VFAST Transactions on Software Engineering
http://vfast.org/journals/index.php/VTESS@ 2015 ISSN(e): 2309-3978; ISSN(p): 2411-6246
Volume 8, Number 1,September-October , 2015

A MOBILE-AGENT ENVIRONMENT FOR SERVICE ORIENTED SYSTEM USING STRONG MOBILITY

RAO SOHAIL IQBAL[1], SHAKEEL AHMAD[2*], SHER AFZAL KHAN[2]

[1]Department of Computer Science, GC University, Faisalabad Pakistan
[2*]Faculty of Computing and Information Technology in Rabigh (FCITR), King Abdul Aziz University
Jeddah, Saudi Arabia
Emails: raosohailiqbal@ucp.edu.pk, sarahmad@kau.edu.sa, Sakhan2@kau.edu.sa

Received January, 2015, Revised May 2015 and Published September 2015

Abstract- In this research, we extended our existing work by applying the concept of strong mobility of mobile agent to make service oriented systems. Initially, an algorithm is developed on the model, based on strong mobility between the clients and server using mobile agent technology via VSAT (very small aperture technology). Where clients are deployed in critical regions which calculate flood discharge based on speed of flood. The proposed algorithm has been verified through the mobile Petri net, a formal language, which is used to formalize it. Reachability tree method of analysis is adopted to verify this model. For practical implementation, a critical area ‘Kalabagh’ is selected and a Java based system is developed of this model. The system is verified by using the historical data values of this critical area and the results verify the model.

Keywords— flood monitoring; service oriented system, strong mobility; mobile agent; Petri net

1. Introduction: Floods are inevitable calamities and there are no chances to prevent them, but only prediction can help to secure the people of inundation area by eviction before time. Some techniques are developed to analyze and predict flood. Most famous technique is the Hydrograph, it uses a graph plots discharge on a daily basis [1]. This is a simple system that has no decision power of prediction except provides information on discharge. Unsteady flow river model is another technique based on software use to calculate discharge by taking different parameters as input like flow and capacity of the river. It calculates only the discharge and does not predict. SPH (Smoothed Particle Hydrodynamics), a simulation base model uses to measure the flood through Kula Lumpur. This model uses GIS (Geographical Information System) with LIDAR (Light Detection and Ranging) and DEM (Digital Elevation Model) to simulate the flood area [1]. For the graphic representation of flood it uses the graphic tool MAYA. However, the model is used only for the historical data, not for real time simulation of flood. Adjoint state theory is a mathematical model used for prediction and assessment of the reliability parameters like catchment and gradient-based search [7]. This model mainly focuses on the calibration of parameters and helps as an initial step to predict flood. FRICS (Foundation of River Basin Integrated Communications) is developed to predict flood before time. This gathers data from 700 different critical points and alerts the inhabitants of inundation area through the website, cell phones and mobile internet for the preparation of evacuation [6]. Often, floods destroy the infrastructure which can disconnect cell phone and internet services, so it is not practical to rely on it. ANFAS- Data Fusion for Flood Analysis and Decision Support System is simulating software with an interactive interface to the user. Users are allowed to put a different securing construction in real images and take impacts of it [1]. It uses heavy graphics and works for limited flood area, it requires heavy bandwidth and processing, which cannot be predicted in limited time which is one of the drawbacks of this method. A project DMS (Disaster Management Support) starts with collaboration of GOI (Government of India) and USAID (United States Agency for International Development) for flood forecasting at Mahanandi River basin based on topography information. This only provides topography information which has insufficient resolution, which failed it [1]. A collaborative effort was done by Turkey and Bulgaria in predicting flood. They developed a system

HydraHMD, running on the internet and manual as well. This system only share data without any prediction technique [4].

Inhabitants of inundation area can impart valuable information to evacuate through a prediction system. The importance of the system can be seen from the above discussion; however, we observed problems like limited routing capability, heavy graphics, low resolution and huge network traffic as explain above.

In the third phase, we developed an algorithm which can predict based on the discharge values of more than one client agent. This model based on strong mobility to predict the flood at any critical area. Strong mobility is a type of mobility to transmit code, data and state of the agent. This system is based on one server along with client agents distributed over the critical regions of Kabul, Tunisia, Chasma, Kalabagh rivers, etc. The server predicts the inundation area based on the information it obtained from the clients. In this phase, we take the most critical region Kalabagh, located on the intersection of two big rivers the Indus and the Kabul. The server predicts, based on the discharge of rivers calculated by the respective client agents. Further, on the basis of collective discharge the server can alert the other client agents located in the upcoming critical areas situated throughout the Indus River. This alert is generated based on the discharge of flood with the historical records of critical areas. The development of the system is initiated by the algorithm with strong mobility. In this request mobile agent from the sever, executes Client 1, takes discharge from it. Further, it moves to another client if required and so on. Finally, the mobile agent is received by the server. It executes the server agent and calculates the decision based on information extracted.

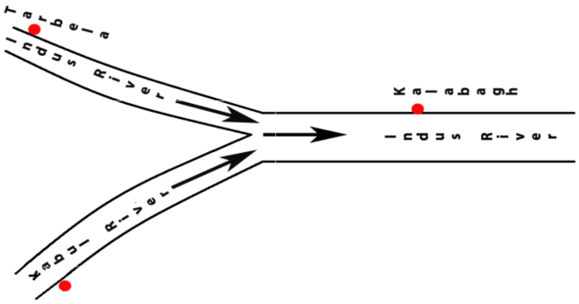


Fig. 1. Kabul River and Tarbela

The algorithm is then specified and verified by the Petri net and reachability tree. For proving it, a Java based server and client agents are implemented. Finally, a generalized mobile Petri net model is developed.

2. Conceptual Model: In this research work, we proposed a solution for the most critical area of Pakistan known as 'Kalabagh' that is a merge client agent. Flood destroys the Kalabagh brutally, because it depends upon two rivers Tarbela (Indus) and Kabul River, as shown in Fig. 1. Here the focus is to predict flood at Kalabagh for this purpose, we take two client mobile agents and one server mobile agent for prediction of flood. The first client mobile agent is situated at Kabul River and the second client at Tarbela. Both clients MA in the initial step of execution, calculate area of particular locations, through which food is passing. Due to the irregular shape of rivers, area calculation is not simple enough as depicted in the Fig. 2. Client MA calculates area with the help of Simpson's 1/3 formula and then discharge.

A. *Components of Client Mobile Agent*

Level of water: As described above, for calculating the area of a river, it requires length and width. Width can be measured easily, but length is not simply due to its irregular shape. To cater this, we divide it into different lengths in a river, getting more precise data for decision. This component is responsible for measuring levels of water.

Speed of water: On how much speed water is passing from a river, can be calculated with the help of this component.

Discharge: This component takes values calculated by above components. As we know

$$Discharge = Area * speed$$

It is responsible for calculating discharge of the river at a particular client.

Security: Making client code secure from intrusion is a major responsibility of this component.

B. *Components of Server Mobile Agent*

Security: This component is securing code from external attacks for making decisions efficiently and without eruptions. When agents are communicating then making it free from malicious code is also its duty.

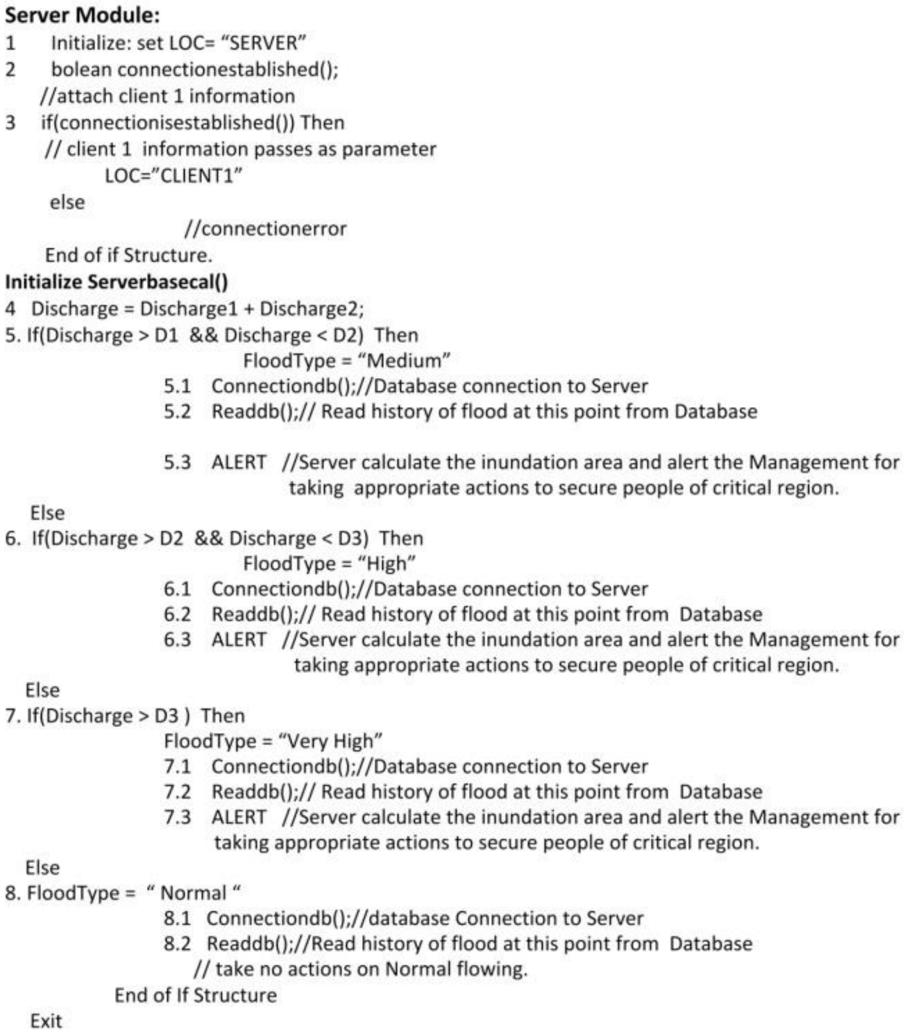
User Messages: For making model interactive and responsive to impart user is the aim of this component.

Database: This is a database of events according to flood inundation. This component helps the server, mobile agent in taking the decision of prediction and calculating inundation area in case of critical situation.

Taking Decision: The major component in flood prediction is taking decision. It stores the discharge values coming from different client mobile agents and process it. It classifies the flood in different categories like normal, medium, high and very high. Except normal, all are flooding situations and require to calculate the inundation area for securing the inhabitants of that area. In prediction, database component also helps as a source of input for decision component.

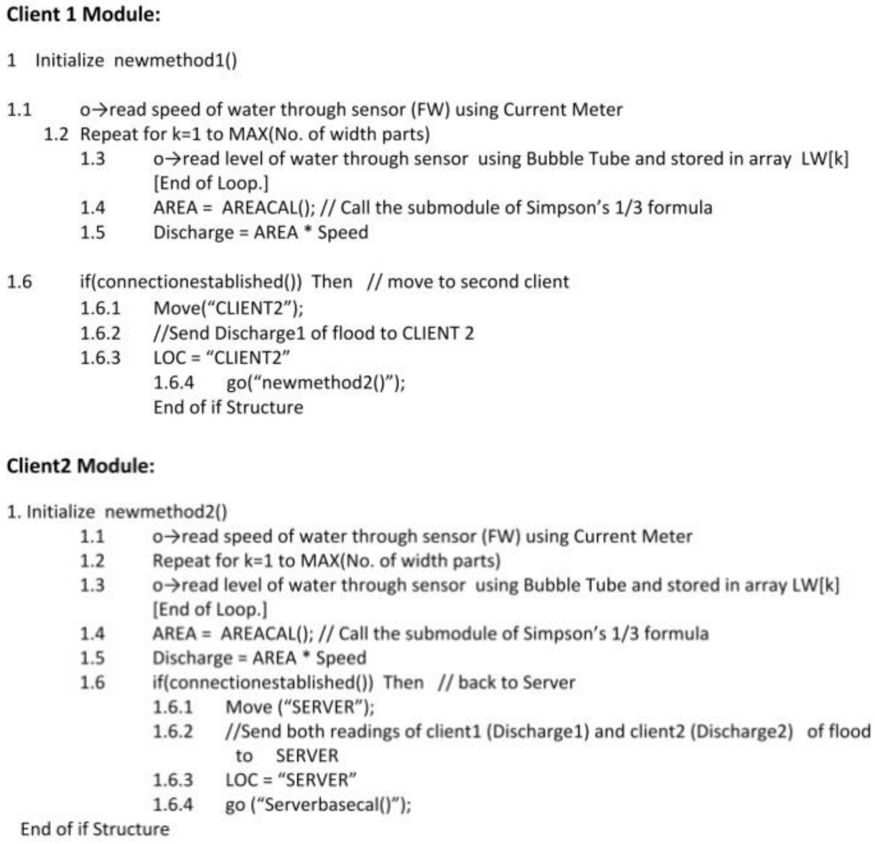
Alarm: In medium, high and very high situation, the major focus of our model is to evacuate the inundation area. To appraise higher management and local administrative staff about calculating the inundation area for securing inhabitants is the main responsibility of this component.

2. Algorithum of Computational Model: This algorithm shows the working and behavior of our development model. In this study, develop a model with strong mobility of mobile agents. The mobile agent can be classified into two categories strong and weak mobility. In weak mobility, it migrates agent's data and code, but rather than its, strong mobility transmits code, data and state of the agent. The reusability of code is reduced in strong mobility, which has elegant support to design different applications.



```
Server Module:
1    Initialize: set LOC= "SERVER"
2    bolean connectionestablished();
   //attach client 1 information
3   if(connectionisestablished()) Then
    // client 1  information passes as parameter
          LOC="CLIENT1"
     else
                    //connectionerror
    End of if Structure.
Initialize Serverbasecal()
4  Discharge = Discharge1 + Discharge2;
5. If(Discharge > D1  && Discharge < D2)  Then
                          FloodType = "Medium"
                    5.1   Connectiondb();//Database connection to Server
                    5.2   Readdb();// Read history of flood at this point from Database

                    5.3   ALERT   //Server calculate the inundation area and alert the Management for
                                  taking  appropriate actions to secure people of critical region.
   Else
6.  If(Discharge > D2  && Discharge < D3)  Then
                          FloodType = "High"
                    6.1   Connectiondb();//Database connection to Server
                    6.2   Readdb();// Read history of flood at this point from  Database
                    6.3   ALERT   //Server calculate the inundation area and alert the Management for
                                  taking appropriate actions to secure people of critical region.
  Else
7. If(Discharge > D3 )  Then
                    FloodType = "Very High"
                    7.1   Connectiondb();//Database connection to Server
                    7.2   Readdb();// Read history of flood at this point from  Database
                    7.3   ALERT   //Server calculate the inundation area and alert the Management for
                          taking appropriate actions to secure people of critical region.
  Else
8. FloodType =  " Normal "
                    8.1  Connectiondb();//database Connection to Server
                    8.2  Readdb();//Read history of flood at this point from  Database
                        // take no actions on Normal flowing.
              End of If Structure
   Exit
```



Client 1 Module:

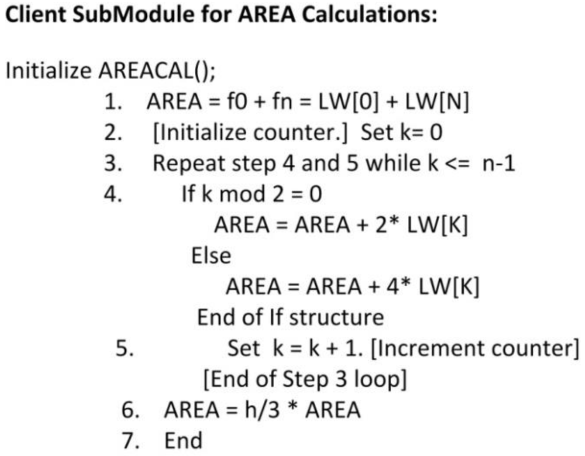
```
1   Initialize  newmethod1()

1.1       o→read speed of water through sensor (FW) using Current Meter
     1.2  Repeat for k=1 to MAX(No. of width parts)
          1.3     o→read level of water through sensor  using Bubble Tube and stored in array  LW[k]
                  [End of Loop.]
          1.4     AREA =  AREACAL(); // Call the submodule of Simpson's 1/3 formula
          1.5     Discharge = AREA * Speed

1.6       if(connectionestablished())  Then   // move to second client
          1.6.1   Move("CLIENT2");
          1.6.2   //Send Discharge1 of flood to CLIENT 2
          1.6.3   LOC = "CLIENT2"
                  1.6.4     go("newmethod2()");
                  End of if Structure
```

Client2 Module:

```
1. Initialize  newmethod2()
          1.1     o→read speed of water through sensor (FW) using Current Meter
          1.2     Repeat for k=1 to MAX(No. of width parts)
          1.3     o→read level of water through sensor  using Bubble Tube and stored in array LW[k]
                  [End of Loop.]
          1.4     AREA =  AREACAL(); // Call the submodule of Simpson's 1/3 formula
          1.5     Discharge = AREA * Speed
          1.6     if(connectionestablished())  Then   // back to Server
                  1.6.1     Move ("SERVER");
                  1.6.2     //Send both readings of client1 (Discharge1) and client2 (Discharge2)   of flood
                              to    SERVER
                  1.6.3     LOC = "SERVER"
                  1.6.4     go ("Serverbasecal()");
  End of if Structure
```



Client SubModule for AREA Calculations:

```
Initialize AREACAL();
          1.   AREA = f0 + fn = LW[0] + LW[N]
          2.    [Initialize counter.]  Set k= 0
          3.    Repeat step 4 and 5 while k <=  n-1
          4.        If k mod 2 = 0
                         AREA = AREA + 2* LW[K]
                     Else
                          AREA = AREA + 4* LW[K]
                     End of If structure
           5.             Set  k = k + 1. [Increment counter]
                      [End of Step 3 loop]
            6.   AREA = h/3 * AREA
            7.   End
```

The execution starts with the creation of mobile agent at the server node by setting the location of mobile agent to the server. In server module, few lines of code execute and move to client A, as shown at line third of server module. Now code will move to client A, it resumes execution from newmethod1( ) as shown in client A module. Client A calculates discharge with the help of Simpson's 1/3 formula as a sub module to calculate area. After complete execution of client A, mobile agent moves from client A to client B with calculated discharge information of client A. Because this algorithm is based on strong mobility then it moves from server to client A and then client A to client B, it also transfers state as well. State describes where the execution stops, stores it and restarts it from that point of suspension. Now client B will accept the rest of code with discharge information of client A and state. Client B will also calculate the discharge at a particular location like client A mobile agent. At the end of execution of client B, it will move back to server with remaining code and calculated discharge information of both clients. Now server will execute the rest of code for prediction, depending upon information extracted from both clients. Server mobile agent finds the flood's type as described in algorithm, i.e. normal, medium, high and very high, depends upon the values of D1, D2 and D3. The conditions of D1, D2, and D3 depend upon the river area. It's quite possible that D1 value is normal in one river, but medium on another river. Here, we select Indus River as a practical example and values of D1, D2, and D3 are 200,000, 400,000 and 600,000 respectively. The values may change according to the river, where it will be implemented.

Simpson's 1/3 approach to calculate the Area of Cross Section: Prediction depends upon the calculated discharge values. A discharge is a result of area (river) and velocity (of water). Area calculation is not easy because of the irregular shape of the river, as shown in Fig. 2. In our model, we used Simpson's 1/3 formula for calculating area shown in the Simpson's section of the algorithm. River width is divided into fix sizes. Corresponding to each width, length of a river is measured. Mostly lengths are not the same because of sedimentation and natural shape of the river. This formula used in our developed model, it helps to calculate maximum accurate discharge. That enables to predict the flood properly.

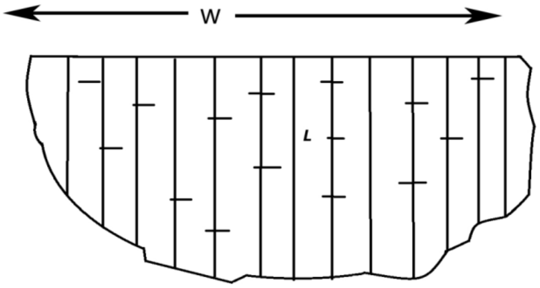


Fig. 2. Showing Irregular Shape of River Base

3. Specification and Verification of the System: In the Fig.3 the mobile Petri net MPN is constructed which shows the process of communication of the server agent with the client agents based on mobile agent approach. In the MPN, the marked place and the transition represent the sequence of operations. The transitions denote starting or finishing of the operations while places represent the operations and resource types. The interpretation of places and transitions are given in Table 1.

In the specification the server and the clients shown in the figure 6.5 consists of three mobile Petri nets.

P_s = {p1, p2, p3, p4, p5} is the server places and this = {t1, t2, t10} are the server transitions. The specification of first client mobile Petri net containing

P_{c1} = {p6, p7, p12, p13, p14, p15, p17}, T_{c1} = {t3, t7, t8, t9} and M_{cc} = {p7}.

The specification of second client mobile Petri net containing

P_{c2}= {p8, p9, p10, p11, p16} and T_{c2} = {t4, t5, t6}.

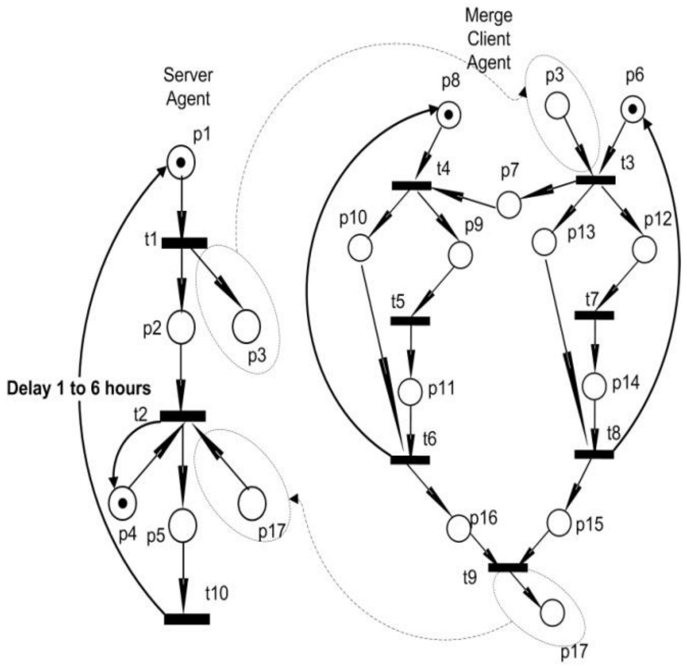


Fig. 3 Specification of the model

In Fig. 6.5 p1 represents server, whereas p6 and p8 are clients. When the server is required to check the status of the first client, it dispatches the mobile agent {p3}. This describes the request of the server to know the current status of the first client. In the real sense the data of a flood are usually required with the interval of one to six hours, where one hour is the critical condition and six hours in the normal situation. The mobile agent {p3} of the server enables the transition t3 of the first client of the merge agent. This further activates the activities denoted by p13 and p12. First client also creates mobile agent {p7} to initiate the second client.

Now both clients execute independently. The p13 reads the flood speed and p12 the flood levels. Moreover, the condition p12 enables the transition t7 uses the level of a flood to calculate the cross-section area of the region denoted by p14 through which the flood is passing. The p9 in second client works same like p12 and p10 works like p13. The P11 is the cross-section area of the second client agent. The transition t8 calculates the discharge of the first client denoted by p15 whereas; t6 calculates the discharge of the second client denoted by p16. The transition t9 calculates the total discharge of merged client by summing up p15 and p16 and enables the server transition t2 by p17. The activity p4 represents the existing database of the flood and disaster caused. The transition t2 activates p5, provide the prediction of the transition of the inundation areas of the merged client. Finally, p5 enables the transition t6 which re-initiates the server after the fixed interval varying from 1 to 6 hours, depends on the situation of a flood. This is designed by the special arc with delay option.

Table 1 Description of Places and Transitions

S#	Places/Transition	Description
1	P1	Server Agent
2	P2	Client Execution at Server
3	P3	Request to Client
4	P4	Flood Database
5	P5	Prediction of Flood
6	P6, p8	Client Agents
7	P7	Request to second client
8	P9,912	Read the Level of flood
9	P10, p13	Read the Speed of flood
10	P11,p14	Area of the Cross-section
11	P15, p16	Discharge of flood
12	P17	Total Discharge of Clients
13	t1	Initiation of Server
14	t2	Prediction by Server
15	t3, t4	Reading the Flood
16	t5, t7	Calculating the Area
17	t6, t8	Calculating the Discharge
18	t9	Calculate total Discharge
19	t 10	Initiation of the Server after the fixed interval

A. *Verification of the System*

To check whether there exists a one-to-one functional correspondence between the Petri net model and the original requirements specification. For this we use the reachability tree which is extremely useful for the analysis of PNs. This method involves the enumeration of all reachable markings or their coverable markings. The Fig. 4 shows the reachability tree of the model and each marking in the tree lists the places holding a token. The reachability tree represents the reachable states of a PN from initial marking p1+p4+p6+ p8.

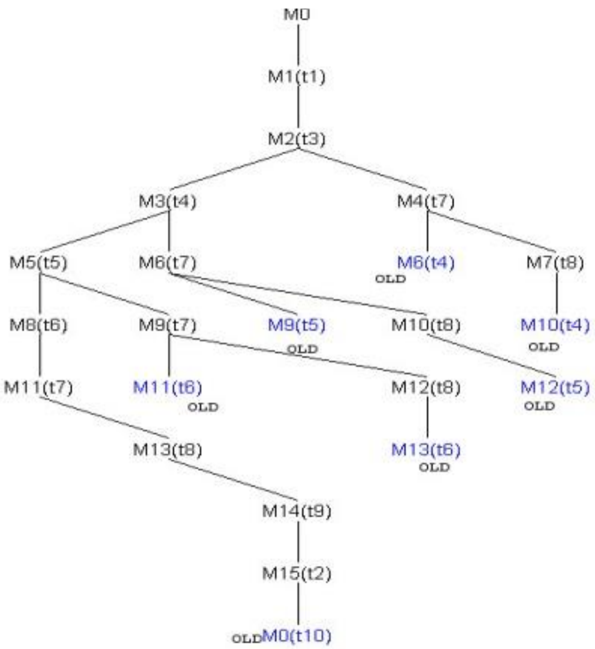


Fig. 4 Reachability Tree

Table 2. Description of the reachability process

Mi	Marking	Description
M0	P1+P4+P6+P8	Initiation of server
M1	p2+p3+p4+p6+p8	Initiation of merge client
M2	p2+p4+p7+p8+p12+p13	Initiation of sub-merge client or calculation of area at merge client
M3	p2+p4+p9+p10+p12+p13	Discharge of merge client or discharge of sub-merge client
M4	P2+p4+p7+p8+p13+p14	Initialization of sub-merge client or calculate discharge of merge client
M5	P2+p4+p10+p11+p12+p13	Calculation of discharge of sub-merge client or cross-section area of merge client
M6	P2+p4+p9+p10+p13+p14	Calculate cross-section area of sub-merge client or calculate discharge of merge client
M7	P2+p4+p6+p7+p8+p15	Initiation of sub-merge client
M8	P2+p4+p8+p12+p13+p16	Calculate cross-section area of merge client
M9	P2+p4+p10+p11+p13+p14	Calculate the discharge of sub-merge client or calculate the discharge of merge client
M10	P2+p4+p6+p9+p10+p15	Calculate the cross-section area of sub-merge client
M11	P2+p4+p8+p13+p14+p16	Calculate the discharge of merge client
M12	**P2+p4+p6+p10+p11+p15**	Calculate the discharge of sub-merge client
M13	**P2+p4+p6+p8+p15+p16**	Calculate the total discharge of both client
M14	**P2+p4+p6+p8+p17**	Prediction of disaster area
M15	**P2+p5+p6+p8**	Initiation of server after fixed interval

In the tree every node represents markings generated and edge from one marking to another represents the transition fired. Moreover, from the Table 2 it is depicted that each new marking enables single transition and leads to single new marking. It is clear from the tree that the final marking p_5 is reachable from their respective initial marking which represents the passing message from the server to the client and then it ends at the prediction of inundation areas. Every marking includes the set of places holding a token showing that the modeled system is safe and the system is under control. The final marking represents the successful operation of the server and client, which predicts at the server. Moreover, every marking enables a transition, showing the absence of deadlock in the system.

B. Practical Implementation of Model

A system is developed to simulate the flood situation in Pakistan. It is developed using a Java, a language, one of the popular languages. This language is preferred because of its versatility i.e. it supports process management, graphics manipulation. The system includes the client mobile agents, shown in figure 6.3 with different color square brackets on rivers, distributed on the critical points where the flood is passing with a single server mobile agent, placed in a secure place. The server creates mobile agent and transfers it towards clients. Where clients accept mobile agents, calculate the discharge and return back to the server. The server calculates the flood and predicts whether flow at a particular location will be normal or critical.

The developed system has some characteristics.

C. Visualization and Interactive Interface

It has interactive interface for user to understand the current situation and is easy to use. Different visual tools are used to enhance the understandability of the system because the system is critical and visual effects have more power to describe the situation. The software provides the visual effects of any critical area with the help of graphics and discharge map. It presents the flood situation with different colors, like sky blue for normal, dark blue for medium, orange for high and red for very high situations.

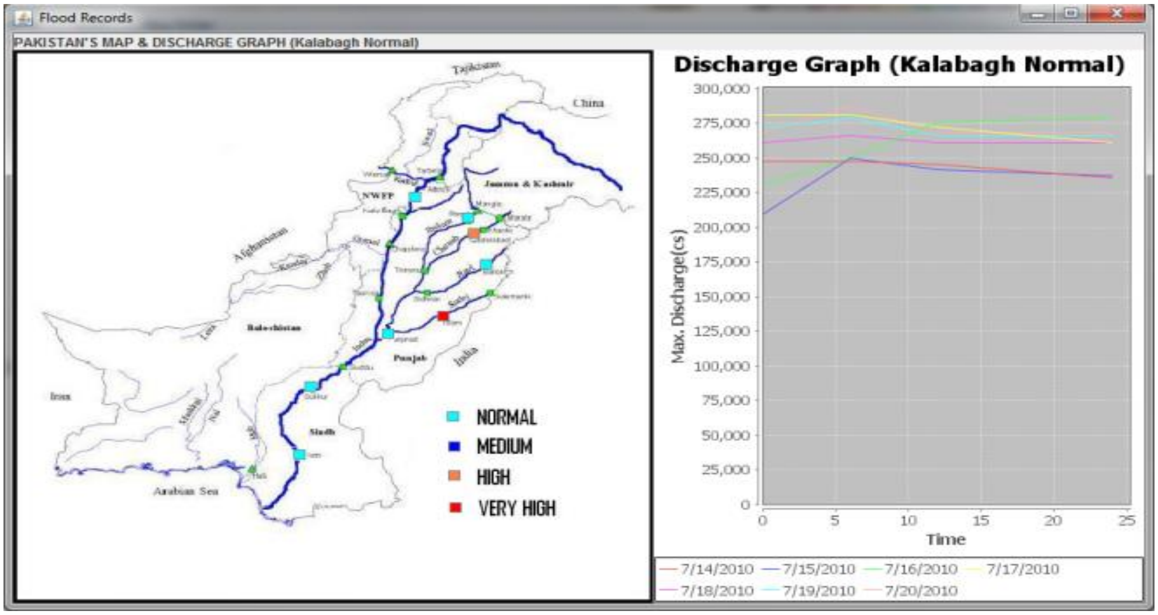


Fig. 5. Normal Discharge of Kalabagh

D. Presenting Critical Areas

Client mobile agents are fixed on critical areas represented by color square brackets, and vary their colors according to the situation. Every client mobile agent executes the mobile agent and transfers current values to server. The server predicts either the client is in normal or in a critical situation. If user, click on particular client mobile agent, server assesses the flood situation and shows it in graphical representation, client current situation by a color and its historical data in tabular form as shown in Fig. 6.

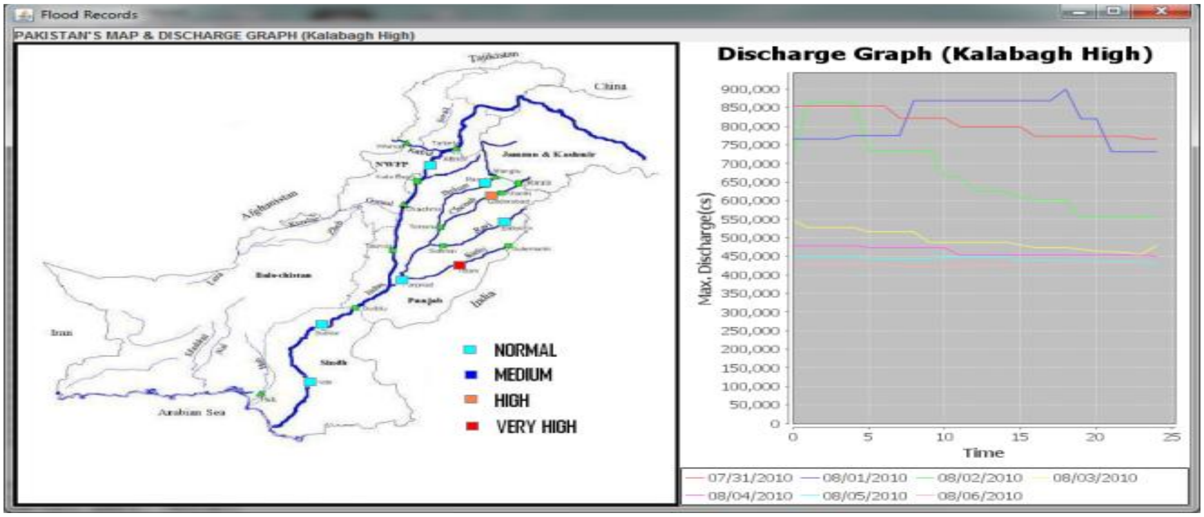


Fig. 6 High Discharge of Kalabagh

Figures are showing the current situation of Kalabagh and also its historical data.

Conclusion: In this work, a new model is developed using strong mobility for one server, mobile agent and two client mobile agents. A critical area 'Kalabagh' is also selected for this model. Because, flood in 'Kalabagh', is coming from two sources, Kabul River and Tarbela. We applied client mobile agents at both places, Kabul River and Tarbela. We showed the communication mechanism among all mobile agents and responsibilities of each component. For communication, we proposed VSAT (Very Small Aperture Terminal) due to its reliability and availability in critical situation. In the next step, an algorithm is developed for this model. Then, we formalize the algorithm using the mobile Petri net. Mobile Petri net is used to represent a system mathematically. Further, reachability graph used to verify this mathematical model. The results of the reachability graph showed that, there is no deadlock in the system.

At the end of this phase, we developed a system of this model using Java language. The software is very dynamic, calculating the discharge, plotting the graph and also showed the critical areas on the map. The system also showed the different states like normal, medium, high and very high. The results of the software proved the perfection of our model. For further directions, the generalized mobile Petri net model will be developed to implement and handle any critical situation. The above results proved our proposed model that means it can be applied in large areas for prediction.

REFERENCES

[1] Breckpot, M., Agudelo, O. M., & De Moor, B. (2013). Flood control with model predictive control for river systems with water reservoirs. *Journal of Irrigation and Drainage Engineering*, *139*(7), 532-541.

[2] Ghazali, J. N., & Kamsin, A. (2008, July). A real time simulation and modeling of flood hazard. In N. E. Mastorakis, V. Mladenov, Z. Bojkovic, D. Simian, S. Kartalopoulos, A. Varonides, ... & K. L. Man (Eds.), *WSEAS International Conference. Proceedings. Mathematics and Computers in Science and Engineering* (No. 12). WSEAS.

[3] Bessiere, H., Roux, H., & Dartus, D. (2007). Data assimilation and distributed flash flood modeling. S. A. Nelson, "River flooding," Tulane University.

[4] Sengupta, S. K., Bales, J. D., Jubach, R., Scott, A. C., & Kane, M. D. Flood Forecasting and Inundation Mapping in the Mahanadi River Basin: A Collaborative Effort between India and the United States. N. Sezen and N. Gunduz, " Meric River Floods And Turkish- Bulgarian Cooperations, " International Congress on River Basin Management, Turcia, 2007.

[5] ANFAS, "Data Fusion For Flood Analysis and Decision Support (2009)" European Research Consortium for Information and Mathematics, ANFAS, http://www.ercim.org/ANFAS, Retrieved March, 2015.

[6] Ministry of lands Japan, " Japan Tokai Heavy Rain (2004) WMO/GWP Associated Programme on Flood Management, September 2000. http://www.apfm.info/publications/casestudies/cs_japan_full.pdf, Retrived March 2015.

[7] US Army Corps of Engineers, "The Hydrologic Modelling System," http://www.hec.usace.army.mil/software/hec-hms/, Retrived March 215.

[8] Nakamura, K., Tockner, K., & Amano, K. (2006). River and wetland restoration: lessons from Japan. *BioScience*, *56*(5), 419-429.