Development of a Network Flow Simulator (NESSY-IV) for Analyses of Mass Evacuation in Case of Emergency

TOSHISUKE HIRAMATSU*

An interactive network flow simulator: NESSY-IV (Net Structure Analyzing System IV) has been developed to analyze effective countermeasures to prevent systems panic which will be caused by mass evacuation from a downtown area after a warning of an earthquake.

A network flow model is represented by three kinds of data (relation data, node data, and modify data), by which interactions among human flows and information flows are described and analyzed. The network structure of a simulation model is described using a set of relation data which is composed of several conditional variables such as: node name, flow type, delay time, transition probability, threshold of flux, effective step interval, and relation type. An initial state value and physical conditions of a node are given by node data. A simulation model is not fixed, but the model structure is modified dynamically according to description of modify data while NESSY-IV is running. In addition, in order to represent a drastic change of a situation, it is possible to add some preparatory relations to a running simulation model at a prescribed time.

As a case study, NESSY-IV was applied to analyses of "Mass evacuation from Sakae area at Nagoya City after a warning of an earthquake is issued". As a result several useful countermeasures to prevent systems panic are obtained. Consequently, it is concluded that NESSY-IV is an efficient network flow simulator for analyses of mass evacuation in emergencies.

1. Introduction

After hearing a warning of an earthquake, the "go home" behavior is the prevailing action for people in a downtown area of a large city. Because of simultaneous actions for evacuation and rushes of people to limited places such as subway stations and bus-stops, great confusion will be caused. Under such *emergent* conditions, analyses for effective countermeasures to prevent *systems panic* and to evacuate many people in a downtown area safely are an urgent matter for serious thought.

The purpose of the present study is to develop a dynamic network flow simulator which analyzes problems of mass evacuation under such emergencies as an earthquake warning.

2. Necessary Conditions to Simulate Problems of Mass Evacuation

In order to simulate a mass evacuation after a warning, it is necessary to take account of various circumstances and conditions as follows.

1. Human actions are controlled by information, and the information is diffused by human actions. Two kinds of flows, the mass flow (human flow) and the information flow, (such as earthquake warnings), should be distinguished and the interactions between the two kinds of flows that constitute a network structure including loops should be analyzed.
2. People are classified into several groups. Their modes of actions are affected by manners of information transmission, control of action, time used for decision making and dispersion of action. Time for decision making is one of the operational parameters, whose value is defined as a delay time to start action.

(A) OFFICE = employees belonging to big companies and public offices.
The issue reaches the security staff→time is required for decision making→the information is announced→and then action starts. The control of their actions is possible. They will prefer, for evacuation, to use the type of transportation they use every day.

(B) DEPART = shoppers or visitors to department stores, supermarkets, theaters, etc.
Processes of issue and decision making are similar to OFFICE. However, the control of information and actions is not controlled. They are not familiar with either the inside or the outside of the buildings, and they prefer, for evacuation, to use the type of transportation they used to come.

(C) STREET = walkers on the street and shoppers at stores, coffee shops, etc.
The issue is received by public broadcasting and the information is diffused. Action is started as soon as the information is received. Therefore, control is almost impossible. In some cases, actions will be started by other's action without any understanding of the situation, and they are entangled with mass flow.

(D) SUBROAD = people on underground streets and shoppers at underground

*Computation Center, Nagoya University.

stores, coffee shops, etc.
They have little chance of contact with public broadcasting and the announcement will reach the security center. The information is announced through the BGM (background music) system. The efficiency is not high and diffusion of actions will be prevailing.

3. It is necessary to consider several cases such as: weekday case, weekend case, and night case. The initial distribution of persons from the four groups (OFFICE, DEPART, STREET and SUBROAD) is different, for each case.

4. Methods of evacuation are classified into several groups, of which critical flux, capacity of transportation and delay time are different for each. In addition, the effect of reduction in transport ratio due to the regulation of speed of subways and buses should be considered.

(A) FOOT = run away through a street.
The flux and capacity are determined by road circumstances.

(B) BUS = go to a terminal and catch a public bus.
The flux will decrease rapidly with increase in FOOT. If terminal becomes crowded, some people arriving at a terminal will change their minds and choose METRO or FOOT to evacuate.

(C) CAR = go to underground parking area and start to evacuate in cars.
The flux will decrease rapidly with increase in FOOT. If parking becomes crowded, some people will change their minds and choose METRO or FOOT to evacuate.

(D) METRO = go to a station through an underground passage, pass through a wicket, enter a platform to wait, and catch a train.
If a platform becomes crowded, some people will change their minds and choose FOOT.

5. The following circumstances and conditions should be considered with respect to subway stations and stairs connecting the ground and underground. People, intending to evacuate by subway, go to a station through an underground passage, pass through a wicket, enter a platform to wait, and catch a train. If a platform becomes crowded, some people will change their minds to evacuate by FOOT, and retrace their steps to the ground.

The number of people staying at platform is determined by the difference between the flow into the platform and the flow from the platform. If many people rush to a station in a short time, a mass level at the station exceeds its critical capacity and a mass panic will occur. In that case, the function of the station stops for a while until the mass level is reduced to a noncritical level. In the worse case, a riot developes and facilities in the station are destroyed, and so the station will be of no use.

In addition, it is necessary to take account of variations such as the following: critical flux at the underground, department junction to subpassage, etc., and capacity at the terminal, stairs, etc. due to the change of circumstances.

6. The duration of simulation is divided into several phases, by which the change of circumstances and the variation of external conditions is represented as:

Phase I (time $=0$ —T_1): normal state
Phase II (time $=T_1$—T_2): speed of traffic is reduced
Phase III (time $=T_2$—T_3): traffic is affected by streams from outsides of the restrited area.

In addition, it is necessary to add some groups of phenomena at a prescribed time, by which the drastic change of circumstances (e.g. the occurrence of earthquake) can be described.

7. In order to simulate various situations continuously, a simulation model should be dynamically modified depending on the results of simulation while the simulator is running. Modifications of a simulation model are carried out as follows. If a function in a simulation model is stopped, the function is replaced by a preliminary function until its recovery, and switching the status of the function (to sleep ⇄ to awake) are repeated depending on the situation. In addition, if a function is damaged and the function is of no use, the function is substituted to an alternative function which is not damaged.

3. The Trend of Simulation Languages for Network Flow Problem

A considerable number of simulation languages, that seem to be adequate to the analyses of mass evacuation problems, are currently in use.

DYNAMO [10] is one of the simulation languages suitable for continuous models which are usually represented mathematically by difference equations. In DYNAMO, the equations of the various state variables are evaluated at fixed time intervals (DT). This language is based on the industrial dynamics concept [1], [2], which focuses on the stability of the system and its response to exogenous factors and to internal changes. The fundamental structure of a dynamic system is represented in terms of varying rates of flow and the resultant changes in level of the state variables. In DYNAMO, activities of enterprises are represented by six kinds of flows (i.e. material, money, order, personnel, capital equipment, and information), and a simulation model is described using six kinds of equations (i.e. level eq., rate eq., auxiliary eq., supplementary eq., constant eq., and initial value eq.). Nonlinears are included in the model through the use of table functions, delays, boxcar train, and other special functions. In addition, a convenient report generator is prepared and output can be obtained in the form of table and/or print-plotted graphs at selected increments of time.

It is true that DYNAMO is a useful simulation language to deal with continuous flow problems, but the simulation model is fixed and it is difficult to modify the simulation model dynamically depending on the results of simulation while DYNAMO is running.

As concerns the network flow problems, several network simulation languages have been developed.

Q-GERT [9] is a typical network simulation languages. A network model of Q-GERT consists of nodes and branches. Nodes are used to separate branches and are used to model milestones, queues, and decision points. Different types of nodes (e.g. source node, statistics node, sink node etc.) are included in Q-GERT to allow for the modelling of complex queueing situations and project management systems. A branch represents an activity that involves a processing time or a delay. Each branch has a start node and an end node. Four classes of branching (deterministic; probabilistic; conditional take-first; and conditional take-all) are provided to specify activities emanating from a node.

Items, referred to as transactions, flow through a network of nodes and branches. Transactions originate at source nodes and travel along the branches of the networks. Transactions can represent physical objects, information, or a combination of the two. Transactions have attribute values that allow different types of objects (or the same type of objects with different attribute values) to flow through the network. Procedures are available to assign and change attribute values of transactions at the various nodes of the network.

As transactions flow through the network model, statistics are collected on travel times, the status of servers and queues, and the times at which nodes are released.

Another network flow language that has been developed is INS [11], which contains new network programming and design concepts. INS has a dual flow orientation where both transactions and resources flow through the system. Resource networks are built to represent the sequence by which resources are assigned to activities.

As described above, Q-GERT and INS are general purpose network simulation languages, and it may be possible to apply two simulators successfully to the problem of mass evacuation.

However, taking account of the following situations such as: ease of coding of the simulation model; language availability on other computers (portability); degree to which language supports and different modelling concepts (flexibility); efficiency in the execution; and reliability of simulation language and support systems, it is not practical to apply a general purpose network simulator to special kinds of problem such as mass evacuation.

Therefore, it is preferable to develop a special purpose simulation language which is suitable to the analyses of mass evacuation.

In recent years, the author has developed three kinds of simulation languages to process network causal relations: NESSY-I, NESSY-II, and NESSY-III.

NESSY-I is a structural analyzing system for net thesaurus [3], and is able to find broader terms, narrower terms, and synonyms of a specified keyword. In addition, it is also possible to search routes between two keywords, or to detect loops passing through a specified keyword.

NESSY-II is a revised relational model simulator which is applicable not only to a structural analysis but also to a numerical analysis of a network flow model [4]. The process of the diffusion of issued information on earthquake is simulated using NESSY-II [8], [12], [13], [14].

NESSY-III is a dynamic net structure analyzing system for causal relations, which is able to analyze net structure including loops [6]. In addition, it is also possible to modify the structure of a network flow model by deleting or adding some causal relations while NESSY-III is running. NESSY-III is applied to the analysis of the responses of individuals with respect to the timing of earthquake warnings [8], [12], [13], [14].

4. The Principles and Basic Functions of NESSY-IV

In order to analyze a problem of mass evacuation, a network flow simulator: NESSY-IV (Net Structure Analyzing System IV) [5], [7] has been developed depending on the implemental experiences of NESSY-III.

Principles employed in the design of NESSY-IV are as follows:

(1) NESSY-IV works in an interactive mode, and a network flow model including multiple loops is simulated and analyzed using interactive commands.

(2) Three kinds of data (i.e. relation data, node data and modify data) are employed to represent the network flow model of causal relations to be analyzed.

(3) A network flow consists of two kinds of flows: mass flow and information flow. Mass flow and information flow are represented using two types of relation data, and the rule of mass conservation holds to the former.

(4) Two types of output functions: a saturated type (pipe model) and a linear type (transition probability model) are employed to calculate the node level in a mass flow.

(5) Relation data is represented using a set of binary relations, by which a flow direction, delay time, transition probability, threshold of flux, effective step interval, and type of output function are described.

(6) There are three kinds of nodes (i.e. general node, monitor node and dummy node), in which the former two nodes are given by users and the third one is generated automatically by NESSY-IV in order to adjust delay time between two nodes.

(7) In order to monitor the status of a node, conditional variables (level, inflow, and outflow) are printed

for monitor nodes while NESSY-IV is running.

(8) Each node has information on its modify type, node level, inflow threshold and two level thresholds (upper and lower), by which causal relations are modified while NESSY-IV is running.

(9) Besides an activation and a cut function of causal relations, a switching function (to sleep⇄to awake) of binary relation and sequential activation of some binary relations is also possible.

(10) A relation table and a hashing technique are employed for the representation of relation data and fast word processing respectively.

(11) Programs are coded mainly by FORTRAN for portability.

5. Implementation of NESSY-IV

NESSY-IV is composed of one main FORTRAN program, 73 FORTRAN subroutines and three assembler subroutines (totals about 3,300 lines and occupies 203 KB (byte) of memory), and uses facilities offered by FACOM M-200 OS IV/F4 TSS. About 500 nodes including dummy nodes and 1,200 binary relations can be stored and analyzed at a time. A network flow model of NESSY-IV is represented by three kinds of data: relation data, node data, and modify data as follows.

5.1 Relation Data

A network flow model is defined using two types of relation data as follows.

(1) ($A>B$ (T, P, Phase No., TYPE))
A level value of node A is transfered from node A to node B depending on a specified delay time T, P and TYPE. This type of binary relation is used to express "mass flow" between node A and node B, where the rule of mass conservation holds.

(2) ($A=B$ (T, P, Phase No., TYPE))
A level value of node A is copied to node B depending on a specified delay time T and transition probability P. This type of relation is used to express "information flow" between node A and node B, where no rule of mass conservation holds.

Here, the meanings of variables employed in the above two expressions are as follows:

A and B are node names expressed by alphanumerical characters including blanks. A node name prefixed a letter "*" (e.g. "*STAIR1") is a monitor node. For monitor nodes, in order to collect statistical data, several conditional variables: level, inflow, and outflow etc., are printed at prescribed time interval.

T shows a delay time to move from node A to node B.

P shows a transition probability for TYPE=1, or a threshold of flux for TYPE=2. The details of output function expressed by TYPE=1 or TYPE=2 are described in the section "**5.5 Calculation of Node Level**".

Phase No. shows a step interval during which the relation is effective. The relation between Phase No. and a simulation time is shown in Fig. 1. Phase No. has fifteen kinds of integer values (0, 1, 2, ···, 10, 20, 88, 99, 100), and the meanings of fifteen values are as follows.

0: effective at all stages.

1–10: effective at respective stages preset by an interactive command "PH(t_1, t_2, ···, t_{10})".

20: to be activated at preset time "t_n", which is prescribed using an interactive command "PIt_n". Phase No. is changed from 20 to zero after activation.

88: conditional activation. Phase No. is changed from 88 to zero after activation.

99: conditional sleep→awake. Phase No. is changed from 99 to 100 after a relation is awaked.

100: conditional sleep⇄awake.

TYPE shows a linear type of output function for TYPE=1, or a saturated type of output function for TYPE=2, of which details are shown in Fig. 9.

Two types of binary relation data expressed in a form (1) or (2) are represented using a relation table as shown in Fig. 2. The relation table consists of nine areas: NODE1, NRELA, NODE2, T, P, Phase No., TYPE,

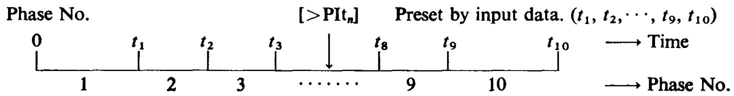

Fig. 1 Phase No. and Time.

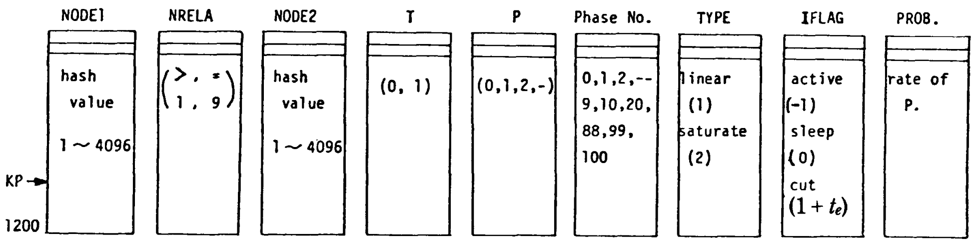

Fig. 2 Relation Table.

IFLAG and PROB. NODE1 and NODE2 are used to store hash values obtained from node names (A and B) using a hash function. Two operators, ">" and "=", are converted to two integer values 1 and 9 respectively and stored in NRELA area. Four parameter values: T, P, Phase No. and TYPE are stored in areas T, P, Phase No. and TYPE respectively. IFLAG is used to store a status of binary relation (-1 = active, 0 = sleep, and t_e = time when the relation is cut). The PROB. area is used to store the transition probability P (for TYPE = 1) or the ratio of threshold of flux P to the total flux from one node (for TYPE = 2). Each area has 1,200 words, therefore 1,200 binary relations are processed at a time.

5.2 Node Data

Node data is used to define an initial state and physical conditions of a node. The expression of a node data is as follows.

A (MDTYPE, LEVEL, LT1, LT2, VT)

Three variables (LT1, LT2, and VT) are employed to prescribe the physical conditions of a node. Relations among four variables (LEVEL, LT1, LT2, and VT) are explained using a schematic of a node shown in Fig. 3, where three kinds of conditions (level overflow, level underflow, and variation overflow) are expressed as:

1. level overflow —LEVEL > LT1,
2. level underflow—LEVEL < LT2,
3. variation overflow—inflow within the unit time > VT.

The meanings of the six variables employed in the expression of node data are as follows.

A shows a node name described in the relation data.

MDTYPE (Modify Type) shows the type of a node. There are five kinds of MDTYPE (0, 1, 2, 3, and 4), by which the manner of relation cut or sleep is defined in respect of node A as shown in Fig. 4. The meanings of five values are as follows.

0 = no change.

1 = inflow relations to A are cut if level overflow or variation overflow occurred.

2 = outflow relations from A are cut if level overflow or variation overflow occurred.

3 = both inflow and outflow relations of A are cut if level overflow or variation overflow occurred.

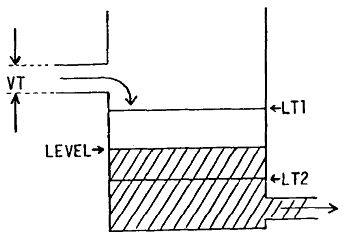

Fig. 3 Schematic Representation of a Node.

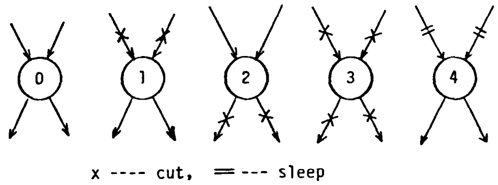

Fig. 4 Cut and Sleep of Relations Due to Five Modify Types.

4 = inflow relations to A sleep if level overflow or variation overflow occurred.

LEVEL shows an initial state value of A. The default value of LEVEL is zero.

LT1 shows the upper threshold level to give a decision of relation cut or relation sleep.

LT2 shows the lower threshold level to give a decision of relation awakening.

VT shows the threshold value to the increase of inflow within the unit time to give a decision of relation cut.

Node data is represented in memory using a HASH TABLE and NODE TABLE as shown in Fig. 5. A node name is converted to a hash value ($1 \sim 4{,}096$) using a hash function. An address of NODE TABLE ($1 \sim 6{,}000$) is stored in an area of HASH TABLE indicated by the obtained hash value. If the HASH Table area indicated by the obtained hash value is occupied (i.e. if a collision occurred), an unoccupied area of the HASH TABLE is searched circularly by adding a unit to the obtained hash value. NP is a pointer which indicates a head address of unused area in the NODE TABLE. Detailed information of the NODE TABLE is shown in the right part of Fig. 5. Hatched areas in the upper right corner are preset by node data and the other areas are used in the calculation of level value while NESSY-IV is running.

The NODE TABLE has 6,000 words, therefore 500 nodes including dummy nodes can be stored at a time, if 12 words are used to store one node on the average.

5.3 Modify Data

If one of the conditions of level overflow, level underflow, and variation overflow occurred, the structure of network flow model is modified depending on the five types of modify data. Expressions of modify data and representations of modify data in memory are shown as follows.

(1) (A/A, B, C, D)—Cut of binary relation.

If level overflow or variation overflow occurred in node A, inflow relations or outflow relations of A, B, C, and D are cut depending on the value of MDTYPE (modify type) of node A.

(2) ($A+A$, B, C, D)—Activation of binary relation.

If level overflow or variation overflow occurred in A, both inflow relations and outflow relations of A, B, C, and D, of which Phase No. are 88, are activated. Phase No. of activated relation is changed

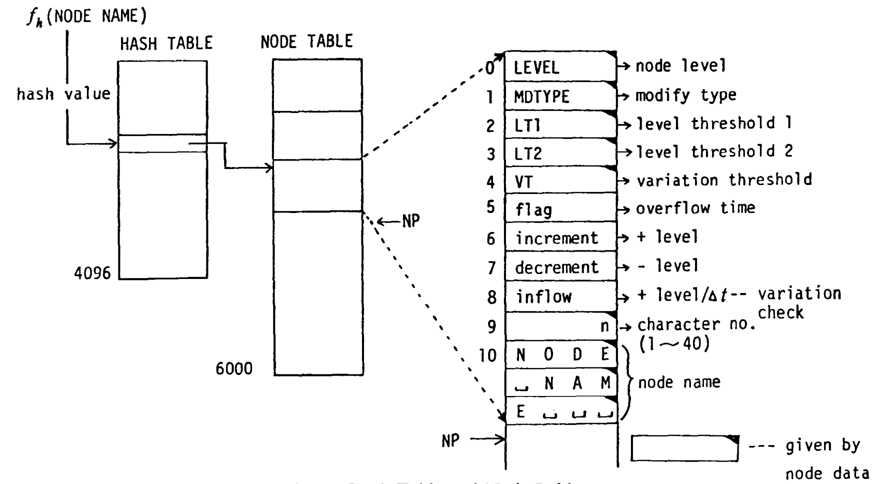

Fig. 5 Hash Table and Node Table.

from 88 to zero and the relation is effective through all phases.

(3) ($A\%A, B, C, D$)—Sleep of binary relation.
If level overflow occurred in node A, active inflow relations of A, B, C, and D, of which Phase No. are 100, are slept, whereas if level underflow occurred in A, sleeping inflow relations of A, B, C, and D, of which Phase No. are 100, are awaked.

(4) ($A\#A, B, C, D$)—Awakening of binary relation.
If level overflow occurred in node A, sleeping inflow relations of A, B, C, and D, of which Phase No. is 99 or 100, are awaked, whereas if level underflow occurred in A, active inflow relations of A, B, C, and D, of which Phase No. are 100, are slept.

(5) ($A>B, C, D, E$)—Sequential activation of binary relation.
If level overflow or variation overflow occurred in node A, a node whose Phase No. is 88, is searched from left to right as: $B \rightarrow C \rightarrow D \rightarrow E$. If an unoverflowing node is found, both inflow relations and outflow relations of the node are activated. The Phase No. of the activated relation is changed from 88 to zero and the relation is effective through all phases.

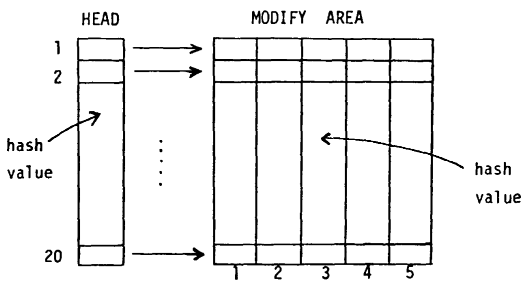

Fig. 6 Head and Modify Area.

Here, A, B, C, D, and E represent node names that are described in relation data.

Five types of modify data are represented in memory using a set of areas (i.e. HEAD and MODIFY AREA) respectively as shown in Fig. 6. Initial values of HEAD and MODIFY AREA are zeroes. If modify data (e.g. ($A/A, B, C, D$)) are given, hash values are obtained from four node names (i.e. A, B, C and D) using a hashing function: f_h (node name), and stored in HEAD and MODIFY AREA as follows.

HEAD(1) $=f_h(A)$,
MODIFY AREA (1, 1) $=f_h(A)$,
MODIFY AREA (1, 2) $=f_h(B)$,
MODIFY AREA (1, 3) $=f_h(C)$,
MODIFY AREA (1, 4) $=f_h(D)$.

5.4 Generation of Dummy Nodes

A lag between two nodes A and B is expressed using a delay time (T) in relation data. If T is greater than or equal to a unit, T dummy nodes are generated by NESSY-IV automatically. For example, following two binary relations: ($A>B$ (3, 20, 1, 1)) and ($A>B$ (2, 80, 1, 1)) are converted as shown in Fig. 7. Here, seven binary relations are generated. In respect of the first two binary relations: ($A>$Z000 (0, 20, 1, 1)) and ($A>$Z003 (0, 80, 1, 1)), a delay time is zero and a transition probability is a given value (i.e. 20 and 80 respectively), while for the other binary relations, delay time is a unit, and transition probability is 100. In order to give a unique definition of node names, the name of dummy node is represented using the letter "Z" and three digits (000~999), thus 1,000 dummy nodes can be generated in a simulation.

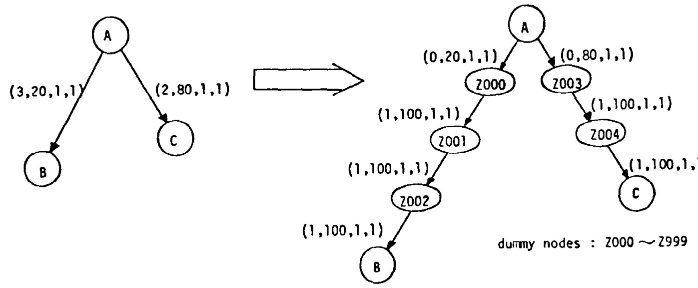

Fig. 7 Generation of Dummy Nodes.

5.5 Calculation of Node Level

Calculations of level variation of nodes are continued till time $=t_e$ repeating two steps (step 1 and step 2) as shown in Fig. 8, where the final time of simulation (t_e) is given by an interactive command "TLt_e". Because of generation of dummy nodes, there are only two kinds of delay time (i.e. zero and unit) between two nodes. Step 1 and step 2 deal with binary relations of which delay time is zero and unit respectively. Calculations in step 1 and step 2 are carried out as follows.

Step 1: In respect to binary relations with delay time of zero, a level variation of a node is calculated depending on level value and P (transition probability or threshold of flux) until total variation converges to zero. After an iteration is completed, level overflow, level underflow and variation overflow are checked. If time $=t_e$ NESSY-IV stops, or goes to step 2.

Step 2: Once for each binary relation with delay time of unit, a level variation of a node is calculated depending on level value and P (transition probability or threshold of flux). After executing step 2, a unit time advances, and returns to step 1.

The amount of level variation of a node is calculated from node level and P. The obtained level variation is increased or decreased to each node level. For example,

Time	Step
0	0 (step 1)
	+1 (step 2)
1	0 (step 1)
	+1 (step 2)
2	0 (step 1)
	+1 (step 2)
3	0 (step 1)
⋮	⋮
	+1 (step 2)
t_e	0 (step 1)

Fig. 8 Sequences of Execution.

the level variation for the following two binary relations is calculated as:

(1) ($A>B$ (T, P, Phase No., TYPE)): the rule of mass conservation holds.

For the case of TYPE = 1, the mass flow is calculated as follows:

$$A(i+1)=A(i)-A(i)*p, \quad B(i+1)=B(i)+A(i)*p.$$

Here, p is a rate of P to the total flow from node A. For the case of TYPE = 2, the mass flow (dv) from node A to node B is calculated as follows:

$$\text{if } A(i)\geqq \Sigma P \text{ then dv}=P(i),$$
$$\text{else dv}=A(i)*P(i)/\Sigma P,$$
$$A(i+1)=A(i)-\text{dv}, \quad B(i+1)=B(i)+\text{dv}.$$

Here, $P(i)$ represents the value of threshold of flux from node A to node B. A schematic representation of output function for TYPE = 1 and TYPE = 2 is shown in Fig. 9.

(2) ($A=B$ (T, P, Phase No., TYPE)): no rule of mass conservation holds. Here, parameter TYPE has no meaning except for TYPE = 1.

$$A(i+1)=A(i), \quad B(i+1)=B(i)+A(i)*p.$$

Generally, different results of simulation are obtained if the order of calculation of binary relations is different. In order to obtain unique results of simulation, independently of the order of calculation of binary relations, it is necessary to separate a calculation of level variation in step 1 and step 2 into the following two parts.

(1) In both step 1 and step 2, increments and decrements of level variation in each node are

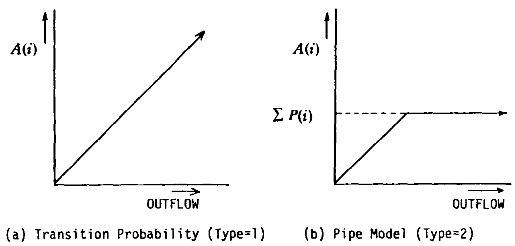

Fig. 9 Schematic Representation of Output Function for Type = 1 and Type = 2.

accumulated using an increment area and a decrement area shown in Fig. 5.

(2) Thereafter a node level is calculated using a value of level variation which has been accumulated in an increment area and in a decrement area.

Separating a calculation into two parts (i.e. an accumulation of level variation and its calculation) as described above, level values of each binary relation have been calculated at the same time, by which a unique result of simulation is obtained independently of the order of calculation of binary relations. In step 1, the calculation is continued in respect of binary relations for delay time of zero until a level variation converges to zero. Here, the criterion of convergence concerns not with binary relations as "$A=B$" but with binary relations as "$A>B$". The iteration in step 1 completes when the amount of absolute value of level variation converges to zero.

5.6 Interactive Commands

NESSY-IV is an interactive simulator for a network flow model, which works under a TSS mode of FACOM M-200 OS IV/F4. A simulation is carried out using conversational commands as shown in Table 1, where commands and their functions are given. Commands are composed of one or two command letters (e.g. DM, *E*, QC etc.) and arguments follow if necessary. In Table 1, "element name" (cf. command *M*, *N*, or *R*) represents an element name in a partitioned organization file. A concrete example of conversation is shown in the paper [7] which deals with the study of "the mass evacuation from downtown area after an earthquake warning" [8].

Table 1 Commands and Functions.

	Commands	Functions
1.	DM	Dump modify data
	DN	Dump node data
	DR	Dump relation data
2.	E	Terminate an execution of the NESSY-IV
3.	G	Start on a simulation
4.	I	Initialize the NESSY-IV
5.	M<u>element name</u>	Read modify data (sepcified by <u>element name</u>) from a file
6.	N<u>element name</u>	Read node data (specified by <u>element name</u>) from a file
7.	PH ($t_1, t_2, \cdots, t_{10}$)	Set phase interval using ten integer values t_i
	PIt_n	Set an interrupt time (t_n)
8.	QC	Querry terminal time and an interval time for monitoring
	QN	Querry informations on nodes
	QP	Querry phase interval data
	QR	Querry the state of used system resources.
9.	R<u>element name</u>	Read relation data (specified by <u>element name</u>) from a file
10.	TLt_e	Set a terminal time (t_e) of a simulation
	TMt_m	Set an interval time (t_m) for monitoring

6. Summary and Conclusion

In order to analyze effective countermeasures to prevent *systems panic* which will be caused by mass evacuations from a downtown area after an earthquake warning a network flow simulator: NESSY-IV has been developed. Simplicity in the description of network flow model, flexibility in the modification of network flow model during a simulation run, and efficiency in the execution are major characteristics of NESSY-IV. A network flow model for mass evacuation is represented by three kinds of data: relation data, node data, and modify data, by which interactions among two types of flows, i.e. mass flows and information flows, are described and analyzed.

Relation data is used to define characteristics of flow between two nodes such as: type of flow (mass flow or information flow); delay time; transition probability or threshold of flux; step interval during which the flow is effective, or characteristics of flow; and a mode of output function applied to the flow.

Node data is used to prescribe the physical capacity of a node using three parameters (upper threshold, lower threshold, and inflow threshold), by which changes of conditions in a node such as: level overflow, level underflow and variation overflow, are represented. In addition, a node has a parameter on a modify type which indicates the manner of modification for the simulation model.

Modify data is used to prescribe the way in which the simulation model is modified due to the changes of conditions occurred in a node.

As a case study, NESSY-IV is applied to the simulation: "Mass Evacuation from Sakae Area at Nagoya City after an Earthquake Warning", from which several countermeasures to prevent *systems panic* are obtained [8]. From these results, it is concluded that NESSY-IV is an effective network flow simulator for the analyses of mass evacuation in case of emergency.

Acknowledgements

The author would like to thank Professor Y. Shimazu of Department of Earch Sciences, Nagoya University, for many useful suggestions he has made. I also wish to thank Professor T. Fukumura of Department of Information Engineering, Nagoya University, for his constant interests in my work.

References

1. Forrester, J. W. Industrial Dynamics, M.I.T. Press (1961).
2. Forrester, J. W. Principles of Systems, Wright-Allen Press (1971).
3. Hiramatsu, T. and Fukumura, T. Interactive Processing System for Net Thesaurus (NESSY-I), Johoshori (*Trans. Information Proc. Soc. Japan*), **20**, 3 (1979) 265–271 (in Japanese).
4. Hiramatsu, T. and Shimazu, Y. Net Structure Analyzing System of Causal Relations—NESSY-II, SMLES Contribution 131, (1979) pp. 21 (in Japanese).
5. Hiramatsu, T. Implementation of Dynamic Net Structure Analyzing System NESSY-IV and its Application, Joint Computer Conference of Seven Computation Centers in Japan, 1, 51–59 (1979).
6. Hiramatsu, T. and Fukumura, T. Dynamic Net Structure Analyzing System of Causal Relations—NESSY-III, Johoshori (*Trans. Information Proc. Soc. Japan*), 21, 3 (1980) 191–199 (in Japanese).
7. Hiramatsu, T. Dynamic Net Structure Analyzing System of Causal Relations NESSY-IV, Memoris of Research and Development Division, Computation Center, Nagoya University, 6, (1980) 27–46.
8. Hiramatsu, T. Urban Earthquake Hazards and Risk Assessment of Earthquake Prediction, Jour. Physics of the Earth, **28**, 2, (1980) 43–84.
9. Pritsker, A. A. B. Modeling and Analysis Using Q-GERT Networks, 2nd ed. A Halsted Press Book, John Wiley & Sons, 1979.
10. Pugh, A. L., III, DYNAMO II User's Mannual, M.I.T. Press, 1970.
11. Roberts, S. D., Schier, J. and Sulanke, T. An Introduction to Modeling with INS, Proc. of the 1978 Winter Simulation Conference, (1978) 343–354.
12. Shimazu, Y. and Hiramatsu, T. Earthquake Hazards and Risk Assessment of Warning Issue, Zisin (J. Seismol. Soc. Japan), **31** (1979) 147–160 (in Japanese).
13. Shimazu, Y. and Hiramatsu, T. Earthquake Hazards and Risk Assessment of Warning Issue, International Symposium on Earthquake Prediction, UNESCO SC-79/Conf. 801/Col. 14/IV-3, (1979) pp. 16.
14. Sugiyama, K., Hiramatsu, T. and Shimazu, Y. Toward Development of Earthquake Disaster Relational Model, Proc. International Conference on Cybernetics and Society, 1 (1978) 788–793.

(Received July 2, 1980)