View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by International Institute for Applied Systems Analysis (IIASA)

International Institute for Applied Systems Analysis
IIASA www.iiasa.ac.at

STRUM - An Interactive Computer System for Modeling Binary Relations

Ganin, I.A. and Solomatin, D.P.

IIASA Collaborative Paper
November 1984

Ganin, I.A. and Solomatin, D.P. (1984) STRUM - An Interactive Computer System for Modeling Binary Relations. IIASA Collaborative Paper. Copyright © November 1984 by the author(s). http://pure.iiasa.ac.at/2525/ All rights reserved. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage. All copies must bear this notice and the full citation on the first page. For other purposes, to republish, to post on servers or to redistribute to lists, permission must be sought by contacting repository@iiasa.ac.at

NOT FOR QUOTATION
WITHOUT PERMISSION
OF THE AUTHOR

STRUM – AN INTERACTIVE COMPUTER SYSTEM FOR MODELING BINARY RELATIONS

I.A. Ganin
D.P. Solomatin
All-Union Institute for Systems Studies, Moscow

November 1984
CP-84-52

Collaborative Papers report work which has not been performed solely at the International Institute for Applied Systems Analysis and which has received only limited review. Views or opinions expressed herein do not necessarily represent those of the Institute, its National Member Organizations, or other organizations supporting the work.

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
2361 Laxenburg, Austria

PREFACE

System identification and specification is essential in every systems study. The development of structural models (which describe the geometric relationships between the elements of the system) is an important part of this procedure. However, when the system is very complex or the number of elements is large it becomes difficult to construct such models without some technical assistance. In this paper, the authors describe an interactive computer system called STRUM which facilitates the structural modeling process. The use of the system (which is implemented on the IIASA VAX 11/780 computer) is illustrated by application to a specific example.

This paper is a contribution to research currently underway in the System and Decision Sciences Program.

A.B. Kurzhanski
Chairman
System and Decision Sciences
Program

STRUM – AN INTERACTIVE COMPUTER SYSTEM FOR MODELING BINARY RELATIONS

I.A. Ganin and D.P. Solomatin
All-Union Institute for Systems Studies, Moscow, USSR

1. INTRODUCTION

A necessary step in every complex systems study is to identify the system and specify its structure. This step is never omitted, although the structural aspects are often not considered explicitly but rather taken into account in some intuitive fashion.

The term *structural modeling* is used to describe those modeling activities in which the intention is to embody the geometric rather than the algebraic aspects, to describe form rather than to calculate or measure quantitative output. Thus, structural modeling is concerned with the relationships between the various elements of the system.

However, it is virtually impossible to construct structural models by hand when, for example, the number of elements is large or a multidisciplinary approach is adopted. To help to overcome these problems an interactive computer system called STRUM which facilitates the structural modeling process has been developed.

This paper describes the theoretical basis of STRUM and explains how it may be used in practice.

2. THE STRUCTURAL MODELING PROCESS

The process of structural modeling involves four steps:

1. The main elements of the system to be modeled are identified.
2. The relations by which it is desirable to connect the elements are selected.
3. Structural models are constructed on the basis of the chosen set of elements and relations.
4. The structural models are analyzed, corrected and verified.

If V is the set of elements and the relation R between them is assumed to be binary, then we may introduce $G(V,R)$ as a directed graph (or graph for short) describing the structural model. In this way we restrict our attention to the class of binary relations, allowing only pairwise relationships between the elements. There are several reasons for this:

- in many cases binary relations are basic and represent the simplest components of higher-order relations;
- binary relations are widely used by experts in many disciplines and structural models based on them are easily understood;
- the process of binary relation construction may be computerized and optimized.

Structural models involving binary relations (graphs) are used quite widely and there are many ways of generating their elements (brainstorming, DELPHI, etc.).

In most cases the element generation and structuring phases overlap as, for example, in decomposition procedures. Warfield [1] seems to have been the first to suggest that element generation and structuring should be separated, and the most suitable techniques used for each phase. This is especially important when an interdisciplinary expert group is participating in the structural modeling process.

Experience has shown that there are a number of identifiable situations in which it is difficult or even impossible to structure a model by hand and a computer can be useful:

1. Situations in which the number of elements is large and problems of scale arise.
2. Situations in which the elements are related in a complex way and it is difficult to maintain and analyze their interconnections.
3. Situations in which structuring is performed by a multidisciplinary team of experts and a systematic procedure for organizing their work is necessary.
4. Situations in which the participants in the structural modeling process are far apart and it is necessary to organize a computerized teleconference.

Of course, these are not the only situations in which a computer can help; they do however summarize the cases in which a computer is either essential or can considerably improve the quality of the modeling.

It is generally accepted that the human ability to perceive, remember and process large numbers of data is limited. For this reason, one of the most reliable means of obtaining expert judgements is to ask the experts to compare pairs of objects in a systematic fashion.

The properties of the relation are also important here. If the binary relation is transitive it becomes possible to significantly reduce the number of questions which must be put to the experts before the structure can be determined. The basic idea is quite simple: if it is known that $a\ R\ b$ and $b\ R\ c$, then it is not necessary to ask whether $a\ R\ c$ is true – it must be, because of transitivity. This information should be inferred by the computer.

The main reasons for our preference for transitive relations (transitive graphs) in structural models are the following:

- transitive relations are common in the analysis of social systems (objective trees, graphs of PERT networks, etc.);
- the construction of a structural model involving transitive relations may be optimized. The optimization boundaries are determined by the completeness of the transitive inferences from the experts' answers;
- the procedures and algorithms developed for the optimization process outlined above may in some cases be used to construct a structural model based on a non-transitive relation R. In this case the expert group first interactively forms a structural model using the "transitive analog" $\hat{R}$ of the relation R and this structural model is then corrected by the group.

Two main principles were therefore built into the STRUM structuring algorithms:

1. Experts are asked by the computer to answer questions about the links between only two elements at a time – this method of pairwise comparison concentrates the experts' attention and is the most reliable way of structuring expert information (if the experts disagree, the majority view should be accepted).
2. The number of questions put to the expert group should be as small as possible (this is a common and natural requirement in interactive systems).

3. USE OF STRUM IN THE STRUCTURAL MODELING PROCESS

After the elements have been selected and the relation R has been chosen, their verbal representations are stored in the computer. The construction of the structural model is then an interactive procedure, which takes the following form.

STRUM first puts questions to the expert group in quasi-natural English: "Indicate the direction of the relation R between elements a and b" (where R, a and b are verbal representations). On obtaining the answer STRUM automatically introduces the transitive inferences into the structural model (if the relation R is assumed to be transitive). Then, using special optimizing rules, STRUM chooses the next pair of elements to be considered, asks the next question and the process is repeated until the structural model is complete.

When the model is complete it must be analyzed and corrected. Visual analysis is aided by a graphics facility which displays the graph of the structural model on either the terminal screen or a graph plotter. Automatic analysis includes the identification of levels, cycles, subgraphs on subsets of elements, reachability subsets, etc. Individual elements and their links or groups of elements and their links may be added or deleted during the correction process.

These and other options for analysis and correction of the structural model may be selected through an interactive on-screen dialogue: a simple command language allows quick activation of the various STRUM procedures. It is also possible to interrupt an interactive session and continue it later without any loss of information.

4. AN ALGORITHM FOR INTERACTIVE CONSTRUCTION OF A TRANSITIVE GRAPH

4.1. Posing the problem

Assume the set V of elements to be given. The problem is then to construct a directed graph from experts' answers to questions about the links between pairs of elements (in terms of the transitive relation R). It is also necessary to minimize the total number of questions asked. (We shall assume that the relation R is also reflexive so that R represents a partial quasi-order.)

Similar problems arise in sorting (where V is a set of numbers and R is the linear order), in decision theory (where R is a preference relation which may or may not be transitive) and other fields.

The interactive reconstruction of a partial quasi-ordered system was first considered in J.N. Warfield's pioneering work on structural modeling (see [1]). He suggested that an algorithm based on the sorting algorithm QUICKSORT [2] could be used for this purpose.

Unfortunately our experience has shown that his algorithm is not flexible enough to be useful in the construction of large structural models (for example, it does not allow the structural model to be corrected during the modeling process). In addition, its computer memory requirements are very considerable.

4.2. Algorithmic skeleton (algorithm for successive d-transitive closure)

The algorithm for successive d-transitive closure (d-ASTC) described briefly in the following section is considered in more detail in [3]. It is based upon the following skeleton:

1. Choose any pair of elements. Go to step 3
2. Choose a pair of elements according to certain "quality" criteria
3. Ask the experts about the relationship between these two elements
4. Add this information to the graph
5. Identify all the logical inferences from the answer and add them to the graph (d-transitive closure)
6. If the graph is not complete then go to step 2
7. Stop

The main problems arise on steps 2 (choosing the "best" pair of elements) and 5 (identifying all the inferences).

4.3. Identifying all transitive inferences

Conventional transitive closure (TC) of a graph (relation, binary matrix) assumes that

$$aRb \;\&\; bRc \rightarrow aRc$$

One possible and indeed frequent answer obtained in the process of graph reconstruction is $a\bar{R}b$, i.e., no connection between a and b. This information is not used by any of the TC algorithms, which assume that the graph is complete and no connections will be added or removed.

However, it is easy to see that this answer implies

$$a\,\bar{R}\,b \;\&\; c\,R\,b \rightarrow a\,\bar{R}\,c$$

$$b\,R\,a \;\&\; b\,\bar{R}\,c \rightarrow a\,\bar{R}\,c$$

These statements yield information about the impossibility of the connection $a\,R\,c$.

If it is known that $a\,\bar{R}\,c$ then we can say that $a\,Q\,c$ is true and call this a 0-connection. Let connections of type $a\,R\,b$ be called 1-connections, where R and Q are the corresponding adjacency matrices. Using this information about the 0-connections leads to a decrease in the total number of questions asked.

It is proved in [3] that the following algorithm enters all possible inferences in the adjacency matrices of the graph (in R and Q).

d-transitive closure algorithm

```
1.  Transitive closure $\hat{R}$ of matrix $R$;
2.  For $k = 1$ to $|V|$ carry out the following steps:
        for $i = 1$ to $|V|$ do if $(\hat{R}_{ki} \vee Q_{ik})$ then
            for $j = 1$ to $|V|$ do if $(Q_{ij} = 0)$ then
                if $(\hat{R}_{ki} \;\&\; Q_{kj} = 1)$ then $Q_{ij} \leftarrow 1$
                if $(Q_{ik} \;\&\; \hat{R}_{jk} = 1)$ then $Q_{ij} \leftarrow 1$
```

4.4. Choosing the "best" question

Steps 2–3 of algorithm d-ASTC are performed many times during the construction of the graph and so statistically we have a repeating event with four possible outcomes (answers) with different probabilities. If the question concerns elements i and j then these answers are:

1. No connection, i.e., $i\,Q\,j \;\&\; j\,Q\,i$;
2. Connection from i to j, i.e., $i\,R\,j \;\&\; j\,Q\,i$;
3. Connection from j to i, i.e., $i\,Q\,j \;\&\; j\,R\,i$;
4. Bilateral connection, i.e., $i\,R\,j \;\&\; j\,R\,i$.

The basic principle used in developing the choice criterion is to count all the 1- and 0-inferences for each answer and then choose the question with the maximum potential number of inferences.

We deduce that 1-connections have higher priority – it is not difficult to show that adding a 1-connection (i,j) to the graph at step 1 of the d-TC algorithm implies the entering of a 0-connection (i,j) at step 2. This means that no question about the pair (i,j) will be necessary later. The same is not true for 0-connections, but experience shows that if a 0-connection (i,j) is entered by experts or inferred by the algorithm, it is in many cases complemented by a connection (j,i).

On the basis of this argument we develop four partial criteria for every answer. Since the answer is not known *a priori*, we have to introduce subjective probabilities for each answer (which change at every iteration). The full criterion is then constructed as the weighted sum of the partial criteria.

Other graph construction methods based on the d-TC algorithm may also be implemented.

4.5. Comparison of various algorithms

We compared three algorithms:

(i) a modified Warfield algorithm [1];

(ii) an algorithm based on conventional transitive closure [4];

(iii) the d-ASTC algorithm.

To compare the average numbers of questions put by the algorithms, a simulation experiment was conducted in which binary matrices filled by a random number generator were used as the input (expert answers). The ratio of the number of units to the total number of matrix entries was assumed to be approximately equal to this ratio for matrices of transitively closed trees.

The experiment showed that, on average, d-ASTC suggests 15% questions less than the modified Warfield algorithm, 25% less than the algorithm based on successive conventional transitive closure and 40% less than the method of pairwise comparisons. In practice this last figure was more like 50–70%.

Other criteria for comparison were: user friendliness, closeness of the computer-built model to the user's mental model, and ease of computer realization.

As a whole the comparison showed that the algorithm d-ASTC has considerable advantages over the others for use in the structural modeling process.

5. ALGORITHMS FOR THE ANALYSIS AND GRAPHICAL REPRESENTATION OF TRANSITIVE GRAPHS

5.1. Graph analysis

There are many different algorithms available for graph analysis. Our choice was based on two competing criteria: operating speed and computer memory requirements.

The algorithm used to identify graph levels (an algorithm for topological sorting) is taken from [2]. It is only applicable to acyclic graphs and so in the case of cyclic graphs the cycles are deleted before using the algorithm (but after storing). When identification of the levels is complete the cycles are restored. The complexity of the algorithm is $O(n^2)$.

Cycles (strong components) may also be found in $O(n^2)$ operations or even in $O(n)$ operations [2]. The main idea of the algorithm is to identify all the pairs of elements for which $R_{ij} = R_{ji} = 1$.

We used Warshall's algorithm [2] for transitive closure – this requires $O(n^3)$ operations. (There are more efficient algorithms available but these are much more memory-consuming.) Because of the compact bitwise storage of the matrices, the operation OR (which is repeated in the algorithm n^2 times on two matrix columns) is actually performed not upon n bits of the column but rather on a small number of words.

This is also true for the transitive reduction algorithm [5], whose theoretical complexity of $O(n^3)$ is considerably reduced in practice. This algorithm requires additional computer memory which is related to the size of the matrix being processed.

5.2. Pictorial representation (depiction) of the graph

The set of algorithms which draw the graphs forms the main part of the depiction subsystem. Input for these algorithms comes from the (transitively reduced) adjacency matrix R or some chosen part of it. Output (the picture of the graph) is directed either to the terminal screen or to a graph plotter, as requested by the user (see [6]).

These algorithms perform the following tasks:

(i) they reduce the number of cross-overs of lines connecting elements;

(ii) they lay out the elements (which are represented as squares) in such a way that the whole picture is clear and legible;

(iii) they send the final result to the chosen output device.

The layout algorithm formalizes certain heuristic rules for "good" layout which appear to be similar to those used in [7,8]. Reduction of the number of crossings is performed by a specially designed algorithm described in detail in [3].

It should be mentioned that one such graphics system is described in an IIASA publication [8]. We feel that it would be useful to conduct a study on the joint use of this system and STRUM at IIASA.

6. IIASA VAX-11/780 IMPLEMENTATION

6.1. Some notes on the storage of the structural model in the computer memory

There are several different methods of graph representation (based on incidence matrices, adjacency matrices, lists of arcs, and adjacency sets) – we decided to use the adjacency matrix method. There were two main reasons for this:

(i) the number of arcs in the graph changes at every step of the interactive process, this number is not known *a priori* and therefore other methods of graph storage are more memory consuming;

(ii) the STRUM algorithms refer to the arcs of the graph in terms of the numbers of the vertices connected by the arc and so the adjacency matrix method is more convenient.

The matrices are stored in packed bitwise form. Access to every bit entry is achieved by activation of specially written FORTRAN subroutines.

6.2. STRUM subsystems

STRUM can be disaggregated into a number of subsystems with distinct functions:

- a structuring subsystem consisting of the algorithms for synthesis of the structural model;
- a subsystem for linking structural models;
- a subsystem for analysis of the structural model (defining levels, cycles, subgraphs, carrying out transitive reduction, etc.);
- a subsystem for correction of the structural model (removing and adding connections and elements, splitting and merging elements);
- a subsystem for dialogue support (interpreting instructions and activating other subsystems);
- a subsystem controlling communication with the computer disc (to provide the possibility of restarting an interactive session after a break and to record the steps in the dialogue);
- a depiction subsystem (uses subroutines from [6]).

6.3. Use of STRUM on the IIASA VAX-11/780

In order to use the STRUM system the user[*] must first move to the directory */uc/ext/solo* (the user's input is given in **bold type**):

```
% cd /uc/ext/solo
```

He must then check that the files *strum*, *comput.str* and *messagee.str* are present in this directory. The files connected with STRUM are shown in Fig. 1, and their functions described below.

strum	executable file
comput.str	information about the computer operating system
messagee.str	STRUM messages
perm.str	information on the structural models developed so far (necessary for a restart) – this will be called the permanent file
smname.est	verbal representations of the relation and elements of a structural model called "smname"

[*]By "user" we mean one or more experts working on a specific structural model.

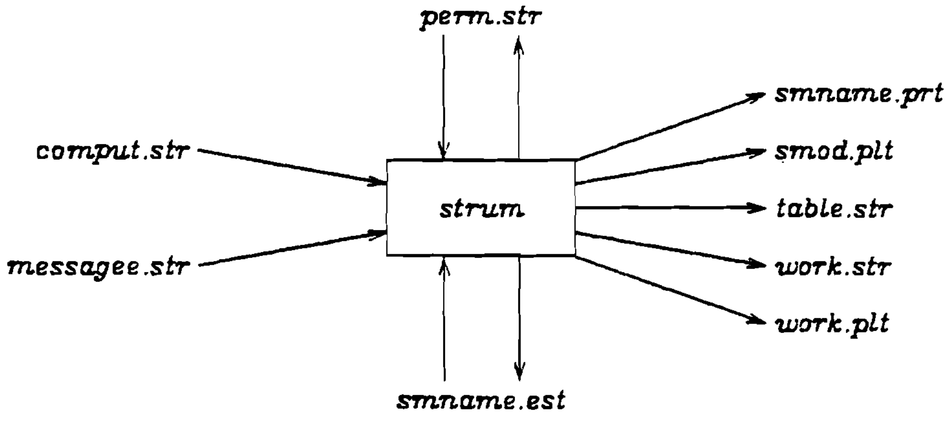

Figure 1. STRUM files.

smname.prt	protocol of user—computer interaction in the development of a structural model called "smname"
smod.plt	picture of the graph in "di-format"
table.str	table of hierarchical levels (ready for printing)
work.str	working file
work.plt	working file

The Appendix to this paper contains a description of an example in which STRUM was applied to one of the World Health Organization programs. We shall now illustrate the use of STRUM with the help of this example.

Suppose that the user intends to construct a structural model called *tobac* consisting of 21 elements. These elements represent legislative, health-care and other actions aimed at reducing tobacco smoking. The binary (transitive) relation chosen is "action A helps to achieve action B."

In this case the user should use one of the UNIX text editors to prepare the file *tobac.est* (consisting of 22 lines which may be wrapped):

Indicate by typing one of the symbols a, v, x, 0 whether one of the following actions helps to achieve the other:

Ban, or at least restrictions, on production, sales

Ban on products yielding certain levels of a particular substance

Development of less hazardous materials to be used as smoking agents

. . .

. . .

. . .

Making special information available to parents of "vulnerable" children

(The full list of elements may be found in the Appendix; the meanings of the symbols **a**, **v**, **x**, **0** are given later in this section.)

Preparation of the file *tobac.est* is optional. If STRUM cannot find this file in the directory, the elements of the structural model will be referred to only by their numbers.

Each element should occupy only one line (long lines are wrapped by the text editor) and must not contain more than 132 characters. The total number of elements should not exceed 200.

Before invoking STRUM it is advisable to make a copy of the file *tobac.est*, delete the first line of the copy and obtain a printout of the resulting file with line numbers (using the UNIX command **p -number**).

A typical interactive session could then proceed as follows (the user's input is given in **bold type**):

```
% strum
- *** structural modeling system ***
- computer type:
IIASA VAX-11/780
* enter line number
    1. begin new model
    2. continue existing model
1
* model name?:
tobac
* your name & phone number?:
solomatin, 224
- model is activated
- opening new protocol
* number of elements?:
21
* enter or remove connections? (line number)
    1. enter new connections
    2. remove connections
    3. remove bilateral connections
    4. exit
1
```

```
— entering new connections
* elements from (* means exit)?:
17,20
* elements towards (* means exit)?:
14
— entering new connections
* elements from (* means exit)?:
13
* elements towards (* means exit)?:
7
— entering new connections
* elements from (* means exit):
*
```

Indicate by typing one of the symbols a, v, x, 0 whether one of the following actions helps to achieve the other:
(13) Health education of influential groups (politicians, sportsmen, actors, etc.)
(14) Ban on the availability and use of tobacco in health facilities

v

Indicate by typing one of the symbols a, v, x, 0 whether one of the following actions helps to achieve the other:
(13) Health education of influential groups (politicians, sportsmen, actors, etc.)
(17) Education in schools to prevent children from starting the habit

v

Indicate by typing one of the symbols a, v, x, 0 whether one of the following actions helps to achieve the other:
(13) Health education of influential groups (politicians, sportsmen, actors, etc.)
(20) Encouragement of peer group pressure against cigarette smoking habits

0

---

and so on.

Questioning is continued until the graph is complete. The symbols **a**, **v**, **x**, **0** have the following meaning:

a	connection from the second element to the first (connection upwards on the screen)
v	connection from the first element to the second (connection downwards on the screen)
x	bilateral connection
0	no connection

The systematic procedure of graph construction may be interrupted by entering ? as the answer instead of **a**, **v**, **x** or **0**. In this case STRUM responds:

SMS>

and waits for the user to enter a command.

The most frequently used STRUM commands are:

c/ac	add connection
c/dc	delete connection (or note its absence)
c/ae	add element
c/db	delete bilateral connection (or note the absence of any connections between two elements)
t	display the table of hierarchical levels on the screen
t/p	send the table of hierarchical levels to the file *table.str*
pic	activate the depiction subsystem
subpic	pick out any subgraph and perform the **pic** command
seq=1	display transitive inferences after every answer
seq=0	do not display transitive inferences after every answer
involved	display the numbers of the elements already involved in the structuring process
par	display the values of the parameters (number of elements, session number, number of question, etc.)
mdl	display the names of the structural models stored in the permanent file *perm.str* (they may also be accessed conventionally)
return	return to interactive construction of the structural model

store	store the results of the current session (changed model) in the permanent file *perm.str* and remain in STRUM
stop	store the changed model in the permanent file *perm.str* and leave ·STRUM. On re-entering STRUM the dialogue will begin from the point of interruption.

The full list of commands may be displayed on the screen by entering **?** as a command:

SMS> ?

The above commands may be entered not only during the graph construction process but also in response to most of the other STRUM questions to the experts. Pressing the RETURN key as an answer (without entering any command) in most cases causes the previous question to be repeated or the current question to be retyped.

Almost all of the STRUM messages are stored in the file *messagee.str*. If the user considers any of these messages cumbersome and wishes to clarify them he may modify them using one of the UNIX text editors. The only restrictions are:

(i) message length must not exceed 80 characters;

(ii) lines in *messagee.str* must not be permuted or deleted because STRUM refers to these messages by the line numbers.

It may be seen from the list of commands that there are two ways of obtaining a representation of the structural model:

(i) using the **t** or **t/p** command;

(ii) using the **pic** or **subpic** command.

The table of hierarchical levels shows the numbers of elements situated at each level, their connections with elements at the lower levels and any cycles. The command **t** sends this table to the screen; command **t/p** sends it to the file *table.str* which may be printed after leaving STRUM by the UNIX command **p**.

The command **subpic** activates subroutines which ask the user about the elements to be included in the subgraph and pass this subgraph to the depiction subsystem.

The command **pic** activates the subsystem which depicts the structural model in the form of a graph. The subsystem asks the user about the device to which the picture will ultimately be sent. The result is in "di-format" (see [6]) and is stored in the file *smod.plt.*

It should be mentioned that this file may be prepared for printing/display on any graphics device. The user only has to check that the page size of this device is not smaller than the page size of the device already indicated to the depiction subsystem.

A detailed description of how a file in "di-format" should be prepared for printing is given in [6]. For example, to send the picture to the BBC-plotter it is necessary to use the following UNIX command:

```
% di-bbc < smod.plt | p -pri:bbc
```

To display the picture on the graphics terminal the file *smod.plt* should be sent to this terminal by the UNIX command **cat**. The user may then re-enter STRUM and correct the structural model or continue its construction.

It should be noted that each time the depiction subsystem is used the old files *table.str* or *smod.plt* are overwritten. To save these files it is necessary to change their names.

ACKNOWLEDGEMENTS

The authors are grateful to Prof. A. Wierzbicki for his support and interest in this work, to Dr. A. Umnov for his constant practical help and fruitful discussions, and to Prof. A. Kochetkov and Dr. A. Yashin for their valuable advice and cooperation in implementing the STRUM system at IIASA.

We would also like to express our gratitude to Prof. J. Warfield (University of Virginia, USA), who drew the attention of one of the authors to structural modeling.

Finally, the authors wish to thank all SDS and Computer Services staff for their help.

REFERENCES

1. J.N. Warfield. *Societal Systems: Planning, Policy and Complexity.* N.Y.: Wiley, 1976.

2. E. Reingold, J. Nivergelt and N. Deo. *Combinatorial Algorithms: Theory and Practice.* Englewood Cliffs: Prentice-Hall, 1977.

3. D.P. Solomatin. Mathematical concepts for an interactive structural modeling system. Preprint. Moscow: VNIISI, 1982 (in Russian).

4. I.A. Ganin and D.P. Solomatin. Constructing structural models through dialogue with a computer. In: *Control Issues in Engineering, Economics and Biology.* Moscow: Nauka, 1981 (in Russian).

5. A. Aho, M. Garey and J. Ullman. The transitive reduction of a directed graph. *SIAM J. Comput.*, 1 (1972) 131–137.

6. B. Schweeger. *Description of the Library of Graphics Subroutines.* Laxenburg: IIASA, 1982.

7. M.-J. Carpano and A. Leduc-Leballeur. Interactive computer graphics in computer-aided decision analysis. EUROGRAPHICS 80. Proc. Int. Conf. and Exhibition, Geneva, Switzerland, 3–5 Sept. 1980. Amsterdam: North-Holland, 1980.

8. K. Sugiyama. Drawing and understanding systems structures: an introduction to the SKETCH system. IIASA Working Paper WP-82-97. Laxenburg: IIASA, 1982.

9. *An Integrated Programme for the Prevention and Control of Noncommunicable Diseases.* World Health Organisation. Report NCD/82.2. 1982.

APPENDIX: APPLICATION OF STRUM TO A SPECIFIC PROBLEM

The authors have considerable experience in using STRUM for management purposes, particularly in regional development, program planning, designing large-scale organizational structures, and the management of East–West scientific and technological cooperation.

At IIASA the STRUM system was used to structure of one of the World Health Organization programs – an integrated program for the prevention and control of noncommunicable diseases [9]. We chose to study the proposed subprogram aimed at reducing tobacco smoking. The list of possible actions designed to curb smoking is as follows:

1. Ban, or at least restrictions, on production, sales
2. Ban on products yielding certain levels of a particular substance
3. Development of less hazardous materials to be used as smoking agents
4. Ban on the sale of tobacco products to certain persons
5. Restriction on the use of tobacco in certain places
6. Modify price by taxation
7. Exclude tobacco from retail price index
8. Differentiate taxes by type of tobacco product
9. Selective restrictions or bans on advertizing
10. Restriction of promotion of events (art, sport, etc.) by tobacco companies
11. Health education of the public, pressure groups, campaigns
12. Growth of individual anti-smoking clinics
13. Health education of influential groups (politicians, sportsmen, actors, etc.)
14. Ban on the availability and use of tobacco in health facilities
15. Education directed at health personnel
16. Development and use of specific tobacco-related diagnostics
17. Education in schools to prevent children from starting the habit
18. Support to voluntary organizations in anti-smoking campaigns
19. Self-help anti-smoking (for smoking cessation) groups
20. Encouragement of peer group pressure against cigarette smoking habits
21. Making special information available to parents of "vulnerable" children

It was decided to structure this set using the relation "action A helps to achieve action B".

The beginning of an interactive dialogue aimed at developing a structural model from these elements and relation was given in Section 6.3. The session lasted for 40 minutes and 74 questions were asked. The resulting structural

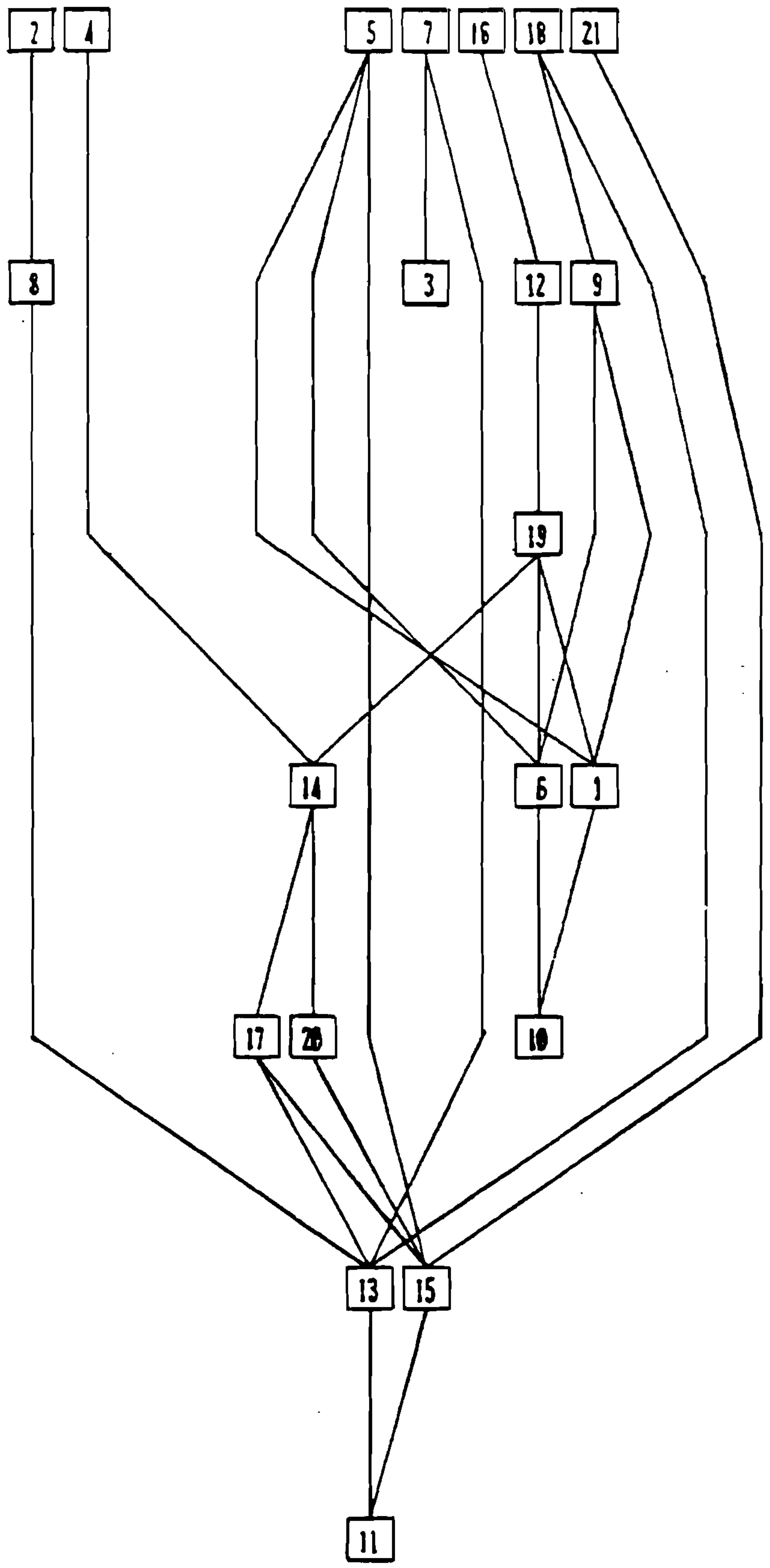

Figure 2. The structural model TOBAC.

model showing the relations between different actions designed to curb smoking is presented in Fig. 2.

This approach made it possible to clarify the interrelationships between elements, to single out the critical actions which affect many of the others, to identify possible ways of carrying out these actions and the connections between them.

The resulting structural model allows us to identify an organization responsible for carrying out each action. In this case the structural model may be used to develop the organizational structure for program management.

The example considered here is purely illustrative; the advantages of using STRUM in the structural modeling process are seen much more clearly when the structural model consists of several dozen elements. However, the application of STRUM even to this rather simple example demonstrates the potential of the system for use in various areas of systems analysis.