Multi-Attribute Decision Making using Weighted Description Logics

Erman Acar, Christian Meilicke

Research Group Data and Web Science
Universität Mannheim
Mannheim, Germany
{erman, christian}@informatik.uni-mannheim.de

Abstract. We introduce a framework based on Description Logics, which can be used to encode and solve decision problems in terms of combining inference services in DL and utility theory to represent preferences of the agent. The novelty of the approach is that we consider ABoxes as alternatives and weighted concept and role assertions as preferences in terms of possible outcomes. We discuss a relevant use case to show the benefits of the approach from the decision theory point of view.

1 Introduction

Preference representation is an ongoing research subject in artificial intelligence, gaining more popularity every day. Since the first attention of multi-attribute utility theory in [10, 6], numerous approaches have been done, including probabilistic, possibilistic, fuzzy and graphical models [3, 19, 9, 8] amongst others. One recent approach stepping forward over the last decade is logical languages [2, 20, 4, 11, 21, 15, 16, 18, 17] to encode decision-theoretic problems

Description Logics (DL) is a family of logic languages which is mainly based on decidable fragments of first order logic. It has been designed to be used as a formalism in the field of knowledge representation, and it has become one of the major approaches over the last decade. In the context of the Semantic Web, it embodies a theoretical foundation for the OWL Web Ontology Language, a standard defined by the World Wide Web Consortium.

In this paper we introduce a Description Logic framework, which can be used to encode and solve decision problems in terms of combining inference services in Description Logics and utility theory to represent preferences of the decision maker. Within our approach we consider ABoxes as alternatives and weighted concept and role assertions as preferences in terms of possible outcomes. We discuss some relevant cases and restrictions about our framework.

The framework that we propose in this paper works with classical-DLs, and it can be applied to decision making scenarios where uncertainty is not involved e.g,. transportation model, or the theory of consumer choice [5]. It can be used, for instance, as a core component of a web-based decision support system for e-shopping. In general, it can be applied to every domain where background knowledge which is relevant for our decisions, can be shared, matched and related in terms of ontologies. Within a logic-based decision making framework it is possible to evaluate an *alternative*, or *choice* in terms of its logical implications. This is important in terms of providing the (logical) *rationality* of the agent. In the case of DL, representing *attributes* or *criteria* in terms of concepts, one can express also the dependency between attributes using the concept hierarchy. This in principle, defines indirectly a multi-attribute utility function, using only the relevant attributes. In general, using logical implication with weighted (logical) formulae, allows one to (partially) define a multiattribute utility function [11, 21, 15, 16]. Such a function parametrized over some formulae, is (partially) additive in terms of weights of the implied formulae. This feature provides convenience in preference elicitation as well as computational complexity of the utility function. We remark that the scope of this paper does not include elicitation of preferences, and the complexity of the employed approach, which is a part of the future work plan.

In our work, we have not specified any specific language of DL, since the core idea is regardless of the chosen language. Therefore, we used the basic DL language $\mathcal{ALC}$ to introduce our framework. However, for convenience, we have used the DL language with concrete domains in the example section (Section 3.3), since numerical domains are typically used in Decision Theory.

In the remainder of the paper, we first briefly present preliminaries in utility theory and DL in Section 2. Then, we introduce our framework and discuss an example (Section 3). In the example, an agent (car buyer) is giving a decision between two alternatives (two cars), according to her criteria. In Section 4, we discuss the related works. Finally, we conclude and give a brief outline for future research in Section 5.

2 Preliminaries

In this section, first we will give a basic introduction to preferences and utility theory. Then we briefly inform the reader about our notation for DL.

2.1 Preferences and Utility

In *prescriptive decision theory* [10] it is useful to suppose the existence of a hypothetical *preference order*, a relation defined over choices of the agent.

Definition 1 (Preference). *Let $X = \{x_1, \ldots, x_n\}$ be a set of choices, and a rational* preference *is a complete and transitive binary relation $\succeq$ on X. Then, for any x_i, $x_j \in X$ where $i, j \in \{1 \ldots n\}$,* strict preference *and* indifference *is defined as follows:*

- $x_i \succ x_j$ *iff* $x_i \succeq x_j$ *and* $x_j \not\succeq x_i$ (Strict preference),
- $x_i \sim x_j$ *iff* $x_i \succeq x_j$ *and* $x_j \succeq x_i$ (Indifference).

It is said that, a is *weakly preferred*[1] (*strictly preferred*) to b whenever $a \succeq b$ ($a \succ b$), a is indifferent to b whenever $a \sim b$.

In order to represent the preference relation numerically, one introduces the term *utility*, which is is a function that maps a choice from the choice set to a positive real number reflecting the degree of usefulness. From now on, we will consider only the case which X is finite.

Definition 2 (Utility Function). *Given a finite choice set $X = \{x_1, \ldots, x_n\}$, and preference $\succeq$ on X. Then $u : X \to \mathbb{R}$ is a* ***utility function*** *if for any x_i, $x_j \in X$ with $i, j \leq n$, the following holds:*

$$\begin{aligned} x_i \succ x_j &\iff u(x_i) > u(x_j) \ , \\ x_i \succeq x_j &\iff u(x_i) \geq u(x_j) \ , \\ x_i \sim x_j &\iff u(x_i) = u(x_j) \ . \end{aligned}$$

For the proof of such a function exist, we refer the reader to the so-called representation theorems in [7]. Occasionally, we will represent $\succeq$ in terms of the (respective) utility function as a set (of pairs) $U_{\succeq} = \{\langle x_1, u(x_1)\rangle, \langle x_2, u(x_2)\rangle, \ldots, \langle x_n, u(x_n)\rangle\}$, where $x_1, \ldots, x_n \in X$ (the choice set that $\succeq$ is defined) and $u(x_1) \ldots u(x_n) \in \mathbb{R}^+$ with $u(x_i) \neq u(x_j) \implies x_i \neq x_j$.

The basic principle in utility theory is that a rational agent should always try to maximize its utility, or *should take the choice with the highest utility.* Note that the decisions in real world are far more complex than requiring to consider just a single *criteria* (e.g. unary utility functions). Multi-attribute utility functions is an approach to deal with such decision problems.

Definition 3 (Multiattribute Utility Function). *Let $X = X_1 \times \ldots \times X_n$ be the set of multiple attributes over which the decision maker has preferences where $n \geq 2$. Let $\succeq$ be the preference relation defined on X, then u is a multiattribute utility function representing $\succeq$ if and only if $\forall (x_1, \ldots, x_n), (y_1, \ldots, y_n) \in X$,*

$$(x_1, \ldots, x_n) \succeq (y_1, \ldots, y_n) \iff u(x_1, \ldots, x_n) \geq u(y_1, \ldots, y_n) \ . \tag{1}$$

2.2 Description Logics

It is assumed that the reader has some familiarity with DL. If that is not the case, we refer the interested reader to [1]. The framework that we are presenting is independent from the choice of a specific DL language. We will recall the brief information of DL, to clarify the notation used and to cover the needed knowledge for the framework and the example.

The signature of the DL language we use, is (N_C, N_R, N_I), where N_C is the set of atomic concepts, N_R is the set of role names, and N_I is the set of individuals. Along the text, we assume the *unique name assumption*, which means that different individuals have different names. We denote concepts by

[1] It is also called *preference-indifference* relation, since it is the union of strict preference and indifference relation.

C and D, roles by R and S, and individuals as a and b. Concept descriptions are defined inductively by N_C, $\neg C$, $C \sqcap D$, and $C \sqcup D$ if C and D are concept descriptions, and $\exists R.C$ and $\forall R.C$ if $R \in N_R$ and C is a concept description. The top concept $\top$ is abbreviation for $C \sqcup \neg C$ and the bottom concept $\bot$ is for $\neg\top$. An interpretation is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ where the domain $\Delta^{\mathcal{I}}$ is a non-empty set and $\cdot^{\mathcal{I}}$ is interpretation function that assigns to every concept name C a set $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and to every role name R a binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. It is defined inductively for every concept description as follows; $(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$, $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$, $(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$, $(\exists R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \exists b.(a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}$, and $(\forall R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \forall b.(a,b) \in R^{\mathcal{I}} \implies b \in C^{\mathcal{I}}\}$. Any other extension is defined accordingly, and will be clarified when it is necessary.

In Description Logics, there is a distinction between *terminological knowledge* (TBox) and *assertional knowledge* (Abox). TBox is a set of concept inclusion axioms: $C \sqsubseteq D$ where the interpretation is $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. $C \equiv D$ if $C \sqsubseteq D$ and $C \sqsubseteq D$. ABox is a set of concept assertions $C(a)$ where $a \in N_I$ and $C(a)^{\mathcal{I}} = a^{\mathcal{I}} \in C^{\mathcal{I}}$, and role assertions $R(a,b)$ where $(a,b) \in N_I \times N_I$ and $R(a,b)^{\mathcal{I}} = (a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$.

A concept is satisfiable if there is an interpretation $\mathcal{I}$ such that $C^{\mathcal{I}} \neq \emptyset$. A concept is satisfiable with respect to $\mathcal{T}$ if and only if there is a model $\mathcal{I}$ of $\mathcal{T}$ such that $C^{\mathcal{I}} \neq \emptyset$. A concept inclusion $C \sqsubseteq D$ is said to be satisfiable if and only if there is an $\mathcal{I}$ which respects $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ (i.e. $I \models C \sqsubseteq D$). A concept C is subsumed by a concept D with respect to $\mathcal{T}$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for every model of $\mathcal{I}$ of $\mathcal{T}$ (i.e. $C \sqsubseteq_{\mathcal{T}} D$ or $\mathcal{T} \models C \sqsubseteq D$). If $\mathcal{T}$ is a set of axioms, then $\mathcal{I}$ is a model of $\mathcal{T}$ if and only if $\mathcal{I}$ satisfies every element of $\mathcal{T}$. Such a TBox is called *coherent*. We say that an assertion α is entailed by ABox $\mathcal{A}$ (i.e. $\mathcal{A} \models \alpha$) if every model of $\mathcal{A}$ also satisfies α. One basic reasoning service we will use is *instance check*; to check for a given ABox $\mathcal{A}$ and α, weather $\mathcal{A} \models \alpha$ holds. An ABox $\mathcal{A}$ is consistent w.r.t. a TBox $\mathcal{T}$ if there is a model of both $\mathcal{T}$ and $\mathcal{A}$. We call the pair $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ a knowledge base, and also say that K is satisfiable if $\mathcal{A}$ is consistent w.r.t. $\mathcal{T}$.

A concrete domain $\mathcal{D}$ is a pair $(\Delta^{\mathcal{D}}, \mathit{pred}(\mathcal{D}))$ where $\Delta^{\mathcal{D}}$ is the domain of $\mathcal{D}$ and $\mathit{pred}(\mathcal{D})$ is the set of predicate names of $\mathcal{D}$. It is assumed that $\Delta^{\mathcal{I}} \cap \Delta^{\mathcal{D}} = \emptyset$, and each $P \in \mathit{pred}(\mathcal{D})$ which is of arity n, is associated with $P^{\mathcal{D}} \subseteq (\Delta^{\mathcal{D}})^n$. We will denote *functional roles* with lower case r. In DL with concrete domains, it is assumed that N_R is partitioned into a set of *functional roles* and the set of ordinary roles. A role r is *functional* if for every $(x,y) \in r$ and $(w,z) \in r$ it implies that $x = w \implies y = z$. Functional roles, in the extended language, is interpreted as partial functions from $\Delta^{\mathcal{I}}$ to $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{D}}$. Functional roles and ordinary roles are both allowed to be used with both the existential quantification and the universal quantification. Concrete domain is required to be closed under negation (denoted by $\overline{P}$), in order to be able to compute the negation normal form of the concepts defined via extended constructs. For more information about DL, we refer the interested reader to [1].

3 Approach

We consider a decision problem (in the terminology of decision theory) from the agent perspective: *in the light of background knowledge and preferences which alternative should be chosen?* We should note that, in this paper we do not concern ourselves with problems regarding elicitation or uncertainty. In this regard, we assume that agents preferences are elicited and there is no uncertainty. This is usually the case for decision making in the domain of consumer choice theory, ([5]). Furthermore, we note that the formalism is created concerning further possible extensions to formalize sequential decisions, policies and game theoretical concepts later.

3.1 Representing a Decision Problem and Utilities

We represent the background knowledge of the agent by the DL knowledge base $\mathcal{K}$, which includes the concept hierarchy $\mathcal{T}$ and *known assertions* about individuals which are represented in $\mathcal{A}$. The choice set $\mathcal{C}$ represents *a priori* alternatives which utilities yet are unknown by the agent. $\mathcal{U}$ is the set of criteria or outcomes where each element consists of an assertion a_i and a value u_i assigned to this assertion with the condition that every a_i has a unique u_i. We will use the terms *criteria*, *attributes* and *outcomes* interchangeably.

Definition 4 (Decision Base). *A decision base, $\mathcal{D} = (\mathcal{K}, \mathcal{C}, \mathcal{U})$ is a triple with;*

- $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ *is a description logic* knowledge base *(background knowledge) in which $\mathcal{T}$ is an acyclic TBox and $\mathcal{A}$ is an ABox,*
- $\mathcal{C} = \{C_1, \ldots, C_n\}$ *is a* choice box*, a non-empty finite set of* choices*, each being an ABox,*
- $\mathcal{U} = \{\langle a_1, u_1\rangle, \ldots, \langle a_m, u_m\rangle\}$ *is a* utility box *(UBox), a finite set of* utility assertions*, in which a_i being an assertion and $u_i \in \mathbb{R}$ is the assigned basic* utility value *for a_i, with the restriction $a_i \equiv_{\mathcal{T}} a_j \implies u_i = u_j$.*

Note that $\mathcal{U}$ can be inconsistent in terms of arbitrary unions of a_is with regard to $\mathcal{K}$. Another way to look at it, is to think of it as a (possibly inconsistent) union of ABoxes. One can think of a utility assertion as an outcome, or an instantiation of an attribute, and the value u_i is the corresponding basic (uninferred) utility value for that outcome. This gives us the flexibility to model a decision problem in various ways in terms of preferences.

The first way is, if one interprets a concept name in UBox, as a single attribute then the preference is defined just as they were defined in parallel to the standard multi attribute utility theory [10]. For instance, if $Colour$ is an attribute, the instantiation of its value red corresponds to an outcome $Colour(red)$ with its basic utility u_i, which is expressed in the form of $\langle Colour(red), u_i\rangle$. In some cases, this might seem contrary to the usual way of expressing ontologies in DL. However, by this example we emphasize the enough expressivity of DL. Furthermore, it allows us to define utility values for complex outcomes explicitly; $\langle Colour(red) \sqcap Size(small), u_j\rangle$, *having the colour red and the size is small.*

Another obvious way to model preferences in a decision problem is to regard a concept name in UBox as a criteria or property (possibly defined or interrelated to other concepts via background knowledge $\mathcal{K}$), e.g. $GourmetTrip(trip_1)$, $EconomicTrip(trip_2)$. The utility of a choice is defined straightforward.

Definition 5 (Utility). *The* utility U *of a choice* C *w.r.t* $\mathcal{D} = (\mathcal{K}, \mathcal{C}, \mathcal{U})$ *is,*

$$U(C) = \Sigma_{\{\langle a,u\rangle \in \mathcal{U} | \mathcal{K} \cup C \models a\}} u$$

where $C \in \mathcal{C}$ *and* $\mathcal{K} \cup C$ *is consistent.*

From the definition above, it follows that the utility of an inconsistent alternative (with respect to the knowledge base) is undefined. Thus we restrict ourselves to assess only the consistent decisions. This naturally provides a service to eliminate alternatives which can cause inconsistencies. Note that one could also define utilities for inconsistent decisions simply by extending the definition, e.g., assigning zero, however, this causes an inconsistent decision and a possible zero-utility choice to be regarded indifferently in terms of utility score, if no restriction on $\mathcal{U}$ is applied, e.g., $u_i \geq 1$. As the consistency of a decision with respect to background knowledge or a previously taken decision is critical to a decision maker, so it is for a decision support system. In case of logical languages, hence of DL-based ontologies, *consistency checking* is a standard reasoning service. A decision support system based on this framework, whose choices are given, could easily help the decision maker in cases it is hard to see the logical implications and possible inconsistencies.

Notice that calculating the utility of a choice, can be thought of as answering a series of *instance checking* inferences i.e. $\mathcal{K} \cup C_i \models^? a_1, \mathcal{K} \cup C_i \models^? a_2, \ldots, \mathcal{K} \cup C_i \models^? a_{|\mathcal{U}|}$ and collecting positive answers. Now, given the decision base and the utility of a choice C_i, one can define a decision problem. A typical form of the decision problem would be finding the best decision expressed as choosing the choice with the maximum utility:

$$C_{max} = \arg\max_{C} \{U(C) \mid C \in \mathcal{C}\} \tag{2}$$

This can be generalized in terms of picking up the best n-choices together.

$$C^n_{max} = \arg \max_{(C_1,\ldots,C_n)} \{U(\bigcup_{i=1}^{n} C_i) \mid C_1, \ldots, C_n \in \mathcal{C} \text{ and } n \leq |\mathcal{C}|\} \tag{3}$$

Or it can be logically restricted to a level that the decision maker can pick up at most one choice (mutually exclusive), with the following definition.

Definition 6 (Mutual Exclusion). *A decision base* $\mathcal{D} = (\mathcal{K}, \mathcal{C}, \mathcal{U})$ *is* mutually exclusive *if for every* $C_i, C_j \in \mathcal{C}$ *with* $i \neq j$, $C_i \cup C_j \cup \mathcal{K}$ *is inconsistent.*

In general, in order to model the concerned type of a decision problem, one can bring some restrictions on $\mathcal{C}$ and $\mathcal{U}$.

Proposition 1. *The utility function U induces a rational preference relation.*

Proof. Since the codomain of U is R^+, $\geq$ is a complete quasiorder (complete and transitive). For any two consistent (w.r.t $\mathcal{K}$) choices C_1 and C_2, set $C_1 \succeq C_2$ if $U(C_1) \geq U(C_2)$, and set $C_1 \sim C_2$ if $C_1 \geq C_2$ and $C_2 \geq C_1$.

□

3.2 About the Expressivity of $\mathcal{D}$

Since utility functions represent preferences, it is well-known that certain classes of utility functions correspond to certain type of preferences. In this section, we will discuss some of the expressivity of $\mathcal{D}$ (defined in Definition 5) in terms of the utility functions. Following [4], let us give some definitions of well-known utility classes first.

Definition 7 (Utility Function Classes). *Let U be a utility function (as in Definition 5), and C_1, C_2, C_3 are pairwise consistent ABoxes. Then,*

1. *U is non-negative iff $U(C) \geq 0$ for all C.*
2. *U is monotonic iff $U(C_1) \leq U(C_2)$ whenever $C_1 \subseteq C_2$.*
3. *U is subadditive iff $U(C_1 \cup C_2) \leq U(C_1) + U(C_2) - U(C_1 \cap C_2)$ for all C_1 and C_2.*
4. *U is superadditive iff $U(C_1 \cup C_2) \geq U(C_1) + U(C_2) - U(C_1 \cap C_2)$ for all C_1 and C_2.*
5. *U is concave iff $U(C_1 \cup C_2) - U(C_2) \leq U(C_1 \cup C_3) - U(C_3)$ for all C_1 whenever $C_3 \subseteq C_2$.*
6. *U is convex iff $U(C_1 \cup C_2) - U(C_2) \geq U(C_1 \cup C_3) - U(C_3)$ for all C_1 whenever $C_3 \subseteq C_2$.*
7. *U is modular iff $U(C_1 \cup C_2) = U(C_1) + U(C_2) - U(C_1 \cap C_2)$ for all C_1 and C_2.*

Monotonicity means, more of a good (or choice) is better. Concavity means that if we move (from C_3) to a better position (or the choice C_2), the marginal utility (of the choice C_1) decreases. This describes the behaviour of *risk-averse* agents. The opposite occurs when the function is convex; exposing a *risk-seeking* behaviour. Modularity is the intersection of both classes. From the informal argument in [4], we state the following proposition.

Proposition 2. *Let U be a utility function, then: (1) if U is concave, then it is subadditive, (2) if U is convex, then it is superadditive.*

Proof. Set C_3 as $C_1 \cap C_2$.

□

The following negative result will help us to discuss the expressive power of U w.r.t. $\mathcal{D}$.

Theorem 1. *U is (1) not non-negative, (2) not monotonic, (3) not subadditive, (4) not superadditive, (5) not concave, (6) not convex, (7) not modular w.r.t. some $\mathcal{D}$.*

Proof. (1) follows from Definition 4 and Definition 5 by setting basic utilities as negative reals. *(2)* follows from *(1)*. To prove *(3)*, set $\mathcal{K} = \emptyset$, $C_1 = \{D(a)\}$, $C_2 = \{E(a)\}$ and $\mathcal{U} = \{\langle D(a), 10\rangle, \langle E(a), 10\rangle \langle (D \sqcap E)(a), 100\rangle\}$. *(4)* follows from setting $\mathcal{K} = \emptyset$, $C_3 \neq \emptyset$, $C_3 \subset C_2 \subset C_1$, and UBox $\mathcal{U}$ as the non-negative assertions of C_1. *(5)* follows from the contrapositive of Proposition 2- *1.* *(6)* follows from the contrapositive of Proposition 2-*2.* *(7)* follows from both *(5)* and also *(6)*. □

Observe that Theorem 1 follows from the expressive power and therefore the flexibility of U (w.r.t $\mathcal{D}$). Therefore, with adequate restrictions on $\mathcal{D}$ (in particular UBox $\mathcal{U}$), one might change U into one of the aforementioned class (in Definition 7). For instance defining basic utilities non-negative ($u_i \in \mathbb{R}^+$) would guarantee the non-negativity (trivially), and monotonicity (since $\models$ is also monotone). Observe that in $\mathcal{U}$ one can express *complement attributes* (as in the proof of Theorem 1.3 that is, the utility of having both criteria is greater than sum of each (e.g. Hotel reservation and Plane ticket of a holiday). Similarly one can express *substitute attributes* e.g., assigning a negative value to having both attribute. Not allowing *complementary* attributes and setting $\mathcal{U}$ non-negative would guarantee modularity. Certainly, investigations over such restrictions and their interrelations need a closer inspection, which we plan to do in future work.

3.3 Example: Car Buyer

Consider an agent who wants to buy a second hand sports car. After visiting various car dealers, he finds two alternatives as fair deals; a sport Mazda (*Mx-5 Miata Roadster, 2013*) which fits his original purpose and a BMW (*335i Sedan, 2008*) which is also worth considering since it has a very strong engine (*300 horsepower (hp)*) and also comes with a sport kit. The car buyer's decision base (background knowledge $(\mathcal{T}, \emptyset)$, choices $\mathcal{C}$, and criteria $\mathcal{U}$) is as in Figure1.

As the use of numerical domains is common to classical Decision Theory, we will use the language with concrete domains. If the reader is already familiar with concrete domains, she can skip the technical definitions and move directly to Figure 1.

Let us clarify concrete domains and predicates which are used in the example. We take the concrete domain Car and $\Delta^{Car} = \Delta^{\$} \cup \Delta^{sec} \cup \Delta^{mpg} \cup \Delta^{mph} \cup \Delta^{hp}$ with $\Delta^{\$} \cap \Delta^{sec} \cap \Delta^{mpg} \cap \Delta^{mph} \cap \Delta^{hp} = \emptyset$, and $pred(Car) = pred(\$) \cup pred(mpg) \cup pred(mph) \cup pred(sec)$. We define the partition (of the domain Δ^{Car}) $\Delta^{\$}$ as a denumerable set $\{i\$\}$ where $i \in \mathbb{N}$, $pred(\$) = \{<_{\$}, >_{\$}, \geq_{\$}, \leq_{\$}, =_{\$}, \neq_{\$}\}$. $(<_{\$})^{\$}(x, y) = \{(x, y) \in \Delta^{\$} \times \Delta^{\$} \mid i, j \in \mathbb{N}$ with $i\$ = x$ and $j\$ = y$ such that $i < j\}$.

$$\mathcal{T} = \{\exists hasPrice. \leq_{30000\ \$} \equiv InexpensiveCar,$$
$$ExpensiveCar \sqsubseteq HighClassCar,$$
$$HighClassCar \sqsubseteq PrestigiousCar,$$
$$\exists hasFuelConsumpt. \geq_{20mpg} \equiv EconomicCar,$$
$$Roadster \sqsubseteq PrestigiousCar,$$
$$MiddleClassCar \sqcap HighClassCar \sqsubseteq \bot,$$
$$SportsCar \sqcup \exists hasHP. \geq_{200hp} \sqsubseteq StrongCar,$$
$$2Doors \sqcap 4Doors \sqsubseteq \bot,$$
$$\exists has0-60mph. \leq_{7.0sec} \sqcap \exists hasHP. \geq_{270hp} \sqsubseteq VeryStrongCar,$$
$$2Doors \sqcap \neg Convertible \equiv Coupé,$$
$$\neg\ Coupé \sqcap \neg Convertible \sqcap \neg Hatchback \sqsubseteq Sedan,$$
$$2Doors \sqcap \exists has0-60mph. \leq_{7.0sec} \sqcap \forall hasKit.SportKit \sqsubseteq SportsCar,$$
$$Bmw \sqcap Mazda \sqsubseteq \bot,$$
$$Bmw335i \sqsubseteq Bmw,$$
$$\exists hasModelYear. \geq_{2012} \equiv YoungCar,$$
$$Bmw \sqsubseteq PrestigiousCar,$$
$$SportsCar \sqcap Convertible \equiv Roadster,$$
$$ClassicalKit \sqsubseteq Kit,$$
$$SportKit \sqsubseteq Kit,$$
$$Car \sqcap Kit \sqsubseteq \bot,$$
$$ClassicalKit \sqcap SportKit \sqsubseteq \bot\}$$

$$C_1 = \{\ MazdaMx5Miata(car),$$
$$hasHP(car, 167hp),$$
$$hasFuelConsumption(car, 24mpg),$$
$$hasModelYear(car, 2013),$$
$$has0-60mph(car, 6.9sec),$$
$$hasPrice(car, 29960\$),$$
$$Convertible(car),$$
$$2Doors(car),$$
$$hasKit(car, kit)\}$$

$$C_2 = \{\ Bmw335i(car),$$
$$hasHP(car, 300hp),$$
$$hasFuelConsumption(car, 19mpg),$$
$$hasModelYear(car, 2008),$$
$$has0-60mph(car, 4, 8sec),$$
$$hasPrice(car, 42560\$),$$
$$Sedan(car),$$
$$4doors(car),$$
$$SportKit(kit),$$
$$hasKit(car, kit)\}$$

$$\mathcal{U} = \{\ \langle InexpensiveCar(car), 30\rangle,$$
$$\langle PrestigiousCar(car), 55\rangle,$$
$$\langle VeryStrongCar(car), 50\rangle,$$
$$\langle StrongCar(car), 40\rangle,$$
$$\langle EconomicCar(car), 30\rangle,$$
$$\langle YoungCar(car), 35\rangle,$$
$$\langle Convertible(car), 10\rangle,$$
$$\langle Sedan(car), 5\rangle,$$
$$\langle SportKit(kit), 20\rangle,$$
$$\langle ClassicalKit(kit), 10\rangle\}$$

Fig. 1. The car buyer's background knowledge $\mathcal{K} = (\mathcal{T}, \emptyset)$, choices set CBox $\mathcal{C} = \{C_1, C_2\}$, and UBox $\mathcal{U}$. We omit the trivial axioms with the super concept Car: $HighClassCar \sqsubseteq Car$, $PrestigiousCar \sqsubseteq Car \ldots$ etc.

Other predicates are defined similarly in an obvious way parallel to usual binary relations over $\mathbb{N}$. For convenience, we extend *pred*($) with finitely many unary predicates in the form of $<_x = \{\forall y \in \Delta^{\$} \mid <_{\$} (x, y)\}$ and also of $>_x$, $\leq_x$, $\geq_x$, $=_x$, $\neq_x$ which are similarly defined, enough to express the intended TBox. Note that *pred*($) is closed under negation: $\overline{<_{\$}}(x, y) = \geq_{\$} (x, y)$, etc. For other partitions, we take $\Delta^{sec} = \{i\ sec \mid i \in \mathbb{R}^+ - \{0\}\}$, $\Delta^{mpg} = \{i\ mpg \mid i \in \mathbb{N}\}$, $\Delta^{mph} = \{i\ mph \mid i \in \mathbb{N}\}$, $\Delta^{hp} = \{i\ hp \mid i \in \mathbb{N} - \{0\}\}$. The rest of the respective predicate names and functional roles are defined in an obvious way ($hasPrice^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta^{\$}$, $hasKit : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, etc).

According to the agent, taking $\mathcal{T}$ into account, a *Bmw* is a *prestigious car*. Considering a *200 hp* or above is enough to refer to a car as strong. An economic car should go for more than *20 miles per gallon* (*mpg*). A car is *young* if it was manufactured in *2012* or later.

Considering $\mathcal{U}$ in Figure1, the agent (car buyer) is more interested in having a *prestigious car* than having an *inexpensive car*. He prefers *convertible* to *sedan*. However, these are not as important as a car to be an *economic car*, or a *strong car*. Using the given decision base, we can calculate the utility of each choice ($U(C_1) = 220$, $U(C_2) = 170$), which implies (by the assumption: the higher the utility, the more desirable is the choice) that $C_1 \succ C_2$.

The example of the decision problem given above is (intrinsically) mutually exclusive since it was obvious that we were deciding between two choices (buying just one car). Therefore we used a unique individual *car* instead of *car1*, *car2*. Mutual exclusion is implied (e.g., by $Bmw \sqcap Mazda \sqsubseteq \bot$).

3.4 Some Useful Notions

Consider the case where there is an assertion in a particular choice or an expansion of it which is not implied by any utility assertion. For instance, assume that in the *car buyer* example, C_1 has $hasColour(car, red)$. This might be quite important for the decision maker. However, as this outcome is not included in $\mathcal{U}$, it will not be reflected in the evaluation of the utility for C_1. In this case, the utility function, which is implicitly defined for the choice C, does not really capture the implications of C. That means preferences are not *comprehensive* enough in terms of having all the necessary outcome information. This in turn diminishes the quality of accuracy in evaluating the utility of a decision (perhaps in trade-off regarding to ease the storage of preference and save of computational resources). By a comprehensive preference with respect to a choice, we understand a preference structure which captures (having a value assigned) all of its logical implications.

Definition 8 (Comprehensiveness). *Let $\mathcal{U}'$ be the entire set of outcomes ($\{a_i\}$) in $\mathcal{U}$, and $cl_{\mathcal{T}}(C) = \{x \mid C \cup \mathcal{T} \models x\}$ be the closure of a choice C (w.r.t $\mathcal{T}$). Then $\mathcal{U}$ is called* comprehensive *w.r.t. C iff $cl_{\mathcal{T}}(C) \subseteq \mathcal{U}'$.*

Comprehensiveness is also an important property that should be taken into account for a possible interest of automated generation of UBoxes.

Note that one can extend the present framework in terms of considering not only the choices but also an extra available information prior to giving a decision. This case is especially relevant when the agent is considered to have an incomplete background knowledge. The extra information can encoded in terms of an axiom or assertion. This allows us to evaluate the value of information in terms of its utility, with respect to a choice. Informally, the *value of information* w.r.t. a choice C is the difference between the utility of C with the extra information and without. We will consider the axiom case.

Definition 9 (Value of Information). *Let* $\mathcal{D} = ((\mathcal{T}, \mathcal{A}), \mathcal{C}, \mathcal{U})$ *be a decision base and* $\mathcal{D}' = ((\mathcal{T} \cup \mathcal{T}_\alpha, \mathcal{A}), \mathcal{C}, \mathcal{U})$ *be the decision base extended with the additional information* $\mathcal{T}_\alpha$*. Then, the* value of information *with respect to a choice* C *and decision base* $\mathcal{D}$ *is* $U(\mathcal{T}_\alpha) = U_{\mathcal{D}'}(C) - U_{\mathcal{D}}(C)$*, whenever* $\mathcal{T} \cup \mathcal{T}_\alpha \cup \mathcal{C}$ *is consistent.*

For instance, assume that in car buyer example, our agent does not know what a roadster car really is (which means we assume that the TBox in Figure1 does not include the axiom $SportsCar \sqcap Convertible \equiv Roadster$), even though he knows that a roadster is prestigious. Without that information the Mazda does not become a prestigious car in all models of $\mathcal{T}$. That means the utility of C_1 can not get extra 55 utility score for being a $PrestigiousCar$. Thus the value of the regarded information is 55 for C_1 whereas it is 0 for C_2.

4 Related Work

Preference representation using logical languages has become popular over the last decade. Many of these approaches are based on propositional logic [2, 4, 11, 21]. DL languages are used for preference representation in [13, 15, 16, 18, 17]. In [13], Lukasiewiecz and Schellhase introduce a framework in DL to model conditional preferences for matchmaking and ranking objects under conditional preferences with an application to literature search. However utility functions is not a part of their approach. In [18, 17], Ragone et al. use DL in order to work on multi-issue bilateral negotiation via focusing on utilities. In [18], they explain how to use DL to describe request and offers from buyers and sellers. Using the non-standard reasoning service of *concept contraction* to handle conflicts in goods and service descriptions, they present an alternating-offers protocol. In their subsequent work [17], they focus on multi-issue bilateral negotiation with incomplete information. There, for the first time, they introduce the utility of a concept. The utility of a concept (proposal) is defined as the sum of the weights of its superconcepts. In [15, 16], they mainly discuss how to compute utilities. Although their work was mainly developed in the context of multiattribute negotiation, to our knowledge this is the most similar work to our approach.

In [15], Ragone et al. show how to represent preferences using weighted DL-formulas. Claiming that the definition of utility by subsumption yields unintuitive results, they base their modified definition of utility on semantic implication. This means that the utility of a concept C w.r.t. a TBox is defined as the sum of the weights of the concepts that are logically implied by C. According to terminology they used, our approach can be understood as an *implication-based* approach. However, they define logical implication in terms of membership, i.e., $m \models C$ iff $m \in C^{\mathcal{I}}$. The *minimal model* that they introduced in order to define the *minimal utility value* is more restrictive than ordinary models in DL. They change this definition to ordinary models in their next paper [16], while keeping the formal machinery the same (except the way they compute utilities). We should note that their *preference set*, which is a set of weighted concepts, is similar to our UBox. Hence, the main difference of our approach is the formal extension to multiple alternatives and the use of ABoxes, which in turn provides extra expressivity (i.e., one can induce a preference relation over the membership of distinct individuals to the same class e.g., $\mathcal{U} = \{\langle C(a), 20\rangle, \langle C(b), 30\rangle\}$).

In [20], authors show how to encode fuzzy MCDM problems in the formalism of fuzzy DL. They base their work on a standard MCDM feature, a *decision matrix* wherein the performance score of each alternative over each criteria is explicitly stated. Criteria are expressed as fuzzy concepts. Among alternatives, the optimal alternative (w.r.t the fuzzy knowledge base) is the one with the highest *maximum satisfiability degree*. The authors do not explicitly make a distinction between the knowledge base and the set of criteria. In general, the focus of the work is to show the potential and flexibility of fuzzy DL in encompassing the usual numerical methods used in MCDM, rather than leveraging a formal concept hierarchy in MCDM for expressing relations and handling inconsistencies between criteria, alternatives, and the knowledge base.

5 Future Work

We have introduced a framework based on knowledge representation formalism DL, for it can be applied to solve decision problems, i.e., multi-attribute discrete alternatives. Using our formalism, we have also defined formally some concepts such as mutually exclusive choices, comprehensiveness ,value of information which is promising for future directions. One aim is to make a closer investigation on expressive power of $\mathcal{D}$.

As the major part of the utility theory literature is concerned with uncertainty, one major future research direction is to extend the framework with probabilistic description logics, e.g., [12, 14]. This would allow us to access the essential utility theory literature from the DL perspective, along with lots of new application possibilities. In particular, the probabilistic extension would allow us to compute the expected utility of choices (as lotteries) in terms of their logical implications according to the type of the probability the framework is defined (e.g. subjective, statistical).

A second major research direction is to extend the framework to sequential decisions (e.g. $\mathcal{D}_i \rightarrow \mathcal{D}_{i+1}$, sequence of decision bases). Once sequential decisions are defined, one can represent policies, strategies and define a planner.

It can be extended to represent collaborative decision making scenarios as well as game theoretical set-ups by considering more than one agent and specifying restrictions between their choice sets and knowledge bases. As an example, in an arbitrary set-up, rules of the game could be a subset of intersection of both agent's knowledge bases, then the knowledge bases would get extended according to each players choices if each player can see what others choose. It can be checked whether a game-theoretical condition is satisfied, in terms of ontologies.

Currently, we are working on the implementation of the basic framework as a Protégé[2] plug-in. Our plugin is planned to consist, first of all, of an editor for the definition of UBoxes and choices, while the background knowledge is loaded via the standard interfaces of Protégé. Our extension will then be able to compute the utility of the given choices in order to display a ranking. The development of our Protégé plugin is motivated by the idea to demonstrate the benefits of our approach to a set of different application scenarios.

References

[1] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[2] Sylvain Bouveret, Michel Lemaître, Hélène Fargier, and Jérôme Lang. Allocation of indivisible goods: a general model and some complexity results. In Frank Dignum, Virginia Dignum, Sven Koenig, Sarit Kraus, Munindar P. Singh, and Michael Wooldridge, editors, *4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005), July 25-29, 2005, Utrecht, The Netherlands*, pages 1309–1310. ACM, 2005.

[3] Urszula Chajewska, Daphne Koller, and Ronald Parr. Making rational decisions using adaptive utility elicitation. In *Proceedings of the 7th Conference on Artificial Intelligence (AAAI-00) and of the 12th Conference on Innovative Applications of Artificial Intelligence (IAAI-00)*, pages 363–369, Menlo Park, CA, July 30– 3 2000. AAAI Press.

[4] Yann Chevaleyre, Ulle Endriss, and Jérôme Lang. Expressive power of weighted propositional formulas for cardinal preference modelling, December 08 2006.

[5] W. Suen E. Silberberg. *The Structure of Economics: A Mathematical Analysis*. McGraw-Hill/Irwin, June 27, 2000.

[6] P. C. Fishburn. Interdependence and additivity in multivariate, unidimensional expected utility theory. *Intl. Economic Review*, 8:335342, 1967.

[7] Peter C. Fishburn. *Utility Theory for Decision Making*. Robert E. Krieger Publishing Co., Huntington, New York, 1969.

[8] Phan H. Giang and Prakash P. Shenoy. Two axiomatic approaches to decision making using possibility theory. *European Journal of Operational Research*, 162(2):450–467, April 16 2005.

[9] Cengiz Kahraman. *Multi-Criteria Decision Making: Theory and Applications with Recent Developments*. Springer, 2008.

[10] R.L. Keeney and H. Raiffa. *Decisions with multiple objectives: Preferences and value tradeoffs*. J. Wiley, New York, 1976.

[11] Céline Lafage and Jérôme Lang. Logical representation of preferences for group decision making. In Anthony G. Cohn, Fausto Giunchiglia, and Bart Selman, editors, *KR2000: Principles of Knowledge Representation and Reasoning*, pages 457–468, San Francisco, 2000. Morgan Kaufmann.

[2] `http://protege.stanford.edu/`

[12] Thomas Lukasiewicz. Expressive probabilistic description logics. *Artificial Intelligence*, 172(6–7):852–883, 2008.

[13] Thomas Lukasiewicz and Jörg Schellhase. Variable-strength conditional preferences for ranking objects in ontologies. *J. Web Sem*, 5(3):180–194, 2007.

[14] Carsten Lutz and Lutz Schröder. Probabilistic description logics for subjective uncertainty. In Fangzhen Lin, Ulrike Sattler, and Miroslaw Truszczynski, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010, Toronto, Ontario, Canada, May 9-13, 2010*. AAAI Press, 2010.

[15] Azzurra Ragone, Tommaso Di Noia, Francesco M. Donini, Eugenio Di Sciascio, and Michael P. Wellman. Computing utility from weighted description logic preference formulas. In Matteo Baldoni, Jamal Bentahar, M. Birna van Riemsdijk, and John Lloyd, editors, *DALT*, volume 5948 of *Lecture Notes in Computer Science*, pages 158–173. Springer, 2009.

[16] Azzurra Ragone, Tommaso Di Noia, Francesco M. Donini, Eugenio Di Sciascio, and Michael P. Wellman. Weighted description logics preference formulas for multiattribute negotiation. In Lluis Godo and Andrea Pugliese, editors, *Scalable Uncertainty Management, Third International Conference, SUM 2009, Washington, DC, USA, September 28-30, 2009. Proceedings*, volume 5785 of *Lecture Notes in Computer Science*, pages 193–205. Springer, 2009.

[17] Azzurra Ragone, Tommaso Di Noia, Eugenio Di Sciascio, and Francesco M. Donini. Description logics for multi-issue bilateral negotiation with incomplete information. In *AAAI*, pages 477–482. AAAI Press, 2007.

[18] Azzurra Ragone, Tommaso Di Noia, Eugenio Di Sciascio, and Francesco M. Donini. DL-based alternating-offers protocol for automated multi-issue bilateral negotiation. In Diego Calvanese, Enrico Franconi, Volker Haarslev, Domenico Lembo, Boris Motik, Anni-Yasmin Turhan, and Sergio Tessaris, editors, *Proceedings of the 2007 International Workshop on Description Logics (DL2007), Brixen-Bressanone, near Bozen-Bolzano, Italy, 8-10 June, 2007*, volume 250 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.

[19] Yoav Shoham. Conditional utility, utility independence, and utility networks. *CoRR*, abs/1302.1568, 2013.

[20] Umberto Straccia. Multi criteria decision making in fuzzy description logics: A first step. In Juan D. Velásquez, Sebastián A. Ríos, Robert J. Howlett, and Lakhmi C. Jain, editors, *KES (1)*, volume 5711 of *Lecture Notes in Computer Science*, pages 78–86. Springer, 2009.

[21] Dongmo Zhang and Yan Zhang. A computational model of logic-based negotiation. In *AAAI*, pages 728–733. AAAI Press, 2006.