A Proposal for Describing Services with DLs

Carsten Lutz and Ulrike Sattler
{lutz,sattler}@cs.rwth-aachen.de

Abstract

Motivated by the semantic web application, we present a generic extension of description logics to describe actions. These actions can then be chained to service descriptions. A web page providing a service can be annotated with a description of this service, which can then be taken into account by agents searching for a web service. Besides syntax and semantics of this extension of DLs, we define and discuss inference problems which are useful to annotate web pages with a description of the service they provide.

1 Motivation

DLs have proven to provide useful support for the definition, integration, and maintenance of ontologies [7, 17]—a feature which makes DLs important for the semantic web [3], where ontologies will play a central role. More precisely, ontologies are envisioned to be used in annotations describing the *content* of web documents, and will then be used by agents searching the semantic web.

However, these agents, besides searching for information or web pages with a certain content, should also be able to search for *services* offered in web pages such as ordering a book. Hence the ontologies referred to in the annotations should also allow for descriptions of services. Now services differ in principle from other concepts described in an ontology in that, when describing them, we need to describe their dynamic behaviour. That is, we want to say what a service S expects to hold prior to its invocation and how its invocation changes the world, e.g., how the world looks like after the service has been "carried out". This is, unsurprisingly, pretty similar to the actions described in planning and reasoning about actions (see, e.g., Chapter 4 of [16] for an overview on planning, [15] for an overview on logics for reasoning about actions, and [5] for DLs for planning and reasoning about actions): a service/action is described by pre-conditions, changes the action yields, and post-conditions.

In this paper, we propose a generic formalism which allows to describe the dynamic behaviour of services. This formalism allows to refer to concepts defined in an ontology described in an expressive DL well-suited for the semantic web. Besides defining (the syntax and semantics of) this formalism, we propose interesting inference problems, i.e., deciding whether a service is realizable, whether a service yields certain results, and whether one service subsumes another one. Algorithms deciding these problems can be used, for example, to support the annotation of web pages, to structure services, and to find services on the web.

2 Describing Services

We assume the reader to be familiar with DLs. The following framework is generic in that it can be instantiated with any DL $\mathcal{L}$, e.g., $\mathcal{ALC}$, $\mathcal{SHIQ}$, $\mathcal{DLR}$, $\mathcal{SHOQ}(\mathcal{D})$, or $\mathbb{Q}$-$\mathcal{SHIQ}$ [19, 10, 4, 9, 12]. We start with the definition of the syntax of actions, then explain the intuition of actions and give some examples, and finally continue with the definition of the semantics of actions.

Definition 1 Let N_C be a set of individual names, N_X a set of individual variables (variables), and $\mathcal{L}$ a (description) logic. We use N_I as an abbreviation for $N_X \cup N_C$. A *condition* is an expression of the form

$$\forall C, \quad C(a), \quad R(a,b), \quad a \neq b$$

for $a, b \in N_I$, C an $\mathcal{L}$-concept and R a possibly negated $\mathcal{L}$-role. A *relaxation* is an expression of the form

$$?C_p \quad ?C_p(a) \quad ?C_p, Q(a) \quad ?Q \quad ?Q(a) \quad \text{or } ?Q(a,b)$$

for $a, b \in N_I$, C_p a concept name, C, D $\mathcal{L}$-concepts, and Q a role name.

An *action* $A = (\mathsf{pre}, \mathsf{rel}, \mathsf{post})$ is a triple consisting of

- a set pre of conditions, the so-called pre-conditions,
- a set rel of relaxations, and
- a set post of pairs π/c of a set of conditions π and a condition c, the so-called post-conditions.

Next, we describe such an action A intuitively. Firstly, actions define a relation on interpretations, i.e., an action A relates an interpretation $\mathcal{I}$ to an interpretation $\mathcal{I}'$ if A can yield $\mathcal{I}'$ when "applied" to $\mathcal{I}$. This can only be the case if $\mathcal{I}$ and $\mathcal{I}'$ satisfy A's pre- and post-conditions:

- conditions that must be satisfied for A to be carried out are described in pre. A condition of the form $\forall C$ requires that each individual in $\mathcal{I}$ is an instance of C. The other conditions correspond to ABox assertions in DLs; see, e.g., [18].

- conditions that must be satisfied after A has been carried out (i.e., by $\mathcal{I}'$) are described in post. Since the state of the world *before* A may influence the effects of A, the post-conditions differ slightly from pre-conditions. The idea behind each π/c is that, if each condition in π holds in $\mathcal{I}$, then c must hold in $\mathcal{I}'$ (e.g., if a gun is loaded, then pulling its trigger yields a shot, whereas nothing happens when pulling the trigger of an unloaded gun).

Moreover, we want that $\mathcal{I}'$ differs only *minimally* from $\mathcal{I}$. That is, we only want that $\mathcal{I}'$ differs in those aspects from $\mathcal{I}$ that are required by the post-conditions. This will be ensured by the semantics of actions. However, one might not want to foresee *all* effects an action can have, and thus might want that the interpretation of certain role or concept names X may change freely from $\mathcal{I}$ to $\mathcal{I}'$. In this case, one can use X in a *relaxation* of A. Such a relaxation can be global as in $?C_p$, or local as in $?C_p(a)$. For example, when describing the service of buying a house, we might want to say that the happiness of the new house owner o_2 may change freely using $?\mathtt{Happy}(o_2)$.

This "built-in" minimization of changes together with the possibility to relax it for certain role or concept names is one of the central features of our proposal.

Before defining the semantics of actions, we will give some intuitive examples. The following is a simple bicycle-selling action:

$$\begin{aligned}&(\{\mathtt{owns}(a_1,b),\mathtt{wants}(a_2,b),\mathtt{owns}(a_2,p),\mathtt{Bicycle}(b)\},\\&\emptyset,\\&\{\emptyset/\mathtt{owns}(a_2,b),\emptyset/\mathtt{owns}(a_1,p)\}),\end{aligned} \tag{1}$$

where $\emptyset$ is the (empty) set. As condition, the empty set is satisfied by each interpretation. As a relaxation, the empty set ensures that the changes of all concept and role names are minimized.

Provided that (1) $\mathtt{owns}^-$ is a functional role (or the background knowledge base contains an axiom like $\top \sqsubseteq (\leq 1\,\mathtt{owns}^-)$), (2) we are happy to model the price of the bicycle b using the abstract object p, and (3) nothing else should change by selling/buying a bicycle, this simple action describes selling a bicycle in a sufficient way: due to the semantics, only the ownerships of the bicycle and its prize will change by this action. For example, b remains a bicycle. If the bicycle is a bad one, the new owner will be unhappy and the former owner happy, which can be modeled in the following way:

$$\begin{aligned}&(\{\mathtt{owns}(a_1,b),\mathtt{wants}(a_2,b),\mathtt{owns}(a_2,p),\mathtt{Bicycle}(b)\},\\&\emptyset,\\&\{\ \emptyset/\mathtt{owns}(a_2,b),\quad \{\mathtt{Bad}(b)\}/\neg\mathtt{Happy}(a_2),\\&\ \ \emptyset/\mathtt{owns}(a_1,p),\quad \{\mathtt{Bad}(b)\}/\mathtt{Happy}(a_1)\})\end{aligned} \tag{2}$$

If we want to allow models where buying a bicycle might make the former or the new owner (un)happy, we can modify the action in the following way:

$$\begin{aligned}&(\{\mathtt{owns}(a_1,b),\mathtt{wants}(a_2,b),\mathtt{owns}(a_2,p),\mathtt{Bicycle}(b)\},\\&\{?\mathtt{Happy}(a_1),?\mathtt{Happy}(a_2)\},\\&\{\emptyset/\mathtt{owns}(a_2,b),\emptyset/\mathtt{owns}(a_1,p)\})\end{aligned} \tag{3}$$

Finally, if we want to express that the happiness of a_2's parents can change freely (i.e., they can remain (un)happy or become (un)happy) through a_2 buying the bicycle, we simply add $?\mathtt{Happy}, \mathtt{parent}(a_2)$ to the relaxations.

In these examples, we did not say whether a_i, p, and b are individual names or variables. The difference made by using individual names or variables will become clear when defining the inference problems.

The semantics of conditions is defined as follows.

Definition 2 Let $\mathcal{L}$ be a description logic and $A = (\mathsf{pre}, \mathsf{rel}, \mathsf{post})$ an action with N_X^A the set of variables occurring in pre, rel, or post.

Let $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ be an $\mathcal{L}$-interpretation and $\tau : N_X^A \to \Delta^\mathcal{I}$ an assignment of the variables in A. As in ABoxes, individual names $\alpha \in N_C$ are mapped by $\cdot^\mathcal{I}$ to individuals $\alpha^\mathcal{I} \in \Delta^\mathcal{I}$. For the sake of succinctness, for variables or individual names $a \in N_I$, we define their interpretation as follows:

$$a^{\mathcal{I},\tau} := \begin{cases} \tau(a) & \text{if } a \in N_X \\ a^\mathcal{I} & \text{if } a \in N_C \end{cases}$$

Then $\mathcal{I}$ and τ *satisfy a condition* of the form

$$\begin{array}{llllll} \forall C & \text{if} & C^\mathcal{I} = \Delta^\mathcal{I}, & C(a) & \text{if} & a^{\mathcal{I},\tau} \in C^\mathcal{I}, \\ a \neq b & \text{if} & a^{\mathcal{I},\tau} \neq b^{\mathcal{I},\tau}, \text{ and} & R(a,b) & \text{if} & \langle a^{\mathcal{I},\tau}, b^{\mathcal{I},\tau}\rangle \in R^\mathcal{I}, \end{array}$$

where negated roles are interpreted as usual, i.e., $(\neg R)^\mathcal{I} = \Delta^\mathcal{I} \times \Delta^\mathcal{I} \setminus R^\mathcal{I}$.

To define the semantics of an action, we will first define when the application of an action A to a model $\mathcal{I}$ *possibly yields* another model $\mathcal{I}'$.

Definition 3 Let $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ and $\mathcal{I}' = (\Delta^\mathcal{I}, \cdot^{\mathcal{I}'})$ be two $\mathcal{L}$-interpretations sharing the same interpretation domain and coinciding on the interpretation of individual names $a \in N_C$. Then an action A *possibly yields $\mathcal{I}'$ when applied to $\mathcal{I}$* (written $\mathcal{I} \rightsquigarrow_A \mathcal{I}'$) if there exists an assignment τ such that τ, $\mathcal{I}$, and $\mathcal{I}'$ satisfy the following conditions:

- $\mathcal{I}$ and τ satisfy each pre-condition in pre,
- for each c_1/c_2 in the post-condition in post, if $\mathcal{I}$ and τ satisfy c_1, then $\mathcal{I}'$ and τ satisfy c_2,

In this case, we say that A *possibly yields $\mathcal{I}'$ with τ* when applied to $\mathcal{I}$ (written $\mathcal{I} \rightsquigarrow_A^\tau \mathcal{I}'$).

As mentioned above, this notion of an action transforming one interpretation $\mathcal{I}$ into $\mathcal{I}'$ is too weak. More precisely, $\mathcal{I}$ and $\mathcal{I}'$ may differ largely—as long as they satisfy the pre- and post-conditions. Clearly, this is not what is intended when describing an action. In general, one only wants as few changes/differences between $\mathcal{I}$ and $\mathcal{I}'$ as necessary—with the exception of certain concept or role names X which are mentioned in relaxations $?X$. This idea is formalized in the notion of an action A "yielding" a model $\mathcal{I}'$ when applied to a model $\mathcal{I}$: this is the case if A *possibly* yields $\mathcal{I}'$ *and* if $\mathcal{I}'$ is a model with minimal changes compared to $\mathcal{I}$, i.e., taking back any difference between $\mathcal{I}$ and $\mathcal{I}'$ but those mentioned in relaxations $?X$ would result in a model $\mathcal{I}''$ which A does *not* possibly yield when applied to $\mathcal{I}$.

Definition 4 Two interpretations $\mathcal{I}$ and $\mathcal{I}'$ *differ w.r.t.* τ *in* $d \in \Delta^{\mathcal{I}}$ *and a concept name* C_p if

- $d \in C_p^{\mathcal{I}}$ and $d \notin C_p^{\mathcal{I}'}$ or $d \notin C_p^{\mathcal{I}}$ and $d \in C_p^{\mathcal{I}'}$,
- $?C_p \notin \mathsf{rel}$,
- $?C_p(a) \notin \mathsf{rel}$ for each a with $a^{\mathcal{I},\tau} = d$, and
- $?C_p, R(a) \notin \mathsf{rel}$ for each R and a with $\langle a^{\mathcal{I},\tau}, d\rangle \in R^{\mathcal{I}}$.

Analogously, $\mathcal{I}$ and $\mathcal{I}'$ *differ w.r.t.* τ *in* $\langle d, e\rangle \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ *and a role name* $R \in N_R$ if

- $\langle d, e\rangle \in R^{\mathcal{I}}$ and $\langle d, e\rangle \notin R^{\mathcal{I}'}$ or $\langle d, e\rangle \notin R^{\mathcal{I}}$ and $\langle d, e\rangle \in R^{\mathcal{I}'}$,
- $?R \notin \mathsf{rel}$,
- $?R(a) \notin \mathsf{rel}$ for each a with $a^{\mathcal{I},\tau} = d$,
- $?R(a, b) \notin \mathsf{rel}$ for each a, b with $a^{\mathcal{I},\tau} = d$ and $b^{\mathcal{I},\tau} = e$.

Next, for an action A, an $\mathcal{L}$-interpretation $\mathcal{I}$ and an assignment τ, we define an ordering $\preccurlyeq_{\mathcal{I},\tau}$ on interpretations which characterizes their "proximity" to $\mathcal{I}$ as follows: for two interpretations $\mathcal{I}' \neq \mathcal{I}''$, we say that $\mathcal{I}' \preccurlyeq_{\mathcal{I},\tau} \mathcal{I}''$ if, for all concept names C_p, role names R, and individuals $d, e \in \Delta^{\mathcal{I}}$,

- if $\mathcal{I}$ and $\mathcal{I}'$ differ w.r.t. τ in d and C_p, then $\mathcal{I}$ and $\mathcal{I}''$ differ w.r.t. τ in d and C_p and
- if $\mathcal{I}$ and $\mathcal{I}'$ differ in w.r.t. τ in $\langle d, e\rangle$ and R, then $\mathcal{I}$ and $\mathcal{I}''$ differ in w.r.t. τ in $\langle d, e\rangle$ and R.

An action A *yields* $\mathcal{I}'$ *with* τ when applied to $\mathcal{I}$ (written $\mathcal{I} \longrightarrow_A^{\tau} \mathcal{I}'$) if $\mathcal{I} \rightsquigarrow_A^{\tau} \mathcal{I}'$ and no such interpretation is closer to $\mathcal{I}$ than $\mathcal{I}'$, i.e., there is no $\mathcal{J} \neq \mathcal{I}$ with $\mathcal{I} \rightsquigarrow_A^{\tau} \mathcal{J}$ and $\mathcal{J} \preccurlyeq_{\mathcal{I},\tau} \mathcal{I}'$.

A *service* is a sequence of actions. For a service $S = A_1 \cdots A_n$ and $\mathcal{I}, \mathcal{I}'$ two $\mathcal{L}$-interpretations, we say that S *yields* $\mathcal{I}'$ *with* τ *when applied to* $\mathcal{I}$ (written $\mathcal{I} \longrightarrow_S^{\tau} \mathcal{I}'$) if there exist interpretations $\mathcal{I}_1, \ldots, \mathcal{I}_n$ with $\mathcal{I}_1 = \mathcal{I}$ and $\mathcal{I}_n = \mathcal{I}'$ such that A_i yields $\mathcal{I}_i$ with τ when applied to $\mathcal{I}_{i-1}$ for $2 \leq i \leq n$. We say that S *yields* $\mathcal{I}'$ *when applied to* $\mathcal{I}$ and write $\mathcal{I} \longrightarrow_S \mathcal{I}'$ if there exists an assignment τ with $\mathcal{I} \longrightarrow_S^{\tau} \mathcal{I}'$.

A service S is *realizable* if there exist $\mathcal{L}$-interpretations $\mathcal{I}$ and $\mathcal{I}'$ with $\mathcal{I} \longrightarrow_S \mathcal{I}'$. A service S is *realizable in an $\mathcal{L}$-ABox $\mathcal{A}$* if there exists an $\mathcal{L}$-model $\mathcal{I}$ of $\mathcal{A}$ and an $\mathcal{L}$-interpretation $\mathcal{I}'$ with $\mathcal{I} \longrightarrow_S \mathcal{I}'$.

A service $S = A_1 \cdots A_n$ *is subsumed by* a service $S' = A'_1 \cdots A'_{n'}$ (written $S \sqsubseteq S'$) if, for all $\mathcal{I}$ and $\mathcal{I}'$, if $\mathcal{I} \longrightarrow_S \mathcal{I}'$, then $\mathcal{I} \longrightarrow_{S'} \mathcal{I}'$.

Let $\mathcal{A}$ be an ABox (i.e., a set of concept-, role-, and inequality assertions) and Θ a set of conditions (cf. Definition 1) on individual names occurring in $\mathcal{A}$. Then we say that a service S *generates* Θ from $\mathcal{A}$ if, for all conditions $\theta \in \Theta$, all models $\mathcal{I}$ of $\mathcal{A}$, and all interpretations $\mathcal{I}'$ with $\mathcal{I} \longrightarrow_S \mathcal{I}'$, $\mathcal{I}'$ satisfies θ.

Finally, subsumption, realizability, and generation are defined w.r.t. an $\mathcal{L}$-TBox $\mathcal{T}$ in the obvious way, i.e., by replacing each occurrence of "interpretation" with "model of $\mathcal{T}$" in the respective definition.

Continuing the bicycle selling example, we can easily see that

$$\begin{array}{l}(\{\texttt{owns}(a_1,b),\texttt{wants}(a_2,b),\texttt{owns}(a_2,p),\texttt{Bicycle}(b)\},\\ \emptyset,\\ \{\emptyset/\texttt{owns}(a_2,b),\emptyset/\texttt{owns}(a_1,p)\})\end{array} \tag{4}$$

does *not* subsume

$$\begin{array}{l}(\{\texttt{owns}(a_1,b),\texttt{wants}(a_2,b),\texttt{owns}(a_2,p),\texttt{Bicycle}(b)\},\\ \emptyset,\\ \{\emptyset/\texttt{owns}(a_2,b),\emptyset/\texttt{owns}(a_1,p),\emptyset/\texttt{Unhappy}(a_1)\}).\end{array} \tag{5}$$

In contrast, adding the relaxation ?$\texttt{Unhappy}$ or ?$\texttt{Unhappy}(a_1)$ to the Service 4 yields a service which *does* subsume the Service 5.

Some remarks are in order: (a) The formulation "a service yields an interpretation when applied to another interpretation" might imply a determinism of services. However, relaxations and the existential quantification of assignments introduce non-determinism in the sense that a service, when applied to one interpretation, may yield a variety of interpretations.

(b) The semantics is such that, if $\mathcal{I} \longrightarrow_S \mathcal{I}'$, then individual names and variables are mapped to the same individuals by $\mathcal{I}$ and $\mathcal{I}'$. More precisely, if $S = A_1 \cdots A_n$, then the individual names and variables in A_i are mapped to the same individuals in $\mathcal{I}$, $\mathcal{I}'$, and all "intermediate" interpretations (please note that all these interpretations share the same domain $\Delta^{\mathcal{I}}$). However, in the definition of subsumption, $\mathcal{I} \longrightarrow_S \mathcal{I}'$ might involve an assignment different from the one for $\mathcal{I} \longrightarrow_{S'} \mathcal{I}'$.

Hence, in the following examples, for $\alpha, \hat{\alpha}, \beta, \hat{\beta} \in N_C$ and $x, y \in N_X$, S is *not* subsumed by S_1, but by S_2, which subsumes (and is subsumed by) S_3. Please note that this difference is solely due to the usage of variables in the place of individual names.

$$
\begin{aligned}
S &:= (\{A(\alpha), \neg A(\beta)\}, \emptyset, \{\emptyset/\neg A(\alpha), \emptyset/A(\beta)\}) \\
S_1 &:= (\{A(\hat{\alpha}), \neg A(\hat{\beta})\}, \emptyset, \{\emptyset/\neg A(\hat{\alpha}), \emptyset/A(\hat{\beta})\}) \\
S_2 &:= (\{A(x), \neg A(y)\}, \emptyset, \{\emptyset/\neg A(x), \emptyset/A(y)\}) \\
S_3 &:= (\{A(y), \neg A(x)\}, \emptyset, \{\emptyset/\neg A(y), \emptyset/A(x)\})
\end{aligned}
$$

Algorithms deciding these inference problems can be used for the aforementioned support in the annotation of web pages and service discovery. Firstly, a web page providing a service should be annotated with a description of this service that is *realizable*, i.e., a system service deciding realizability would be useful. Next, a hierarchy of services w.r.t. the subsumption relation can be useful when constructing an ontology of services, and this hierarchy can be computed using a decision procedure for the subsumption problem of services. Moreover, an agent searching for a service S can return all web pages providing a service S' subsumed by (equivalent to) S. Finally, when trying to find out whether a service S is appropriate for a given task, one is interested in the consequences generated by S when applied in a specific situation. Then one can specify this situation by an ABox $\mathcal{A}$ and the consequences by a set of conditions Θ, and ask the system to test whether S generates Θ from $\mathcal{A}$.

2.1 Complexity of the inference problems

Firstly, it can be easily seen that a service S is realizable iff

$$S \not\sqsubseteq (\{a \neq a\}, \emptyset, \emptyset).$$

Hence realizability can be reduced to subsumption.

Secondly, generation can also be reduced to subsumption since $S = (\mathsf{pre}, \mathsf{rel}, \mathsf{post})$ generates Θ from an ABox $\mathcal{A}$ iff

$$S \sqsubseteq (\mathsf{pre}, \Gamma(S), \{\mathcal{A}/c \mid c \in \Theta\}),$$

for $\Gamma(S)$ the set of relaxations containing $?X$ for each concept or role name X that occurs in S. Intuitively, the presence of $\Gamma(S)$ in the relaxation component of the right-hand service expresses that the "minimality of changes" is cancelled. This is necessary since S may enforce other assertions besides the one in Θ and this should not be prohibited by the definition of the service on the right-hand side.

The decidability of the realizability of certain services follow from results on Temporalized DLs (TDLs). TDLs are temporal logics where worlds are DL interpretations. Different variants were investigated, the most expressive ones can, e.g., be found in [21, 22, 23, 24]. Our approach is closely related to TDLs: if $\mathcal{I} \longrightarrow_S \mathcal{I}'$, then $\mathcal{I}'$ can be said to "be after" $\mathcal{I}$, and the intermediate interpretations are ordered linearly. Thus it is natural to try to translate a service into a TDL concept. Due to the (finite) sequential structure of services, we should be able to translate a service into a TDL concept which only uses the "next" operator. However, TDLs do not provide expressive means to minimize changes.

Let us consider services where *all* role and concept names and role names are relaxed, i.e., we do not require that the changes of the interpretation of any concept or role name is minimized. Then the above mentioned translation of a service into a TDL concept (which uses only the "next" operator) is possible. As a consequence, decidability of the realizability of such services for $\mathcal{ALC}$ or $\mathcal{DLR}$ as the underlying description logics and general TBoxes as background knowledge bases $\mathcal{T}$ is an immediate consequence of the decidability results in [21, 2].

In case we want to minimize the changes w.r.t. to certain role- or concept names (by not mentioning them in relaxations $?X$), these results do not help: please note that minimization of changes of X does not mean that the extensions of X before and after the application of a service coincide. Moreover, we believe that subsumption of services cannot be reduced to any standard inference problem in TDLs.

Summing up, we can reduce all inference problems defined for services to (non-) subsumption of services, and realizability of "fully relaxed" services is decidable if the underlying description logic is $\mathcal{ALC}$ or $\mathcal{DLR}$ and general TBoxes are used as background knowledge bases.

3 Comparison with other formalisms

Frameworks similar to the one presented here have been introduced and investigated both in description logics and in other areas of AI, mainly in reaoning about actions and in planning. In the following, we will briefly compare our framework with three of them. For non-description logic formalisms, we restrict our attention to those which provide more than propositional logic for the description of worlds/situations, i.e., which also allow to make assertions concerning the relational structure of a world. To the best of our knowledge, the non-description logic formalisms differ from the one presented here in that they do not have a notion of subsumption between services, and that intensional inference problems seem to be mostly undecidable, whereas we are aiming at the decidability of these problems.

Planning in Description Logics Description logics have previously been extended to describe actions, their effects, and plans which are composed of actions. The two systems Clasp [6] and Rat [8] are knowledge representation systems based on such extensions. Similar to the formalism presented here, both systems provide a notion of subsumption between plans, and regard the computation of "plan hierarchies" as an important system service. However, they differ in (1) the underlying description logic, (3) the constructors to build complex plans (here called services) from atomic actions, and (3) the semantics of actions: (1) CLASP is based on CLASSIC, whereas RAT is based on $\mathcal{KRIS}$. All three formalisms use pre- and post-conditions to describe actions, but in CLASP, these conditions are restricted to pure conjunctions of atomic concepts, whereas they are extended ABox assertions here and restricted ABox assertions in RAT. (2) Both in the formalism presented here and in RAT, atomic actions can only be chained to (finite) sequences. In contrast, CLASP provides richer constructors to build complex plans/services from atomic ones: it offers a non-deterministic choice operator and an operator repeating some action any finite number of times (i.e., a Kleene-star). (3) Only the framework presented here seems to provide a strong notion of minimal changes: regarding this feature, CLASP is very similar to STRIPS, and thus provides a notion of minimality due to its "add"- and "delete"-lists of atoms.

Reasoning about Actions: Situation Calculus (SC) [13, 14, 15] is a family of logics designed for the representation of and reasoning about actions. In SC, *situations* are objects and actions are functions or relations acting on situations and properties of objects. There exist a great variety of different SCs, and they differ in the following aspects with the framework presented here: *The* Situation Calculus is a (sorted) second order logic in which objects and situations are distinguished, and which provides a special function "do(action(parameters),situation)" which maps a situation, an action, and its parameters to the corresponding successor situation. To describe the effect of actions, so-called effect axioms are formulated. Moreover, one also needs to specify what is left unchanged by an action; this is formulated in so-called frame axioms. Recently, several solutions were proposed to the problem of writing down all these frame axioms. Roughly speaking, in certain settings, the frame axioms can be computed automatically from an axiomatization of the world (which corresponds to our background TBox) and the effect axioms: even though there is still a large number of axioms that has to be taken into account, the user does not need to write them down by hand, but they are generated automatically.

In general, "executability" of a sequence of actions is an interesting, yet undecidable problem. However, one is mostly concerned with plan synthesis, i.e., the automatic generation of a plan which yields a certain goal situation from a certain initial situation. There exists a variety of SC fragments for which plan synthesis is decidable but, to the best of our knowledge, each fragment for which satisfiability or entailment was proven to be decidable [11, 20] cannot describe the relational structure of situations.

Web services: DAML-S is both a language for and an ontology of services for the semantic web [1]. It distinguishes three aspects of a service: (1) the service profile, (2) the service model, and (3) the service grounding, and is designed to support various web-related activities involving services, e.g., discovering, invoking, composing, inter-operating, and monitoring a service. Our framework is designed only for the first task, i.e., for the annotation of web pages with a description of the services they provide such that these services can then be discovered by agents, and to support the user in annotating her web pages with a description of the services they provide. In the DAML-S model, the properties of a service related to this task are described in the service profile. However, it does not become quite clear how these properties will be modeled. Most importantly, it is not clear how the dynamic behaviour of a service is represented declaratively such that an agent can reason about it or compare it with other service descriptions.

4 Summary and Future work

So far, we proposed a framework for the representation of services and defined inference problems that we believe to be useful when annotating web pages or searching for services. The main features of this framework can be summarized as follows.

Actions and hence services non-deterministically transform one description of the world into another one. This is crucial since we thus incorporate an explicit notion of dynamic changes. Moreover, our formalism takes into account the ontological formalisms developed for the semantic web by allowing to describe (the relational structure of) worlds in a standard, very expressive terminological formalism.

Next, the semantics of actions and services is such that they only yield *minimal* changes—besides for those concepts and roles that are explicitly relaxed. This allows a natural, succinct representation of actions and services and an elegant solution of the frame problem.

Future work will firstly include the investigation of the decidability and complexity of these inference problems, which obviously depend on the underlying description logic. Secondly, we think of extending the expressive power of the current framework. For example, one might want to state more precisely in which way the activation of a service changes a world. This could be done, e.g., by stating that carrying out a certain action A only increases (decreases) the interpretation of a role or a concept name. In parallel, we plan to thoroughly compare the expressive power provided by our framework with the one required by the semantic web.

References

[1] A. Ankolekar, M. Burstein, J. Hobbs, O. Lassila, D. Martin, S. McIlraith, S. Narayanan, M. Paolucci, T. Payne, K. Sycara, and H. Zeng. DAML-S: Semantic markup for web services. In *Proceedings of the International Semantic Web Working Symposium (SWWS)*, electronically available at `http://www.semanticweb.org/SWWS/report/`, 2001.

[2] A. Artale, E. Franconi, M. Mosurović, F. Wolter, and M. Zakharyaschev. The $\mathcal{DLR}_{\mathcal{US}}$ temporal description logic. In *Proceedings of the 2001 Description Logic Workshop (DL 2001)*, number 49 in CEUR-WS (http://ceur-ws.org/), pages 96–105, 2001.

[3] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic Web. *Scientific American*, 284(5):34–43, 2001.

[4] D. Calvanese, G. De Giacomo, and M. Lenzerini. On the decidability of query containment under constraints. In *Proc. of the 17th ACM SIGACT SIGMOD SIGART Sym. on Principles of Database Systems (PODS'98)*, pages 149–158, 1998.

[5] G. De Giacomo, L. Iocchi, D. Nardi, and R. Rosati. Moving a robot: the KR&R approach at work. In *Proceedings of the Fifth International Conference on the Principles of Knowledge Representation and Reasoning (KR-96)*, pages 198–209. Morgan Kaufmann, Los Altos, 1996.

[6] P. Devanbu and D. J. Litman. Taxonomic plan reasoning. *Artificial Intelligence*, 84(1–2):1–35, 1996.

[7] D. Fensel, F. van Harmelen, I. Horrocks, D. McGuinness, and P. F. Patel-Schneider. OIL: An ontology infrastructure for the semantic web. *IEEE Intelligent Systems*, 16(2):38–45, 2001.

[8] J. Heinsohn, D. Kudenko, B. Nebel, and H.-J. Profitlich. RAT — representation of actions using terminological logics. In J. Heinsohn and B. Hollunder, editors, *DFKI Workshop on Taxonomic Reasoning — Proceedings*, number D-92-08 in DFKI Document. DFKI, Saarbrücken/Kaiserslautern, Germany, 1992.

[9] I. Horrocks and U. Sattler. Ontology reasoning in the $\mathcal{SHOQ}(D)$ description logic. In B. Nebel, editor, *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-01)*, pages 199–204. Morgan Kaufmann, Los Altos, 2001.

[10] I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for very expressive description logics. *Logic Journal of the IGPL*, 8(3):239–264, 2000.

[11] P. Liberatore. The complexity of the language $\mathcal{A}$. *Electronic Transactions on Artificial Intelligence*, 1(1-3):13–38, 1997.

[12] C. Lutz. Adding numbers to the $\mathcal{SHIQ}$ description logic—First results. In *Proceedings of the Eighth International Conference on the Principles of Knowledge Representation and Reasoning (KR-02)*. Morgan Kaufmann, Los Altos, 2002. To appear.

[13] J. McCarthy and P. J. Hayes. Some philosphical problems from the standpoint of artificial intelligence. In *Machine Intelligence 4*. Edinburgh University Press, Edinburgh, 1969.

[14] R. Reiter. The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In V. Lifschitz, editor, *Artificial Intelligence and Mathematical Theory of Computation*, pages 359–380. Academic Press, 1991.

[15] R. Reiter. *Knowledge in Action.* The MIT Press, 2001.

[16] S. Russell and P. Norvig. *Artificial Intelligence. A Modern Approach.* Prentice-Hall, Englewood Cliffs, 1995.

[17] U. Sattler. Description logics for the representation of aggregated objects. In W.Horn, editor, *Proceedings of the 14th European Conference on Artificial Intelligence.* IOS Press, Amsterdam, 2000.

[18] A. Schaerf. Reasoning with individuals in concept languages. In *Proceedings of the Third Conference of the Italian Association for Artificial Intelligence (AI*IA-93)*, Lecture Notes in Artificial Intelligence. Springer-Verlag, 1993.

[19] M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.

[20] E. Ternovskaia. Automata theory for reasoning about actions. In D. Thomas, editor, *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 153–159. Morgan Kaufmann, Los Altos, 1999.

[21] F. Wolter and M. Zakharyaschev. Satisfiability problem in description logics with modal operators. In A. G. Cohn, L. Schubert, and S. C. Shapiro, editors, *Proceedings of the Sixth International Conference on the Principles of Knowledge Representation and Reasoning (KR-98)*, pages 512–523. Morgan Kaufmann, Los Altos, 1998.

[22] F. Wolter and M. Zakharyaschev. Modal descriptiopn logics: modalizing roles. *Fundamentae Informaticae*, 39:411 – 438, 1999.

[23] F. Wolter and M. Zakharyaschev. Temporalizing description logic. In D. Gabbay and M. de Rijke, editors, *Frontiers of Combining Systems*, pages 379 – 402. Studies Press/Wiley, 1999.

[24] F. Wolter and M. Zakharyaschev. Dynamic description logic. In K. Segerberg, M. de Rijke, H. Wansing, and M. Zakharyaschev, editors, *Advances in Modal Logic, Volume 2.* CSLI Publications, 2000.