A Disaster Management System for High Water Response Based on Description Logics

Martina Kullmann[1] and **David Rudloff** and **François de Bertrand de Beuvron**[2]

Abstract. This paper describes a disaster management system for high water response, which has an intelligent description logic system as kernel. A description logic model of a high water protection plan allows for using corresponding reasoning methods to process high water situation descriptions, and thus to propose reaction plans. So, in a disaster situation the system can support the decision process as well as the management of reactions. Other important features are, that means are provided for checking the consistency of the data model, assessing disaster situations, taking into account the situations' changes, and maintaining the model in a comfortable way. The existing prototype provides all the facilities needed for an efficient disaster management. Comparing this approach to conventional database systems, the benefits of using an intelligent system instead are pointed out.

1 INTRODUCTION

The area of disaster management deals with three main aspects, which are disaster mitigation, disaster relief and rehabilitation (see [4]). Disaster mitigation and relief notably comprise the planning of immediate emergency reactions to a disaster situation as well as its management in an emergency case. Such disaster response aims at prohibiting or at least limiting the damages caused by disasters. Since disaster plans usually are quite complex, to allow for an efficient execution their management has to be supported by a computer system. Such a system has to provide means to process a user defined disaster situation description to propose appropriate measures to be taken. In this paper, a disaster management system is presented which uses artificial intelligence methods to process disaster data.

As an example disaster scenario, high water situations in the city of Heidelberg on the Neckar in Germany are chosen. There, reaction measures are specified in the so-called high water protection plan, which is currently managed by a database system. Disaster relevant knowledge is stored in a database and can be queried in a database-typical way. The system provides a library of predefined disaster situations, among which a concrete situation can be chosen. With respect to the chosen situation an appropriate disaster plan is provided.

In this paper, it is proposed to use description logics for the representation of the relevant knowledge. It is aimed at exploiting description logic inferences for querying the data, to provide a more efficient disaster management. Therefore, a description logic model for the high water protection plan of Heidelberg is presented. Using description logic inferences the model supports the decision making in a disaster situation as well as the management of reaction plans. So far, a research prototype of a disaster management system has been developed, which provides means to process an appropriate disaster data model. Description logics have been chosen for representing the disaster data, because they combine high expressiveness with powerful inference methods. Therefore, using description logics for knowledge representation and reasoning is much more natural and elegant, than using relational or object-oriented databases.

The structure of this paper is as follows. First, the problem of high water management is illustrated by describing the high water protection plan of Heidelberg, and how it is managed by a database system. Then, a description logic model of this disaster plan is presented, and how a corresponding reasoner can be used to process it for managing high water situations. Also, it is stated what kind of advantages such a knowledge-based system has in comparison to a database system. Afterwards, the current status of the research prototype is described as well as the architecture planned for the final high water management system. To conclude the features of the system are summarized.

2 DATABASE-SUPPORTED HIGH WATER MANAGEMENT IN HEIDELBERG

To introduce to the problem of disaster management, it is described how high water situations are currently managed in the city of Heidelberg. When a high water is announced, reaction measures have to be taken to prepare for the situation. For Heidelberg these measures are specified in the so-called high water protection plan. In this section, the structure of this disaster plan is described, and how it is managed by a database system (see also [5]). Besides, several drawbacks of this system are stated.

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 (see [10]). 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 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 organizing measures in a high water situation presently uses a database system to manage the disaster plan described above [7]. 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 re-

---

[1] promoted by the fellowship program of the German Federal Environmental Foundation, Osnabrück, Germany
[2] LIIA, ENSAIS, 24, bd de la Victoire, F-67084 Strasbourg Cedex, France, email: martina@liia.u-strasbg.fr

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

spect 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.

However, the described way of reacting in high water situations has some obvious drawbacks. One is, that the actual system is only able to handle predefined situations characterized only by their water level, which is not sufficient. Disasters are that difficult to respond because they entail situations which can not be foreseen. The reactions can depend on events occurring independently from the water level. Our aim is to develop a system which is able to propose reactions to situations specified just during the course of disaster.

Besides, the disaster plan of Heidelberg has a very simple structure. As it can easily be seen, 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 relations between the actions. However, to provide methods for a more detailed disaster management, it is necessary to represent and reason about *real* plans. Therefore, a disaster response system should incorporate the notion of time.

Furthermore, it should be possible to take into account new circumstances occurring during the course of a 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, its description should permanently be adapted to the new circumstances. Reprocessing the updated description would be necessary to allow for reactions to the situation's changes. The proposed activities also should be chosen with respect to the activities which are currently being executed.

3 HIGH WATER MANAGEMENT BASED ON DESCRIPTION LOGICS

Having illustrated the problem of high water management by giving the example of Heidelberg on the Neckar, now a description logic approach is presented which tackles the sketched problem. First, knowledge representation with description logics is briefly introduced. Then, a general overview of the description logic model of the high water protection plan of Heidelberg is given. Furthermore, some model details are considered like the relations between events and activities, the water level definition, as well as the water level dependency of events. It is explained, how the presented model can be used for managing the reactions in a high water situation. Finally, the advantages of the description logic data representation in comparison to the conventional database approach are pointed out.

3.1 Description logics

The description logic knowledge representation formalism consists of two parts. The terminological part of the formalism enables defining the abstract conceptual frame through which the real world is to be seen. *Concepts* and relations can be defined. They form a *terminology*. Based on a given terminology the assertional part of the formalism can be used to describe concrete facts about the real world. Particular objects, the so-called *individuals* can be introduced to be instances of concepts. Besides, relations between individuals can be defined. These individual or relation instantiations form a *world description*.

Description logics are provided with a formal syntax as well as a formal model-theoretic semantics, which indicate how concept terms and axioms have to be defined and interpreted, respectively. The important feature of description logics is, that some of these languages are decidable, and that their knowledge representation supports several inference methods for reasoning about the represented knowledge. The formal semantics allows the formulation of sound and complete algorithms for the implementation of these inferences.

For the terminology, inferences exist to test the *satisfiability* of a concept description and to check whether a concept *subsumes* another one. The *classification* inference automatically organizes the concepts in a taxonomical order. The *consistency test* checks, whether the individuals in the world description have contradiction-free descriptions. By the *instantiation test* it is checked whether an individual belongs to a given concept. The *realization* inference computes the most specific concepts corresponding to a given individual, and the *retrieval* inference yields for a given concept the individuals which instantiate it. More details about description logics in general can be found in [2], for instance.

So far, several description logic extensions to represent and reason about *temporal* knowledge have been developed (see [1] for an overview). One of the recent approaches is the logic $\mathcal{ALCF}(A)$, which associates the language with a temporal so-called concrete domain. This logic is decidable and provided with a sound and complete tableau calculus for checking concept satisfiability and knowledge base consistency.

3.2 General model overview

The description logic model presented in the following is an extension of the one given in [5]. Its terminology is constructed mainly based on three concepts which are `situation`, `event`, and `activity`. The situation concept is characterized by a corresponding water level as well as by events which might occur independently of the water level. This way a more flexible situation description is possible than provided in the plan of Heidelberg. Also, different situation concepts can be constructed by relating them to different water levels and events. Furthermore, events are modeled to require certain activities. However, to model disaster reaction plans, it is not sufficient to be able to represent and reason about simple activities only.

To incorporate the notion of time and *real* plans in the model, temporal description logics are used. They provide means for modeling temporal relations between actions and object states.

Based on the terminology sketched above, the disaster plan itself can be modeled. Activity, event, and situation individuals can be defined and related to one another. The model contains one situation individual which represents the current situation. It is related to the individual representing the highest water level which is expected to occur in the given situation, as well as to water level independent event individuals. Besides, relations between the water level individual and water level dependent event individuals are established, and the event individuals are associated with corresponding activity individuals. By means of the temporal description logic formalism, activity individuals can be temporally related to form plan individuals. Besides, they are characterized by how their execution changes the states of their object individuals.

The described representation of the Heidelberg disaster plan allows exploiting corresponding description logic inference mechanisms. As stated above, they also take into account the temporal knowledge in the model. So, the model consistency can automatically be tested. Besides, subsumption relations between concepts can be calculated, and therefore also a situation taxonomy using classification. Then, realization allows for determining the situation concept for a given situation individual, i.e. the assessment of newly defined concrete disaster situations. Furthermore, the activities to be executed in a given situation can be queried by means of the retrieval inference. These activities can also be complex plans.

3.3 Relating events and activities

When in a high water situation a certain event is occurring, some reaction measures have to be executed with respect to this event. Therefore, in the description logic model, a relationship between the activity and the event concept has to be established. Figure 2 and 3 show an extract of the description logic model terminology, containing also the activity and the event concept definition. Based on these definitions more special activity and event concepts are defined, as e.g. the event `street_flooded` and the activity `blocking`.

To model the implication relationship between events and activities, the language construct implies is used. The fact, that an individual belongs to the activity concept implies that it also is required by an event. The corresponding inverse relation is established analogously: `event` implies, that an activity is required. The definition of subconcepts of `event` and `activity` follows the same schema. For instance, the event `street_flooded` implies the requirement of an activity of type `blocking`. For more details about the representation and reasoning about plans, it is referred to [6].

```
(define-primitive-attribute label)
(define-primitive-attribute height)
(define-primitive-attribute object)
(define-primitive-role happens)
(define-primitive-role happens_in)
(define-primitive-role requires)
(define-primitive-role required_by)
(define-primitive-role indicator)
(define-primitive-role indicates)
(role-implies happens_in (inverse happens))
(role-implies happens (inverse happens_in))
(role-implies required_by (inverse requires))
(role-implies requires (inverse required_by))
(role-implies indicator (inverse indicates))
(role-implies indicates (inverse indicator))
```

Figure 2. Relation definitions in the high water protection plan model

```
(define-primitive-concept activity)
(define-concept blocking
  (and activity (some object barrier)))
(define-concept avoiding
  (and activity (some object street)))
(define-concept information
  (and activity (some object person)))
(define-primitive-concept event)
(define-primitive-concept street_flooded event)
(define-primitive-concept traffic_jam event)
(define-string-concept name TOP)
(define-concept water_level (some height TOP))
(define-concept situation (and (some label name)
  (some indicator water_level)
  (all indicator water_level)
  (some happens event) (all happens event)))
(define-concept emergency_situation_with_traffic_jam
  (and situation (some happens traffic_jam)
  (some indicator (some height (>= 2.90)))))
(implies activity (and (some required_by event)
  (all required_by event)))
(implies event (and (some requires activity)
  (all requires activity)
  (some happens_in water_level)
  (some happens_in situation)))
(implies street_flooded (some requires blocking))
(implies traffic_jam (some requires avoiding))
(implies water_level (and
  (some indicates situation)
  (all indicates situation)
  (some happens event) (all happens event)))
```

Figure 3. Concept definitions in the high water protection plan model

Considering the relations between the event and the situation concept, partially the same constellation is encountered as for the event-activity-relations. A situation is characterized by the events occurring in it, and therefore the situation concept is related by definition to the event concept by means of the role `happens`. The inverse relation however, is established by using the implies command following the same schema as for the event-activity-relations.

3.4 Water level definition

As stated above, a situation is mainly characterized by the water level occurring in it. Therefore, the description logic terminology in Figure 3 contains a water level concept to which the situation concept is related by the role `indicator`. To model the water level concept a numerical concrete domain is used. The water level is associated with a height which is a numerical concept and therefore defined by a numerical value (see [3]). In the definition of the general situation concept this value is not specified. However, more specific situation concepts can be characterized by indicating a precise boundary value for the water level height supposed to occur. In Figure 3 for instance, the specific situation concept `emergency_situation_with_traffic_jam` is associated with a water level higher than or equal to 2.90 m.

```
;; water level dependent events
(instance Neckarweg_flooded street_flooded)
(instance block_Neckarweg blocking)
(related Neckarweg_flooded
  block_Neckarweg requires)
(instance water_level_2.90 water_level)
(instance water_level_2.90 (some height (= 2.90)))
(related water_level_2.90
  Neckarweg_flooded happens)
(instance water_level_3.55 water_level)
(instance water_level_3.55 (some height (= 3.55)))
(instance B37_Alte_Bruecke_flooded street_flooded)
(related water_level_3.55
  B37_Alte_Bruecke_flooded happens)
;; water level independent events
(instance avoid_Hauptstrasse avoiding)
(instance traffic_jam_Hauptstrasse traffic_jam)
(related traffic_jam_Hauptstrasse
  avoid_Hauptstrasse requires)
;; current situation description
(instance current_situation
   (and situation (some label (aset now))))
(related current_situation
  water_level_2.90 indicator)
(related current_situation
  traffic_jam_Hauptstrasse happens)
```

Figure 4. World model of the high water protection plan

According to this terminological framework, the world model contains several water level individuals (see Figure 4). They correspond to the different water level heights which can possibly occur and which have significantly different consequences. Therefore, such an individual is characterized by a concrete water level height. Besides, it is related to several event individuals representing the events expected to occur with respect to it. Finally, a relation between the current situation individual and its corresponding water level has to be established. In the example the individual `water_level_2.90` is chosen.

Since the relation `indicator` between a situation and a water level is defined to be a role, multiple water level individuals can be related to the current situation individual. This is necessary if the occurring water level is a higher one, which therefore implies all the existing lower ones. Unfortunately, these relations have to be established explicitly, since associating only one high water level does not implicitly associate the lower ones. Even if on the conceptual level, a water level like `(>= 3.55)` inherits from one like `(>= 2.90)`, this is not of use here.

As stated above, the water level values can be used to specialize a situation concept with respect to a more general one with a larger water level height boundary. Furthermore, a concrete water level height characterizes the current situation individual via the water level individual, as it can be seen in the example world model. So, the concrete water level values affect the subsumption and the recognition inference, and therefore also classification and realization. However, the values do not have any impact on the retrieval of situation-dependent activities.

3.5 Water level dependency of events

Most of the events which are occurring in a high water situation only depend on the water level. However, there still might occur some events which are water level independent like a traffic jam in some street, for instance. While in the high water protection plan of Heidelberg only water level dependent events are considered, in the description logic model also events of the second type are taken into account. To discuss the model of the water-level-dependency or -independency of the events the situation concept definition in Figure 3 is considered. First, the situation concept is characterized by the water level. The water level implies certain events, which is modeled by the language construct implies. Second, the situation concept is also directly related to the event concept by means of the role `happens`. This allows for a situation description based on events which are not only the consequence of a certain water level height.

The situation characterization described above yields two types of events in the corresponding world model (see Figure 4). On the one hand, the current situation individual is related to a water level individual, which is related by the role `happens` to its consequence event individuals. For a water level of 2.90 m this is the event `Neckarweg_flooded`, for instance. On the other hand, water level independent events like `traffic_jam_Hauptstrasse` are directly related to the current situation individual. So, in the presented model *two* levels of situation-related events are encountered. Water level independent events, which cannot be foreseen with respect to a given water level, are directly related to the situation. Water level dependent events however, are related to the water level which, as an indicator, forms the situation description. So, the latter events are only indirectly related to the situation concept. There is an intermediate concept which establishes the relation. In any case, the presented model enables a more flexible situation description with respect to the original high water protection plan of Heidelberg.

3.6 The model at work

In this section it is illustrated, how the presented description logic model can be used to manage the reactions in a high water situation. For this task, the description logic inferences are of particular interest. As already stated, the description logic inference for testing the model consistency can be applied. Besides, a situation taxonomy can be constructed using classification. The concept `emergency_situation_with_traffic_jam` for instance, can be classified to be a more special one than `situation`. Then, realization enables the assessment of concrete disaster situations. In the given example, the current situation individual can be realized to be of the type `emergency_situation_with_traffic_jam`. Furthermore, the activities to be executed in a given situation can be queried by means of the retrieval inference. In the example world model, these are the activities `block_Neckarweg` and `avoid_Hauptstrasse`.

The retrieval of situation-dependent activities, which is the most important reasoning task supported by the presented model, still needs some more explanation. As illustrated in Figure 3, 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. In the example, the current situation is associated with the label now. By referring to the current situation individual on the conceptual level the following concept description can be constructed:

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

The retrieval of this description yields the activities depending on the current situation. The first argument of the or expression refers to the activities required by water level independent events, whereas by its second argument the activities corresponding to water level dependent events are addressed. The union of both argument expressions supports the retrieval of activities required by both types of events (see also [5]).

In a high water situation, the presented model can first be used to construct a description of the current situation. This is done by means of the corresponding water level and a library of predefined water level independent events. Therefore, such a situation description does not only depend on water level-implied events, but can be constructed by means of not already predefined combinations of events. Then, the realization inference allows to map the specified current situation to its corresponding situation concept in the situation taxonomy. This corresponds to an assessment of the situation. Furthermore, the retrieval of the activities and plans to be executed in the given situation can be done. By proposing these plans, the description logic system supports the process of deciding what to do in a disaster situation.

For being able to react also with respect to the situation's changes, the process of situation description and activity retrieval described above can be viewed as a step in a more complex description-retrieval-interaction, based on a world model which changes corresponding to the current situation. First, based on the water level forecast, a kind of standard situation development can be introduced into the world model. This means that individuals corresponding to the different water level heights supposed to occur are defined, associated with their occurrence time. This standard situation development implies a kind of standard plan, in fact being the predefined plan how to react to water level dependent events. Now, during the course of the situation, supplementary water level independent events, which have not been foreseen, can be added to the standard situation description for it better reflects the current situation. Thus, the activity retrieval with respect to this updated situation description also yields activities corresponding to the water level independent events. So, new circumstances occurring during the course of the disaster are taken into account. Repeating the situation description modification and activity retrieval procedure permanently, appropriate reaction measures can be proposed during the entire course of the disaster. Thus, a disaster plan is constructed, which adapts to the changing disaster situation (see [6] for details).

3.7 Description logics versus databases

In general, description logic knowledge bases have several advantages in comparison to relational or object-oriented data bases, since they allow for a representation incorporating also the semantics of the data to be stored. This semantics makes the *data* become *knowledge*. It allows for representing and reasoning about the data on a higher, semantically richer level. Concerning the data representation, description logics provide a highly expressive language. They incorporate the notion of inheritance together with the notion of relations. In database systems, hierarchically structuring the data and modeling relations at the same time cannot be realized that elegantly. To query data in description logic representation, corresponding reasoning procedures are used. They have important computational properties, which are decidability, soundness and completeness. The latter however, can not be provided by database systems, if they support such kind of reasoning at all.

For disaster management, some of the benefits of using description logics have already been mentioned above. First of all, the consistency of the data model can automatically be checked. Therefore, the system is also able to validate knowledge base modifications. Besides, a disaster situation description can be realized, i.e. assessed, to be of a certain situation type. Finally, with respect to a situation description the required activities and plans can be retrieved. Even if using a database system the activity retrieval would also be possible, in this case the answer would not necessarily be complete.

Another major advantage of a description logic knowledge base concerns its maintenance. It is, that the subsumption relations between concepts can incrementally be changed. These updates are possible, even if there have already individuals been defined. Since a concept as such cannot be redefined, modifications can be achieved by using so-called general concept inclusion axioms. They can define a concept expression to imply another one, and thus change the hierarchical order of the concepts. Using the classification inference, the concept hierarchy can be calculated at any state of the data input procedure, independently of the corresponding individual definitions. With respect to disaster management this means, that situation descriptions and reaction plans can incrementally be modified.

Another advantage of a description logic representation is, that an individual can be defined without indicating its precise type, i.e. one of its most specific concepts. Instead, individuals can be defined to belong to a more general concept. This is of interest, especially if an individual's precise type is unknown. Besides, for a given individual the set of most specific concepts can be calculated by means of realization. With respect to the application described here, this is particularly interesting for the assessment of a given disaster situation.

Even if description logics allow for representing and reasoning about data in a much more natural and elegant way than databases do, they have one main disadvantage. This is, that the worst case runtime of the algorithms underlying the reasoning inferences is not polynomial. However, experience shows that this is not a problem as far as knowledge representation for disaster management is concerned. On the one hand, various optimizations can be applied to the algorithms, and on the other hand, the knowledge bases corresponding to this application are not that big. In the case of Heidelberg for instance, the knowledge base only contains about 50 concepts and 1000 individuals. For other disaster scenarios the knowledge base may be larger, but the order of magnitude should be the same. Also, most of the necessary reasoning can already be done *before* the disaster situation occurs. This concerns checking the model consistency as well as constructing the concept hierarchy, which is the most costly task. Then, in a concrete disaster situation, only the situation realization and the retrieval of situation-dependent activities is needed.

Finally, it has to be mentioned, that for constructing a description logic model a knowledge engineer is needed. Such an expert has to acquire the knowledge of the application domain, and then to adequately introduce it into the description logic system, for which good knowledge about the representation formalism is necessary. Also, the updates of the base can only be done by such an expert. However, since the final system will be provided with a user interface, it is sufficient that the person working with the system during the course of a disaster is familiar with the appropriate decision making.

4 TOWARDS A REAL APPLICATION

Since the so far existing disaster management system is only a research prototype, it needs for more development to make it a system, which is usable for managing real high water cases. In this section, the actual status of the system is sketched. The means are described with which the prototype has been developed, and its features are summarized, also pointing out some remaining issues. Besides, the architecture planned for the final high water management system is presented.

4.1 The prototypical implementation

The main component of the presented disaster management system is the description logic reasoner. Here, the description logic system CICLOP is used [3], which has been developed at the LIIA of the ENSAIS in Strasbourg. The system is implemented in JAVA. It supports the standard description logic reasoning facilities for the highly expressive language $\mathcal{ALC}\mathbf{f}\mathcal{IHR}+$ with concrete domains[3]. The reasoning algorithms are tableau-based and thus sound and complete. Their implementation uses state of the art optimization techniques. For research purposes, CICLOP is freely available. At the LIIA, the system has been used in several applications (see [8], [9], for instance). Currently, representation and reasoning facilities to deal with temporal knowledge are integrated into the system. So, the reasoner is able to process description logic models as presented in this paper.

To conclude, the prototype provides the facilities needed, to allow for an efficient disaster management, i.e. mainly the retrieval of situation-dependent activities. Furthermore, it allows for checking the knowledge base consistency, assessing a given disaster situation, reacting to the situation's changes, and maintaining the knowledge base in a comfortable way. This is, why potential users, like also the high water board in Cologne on the Rhine, for instance, have shown their strong interest in this project. In Cologne, also a database system is currently used for high water management.

However, the current state of the system still requires further work to make it a real application. One limitation of the system is, that it cannot visualize plans. This however, obviously is one of the characteristics of a user-friendly disaster management system. So, one of the plans for the near future is to integrate a plan visualization tool into CICLOP to close this gap. It is intended to use the visualization component of ROGSI/DMS, a conventional commercial system for documenting and managing emergency cases in computing centers. ROGSI/DMS has been developed by ROG GmbH, Hirschhorn, Germany, with which the LIIA is collaborating.

Furthermore, the disaster scenario model used so far for the validation of the system still has to be extended. However, this model already incorporates an amount of data, which is suitable to illustrate the system's important features. Also, the system's decision making should be verified based on data of a former disaster case, for allowing to compare its proposals to the ones of a conventional system.

4.2 Architecture of the final system

The design of the real application will be as illustrated in Figure 5. Via a user interface the knowledge engineer and the end user can communicate with the description logic reasoner. The interface on the one hand provides means for the input of the disaster situation model and corresponding modifications as well as for displaying, whether the input is valid. On the other hand, it visualizes the modeled event library to allow the end user to construct a current situation description in an emergency case. It also displays the result of the situation assessment as well as the retrieved disaster plans. Besides, the water level forecast is incorporated into the system. Event individuals corresponding to the water level heights expected for certain times should automatically be introduced into the current world model. The world model can be divided into two parts. One part contains the static knowledge about the disaster scenario, the other the dynamic knowledge which is the water level forecast and the current situation description, for instance.

Besides, the system should support client server applications. A high water board usually consists of multiple persons, each of which having the responsibility for the execution of a certain part of the disaster plan. Furthermore, there is one overall responsible. Therefore, a disaster management system should be able to be queried in a decentralized way. Every staff member should be able to work in its office querying a common knowledge base. Besides it would be interesting to introduce access rights corresponding to the needs of the different staff members. The conceptual knowledge has to be shared by them all, but the world model can be partitioned corresponding to their responsibilities (see also Figure 5). Thus, each staff member has read and write access to its distinct world model part. The reasoning however, takes the whole knowledge base into account. The overall responsible can read the whole world model, and can thus follow the execution of the entire plan.

The maintenance of the presented disaster management system mainly consists in the maintenance of the knowledge base. Only very fundamental changes of the knowledge base might need for a change also in the user interface. As already stated above, maintaining description logic knowledge bases is quite straightforward. The satisfiability and the consistency test allow for continuously checking the consistency of a modified knowledge base. The expected necessary modifications of the relevant knowledge mainly concern statements about current circumstances, i.e. new building sites, or changed re-

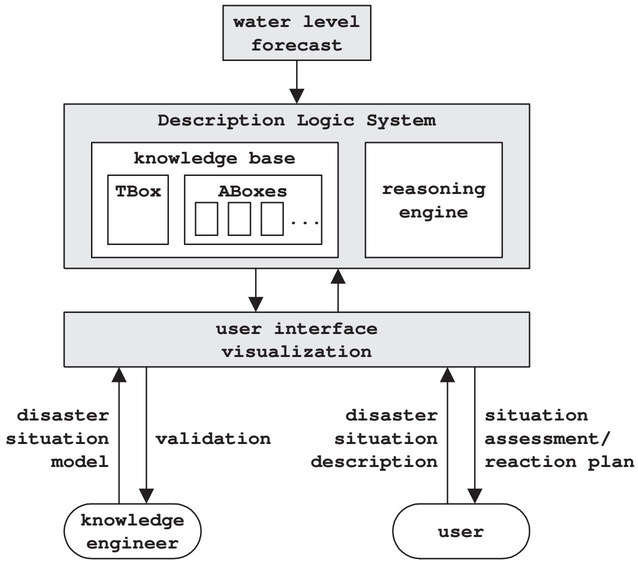

Figure 5. Design of the final high water management system

[3] See [3] for details about concrete domains provided by CICLOP.

sponsibilities for instance. Furthermore, knowledge gained by experience has to be incorporated into the system, new knowledge about more people wishing to be informed, for instance.

5 CONCLUSION

Using a description logic system as the kernel of a disaster management system for high water response, the benefits have been illustrated, which an artificial intelligence approach has in comparison to a conventional approach like using a relational or object-oriented database system. First, an introduction to the problem of high water management was given by means of the example of Heidelberg, where currently a database system is used to manage the so-called high water protection plan. Then, a description logic model of this disaster plan is presented, allowing for a high water management based on description logic reasoning methods. These methods have important computational properties, which are decidability, soundness and completeness.

The description logic model of the high water protection plan of Heidelberg allows for describing a disaster situation. This situation description is based on the water level forecast, but events which are water level independent are also taken into account. Besides, the model incorporates the notion of time and thus *real* plans, which allows for an appropriate description of reaction measures. Applying the description logic consistency test, the consistency of the model can automatically be checked and therefore possible modifications be validated. Furthermore, a concrete situation can be defined without indicating its precise situation type, which can then be calculated by the realization inference. This allows for assessing a given disaster situation description. Besides, with respect to a situation description the required activities and plans can be retrieved. For being able to react also with respect to the situation's changes during the course of disaster, the situation description can permanently be updated and reprocessed to propose corresponding reaction measures. Thus, a disaster plan is composed, which adapts to the changing situation. By proposing this plan, the system can support the decision process in a disaster situation. Finally, the model once being constructed and maintained by a knowledge engineer, can be used by a non-computer-scientist, if an appropriate user interface is provided. Concerning the maintenance of the model, thanks to the classification inference it is possible to incrementally modify situation descriptions and reaction plans.

To characterize the actual status of the presented research prototype, it can be stated that it provides the facilities needed, to allow for an efficient disaster management. In the near future, it is intended to integrate a plan visualization tool into the system. Besides, the systems' performance will be validated by means of a complete real world disaster scenario model. The verification of the systems' decision making will be done based on data of a former disaster case. However, to make the prototype a real high water management application corresponding to the design presented at the end of this paper, there still remain some steps to be done. It is planned to realize this application in collaboration with the ROG GmbH, Hirschhorn, Germany, already mentioned above. This application development project is strongly motivated by the interest of potential users like the high water staff in Cologne, for instance.

Finally, the description logic approach is also interesting for managing disaster scenarios and reaction processes corresponding to other types of disasters than high water. The principal approach stays the same even if the disaster environment, i.e. the local circumstances and the disaster type are different. Possible scenarios reach from other natural or environmental disasters, over problems in computing centers, to aeroplane kidnapping, for instance.

ACKNOWLEDGEMENTS

We thank Reinhard Vogt and Patrick Antoch from the high water protection staff in Cologne on the Rhine, and Peter Kullmann from the ROG GmbH, Hirschhorn, for the many interesting discussions.

REFERENCES

[1] A. Artale and E. Franconi, 'Temporal description logics'. In: *Handbook of Time and Temporal Reasoning in Artificial Intelligence*. MIT Press, to appear.

[2] F. Baader and B. Hollunder, 'A terminological knowledge representation system with complete inference algorithms', in *Proc. of the International Workshop on Processing Declarative Knowledge*, volume 572 of *Lecture Notes in Computer Science*, pp. 67–85, Kaiserslautern, Germany, (1991). Springer-Verlag.

[3] 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).

[4] C. Eikenberg. Journalist's manual on disaster management – 1998 – . German IDNDR-Committee for Natural Disaster Reduction, Bonn, Germany, 1998.

[5] 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).

[6] M. Kullmann, F. de Bertrand de Beuvron, and F. Rousselot, 'A description logic model for reacting in a dynamic environment', in *Proc. of the International Workshop on Description Logics 2000*, Aachen, Germany, (2000).

[7] J. Picht, 'Hochwasser-Informations und Schutz-System (HoWISS)', Technical report, CreativWare, Karlsruhe, Germany, (1999).

[8] D. Rudloff, *Modélisation conceptuelle et optimization des requètes dans une interface en langue naturelle pour des bases de données*, Ph.D. dissertation, Université Louis Pasteur, Strasbourg, France, 2000.

[9] M. Schlick, *CICLOP – Application of Description Logics to Configuration*, Ph.D. dissertation, Université de Haute Alsace de Mulhouse, France, 1999.

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