runtime reasoning engine utilizes capability descriptions to identify SWS (*wsmo:Web Service*) which suit a given Goal. In contrast, the preliminary selection of the most appropriate goal description for a given situation is addressed by classification of situation parameters through CSS. For instance, given a set of real-world situation parameters, described as members in a CSS, their semantic similarity with predefined prototypical situation descriptions (*css:Prototypical Member*) is calculated. Given such a classification of a particular real-world situation, a goal representation which assumes matching prototypical parameter instances can be selected and achieved through the reasoning engine.

## 4 Spanning a Conceptual Learning Situation Space

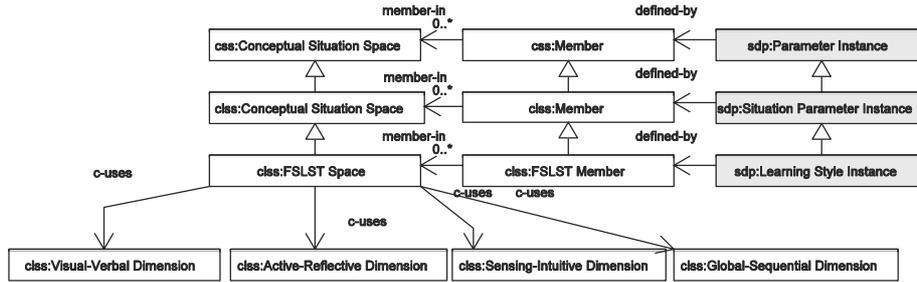
As Gärdenfors states in [11], the prioritization of certain quality dimensions within a CS is highly dependent on the context of the space. The same applies to situations which are described within a CSS. In order to validate the applicability of our approach, we defined a CSS for the eLearning domain, a *Conceptual Learning Situation Space (CLSS)*.

Since situation parameters usually are complex theoretical constructs, each parameter itself is described as a CSS subspace (Section 3). In this Section we focus exemplarily on the representation of one parameter, which is of particular interest within the eLearning 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 [19]. Whereas each individual has his/her distinct learning style, it affects the learning process.

We refer to the *Felder-Silverman Learning Style Theory (FSLST)* [6], which is supposed to be suitable to describe learning styles within computer-aided educational environments. It is important to note that distinct theories can be applied to describe each situation parameter, and FSLST just serves as example to illustrate the application of CSS in this paper. Following FSLST, a learning style can be described by four quality dimensions [6]. In short, the *Active-Reflective* dimension describes whether or not a learner prefers to interact with learning material, whereas the *Sensing-Intuitive* dimension, describes whether a learner tends to focus on facts and details (Sensing) rather than abstract theories (Intuitive). The *Visual-Verbal* dimension obviously covers, whether a learner prefers visual rather than verbal learning material, while the *Global-Sequential* dimension describes, whether a learner tends to learn gradually in small steps (Sequential) rather than following a holistic learning process marked by large learning leaps. Literature shows [8][15][27], that these dimensions can be assumed to be virtually linearly independent. Consequently, we define a *CLSS L* with 4 quality dimensions  $l_i$ :

$$L^4 = \{(l_1, l_2, l_3, l_4) | l_i \in L\}$$

Figure 4 depicts the key concepts describing  $L$  as subspace (*cls:FSLST Space*) within CLSS representing FSLST.



**Fig. 4.** Key concepts of an ontology representing the FSLST as particular subspace.

Moreover, Figure 4 depicts the alignment of subspace  $L$  (*cls:FSLST Space*) with SDP (Section 2), represented via grey-colored concepts. Symbolic representations of parameter instances within the SDP are defined by members within CLSS-based representations (*cls:Member*), where a particular learning style is a specific SDP parameter instance (*sdp:Learning Style Instance*) and is defined by a particular member (*cls:FSLST Member*). The metric scale, datatype, value range and prominence value for each dimension  $l_i$  are presented in Table 1:

**Table 1.** Quality dimensions  $l_1 - l_4$  of CLSS  $L$  describing learning styles following FSLST.

	Quality Dimension	Metric Scale	Datatype	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 1, each quality dimension is ranked on an interval scale with its value range being an integer between -11 and +11. This particular measurement scale was derived from an established assessment method, the Index of Learning Styles (ILS) questionnaire defined by Felder and Soloman [7], aimed at identifying and rating a particular learning style of an individual.

The authors would like to highlight, that prominence values have been assigned which rank the first ( $I_1$ ) and the third dimension ( $I_3$ ) higher than the other two, since these have a higher impact on the context of the learning situation, which is focused on the aim to deliver appropriate learning material to the learner. Since dimensions  $I_1$  and  $I_3$  are highly critical for context-adaptation and SWS discovery (Section 5), a higher prominence value was assigned. It is obvious, that the assignment of prominence values is a highly subjective process, strongly dependent on the purpose, context and individual preferences. Therefore, future work is aimed at enabling learners to assign rankings of quality dimensions themselves individually.

To classify an individual learning style (*class:FSLST Member*), we define prototypical members (*class:FSLST Prototypical Member*) 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 [8][27]. Particularly, we refer to correlation coefficients identified in [27] which led to the description of the following five prototypical members and their characteristic vectors:

**Table 2.** Prototypical learning styles defined as prototypical members in the CLSS ontology.

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

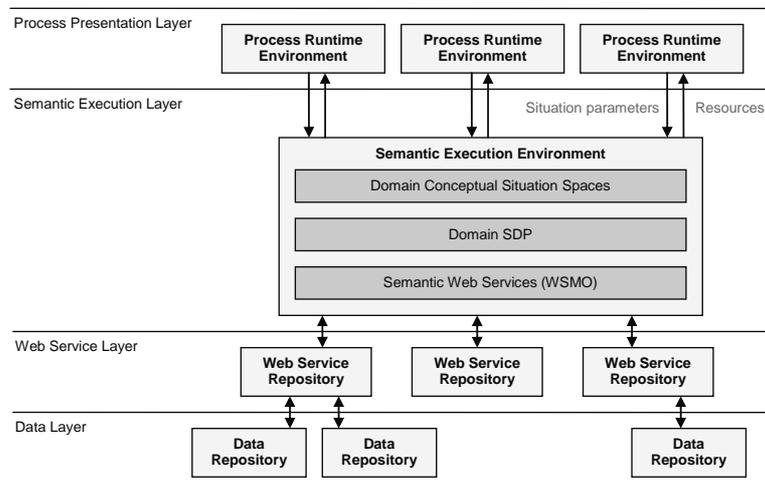
## 5 Context-Adaptive Composition and Accomplishment of SDP

To prove the feasibility of our approach, a proof-of-concept prototype application has been provided which applies CSS (Section 3) to the domain of eLearning<sup>2</sup>. The lifecycle of a SDP instance consists of the following three stages:

- S1. Classification of a situation given a domain-specific CSS;
- S2. Composition of SDP as sequence of BG which satisfy a given situation;
- S3. Runtime execution of SDP, in terms of BG achievements.

Figure 5 depicts the architecture used to support reasoning on SDP and CSS through a Semantic Execution Environment (SEE) which is implemented through IRS-III [2] in our case.

<sup>2</sup> The application and ontologies have been developed in the context of the EU FP6 project LUISA [18].



**Fig. 5.** Architecture to support runtime reasoning on SDP/CSS.

SEE makes use of WSMO-based SWS descriptions, semantic representations of the SDP and CSS metamodels, and particularly their derivations for the eLearning domain. Multiple runtime environments interact with SEE to provide information about the current real-world situation on the one hand and to present and accomplish SDP-based processes on the other hand.

Starting from a set of real-world situation parameters, their semantic similarity with a set of prototypical situation parameters (Section 4) is calculated to support *S1*. Given such a classification of an identified prototypical situation, the appropriate SDP (Section 2) is composed as a sequence of Goals and supportive Brokered Goals (BG) to accomplish *S2*. Finally, SEE accomplishes *S3* by discovering and orchestrating appropriate Web services, which show the capabilities to suit the Brokered Goals associated with the user Goals within the SDP.

### 5.1. Context Classification through CSS

To enable a description of a situation, a learner is first authenticated in order to retrieve information about his/her actual preferences. During the following situation refinement, learners are enabled to define situation-specific parameters, such as his/her current learning aim or the available learning time, while other parameters are calculated automatically, such as the competency gap between the current learner situation and the desired aim. Referring to the learning style as particular situation parameter defined by the CLSS subspace  $L$  in Section 4, 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  $V$  can be calculated disregarding a Z-transformation (Section 3) for each vector as follows:

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

The calculation of similarities using the formula shown above is performed by a standard Web service that is exposed as SWS. Given a particular CSS description, a member (representing a specific parameter instance) as well as a set of prototypical members, similarities are calculated 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 CLSS subspace  $L$  (*class:FSLST Space*) 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 introduced to individual learners by utilizing the ILS Questionnaire [7] as assessment method. Calculating distances between  $U$  and each of the prototypes described in Table 2 of Section 4 led to the following results:

**Table 3.** Euclidean distances between  $U$  and prototypical learning styles.

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 3, the lowest Euclidean distance applies to  $PI$ , indicating a rather active and visual learning style described as in Table 2 of Section 4. Utilizing such similarity-based classifications enables the gradual refinement of learning situation description and fuzzy matchmaking between real-world situation parameters, such as  $U$ , and prototypical parameters such as  $PI$ .

## 5.2. Situation-driven Composition and Accomplishment of SDP

The authors would like to highlight, that not only processes (SDP) but also entire application scenarios are accomplished by achieving BG at runtime. Given a classified situation description, the SEE first identifies semantic Goal representations (*sdp:Brokered Goal*) and finally discovers and orchestrates Web services to suit the given runtime situation. Given the alignment of CSS and WSMO (Section 3.2) we address the issue of discovering the most appropriate BG representation for a given situation by enabling a classification of an individual situation based on semantic similarity calculations as described in the previous sections.

For instance, referring to Section 5.1, given a classified learning style, together with classifications of all further situation parameters, a BG representation (*sdp:Brokered Goal*) which assumes equivalent situation parameter instances

(*sdp:Parameter Instance*) is achieved at runtime through SEE. Consequently, following the alignment of CSS and WSMO, 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.

Following the lifecycle stages *S1-S3*, the application supports the automatic composition of SDP (*S2*), their transformation into non-semantic process metadata manifestations for two metadata standards - ADL SCORM [1] and IMS LD [16] - and the accomplishment of the SDP-based process (*S3*). Process composition is accomplished by a Web service which composes a SDP as a set of Goals and Brokered Goals, which show the appropriate effects and assumptions to progress from an initial situation *Si* to the final situation *Sf*. The service takes into account the situation parameters of *Si*, for instance the available learning time of the learner or his/her learning style (Section 4). A particular parameter, the actual aim of a learner, is linked to a set of desired competencies which consequently are part of the final situation *Sf*. Composition functionalities could be provided by different Web services following distinct composition strategies. At runtime, the most appropriate composition and transformation service for a given situation is selected and invoked automatically by the SEE. The achievement of BG at runtime considers the actual learning situation parameters, and enables a more fine-grain adaptation to the actual learning context. At runtime, a process is executed (*S3*) whether through a metadata standard specific runtime environment or a runtime environment dedicated to interpret semantic SDP models. Whereas a web-based user interface is utilized to interpret and present semantic process instances, software clients of the RELOAD-project [25] are utilized to present dynamically transformed XML-manifestations following the IMS LD and ADL SCORM standard.

## 6 Discussion and Conclusions

We proposed the CSS approach aimed at the classification of situations to enable the automatic discovery, composition and accomplishment of SWS-based Goal representations to suit a given process situation. We introduced the notion of a Situation-Driven Process (SDP), which perceives a process from two perspectives, the user perspective which describes a process in terms of user Goals and situations and the system perspective, which links user Goals to automatically achievable SWS Goals. Whereas current SWS frameworks such as WSMO and OWL-S support the system perspective of a process by enabling 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 situation. Consequently, by aligning CSS to established SWS frameworks, the expressiveness of symbolic SWS standards is extended with context information on a conceptual level to enable fuzzy context aware delivery of resources at runtime. Deriving the CSS metamodel for specific domains enables the composition of domain-specific SDP at runtime while considering the specific characteristics of the runtime situation. Based on mappings with domain-specific process metadata

standards, a semantic SDP based process model can be transformed into non-semantic metadata standards to support interoperability of a process.

To prove the feasibility of our approach, a proof-of-concept prototype was described, which applies CSS to the eLearning domain. Whereas the Felder-Silverman Learning Style Theory (FSLST) was exemplarily represented as CSS, the authors would like to highlight that distinct theories could be applied to represent situation parameters. In this paper, FSLST just serves the purpose to illustrate the application of CSS but is not explicitly supported by the authors. The application supports the automatic classification of a situation based on similarity calculation within CSS, the situation-driven composition of semantic process instances following the SDP approach at runtime, their transformation into non-semantic metadata standards and the runtime execution of a SDP-based model by automatically achieving SWS to retrieve resources which suit the actual process situation.

However, even though our approach aims at solving SWS-related issues such as the symbol grounding problem, several criticisms still have to be taken into account. Whereas defining objects, 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 a CSS itself is not trivial at all and strongly dependent on individual perspectives and appraisals. Whereas semantics of an object are grounded to metrics in geometrical vector spaces within a CSS, the quality dimensions themselves are subject to ones perspective what may lead to ambiguity issues. Consequently, the approach of CSS does not appear to completely solve the symbol grounding issue but to shift it from the process of describing instances to the definition of a CSS. Apart from that, whereas the size and resolution of a CSS is indefinite, defining a reasonable CSS for a specific context may become a challenging task. Moreover, distance calculation as major contribution of the CSS approach always relies on the fact, that objects are described in the same geometrical space. Consequently, CSS may be perceived as step forward but do not fully solve the issues related to symbolic Semantic Web (Services)-based knowledge representations.

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 preferences, since prioritization of dimensions is a highly individual and subjective process. Besides that, we consider the representation of further situation parameters as CSS subspaces.

The authors are aware, that the current prototype applies SDP/CSS to one domain only. However, we are strongly convinced that applying the idea of SDP in further process domains is feasible, since processes across several domains share similar notions, as well as concepts and have to deal with related issues, such as process design, process resource allocation, and context-sensitivity. Therefore, further research will be concerned with the application of our approach to further domains. It is apparent, that our approach requires an initial effort to produce domain models following SDP/CSS which can be instantiated at runtime. However, once these derivations are available, these can be reused across multiple application settings to enable context-aware runtime composition and accomplishment of processes.

## 7 References

- [1] Advanced Distributed Learning (ADL) SCORM 2004 Specification (<http://www.adlnet.org>).
- [2] Cabral, L., Domingue, J., Galizia, S., Gugliotta, A., Norton, B., Tanasescu, V., Pedrinaci, C.: IRS-III: A Broker for Semantic Web Services based Applications. In proceedings of the 5<sup>th</sup> International Semantic Web Conference (ISWC 2006), Athens, USA (2006).
- [3] Cregan, A. (2007), Symbol Grounding for the Semantic Web. 4th European Semantic Web Conference 2007, Innsbruck, Austria.
- [4] Devore, J., Peck, R., (2001) Statistics - The Exploration and Analysis of Data, 4th ed. Pacific Grove, CA: Duxbury, 2001.
- [5] Dietze, S., Gugliotta, A., Domingue, J., (2007). A Semantic Web Services-based Infrastructure for Context-Adaptive Process Support. Proceedings of IEEE 2007 International Conference on Web Services (ICWS), Salt Lake City, Utah, USA.
- [6] Felder, R. M., Silverman, L. K. (1988). Learning and Teaching Styles in Engineering Education, *Engineering Education*, 78, 1988, pp. 674–681. Preceded by a preface in 2002: <http://www.ncsu.edu/felderpublic/Papers/LS-1988.pdf>.
- [7] Felder, R. M. and Soloman, L. K. (1997) Index of Learning Styles Questionnaire. Online version <http://www.engr.ncsu.edu/learningstyles/ilsweb.html> (retrieved October, 2007).
- [8] Felder, R.M. and Spurlin, J., (2005). Applications, Reliability and Validity of the Index of Learning Styles. *International Journal on Engineering Education*, Vol. 21, No. 1, pp. 103-112.
- [9] Fensel, D., Lausen, H., Polleres, A., de Bruijn, J., Stollberg, M., Roman, D., Domingue, J. (2006): *Enabling Semantic Web Services – The Web service Modelling Ontology*, Springer 2006.
- [10] Gärdenfors, P. (2000), *Conceptual Spaces - The Geometry of Thought*. MIT Press, 2000.
- [11] Gärdenfors, P. (2004), How to make the semantic web more semantic. In A.C. Vieu and L. Varzi, editors, *Formal Ontology in Information Systems*, pages 19–36. IOS Press, 2004.
- [12] Gangemi, A., and Mika, P. (2003), Understanding the Semantic Web through Descriptions and Situations. In Meersman, R.; Tari, Z.; and et al., D. S., eds., *Proceedings of the On The Move Federated Conferences (OTM'03)*, LNCS. Springer Verlag.
- [13] Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., Schneider, L.(2002), Sweetening Ontologies with DOLCE. In: A. Gómez-Pérez , V. Richard Benjamins (Eds.) *Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web: 13th International Conference, EKAW 2002, Sigüenza, Spain, October 1-4, 2002*.
- [14] Gellersen, H-W., Schmidt, A. and Beigl, M. (2002): Multi-Sensor Context-Awareness in Mobile Devices and Smart Artefacts. *ACM journal Mobile Networks and Applications (MONET)*, Vol. 7, No. 5. October 2002.
- [15] Graf, S., Viola, S.R., Kinshuk, and Leo, T. (2006). Representative Characteristics of Felder-Silverman Learning Styles: An Empirical Model. Proceedings of the IADIS International Conference on Cognition and Exploratory Learning in Digital Age (CELDA 2006), Barcelona, Spain, December 8-10.
- [16] IMS Learning Design Specification (<http://www.imsglobal.org>).
- [17] Keßler, C. (2006). Conceptual Spaces for Data Descriptions. The Cognitive Approach to Modeling Environments (CAME), Workshop at GIScience 2006, Münster, Germany. SFB/TR 8 Report No. 009-08/2006: 29-35.
- [18] LUISA Project - Learning Content Management System Using Innovative Semantic Web Services Architecture (<http://www.luisa-project.eu/www/>).
- [19] Johnson, C., Orwig, C.(1998), What is Learning Style?, <http://www.sil.org/LinguaLinks/LanguageLearning/OtherResources/YorLrnngStylAndLn ggLrnng/WhatIsALearningStyle.htm>.

- [20] Motta, E. (1998). An Overview of the OCML Modelling Language, the 8th Workshop on Methods and Languages, 1998.
- [21] Nosofsky, R. (1992), Similarity, scaling and cognitive process models, Annual Review of Psychology 43, pp. 25- 53, (1992).
- [22] Object Management Group: Business Process Modelling Notation Specification (<http://www.omg.org/docs/dtc/06-02-01.pdf>).
- [23] OWL-S 1.0 Release. <http://www.daml.org/services/owl-s/1.0/>.
- [24] Raubal, M. (2004). Formalizing Conceptual Spaces. in: A. Varzi and L. Vieu (Eds.), Formal Ontology in Information Systems, Proceedings of the Third International Conference (FOIS 2004).Frontiers in Artificial Intelligence and Applications 114, pp. 153-164, IOS Press, Amsterdam, NL.
- [25] Reload Project (<http://www.reload.ac.uk/>).
- [26] Schmidt, A., Winterhalter, C. (2004). User Context Aware Delivery of E-Learning Material: Approach and Architecture, Journal of Universal Computer Science (JUICS) vol.10, no.1, January 2004.
- [27] Viola, S. R., Graf, S., Kinshuk, and Leo T. (2007). Investigating Relationships within the Index of Learning Styles: A Data-Driven Approach. International Journal of Interactive Technology and Smart Education, Vol. 4, No. 1, pp. 7-18.
- [28] WSMO Working Group, D2v1.0: Web service Modeling Ontology (WSMO). WSMO Working Draft, (2004). (<http://www.wsmo.org/2004/d2/v1.0/>).