Personalization of Semantic Web Services*

Freddy Lécué

The University of Manchester
Booth Street East, Manchester, UK
{(firstname.lastname)@manchester.ac.uk}

Abstract. Nowadays web users have clearly expressed their wishes to receive and interact with personalized services directly. However, existing approaches, largely syntactic content-based, fail to provide robust, accurate and useful personalized services to its users. Towards such an issue, the semantic web provides enabling technologies to annotate and match services' descriptions with a users' features, interests and preferences, thus allowing for more efficient access to services and then information. The aim of our work, part of service personalization, is on automated instantiation of services which is crucial for advanced usability i.e., how to prepare and present services ready to be executed while limiting useless interactions with users? To this end, we exploit Description Logics reasoning through semantic matching to i) identify useful parts of a user profile that satisfy services requirements (i.e., input parameters) and ii) compute the description required by a service to be executed but not provided by the profile. Finally, the scalability of our approach has been evaluated through its integration in the service consumption of the EC-funded project SOA4All.

Key words: Semantic web, web service, Personalization, Automated reasoning.

1 Introduction

Personalization in web-based applications [1], as a global tendency nowadays, aims at alleviating the burden of information overload by tailoring the information presented based on an individual and immediate user's needs. Between the numerous examples that can be found all across the web, we can highlight the proliferation of personalized home sites, such as iGoogle (http://www.google.com/ig) or netvibes (http://www.netvibes.com), but also the fact that many other web applications of different kinds treat user configuration as one of their most prominent characteristics. From collaborative [2] and content [3] to hybrid-based [4], various personalization techniques have been introduced, depending on the data they manipulate and personalization levels they achieve. In most of these approaches, the user profile, as a collection of data modelling the user extended with its interests, its preferences and context, is a prominent element to ensure accurate and efficient personalized access to information.

In recent years, web service [5], as an emergent technology to consume information on the web, has benefited from research progress in web personalization. Indeed, many approaches addressing user-centric and preference-based consumption of services and

* Foundation Project: Supported by European Commission VII Framework IP Project Soa4All.

more specially their publication [6], discovery [7], selection [8] and execution [9] have emerged. The possibility to customize their results [10] even goes further by giving the users the chance to experience those services in a personalized fashion, which is prominent in order to permit the users to fulfil their desires more suitably.

However most of these approaches ensure personalization by collecting and analizing syntactic content of user profile and services description e.g., [11]. This under specification limits the accuracy of personalization and their automation. Towards this issue, the *semantic web* [12], where the semantic content of the information is tagged using machine-processable languages, provides many advantages over the current "formatting only" version of the web, its services and users. OWL [13], as one of its *Web Ontology Language* which is based on Description Logics (DLs) [14], aims at modelling knowledge on the web through ontologies i.e., formal conceptualization of a particular domain. Therefore, services with their functionalities (i.e., input and output parameters, preconditions, effects and invariants) and user profile with their interests, preferences can be both annotated and then enhanced using semantic descriptions. Such annotations are one important features to enable reasoning on services and user profile descriptions, hence automation of personalized tasks such as the consumption of services.

In this work, we address automated instantiation of services, part of personalization in service consumption, which is crucial for advanced usability. Service instantiation, which is between in selection and execution, aims at preparing and presenting preselected services ready to be executed while limiting useless interactions with users. To this end, execution-time constraints attached to services descriptions are required to be satisfy before their execution. Most of existing approaches [10] undervalue this issue by rarely considering suitable and efficient methods for such a personalization level. Contrary to the latter that consider several levels of interaction to manually collect information (from the users) required by input parameters of services to be executed, we consider automation of this process through semantic instantiation of services.

By addressing the latter, we thus aim at i) improving and easing the user interaction with services, beneficial for both parties and ii) better supporting the user by anticipating her needs. To reach the goal of automated and personalized consumption of services and more specially to suggest accurate and personalized presentations of services to users, we benefit from the semantic augmentation of service and user profile descriptions. In this direction, we define a framework, where potential matching between both descriptions is defined as a reasoning task to be solved for service instantiation (in the rest of the paper we refer to it as service personalization and adaptation). The semantic matching, core of our approach, exploits standard DL reasoning [15, 16] and abduction [17] to i) adapt services to the user by identifying useful parts of its profile that satisfy the service requirements (i.e., input parameters) and ii) compute the descriptions required by a service to be executed but not provided by the user profile.

The remainder of this paper is organised as follows. Section 2 briefly reviews i) DL reasoning and abduction, both required for automated personalization, ii) semantic web services and iii) semantic user profiles. Section 3 presents our approach to personalize semantic web services. Section 4 presents details about the prototype implementation and reports some experiment results. Section 5 briefly comments on related work. Finally Section 6 draws some conclusions and talks about possible future directions.

2 Background

In this section we review i) DL as a semantic language and abduction reasoning to perform personalization. Then we remain the definitions of ii) semantic web service and iii) semantic user profile, as core elements in our approach.

2.1 Description Logic and Abduction Reasoning

The model we considered to represent semantics of services and user profiles is provided by an ontology. In more detail, we focused on DL as a formal knowledge representation language to define ontologies since the latter offers good reasoning support for most of its expressive families and compatibility to current W3C standards e.g., OWL. Terminological Box $\mathcal{T}$ (or TBox i.e., intentional knowledge) and Assertional Box $\mathcal{A}$ (or ABox i.e., extensional knowledge) are core elements to represent knowledge in DL systems. In the following, we will focus on the TBox $\mathcal{T}$ (see Fig. 1 as an example) that supports different level of inference by means of DL reasoning. As a trade-off between expressivity and complexity, we use the expressiveness of the DL $\mathcal{ALC}$ [14] to perform service personalization, which is the standard DL $\mathcal{AL}$ (Attributive Language) extended with full existential qualification $\mathcal{E}$ and concept union $\mathcal{U}$.

$$
\begin{array}{ll}
\multicolumn{2}{l}{British \equiv Person \sqcap \exists hasSpokenLanguage.English} \\
\multicolumn{2}{l}{Name \equiv \exists hasFN.FirstName \sqcap \exists hasLN.LastName} \\
\multicolumn{2}{l}{BusinessAccount \equiv Account \sqcap \exists hasID.OpenID \sqcap} \\
\multicolumn{2}{l}{\qquad\qquad\qquad\qquad \exists hasSocialNetwork.SocialNetworkAccount} \\
\multicolumn{2}{l}{PersonalAccount \equiv Account \sqcap \exists hasID.ElectronicID \sqcap} \\
\multicolumn{2}{l}{\qquad\qquad\qquad\qquad \exists hasSocialNetwork.LinkedIn} \\
SkypeAccount \sqsubseteq Account & APIKey \sqsubseteq \top,\ BankID \sqsubseteq \top,\ Person \sqsubseteq \top \\
LinkedIn \sqsubseteq SocialNetworkAccount & Latitude \sqsubseteq \top,\ Longitude \sqsubseteq \top,\ ID \sqsubseteq \top \\
SocialNetworkAccount \sqsubseteq Account & BankAccount \sqsubseteq Account,\ English \sqsubseteq \top \\
OpenID \sqsubseteq ElectronicID \sqsubseteq Account & GMail \sqsubseteq MaillingAddress \sqsubseteq OpenID \\
\multicolumn{2}{l}{FirstName \sqsubseteq \top,\ LastName \sqsubseteq \top,\ British \sqsubseteq \top}
\end{array}
$$

Fig. 1. Sample of an $\mathcal{ALC}$ Terminological Box $\mathcal{T}$.

Besides standard DL reasoning approaches such as satisfiability or subsumption to guarantee consistency of DL knowledge bases or build concepts hierarchy, authors of [17] suggest computing the *Abduction* (Definition 1) between concepts C and D, representing what is underspecified in D in order to completely satisfy C taking into account the information modelled in a $\mathcal{ALN}$ (so compliant with $\mathcal{ALC}$) TBox $\mathcal{T}$.

Definition 1 ***(Concept Abduction Problem)***
Let $\mathcal{L}$ be a DL, C, D be two concepts in $\mathcal{L}$, $\mathcal{T}$ be a set of axioms in $\mathcal{L}$ and $\mathcal{A}$ be a set of assertions. A Concept Abduction Problem (CAP), denoted as $\langle \mathcal{L}, D, C, \mathcal{T} \rangle$ (or shortly $C \backslash D$) consists in finding a concept $B \in \mathcal{L}$ such that $\mathcal{T} \models D \sqcap B \sqsubseteq C$.

Example 1 ***(Concept Abduction)***
Let C and D be two $\mathcal{ALC}$ descriptions respectively defined by $BusinessAccount \sqcap$

$\exists hasSkype.SkypeAccount$ and $PersonalAccount$ (Fig.1). According to Definition 1, the description B required by D to satisfy (or more precisely to be subsumed by*) C is denoted by $C\backslash D$ i.e.,*

$$C\backslash D \doteq \exists hasID.OpenID \sqcap \exists hasSkype.SkypeAccount \quad (1)$$

In other words, D needs an OpenID and a SkypeAccount to satisfy C.

2.2 Semantic Web Services

Semantics of web services can be expressed by means of different descriptions, from their process levels [18] (i.e., internal and complex behaviours) and causal levels [19] (i.e., preconditions and effects on the world) to their functional levels (i.e., simple interface). In this work we will focus on the latter and more generally their functional input and output parameters, which are prominent to personalize and execute any service. In the semantic web, these functional parameters are enhanced with DL concepts that determine the semantics of the operations they achieve. Therefore, semantic web services can be expressed as DL concepts in (2).

$$Service \doteq \exists requires.Input \sqcap \exists returns.Output \quad (2)$$

This definition confines a semantic web service to being anything that $requires$ input parameters $Input$ to be processed and $returns$ some output parameters $Output$. Both latter parameters are defined in $\mathcal{T}$ such that $\mathcal{T} \models Input \sqsubseteq \top$ and $\mathcal{T} \models Output \sqsubseteq \top$. According to this model, the OWL-S profile [20], WSMO capability [21] or SA-WSDL [22] can be used to describe the functional level of semantic web services.

Example 2 ***(Semantic Web Service)***
Suppose a semantic web service S_1 locating friends and professional colleagues of a specific person, as its main functionality. This service, starting from a `FirstName`, `LastName`, `BusinessAccount`, `SkypeAccount` and a `GMail` address of this person, returns the list of her nearby `ContactPersons` according to her `Location`. According to (2)*, the semantic description of S_1 is defined by* (3)*.*

$$S_1 \doteq \exists requires.C^1_{S_1} \sqcap \exists requires.C^2_{S_1} \sqcap \exists requires.C^3_{S_1} \sqcap \exists requires.C^4_{S_1} \sqcap \exists returns.ContactPerson \quad (3)$$

where conjuncts $C^{i,1\leq i\leq 4}_{S_1}$, described by means of TBox $\mathcal{T}$ in Fig.1, are defined by:

$$C^1_{S_1} \doteq \exists hasFN.FirstName \sqcap \exists hasLN.LastName \quad (4)$$
$$C^2_{S_1} \doteq BusinessAccount \sqcap \exists hasSkype.SkypeAccount \quad (5)$$
$$C^3_{S_1} \doteq MaillingAddress \sqcap \exists hasMail.GMail \quad (6)$$
$$C^4_{S_1} \doteq Location \sqcap \exists hasLat.Latitude \sqcap \exists hasLong.Longitude \quad (7)$$

2.3 Semantic User Profile

Semantics of user profile can be expressed on different dimensions, mainly on information related to her i) identity and possessions, but could be also extended with her ii) long-term interests in topics [23] and preferences, iii) skills [24], iv) behaviour [25] and v) knowledge or beliefs in certain domains. Since the description of semantic user profiles are tailored, in this work, to access services in a personalized way trough their instantiation, we will focus solely on the former descriptions. Indeed, descriptions from (ii) to (v) such as the user's interests and preferences are mainly relevant information for prioritizing services in a discovery process rather than making services and their (input) parameters adapted to the user. Therefore, a DL expression of semantic user profiles is defined by the conjunction of different parts as expressed in (8).

$$Profile \doteq \bigsqcap_i \exists hasInfo.Info_i \bigsqcap_j \forall hasInfo.(\neg NonInfo_j) \tag{8}$$

This definition is based on a single role or property called $hasInfo$, which describes the user. While descriptions $Info_i$ cover a collection of data identifying users, descriptions expressed by $NonInfo_j$ refer to information users are not inclined to provide. The latter could be sensitive data such as medical record data. All initial information in (8) is collected from a short questionnaire [26] and could be also updated. Alternatively, the profile can be automatically generated from [27].

Example 3 ***(Semantic User Profile)***
Suppose a `British` *lady identified by her* `Name`*,* `ElectronicID` *and connected to the* `LinkedIn` *social network (http://www.linkedin.com/), but without any authorized access to her* `Bank Account` *from third parties. Its semantic profile P_1 is defined by:*

$$P_1 \doteq Female \sqcap \exists hasInfo.C^1_{P_1} \sqcap \exists hasInfo.C^2_{P_1} \sqcap \forall hasInfo.C^3_{P_1} \tag{9}$$

where conjuncts $C^{i,1\leq i\leq 3}_{P_1}$, described by means of TBox $\mathcal{T}$ in Fig.1, are defined by:

$$C^1_{P_1} \doteq Person \sqcap (\exists hasName.Name) \sqcap (\exists hasNationality.British) \tag{10}$$

$$C^2_{P_1} \doteq Account \sqcap (\exists hasID.ElectronicID) \sqcap (\exists hasSocialNetwork.LinkedIn) \tag{11}$$

$$C^3_{P_1} \doteq \neg BankAccount \sqcup \neg(\exists hasBank.BankID) \tag{12}$$

The model (8) we suggest for semantic user profile can be adapted with straightforward modifications and extensions (e.g., roles) depending on the application [24].

3 Semantic Web Service Personalization

The formalization of services (2) and user profiles (8) in DLs is required to compare their descriptions at a semantic level. The comparison is proceeded through a process of DL expressions matching, based on [15, 16] and [17]. This section, illustrated with examples, describes the personalization approach in details. First of all, matching through standard DL reasoning is employed to adapt services by means of user profiles. Then, we suggest an approach to compute relevant information which is missing in the user profile to improve personalization. Finally, we present two ways to extend the user profile with the latter information.

3.1 Personalized Adaptation of Services with Semantic User Profiles

Our approach, based on semantic matching, aims at discovering relevant information in the semantic user profile that could fit (i.e., be used by) the service (i.e., its functional input parameters) to be executed. By considering such a personalization, we suggest an approach which adapts any service to the user. To support this adaptation, the matching is performed over a service S and a user profile P with respect to a TBox $\mathcal{T}$. Therefore, a role hierarchy (13) between $requires$ and $hasInfo$ roles of S (2) and P (8) is required in $\mathcal{T}$, at least for satisfiable compatibility reasons [15] between S and P.

$$\mathcal{T} \models hasInfo \sqsubseteq requires \tag{13}$$

According to (13), some information ($hasInfo$) from the user profile could be used by services to adapt and then personalized its input parameters ($requires$).

In detail, our approach achieves such an personalized adaptation by following steps in Algorithm 1. From a logical point of view, this algorithm evaluates potential matching between conjuncts C_S and C_P of respectively S and P. By emphasizing only robust matching [28] i.e., Exact in line 7 and PlugIn in line 10, this algorithm focuses on the description required by S and provided by P. Therefore, other matching such as $\mathcal{T} \models C_P \sqsupseteq C_S$ (Subsume) or $\mathcal{T} \not\models C_P \sqcap C_S \sqsubseteq \bot$ (Intersection) are not valued since they do not fit our service adaptation purpose.

Algorithm 1: Matching-based Personalized Adaptation: $adapt(S, P, \mathcal{T})$.

1 **Input**: A user **S**ervice S, a **P**rofile P, a **T**erminological Box $\mathcal{T}$.
2 **Result**: $match$: a set of matching triples $(C_S, C_P, type)$ if C_S could be adapted by C_P with a matching $type$, $Incompatibility$ otherwise.
3 **begin**
4 $\quad match \leftarrow \emptyset$;
5 $\quad$ **foreach** $\exists requires.C_S \in Input(S)$ **do**
6 $\quad\quad$ *// Exact: The descriptions of user profile and service match perfectly.*
7 $\quad\quad$ **if** *there exists* $\exists hasInfo.C_P \in Info(P)$ *such that* $\mathcal{T} \models C_P \equiv C_S$ **then**
8 $\quad\quad\quad match \leftarrow match \cup_{set} (C_S, C_P, ''\equiv'')$;
9 $\quad\quad$ *// PlugIn: The user profile description is more specific than service description.*
10 $\quad\quad$ **if** *there exists* $\exists hasInfo.C_P \in Info(P)$ *such that* $\mathcal{T} \models C_P \sqsubseteq C_S$ **then**
11 $\quad\quad\quad match \leftarrow match \cup_{set} (C_S, C_P, ''\sqsubseteq'')$;
12 $\quad\quad$ *// Incompatible: The User profile and service descriptions are incompatible.*
13 $\quad\quad$ **if** *there exists* $\forall hasInfo.(\neg C_P) \in NonInfo(P)$ *such that* $\mathcal{T} \models C_P \sqsubseteq C_S$ **then**
14 $\quad\quad\quad$ **return** *Incompatibility*;
15 $\quad$ **return** $match$;
16 **end**

In Algorithm 1 Exact matching are tested before the PlugIn matching, mainly because of the logical implication relation between these matching. Indeed, if $\mathcal{T} \models C_P \equiv C_S$ (Exact), then $\mathcal{T} \models C_P \sqsubseteq C_S$ (PlugIn). The algorithm is then based on structural algorithms for satisfiability and subsumption [29]. Since it is reasonable to assume that users and service providers do not enter contradicting information, we assume that the services S and users profiles P descriptions are consistent in $\mathcal{T}$.

The result of this personalized adaptation step is a set of matching triples $(C_S, C_P, type)$ (referring to $match$) wherein the description C_P in P could be used to adapt the service parameter C_S in S. The latter descriptions are completed with their matching type, emphasizing the accuracy of the personalized adaptation step (from C_P to C_S). In case S violates some descriptions in P, S and P are returned as incompatible.

Example 4 ***(Semantic and Personalized Adaptation of Services)***
Suppose the service S_1 and user profile P_1 in Examples 2 and 3. According to Algorithm 1, the triple $(C^1_{S_1}, C^1_{P_1}, PlugIn)$ is returned. Indeed, according to the TBox $\mathcal{T}$ in Fig. 1, it seems possible to personalized S_1 by adapting the `FirstName` *and* `LastName` *input parameters of S_1 with information of $C^1_{P_1}$ in the user profile, and more specially with its* `Name`*.*

Even if our approach is able to personalize services with user profile descriptions using subsumption-based DL reasoning, the latter profile maybe not always as accurate as it should be hence limiting its benefits. Indeed, in some cases parts of services could only partially match or even mismatch the profile.

Example 5 ***(Limitation of Semantic and Personalized Adaptation of Services)***
Even if $C^2_{S_1}$ and $C^2_{P_1}$ in Examples 2 and 3 are both related to `Account` *description in $\mathcal{T}$, $C^2_{P_1}$ cannot be used to personalize S_1 and more specially $C^2_{S_1}$ because of missing description* (1) *in its profile P_1 i.e., neither* `BusinessAccount`*, nor* `SkypeAccount`*.*

3.2 Towards Incomplete Semantic User Profile

Since the personalization process may fail because of under specification or missing description in the user profile (e.g., Example 5), we suggest to extend Algorithm 1 with Algorithm 2 by exploiting results from non robust matching cases between conjuncts C_S and C_P i.e., Intersection and Subsume. Therefore, we aim at i) inferring further matching triples from the latter matching cases, and ii) discovering descriptions which are required by services but not provided by the user profile by applying abduction.

Further Matching Triples for Personalized Adaptation of Services: In both Intersection and Subsume matching cases, simple matching triple $(C_S, C_P, type)$ cannot be returned as in Algorithm 1 mainly because *only a part* of C_P is required to adapt *(again) a part* of C_S. Towards this issue, we identify descriptions B and A that need to be removed respectively from C_S and C_P to obtain a PlugIn matching between C_P and C_S i.e., $C_P \Delta C_S$ (Definition 2, adapted from [30]).

Definition 2 ***(Symmetric Difference)***
Let $\mathcal{L}$ be a DL, C, D be two concepts in $\mathcal{L}$, $\mathcal{T}$ be a set of axioms in $\mathcal{L}$ and $\mathcal{A}$ be a set of assertions. The symmetric difference between C, D, denoted as $C\Delta D$ consists in finding two concepts $A, B \in \mathcal{L}$ such that

$$\mathcal{T} \models C \backslash A \sqsubseteq D \backslash B \tag{14}$$

$C^*\Delta D$ and $C^*\Delta D$ refer respectively to A and B in (14)*.*

Example 6 ***(Symmetric Difference)***
Let $C^2_{P_1}$ and $C^2_{S_1}$ be two $\mathcal{ALC}$ descriptions respectively defined in Examples 2, 3 with respect to $\mathcal{T}$ in Figure 1. According to Definition 2, $C\Delta D$ is defined by A and B i.e.,

$$A \doteq \exists hasID.ElectronicID \tag{15}$$

$$B \doteq \exists hasID.OpenID \ \sqcap \ \exists hasSkype.SkypeAccount \tag{16}$$

Algorithm 2: (Refined) Personalized Adaptation: $refinedAdapt(S, P, \mathcal{T})$.

1 **Input**: A **S**ervice S, a user **P**rofile P, a **T**erminological Box $\mathcal{T}$.
2 **Result**: A pair $(match, miss)$ where $match$ is set of matching Triples $(C_S, C_P, type)$ if C_S could be adapted by C_P with a matching $type$, and $miss$ refers to the set of missing description in P in order to adapt S.
3 **begin**
4 $\quad i \leftarrow 0;\ miss \leftarrow \emptyset;\ match \leftarrow \emptyset;$
5 $\quad$ **foreach** $\exists requires.C_S \in Input(S)$ **do**
6 $\quad\quad$ *// Non Robust: Profile description partially covers service description.*
7 $\quad\quad$ **if** *there exists* $\exists hasInfo.C_P \in Info(P)$ *such that* $\mathcal{T} \not\models C_P \sqcap C_S \sqsubseteq \bot$ *and* $\mathcal{T} \not\models C_P \sqsubseteq C_S$ **then**
8 $\quad\quad\quad A \leftarrow C^*_P \Delta C_S$; *// Descriptions A in C_P and B in C_S...*
9 $\quad\quad\quad B \leftarrow C_P \Delta C^*_S$; *// ...that make $\mathcal{T} \not\models C_P \sqsubset C_S$.*
10 $\quad\quad\quad X \leftarrow C_S \backslash B$; *// Descriptions X in C_S and Y in C_P...*
11 $\quad\quad\quad Y \leftarrow C_P \backslash A$; *// ...such that $\mathcal{T} \models Y \sqsubseteq X$.*
12 $\quad\quad\quad match \leftarrow match \cup_{set} (X, Y, ''\sqcap'');$
13 $\quad\quad\quad miss_i \leftarrow C_S \backslash C_P;$
14 $\quad\quad\quad i \leftarrow i + 1;$
15 $\quad miss \leftarrow \inf_{\sqsubseteq}\{miss_j | 0 \leq j \leq i\} \backslash_{set} \{\top\};$
16 $\quad$ **return** $(match, miss)$;
17 **end**

According to Definitions 1 and 2, it is straightforward to identify the parts $X \in \mathcal{P}(C_S)$ and $Y \in \mathcal{P}(C_P)$ respectively from C_S and C_P (where $\mathcal{P}(S)$ refers to power set of S) which are required to ensure that X can be adapted by Y (in the sense of Algorithm 1 i.e., $\mathcal{T} \models C_P \sqsubseteq C_S$). Both descriptions X and Y, parts of the new matching triples, are defined as $C_S \backslash (C_P \Delta C^*_S)$ and $C_P \backslash (C^*_P \Delta C_S)$. Algorithm 2 elaborates these further matching triples (line 12) from respectively lines 9, 10 and lines 8, 11. The first element of the matching triples represents the description in C_S which subsumes C_P, whereas the second element represents the part in C_P which is subsumed by C_S. In other words, these triples present the descriptions in C_P which will be used to further adapt the descriptions in C_S. Finally, as Algorithm 1, results are aggregated in $match$.

Example 7 ***(Matching Profile to Service with an Intersection Match)***
According to Example 2, 3 and $\mathcal{T}$ in Figure 1, $\mathcal{T} \not\models C^2_{P_1} \sqcap C^2_{S_1} \sqsubseteq \bot$. Therefore, only some parts Y of $C^2_{P_1}$ can be used to adapt some parts X of $C^2_{S_1}$. Applying Algorithm 2 (lines from 8 to 12) and using results from Example 6, Y and X are defined as:

$$Y \equiv Account \sqcap (\exists hasSocialNetwork.LinkedIn) \tag{17}$$

$$X \equiv Account \sqcap (\exists hasSocialNetwork.SocialNetworkAccount) \tag{18}$$

This simply means that the `SocialNetworkAccount` *of* $C^2_{P_1}$ *(i.e.,* `LinkedIn`*) can be used to instantiate the* `SocialNetworkAccount` *requirement (i.e., input parameter) of* $C^2_{S_1}$.

Computing Missing Description in a User Profile: In addition, Algorithm 2 (lines 13 and 15) computes descriptions which are required by C_S and not (or partially) provided by C_P. To this end, abduction is applied between the latter conjuncts (Definition 1) and then the result is aggregated in a set of missing description $miss$ (Definition 3).

Definition 3 ***(Set of Missing Description)***
The set of missing description $miss$ *of a service personalization problem* $(S, P, \mathcal{T})$ *is defined by:*

$$miss(S, P, \mathcal{T}) \doteq \inf_{\sqsubseteq} \{C_S \backslash C_P | \mathcal{T} \not\models C_P \sqcap C_S \sqsubseteq \bot\} \backslash_{set} \{\top\} \tag{19}$$

where C_S *and* C_P *are respectively conjuncts of* $\exists hasInfo.C_P$ *and* $\exists requires.C_S$.

According to Definition 3, $miss$ gathers the most specific descriptions of the set $\{C^i_S \backslash C^j_P\}$. Therefore a same description (i.e., the most specific in the service requirements) can be used to satisfy different abduction problems $C^i_S \backslash C^j_P$ and then could be exposed as a description not provided by P but required by some conjuncts (related by subsumption) of S. $miss$ does not only explain why services have not been adapted and personalized (regarding a user profile) but also suggest a solution to extend the personalization of semantic web services.

Property 1 ***(Empty Set of Missing Description)***
The set of missing description $miss$ *of a web service personalization problem* $(S, P, \mathcal{T})$ *with either i)* $\mathcal{T} \models C_P \equiv C_S$*; or ii)* $\mathcal{T} \models C_P \sqsubseteq C_S$ *is the empty set.*

Proof. By Definition 1, $C_S \backslash C_P$ is defined by $\mathcal{T} \models C_P \sqcap (C_S \backslash C_P) \sqsubseteq C_S$. Therefore, we obtain in both cases that $C_S \backslash C_P \equiv \top$ is a solution i.e., $miss$ is defined by the empty set according to Definition 3.

The property 1 justifies our choice of not computing missing descriptions in Algorithm 1. Indeed, such a computation would reach to the empty set.

Example 8 ***(Set of Missing Description)***
Since the description in P_1 *is not enough to totally adapt and personalize* S_1 *(Example 5), Algorithm 2 and Definition 3 are required to discover the missing description* $miss$ *in* P_1*. According to the latter definition,* $miss$ *is constituted by the union of results of the abduction problems* $C^2_{S_1} \backslash C^2_{P_1}$ *(Example 1),* $C^4_{S_1} \backslash \top$ *and* $C^3_{S_1} \backslash C^2_{P_1}$*:*

$$C^2_{S_1} \backslash C^2_{P_1} \equiv Account \sqcap \exists hasID.OpenID \sqcap \exists hasSkype.SkypeAccount \tag{20}$$

$$C^4_{S_1} \backslash \top \equiv Location \sqcap \exists hasLat.Latitude \sqcap \exists hasLong.Longitude \tag{21}$$

$$C^3_{S_1} \backslash C^2_{P_1} \equiv MaillingAddress \sqcap \exists hasMail.GMail \tag{22}$$

Since $GMail \sqsubseteq OpenID$, $hasMail \sqsubseteq hasID$ and $miss$ only considers most specific description (whether subsumption-based comparable), $miss$ is $\{A, C^4_{S_1}\}$ where:

$$A \equiv Account \sqcap \exists hasMail.GMail \sqcap \exists hasSkype.SkypeAccount \tag{23}$$

According to Property 1, conjuncts $C^{i,1\leq i\leq 4}_{S_1}$ and $C^{j,1\leq j\leq 2}_{P_1}$ such that $\mathcal{T} \models C^j_{P_1} \sqsubseteq C^i_{S_1}$ are not considered by Algorithm 2.

3.3 Extending Semantic User Profile with Further (Missing) Descriptions

Once the set of missing descriptions is retrieved through $miss$, two approaches are considered. First of all, an intuitive method consists in discovering [31] which new and appropriate services $S_{i,1\leq i\leq n}$ would be able to return the missing description. Following this approach, this description $miss$ could be identified and satisfied by the conjunction of some output parameters $Out\_S_{i,1\leq i\leq n}$ of these services $S_{i,1\leq i\leq n}$. Therefore, depending on the available description in the user profile P and the description of output parameters of $S_{i,1\leq i\leq n}$, we could proceed to the service S personalization. Indeed, the conjunction C of the output parameters $Out\_S_{i,1\leq i\leq n}$ and the user profile P i.e., $\bigsqcap_{i=1}^{n} Out\_S_i \sqcap P$ can be used to adapt and personalize the service S by applying Algorithm 1 as following: $adapt(S, C, \mathcal{T})$. However, each input parameter of these discovered services S_i has to be known at run time. To this end, we can imagine use the description available in P to adapt this new discovered services. In this direction, Algorithm 1 is applied on the n relevant services as following: $adapt_{i,1\leq i\leq n}(S_i, P, \mathcal{T})$. One constraint of this method is related to the number of services (and their input parameters to be satisfied by P) which are required to satisfy $miss$.

In the second approach, the set of missing description is simply suggested to the user. The user is then responsible of providing the description that the system needed to adapt and personalize the service. The requested information is also used to populate the semantic user profile, hence available for further personalization purposes.

4 Validation

In this section, we discuss the prototype tool that we developed to provide personalized adaptation of semantic web services. Moreover we give a preliminary evaluation of the suggested approach by analyzing some results obtained with the prototype.

4.1 Architecture and Implementation

Figure 2 shows the high level prototype architecture wherein we implemented and tested our personalization approach. In detail, our approach, part of the core architecture of the EU project SOA4All[1] (Service Oriented Architectures for All), has been integrated with three main state-of-the-art modules, namely a *DL Reasoning*, a *Service Discovery* and a *SPARQL Query Engine* module. The main function of the former module is to check satisfiability, subsumption and infer on-line matching between user profile and service

[1] http://www.soa4all.eu/

description. The MAMAS-tng[2] reasoner has been used to compute standard reasoning and evaluate abduction. This reasoner has been extended to compute symmetric difference (Definition 2). The *SPARQL*[3] *Query Engine* module RDF2GO[4], is required to manipulate matching triples i.e., RDF-based C_P, C_S and $miss$ data returned by Algorithms 1 and 2. For instance, this module transforms RDF-based C_P in C_S using a CONSTRUCT query form of SPARQL in order to adapt C_S with C_P.

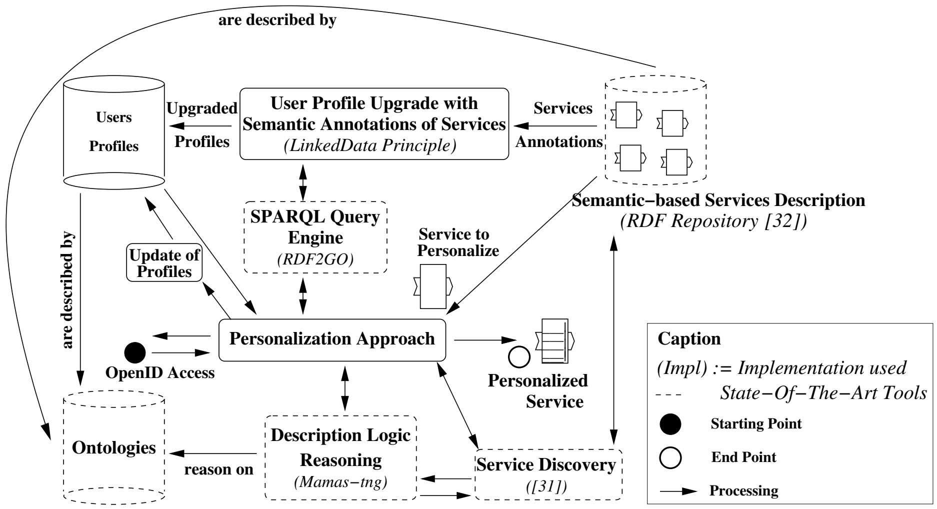

Fig. 2. Core Architecture for Service Personalization.

In addition, a pool of (SA-WSDL) *semantic-based services* (formalized in (2)), and *semantic user profile* (formalized in (8) and identified by OpenID[5] are stored in two different RDF[6] repositories, e.g., [32] for services and a Sesame-based[7] for profiles. Their descriptions are based on different $\mathcal{ALC}$ TBoxes, depending on *ontologies* used to annotate services. Therefore, the semantic descriptions used to annotate a service and a profile may differ, even if the annotated element is the same. Towards this issue of ontology matching [33], we integrated a simple component (*User Profile Upgrade* Component) which aims at semi-manually linking profiles and services descriptions, following the Linked Data principles [34] e.g., by further annotating profiles with relevant `owl:sameAs`, `rdfs:subClassOf`, `owl:equivalentProperty` or `rdfs:subPropertyOf` constructs. To this end, the *SPARQL Query Engine* module is used. Finally, once the service personalization process is achieved, the service is ready to be executed by any execution engine. Moreover, the semantic user profile is updated in case of missing description (Definition 3). To this end, either the user is able to provide it, or a service discovery process is performed (Section 3.3).

[2] http://dee227.poliba.it:8080/MAMAS-tng/DIG
[3] http://www.w3.org/TR/rdf-sparql-query/
[4] http://semanticweb.org/wiki/RDF2Go
[5] http://openid.net/
[6] http://www.w3.org/RDF/
[7] http://coconut.tie.nl:8080/storage/repositories/profiles

4.2 Experimental Results

Evaluation of personalization systems remains a challenge due to the lack of understanding of what factors affect user satisfaction with a personalization system [35]. Personalization systems are in general evaluated and compared on the accuracy of predictions. However, a comparison based on the accuracy of our approach and existing personalization methods is not appropriate. Indeed, our work i) features different expressivity (compared to syntactic-based approaches), and ii) does not only evaluate if a user profile and a service match but also explains how they could match and why they could not (compared to semantic-based approaches).

Therefore we analyze the performances of our approach by i) comparing abduction- with difference-based [36, 37] personalization using different expressivities of DL, and ii) studying the impact of the *User Profile Upgrade* component (Fig.2) on the personalization process. The experiments have been conducted on Intel(R) Core(TM)2 CPU, 2.4GHz and 2GB RAM.

Comparing Abduction- and Difference-based Service Personalization: Since other approaches based on DL difference operator such as [36] or [37] can be used to compute from a given description all the information different in another description, we suggest to compare them with abduction to achieve service personalization (Fig.3). The comparison is driven on three set of ontologies with different DLs used to annotate services and profiles i.e., $\mathcal{ALC}$, $\mathcal{ALN}$ and $\mathcal{ALE}$, from the most to the least expressive. In particular, the two former ontologies are based on the $\mathcal{ALE}$ TBox (formally defined by 1100 concepts and 390 properties) wherein only DL operators changed in descriptions. Personalization of up to 100 services have been considered in this experiment, especially for obtaining convincing results towards their applicability in real (industrial) scenarios.

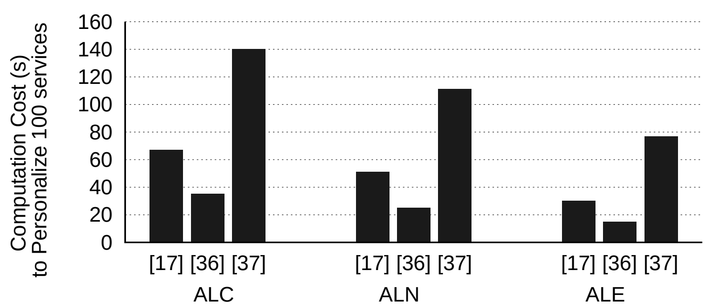

Fig. 3. Abduction vs. Difference.

The [37]'s difference consider a semantic maximum (ordering according to the subsumption operator) between (only) subsumption-based comparable descriptions. Even if they provides sufficient condition (i.e., structural subsumption relation) to characterize the uniqueness of difference, some TBoxes cannot be considered such as $\mathcal{ALN}$. In addition, this is the most time consuming approach regarding the three set of experimentation, mainly because they perform an equivalence between two concept descriptions ($\mathcal{T} \models D \sqcap B \equiv C$) whereas abduction computes (only) a subsumption of concept descriptions ($\mathcal{T} \models D \sqcap B \sqsubseteq C$). The difference operator of [36] is a refinement of

[37]'s difference that considers the syntactic minimum ($\preceq_d$) between incomparable descriptions. Such a consideration, limiting the relevance of its results using expressive DLs, explains its very good performance.

Even if deciding subsumption, computing abduction and difference in $\mathcal{ALE}$ is NP-complete, Figure 3 reports the feasibility and the scalability of the personalization process. However these results depend on size and structure of the used ontologies, size and complexity of user profile and service descriptions. The choice of abduction to personalize services is justified by its performance in the three different DLs studied. Indeed, our process of personalizing services with an expressive DL $\mathcal{ALC}$ over performs the time consuming [37]'s difference-based personalization using $\mathcal{ALN}$ or $\mathcal{ALE}$ DLs.

Impact of the User Profile Upgrade Component: Since our personalization approach aims at matching services to user profiles, it is required that their descriptions can be semantically compared, either using a same ontology, or by establishing subsumption-based relationships between some of them. This experiment studies the qualitative impact of the latter (i.e., *User Profile Upgrade* component in Fig.2) on personalization.

To this end, 55 different initial $\mathcal{ALN}$ TBoxes (i.e., average of 103 concepts and 61 properties) have been used to annotate 100 services $S_{i,1\leq i\leq 100}$ and one profile P. Then, progressively, some inter-connections have been established between them, favouring the matching hence the personalization. Roughly speaking, the progression of connections grows exponentially along 10 rounds $R_{i,1\leq i\leq 10}$ i.e., from adding 2^1 to 2^{10} connections. After each round, we run our personalization approach on these 100 services and evaluate the rate of i) input parameters q_S that can be adapted given P and ii) missing description q_{miss}, both regarding the number of description in S_i.

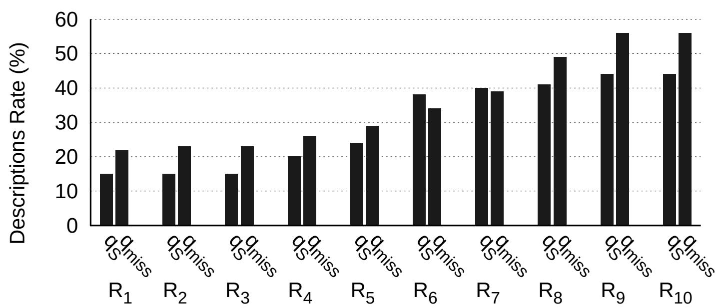

Fig. 4. Impact of the User Profile Upgrade Component on Service Personalization.

As shown in Fig.4, the more interconnections between ontologies, the better personalization (i.e., q_S). In the same way, the rate of description to be updated in the user profile (i.e., q_{miss}) is also improving. Both improvements follow a linear evolution even if an exponential generation of interconnections is applied. In more detail, 37% and 98% of service descriptions can be semantically compared (not matched) to profile description respectively in the first and last round of interconnections generation. The transition between round 5 and 6 (with 64 new connections) is the most significant with respectively 53% and 72% of comparable descriptions in services and profile. While our approach is able to adapt and personalize 44% of services in the last two rounds, the personalization only reaches 15% of services in the first three rounds. This confirms the high impact of the user profile upgrade component in our personalization framework.

q_{miss} is in general higher than q_r since some input parameters of services cannot be captured by personalization e.g., variables such as flight destination, or sensitive data.

5 Related Work

Based on [38], [10] present *Personal Reader Framework*, an approach for RDF-based data extraction, combination, visualization and personalization. In particular, they generate personalized view of data by applying standard subsumption-based matching between data description, user profile and contextual information. To this end they adopt TRIPLE [39], a rule language which is designed for querying and transforming RDF models. Even if their approach is augmented with contextual information, their matching-based personalization approach do not consider automated update of user profile.

Contrary to our approach, [8, 23, 27] perform personalization to obtain more relevant services during the discovery process. Therefore they do not address services instantiation (through their personalized adaptation) but rather service selection. The selection process is based on the compatibility of users' interests, disinterests and service descriptions. Services that do not match a certain profile are discarded on the fly. Since the matching process between the latter descriptions is handled on mobile device, both approaches [23] sacrifice expressivity of DL and use standard DL inferences [15, 16]. Even more (semantically) limited, [27] consider only one-to-one syntactic matching of service and profile descriptions for personalization. [8] consider non-functional parameters, preferences and knowledge that is implicitly given by previous service to personalize the selection of services.

[24] present an approach for matching user profiles for applications such as job recruitment or dating system. The matching, which is performed on a demand profile P_d and a supply profile P_S, aims at evaluating their semantic similarity. In the same way as our work, abduction is used, but only for weighting, ranking purposes and not for extracting and reusing relevant parts of P_d and P_S. In addition they apply the non-standard inference contraction [40] to evaluate the effort (i.e., description) required to make $P_d \sqcap P_s$ satisfiable in $\mathcal{H}$. On the contrary, we assume the latter conjunct to be satisfiable in $\mathcal{H}$ since one goal of our personalization approach is to suggest more specific descriptions (and so satisfiable) to the user profile regarding the service descriptions.

6 Conclusion

In this work we studied service personalization or the way to tailor services to a particular user. In particular, we addressed automated instantiation of services (through personalized adaptation) which is crucial for advanced usability i.e., how to prepare and present services ready to be executed while limiting useless interactions with users? Towards this issue, we considered a semantic augmentation of services and extensible user profiles to infer potential matching between both descriptions. The semantic matching, core of our approach, exploits standard DL reasoning and abduction to i) identify useful parts of a user profile that satisfy the service requirements (i.e., input parameters) and

ii) compute the descriptions required by a service to be consumed but not provided by the user profile. Our approach, integrated in the service consumption of the EC-funded project SOA4All, has been augmented with a process of user profile upgrade in case heterogeneous ontologies are used to describe services. Such an augmentation, aiming at linking data description of services and user profile, has been validated by experimental results. In the same way, the latter results confirm our choice of preferring abduction rather than other difference operators for scalability and expressivity reasons.

In future work we will consider a more precise abduction operator, which is also easy-to-compute in expressive DLs in order to address more complex cases of personalization and user profile update. We will also focus on the context dimension for personalization. Another area of investigation is the policy-based control access [41] of the user profile by third parties during its update. Finally, as reported by experimental results, automating ontologies alignments is a key issue that need to be address.

References

1. Mobasher, B., Cooley, R., Srivastava, J.: Automatic personalization based on web usage mining. Commun. ACM **43**(8) (2000) 142–151
2. Goldberg, D., Nichols, D.A., Oki, B.M., Terry, D.B.: Using collaborative filtering to weave an information tapestry. Commun. ACM **35**(12) (1992) 61–70
3. Lieberman, H.: Letizia: An agent that assists web browsing. In: IJCAI (1). (1995) 924–929
4. Pazzani, M.J.: A framework for collaborative, content-based and demographic filtering. Artif. Intell. Rev. **13**(5-6) (1999) 393–408
5. Alonso, G., Casati, F., Kuno, H., Machiraju, V.: Web Services: Concepts, Architectures and Applictions. Springer-Verlag (2004)
6. Sivashanmugam, K., Verma, K., Sheth, A.P., Miller, J.A.: Adding semantics to web services standards. In: ICWS. (2003) 395–401
7. García, J.M., Ruiz, D., Cortés, A.R.: A model of user preferences for semantic services discovery and ranking. In: ESWC (2). (2010) 1–14
8. Balke, W.T., Wagner, M.: Towards personalized selection of web services. In: WWW (Alternate Paper Tracks). (2003)
9. Mandell, D.J., McIlraith, S.A.: Adapting bpel4ws for the semantic web: The bottom-up approach to web service interoperation. In: ISWC. (2003) 227–241
10. Baumgartner, R., Henze, N., Herzog, M.: The personal publication reader: Illustrating web data extraction, personalization and reasoning for the semantic web. In: ESWC. (2005) 515–530
11. Blake, M.B., Nowlan, M.F.: A web service recommender system using enhanced syntactical matching. In: ICWS. (2007) 575–582
12. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American **284**(5) (2001) 34–43
13. Smith, M.K., Welty, C., McGuinness, D.L.: Owl web ontology language guide. W3c recommendation, W3C (2004)
14. Baader, F., Nutt, W. In: The Description Logic Handbook: Theory, Implementation, and Applications. (2003)
15. Li, L., Horrocks, I.: A software framework for matchmaking based on semantic web technology. In: WWW. (2003) 331–339
16. Paolucci, M., Kawamura, T., Payne, T., Sycara, K.: Semantic matching of web services capabilities. In: ISWC. (2002) 333–347

17. Noia, T.D., Sciascio, E.D., Donini, F.M., Mongiello, M.: Abductive matchmaking using description logics. In: IJCAI. (2003) 337–342
18. Pistore, M., Marconi, A., Bertoli, P., Traverso, P.: Automated composition of web services by planning at the knowledge level. In: IJCAI. (2005) 1252–1259
19. McIlraith, S.A., Son, T.C.: Adapting golog for composition of semantic web services. In: KR. (2002) 482–496
20. Ankolenkar, A., Paolucci, M., Srinivasan, N., Sycara, K.: The owl-s coalition, owl-s 1.1. Technical report (2004)
21. Fensel, D., Kifer, M., de Bruijn, J., Domingue, J.: Web service modeling ontology submission, w3c submission. (2005)
22. Kopecký, J., Vitvar, T., Bournez, C., Farrell, J.: Sawsdl: Semantic annotations for wsdl and xml schema. IEEE Internet Computing **11**(6) (2007) 60–67
23. Kleemann, T., Sinner, A.: User profiles and matchmaking on mobile phones. In: INAP. (2005) 135–147
24. Calì, A., Calvanese, D., Colucci, S., Noia, T.D., Donini, F.M.: A description logic based approach for matching user profiles. In: Description Logics. (2004)
25. Berendt, B., Spiliopoulou, M.: Analysis of navigation behaviour in web sites integrating multiple information systems. VLDB J. **9**(1) (2000) 56–75
26. Ghosh, R., Dekhil, M.: Discovering user profiles. In: WWW. (2009) 1233–1234
27. Weiß, D., Scheuerer, J., Wenleder, M., Erk, A., Gülbahar, M., Linnhoff-Popien, C.: A user profile-based personalization system for digital multimedia content. In: DIMEA. (2008) 281–288
28. Lécué, F., Delteil, A.: Making the difference in semantic web service composition. In: AAAI. (2007) 1383–1388
29. Borgida, A., Patel-Schneider, P.F.: A semantics and complete algorithm for subsumption in the classic description logic. J. Artif. Intell. Res. (JAIR) **1** (1994) 277–308
30. Herzig, A.: The pma revisited. In: KR. (1996) 40–50
31. Benatallah, B., Hacid, M.S., Léger, A., Rey, C., Toumani, F.: On automating web services discovery. VLDB J. **14**(1) (2005) 84–96
32. Pedrinaci, C., Lambert, D., Maleshkova, M., Liu, D., Domingue, J., Krummenacher, R.: Service Engineering: European Research Results. In: Adaptive Service Binding with Lightweight Semantic Web Services. (2010)
33. Euzenat, J., Shvaiko, P.: Ontology matching. Springer-Verlag, Heidelberg (DE) (2007)
34. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. Int. J. Semantic Web Inf. Syst. **5**(3) (2009) 1–22
35. Anand, S.S., Mobasher, B.: Introduction to intelligent techniques for web personalization. ACM Trans. Internet Technol. **7**(4) (2007) 18
36. Brandt, S., Kusters, R., Turhan, A.: Approximation and difference in description logics. In: KR. (2002) 203–214
37. Teege, G.: Making the difference: A subtraction operation for description logics. In: KR. (1994) 540–550
38. Antoniou, G., Baldoni, M., Baroglio, C., Baumgartner, R., Bry, F., Eiter, T., Henze, N., Herzog, M., May, W., Patti, V., Schaffert, S., Schindlauer, R., Tompits, H.: Reasoning methods for personalization on the semantic web. Annals of Mathematics, Computing & Telefinformatics **2(1)** (2004) 1–24
39. Sintek, M., Decker, S.: Triple - a query, inference, and transformation language for the semantic web. In: International Semantic Web Conference. (2002) 364–378
40. Colucci, S., Noia, T.D., Sciascio, E.D., Donini, F.M., Mongiello, M.: Concept abduction and contraction in description logics. In: DL. (2003)
41. Abel, F., Henze, N., Krause, D.: Context-aware ranking algorithms in folksonomies. In: WEBIST. (2009) 167–174