Journal of International Technology and Information Management

Volume 18 | Issue 1 Article 3

2009

Formal Verification of Urban Traffic System Using the Concept of Fuzzy Workflow Simulation

Manuj Darbari
ICFAI Business School, Lucknow

Abhay Kumar Srivastava
ICFAI Business School, Lucknow

Sanjay Medhavi
University of Lucknow

Formal Verification of Urban Traffic System Using the Concept of Fuzzy Workflow Simulation

Manuj Darbari
Abhay Kumar Srivastava
ICFAI Business School Lucknow
INDIA

Sanjay Medhavi
University of Lucknow
INDIA

ABSTRACT

Modeling complex urban traffic system requires extensive use of workflow methodologies which can simulate almost real time traffic situations. Number of studies were done in the field of Urban Traffic Simulation like PLOTS, NETSIM and PALAMICS but none of them could provide a real formal solution. Our paper proposes a methodology using Fuzzy Petri nets for modeling real time traffic system. Fuzzy Petrinets are also visualised and tested by using Flow- Charter and HPSIM software packages. These packages help in providing successive iterations of process model for further refinement.

INTRODUCTION

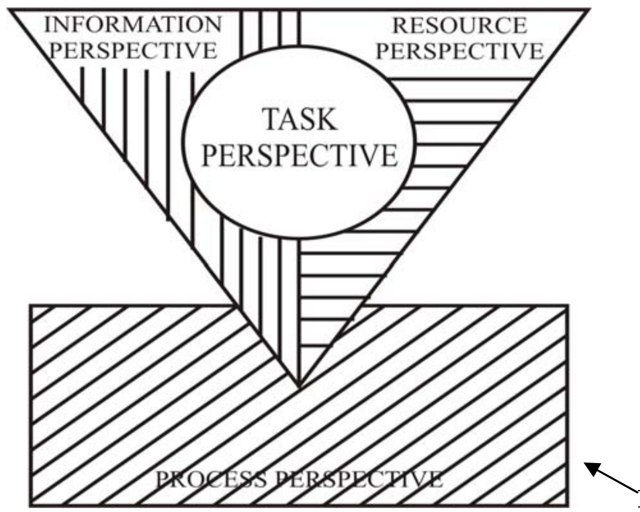
Urban Traffic Congestion is becoming a problem in India. In the past there were number of researches being done that states that jam are not only exasperating but they also cost India a lot in terms of productivity. The study on the subject showed the external cost of road traffic congestion alone amount to 0.5% (Economic Survey of GDP, by MARG).

The problem comes from the fact that transport users do not always cover the costs. The cost of infrastructure, congestion, environmental damage and accidents are not reflected in the price structure. Workflow management as a method of initiation of operations in a real-world processes and systems can e used to describe and analyse the behaviour of an existing and conceptual system. By changing these models (Fish, 2006), the behaviour of the system adopt to its environment.

MODELING OF PROCESS USING PETRINETS

Any business process can be viewed from three different prospective. Figure 1 shows some of the prospective which is relevant for business process management systems (Sheth, Joosten, Scacchi & Wolf, 1996). The process prospective describes the control flow i.e., the ordering of the task. The information perspective describes the structure of the organization and identifies resources, roles and groups. The task perspective describes the content of the individual step in the process (Asgekar, 2003).

Figure 1: Perspective of models driving workflow management system.



PROCESS PERSPECTIVE

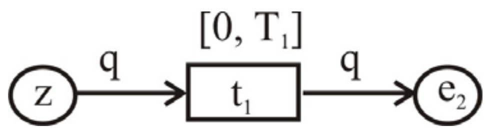
Many techniques have been proposed to model the process perspective. Some of these techniques are informal in the sense that the diagram used have no formal defined semantics. These model are typically very initiatives and the interpretation shifts depending on the application domain, and characteristics of the business process at hand. Examples of informal techniques are ISAC (Lim, Lim & Heinrichs, 2008), DFD, SADT and IDEF. These techniques may serve well for discussing the work processes. However, they are inadequate for directly driving information system since they are incomplete and subject to multiple interpretations. Hence are more specific and precise modeling is required.

Petrinets (Murata, 1989), invented by Carl Adam Petri in early sixties, provide a best means of formalising any operations. In this methodology traffic flows are represented by a visual graph and this graphical net itself serves directly as illustrative representation of simulated outputs. Any rates are essentially driven by common simple rules we would easily follow by hand. In general, places are drawn as circles and transitions as boxes and bars. Directed arcs connected transitions and places either from transitions to a place or from a place to transitions. Arcs are labeled with position integers as their weights. Places may contain tokens. The token is represented by a black dot in the place p1. The firing rules of petrinets as shown in Figure 2 are:

- *A transition t is enabled if each input place of 't' contains at least w(p,t) tokens, where is w(p,t) is the weight of the are from p to t.*

- *The firing of an enabled transition t removes w(p,t) tokens from each input place p of t and adds w(t,p) is the weight on arc from t to p.*

Figure 2: Petrinet Example.



The above classical rules can be extended by associating a time interval and fuzzy decision logic with each transition. Such petrinet net are known as intelligent timed petrinet.

FUZZY PETRINET INTRODUCTION

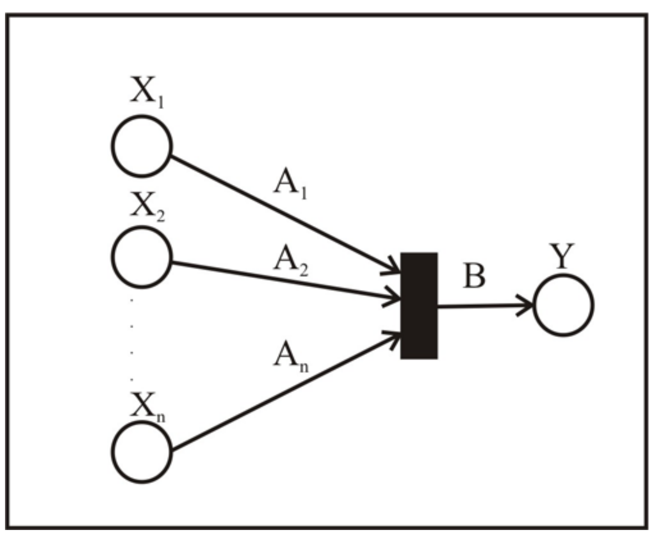
During process simulation by petrinets we have to time to time illustrates the status which we are not sure whether that event will happen or not. For modeling an unpredictable situation like Urban traffic system it is not always possible to simulate by assuming certain conditions that are normally considered as "Ideal". These limitations of petrinet can be resolved by introducing fuzziness (Yenduri, Perkins & Sarder, 2007) in its outcome. Any IF THEN (Knybel, 2005), rule of the previous function can be defined by the help of petrinet as :

The set of IF-THEN rules, which forms the linguistic description:

R_1 : = IF X1 is A11 AND ..... X_n is A_{In} THEN Y is B_1
:
:
Rm : = IF X1 is A_m, AND .... X_n is A_{mn} THEN Y is B_m.

It can be modeled as:

Figure 3: Fuzzy Petrinets.



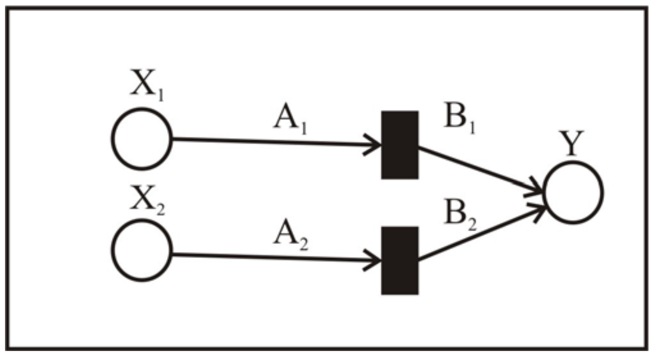
The above condition corresponds to the simple situation of decision making but in some cases when some edges are missing, then we have to introduce undefined condition as "UNDEF" is

R1 : IF Xi is A_1 AND X_2 is UNDEF THEN Y is B_1

R2 : IF X, is UNDEF AND X2 is A_2 THEN Y is B_2

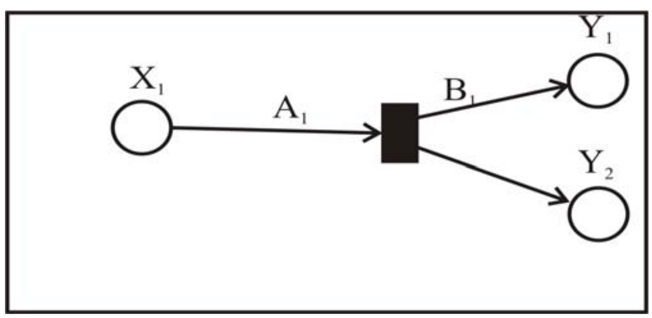
It can be shown by the figure 4.

Figure 4: Fuzzy Modeling when one of the edge is missing.



Similarly for a decision stage when more than one output variable are possible.

Figure 5: More than one output variables

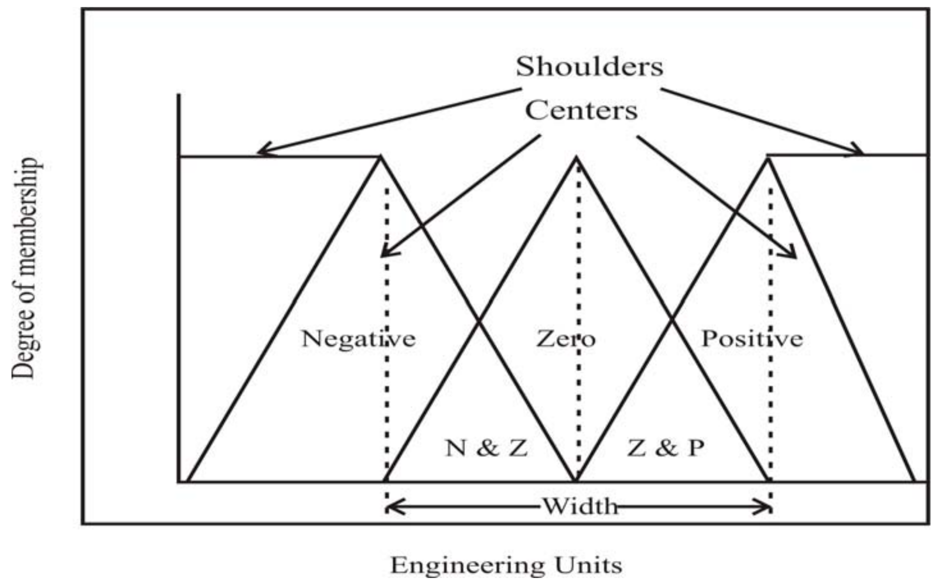


The corresponding description could look like

R : = If X_1 is A, THEN Y , is B_1 and Y_2 is $B_{2.}$

We can define the membership function as graphical representation of the magnitude of participations of each Input. It associates a weighting milk each of the input that are processed, define functional overlap between inputs, and ultimately determines an output response. The rule uses the input membership values as weighting factor to determine their influence on the fuzzy outputs sets of the final output conclusion. Once the functions are inferred, scaled and combined, they are defuzzified into crisp output which drives the system. There are different memberships functions associated with each input and output response shown in Figure 6.

Figure 6: Features of Membership Functions.



The degree of Membership (DOM) is determined by plugging the selected input parameter (error or error, dist.) into the horizontal axis and projecting vertically to the upper boundary of the membership functions).

ERROR AND ERROR - DOTS FUNCTION MEMBERSHIP

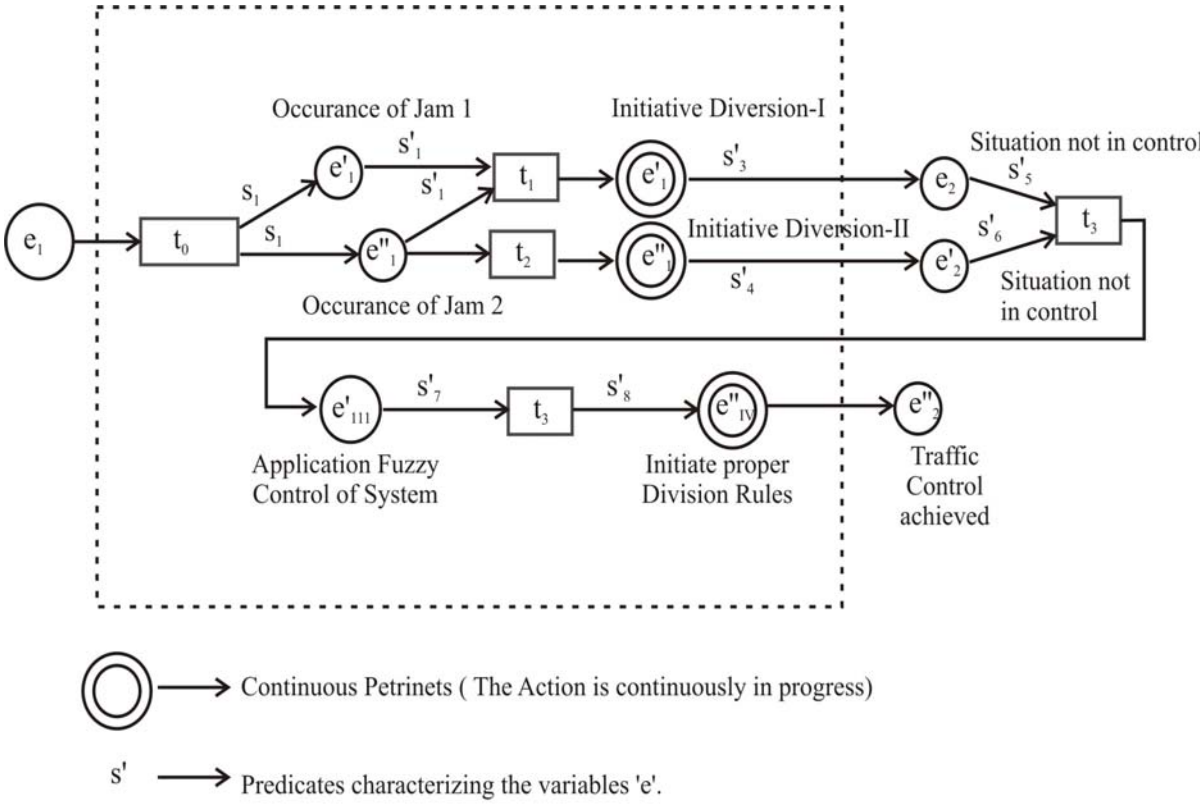
The degree of membership for an "error" of -1.0 projects up to middle of overlapping part of the "negative" and "zero" function so the results is "negative" membership = 0.5 and "zero" membership = 0.5. It can derive from figure 5. These particular input conditions indicate that the feedback has exceeded the command and is still increasing.

Thus, there is a unique membership function associated with each input parameter. The membership associate a weighting factor with values of each input the effective rules. By computing the logical product of the membership weights for each active rule, a set of fuzzy output response magnitudes are produced.

FORMAL VERIFICATION OF URBAN TRAFFIC SITUATION USING FUZZY PETRINETS

Consider a situation of traffic jam (e_1) occurs more than once within period T2, this situation concerns the case where multiple occurrences of one event within a certain time period causing another single event to occur. This situation introduces the motion of expiration time of events . If an event is not consumed by a rule, it may expire after a time interval. The Petrinet model pertaining to these events can be defined by placing token's e', and e'', the transition t_2 and t_3 are enabled, but they cannot be fired immediately. When there are two tokens arriving in place e', and e'', transition t_1 fires immediately and produces the event e_2 is "Notification to Traffic Department".

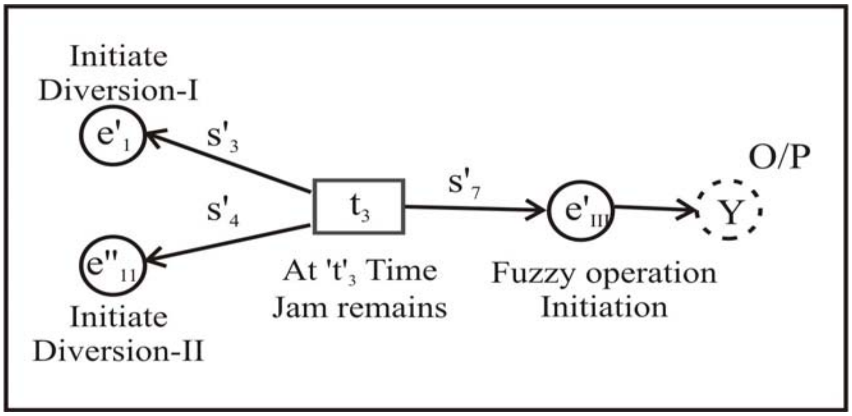
Figure 7: Petrinet Decision Model.



After transition to fires, two tokens are returned to place e', and e", leading two situations, either the traffic department will know about the current situation and take necessary action before the second jam situation or the two jam situation will be handled simultaneously if the necessary time to control the first jam expires.

Under this condition the control efforts of diverting the traffic (Darbari, Medhavi, & Srivastava, 2007) in by lanes and alternative road fails and it is given to the decision center to take the necessary action. Fuzzy Nets (Juhás, Lorenz, & Mauser, 2007) are used to model this deadlock situation of two jams by using IF_THEN Fuzzy rules.

Figure 8: Fuzzy Net representation.



The condition can be expressed in the linguistic format as R_1 : IF e'_1 is s'_3 AND e'_{11} is s'_{4t3} THEN e'_{111} is s'_{7t3}.

Where s'_3, s'_4 and s'_7 set of rules under which an event can occur.

We can write an algorithm for the above condition where the decision of traffic diversion can be initiated as:

input : fuzzy traffic control : *ftc*

output : set of output rules : *sorl*

sorl = ϕ ;

for each output place e' of sort do *// creates set of input variables on whose e depends,*

for each input transition e of sorl e' do

// add all inputs of transition 's' to input set e
inputs = inputs Us inputs ;
end
for each input transition e of sorl do
// construction of rule corresponding to the transition 's'
rule = ϕ *// rule belongs to traffic diversion*
for each element in from inputs do
if rule ≠ ϕ, then rule = rule + AND;
if in ∈ s, inputs.then
rules = rule + in.name is edge (s, it).value;
else

rule = rule + in.name is UNDEF;
end
rule = rule + THEN e'. name is edge (in , e').value;
rb → rule database
rb → rbUrule ; *// add rule of Traffic control to Database for further reference*.
end.

The above algorithm rules can be quantified in the form of Rule Matrix Workflow of Fuzzy Petrinets (Cai, Zhao, Jia, Ye, & Li, 2008). There will be some combination of variables stated below as :

e'N = Jam getting Worst
e'P = Jam being Resolved
eZ = Zero_error
e'_{I} = Controlling Jam - I
e'_{II} Controlling Jam - II
"-" = No change in Jam
cmd = Traffic Jam Resolved
Tr_J = Traffic jam
Error = Cmd_Tr (+ = only Jam - I, - = Only Jam - II)
Error_dot = Time derivative or Error (+ = getting, - = getting worst)
Output = Controlling Jam-I or No change or Controlling Jam-II

Figure 9: Traffic Control System Response.

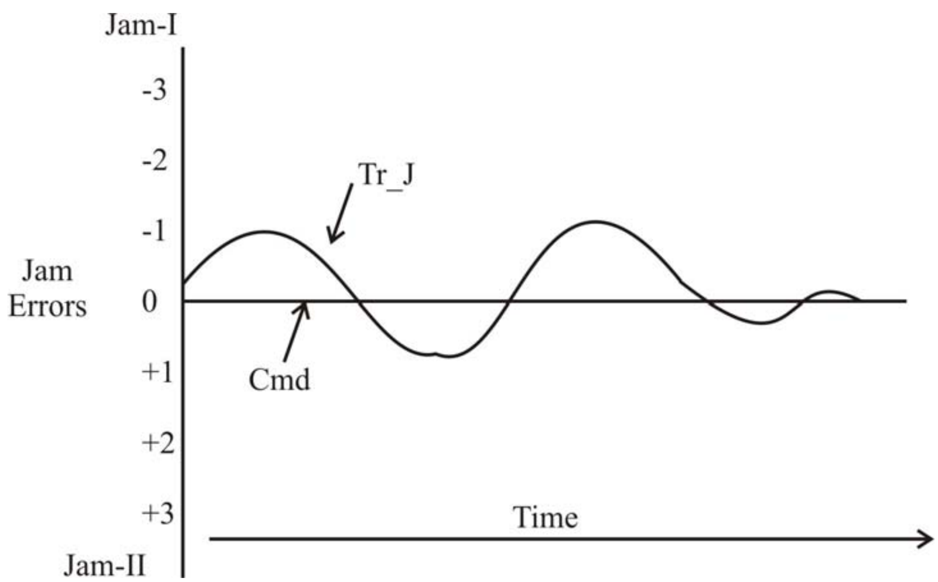


Figure 9 shows what command and error look like in a typical control system relative to command set points as the system hurts for stability. We have to define some value in order to design fuzzy workflow nets.

DEFINITIONS

INPUT # 1 : ("Error", positive (e'_1, p) , Zero (-), negative (e'_1, N)

INPUT # 2: ("Error-dot", positive (e'_1, p) , Zero (-), negative (e'_1, N)

CONCLUSION : ("Output", Controlling Jam 1 (e'_1), No change (-), Controlling Jam II (e'_{11})

OUTPUT

e'_1 = Call for controlling Jam-I only

"-" = No change in controlling levels

e'_{11} = Call for controlling Jam-Ii only.

We can develop Rule Matrix from the above situation with errors and Error -dot functions.

Table 1: Rule Matrix of Traffic Jam.

Error-dot \ Error - function	e'_1N	eZ	e'_{11}P
e'_1N	e'_{11}	e'_1	e'_1
eZ	e'_{11}	-	e'_1
e'_{11}P	e'_{11}	e'_{11}	e'_1

Now in order to apply the rule Matrix derived above we have to use membership function. The membership function is a graphical representation of the magnitude of participation of each input. It associates a weighting with each of the inputs that are processed, define functional overlap between inputs, and ultimately determines an output response. Once the functions are inferred, scaled and combined they are defuzzified (Sen & Matolak, 2007) into a crisp output which drives the system. As discussed, we can draw the membership function by considering the degree of membership from an "error" of -1.0 so the middle of the overlapping part is a negative and zero function. For an error-dot of +2.5, a "zero" and "positive" membership of 0.5 is indicated.

INPUT DEGREE OF MEMBERSHIP

"error" = - 1.0 ; "negative" = 0.5 and "zero" = 0.5 and "error-dot" = +2.5; "zero" = 0.5 and "positive" = 0.5 . We can defuzzify to return to the crisp off rule for (e) = error and (er) = error-dot.

1. If (e<0) AND (er<0) then e'11 0.5 & 0.0 = 0
2. If (e=0) AND (er<0) then e'1 0.5 & 0.0 = 0.0
3. If (e>0) AND (er<0) then e1' 0.0 & 0.0 = 0.0
4. If (e<0) AND (er=0) then e'11 0.5 & 0.5 = 0.5
5. If (e=0) AND (er=0) then '-' Chng 0.5 & 0.5 = 0.5.
6. If (e>0) AND (er=0) then e' 0.0 & 0.5 = 0.0
7. If (e<0) AND (er>0) then e' 0.0 & 0.5 = 0.0
8. If (e=0) AND (er>0) then e'11 0.5 & 0.5 = 0.5
9. If (e>0) AND (er>0) then e'1 0.0 & 0.5 = 0.0

The inputs are combined logically using the AND operator to produce output response values for all expected inputs. The active conclusions are then combined into a logical sum of each membership function. A firing strength for each output membership function is computed. They are then combined into logical sums in a de-fuzzification process to produce the crisp output.

DIAGNOSING THE PROCESS USING WORKFLOW SIMULATOR

Based on some of the results presented in the previous section, the tool HP simulator (Poliakov, Sokolov & Mokhov, 2007) converts the workflow process in Microsoft Flow Charter. The major emphasis of using such tool is to analyse the workflow processes in a logical and meaningful order.

The basic for the diagnosis process of figure 11 is a theorem on workflow net stated as :

(Workflow .net) A P/T net N = (P, T,F) is a workflow net (Wfnet) iff

(i) $i \in P \wedge .i = \phi$

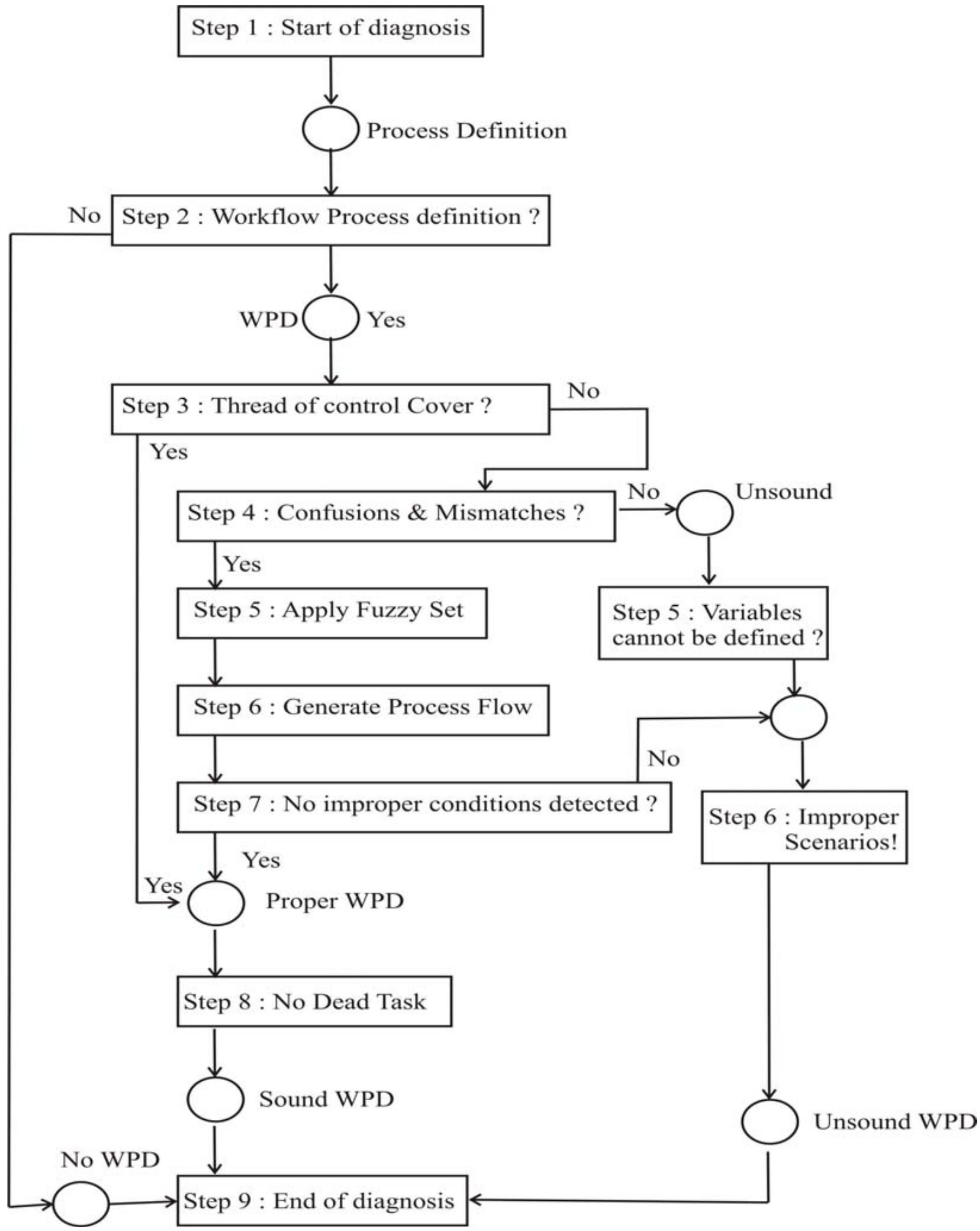
(ii) $o \in P \wedge o. = \phi$

(iii) The short-circuited P/T net (P,TU {t}, F U {(0,t), (t, i)}), devoted N, is strongly connected , where t $\notin$ T.

Where “i” and “0” are called places termed as “start” and “finish”. For every transition in a workflow net, there must be a directed path from “i” to “t” and directed path from t to 0. In P/T net terms, this confirms to strongly corrected ness under the assumption that there can be a directed path from “0” to “i”.

The terminology "**thread of control cover**" refers to set of parallel task to be executed. Each such thread specifies that certain tasks have to be executed in a certain sequential order to get a certain piece of work completed.

Figure 10: Flow Chart of Diagnosis Process for Jam Condition.



APPLICATION OF SMART DRAW IN URBAN TRAFFIC MODELING

Consider a case of Urban Traffic system described previously in which an accident has resulted in two jam situation at distant locations.

Figure 11: Flow Chart of UTS using Smart Draw.

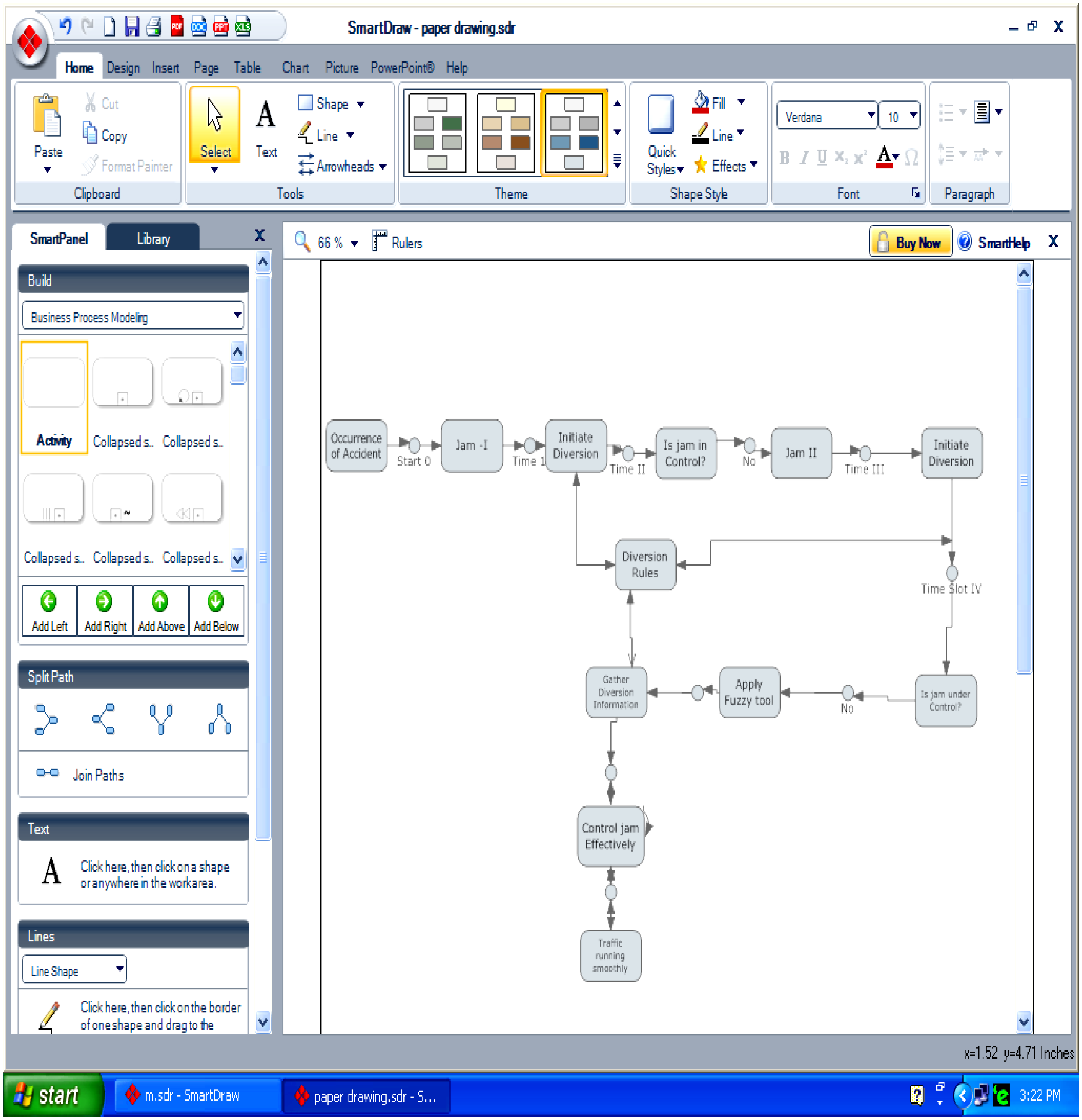
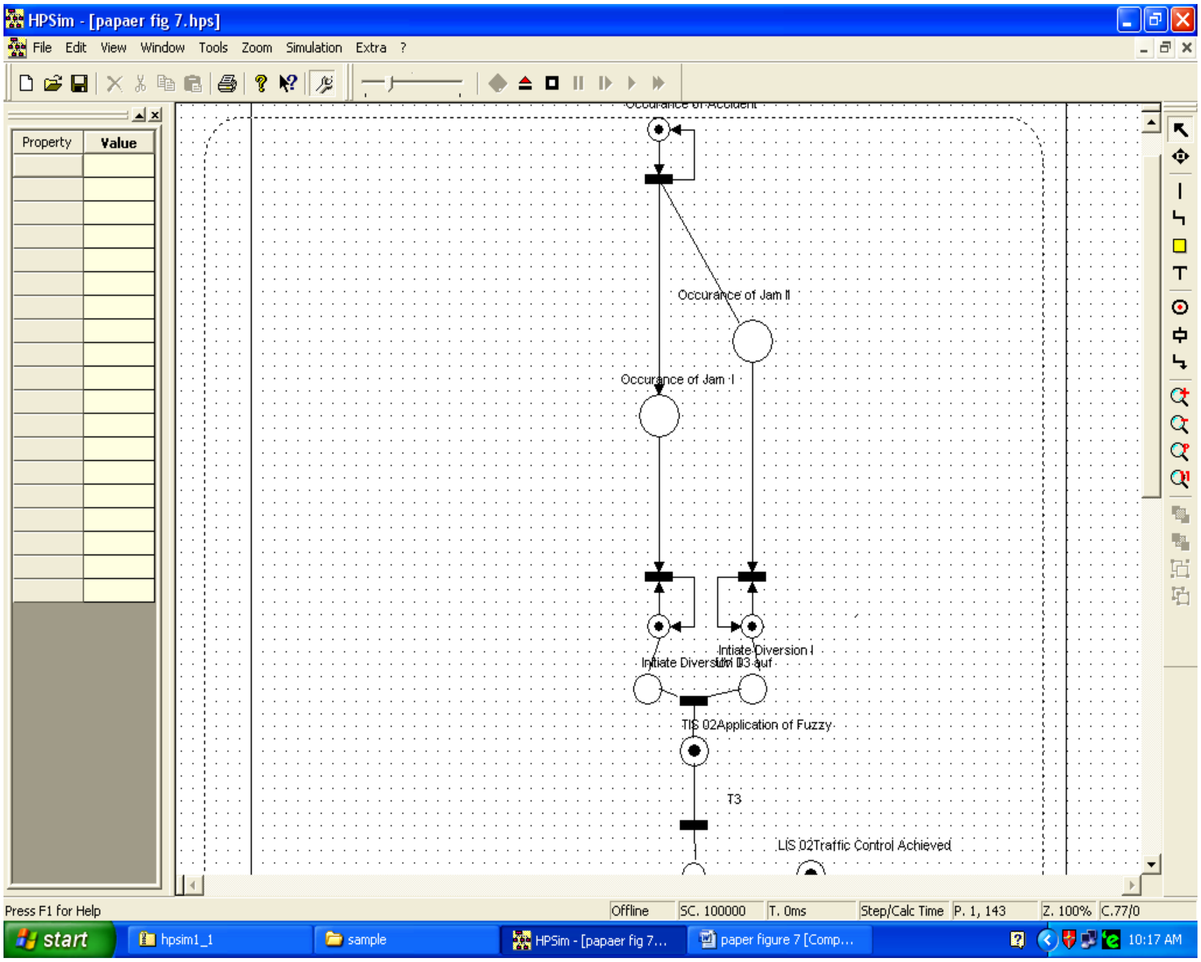


Figure 11 gives the snapshot of flow charts simulated by Smart Draw. The process definition is tested and Fuzziness modeling is also incorporated which is derived from rule- matrix of the Fuzzy logic. It is important to note that corrections to process definition may depend on the workflow system at hand. The flow chart is then analysed and "SIM 2.1(Fang & Elefteriadou, 2008) is used to further formalise it using Petrinets. The petrinet model described in Figure 8 was drawn in HPSM (Figure 12)by generating WFNet and diagnosed. There were three iterations required to completely check each and every process flow. During the first iteration one error was detected during the conversion. A small part of the process definition was not connected to the main part further two structural errors were found during the second iteration where the OR-joint being replaced by AND joint.

Figure 12: Simulation of Accident using Fuzzy Petri-nets through HPSIM 1.1.



In second iteration the fuzzy Logic not were checked and necessary sequence of steps were written for this step based in the token weights and logic levels, the token moved to forward.

Summarizing, the main conclusion of this case study is that HPSIM can be useful aid for detecting and correcting the process charts and finally to WF-Nets HPSIM provides a techniques for checking the consistency of transactional workflow including fuzzy constraints. The software is restricted to acyclic workflow and only gives necessary conditions for consistency. We can also use specific feature of reduction which corresponds to soundness and the class of workflow process.

CONCLUSION AND FUTURE SCOPE

Workflow management technology is rapidly gaining popularity in the support of business process. We have developed an approach for modeling Urban Traffic system using petrinet for vague values. We have clubbed the tools of fuzzy logic to work under special traffic deadlock situations. We have special traffic deadlock situations. WE have taken the "token" as bearer of fuzzy sets and the edges are evaluated by linguistic expression from IF THEN rules. The use of rule_matrix helped us in modeling more precise WF- Net Model. The WF Net developed more precise WF-Net model. The WFNet developed physically is being verified by Smart Draw, for verifying flow-charts and HPSIM for petrinet verification and simulation.

Overall, the approach provides a concrete step in formalising workflow networks for static and dynamic models. The future scope could be a model a stochastic fuzzy petrinets for urban Traffic systems.

REFERENCES

Asgekar, V. (2003). Event Management Graduates with Distinction. *Supply Chain Review*, 15-16.

Sheth, A., Joosten, S., Scacchi, W., & Wolf, A. (1996). A report from NSF workshop on workflow and process automation in information system, *SIGMOD*. In Novak, V. (2000), Fuzzy Sets and their Applications, Adam Hilger, Bristol.

Fang, F.C. & Elefteriadou, L. (2008, December). Capability-Enhanced Microscopic Simulation With Real-Time Traffic Signal Control. *IEEE Transaction on Intelligent Transport System*, 625-632.

Juhás, G. & Mauser, R. S. (2007). Complete Process Semantics for Inhibitor Nets. 28th International Conference on Applications and Theory of Petri Nets and Other Models of Concurrency, ICATPN 2007.

Cai, G., Zhao, S., Jia, L., Ye, Y., & Li, X. (2008). Approach of Train Group Operation Conflict Prevention Reasoning Based on Fuzzy Time Petri Nets. Fourth International Conference on Networked Computing and Advanced Information Management, ICNCAIM-2008, 272-277.

Poliakov, I., Sokolov, D., & Mokhov, A. (2007). Workcraft: A Static Data Flow Structure Editing, Visualisation and Analysis Tool. International Conference on Applications and Theory of Petri Nets and Other Models of Concurrency, ICATPN 2007, 505-507.

Lim, K. S., Lim, J. S., & Heinrichs, J. H. (2008). Validating the End-User Computing satisfaction Instrument: A Confirmatory Factor Analysis Approach Using International Data. *Journal of International Technology and Information Management*, 17(2), 153-165.

Fish, K. E. (2006). An Artificial Intelligence Approach to International Market Screening DSS. J*ournal of International Technology and Information Management*, 15(2), 49-60.

Knybel, J. (2005). Representation of Fuzzy IF-THEN rules by Petrinets. In *ASIS* - 2005 conferences, Ostrava.

Darbari, M., Medhavi, S., & Srivastava, A. K. (2007). Application of UML for modeling Urban Traffic System Using Producer Consumer theory to generate Process Algebra Model. *Journal of International Technology and Information Management*, 16(4), 75-81.

Murata, T. (1989). Petri Nets: Properties, Analysis and Applications, In proceedings of the Institute of Electrical and Electronics Engineers , 77(4).

Sen, I. & Matolak, D.W. (2007), Vehicle-Vehicle Channel Models for the 5-GHz Band. *IEEE Transaction on Intelligent Transport System*, 235-245.

Yenduri, S., Perkins, L., & Sarder, M. (2007), An Analysis of Programming Language "D." *Journal of International Technology and Information Management*, 16(3), 87-92.