VAWKUM Transactions on Computer Sciences
http://vfast.org/journals/index.php/VTCS@ 2021, ISSN(e): 2308-8168, ISSN(p): 2411-6335
Volume 9, Number 1, January-December , 2021 pp: 34-46

Specification of Operating Rules for Water Reservoir to Manage Flood Using Z-Notation

Hasrat Alam

Lecturer, Department of Computer Sciences, Government College Shaikh Maltoon Mardan, Pakistan
*Corresponding author email address: hasrat.cs@gmail.com

ABSTRACT

Flood is natural event; it brings a lot of destruction. Due to flood not only loss of human life and misery of millions of people occur, but tremendous damages occur in public and private property. It cannot be stopped, but can be managed to save the lives and property of the people. To manage a flood, there are two solutions, physical and logical. In physical solutions, different physical constructions are made to manage flood like levee, channel improvement, flood ways, and widening of barrages, but these control works have many deficiencies. In logical solutions, models and techniques are developed to manage the flood, however, these systems also have some deficiencies. To overcome the deficiencies in the existing models, we focus on to develop a flood management model consisting of an off-river reservoir and diverted canals having regulators. Reservoirs are most effective water storage that smooth down extreme inflow. The optimal operations of reservoirs determine the release and accumulation of water over time. For effective operative decision, operating rules will be defined in Z-Specification. For these operating rules, we will develop an algorithm, will verify the rules in Z-Notation and will implement them in the Java programming language.

KEYWORDS

Formal Modeling, Z-Notation, Flood Management, Operating rules

JOURNAL INFO

HISTORY: Received: August 17, 2021 Accepted: September 14, 2021 Published: December 05, 2021

INTRODUCTION

Flood is natural event; it brings a lot of destruction. Due to flood not only loss of human life and misery of millions of people occur, but tremendous damages occur in public and private property. It is the main problem of many countries of the world, including Pakistan, Saudi Arabia, China, Japan, Turkey, Malaysia, India and Bulgaria (Iqbal, et al., 2014). It cannot be stopped, but can be managed to save the lives and property of the people. In Pakistan the financial and social damages of the flood of 2010 in different sectors were tremendous. Social infrastructure got damaged and loss of 115,451 million of Pakistani rupees has occurred. Similarly, in physical infrastructure loss of 102,469 million, in economic sector loss of 330,120 million, and in the cross cutting sector a loss of 4133 million Pakistani rupees occurred (Qadir, 2010). It is conceded that flood cannot be avoided, but can be managed in time to secure people area and infrastructure.

To minimize the damages and to manage flood, methods and techniques of flood management is a challenge for many countries of the world who regularly facing it. The countries like Malaysia, China, USA, and Canada are working to manage it (Hsu & Wei, 2007) (Luis, et al., 2013) (Cheng & Chau, 2004). If a flood is managed in time, then a lot of losses can be minimized.

Flood is a corollary of sudden changes in climate and earth through which it is passing. It is conceded that flood cannot be avoided, but can be controlled in time to secure people. To manage flood there are two solutions physical and logical. In physical solution, a lot of physical control structures are developed like levee, channel improvement, flood ways and widening of barrages. In other solutions, techniques and models are used to manage flood to reduce the disasters of flood.

System Dynamics Model is a model which presents a dynamic interactions of different components in the flood management system. The operation of the reservoir and floodways is simulated and to minimize flooding operating rules are defined. The analysis of flood management policy is done by System Dynamics. System Dynamics is as modeling paradigm based on feedback simulation approach (Ahmad & Simonovic, 2000). However, the used simulation techniques cannot directly generate an optimal solution related to the reservoir operation problem. Secondly, flood routing becomes too complex by System Dynamics.

IMSFCR, Integrated Management System for Flood Control of Reservoir, is used in China. In the study, for high flows of year operating rules are developed and by changing reservoir capacity and outflow alternative rules are explored. Operation of the reservoir and flood ways is simulated by system dynamics. The main challenges to this system are system complexity, interface integration, and standardization of software (Cheng & Chau, 2004).

Reservoir Real-Time Operation is another flood management model. This model determines optimal real time release. The model consists of three sub-models to predict rainfall, reservoir inflow and to forecast reservoir operation optimization. The reservoir operation optimization model releases reservoir release hydrograph (Hsu & Wei, 2007). But the problem with this model is that hydrograph usually uses a graph plot discharge which simply provides discharge data and has no decision power.

Ten-Stage Operation Policies for Routing of Flood, determines the gate opening rules based on current pool level, which classifies flood into ten groups. It routs the flood based on ten fixed rules (Haktanir & Kisi, 2001). In this study, the decision of flood routing is made without predicting the magnitude of the incoming flood which incorporates risk in decision making process.

In another study of reservoirs, rule curves are used for flood regulation and other purposes. Rule curves are used for relating gate opening and reservoir state (Jain, et al., 1998). The problem with rule curve is that its performance is weak under moderate flood. Secondly, it does not depend on external data of reservoirs (Valdes & Marco, 1995).

RFFCS, Reservoir Flood Forecasting and Control System, is a software development system which was initiated in January 1995 and has been applied to several hydropower reservoirs in China during the period 1995–1998 (Guo, et al., 2004). The parameters for flood forecasting models uses historical data, while this software system uses data from the hydrological telemetering system. So, there is a difference in reliability and data source.

(Li, et al., 2010) described a model consists of three modules: pre-release module, second one is a refill operation module is used to retain recession flood, third one is a risk analysis module, which is used to assess flood risk. The dynamic control bound of the reservoir is estimated by using Monte Carlo simulation. (Chen, et al., 2010) Used reservoir operation and management for seasonal design floods, which considers dates of flood occurrence and magnitudes of the peaks (runoff), based on copula function. Further, this model was used in Geheyan Reservoir, China. The methods satisfied the flood prevention standards, however, it cannot be applied to every reservoir due its physical and its prediction requirements.

In logical solutions, models and techniques are developed to manage the flood, however, these systems also have some deficiencies (Luis, et al., 2013) (Ahmad & Simonovic, 2000). To overcome the deficiencies in the existing models, we focus on to develop a flood management model consisting of an off-river reservoir and diverted canals having regulators. Reservoirs are most effective water storage that smooth down extreme inflow. The optimal operations of reservoirs determine the release and accumulation of water over time. For effective operative decision, operating rules will be defined in Z-Specification. For these operating rules, we will develop an algorithm, will verify the rules in Z-Notation and will implement them in the Java programming language.

RESEARCH METHODOLOGY

We propose the proper operation management model of reservoirs during a flood situation which will make precise and timely decisions based on flood prediction received. We divide our propose model into two parts; diverted canals having regulators and Reservoir. In our proposed model the off-river reservoir system is used which is constructed off the main course of the river. The reservoir will get the water through diversion canals having regulators at the start. On the prediction of flood, the flood water can’t hold by the river will be diverted to the reservoir through regulators at the diversion canal, but before diverting the extra water to the reservoir, the level/capacity of the reservoir will be checked. For diverting the extra water and checking the capacity of the reservoir, the system will use operating rules. The proposed work is a safety critical and monetary model, whose failure can cause human and financial losses, therefore we propose z notation for the specification and verification of model to develop a mathematical definition of the objective functions. The Z notation is a specification language based upon set theory and mathematical logic (Bowen, 1988). On the prediction of flood our model will take decisions using operating rules. Based on decision of the model either water will be diverted to reservoir or the spillway will be opened to make space for water and then divert the water.

To make the system practical, we have shown our conceptual model in figure 4.1. If there is the prediction of flooding in the river A or B or in both and there is a place for water in the reservoir, the extra water will be allowed through regulators at point C on river A and/or point D on river B. If there is a flood, but there is no space for water in the reservoir then gates of the spillway, at points G and/or H, will be opened to discharge the water from the reservoir and then the extra water from river A and/or B will be diverted to the reservoir. For the above operations to work properly, we have defined operating rules for the proposed model.

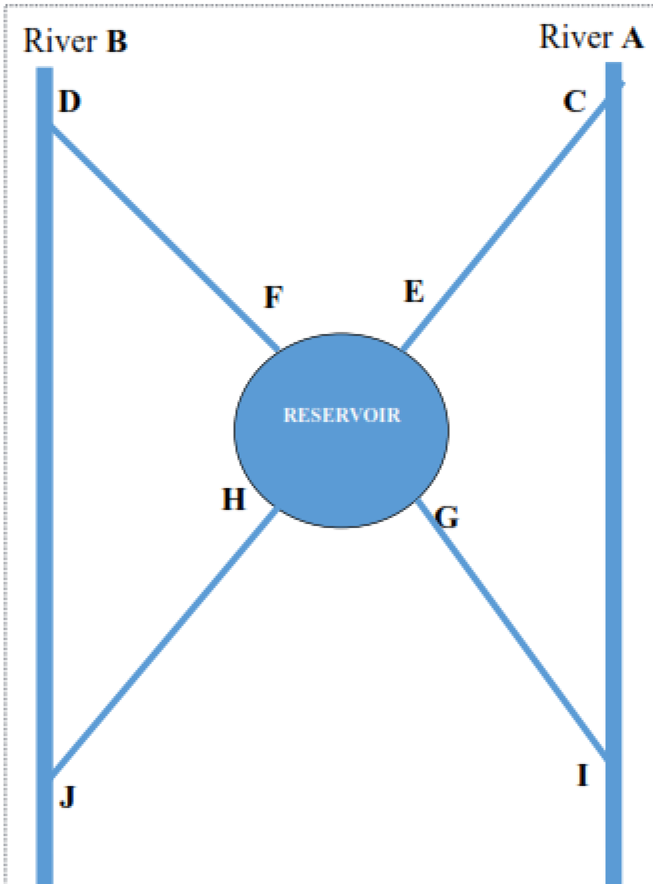

Figure 4.1: Off-River Reservoir (our Conceptual Model)

SAFETY PROPERTIES AND ITS SPECIFICATION

Reservoirs are an important water facility which needs a series of decisions for the release and accumulation of flood water. Following are the properties defined for the proposed model.

IF: Inflow (flow of river)
F1: High Flood
F2: Very High Flood
F3: Exceptionally High Flood
L0: DSL (Dead Storage Level) minimum pool level
L1: NCL (Normal Conversation Level) conservation pool level of reservoir
L2: FRL (Full Reservoir Level i.e. top of the conservation level)
L3: Flood Control Pool
L4: MWL (Maximum Water Level)

1. If IF in river A is F1 and the level of the reservoir is Less or equal to L2, then open regulator of diverting canal on the river A at point C, when the level of reservoir reaches to L4 close the regulator at point C.
2. If IF in river A is F1 and the level of the reservoir is greater than L2, then open the spillway gate at H, when the level reaches to L0 close spillway gate H and open regulator at point C, when the level reaches to L4 close the regulator at point C.
3. If IF in river A is F2 or F3 and level of the reservoir is greater or equal to L1, then open the spillway gate at H, when the level reaches to L0 close spillway gate H and open regulator at point C, when the level of reservoir reaches to L4 close regulator at point C.

FORMAL SPECIFICATION OF MODEL

Here the actual coding of the rules given above is given in the Z notation along with the explanation.

Gate* ::= *open* | *close

Gate is an attribute which can take only two possible values, i.e. on and off.

A schema is a structure that specifies the relationship between the variable values. A state schema specifies snapshot of the system. In the top part of schema variables are typed and declared. In the bottom part of schema the possible values of the declared variables restrain a predicate (axioms).

Reservoir

normal, high, very_high, exceptionally_high: $\mathbb{N}$
flood_type_A, flood_type_B: $\mathbb{N}$
$L_0, L_1, L_2, L_3, L_4: \mathbb{N}$
reservoir_level: $\mathbb{N}$
regulator_C, regulator_D, spillway_G, spillway_H: Gate

$L_4 > L_3 > L_2 > L_1 > L_0 > 0$

Reservoir is the base schema of the system which shows the static view of the reservoir model actually it presents the snapshot of the model. In declaration part, first it declares four variables, normal, high, very_high and exceptionally_high of natural number. These variables represent the type of flood in either river A or river B. It also declares flood_type_A and flood_type_B of natural numbers which shows the type of flood in river A and river B and their values are taken from the above four types of flood. Further, it declares five variables L_0, L_1, L_2, L_3 and L_4 of natural numbers, which represents the four levels of reservoirs i.e. Dead Storage Level (DSL), Normal Conservation Level (NCL), Full Reservoir Level (FRL), Flood Control Pool (FCP) and Maximum Water Level (MWL) respectively. It also declares reservoir_level of natural number whose value will be compared to the above four levels of the reservoir. At last it declares regulator_C, regulator_D, spillway_G and spillway_H of type Gate.

In predicate part, it only checks that the value of L_4 is greater than L_3, L_3 is greater than L_2, L_2 is greater than L_1, L_1 is greater than L_0 and L_0 is greater than 0.

Reservoir_init

Reservoir

---

flood_type_A = normal
$\wedge$ ***flood_type_B = normal***
$\wedge$ ***reservoir_level =*** L_0
$\wedge$ ***regulator_C = close***
$\wedge$ ***regulator_D = close***
$\wedge$ ***spillway_G = close***
$\wedge$ ***spillway_H = close***

To specify the initial state of reservoir model, we specify *Reservoir_init* as the initial schema of *Reservoir* Schema. *Reservoir_init* includes *Reservoir* schema in declaration part which means that in declaration part included schema is added to new schema and predicate part of included schema is conjoined to the predicate part of the new schema. In declaration part, it initializes flood_type_A, flood_type_B to normal, reservoir_level to L_0, and regulator_C (regulator at point C), regulator_D (regulator at point D), spillway_G (spillway at point G) and spillway_H (spillway at point H) to close. That is initially both rivers A and B are having normal floods, and the gates of regulators and spillways are closed.

The state schema Reservoir gives the static view of the system. In order to specify how the system can change we require to specify operation schema. An operation is an instance of state schema which produces a new instance.

Reservoir_Status

ΔReservoir
read_A?,read_B?:ℕ
read_RL?:ℕ

flood_type_A = read_A? ∧ flood_type_B = read_B? ∧ reservoir_level = read_RL?

Reservoir_Status is the operational schema of our model in which Δ*Reservoir* schema is included and it gives the current read values to the attributes of Reservoir schema. Δ*Reservoir* means that the value of the attributes of *Reservoir* will be changed. *Reservoir_Status* includes *Reservoir* schema in declaration part which means that in declaration part included schema is added to new schema and predicate part of included schema is conjoined to the predicate part of the new schema. In the declaration part it declares two variables read_A? and read_B? of natural numbers, which reads the current status of flood in river A and river B respectively. It also declares read_RL of natural number to read the current level of water in the reservoir. In predicate part it assigns read_A to flood_type_A, read_B to flood_type_B and read_RL to reservoir_level.

Rule_1

ΔReservoir_Status

flood_type_A = high ∧ reservoir_level $\boldsymbol{\leqslant L_2}$ ***⇒ regulator_C = open***
reservoir_level = $\boldsymbol{L_4}$ ***⇒ regulator_C = close***

Rule_1 is the operation schema for the first rule of our model in which Δ*Reservoir_Status* schema is included in declaration part because it will change the current status of the reservoir. In predicate part there is a pre-condition for the inflow into the reservoir. It checks if the magnitude of flood in the river A is high (*flood_type_A = high*) and the level of water in the reservoir is less than or equal to L_2 (*reservoir_level* $\leqslant L_2$), it will open the regulator at point C (*regulator_C = open)* to allow inflow of water to the reservoir at point E from river A. When the level of reservoir reaches to L_4 (*reservoir_level* $= L_4$), it will close the regulator at point C (*regulator_C = close*) because the water in the reservoir has reached to its maximum level.

Rule_2

ΔReservoir_Status

flood_type_A = high ∧ reservoir_level $\boldsymbol{> L_2}$ ***⇒ spillway_H = open***
reservoir_level = $\boldsymbol{L_0}$ ***⇒ spillway_H = close ∧ regulator_C = open***
reservoir_level = $\boldsymbol{L_4}$ ***⇒ regulator_C = close***

Rule_2 is the operation schema for the second rule of our model which specifies how the system can change. In predicate part it checks if the magnitude of flood in river A is high (*flood_type_A = high*) and level of the reservoir is greater than L_2 *(reservoir_level* $> L_2$), it will open the spillway at point H (*spillway_H = open*) to divert some water of the reservoir towards river B to make a free space for water in the reservoir. When the level of the reservoir reaches to L_0 (*reservoir_level* $= L_0$), it will close the spillway at point H (*spillway_H = close*) and open the regulator at point C (*regulator_C = open)* to allow inflow of water to the reservoir at point E. When the level of reservoir reaches to L_4 (*reservoir_level* $= L_4$), it will close the regulator at point C (*regulator_C = close*).

Rule_3

ΔReservoir_Status

flood_type_A ⩾ very_high ∧ reservoir_level $\boldsymbol{\geqslant L_1}$ ***⇒ spillway_H = open***
reservoir_level = $\boldsymbol{L_0}$ ***⇒ spillway_H = close ∧ regulator_C = open***
reservoir_level = $\boldsymbol{L_4}$ ***⇒ regulator_C = close***

Rule_3 is the operation schema for the third rule of our model. In predicate part it checks if the magnitude of flood in river A is very_high or exceptionally_high (*flood_type_A ⩾ very_high*) and the level of the reservoir is greater than or equal to

L_1 (*reservoir_level* ⩾ L_1), it will open the spillway at point H (*spillway_H = open*) to divert some water of the reservoir towards river B. When the level of the reservoir reaches to L_0 (*reservoir_level* = L_0), it will close the spillway at point H (*spillway_H = close*) and open the regulator at point C (*regulator_C = open)* to allow inflow of water to the reservoir at point E. When the level of reservoir reaches to L_4 (*reservoir_level* = L_4), it will close the regulator at point C (*regulator_C = close*).

VERIFICATION AND SPECIFICATION OF FLOOD MANAGEMENT MODEL USING Z NOTATION

In this section, the proposed model is verified using Z-Notation. Reservoir model is specified and a property of Z, schema and auxiliary variables are used to prove that the reservoir model satisfies the property, i.e. inflow into the reservoir is equal to the outflow from the reservoir. Based on auxiliary variables and schemas the proof theorems are developed and applied on every rule.

General Proof Theorem of the Model

The proposed work is a safety critical and monetary model, whose failure can cause human and financial losses. Therefore, the model is verified using Z-Notation. The Z-Notation is a formal modeling tool based on mathematical logic, functions, relations, and set-theory (Gamble, 1995). Two proof theorems for inflow and outflow to the model is developed by using auxiliary variables and schemas. The results show that in any case inflow into the reservoir is equal to the outflow from the reservoir and then every operation rule of the model is verified by it.

[*CUSEC*]

It is the possible amount of flood water that could ever be stored or transmitted.

Suppose CUSEC = {$cusec_1$, $cusec_2$}, where $cusec_1$ is the inflow into the reservoir from river A through regulator C and $cusec_2$ is the inflow into the reservoir from river B through regulator D. Similarly, in case of outflow suppose $cusec_1$ is outflow to River A through spillway G and $cusec_2$ is outflow to River B through spillway H.

| *mx*: ℕ

It is the constant maximum amount of water that can be held in the reservoir at any one time.

```
┌─ Reservoir ──────────────────────────
│ unitFlood: seq CUSEC
├──────────────
│ # unitFlood ⩽ mx
└──────────────────────────────────────
```

Reservoir is the state schema which represents the snapshot and static view of the reservoir. In the declaration part it declares unitFlood which is a sequence of CUSEC. UnitFlood is the amount of water which is present at any one time in the reservoir. To ensure the safe operation of reservoir the unitFlood must be less than or equal to mx.

The state schema reservoir gives the static view of the reservoir. To show how the system can change we have to specify the operation schema which is an instance of state schema. To specify an operation, the relationship between the instance of the state before and after the operation are expressed as predicate. We have adopted the convention that before the operation the values of state variables are denoted by unprimed identifiers and after the operation the values of state variables are denoted by primed identifiers. Similarly variable_name? shows input to the system and variable_name! shows output of the system. Operations of the reservoir are atomic.

For the reservoir model, there are two operations

Inflow: The flood water is added to the reservoir from the river through regulators.

Outflow: The flood water leaves the reservoir through spillways.

As we have a “before” and “after” instance for any operation we make the syntactic simplification.

```
┌─ ΔReservoir ─────────────────────────
│ unitFlood, unitFlood': seq CUSEC
├──────────────
│ # unitFlood ⩽ mx
│ # unitFlood' ⩽ mx
└──────────────────────────────────────
```

$\Delta Reservoir$ is the operation schema which will change the current status of the reservoir. It declares two sequences unitFlood and unitFlood' of type CUSEC which shows before and after instances. In predicate part, it ensures that unitFlood and unitFlood' must be less than or equal to mx. mx is the constant maximum limit of water in the reservoir at any one time, which ensures that water stored in reservoir before and after operation does not exceed the maximum capacity of the reservoir.

Inflow

$\Delta Reservoir$
$cusec_1?, cusec_2?: CUSEC$

$\# unitFlood < mx$
$unitFlood' = unitFlood ^\frown \langle cusec_1? \rangle ^\frown \langle cusec_2? \rangle$

Inflow is the operation schema in which $\Delta Reservoir$ schema is included in declaration part. By doing this, the declaration part of the included schema ($\Delta Reservoir$) is added to new schema and the predicate part of included schema is conjoined to the predicate part of the new schema.

Further, in the declaration part, it declares two input variables $cusec_1?$ and $cusec_2?$ of type CUSEC, which represents the inflow into the reservoir from the river A and river B respectively. In predicate part the first line ($\# unitFlood < mx$) is a precondition for the inflow operation. It shows that for the inflow operation to be possible the water held by the reservoir must be less than the maximum capacity of the reservoir. The last line of the predicate shows the relationship between the reservoir water before and after the operation. In case of inflow into reservoir, either from river A or river B or from both, the unitFlood (water held by reservoir before inflow operation) is updated with $cusec_1$ and/or $cusec_2$ and unitFlood changes to $unitFlood'$ (water held by reservoir after inflow operation).

Outflow

$\Delta Reservoir$
$cusec_1!, cusec_2!: CUSEC$

$unitFlood \neq \emptyset$
$unitFlood = \langle cusec_1! \rangle ^\frown \langle cusec_2! \rangle ^\frown unitFlood'$

Outflow is another operation schema in which ΔReservoir schema is included. In declaration part, it declares two output variables $cusec_1!$ and $cusec_2!$ of type CUSEC. In predicate part, the first line shows the precondition for the outflow operation, which shows that for the outflow operation to be possible, the reservoir must not be empty. The last line of the predicate shows the relationship between the water of reservoir before and after operation. After the outflow operation, the $unitFlood$ (water contained in reservoir before the operation) is equal to $\langle cusec_1! \rangle ^\frown \langle cusec_2! \rangle$ (water transported through spillways) and $unitFlood'$ (water contained in the reservoir after the operation).

Reservoir_INIT

$Reservoir$

$unitFlood = \langle\rangle$

For the completion of reservoir specification, we specify the initial state of the reservoir. *Reservoir_INIT* is the initial schema of reservoir which includes only *Reservoir* schema in its declaration part. In predicate part it initializes $unitFlood$ with $\langle\rangle$ which means that initially the reservoir is empty and then operations inflow and outflow can occur when their preconditions are satisfied.

This reservoir model, as specified, has the property that in any case inflow into the reservoir is equal to the outflow from the reservoir. To prove this property, we need to introduce auxiliary variables, *inhist and outhist.* These variables record the history of the flow of flood water into and out of the reservoir model.

$$\begin{array}{|l}\hline \textit{Recorded\_Reservoir} \\ \textit{Reservoir} \\ \textit{inhist}: \text{seq } \textit{CUSEC} \\ \textit{outhist}: \text{seq } \textit{CUSEC} \\ \hline \end{array}$$

Recorded_Reservoir is the operation schema, in which *Reservoir* schema is included and it also declares two sequences inhist (in history) and outhist (out history) of type CUSEC. Inhist records the history of inflow and outhist records the history of outflow from reservoir.

$$\begin{array}{|l}\hline \textit{Recorded\_Reservoir\_INIT} \\ \textit{Recorded\_Reservoir} \\ \textit{Reservoir\_INIT} \\ \hline \textit{inhist} = \langle\rangle \\ \textit{outhist} = \langle\rangle \\ \hline \end{array}$$

Recorded_Reservoir_INIT is the initial schema of *Recorded_Reservoir* schema which shows that initially the inhist and outhist of the reservoir is empty.

$$\begin{array}{|l}\hline \textit{Recorded\_Inflow} \\ \Delta \textit{Recorded\_Reservoir} \\ \textit{Inflow} \\ \hline \textit{inhist}' = \textit{inhist} ^\frown \langle \textit{cusec}_1? \rangle ^\frown \langle \textit{cusec}_2? \rangle \\ \textit{outhist}' = \textit{outhist} \\ \hline \end{array}$$

Recorded_Inflow is the operation schema, which includes Δ*Recorded_Reservoir and Inflow* schemas in its declaration part. In predicate part it updates *inhist* which means that when *Inflow* occurs *inhist* is updated by $cusec_1?$ and/or $cusec_2?$ and *inhist* changes to *inhist'*. In case of *Inflow* there would be no change in *outhist*.

$$\begin{array}{|l}\hline \textit{Recorded\_Outflow} \\ \Delta \textit{Recorded\_Reservoir} \\ \textit{Outflow} \\ \hline \textit{inhist}' = \textit{inhist} \\ \textit{outhist}' = \textit{outhist} ^\frown \langle \textit{cusec}_1! \rangle ^\frown \langle \textit{cusec}_2! \rangle \\ \hline \end{array}$$

Recorded_Outflow is the operation schema which includes Δ*Recorded_Reservoir and Outflow* schemas in its declaration part. In predicate part, it updates *outhist* which means that when *Outflow* from reservoir occurs, *outhist* is updated by $cusec_1!$ and/or $cusec_2!$ and outhist changes to *outhist'*. In case of *Outflow* there would be no change in *inhist*.

The auxiliary variables are used to prove that the reservoir model satisfies the property that in any case inflow into the reservoir is equal to the outflow from the reservoir.

GENERAL PROOF THEOREM FOR INFLOW

$$\forall \textit{Recorded\_Reservoir} \bullet \textit{inhist} = \textit{outhist} ^\frown \textit{unitFlood}$$

Proof:

Using Structural Induction

Initially $\textit{inhist} = \textit{outhist} = \textit{unitFlood} = \langle\ \rangle$

So predicate is true.

Suppose the predicate is true and *Recorded_Inflow* occurs.
After the operation

$$inhist' = inhist ^\frown \langle cu\sec_1? \rangle ^\frown \langle cu\sec_2? \rangle$$

$$outhist' = outhist$$

$$unitFlood' = unitFlood ^\frown \langle cu\sec_1? \rangle ^\frown \langle cu\sec_2? \rangle$$

Hence:

$$inhist' = inhist ^\frown \langle cu\sec_1? \rangle ^\frown \langle cu\sec_2? \rangle$$

$$= \left(outhist ^\frown unitFlood \right) ^\frown \langle cu\sec_1? \rangle ^\frown \langle cu\sec_2? \rangle$$

$$By\ applying\ Associative\ law,\ we\ get$$

$$= outhist ^\frown \left(unitFlood ^\frown \langle cu\sec_1? \rangle ^\frown \langle cu\sec_2? \rangle \right)$$

$$inhist' = outhist' ^\frown unitFlood'$$

Hence Proved.

General proof theorem for the outflow

$$\forall Recorded\_Reservoir \bullet inhist = outhist ^\frown unitFlood$$

Proof:

Using Structural Induction

Initially $inhist = outhist = unitFlood = \langle\ \rangle$

So predicate is true.

Suppose the predicate is true and *Recorded_Outflow* occurs.
After the operation

$$inhist' = inhist$$

$$outhist' = outhist\ \hat{}\ \langle cu\sec_1! \rangle ^\frown \langle cu\sec_2! \rangle$$

$$unitFlood = \langle cu\sec_1! \rangle ^\frown \langle cu\sec_2! \rangle ^\frown unitFlood'$$

Hence:

$$inhist' = inhist$$

$$= outhist ^\frown unitFlood$$

$$= outhist ^\frown \left(\langle cu\sec_1! \rangle ^\frown \langle cu\sec_2! \rangle ^\frown unitFlood' \right)$$

$$By\ applying\ Associative\ law,\ we\ get$$

$$= \left(outhist\ \hat{}\ \langle cu\sec_1! \rangle ^\frown \langle cu\sec_2! \rangle \right) ^\frown unitFlood'$$

$$inhist' = outhist'\ \hat{}\ unitFlood'$$

PROOF OF INDIVIDUAL RULES BY USING GENERAL PROOF THEOREMS

Now we show that all the operating rules of the model can be proved by general proof theorems for inflow and outflow.

PROOF OF RULE 1

INFLOW THEOREM FOR RULE 1

$$\forall Recorded\_Reservoir \bullet inhist = outhist ^\frown unitFlood$$

PROOF:

Using Structural Induction

Initially $inhist = outhist = unitFlood = \langle \ \rangle$

So predicate is true.

Suppose the predicate is true and Recorded_Inflow occurs.

After the operation

$$inhist' = inhist \frown \langle cu\sec_1? \rangle \frown \langle \phi ? \rangle$$

$$outhist' = outhist$$

$$unitFlood' = unitFlood \frown \langle cu\sec_1? \rangle \frown \langle \phi ? \rangle$$

Hence:

$$inhist' = inhist \frown \langle cu\sec_1? \rangle \frown \langle \phi ? \rangle$$

$$= \left(outhist \frown unitFlood \right) \frown \langle cu\sec_1? \rangle \frown \langle \phi ? \rangle$$

$$By\ applying\ Associative\ law,\ we\ get$$

$$= outhist \frown \left(unitFlood \frown \langle cu\sec_1? \rangle \frown \langle \phi ? \rangle \right)$$

$$= outhist' \frown unitFlood'$$

As there is only inflow from river A, so $cusec_2$ is ϕ.

OUTFLOW THEOREM FOR RULE 1

There is no outflow from reservoir in case of Rule 1, so only inflow theorem is proved.

PROOF OF RULE 2

INFLOW THEOREM FOR RULE 2

Proof of Inflow theorem for Rule 2 is same as that of Rule 1.

OUTFLOW THEOREM FOR RULE 2

$$\forall Recorded\_Reservoir \bullet inhist = outhist \frown unitFlood$$

Proof:

Using Structural Induction

Initially $inhist = outhist = unitFlood = \langle \ \rangle$

So predicate is true.

Suppose the predicate is true and Recorded_Outflow occurs.

After the operation

$$inhist' = inhist$$

$$outhist' = outhist \hat{} \langle \phi ! \rangle \frown \langle cu\sec_2! \rangle$$

$$unitFlood = \langle \phi ! \rangle \frown \langle cu\sec_2! \rangle \frown unitFlood'$$

Hence:

$$
\begin{aligned}
inhist' &= inhist \\
&= outhist \frown unitFlood \\
&= outhist \frown \left(\langle \phi ! \rangle \frown \langle cu\sec_2 ! \rangle \frown unitFlood' \right) \\
& \textit{By applying Associative law, we get} \\
&= \left(outhist \frown \langle \phi ! \rangle \frown \langle cu\sec_2 ! \rangle \right) \frown unitFlood' \\
inhist' &= outhist' \frown unitFlood'
\end{aligned}
$$

As there is only outflow through spillway H therefore $cusec_1$ is ϕ

PROOF OF RULE 3

INFLOW THEOREM FOR RULE 3

Proof of Inflow theorem for Rule 3 is same as that of Rule 1.

OUTFLOW THEOREM FOR RULE 3

Proof of Outflow theorem for Rule 3 is same as that of Rule 2.

CONCLUSION

Flood is natural disaster, it brings a lot of destruction. Efficient flood management *techniques* can help in securing people, land, and infrastructure. In recent era the existing flood management techniques are not taking the benefits of technology advancement. We propose a model based concept for flood management. In the initial phase, a conceptual model is developed with formal specification and its varification. The model will make precise and timely decisions based on flood prediction received. In the second phase, the procedures will contain properties of opening and closing various gates of reservoirs to release/storage, plans to control or divert the flood. In the work three out of ten Operating rules of the model are developed in the if-then-else form, then these rules are specified and verified in Z/Specification language. In the third phase, the algorithm of the model is developed that shows the behavior and working of the model.

REFERENCES:

[1]. Ahmad, S. & Simonovic, S. P., 2000. *Dynamic modeling of flood management policies.* Bergen, Norway, In Proceedings of the 18th International Conference of the System Dynamics Society: Sustainability in the Third Millennium, pp. 6-10.

[2]. Ahmad, S. & Slobodan, P. S., 2000. System dynamics modeling of reservoir operations for flood management. *Journal of Computing in Civil Engineering,* 7, 14(3), pp. 190-198.

[3]. Allen, P., 2009. *Operation of spillway gates—how to avoid the problems and pitfalls.* Caloundra, Peter Allen, 34th Annual Qld Water Industry Operations Workshop, pp. 98-105.

[4]. Asif, R. S. I., 2012. *Analysis and Design of Flood Prediction Model using Mobile Agents,* Lahore: Roa Sohail Iqbal Asif, University of Central Punjab.

[5]. Bakker, M., Baker, R., Meinzen-Dick, R. & Konradsen, F., 1999. *Multiple Uses of Water in Irrigated Areas: A Case Study from Sri Lanka,* Colombo: International Water Management Institute.

[6]. Barlas, Y., 2007. *System Dynamics: Systemic Feedback Modeling for Policy Analysis.* 1st ed. Istanbul: Boğaziçi University.

[7]. Bartlett, R. E., 1981. *Surface Water Sewerage.* [Online]
Available at: http://trid.trb.org/view.aspx?id=165264#
[Accessed 05 02 2015].

[8]. Batuca, D. G. & Jordaan, J. J. M., 2000. General Consideration about Reservoirs. In: *Silting and Desilting of Reservoirs.* Rotterdam: A. A. Balkema Publishers, pp. 11-19.

[9]. Bowen, P. J., 1988. *Formal specification in Z as a design and documentation tool.* Livepool, IET, pp. 164-168.

[10]. Bowles, D. S., Mathias, J. D., Chauhan, S. S. & Countryman, J. D., 2004. *Reservoir Release Forecast Model for Flood Operation of the Folsom Project Including Pre-Releases.* Denver, United States Society On Dams.

[11]. Chang, F. J. & Chang, Y. T., 2006. Adaptive neuro-fuzzy inference system for prediction of water level in reservoir. *Advances in Water Resources,* 29(1), pp. 1-10.
[12]. Cheng, C. T. & Chau, K. W., 2004. Flood control management system for reservoirs. *Environmental Modelling & Software*, 19(12), p. 1141–1150.
[13]. Chen, L. et al., 2010. A new seasonal design flood method based on bivariate joint distribution of flood magnitude and date of occurrence. *Hydrological Sciences Journal–Journal des Sciences Hydrologiques,* 55(8), pp. 1264-1280.
[14]. Cigizoglu, H. K. et al., 2007. *Artificial Neural Network Models in Rainfall-Runoff Modelling of Turkish Rivers.* Turcia, Scientific and Technical Research Council of Turkey (TÜBİTAK), pp. 560-571.
[15]. Fang, L. et al., 2010. *A flood emergency response system based on flood disaster evaluation models.* Beijing, IEEE, pp. 1-4.
[16]. Firat, M., 2007. Artificial intelligence techniques for river flow forecastint in the seyhan river catchement. *Hydrology and Earth System Sciences (HESS),* 4(3), pp. 1369-1406.
[17]. Gamble, R. F., 1995. Integrating a formal specification course with a software projects course via an editing tool. *ACM SIGCSE Bulletin*, 27(1), pp. 312-316.
[18]. Ghazali, J. N. & Kamisn, A., 2008. *A Real Time Simulation and Modeling of Flood Hazard.* Heraklion, WSEAS International Conference. Proceedings. Mathematics and Computers in Science and Engineering (No. 12). WSEAS., pp. 438-443.
[19]. Golian, S., Moazami, S., Yazdi, J. & Sheshangosht, S., 2010. *An IFM-based flood damage reduction: A small study area in Iran.* Kyoto, IEEE.
[20]. Guo, S. et al., 2004. A reservoir flood forecasting and control system for China/Un système chinois de prévision et de contrôle de crue en barrage. *Hydrological sciences journal,* 49(6).
[21]. Haktanir, T. & Kisi, Ö., 2001. Ten-stage discrete flood routing for dams having gated spillways. *Journal of Hydrologic Engineering,* 1, 6(1), pp. 86-90.
[22]. Hsu, N. S. & Wei, C. C., 2007. A multipurpose reservoir real-time operation model for flood control during typhoon invasion. *Journal of Hydrology,* 7 4, 336(3), pp. 282-293.
[23]. Iqbal, R. S. et al., 2013. A Mobile Agent-Based Algorithm for Prediction of Inundation Area. *Research Journal of Recent Sciences,* January, 3(1), pp. 72-77.
[24]. Jain, S. K., Goel, M. K. & Agarwal, P. K., 1998. Reservoir operation studies of Sabarmati system, India. *Journal of water resources planning and management*, 124(1), pp. 31-37.
[25]. Javadi, S., Kiapasha, M. S. & Mohammadi, K., 2009. Object Oriented Simulation; Its Application in Water Reservoir Management and Operation. *Journal of Agricultural Science and Technology,* Volume 11, pp. 331-340.
[26]. Jin, Q., Chen, J., Wang, H. & Zhao, S., 2010. *Research on Regional Flood Disaster Risk Assessment Based on PCA and BP Neural Network.* Nanjing, IEEE.
[27]. Kang, L. & Guo, X., 2010. *Research on application of cross structure flood risk assessment decision support system using Bayesian Network.* Chengdu, IEEE.
[28]. Lee, S., 2011. *Formal Methods in Software Enginering.* Kuala Lumpur: FTMS Consultants (M) Sdn Bhd Kuala Lumpur, Malaysia.
[29]. Liaqat, W., 2012. *Types of Spillways.* [Online]
Available at: http://www.slideshare.net/waleedliaqat/spillways
[Accessed 31 08 2015].
[30]. Lin, Z. & Qiang, Q., 2010. *Notice of Retraction The Research on Problem of Flood Insurance Pricing Based on the DFA in PR China.* Wuhan, IEEE, pp. 1-7.
[31]. Li, X., Guo, S., Liu, P. & Chen, G., 2010. Dynamic control of flood limited water level for reservoir operation by considering inflow uncertainty. *Journal of hydrology,* 391(1), pp. 124-132.
[32]. Luk, S.-h. & Whitney, J., 1993. *MEGAPROJECT A Case Study of China's three Gorges Project,* London: 3 Henrietta Street, London WC2E 8LU.
[33]. Matei, A. M., 2008. An analysis of modern monitoring and preventing techniques of floods. *Mathematics-Informatics-Physics Series,* 60(2), pp. 86-92.
[34]. Naliki, J. A., 2014. *Design of a multipurpose reservoir in Nzeveni area Makueni County.* [Online]
Available at: http://erepository.uonbi.ac.ke/handle/11295/78579
[Accessed 21 08 2015].
[35]. Noori, M., Othman, F., Sharifi, M. B. & Heydari, M., 2013. *Multi Objective Operation Optimization of Reservoirs Using Genetic Algorithm.* Singapore, IACSIT Press.
[36]. Perry, C. A., 1993. *Effects of Reservoirs on Flood Discharges in the Kansas and the Missouri River Basins,* Denver: United States Government Printing Office.

[37]. Potter, B., Till, D. & Sinclair, J., 1996. *An Introduction to Formal Specification ans Z.* USA: Prentice Hall PTR Upper Saddle river,NJ,USA.

[38]. Qadir, C. G., 2010. *Flood 2010 in Pakistan and its damages.* Lahore, Director / PRO (Hydraulics) Irrigation Research Institute, Punjab Irrigation and Power Department, Lahore., pp. 197-219.

[39]. Sawunyama, T., 2005. *Estimation of small reservoir storage capabilities in Limpopo river basin using geographical information system and remote sensed surface areas.* [Online] Available at: http://www.smallreservoirs.org/full/publications/reports/UZ-Sawunyama-MSc-thesis-0605.pdf [Accessed 05 01 2015].

[40]. Sengupta, S. K. et al., 2006. *Flood Forecasting and Inundation Mapping in the Mahanadi River Basin,* Odisha: A Collaborative Effort between India and the United States.

[41]. Shao-fei, L., 2011. *Assessment of flood hazard risk based on catastrophe theory in flood detention basins.* Lushan, IEEE, pp. 139-142.

[42]. Siddiqui, I., 2008. *Dams and Reservoirs: Planning Engineering.* 1st ed. Oxford: Oxford University Press.

[43]. Spivey, J. M., 1992. *The Z Notation.* 2nd ed. UK: Prentic Hall international (UK) Ltd.

[44]. Tariq, A., 2012. *Dam and Reservoir Engineering.* 1st ed. Punjav: Punjab Irrigation Agriculture Investment Program (PIAIP).

[45]. Tariq, A., 2012. Dam Spillways. In: A. Tariq, ed. *Dam and Reservoir Engineering.* Lahore: Punjab Irrigation Agriculture Investment Program (PIAIP), pp. 381-495.

[46]. Tariq, A., 2012. Introduction. In: A. Tariq, ed. *Dam and Resevoir Engineering.* Lahore: Punjab Irrigation Agriculture Investment Program (PIAIP), pp. 1-48.

[47]. Thornton, J., Steel, A. & Rast, W., 1996. Reservoirs. In: D. Chapman, ed. *Water Quality Assesments – Guide to Use of Biota, Sediments and Water in Environmental Monitoring.* 2nd ed. Cambridge: E&FN Spon, an imprint of Chapman & Hall, pp. 371-411.

[48]. Tomac, J., 2014. *Design and Evaluation of an Irrigation Storage Reservoir,* Jimmy Tomac, Polytechnic State University: California.

[49]. Valdes, J. B. & Marco, J. B., 1995. *Managing reservoirs for flood control.* Perugia, Italy, In Proceedings of the US–Italy Research Workshop on the Hydrometeorology, Impacts, and Management of Extreme Floods.

[50]. Woodcock, J. & Davies, J., 1996. *Using Z.* 1st ed. Oxford: University of Oxford.

[51]. Wu, C. et al., 2010. *Set pair analysis and BP neural network coupling model for optimal selection of flood control operation.* Xuzhou, IEEE, pp. 2238-2243.

[52]. Wurbs, R. A., 1991. *Optimization of Multiple-Purpose Reservoir System Operations: A Review of Modeling and Analysis Approaches,* Davis: US Army Corps of Engineers Hydrologic Engineering Center.

[53]. Yuliang, Z., Ping, Z., Juliang, J. & Libing, Z., 2010. *Application of Interpolation Model Based on Genetic Algorithm to Comprehensive Evaluation of Flood Disaster Loss.* Wuhan, IEEE, pp. 272-275.

[54]. Zhang, X. Z., Jing, R. L. & Tang, K. D., 2011. *Evaluation for flood control influence on Zhangjiazhuang (Cihe River) bridge.* Xi'an, IEEE, pp. 474-477.