A New Application for Description Logics: Disaster Management

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

Abstract

By presenting a description logic model of the high water protection plan of Heidelberg we demonstrate that description logics are an appropriate tool for supporting the management of natural disasters. The model enables testing its consistency, assessing disaster situations by means of realization, and retrieving situation-dependent activities. Finally, we discuss the remaining problems on which we will work in the future.

1 Introduction

The field of disaster management comprises disaster mitigation, disaster relief and rehabilitation ([2]). Disaster mitigation aims at reducing the disaster risk or a disaster's severeness by preparing for appropriate response. Disaster relief denotes the immediate humanitarian assistance as well as subsequent long-term relief assistance. Rehabilitation finally comprises long-term measures for reestablishing economical and social structures.

In our paper we contribute to disaster mitigation and disaster relief by proposing methods for supporting the immediate emergency aid in the case of disaster. Disaster response aims at prohibiting or at least limiting the damages caused by disasters. It requires a detailed planning of actions to be performed in the case of disaster. Furthermore, to enable an efficient execution of disaster plans the management of reactions has to be supported by a computer system.

So far, computer systems for supporting disaster response are mainly based on databases ([6]). Disaster relevant knowledge is stored in a database and can be queried in a database-typical way. These systems provide a library of predefined disaster situations among which a concrete situation has to be chosen. With respect to the chosen situation an appropriate disaster plan is provided.

However, it is obvious that it is not sufficient being able to handle only predefined situations. Disasters are that difficult to respond because they entail situations which can not be foreseen. Therefore, our aim is to develop a system which is able to propose reactions to situations specified just during the course of disaster. We propose to use description logics for the representation of the relevant knowledge. Thus, the powerful description logic inferences can be used to query the data.

In our work we choose as an example the high water protection plan of the city of Heidelberg on the Neckar in Germany ([4]). We give a simple description logic model of the disaster plan and propose an extension using more expressive description logics. The final model enables exploiting the description logic inferences for testing the model consistency, for assessing a given concrete disaster situation, and for selecting measures to be executed with respect to this situation. For validating our model we use the description logic-based system CICLOP ([1]).

2 The High Water Protection Plan of Heidelberg

The schema which is currently used in the city of Heidelberg for the planning of reactions in the case of high water is quite simple. 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 measures which consist of several activities. A measure is furthermore associated with a duration of execution, a responsible person and a competent office. The activities forming the measures can basically be described by means of an object and a location of execution. Sometimes some more characteristics have to be added. The structure of the high water protection plan of the city of Heidelberg is illustrated in Figure 1, which shows extracts of the planning for the situation with a water level of 2.90 m.

In Heidelberg, the office which is competent for orga-

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

```
Water level 2.90 m
Event 01:             high water danger
  Measure 01.01:      information
    Activities:       inform Mr. Schmidt
                      inform Mr. Meier
                      inform building sites
                      ...
    Competent:        office of civil engineering
    Responsible:      Mr. Schulze
    Duration:         0.80 hours
  ...
Event 02:             underpass Neckarweg flooded
  Measure 02.01:      blocking
    Activities:       block underpass Neckarweg
                      ...
    Competent:        office of civil engineering
    Responsible:      Mr. Schulze
    Duration:         2.50 hours
  ...
...
```

Figure 1: Extract from the High Water Protection Plan of Heidelberg.

nizing measures in a high water situation presently uses a database system to manage the disaster plan described above. In the case of disaster the water level forecast is considered to calculate the highest water level expected to occur and when it is expected to occur. With respect to this information the planned actions are chosen. Based on their duration of execution they are provided with time limits when they have to be started the latest. The execution of the disaster plan updated this way is coordinated by a high water board.

Note, that since an emergency situation is characterized only by its water level, the corresponding reactions only depend on the water level as well. Furthermore, the term disaster *plan* is not really appropriate here, because the so-called plan is just a set of actions which are associated with a duration of execution. There are no temporal relationships between the actions. For simplicity however, we will still use the term *plan* in our case.

3 Description Logic Model

The simple structure of the described disaster plan suggests the use of description logics for its modelling. The aim is to exploit description logic inferences to provide an efficient disaster management.

We will first present a simple description logic model for which we can test the model consistency and which provides the assessment of disaster situations by the use of realization. Afterwards, we extend the model using inverse roles and concrete domains to be able to provide the retrieval of situation-dependant activities.

```
(define-primitive-role happens)
(define-primitive-role requires)
(define-primitive-attribute water_level)
(define-primitive-attribute object)
(define-primitive-attribute flooded)
(define-primitive-concept activity)
(define-concept blocking
  (and activity (some object barrier)))
(define-concept information
  (and activity (some object person)))
(define-concept situation
  (and (some happens event) (all happens event)))
(define-concept high_water_situation (and situation
  (some happens high_water_level)
  (some happens high_water_danger)
  (some happens street_flooded)))
(define-concept event (all requires activity))
(define-concept high_water_level
  (and event (some water_level height)))
(define-concept high_water_danger
  (and event (some requires information)))
(define-concept street_flooded (and event
  (some flooded street)
  (some requires blocking)))
```

Figure 2: Terminology of the Simple Model.

3.1 Simple Model

One advantage of description logic based knowledge representation is the taxonomical organisation of the represented data. In our case, it is natural to construct a situation taxonomy for being able to realize, i.e. to assess a given disaster situation. This is especially important if we do not want to rely only on predefined situations.

We propose a terminology where situations are not only characterized by their corresponding water level which implies certain events. In our model, events can arbitrarily be grouped to form a situation. For simplicity, we neglect the measure characteristics `duration of execution`, `responsible person` and `competent office`. We relate activities directly to events and do not regroup them to measures.

Using the description logic with expressiveness $\mathcal{ALCF}$ as given in [1] we construct an example terminology of the simple model described above, which is shown in Figure 2. We define activity, situation and event concepts. The concept `high_water_situation` is defined as a situation which is characterized by the three events `high_water_level`, `high_water_danger`, and `street_flooded` occurring in it.

By means of the classification inference, a situation taxonomy can now be constructed. The concept `high_water_situation` is classified to be a subconcept of the concept `situation`.

Having defined the terminology, we are able to model the disaster plan given in Figure 1. We do this by defining activity, situation, and event individuals. However, we have the advantage, being able to use these predefined event and situation individuals to compose a new situation individual, which represents the current situation. We give an example in Figure 3. The individual situation **current_situation** is composed by relating it to the three event individuals **Neckarweg_flooded**, **water_level_2.90**, and **danger_2.90**.

```
(instance current_situation situation)
(instance Neckarweg_flooded street_flooded)
(instance block_Neckarweg blocking)
(related Neckarweg_flooded
    block_Neckarweg requires)
(instance water_level_2.90 high_water_level)
(instance danger_2.90 high_water_danger)
(related current_situation
    water_level_2.90 happens)
(related current_situation danger_2.90 happens)
(related current_situation
    Neckarweg_flooded happens)
```

Figure 3: Simple World Model.

The description logic based model of the considered disaster plan allows the use of corresponding inferences. One of them is the consistency test. This test allows detecting logical errors in the data model. Besides, the situation taxonomy allows the use of the realization inference. Having described a current emergency situation by means of the predefined events and situations, the realization inference can be used to find the corresponding situation concept. This way, the current situation can be assessed. In our example the concept **high_water_situation** corresponds to the individual **current_situation**.

In comparison with the disaster plan, our model provides a bigger liberty in describing situations. Furthermore, we are able to assess a given situation. The practice shows that a very small terminology of about 50 concepts is fully sufficient to model our data. However, we have to deal with about 1000 individuals.

Next, it would be interesting to use inferences to obtain all the activities to be executed in a given situation. To achieve this, we extend the simple model using a higher description logic expressiveness. We will be able to retrieve all the activities corresponding to the events forming a given situation. But there might be activities to be executed only with respect to a situations type. For being able to retrieve these activities also, which are not directly required by the events occurring in the situation, the situation assessment described in this section is particularly of interest.

3.2 Model with Inverse Roles and Concrete Domain

A description logic-based data representation allows querying the data via relations between concepts or individuals, respectively. We are making use of this property to find the activities to be executed in a given disaster situation.

Considering our simple model, a situation concept is related to event concepts which themselves are related to a corresponding activity concept. Therefore, the activities corresponding to a given situation can be accessed by following the roles from the situation concept via the related event concepts to the associated activity concepts. However, for being able to elegantly define a concept containing all the activities in relation with a given situation we need a higher description logic expressiveness, i.e. we need inverse roles. Not using inverse roles would have meant a much less elegant way of representing and reasoning about the data.

The extension of our simple terminology using inverse roles can be described as follows. The concepts **situation** and **event** are now related inversely to one another, as well as the concepts **event** and **activity**. In our example given in Figure 4 we associate only *one* event to the high water situation concept, for simplicity. Using the introduced inverse roles **happens_in** and **required_by** we are able to define the concept **to_do_in_situation**. It is equivalent to the activity concept, but **activity** is not referred to in its definition. The definition is based on the situation concept only.

However, in a given disaster situation it is necessary to refer to a situation *individual* and then to access all the activity individuals related to it. To solve this problem we associate a label to the situation concept. This allows to use a specific label for the definition of a more specific situation concept which is intended to contain only one individual being the current situation in our case. So, referring to the current situation by referring to the labeled concept is possible.

We could label concepts by using the description logic language construct `set` ([5]), i.e. by using individuals for labelling concepts. Another possibility is to use a concrete domain, i.e. strings. Since we do not want to mix up terminology and assertions we continue in the second direction. Strings allow to refer to individuals on the conceptual level. So, we allow concepts to have a description which consists of a set of strings, more precisely of a set of possible or impossible ones (see [1]).

We start the extension of our model by defining a concept **name** which is described by a set of strings. The situation concept is redefined to have a label of type **name**, i.e. a string label. Now we can define the special situation concept `(and situation (some label (aset now)))` which is labeled by the string `now`. This concept

```
(define-string-concept name TOP)
(define-primitive-attribute label)
(define-primitive-role happens_in)
(role-implies happens_in (inverse happens))
(role-implies happens (inverse happens_in))
(define-primitive-role required_by)
(role-implies required_by (inverse requires))
(role-implies requires (inverse required_by))
(define-primitive-concept activity
  (some required_by event))
(define-concept blocking (and activity
  (some object barrier)
  (some required_by street_flooded)))
(define-concept situation (and (some label name)
  (some happens event) (all happens event)))
(define-concept high_water_situation (and situation
  (some happens street_flooded)))
(define-concept event (and
  (some requires activity)
  (all requires activity)
  (some happens_in situation)
  (all happens_in situation)))
(define-concept street_flooded (and event
  (some flooded street)
  (some requires blocking)
  (some happens_in high_water_situation)))
(define-concept to_do_in_situation
  (some required_by (some happens_in situation)))
(define-concept to_do_now
  (some required_by (some happens_in
  (and situation (some label (aset now))))))
```

Figure 4: Terminology of the Model with Inverse Roles and Concrete Domain.

is intended to contain only the current situation. By labeling the current situation individual with the same name we can refer to it using the special situation concept. If the current situation individual is the only one with this label then it is the only one which is refered to via the special situation concept. To retrieve all the activities to be executed in the current situation we can now use a concept like `to_do_in_situation` which refers to the special situation concept. In our example terminology in Figure 4 we call it `to_do_now`.

The corresponding world model with the modified definition of the current situation individual is shown in Figure 5. We define the event `Neckarweg_flooded` to occurr in the current situation requiring the activity `block_Neckarweg`. Besides, we define another activity `inform_MrSchmidt` not being related to the current situation. Retrieving the concept `to_do_now` only yields the activity `block_Neckarweg` in this case.

Note, that it is not necessary to explicitly define the concept `to_do_now`. It would even be better to separate the conceptual from the individual level as much as possible. In this case, the retrieval inference could be called

```
(instance current_situation
    (and situation (some label (aset now))))
(instance Neckarweg_flooded street_flooded)
(instance block_Neckarweg blocking)
(related Neckarweg_flooded
    block_Neckarweg requires)
(instance inform_MrSchmidt information)
(related current_situation
    Neckarweg_flooded happens)
```

Figure 5: World Model with Inverse Roles and Concrete Domain.

to retrieve the *description* of the concept `to_do_now`.

The extended model discussed in this section offers the realization of a non-predefined disaster situation as well as the retrieval of all the activities which have to be executed in it. The model is based on the description logic expressiveness $\mathcal{ALCFI}$ with strings which is a subset of the logic $\mathcal{ALCFIHR}_+$ with strings given in [1]. The decidability of the logic $\mathcal{ALCFI}$ has been shown in [3].

4 Conclusion

By presenting a description logic model of the high water protection plan of Heidelberg we demonstrated that description logics are an appropriate tool for representing and reasoning about disaster data for supporting the management of natural disasters. We started with a simple model which exploits the description logic inference to test the model consistency, which uses classification for constructing a situation taxonomy, and which provides the realization of situations. Then, we proposed an extended model providing the retrieval of situation-dependent activities, i.e. making use of the retrieval inference.

In a disaster situation, the proposed disaster data model allows the following. First, the current situation can be described by means of an event library and predefined disaster situations. So, we do not only have to rely on predefined situations. Then, the realization inference allows to map the specified current situation to its corresponding situation concept in the situation taxonomy. This way, a given situation can be assessed. Furthermore, using inverse roles and strings to build the model allows the retrieval of all the activities to be executed in the given disaster situation. By proposing these activities, the description logic system can support the process of deciding what to do in a disaster situation.

We still have to investigate whether description logic inferences are appropriate to treat realistic knowledge bases. Hopefully, the high complexity of the corresponding algorithms will not be a problem, because the consistency test, the classification and the realization of predefined individuals can already be executed before using the system for disaster management. The inferences used in the disaster situation, which have to be fast, are the realization of the current situation and the retrieval of the activities to be executed.

The presented methods can be applied to other types of disasters than high water, as well. However, the data model has to be redeveloped for each special disaster environment, for it depends on the local circumstances and the disaster type. To conclude, we will enumerate some points which should also be taken into account by a description logic model for disaster management.

5 Future Work

Even if we were able to demonstrate that description logics are an appropriate tool for disaster management, the presented representation of the Heidelberg disaster plan shows some problems which require further consideration. In the following we will state some aspects of disaster management which have not been taken into account by the presented model and on which we will be working in the future.

Our demonstration of the usefulness of description logics in disaster management was based on a very simple disaster plan. As we already pointed out, our example *plan* does not contain any temporal knowledge besides the duration of measures composed by certain activities. Starting from existing approaches for representing and reasoning about temporal knowledge with description logics, our major task will be to investigate how the notion of time and real plans could be incorporated in our model. The need of extending our model in this direction is obvious, if we want to provide methods for a more detailed disaster management. Besides, it would be desirerable being able to graphically visualize a disaster plan.

Another aspect which is neglected in the high water protection plan of Heidelberg is the management of resources. Each activity to be executed needs some resources, at least its actor. Managing disasters, we have to answer questions about the number of resources available, about the kind of resources needed, and where and how resources can be distributed to satisfy all demands. To answer these questions we have to deal with priorities of activities and strategies to solve conflicts, as well.

Furthermore, it should be possible to take into account new circumstances occurring in the current case of disaster. It can happen that activities can not be executed, or that the successful execution of an activity changes the situation in different directions. As the disaster situation changes permanently, it seems to be neccessary to permanently input the current situation in the management system. So, the system would be able to react to the situation's changes by proposing appropriate activities.

Besides, it is desirable to take into account background knowledge which could be useful in the case of disaster. This is, for example, geographical knowledge, and knowledge gained by experience. With respect to a water level forecast the constraints on the execution time for activities should be calculated and incorporated in the disaster plan proposed by the system. Considering disasters in regions which require international coordination it should be possible to integrate multiple disaster plans.

In the future, we will investigate how description logics might contribute or might be extended to contribute to the solution of these open problems. The approach for managing disasters with description logics proposed in this paper will serve as a basis.

References

[1] F. de Bertrand de Beuvron, F. Rousselot, M. Grathwohl, D. Rudloff, and M. Schlick. "CICLOP." System Comparison of the International Workshop on Description Logics '99, Linköping, Sweden, 1999.

[2] C. Eikenberg. "Journalist's Manual on Disaster Management -1998-." German IDNDR-Committee for Natural Disaster Reduction, Bonn, Germany, 1998.

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

[4] "Hochwasserschutzplan der Stadt Heidelberg - Stand: März 1998." Stadt Heidelberg, Germany, 1998.

[5] P.F. Patel-Schneider, B. Swartout. "Description-Logic Knowledge Representation System Specification from the KRSS Group of the ARPA Knowledge Sharing Effort." 1993.

[6] J. Picht. "Hochwasser-Informations und Schutz-System (HoWISS)." CreativWare, Karlsruhe, Germany, 1999.