A Description Logic Model for Reacting in a Dynamic Environment

Martina Kullmann* François de Bertrand de Beuvron
François Rousselot
LIIA, ENSAIS, 24, bd de la Victoire, F-67084 Strasbourg Cedex,
Email: martina@liia.u-strasbg.fr

Abstract

The dynamics of the world can be incorporated into a description logic model by continuously adapting it to the occurring changes. A world description formed by events occurring and actions to be taken with respect to them, allows for retrieving actions. When action retrieval is continuously repeated based on a permanently updated model, it corresponds to the construction of a plan for reacting in a dynamic environment. Using temporal description logics reasoning about plans and object states is possible. As an example application, disaster response is mentioned.

1 Introduction

The world, i.e. the state of the objects in the world, is permanently changing. Thus, in several applications of knowledge-based systems the underlying world model has to incorporate some kind of dynamics. A simple way of realizing this is to use a *static* model, which is permanently adapted to occurring changes in the world. The advantage of this approach is that also changes can be taken into account, which are not necessarily known when building the model, but which occur only during its application. Another possibility for incorporating the dynamics of the world is to directly construct a *dynamic* model, i.e. a model incorporating the notion of time.

Using description logics for knowledge representation, modifying the world model is rather simple. Besides, temporal extensions of description logics provide means for representing dynamic knowledge. Depending on the expressiveness of

*promoted by the fellowship program of the German Federal Environmental Foundation, Osnabrück, Germany

the language, sound, complete and terminating algorithms for reasoning such as satisfiability, subsumption and consistency checking are available. A continuous adaptation of such a model allows for reasoning with respect to the current circumstances at all time. Consider a world description by means of certain events occurring and actions to be taken with respect to these events. Then reasoning can be used for action retrieval, and based on a permanently updated model a sort of *reactions* to the changes in the world can be obtained. By defining temporal relations between these reactions a whole reaction *plan* can be composed. Being able to react in a dynamic environment is crucial for different kinds of applications. As an example, a description logic model of a disaster response plan is sketched. The model allows for processing disaster situation descriptions to retrieve corresponding reaction plans.

The structure of this paper is as follows. First it is described, how a static world model can be used for reacting in a dynamic environment. Then, the model is refined to directly incorporate the notion of time by means of temporal description logics. Afterwards, a description logic model of a disaster response plan is given. It is explained, how such a model can be adapted to allow for the retrieval of reaction plans in a disaster situation. The paper concludes by pointing out remaining problems and future work.

2 Adapting a Static World Model

As already stated, one possibility to model the permanent changes in the world is to use a static world model, and to continuously adapt it to the new circumstances, i.e. new events occurring, for instance. Such a model reflects the current state of the world at any time. Using description logics for the modeling, modifications like adding or deleting facts can be realized without problems. Only, in the case of deletion, facts which are in some way related to the deleted ones, have to be reviewed. Since there exist facts about the world, which do not change over time, not the whole world description has to be modified. Thus, the model can be divided into two parts, a static part and a second one which is intended to be changed, and thus is "dynamic" in a sense. The static part contains the knowledge which is always valid, like knowledge about names of objects or events which might occur, for instance. The knowledge in the dynamic part is knowledge which changes over time, and which therefore is subject to modification. For instance, new events occurring in the world have to be modeled to compose the current situation's model. Since, necessarily, there are relations between objects in the static and objects in the dynamic model part, the described separation is not strict. It only aims at structuring the knowledge base, in order to indicate to the user, which parts of the knowledge can be modified.

The basic concept definitions of a description logic model as sketched above

```
(define-primitive-concept activity)
(define-primitive-concept event)
(define-string-concept name TOP)
(define-concept situation (and (some label name)
  (some happens event) (all happens event)))
(implies activity (and (some required_by event) (all required_by event)))
(implies event (and (some requires activity) (all requires activity)
  (some happens_in situation)))
```

Figure 1: Basic Concept Definitions in the Terminology

are shown in Figure 1. The model is constructed mainly based on three concepts, which are the situation, the event, and the activity concept. The situation concept is characterized by possibly occurring events, events are modeled to require the execution of certain activities. The more complex concept terms in the model are constructed by means of the relations `happens` and `requires`, as well as their corresponding inverses `happens_in` and `required_by`. The expressiveness of the logic used is $\mathcal{ALC}\mathrm{f}\mathcal{I}$ with strings, which is a subset of the logic $\mathcal{ALC}\mathrm{f}\mathcal{IHR}_+$ with strings given in [4]. The decidability of the logic $\mathcal{ALC}\mathrm{f}\mathcal{I}$ has been shown in [5].

By means the given terminology a description logic world model can be constructed, i.e. event and activity individuals can be defined. Applying description logic reasoning methods, the retrieval of situation-dependent activities can be done. As illustrated in Figure 1, the situation concept is associated with a string label. This label allows the definition of a special situation concept, which is instantiated only by the current situation individual, if this is the only individual associated with the same label. Therefore, referring to such a special situation concept is equivalent to referring to the current situation individual. Suppose, the current situation individual is associated with the label `now`. Then, the following concept description can be constructed, the retrieval of which yields the activities depending on the current situation (see also [6]):

```
(some required_by (some happens_in (and situation (some label (aset now)))))
```

Permanently modifying the world model sketched above makes it a model of the dynamic world. A continuously adapted model allows for reasoning always with respect to the current circumstances. So, when reasoning consists in retrieving actions, the retrieval result can be viewed at as *reactions* to the current state of the world. Repeating the retrieval continuously, corresponds to the construction of a kind of reaction plan. Since such a plan evolves together with the world, it meets the demands in the world at any time. It has to be mentioned, that in this case the term *plan* denotes a set of actions only. Temporal relations between actions are imposed by the actions' retrieval time, and thus are only implicit. Such a plan necessarily is consistent, if all its basic actions are so, since they do not have any prerequisites and consequences, and thus are completely independent from one another. The consistency of a world description containing the loose actions can be checked by means of the description logic consistency test. Even if the considered reaction plan is constructed based on simple actions only, its overall structure can be quite complex. Every action retrieval yields actions to be launched in parallel, and can thus be considered as the beginning of a new step in the plan. However, actions can take more time then there is between two retrievals, and thus may last over multiple steps.

Furthermore, updates of the current circumstances allow for also refining reaction measures. Suppose, that the action to extinct a fire has been retrieved, and additionally it is found out, that there is magnesium burning. By indicating this fact, the extinct action is refined to be of the type `extinct_by_sand`. So, the concepts an action belongs to can be specialized by indicating new circumstances under which the action is to be taken.

3 Building a Dynamic World Model

Having demonstrated how the dynamics of the world can be reflected by means of a static model, in the following the model is refined to directly incorporate the notion of time by using a *temporal* description logic. Objects are now associated with time intervals, and temporal relations between these intervals can be expressed. Since objects can be modeled to have states which differ with respect to time, means are provided to implicitly indicate prerequisites and consequences of actions. By relating actions temporally to one another, complex plans can explicitly be represented. So, the term *plan* here denotes a set of actions which are temporally related to one another and which may have prerequisites and consequences. Therefore, the use of temporal description logics allows for a detailed explicit description of plans.

Based on such a dynamic model which allows for the representation of object states and plans, temporal reasoning like satisfiability, subsumption and consistency checking with respect to the temporal knowledge can be applied. Thus, the consistency of plans can be verified, while taking into account the temporal dependencies between actions and the states of their objects. The two most recent approaches for temporal description logics are the decidable logics $\mathcal{TL}$-$\mathcal{ALCF}$ [1] and $\mathcal{ALCF}(A)$ [2], which is an extension of $\mathcal{ALC}(\mathcal{D})$ as introduced in [3]. The logic $\mathcal{TL}$-$\mathcal{ALCF}$ has the advantage that it allows for the hierarchical decomposition of plans. However, for this logic so far not all of the above mentioned reasoning tasks can be realized by means of a tableau-calculus. Since here the tableau-based description logic system CICLOP [4] is used, a logic like

$\mathcal{ALCF}(A)$ with its tableau-based algorithms can easier be integrated and therefore is more appropriate. Besides, it provides sound, complete and terminating algorithms for all of the above mentioned reasoning tasks. However, for decidability reasons the temporal model part has to be defined and treated separately from the static one so far.

Additionally to the application of description logic reasoning methods, the dynamic model allows for the same adaptation and reaction retrieval procedure as described in Section 2. Since now plan representation is possible, action retrieval can also yield more complex reaction plans, i.e. the name of a complex plan can be retrieved. By introducing temporal relations between the retrieved plans, an explicit representation of the final overall plan can be constructed.

However, plan consistency checking so far has one limitation. It is, that inconsistencies can only be detected either on the conceptual or on the concrete level, and therefore inconsistencies caused by an interaction of these levels cannot be found. In the logic $\mathcal{ALCF}(A)$, a temporal object is represented by means of multiple objects, each of which describes the object at a different time interval, and which are combined by relating them via attributes to a kind of meta-object. Such a representation allows for detecting inconsistencies on the conceptual level as usual. Besides, inconsistencies in the concrete temporal domain can be detected, i.e. if intervals do not respect the constraints imposed on them. However, there is no means to unify objects corresponding to overlapping intervals, to have for each interval in time only *one* object describing the meta-object. This would be interesting in case that a *state* of an object is to be described. Consider the definition of the following concept `test`:

```
(define-concept test (and (some step1 state) (some step2 (not state))
    (some (compose step1 time) (compose step2 time) during)))
```

This concept denotes objects which are associated with two "time steps", the fillers of which are in the extension of the concept `state` and `(not state)`, respectively. Besides, the time intervals corresponding to the steps are related by the temporal relation `during`. Since the step fillers are intended to represent the same `test` meta-object at their corresponding intervals, it should be possible to undertake a kind of unification of these fillers, in the case that their intervals overlap as in the example, and the corresponding concept descriptions denote states. Then, a concept like `test` would not be consistent, i.e. objects corresponding to the same meta-object could not be in the extension of contradictory state concepts at overlapping time intervals.

Since in a temporal description logic context actions and plans can have prerequisites and consequences, inconsistencies of the type as described above can occur in the overall plan constructed during the adaptation-retrieval-procedure. Since such a plan is combined out of several actions and sub-plans, these may involve and thus change the states of the same objects. These objects are represented by a meta-object and several sub-objects, the latter of which might be defined with respect to different actions and at overlapping time intervals.

4 An Example Application: Disaster Response

In this section, an example application of using description logics for reacting in a dynamic environment is given, which is disaster response. For an appropriate disaster response a detailed planning of emergency reactions is essential, as well as their efficient management in an emergency case. By means of a description logic knowledge base containing definitions of disaster situations and reaction plans, a description logic system can support the decision making in a specified disaster case. Since during the course of a disaster the situation changes permanently, also permanently new reaction measures have to be selected. Here, a model of high water situations and response plans in the city of Heidelberg on the Neckar in Germany is considered. First, a static model is given, which is then refined by means of temporal description logics.

4.1 Static Disaster Response Plan Model

For the city of Heidelberg the so-called high water protection plan specifies how to react in the case of high water. The schema of this disaster plan is quite simple (see [7]). An emergency situation is characterized only by its corresponding water level. Each situation is associated with events which are expected to occur. An event requires the execution of a certain number of activities. In the case of disaster, the actions to be executed are chosen with respect to the highest water level expected to occur, and their execution is coordinated by a high water board.

The description logic model sketched in the following is discussed comprehensively in [6]. Based on the concepts given in Figure 1 the terminology shown in Figure 2 can be constructed, i.e. different kinds of more special situation, event, and activity concepts. The high water situation concept is characterized by a corresponding water level as well as by events which might occur independently of the water level. Different situation concepts can be constructed by relating them to different water levels and events. Figure 3 shows an extract of the disaster plan, which is modeled based on the terminology sketched before. Such a representation now allows for applying reasoning methods, and can thus be used to propose and manage the reactions in a high water situation. First, a description of the current situation can be constructed by means of the water level and a library of events, and the consistency of the model can be checked. Then, the realization inference allows for assessing the specified current situation, and besides the activities to be executed can be retrieved. So, the model supports

```
(define-concept blocking (and activity (some object barrier)))
(define-concept avoiding (and activity (some object street)))
(define-concept install_barrier (and activity (some object barrier)))
(define-concept install_wall (and install_barrier (some object wall))))
(define-primitive-concept street_flooded event)
(define-primitive-concept traffic_jam event)
(define-concept high_water_situation (and situation
  (some indicator water_level) (all indicator water_level)))
(define-concept emergency_situation_with_traffic_jam
  (and high_water_situation (some happens traffic_jam)))
(implies event (some happens_in water_level)))
(implies street_flooded (some requires blocking))
(implies traffic_jam (some requires avoiding))
(implies rail_precaution install_wall)
(implies water_level (and (some happens event) (all happens event)
  (some indicates situation) (all indicates situation)))
```

Figure 2: Concept Definitions in the High Water Protection Plan Model

the process of deciding what to do in this situation. In the case considered here, the concept description for activity retrieval as proposed in Section 2 has to be slightly modified, since also water level dependent events may occur.

However, for an appropriate disaster response it is not sufficient to construct an adequate situation description only once and then to execute the corresponding reaction measures. During the course of a disaster the situation changes permanently and its particular circumstances are not known in advance. To allow for retrieving appropriate reactions to the changes in the world, the situation description and disaster plan have to reflect these changes. Updating the model can be done manually. Thanks to the water level forecast, time is usually not pressing when retrieving actions to be executed. In the example, corresponding to the current situation the activities `Avoid_Hauptstrasse` and `Block_Neckarweg` can be retrieved. Suppose that some time later the water is 3.55 m high. Then the statement `(related Current_situation Water_level_2.90 indicator)` is replaced by the one for a water level of 3.55 m. Also other events might be defined, to happen in the updated current situation. Repeating the activity retrieval, it now yields reactions to the new situation model, i.e. instead of `Block_Neckarweg` the action `Install_barrier_Neckarweg` is obtained. The activities, which are retrieved over time, implicitly form a consistent reaction plan.

Furthermore, updates of the current circumstances allow for refining reaction measures. For instance, consider the action `Install_barrier_Neckarweg`. Suppose, that the circumstances concerning this action have changed, i.e. that at Neckarweg there has been installed a rail to fix a mobile wall. Then, the asser-

```
;; static world description part
(instance Neckarweg_flooded street_flooded)
(instance Block_Neckarweg blocking)
(related Neckarweg_flooded Block_Neckarweg requires)
(instance Water_level_2.90 water_level)
(related Water_level_2.90 Neckarweg_flooded happens)
(instance B37_Alte_Bruecke_flooded street_flooded)
(instance Install_barrier_Neckarweg install_barrier)
(related B37_Alte_Bruecke_flooded Install_barrier_Neckarweg requires)
(instance Water_level_3.55 water_level)
(related Water_level_3.55 B37_Alte_Bruecke_flooded happens)
(instance Avoid_Hauptstrasse avoiding)
(instance Traffic_jam_Hauptstrasse traffic_jam)
(related Traffic_jam_Hauptstrasse Avoid_Hauptstrasse requires)
(instance Current_situation (and situation (some label (aset now))))
;; dynamic world description part
(related Current_situation Water_level_2.90 indicator)
(related Current_situation Traffic_jam_Hauptstrasse happens)
```

Figure 3: Model of the High Water Protection Plan

tion `(instance Install_barrier_Neckarweg rail_precaution)` can be made, which implicitly makes `Install_barrier_Neckarweg` an `install_wall` action. So, indicating new circumstances under which the action is to be taken can have an effect on the concept membership of the action, i.e. the most specific concepts an action belongs to can be specialized.

4.2 Dynamic Model Extension

Since for a detailed modeling of disaster reactions representing and reasoning about plans is needed, the basic model of the high water protection plan is extended by means of the temporal description logic $\mathcal{ALCF}(A)$. The definition of the action `install_wall` for instance, can be refined to be a plan, which is composed of two actions. First, some material has to be brought, then the wall can be installed. The two actions are temporally related by the relation before.

```
(define-concept install_wall (and (some step1 bring_material)
    (some step2 install_high_water_wall)
    (some (compose step1 time) (compose step2 time) before)))
```

Besides plan representation, temporal description logics allow for representing object states depending on time. This is needed, since actions or plans change the state of their objects, like the action of blocking a street, for instance:

```
(define-concept blocking (and (some step1 (some street non_blocked))
    (some step2 (some street blocked))
    (same-as (compose step1 street) (compose step2 street))
    (some (compose step1 time) (time) meets)
    (some (time) (compose step2 time) meets)))
```

Here, the street is `non_blocked` before the execution of the action and it is `blocked` afterwards. So, by means of object state definitions, action prerequisites and consequences can implicitly be described.

Incorporating plans and object states in the example high water plan, the retrieval of refined plans describing how to react in a disaster situation is possible. Besides, such a representation allows for reasoning taking also the temporal knowledge into account. So, satisfiability and subsumption checking for plan concepts is possible, as well as plan classification. Consistency checking can be done for world descriptions with plan individuals, as well as plan recognition. By relating intervals temporally to one another, it is now possible to construct an explicit representation of the final overall plan. In the example, first the action `Block_Neckarweg` is retrieved and afterwards the action `Install_barrier_Neckarweg`. Suppose, `Block_Neckarweg` is associated with the interval i_1 and `Install_barrier_Neckarweg` with the interval i_2. The temporal relation between the two actions can then be expressed by stating after(i_2,i_1). If `Install_barrier_Neckarweg` had the prerequisite, that just before its execution the street Neckarweg had to be `non_blocked`, this would contradict with a former action's consequence without being detected.

5 Conclusion and Future Work

In this paper, it is illustrated, how a static description logic model which is permanently modified to reflect the relevant changes in the world can be used for retrieving reactions to a dynamic environment. Refining the model by means of temporal description logics allows for a detailed description of plans and object states. The plans retrieved over time form a kind of overall reaction plan, which can be explicitly represented. Being able to react in a dynamic environment is of interest for different kinds of applications. Here, a model of a disaster response plan is sketched. Another important application area are robotics, for instance. To validate the presented approach, it is planned to extend the description logic reasoner CICLOP [4] to allow for representing and reasoning about temporal knowledge.

The proposed reaction model is a rather simple one. It is still necessary to think about, how reaction plans proposed at different time steps might interact. The proposal of a plan might need for stopping or modifying another one, which is already being executed. Besides, the presented model does not automatically incorporate changes, which are caused by an action with consequences. Also, the feasibility of an action with prerequisites is not checked yet. Besides, if an action's consequence holds already this action does not need to be executed anymore. All these questions have already been addressed in several approaches for planning with description logics, which will be studied in the future to refine the presented model.

Another problem is the question, at what times to retrieve new plans. In the high water response example, some significant water level heights are known, which imply significantly new reactions. So, the application domain can give some hints about when a new retrieval is of need.

References

[1] A. Artale and E. Franconi. A temporal description logic for reasoning about actions and plans. *Journal of Artificial Intelligence Research*, 9:463–506, 1998.

[2] A. Artale and C. Lutz. A correspondence between temporal description logics. In *Proc. of the International Workshop on Description Logics '99*, Linköping, Sweden, 1999.

[3] F. Baader and P. Hanschke. A scheme for integrating concrete domains into concept languages. In *Proc. of the* 12^{th} *International Joint Conference on Artificial Intelligence*, pages 452–457, Sidney, Australia, 1991.

[4] F. de Bertrand de Beuvron, F. Rousselot, M. Grathwohl, D. Rudloff, and M. Schlick. Ciclop. In *Proc. of the International Workshop on Description Logics '99, System Comparison*, Linköping, Sweden, 1999.

[5] G. de Giacomo. *Decidability of Class-Based Knowledge Representation Formalisms.* PhD thesis, Università di Roma "La Sapienza", Italy, 1995.

[6] M. Grathwohl, F. de Bertrand de Beuvron, and F. Rousselot. A new application for description logics: Disaster management. In *Proc. of the International Workshop on Description Logics '99*, Linköping, Sweden, 1999.

[7] Stadt Heidelberg, Germany. *Hochwasserschutzplan der Stadt Heidelberg – Stand: März 1998*, 1998.