River Basin Management with SPIN

María-del-Mar Gallardo, Pedro Merino, Laura Panizo(✉), and Alberto Salmerón

Andalucía Tech, Dept. de Lenguajes y Ciencias de la Computación, Universidad de Málaga, Málaga, Spain
{gallardo,pedro,laurapanizo,salmeron}@lcc.uma.es

Abstract. This paper presents the use of the SPIN model checker as the core engine to build Decision Support Systems (DSSs) to control complex river basins during flood situations. Current DSSs in this domain are mostly based on simulators to predict the rainfall and the water flow along the river basin.

In this paper, we propose a scheme that integrates simulators in the water domain with additional logic in PROMELA to represent basin elements, such as dams, their management rules, the evolution of dam parameters (e.g. level or discharge capacity), and user defined constraints in the whole basin over time. Then, we use the exploration capabilities of SPIN to find out which sequences of operations over the dams produce a global behaviour that mitigates the effect of floods according to user defined constraints along the river basin. Although the method is general for any river basin with dams, it has been evaluated in a real basin in the south of Spain.

1 Introduction

Mediterranean countries, like Spain, have built many big dams which ensure the water supply to the population during typical long drought periods, and also limit the damage caused by floods by means of their flood discharge capacity (Spain is the fourth country in number of big dams, following USA, China and India). However, experience has demonstrated [14] that during a flood episode, the incorrect management of a dam can produce disasters worse than if the dam did not exist. This problem is even more complex when there are several dams in the same river basin, because of the difficulty to predict the cumulative effect of water discharging at several points in parallel.

The most common way to manage dams during flood episodes is based on the combination of weather forecasts and ad-hoc decision rules. The dam operators usually estimate the input of water over time (the *input hydrograph*) with official forecasts, and employ a pre-designed catalogue of management rules to decide water discharges. These rules take into account different parameters, e.g. the

This work has been partially funded by the Regional Government of Andalusia under grant P11-TIC-07659, and the European Comission under FP7 Environment project SAID, Grant agreement 619132, and FEDER.

D. Bošnački and A. Wijs (Eds.): SPIN 2016, LNCS 9641, pp. 78–96, 2016.
DOI: 10.1007/978-3-319-32582-8_5

reservoir level, the weather forecast, the current downstream drainage capacity, etc. One recent trend is the development of software systems that act as reliable Decision Support Systems (DSSs) to assist dam managers in floods [10,11]. These DSSs are based on simulation models that allow a detailed and faithful representation of a real-world system with complex mathematical models. However, they can only show the effect of applying a specific management policy. With this approach, a large number of trials is necessary to establish an optimal policy, which can drastically reduce the time to react to the flood.

In [7], we introduced the use of model checking as a promising novel approach to build more powerful DSSs for flood management in a single dam. The proposal works as follows. We describe the dam's physical components (like spillways to discharge water) with PROMELA as well as a non-deterministic process simulating the dam manager's actions on the physical discharge elements. An external tool provides the representation of the expected input water flow to the dam over time as a hydrograph. Finally, we added constraints to keep the dam level between a minimum and maximum value or to discharge a maximum flow downstream. Constraints are encoded as a *never claim*, a special PROMELA process. SPIN uses these inputs and generates a counterexample that corresponds to the manoeuvres over dams that satisfy the constraints.

Our previous work focused on managing a single dam. Thus, to manage a complex river basin with more than one dam, the dam operators must manually run our DSS for each dam and the hydrologic basin models, appropriately linking the inputs and outputs to simulate the state of the basin. However, this is unfeasible in practice. In this paper, we extend our previous work to use SPIN as the core engine of a DSS for the coordinated management of all the dams in a river basin. We reuse the initial work in [7] to model every dam in the basin in a single PROMELA model, and we integrate an external hydrologic river basin model to simulate the effects of the dams downstream. The constraints over basin locations are checked externally, and the result of the evaluation directly affects the SPIN exploration algorithm. The PROMELA model of the river basin now includes several dams, integrates different external (hydrodynamic) models and safety constraints over the basin, and the management rules modeled as a non-deterministic process. We make extensive use of embedded C code in PROMELA, tracking a minimal number of variables and abstractions to reduce the state space. The embedded C code is also used to deal with discretized continuous variables, and to propagate the effects of dam manoeuvres throughout the basin, using different time references. The output of the verification process is a sequence (or several sequences) of coordinated manoeuvres for all the dams to assist the manager in the decision making process. We have implemented the system for a real river basin in the south of Spain, and validated its performance and usefulness with real scenarios.

To the best of our knowledge, there are no works on the use of model checking to synthesize the manoeuvres in flood episodes. Compared with other works in this domain, like FCROS [9] in Poland, DESMOF [2] in Canada, or IMSFCR [4] in China, our approach offers several novelties. While FCROS and DESMOF

only include simulation of flood policies, our DSS and IMSFCR also calculate the necessary operations. IMSFCR makes multi-objective optimisation based on fuzzy iteration, but it does not consider hydrological models downstream.

The rest of the paper is organized as follows. Section 2 provides some background on dam management and presents the case study used in the paper. Section 3 describes our approach based on model checking, while Sect. 4 details how to build the PROMELA models of the river basin. Section 5 explains how to define constraints over the dam parameters and the basin flows. Section 6 is devoted to the evaluation with the case study, and finally Sect. 7 presents the conclusions and future work.

2 Background on Flood Management

Flood management is a complex task, especially in Mediterranean basins, which are characterized by long drought periods and short but intense rainfalls. Dams are an important element in this kind of basin, as they store water for two main purposes: to supply water to the population in drought periods and to control floods. With correct management, a dam can smooth the peak rainfall and avoid downstream flooding.

Dams are equipped with different types of discharge elements. Figure 1 shows the discharge elements of the Conde del Guadalhorce dam, which is included in our case study. Spillways are gates for flood regulation. They usually have the highest discharge capacity. Outflows can be used for flood regulation or other water uses (supply, irrigation or energy production), and their discharge capacity is lower. In general, the outflow capacity of a dam's outlets depends on their location, which is fixed, their opening degree, which is variable, and the dam

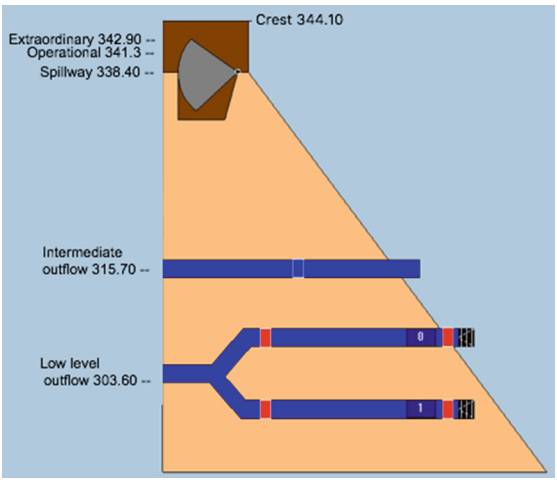


Fig. 1. Dam discharge elements

level, which changes following Eq. 1, where $V(t)$ and $V(t-1)$ are respectively, the water stored at instant t and $t-1$, $Inflow(t)$ represents the water input and $Q_{s_i}(t)$ is the water discharged by outlet s_i. Equation 2 shows the discharge capacity of a spillway gate, where h_{s1} and h_{s2} are the water level and the position of the gate evolving over time. The other components, C and L, depend on the geometry of the gates and can be considered constant.

$$V(t) = V(t-1) + (Inflow(t) - \sum_{i=1}^{n} Q_{s_i}(t)) \tag{1}$$

$$Q_s(t) = CL(\sqrt{h_{s1}(t)^3} - \sqrt{h_{s2}(t)^3}) \tag{2}$$

Basin and dam management are controversial issues, especially in flood scenarios. Dam management has been traditionally carried out by a human operator, who has to manage in parallel the different outflow elements. In addition, a basin can include several dams in parallel and/or cascade, and the management of one dam can have a direct impact on the other dams and on the population downstream. Moreover, in Mediterranean basins, with short and intense rainfalls, dam managers have little time to decide how to operate to ensure dam safety considering the management of the other dams.

2.1 The Guadalhorce Case Study

In this work, the case study is the Guadalhorce River basin, located in the province of Málaga, in the South of Spain. The basin has a total area of 3,175 km^2 and is responsible supplying water to the city of Málaga, a touristic city with a population of more than 500,000 inhabitants. In addition, the basin supplies water and irrigation to other small cities of the province. The Guadalhorce basin has a short concentration time: water flows from the headwater to the mouth in approximately 8 h. Figure 2 shows the basin area. The Guadalhorce is the main river of the basin. Its flow is controlled by means of three dams (Guadalhorce, Guadalteba and Conde del Guadalhorce), which are located at the confluence of the Guadalhorce with the Turón and Guadalteba rivers. The three dams are managed by the Andalusian Regional Ministry (Consejería de Medio Ambiente y Ordenación del Territorio), and are used for flood management and water supply. Table 1 shows the main data of the three dams.

The management of the Guadalhorce and Guadalteba dams is special. These dams are separated by a wall measuring 355 masl (meters above sea level) from the base. During the flood season water is usually over this level and both dams are managed as a single dam. In fact, they have been designed to share the spillway, which is located in the Guadalteba dam. From now on, we will refer to the Conde del Guadalhorce dam as CGH, and to the Guadalhorce and Guadalteba dam jointly as GH-GT. Since the three dams and their outlets are very close, an important aspect of their management is the synchronization of peak discharges to avoid downstream flooding. In the main river channel there are no other dams downstream, but there are many tributaries that flow into the Guadalhorce River. The largest tributaries in volume are the Grande River, which flow

Table 1. Main characteristics of the dams

	Guadalhorce	Guadalteba	Conde G.
Operational level (masl)	362.25	362.25	341.3
Volume at op. level (hm^3)	125.8	153.3	66.5
Extraordinary level	364.0	364.0	342.9
Crest level	367.0	367.0	344.1
Low level outflow			
Number of gates	2	2	2
Level (masl)	302.5	308	304
Spillway			
Number of gates	-	4	2
Level(masl)	-	356	338.4

masl: meters above sea level

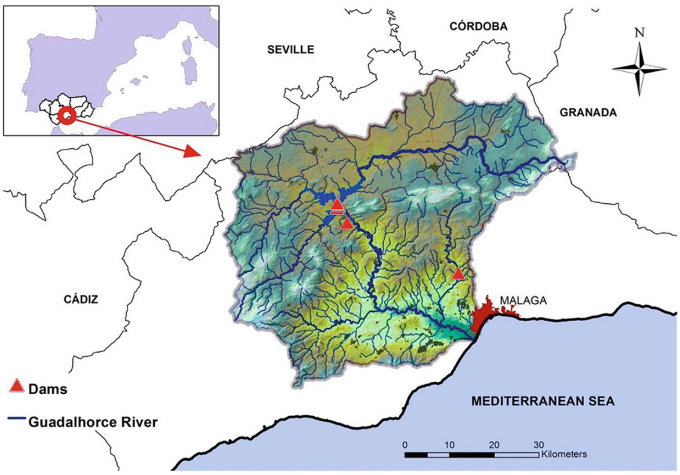


Fig. 2. Guadalhorce river basin

into the Guadalhorce 35 km downstream, and the Campanillas River, which merges near the river mouth.

From the point of view of flood management, the basin has 4 locations in which water flow must be monitored. The first one is La Encantada hydroelectric plant, which is located 7 km downstream of the dams. The second and third locations are at the confluence of the Grande River and the Campanillas River with the main river channel. Finally, the fourth point is the river mouth, which is located in the city of Málaga, near the international airport.

In this work we present a DSS for this basin based on model checking. The dam manager has to define constraints that describe the desired behaviour of

the basin for a specific flood episode. Then, the DSS produces a sequence of manoeuvres that satisfies the constraints. Figure 3 shows an example of the results produced by the DSS. At the top, are the level and total outflows of the dams. Then, the evolution of gates' openings is displayed. Finally, on the bottom the water flows in the basin are shown.

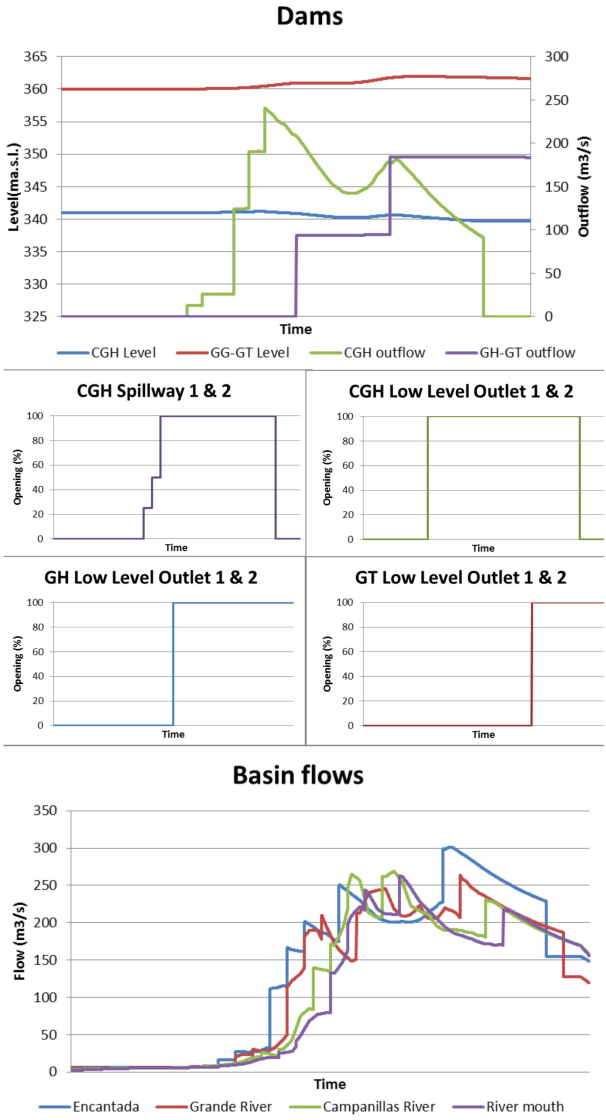


Fig. 3. Synthesis of manouvers

3 Approach with Model Checking

We use model checking in order to synthesize management recommendations that meet the constraints given by the dam manager. We use SPIN [8] as the underlying model checker, and, in consequence, PROMELA as modelling language. In addition, the PROMELA model also uses an external model for the river basin, developed independently. Given a set of constraints over the variables of the dams and the river basin, SPIN will explore exhaustively all possible manoeuvres, and produce a suitable set of recommendations for the dam manager that fulfils the constraints.

Figure 4 shows an overview of our approach, and how the PROMELA model used by SPIN and the external river basin model interact. First, we must model the dam (or dams) which will be operated by the dam manager. The management of the dam outlets is defined in a partially non-deterministic model, which determines when the gates should be opened or closed according to the operation rules, affecting variables such as the water outflow and the dam level over time, and consequently the outflow across the river basin. The latter is provided by an external river basin model, which is not modelled in PROMELA. The external river basin model takes the outflow of the dams and other environmental aspects as input, and computes the flow at several points across the river basin. All these models will be described in Sect. 4. Finally, the user may set restrictions on the outflows of the dam or at points of interest across the river basin, using timed automata, or upper and lower curve bounds, as explained in Sect. 5.

Once the models and the restrictions are in place, the analysis can proceed. The dam manager modelled in SPIN is executed periodically to select and apply one manoeuvre from those available from the rules. The dam model computes the water discharged between manoeuvres. This will serve as input for the external river basin model, which is also executed periodically to compute the outflow along the basin.

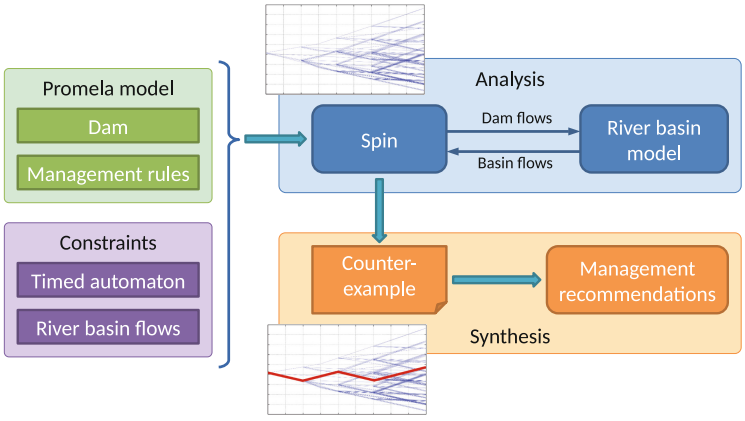


Fig. 4. Overview of synthesis of recommendations for dam management

Depending on the state of the dams and the set of management rules, the management model may have several options available whenever it has to make a decision. These options constitute the state space to be explored. Thanks to the exhaustive exploration provided by Spin, the analysis can obtain all possible manoeuvres that the dam manager can choose during the course of an episode. If a particular series of actions leads to a state that violates the constraints over the dams or the basin, Spin will backtrack and try different manoeuvres, until the end of the episode is reached while fulfilling specified constraints. This will produce a counterexample that contains the manoeuvres that satisfy the constraints.

4 Dam and Basin Modelling

The management of the river basin is based on the analysis of a Promela basin model against a set of properties that describes the constraints of dam and basin parameters, such as dam level or water flow. It is worth noting that some of these parameters have a continuous evolution over time and have to be properly represented to avoid state-space explosion problems.

The global model of the basin comprises different sub-models, such as the model of the dams and their outlets, or the model of the water flow downstream. As mentioned above, in this work, we have used Promela as the modelling language, embedding C code to describe some complex mathematical equations. In addition, we have used C code to embed the interaction with external models developed by third parties.

In this section, we describe the main structure of these sub-models, and some specific issues for the case study.

4.1 Dam Model

There are two main aspects that must be taken into account by a dam model. First, it must describe the evolution of the main variables of the dam over time and how they are related, e.g. the relation between dam volume and dam level (dam's bathymetry), the relation of the stored water volume and the water inflow and outflow over time, etc. Second, it must provide a mechanism to change the state of the dam outlets, i.e. their opening degree, during the analysis.

In [7], we presented a simplified version of a dam model. We describe the dam as a Promela proctype that receives commands from the dam manager (another proctype) to change the opening degree of the outlets. After updating the state of the outlets, the model computes the flow discharged by means of embedded C code that describes the outlet equations. In this work, we have improved the dam model such that it is now possible to describe and analyze the behaviour of dams with more outlets, more outlet opening degrees, and longer flood episodes. In addition, we also allow non operative outlets, i.e. outlets whose state cannot be changed. Figure 5 shows the skeleton of the dam model used in the case study.

To reduce the state space to be explored, we make extensive use of embedded C code, and export some of the C variables into Spin's state. Some of these variables have been declared as UnMatched, i.e. outside the scope of Spin's state matching algorithm, to reduce the of number states. In other cases, we have abstracted the values of several UnMatched variables into a single variable in Spin's state. For instance, the current opening degree of each gate of an outlet is UnMatched, but we include a single matched variable that abstracts these values. This abstract variable only provides the number of gates that are opened or closed, and not which ones are opened or close, which is not of interest from the point of view of the exploration. However, if Spin backtracks, the exact state of each gate will be recovered.

Finally, we have defined a systematic way of defining this kind of dam model, which has been implemented in a prototype tool as part of the SAID project [1]. Using this tool, it is possible to easily develop models of new dams without errors.

4.2 River Basin Model

To manage a complete river basin, we need a hydrological model that simulates the water inflow to the dams and the flow downstream. There exist different hydrological models and simulation engines that fulfil our needs. In particular, other partners in the SAID project have used a basin model through the WiMMed tool [5,13]. Instead of translating these models to Promela code, we treat them as black boxes that produce the required output given the appropriate inputs, such as the outflows from the dams and the environmental inflows.

Before the analysis, we first run the black box to produce the inflow hydrographs of the dams for the particular flood episode we are analysing. These inflow hydrographs are independent of the manoeuvres performed during the analysis. Then, during the analysis with Spin, the black box model will be executed periodically using embedded C code to simulate the water flow downstream for different sets of manoeuvres. The model will return the resulting hydrographs at predefined locations in the basin, showing how the manoeuvres affect the flow along the river basin.

While the Promela model tries different manoeuvres with a short time period, e.g. every hour, external river basin models are usually meant to simulate longer periods of time, e.g. several days. Executing the external river basin model for each new manoeuvre to find out their effect downstream can be very time consuming. To solve this problem, we use a longer period to execute the external model, i.e. the external model will be executed after the management model has selected the manoeuvres for the past few hours.

In addition, we are only interested in the portion of the simulation which was affected by the chosen manoeuvres. The external model provides hydrographs at several points of interest along the river basin, which are increasingly further away from the dam. Although the distances are constant, the time elapsed between the manoeuvres and the water affecting these points downstream varies

```
/* Macro definition */
#define action_spill_gg(id,ap)
  c_code{if(spill_gg_enabled[PDam->id]==1){spill_gg_opening[PDam->id]=ap;}}
#define outflow_spill_gg(id)
  c_code{spill_gg_outflow[PDam->id]=spill_gg_contribution
  (spill_gg_opening[PDam->id],dam_h_gg,1);}
#define update_state_spill_gg
  c_code{now.spill_gg_outlet_type_state = update_outlet_type_state
  (&spill_gg_opening, MAX_SPILL_GG);}
...
/* GH-GT Dam variables*/
c_track "&dam_h_gg" "sizeof(double)" "UnMatched";
c_track "&dam_v_gg" "sizeof(double)" "Matched";
c_track "&inflow_gg" "sizeof(double)" "UnMatched";
c_track "&outflow_gg" "sizeof(double)" "UnMatched";
/*Spillway GH-GT Dam - Variables*/
c_track "&spill_gg_outflow" "sizeof(spill_gg_outflow)" "UnMatched";
c_track "&spill_gg_opening" "sizeof(spill_gg_opening)" "UnMatched";
c_track "&spill_gg_enabled" "sizeof(spill_gg_enabled)" "UnMatched";
int spill_gg_outlet_type_state;
mtype={spill_gg_ap0, spill_gg_ap1, spill_gg_ap2, spill_gg_ap3};
chan cmd_spill_gg[MAX_SPILL_GG] = [1] of {mtype};
/* LLO GH-GT Dam - Variables */
...
/* CGH Dam and outlets variables */
...
proctype Dam() provided(current==1)
{
 int id;
 atomic{
   do
   ::(c_expr{t==0})-> break;
   ::else ->   id = 0;
     do  /* Spillway GH-GT Dam - Command reception */
     ::(id<MAX_SPILL_GG)->
        if
        ::(cmd_spill_gg[id]?[spill_gg_ap0])-> cmd_spill_gg[id]?_;
            action_spill_gg(id,SPILL_GG_AP0);
        ::(cmd_spill_gg[id]?[spill_gg_ap1])-> cmd_spill_gg[id]?_;
            action_spill_gg(id,SPILL_GG_AP1);
        ::(cmd_spill_gg[id]?[spill_gg_ap2])-> cmd_spill_gg[id]?_;
            action_spill_gg(id,SPILL_GG_AP2);
        ::(cmd_spill_gg[id]?[spill_gg_ap3])-> cmd_spill_gg[id]?_;
            action_spill_gg(id,SPILL_GG_AP3);
        ::else -> skip;
        fi;
        id= id +1;
     ::else-> id=0; break;
     od;
     do /* Spillway GH-GT Dam - workout outflow */
     ::(id<MAX_SPILL_GG)-> outflow_spill_gg(id); id=id+1;
     ::else-> id=0; break;
     od;
    update_state_spill_gg;
/* Rest of outlets GH-GT and CGH Dams */
...
     current=_pid+1;
   od;
   current=_pid+1;
 };
}
```

Fig. 5. Promela dam model

dynamically depending on several conditions. For our analysis, we use the estimated minimum of these times for each point (provided together with the river basin model) to determine which part of the basin flows can be safely analyzed. If a property is violated in this part, SPIN will backtrack and try another set of manoeuvres, as explained previously.

It is worth noting that we do not check the constraints in a portion of the basin flows that has not been affected by the water discharged from the dam. If we did, SPIN could detect a constraint violation in an unaffected portion of the basin flows, and then incorrectly assume that the chosen manoeuvres had a negative impact. This would lead to backtracking and choosing a different set of manoeuvres, while in reality the discarded set could be valid. If these manoeuvres did in fact have a negative impact, this will be eventually detected by the analysis, and they will be discarded during backtracking.

This approach to timing can be seen in Fig. 6, which shows a dam and two points (#1 and #2) along the river basin. The Y axis shows the minimum distance in hours between the dam and the two points. The dots along the dam line represent manoeuvres chosen by the management model. The dashed lines show the minimum time it takes the water released from the dam to reach and influence the two basin points. For instance, water released in t_0 will reach points #1 and #2 at t'_0 and t''_0 at the earliest, respectively. A flow is shown for each element above its line, e.g. showing how the peak discharge in the dam is smoothed as it flows downstream. Also note that any flow from the river basin model before t'_0 and t''_0 will not be affected by any of the manoeuvres.

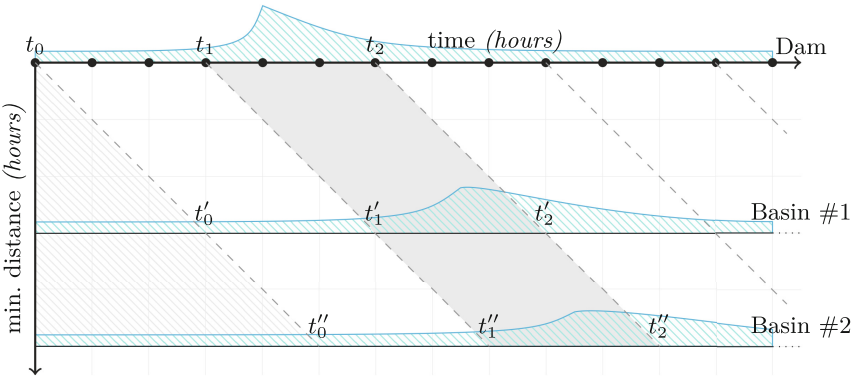


Fig. 6. Timeline of different basin elements

In this example, the dam manager chooses a manoeuvre every hour, but the external model is executed every three hours. Between t_1 (inclusive) and t_2 (non-inclusive) the manager performs three manoeuvres. The shaded area shows the part of the river basin that will be affected by these manoeuvres, i.e. interval $[t'_1, t'_2)$ for point #1 and $[t''_1, t''_2)$ for point #2. The constraints set by the user in

the river basin will be checked for these intervals. If one of the constraints is not met in these intervals, Spin will backtrack and try a different set of manoeuvres. If the water is slower than the minimum time, possible constraint violations will be detected later, but will result in backtracking to try new manoeuvres as well.

4.3 Management Rules

The management rules define how the dam manager has to act in flood episodes. These rules are included in the dam manual and consider average and maximum rainfalls. Rules are usually described as *if-then* statements to simplify their application during flood episodes. Our objective is to provide the dam (basin) manager with a set of manoeuvres that leave the basin and its dams in a safe and desired state. To this end, we have extended and modelled the management rules defined for the three dams of the Guadalhorce basin. Figure 7 shows the skeleton of the current management rule model. It describes most of the original if-then rules included in the dam manual. For instance, line 12 implements a rule that closes all outlets if the dam level is under $NMN\_C - SHELTER$ and the dam level is decreasing. In addition, this model monitors the dams and operates (or not) periodically to model the real management and also reduce the state space. In this case study, the model can operate the dams each one or two hours (e.g. lines 24 and 28) depending on dam's state.

The management rule model includes non-deterministic choices, making it possible to synthesize manoeuvres that satisfy different constraints. The number of non-deterministic choices directly affects the state space of the model. In addition, the coding of the model directly affects the analysis performance and the results. For instance, we have used the order of non-deterministic choices to first explore sequences of manoeuvres with a lower cost; that is, the DSS will return solutions with fewer operations if possible, which are more suitable in real flood management. Thus, this model can be refined to produce appropriate manoeuvres in a short period of time with the resources available.

5 Constraints for Synthesis of Management Decisions

The objective of our DSS is to provide different alternatives to manage the dams of the basin in flood episodes. Given a particular flood scenario, the DSS has to synthesize a set of manoeuvres that preserve the safety of the dams and the basin. For each scenario, we describe the safety of the dams and the basin as a set of constraints. For instance, during the flood season it is desirable to maintain dam levels lower than in other seasons, and keep the flow at the river mouth under a threshold to avoid flooding the airport. These constraints are then transformed into safety properties that are analyzed on the model using Spin. The non-deterministic behaviour of the operation rule model, presented in Sect. 4.3, allows the DSS to come up with different basin management alternatives.

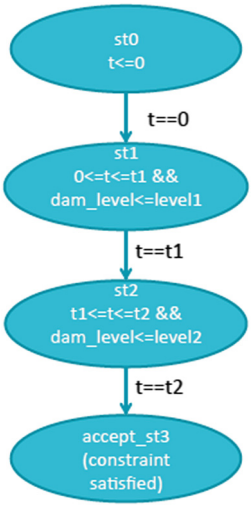
In [6], we described the constraints as LTL formulas that Spin automatically translates to a *never claim* proctype that represents the Büchi automaton

```
#define wait(x) if ::c_expr{PRules->x<t}->set(t_user, x);current=1;
                   ::else-> break; fi
c_decl{ double last_dam_h_c,last_dam_h_gg, QNMN_C, QNMN_GG;}
c_track "&last_dam_h_c" "sizeof(double)" "UnMatched";
c_track "&last_dam_h_gg" "sizeof(double)" "UnMatched";
proctype Rules() provided(current==_pid)
{
  atomic{
    do
    ::(1)->
      if /* Rules for CGH Dam */
      :: c_expr{(dam_h_c<NMN_C-SHELTER_C)&&(dam_h_c<=last_dam_h_c)}->
           cmd_spill_c[0]!spill_c_ap0; /* ... close all */
      :: c_expr{(dam_h_c<NMN_C)&&(dam_h_gg>NMN_GG)}-> /* close all to let
           GH_GT discharge */
      :: c_expr{(dam_h_c>=NME_C)&&(dam_h_c>last_dam_h_c)}-> cmd_spill_c[0]!
           spill_c_ap3; /* ... open all */
      ::else->  /* Open non-deterministically CGH Dam outlets */
      fi;
      if /* Rules for GH-GT */
      ::c_expr{(dam_h_gg<NMN_GG-SHELTER_GG)&&(dam_h_gg<=last_dam_h_gg)}->
            /* close all */
            wait(120);
      ::c_expr{(dam_h_gg<NMN_GG-SHELTER_GG)&&(dam_h_gg>last_dam_h_gg)&&
                 (inflow_gg<QNMN_GG)}&&(spill_gg_outlet_type_state != 0)->
            /* close spillway */
            wait(120);
      ::c_expr{(dam_h_gg>=NME_GG)&&(dam_h_gg>last_dam_h_gg)}-> /* open all */
            wait(120);
      ::else-> /* Open non-deterministically GH-GT Dam outlets */
      wait(60);
      fi;
    od;
    current = 1;
  }
}
```

Fig. 7. PROMELA operation rules

associated with the LTL. However, LTL is not suitable for describing properties that refer to precise time instants. In [12], we defined the constraints as Timed Automata [3], which are automata extended with real-valued clocks, and we proposed a translation from Timed Automata to never claim, using a discretized clock variable. In both cases, constraints were always relative to dam parameters, such as the dam level or the outflow. The state space of the discretized automaton is a subset of the original, thus we ensure that in this discrete time instant the dam model satisfies the constraints. However, given the nature of the variables modeled (dam level, water flow, etc.) and the small time step used, the evolution of variables can be considered lineal between two time instants, which allow us to guarantee that the constraints are also satisfied between two discrete time instants.

In this work, we allow the definition of constraints over dam parameters and flows at locations of interest in the river basin. The evaluation of these two types of constraints is slightly different. We use the approach presented in [12] to define and evaluate constraints over dam parameters. In this case, the constraint is described as a timed automaton and translated into a never claim with an acceptance state that is only reached if the constraint is satisfied. When SPIN analysis reaches the acceptance state, the analysis ends and returns the




```
1  #define inv0 c_expr{ t<=0 }
2  #define guard0 c_expr{ t==0 }
3  #define inv1 c_expr{ dam_level<=level1 && t
       <=t1 && t>=0 }
4  #define guard1 c_expr{ mitime == t1}
5  #define inv2 c_expr{ dam_level<=level2 && t
       <= t2 && t>=t1 }
6  #define guard2 c_expr{ t==t2 }
7  never {
8  st0: if
9       :: (guard0 && inv0 && inv1) -> goto st1
10      :: (inv0) -> goto st0
11      :: (Inicio[0]@l_init) -> goto st0
12      fi;
13 st1: if
14      :: (guard1 && inv1) -> goto st2
15      :: (inv1) -> goto st1
16      fi;
17 st2: if
18      :: (guard2 && inv2) -> goto accept_st3
19      :: (inv2) -> goto st2
20      fi;
21 accept_st3:
22      if
23      :: (1) -> skip
24      fi
25 }
```

Fig. 8. Constraint described as (a) timed automaton and (b) never claim

sequence of states leading to this *error* state. The sequence of states includes the scheduling of manoeuvres performed by the operation rule model. Figure 8 shows an example of a timed automaton and never claim used to synthesize a set of manoeuvres. The constraint is to maintain the `dam_level` under a threshold `level1` in period $[0, t1]$ and under threshold `level2` in period $[t1, t2]$. When the never claim reaches the state `accept_st3`, the analysis will stop and return the execution trace of the basin model, including the management rules applied to dams.

To analyze constraints over basin flows, we have to extend this approach. The main reason is that the external hydrological model returns the temporal evolution of the flows for future time instants that are not easily synchronized with the timing of the PROMELA model. The constraints over basin flows are described as curves that serve as the upper or lower limit for some of these flows. In Sect. 4.2 we explained how the hydrological model is periodically executed to compute the effects of the manoeuvres downstream. Figure 9 shows how the external model is called (line 5) and how the constraints over basin flows are evaluated (line 7). When execution of the external hydrological model finishes, its results are stored in hidden C structures. These values are checked by the function `basin_check_constraints`, which compares the results against the constraints set by the user. Only the interval affected by the manoeuvres since the last time the external model was executed is checked, taking into account the distance from the dams to each basin point of interest. Observe that the function is called using the primitive `c_expr` instead of `c_code`. If the checks succeed, the analysis can continue, but if the checks fail, the instruction is not

executable and SPIN has to backtrack to a state where different operation rules can be selected.

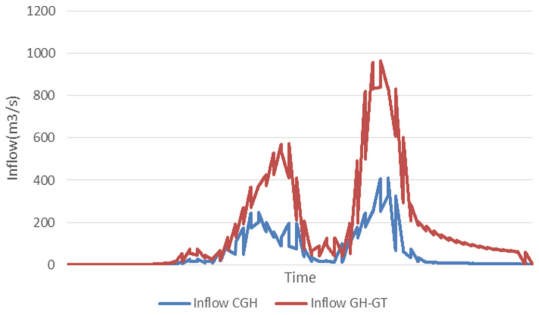
```
proctype Timer()
{ ...
if
:: (t_basin == 0) ->
  c_code { basin_execute_model(t, cycles); }; /* Run hydrological model */
  /* Check constraints over basin flows; block if not satisfied */
  c_expr { basin_check_constraints(t-BASIN_TIME_STEPS, BASIN_TIME_STEPS) };
  set(t_basin, BASIN_TIME_STEPS)
:: else -> skip
fi;
... }
```

Fig. 9. Evaluation of constraints over basin flows

When constraints are only specified over the basin flows, the never claim has to check that time t reaches the end of the episode.

6 Evaluation

In this section, we analyze a flood episode of 60 h to evaluate the performance of the DSS. Figure 10 shows the dam inflows and their initial state. Since the levels of the Guadalhorce and Guadalteba were above the separation wall, we can manage them as a single dam.



	GH-GT	CGH
Level (m.a.s.l.)	360	341
Volume (hm^3)	246.83	66.49
Spillway	close	close
Low level outflow	close	close

Fig. 10. Flood episode (a) inflow and (b) initial dam state

Using this initial configuration and the inflow hydrographs, we carry out different analyses. The first one checks that the model (PROMELA plus embedded C code) does not end in invalid states. For this analysis there are no constraints over the basin or the dam, thus SPIN explores all the possible execution branches produced by the non-deterministic behaviour of the management rule model. The analysis ends without errors, and we have obtained 15 different manoeuvre sets for this episode.

The following analyses include constraints to synthesize specific manoeuvres. To this end, we configure Spin to analyze the system plus a never claim, and to stop when the first error occurs. Constraints can be defined over dam parameters and the basin flows, in an independent or combined way. The objective of the second analysis is to limit the outflow of GH-GT and CGH to under 310 m^3/s, and the flow at the four locations to under 310 m^3/s. Figure 11 shows the never claim used to describe these constraints. The analysis ends with an error, which means that there is at least one set of manoeuvres that satisfies the constraint. Figure 3 shows the evolution of dam parameters, the flow downstream in different locations, and the manoeuvres of the different outlets. In this case, the spillway of the GH-GT dam remains closed, and the other gates are opened at different degrees over time.

```
#define inv0 c_expr{t <= 0}
#define inv1 c_expr{outflow_c < 310 && outflow_gg < 310 && t <= 3600}
#define guard0 c_expr{t == 0}
#define guard1 c_expr{t == 3600}
never {
st0:
  if
  :: (guard0 && inv0 && inv1)->goto st1
  :: (inv0)->goto st0
  :: (Inicio[0]@l_init)->goto st0
  fi;
st1:
  if
  :: (guard1 && inv1)->goto accept_st2
  :: (inv1)->goto st1
  fi;
accept_st2:
  if
  :: (1) -> skip
  fi
}
```

Fig. 11. Never claim for constant constraints

The last analysis uses variable constraints to synthesize manoeuvres. There are two ways of defining variable constraints over dam parameters. The first approach is to define constraints as curves that define the upper and lower bounds of the parameters. These curves are stored in UnMatched C structures. The never claim is modified to compare the parameter with the curves. For example, the definition of `inv1` in Fig. 11 can be modified to check that `outflow_c` is always under the curve stored in `curve[0]` as follows:

```
#define inv1 e_expr{outflow_c< curve[0][t]}
```

The second approach is to define constraints as a timed automaton that represents sequences of intervals. This approach does not require C structures, which reduces the memory and time required. The timed automaton is transformed into a never claim, as explained in Sect. 5. We use this approach to restrict the level of CGH dam at four different time intervals. Figure 12 shows the timed automaton that represent the variable constraint. The analysis ends with an

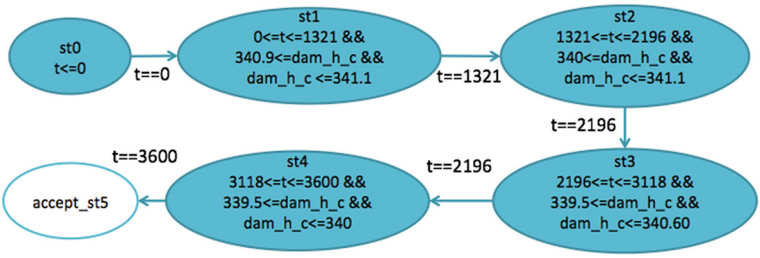


Fig. 12. Timed automaton for variable constraints

Table 2. SPIN statistics

	Invalid end state	Const. constraint	Var. constraint
Depth	421573	432403	432393
States stored	102530	10834	12018
States matched	1	0	1
Atomic steps	3989109	421565	467584
Memory usage (MB)			
For states	17.133	2.104	2.300
For hash table	2.000	2.000	2.000
For DFS stack	26.703	26.703	26.703
Other(proc and chan stacks)	29.821	30.127	30.127
Total memory	75.773	61.027	61.222
Time (sec)			
Total elapsed time	175	19.2	22.4
External model	89.6	9.1	10.6

error that corresponds to the manoeuvres, which are very similar to the previous ones. In this case, the CGH spillway is completely open in two steps, while in the previous analysis, it is opened in three steps. Since the spillways are the gates with greatest discharge capacity, this small change has a great influence on constraint satisfaction.

Table 2 shows the statistics of SPIN for each analysis. Note that the state space is fairly small, this is thanks to the use of UnMatched C variables and the abstraction of outlet states described in Sect. 4.1. The time elapsed in each analysis depends on the calls to the external model. We have measured the execution time of the external model to determine how much time is spent on these calls. Observe that the depth in the second and third analysis has increased, because of the interleaved execution of the PROMELA model and the never claim that defines the constraints. Finally, note that the number of matched states is 0 or 1, which means that there are no repeated states. This is mainly because of a global timer in the PROMELA model, which is defined as a C Matched variable

that counts the number of minutes of the flood episode. In addition, when SPIN backtracks to a state, the management rule model operates the dam outlets in a different way, which causes a different evolution of the other model variables.

7 Conclusions and Future Work

We have provided a complete case study to show how the SPIN model checker can be a central part of future DSSs to help in mitigating the effects of floods. The methodology to generate the dam and management rule models, which exports a reduced number of C variables into SPIN's state, and reduces the interleaving of the different process, makes the approach effective enough regarding to both the effort to write the PROMELA models for each specific river basin and also to the time needed to synthesize the appropriate manoeuvres. Since the simulators for hydrologic models are integrated as a black box, more accurate versions of such simulators can be easily integrated. This novel application domain opens the use of the SPIN model checker as a central component of (commercial) DSSs demanded by the authorities that manage big dams in many countries. This is a real need identified in the current European Research Project SAID (Smart wAter management with Integrated DSSs) [1]. In the final stage of the project the DSS will be fully operative, and the dam manager will evaluate the quality of synthesized manoeuvres and the time required.

The work could be further extended to introduce additional optimisation when there are many dams in cascade in the same basin. We are also working on a different way of building the models in order to exploit parallel execution of SPIN for very complex river basins.

References

1. SAID Project 12 Feb 2015. http://www.said-project.eu
2. Ahmad, S., Simonovic, S.: An intelligent decision support system for management of floods. Water Resour. Manage. **20**, 391–410 (2006)
3. Alur, R., Dill, D.: The theory of timed automata. In: Huizing, C., de Bakker, J.W., Rozenberg, G., de Roever, W.-P. (eds.) REX 1991. LNCS, vol. 600, pp. 45–73. Springer, Heidelberg (1992)
4. Cheng, C.T., Chau, K.W.: Flood control management system for reservoirs. Environ. Model. Softw. **19**(12), 1141–1150 (2004)
5. Díaz, M., Soler, E., Romero, S., Gallardo, M.M., Merino, P., Panizo, L., Salmerón, A.: Technical specification of the DSS for flood management. Deliverable 1.3, SAID Project (2015)
6. Gallardo, M.M., Merino, P., Panizo, L., Linares, A.: Developing a decision support tool for dam management with spin. In: Alpuente, M., Cook, B., Joubert, C. (eds.) FMICS 2009. LNCS, vol. 5825, pp. 210–212. Springer, Heidelberg (2009)
7. Gallardo, M.M., Merino, P., Panizo, L., Linares, A.: A practical use of model checking for synthesis: generating a dam controller for flood management. Softw. Pract. Experience **41**(11), 1329–1347 (2011)
8. Holzmann, G.: The SPIN Model Checker: Primer and Reference Manual. Addison-Wesley Professional, Reading (2003)

9. Karbowski, A.: Fc-ros - decision support system for reservoir operators during flood. Environ. Softw. **6**(1), 11–15 (1991)
10. Labadie, J.W.: Optimal operation of multireservoir systems: state-of-the-art review. J. Water Resour. Plan. Manage. **130**(2), 93–111 (2004)
11. McCartney, M.P.: Decision support systems for dam planning and operation in Africa. International Water Managment Institute, Colombo (2007)
12. Panizo, L., Gallardo, M.M., Merino, P., Sanán, D., Linares, A.: Dam management based on model checking techniques. In: 8th International Conference on Software Engineering and Formal Methods. SEFM 2010: Proceedings of the Posters and Tooldemo Session, pp. 9–13. CNR, Pisa, Italy, Sept. 2010
13. Polo, M., Herrero, J., Aguilar, C., Millares, A., Moñino, A., Nieto, S., Losada, M.: Wimmed, a distributed physically-based watershed model (i): Description and validation. Environmental Hydraulics: Theoretical, Experimental & Computational Solutions, pp. 225–228 (2010)
14. Pottinger, L.: A Flood of Dam Safety Problems, 8 Sept. 2010. https://www.internationalrivers.org/resources/a-flood-of-dam-safety-problems-1700