Manipulations of the model structure

Editors: prof.dr. K.M. van Hee
prof.dr. H.G. Sol

MANIPULATIONS OF THE MODEL STRUCTURE

by

M.W.J. van Kraaij
W.Z. Venema
J. Wessels

NFI 11.88/08

EINDHOVEN UNIVERSITY OF TECHNOLOGY
F. du Buisson
Department of Mathematics and Computing Science
P.O. Box 513
5600 MB EINDHOVEN

December 1988

MANIPULATIONS OF THE MODEL STRUCTURE

M.W.I. van Kraaij, W.Z. Venema, J.Wessels
Eindhoven University of Technology
Faculty of Mathematics and computing Science
P.O.Box 513, 5600MB Eindhoven, The Netherlands

ABSTRACT

The purpose of this paper is to describe a way to manipulate model structures of decision support systems. The approach will be worked out within the context of the manpower planning system Formasy.

1. Introduction

A decision support system (DSS) can support the decision making process of a planner by providing facilities to evaluate and generate different scenarios for the future. Each scenario is represented as a model, in which we distinguish the model structure and the model parameters. In the same way we will distinguish the processes that generate solutions of problems posed to the DSS in two different classes. The first class of processes provides the proper model structure. The second class of processes generates possible solutions of the problems, at choice by changing the model parameters on the basis of a steady model structure (see also [1]). The used models are often very extensive and deal with a tremendous amount of data. Changes in the structure of these models will attend with changes in a great deal of these data. As a matter of fact it should not be necessary to specify the required changes in full detail, because major parts of the new model can, at least in principle, be derived from the original model and from a more global description of the required changes. Therefore we would like to execute such transformations of the model structure in an efficient way and with minimal help of the user. In particular, the planner should not need to know about the mathematical models that are involved in the evaluation of different scenarios.

2. Environment

The system will be developed within the framework of an existing DSS for manpower planning. The remainder of this section is devoted to a description of the current planning environment and gives a classification of model structure transformations that we want to support.

2.1. The decision support system Formasy

Formasy is a decision support system for manpower planning problems [2]. With the term manpower planning we mean the medium and long-term matching of personnel requirement and availability. The future requirement is determined by the development of the activities of the organization. The future availability is a result of the actual personnel distribution and the personnel policy (recruitment policy, careers, training, etc.). In Formasy the availability of personnel is modelled by using models, that reflect the classification and the evolution possibilities of the personnel. The personnel is distributed over a set of categories and the transitions from one category to another are governed by a push (Markov) or a pull (renewal) mechanism. Thus the structure of these models can be seen as a directed network structure. The categories in which the personnel is classified are specified by several characteristics, such as grade, age, grade seniority and level of training. These characteristics are chosen such that the personnel flows between the categories themselves and between the categories and the environment can be described correctly through this network structure. Furthermore, the distribution over the categories must reflect the facts the user is interested in. The categories form the nodes of the network, the possible transitions between the categories are reflected by the arcs.

After the network structure is known, the model is completed by specifying the model parameters, such as the transition fractions, the recruitment data and the wastage data. We will call this completely specified model the model instance, the underlying network structure is called the personnel structure. The object is to provide the possibility to manipulate the personnel structure.

2.2. Classification of model structure transformations

Manipulations on the network structure can refer to alterations with respect to the set of categories (i.e. the nodes of the network structure), but they also can refer to alterations of the transition structure only. The alterations with respect to the network structure will be induced by changes of the range of values of the characteristics of the involved categories. However, the kind of alterations in the structure depends on the meaning of the concerned characteristic. For instance changes of the range of the characteristic training are restricted by the possible levels of training of the personnel. Employees with the same level of training will always have the same value for this characteristic. However, changes in the range of the characteristic grade are only restricted by the intentions of the user. In principle the values for such a characteristic can be assigned freely. The reason for this difference lies in the nature of the concerned characteristics. We can classify them in so called variable characteristics and fixed characteristics. Fixed characteristics

reflect properties which will not change by changing the range of values, such as level of training and length of service. Of course the values of these characteristics can be changed, but the assignment of the new values is fixed by the real value of the concerned property. Variable characteristics are also used to determine the career patterns of the personnel. The range of values of these characteristics is dependent on the organization. The assignment of these values to the categories is determined by the position of this category in the entire structure and not (only) of fixed properties of the personnel in such a category. Examples of such characteristics are grade and grade seniority.

As mentioned before, manipulations of the personnel structure can be induced by alterations in the set of categories and/or by alterations in the transition structure. Alterations in the set of categories can be described in terms of the characteristics. We will now give a summary of possible alterations in the network structure and a global description how to realize them:

1 add a new characteristic:
   copy the concerned part of the actual structure and add this new characteristic to all categories with the proper value.
2 remove an existing characteristic:
   remove this characteristic in all the categories and join those categories that have become identical (have the same values for the remaining characteristics).
3 alteration in the range of values of a characteristic:
   if necessary, copy some parts of the structure;
   replace in the categories the values of the considered characteristic by proper values, according to certain rules;
   join the categories that have become identical and clean up the network structure.
   As mentioned before, the rules to replace the values have a different character for fixed and for variable characteristics.
4 changes in the possible transitions between groups of categories, aggregated according to rules based on the possible values of the various characteristics:
   from the desired changes in the transitions between the groups is derived what has to be changed in the original network structure, after which these changes are made.

Combinations of this types of manipulations can be done by successively realizing them. The result of any manipulation will then be used as input for the next manipulation. The need for such manipulations can arise from e.g. a reorganization, by which the grade system is revised (thus the existing grades with structure are replaced by new grades with structure). Other examples are the fusion of two departments or the decision to pursue an active emancipation policy, for which purpose the characteristic sex has to be added.

2.3. Description of the problem type

The class of possible alterations of the model instance we consider refers to alterations of the underlying network structure. The types of characteristics will not be changed (no characteristics will be added or removed). Only the range of values of the considered characteristics might be altered. At least one of these characteristics is a variable one (e.g. grade). Thus we consider alterations of the network structure, induced by

changing the categories (i.e. the nodes) by revaluating the values of some characteristics and/or by changing the transitions (i.e. the arcs) between the categories.

We assume that all alterations of the model instance with respect to the personnel structure, are initiated by the user. As mentioned before, mostly it is not practical to have the user specify the required alterations in detail, because of the extent and complexity of these alterations. To meet this, we require that the user and the system communicate on base of some aggregated view of the model instance. The aggregation involves a grouping of categories based on the values of their characteristics. The aggregated network structure results from the personnel structure and the way of aggregation: the nodes represent the groups of categories, the arcs represent transitions between the categories in the concerned groups. This aggregation enables the user to specify his intentions in terms familiar to the origin of the desired changes.

We make the following assumptions with respect to the way of aggregation:

- the user thinks within the scope of some aggregated structure of the model instance (e.g. the grade structure) and is able to specify this;
- aggregation only happens based on existing characteristics;
- the user is able to specify the desired aggregated structure and to specify the relation between the original and the desired aggregated structures;
- the way of aggregation is the same in the original as well as the desired structure (e.g. when the original model instance is aggregated with respect to the characteristic grade, then the nodes in the desired aggregated structure also represent grades).

A new detailed model instance is generated from a given model instance based on information specified on the level of aggregation. As a result, the aggregated structure belonging to the new detailed model instance will match the intentions of the user exactly. This does not fix the specification of the detailed model instance completely, but from this a specification can be derived which is sufficient. Therefore, for the way to generate model parameters in the new detailed model instances, we assume that the personnel flows in both structures roughly show the same behavior. In this way the user can see what kind of personnel policy, based on the new model structure, would have the same effects as the original situation. This gives him an appropriate starting point in the process of developing policies, based on the new model structure.

3. Functional requirements

The intention is to develop a system that allows a user to specify requirements and to propose changes to a given model structure in a global way, in terms familiar to the user and such that it is not necessary to specify all the required modifications in detail. From a global specification and from a given detailed model instance the system must be able to construct a new detailed model instance in accordance with the intentions of the user.

4. Conceptual model

4.1. Manipulations of the model structure

In our opinion a possible solution to the problem is to perform the communication between user and system via an aggregated view of the model, instead of in terms of the completely specified model itself. The completely specified model we will call the detailed model instance, its structure the detailed structure. The structure of the aggregated view is called the aggregated structure. The level of aggregation has to be chosen such that the system is able to derive the consequences for the detailed model instance from the changes specified by the user in terms of the aggregated view. Further this aggregated view must be in accordance with the way of thinking of the user. In this way, the user can describe the required changes in a more global way and in terms more similar to the problem domain.

We will give now a general description of a system, which is able to manipulate the model structure. We start from a completely specified model. The goal is to transform the structure of this detailed model instance, according to the wishes of the user. As mentioned before, the user and the system communicate in terms of an aggregated view of the model. Therefore, the user has to first chose an appropriate level of aggregation. The aggregated structure, which is what the user actually sees, can then be derived from this aggregation method and from the structure of the detailed model. From this point, the user will describe the changes in terms of the aggregated structure. The system deduces what changes have to be made to the aggregation method and to the underlying detailed model instance, and regenerates a corresponding aggregated model structure which is used for feedback to the user. This process will be repeated until the global structure of the detailed model reflects the intentions of the user. As a final stage in this process, it may be necessary to perform some clean-up on the detailed model (the detailed model structure can have become rather disorderly as a result of all the changes).

Resuming, the transformation process to change the model structure is as follows:
Initial state: the detailed model instance M_o.
Goal: the generation of a new model instance, based on a new structure and in accordance with the intentions of the user.
This process is divided in three stages:

i description of the aggregation level G_o of the original model instance M_o;
construction of the aggregated structure S_o from the model instance M_o and from the description of the aggregation level G_o.

ii transformations, as a result of required alterations in the aggregated structure:
Initial state: given M_o and G_o.
A series of intermediate model instances M_i and aggregated descriptions G_i is generated as follows. The user indicates the changes required in the related aggregated structure S_i. The system translates these changes to the model instance M_i and the aggregated description G_i and executes them, with result M_{i+1} and G_{i+1}. Next S_{i+1} is derived from M_{i+1} and G_{i+1}.

This process ends when the aggregated structure S_{i+1} satisfies the intentions of the user.

Result: M_l and G_l, such that the related aggregated structure S_l has the correct form.

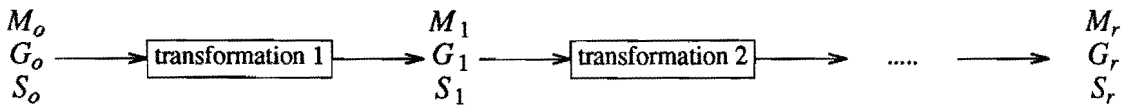
iii transformations to smooth the model structure (these transformations do not have any effect on the related aggregated structure):

Initial state: given M_0 and G_0, the result of stage 2.

A series of intermediate model instances M_i and aggregation descriptions G_i is generated as follows. Every unit of the aggregated description G_i corresponds with some part of the model instance M_i. Select such a part of M_i with the help of G_i. Transform the structure of this part of the model structure to a structure, with properties valid for the model structures under consideration and in conformity with the intentions of the user. This results in M_{i+1} and G_{i+1}. The related aggregated structure M_{i+1} will be the same as M_i, because only changes are made in a part of M_i corresponding to one unit of G_i.

Result: a model instance M_r with a proper structure, whose aggregated structure S_r meets the intentions of the user.

Stage 1 provides for the preparative work. In the stages 2 and 3 is deduced successively what should be the result of the next action and what transformations have to be done to realize this. Next this transformation will be executed. In this way a series of transformations is generated, which converts the original model instance in the model instance as required by the user. This process is shown below.



with M_o the original model structure and M_r the resulting new model instance such that S_r satisfies the intentions of the user.

4.2. The structure of the transformation process in Formasy

The starting point is an existing detailed model instance. The user himself always starts the processes to alter the model structures. The first stage is the definition of the level of aggregation, which will be the interface between the user and the system. The user specifies this aggregation level by grouping the categories according to certain rules. These rules describe the relation between the values of the characteristics of the categories and the way of grouping (e.g. grouping with respect to the characteristic grade).

The construction of the required detailed model instance is based on the regrouping of the categories, such that the new groups agree with the nodes in the required aggregated network structure. Therefore the user is asked to specify the groups in the required aggregated structure and to give the connection between the nodes in both (the original and the required) aggregated structures. From this the system derives the regrouping of

the categories in the original model instance and the corresponding aggregated network structure. The nodes of this aggregated network represent the required groups, the arcs are derived from the structure of the original model instance. Mostly the in this way obtained aggregated structure will not satisfy the intentions of the user.

In the next stage the obtained aggregated network is submitted to the user. Next he indicates what has to be changed in the transitions between the groups. From this, the system decides what has to be changed in the corresponding detailed model instance and executes those changes. The result is a new detailed model instance (with the same categories, only transitions are changed). Again the related aggregated network structure is deduced and submitted to the user. This process is repeated until the user is satisfied with the aggregated network structure.

As a result of the regrouping of the categories and the changing of the transitions, it is possible that the inner structure of the groups has to be adapted. For instance the regrouping of the categories can be based on the revaluating of the values of some characteristics. Therefore the proper values have to be assigned to these characteristics in the concerned categories. After this it is possible that some categories have become the same, so these categories must be joined. It also can happen that a group consists of many categories with a lot of transitions between them, while the inner structure of each group has to be a linear ordering of a fewer number of categories. The adaptation of the inner structure of a group consists if the assignment of new values to characteristics of that group, changes in the number of categories, changes in the transitions between categories and so on. This process of smoothing the structure of the model instance does not change the aggregated structure (only adaptations inside groups are done). So the result of this process will be a detailed model instance of a proper form, whose aggregated network structure agrees with the intentions of the user, thus the required model instance.

Now we will give a formal specification of this structure.
Initial state: the model instance M_o.
Goal: the generation of a new model instance, based on a new structure in accordance to the intentions of the user.
Four stages are distinguished:

1 definition of the aggregation level G_o and the construction of the aggregated network structure S_o
2 specification of the groups in the required aggregated network structure and the construction of the aggregated network structure S_t, related to this new grouping
3 transformations with respect to the transitions:
the system derives the consequences for the underlying model instance from the changes in the transitions of the aggregated network structure as specified by the user and transforms the model instance conformable
4 transformations on the inner structure of the groups of categories:
smooth the model instance to get a well-shaped model, whose detailed structure meets the requirements of the user.

This structure is shown in figure 1.

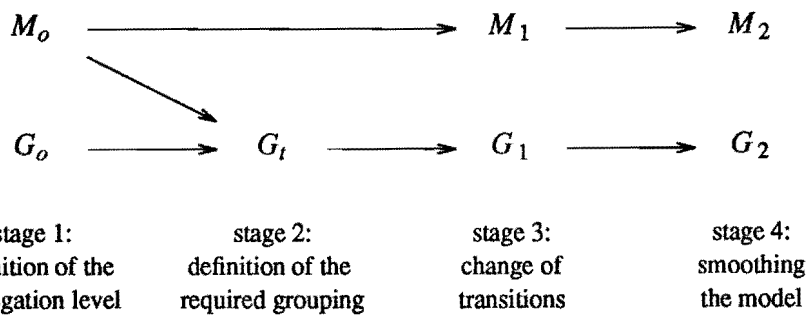


Fig.1: The structure of the transformation process

Next a more detailed description of the stages in this process is given.

Stage 1: definition of the aggregation level G_o and the construction of the aggregated network structure S_o

Given: M_o.
Result: the description of the grouping G_o of the categories in M_o (defining the aggregation level) and the to G_o and M_o related aggregated network structure S_o.

Description of the process:
1 The level of aggregation is defined by grouping the categories of the model instance. The user specifies the rules for this grouping.
2 Derivation of the description of the grouping G_o.
3 Construction of the aggregated network structure S_o, the derivative of G_o and M_o: the groups of G_o form the nodes, the arcs are derived from the structure of M_o.

Stage 2: specification of the groups in the required aggregated network structure and the construction of the aggregated network structure S_t, related to this new grouping

Given: M_o, G_o and S_o.
Result: M_o, G_t (the grouping of the categories as required by the user for the new model instance) and S_t.

Description of the process:
1 The user defines the nodes of the required aggregated network structure and gives the connection between the nodes in the original and the required aggregated networks.
2 Derivation of G_t, the grouping of the categories in M_o for the required aggregated network structure.
3 Construction of the aggregated network structure S_t, the derivative of G_t and M_o.

Stage 3: transformations with respect to the transitions

Given: M_o, G_t and S_t (result of stage 2).
Result: M_l, G_l (=G_t) and S_l, such that S_l is the aggregated network structure as required by the user.

Description of the process:
Let $M_0 = M_o$, $G_0 = G_t$ and $S_0 = S_t$.
1 The user indicates a transition which has to be added in S_i, say from group g_i to group g_j.
The system adds at least one transition from a category in group g_i to a category in group g_j. This results in the model instance M_{i+1}. S_{i+1} is constructed from M_{i+1} and G_0 (G_0 does not change: only transitions are altered, not the grouping of categories).
Repeat this process until the user does not want to add more transitions.
2 The user indicates a transition which has to be removed in S_i, say from group g_i to group g_j.
The system removes all transitions from the categories in group g_i to the categories in group g_j. This results in the model instance M_{i+1}. S_{i+1} is constructed from M_{i+1} and G_0 (G_0 does not change: only transitions are altered, not the grouping of categories).
Repeat this process until the user does not want to remove more transitions.
After these transformations the aggregated network structure is according to the intentions of the user.

Stage 4: transformations on the inner structure of the groups of categories

Given: M_0, G_0 and S_0 (result of stage 3).
Result: M_r, G_r and S_r, such that M_r is a well-shaped model instance, whose related aggregated network structure S_r meets the intentions of the user. M_r is the required new model instance.

Description of the process:
1 Select a group of categories from M_i with the help of G_i.
2 Smooth the inner structure of this group: assign to all characteristics the proper value, join categories that have become the same, change the number of categories on indication of the user, make the structure similar to the inner structures of the groups in the original model instance and so on.
This results in the model instance M_{i+1}. G_{i+1} is similar to G_i, except for the altered group. The related aggregated network structure does not change: in this process only the inner structure of the groups in the aggregated network is changed.
3 Repeat this process until all groups in S_0 are restructured.

5. References

[1] van Kraaij,M.W.I., Venema,W.Z., Wessels,J., *Extensions to the manpower planning systemFormasy*, Report NFI 11.88/06 (1988);

[2] Verhoeven, C.J., *Techniques in corporate manpower planning; methods and applications*, Kluwer/Nijhoff Publishing, Boston (1982).

DESIGNING DECISION SUPPORT SYSTEMS NOTES

In this series appeared:

number	author(s)	title
NFI 11.86/01	K.M. van Hee	Decision Support Systems for Logistics, 70 p.
NFI 11.86/02	K.M. van Hee, R.J. Wijbrands	Decision Support System for Container Terminal Planning, 20 p.
NFI 11.86/03	K.M. van Hee, B. Huitink, D.K. Leegwater	Portplan, Decision Support Systems for Port Terminals, 26 p.
NFI 11.87/01	J.K. Lenstra	Interfaces between Operations Research and Computer Science, 16 p.
NFI 11.87/02	K.M. van Hee, A. Lapinski	OR and AI Approaches to Decision Support Systems, 33 p.
NFI 11.87/03	P.W.G. Bots, H.G. Sol	An Environment to Support Problem Solving, 15 p.
NFI 11.87/04	H.G. Sol, M.B. van der Ven	A Group Decision Support System for International Transfer Pricing Decisions within the Pharmaceutical Industry, 19 p.
NFI 11.87/05	J.M. Anthonisse, J.K. Lenstra, M.W.P. Savelsbergh	Functional Description of CAR, an Interactive System for Computer Aided Routing, 15 p.
NFI 11.87/06	J.M. Anthonisse, K.M. van Hee, J.K. Lenstra	Resource-Constrained Project Scheduling: an International Exercise in DSS Development, 11 p.
NFI 11.87/07	K.M. van Hee	Features of the Architecture of Decision Support Systems, 6 p.
NFI 11.87/08	H.G. Sol	Wrijvingspunten rond Management Ondersteunende Systemen, 11 p.

number	author(s)	title
NFI 11.88/01	M. Desrochers, J.K. Lenstra, M.W.P. Savelsbergh	A classification scheme for vehicle routing and scheduling problems, 12 p.
NFI 11.88/02	J.M. Anthonisse, J.K. Lenstra, M.W.P. Savelsbergh	Behind the screen: DSS from an OR point of view, 6 p.
NFI 11.88/03	A.H. Vellekoop, C.G.E. Boender, A.H.G. Rinnooy Kan	The design of interactive decision support systems, 7 p.
NFI 11.88/04	C.G.E. Boender, A.H.G. Rinnooy Kan, A.H. Vellekoop	A random field model for estimating the content of soil layers, 7 p.
NFI 11.88/05	A.W.J. Kolen, A.P. Woerlee	VIPS, A decision support system for Visual Interactive Production Scheduling, 21 p.
NFI 11.88/06	M.W.J. van Kraaij, W.Z. Venema, J. Wessels,	Extensions to the manpower planning system formasy, 15 p.
NFI 11.88/07	W.Z. Venema, J. Wessels	Systematic modeling and model handling for manpower planning systems, 15 p.