EXTENSIONS TO THE MANPOWER PLANNING SYSTEM FORMASY

by

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

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

October 1988
M.W.J. van Kraaij
Eindhoven University of Technology
Faculty of Mathematics and Computing Science
P.O. Box 513, 5600 MB Eindhoven, the Netherlands

W.Z. Venema
Eindhoven University of Technology
Faculty of Mathematics and Computing Science
P.O. Box 513, 5600 MB Eindhoven, the Netherlands

J. Wessels
Eindhoven University of Technology
Faculty of Mathematics and Computing Science
P.O. Box 513, 5600 MB Eindhoven, the Netherlands
EXTENSIONS TO THE MANPOWER PLANNING SYSTEM FORMASY

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

A restriction of conventional decision support systems is that the available knowledge can only be used for solving a few type of problems. We attempt to eliminate this limitation by introducing a reasoning mechanism that manages and manipulates model-related data and algorithms. This approach is introduced within the context of the manpower planning system Formasy. Formasy provides the possibility to evaluate and generate policies, based on several variants of a model. A restriction of the present system is that it only supports the generation of model variants with the same underlying structure. A first application of the reasoning mechanism will be to provide support for automatic generation of model variants with different structures.

1 Introduction

A specific decision support system only supports the decision maker in solving a specific set of problems. For example, a system for operational planning (e.g. production planning or vehicle routing) contains a great deal of knowledge about the relevant problem domain. However, only questions related to operational planning can be readily answered by such a DSS. It requires a considerable amount of work from the user to use the same system for strategical planning (e.g. what is the impact of buying a new machine or a new vehicle). Although sufficient knowledge is present, it cannot be used due to restrictions in the underlying software.

In our research, we investigate the possibilities to make a system, that is more flexible. With this is meant a system, in which the available knowledge and data can be managed in several ways, to support various kinds of problems and to gain as much as possible by the available information. Another task of a DSS will be to translate requests of the user to the mathematical models and to decide how much assistance of the user is required to generate an answer. We think that these properties can be realized by a
system in which the available knowledge is represented in a set of independent algorithms (to solve subproblems), which will be managed by some sort of a "reasoning mechanism". The object is to investigate in which way such a system can be developed. In this paper we will give a global description of such a system.

A problem domain, where such a flexible system would be of use is the domain of manpower planning problems. As a base for our research, we make use of the already developed manpower planning system Formasy, described in [1], [2] and [3]. At this moment Formasy is not very flexible. Some extensions are required, to create the possibility for more direct support to the many questions a user may have. One of the restrictions of Formasy is, that it cannot support transformations of the model structure. It is only possible to generate and evaluate variants of models, based on the same structure. We attempt to develop an extension of Formasy, which provides the possibility to transform the model structure, in a way as described above.

In section 2 the basic structure of a DSS consisting of sets of algorithms and reasoning mechanisms is described. In section 3 we give a short overview of manpower planning and the models which can be used to simulate and develop personnel policies. In section 4 we describe the manpower planning system Formasy. In section 5 we will see what extensions to this system are desirable. One of these extensions will be further considered and a first step for the development of this extension is described.

2 Structuring a DSS

One of the tasks of a DSS is to support the decision making process by the evaluation and generation of several scenarios for the future. Therefore it must be rather easy to change the models, that reflect these scenarios. To effect this, the DSS must provide tools for the entire process of changing scenarios (i.e. the models describing them) and evaluating the consequences of these scenarios.

In this paper, we consider a DSS, in which the available knowledge is represented in a number of independent algorithms, managed and manipulated by some reasoning mechanism. The idea is that by an intelligent control of these algorithms and the available data, the system can support the decision maker in solving a lot of different types of problems. We expect that in this way also the solving of new types of problems can be supported. Besides in this way, the available knowledge can be represented and used efficiently. The components of such a DSS are:

i a database;
ii groups of independent algorithms (to execute subprocesses);
iii several control mechanisms.

The database consists of one or more complete model instances (i.e. the complete specified model describing the reality). Also parts of model instances and separated data can be included.

The processes that generate solutions of problems posed to the DSS, can be divided in two stages in an obvious way. The first stage will provide the proper model instance. The second stage will generate one or more possible solutions of the problem, based on
the model instance obtained in the first stage. According to this, the available algorithms also will be divided in two main groups, which both can be further divided in some subsets. The first main group of algorithms provides the proper model instance. This group contains
- algorithms to construct a model instance, using parts of existing model instances, data from the database and/or data specified by the user;
- algorithms to transform a given model instance to a new model instance (the so called transformation algorithms).

The algorithms in the second main group provide the execution of, for instance, computations and evaluations, based on a fixed model instance (the so called computational algorithms).

To manage the whole process in a correct way, control mechanisms that show some intelligent behavior are needed at several levels:
- in every subset of algorithms some mechanism is needed to choose and execute them in the proper order and to provide them the proper data;
- in every main group of algorithms a mechanism is needed to activate the proper subsets and to provide them the proper data;
- a mechanism is needed to manage the whole process, on the highest level. This mechanism co-ordinates the database, the two main groups of algorithms and the request of the user.

On the highest level the solution finding process will globally operate as follows. First a proper model instance will be chosen or generated. Depending on the request of the user and the chosen model instance, this model instance will be transformed in a new one. Next the computations can be done, based on the obtained model instance. In this process it may become apparent that a different model instance has to be chosen or that the present model instance has to be transformed (again). Next, more computations will be executed, and so on. The global structure of this process is shown in figure 1.

![Diagram](Fig.1: Global description of the solution finding process.)

We will illustrate the structure of this mechanism within the context of a DSS for manpower planning. A global description will be given of the reasoning mechanism, that will provide support for the automatic generation of model variants with different structures.
3 Manpower planning

In this article, we mean with manpower planning, the medium and long-term matching of the expected requirement for and availability of personnel on the different levels within an organization. This type of manpower planning is essential because of the long-term impact of manpower decisions, the delay in the effects of these decisions and the set of boundary conditions an organization has to face. The planning process comprises the forecast of future requirement for personnel and the forecast of the expected personnel distribution. Out of this, possible discrepancies may be determined and the matching of manpower requirement and availability can be obtained by adapting the future requirement and/or the expected availability.

The expected availability is a result of the actual personnel distribution, the expected turnovers and the pursued personnel policy (careers, retirement ages, recruitment policy, etc.). So the manpower availability will be adjusted by changing the actual personnel policy. Unfortunately, the development of policies is highly restricted by a lot of constraints, caused by the present situation. Think of e.g. the actual occupation and policy, required continuity in this policy especially with regard to the career patterns, protection against dismissal and degradation and the delay between the moment measures have been taken and the moment the effects arise.

The future requirement for personnel follows from the expected activities of the organization. The adaptation of the requirement depends on the flexibility of the organization plans and the situation in which the organization finds itself. Often an organization will have a number of other aims to satisfy, besides the expected requirement. A few examples are a certain desired age distribution, a desired man-woman ratio and a bound on the total salary costs. Because of all these constraints and aims, there are a lot of aspects in the medium and long term manpower planning, which will be of great interest for a personnel manager, such as:
- are there some bottle-necks to expect during the near future and how can they be solved;
- what are the consequences of e.g. a reorganization;
- in which way can certain aims be reached, for example what policy will result in a certain required occupation ratio in the different levels.

The part of the planning process we consider, is the adaptation of the expected manpower availability to a given requirement. We assume that in first instance, the future requirement for personnel and eventually some other aims are known to the planner and that in the matching process he himself makes the changes in these aims. Finally, he knows why these aims are proposed and what changes are possible and allowed, given the actual circumstances.

The object is to support the evaluation and the development of personnel policies, in order to match the expected personnel distribution on the different levels to the given aims. We will consider this process for groups of personnel and not for individual employees. Mathematical models, that can assist this process are the so called push (or Markov) and pull (or renewal) models, see [3] and [4]. For these models, we assume that
the personnel is divided into a number of categories, specified by several characteristics, such as grade, age, grade seniority and level of training. These characteristics have to be chosen such that the personnel flows between the categories themselves and between the categories and the environment can be described correctly by the model. Furthermore, this distribution over the categories must enable the user to get clear the facts he is interested in.

Push models can be used to describe the future manpower evolution and distribution from the present occupation and policy. The future manpower distribution is forecasted from the actual number of employees in all categories, the transition fractions between the categories, the wastage fractions, the retirement ages and the future recruitment in the various categories. So, with this kind of models, it is possible to determine the consequences of the actual policy and of alternative policies, at will corrected for possibly changed external circumstances. In this way the nature and size of eventual bottle-necks become clear. However, a feasible policy, in correspondence with the given aims, can only be found through a process of trial and error. In that case the impacts of alternative policies must be evaluated. Based on these evaluations, the policies will be adjusted and simulated again, until a feasible policy is found. Thus push models are less suitable for the direct development of new policies, which are in accordance with a desired result or which form a solution for signalized bottle-necks.

Pull models describe the personnel flows in organizations, whose policy is based on a formation principle: promotions and recruitment only happen in case of shortage in a higher ranked category. In these models, the actual manpower distribution, as well as the desired manpower distribution in the future and the wastage fractions have to be known. The expected promotion flows of the personnel and the required recruitment follow from this desired future distribution (and not any longer from the career patterns). Thus with a pull model can be computed how the personnel flows and recruitment have to be, in order to reach a certain occupation some years ahead. In this way, the feasibility of the proposed aims and their consequences, e.g. for the career perspectives of the employees in the various categories, become clear. Generally the policy, generated with a pull model, will exhibit a rather unstable course. Still it gives an indication of what a suitable policy will roughly look like. In this way a pull model can assist in the search for a new policy, in a more direct way then push models do.

4 Formasy

4.1 The global structure of Formasy

In this section we will give a global description of the interactive manpower planning system Formasy. We will discuss its main characteristics and the way this system can be used. For a more detailed description we refer to Verhoeven [3].

Formasy is a system to support the decision making process, with regard to manpower planning problems. It is developed to be used by a user without knowledge of the mathematical models, the system is based on. He may concentrate entirely on the
planning problems, meanwhile using Formasy as a resource, to get a view on the impacts of alternative policies or changed circumstances.

The underlying mathematical model is based on a push model. In Formasy, the categories in which the employees are classified, are defined by four labels. Two of them have a fixed meaning: the grade or function level and the grade seniority. The other two may be chosen by the user. When the age of the employees is also an important factor for the promotion or turnover fractions, one of the two free characteristics must be used for the age. Otherwise the age will only be used to predict the retirements.

On the basis of this distribution over categories, the following model properties have to be specified:

- transitions possibilities between categories;
- transition fractions;
- wastage properties, such as wastage fractions, retirement and early retirement ages and early retirement fractions;
- recruitment possibilities and recruitment numbers;
- salary levels.

Formasy is based on the assumption that the structure, with regard to the categories is fixed and cannot be changed, while consulting the system. This structure we will call the personnel structure. Within this steady structure, the user has various possibilities to simulate alternative policies and/or alternative circumstances, by changing the model properties, mentioned above. The simulation of these alternative policies or circumstances can, at choice, refer to:

- the manpower distribution over the various categories;
- the manpower distribution including the age distribution of the employees over the various categories;
- the actual and future salary costs for all grades.

In this way, Formasy can support the decision making process. With the help of Formasy, the effects of a given policy and of alternative policies can be evaluated. It becomes clear, when an alteration of the actual policy has to be considered and what in fact the bottle-necks are. Thus via this evaluation possibility, the system supports the user in the design of a new policy.

A computational model, which is a combination of push and pull models, has been developed by W. Bens [5] (but is not already included into Formasy). With this model the user has the choice for the use of a pure push or a pure pull model, but he also might choose a model which is a mixture of these two. With this extension, the system can support the user in the evaluation of the consequences of proposed aims. It gives an indication if and how these aims can be realized and what kind of policies they will imply. Bottlenecks, as a result of these aims, can be detected and they will aid the user in the decision making process. The design of a new policy can happen through an interaction between the push model and the mixed push-pull model.
4.2 Some restrictions of Formasy

Formasy can support the search for solutions to a lot of questions and problems. But in this process, the user himself has to do a lot of work. In fact, Formasy can answer only one question: what will happen, when a certain policy is performed. The development of new policies, in agreement with the proposed aims is a process of trial and error. The detection of bottlenecks has to be done by the user, using results from simulated policies and he himself has to find out in which way a solution can be find. For the computation of the consequences of e.g. a reorganization, the user has to change the structure with respect to the categories and to provide all new model parameters.

The problems a user has to deal with, can be divided naturally into problems concerning a steady personnel structure, as well as into problems taking account of an altered personnel structure. On this moment, Formasy only can deal with problems concerning a steady personnel structure. Thus, one of the useful extensions will refer to the possibility to change this structure.

The possibility to use both push and pull options will provide a better support of the user with problems, concerning a steady structure. There are also some desired extensions to these models. From results of a computation, a user can detect bottlenecks via an evaluation of the results. But when the aims of a user are given, the system should be able to do this evaluation to indicate these bottlenecks, and give advise to dissolve the arisen problems. Therefore it is desirable, too, that a user should be able to specify more of his aims, rather than only a required personnel distribution or a certain policy. Also the alternate use of these options during the solution finding process, either on request of the user, or suggested by the system, has to be possible and simple, from the point of view of the user.

5 Extensions of Formasy

5.1 Introduction

The object is to investigate if it is possible to realize a system, consisting of some sets of algorithms and mechanisms that are able to manipulate and manage them in an intelligent way. As mentioned before, this approach is investigated within the context of the system Formasy. In chapter 4 is shown that some extensions of this system are desirable. We would like that Formasy is able to support the user in a more direct way and to a more extensive set of questions, concerning the problem domain. We don't have the object to design a system, that produces a satisfying answer to every question, a user might submit. For example the user should made himself the decisions and the choices, that cause critical changes in the organization structure. He is the expert on this domain and knows the scope of the problems. He knows what is possible and desirable, within the actual situation.

We start from the assumption that the user is an expert in manpower planning (e.g. a personnel manager) and has a clear insight in manpower planning problems, but does not
need to have knowledge about the mathematical models on which the system is based. This means that the user must be able to communicate with the system in manpower planning terms. Another requirement for the system is, that it shows some intelligent behavior. With this is meant that the system must be able to decide what actions have to be taken when a user poses a problem to the system. For instance the system itself must determine if it is possible to generate an answer from the present personnel structure, if it is necessary to change this structure, which algorithms and data are needed in the solution finding process and which data must be specified by the user.

However we don't want the user to specify the problem entirely, so that the system only has to do the translation towards the mathematical formulation and next to perform the computations. A reason is, that the amount of data to specify, soon passes all reasonable bounds, while a lot of these data also can be derived from data, already specified, via a reasoning process or via algorithms. Well should be attended that, only if really necessary, the assistance of the user must be called in. We will illustrate this in a global way. Suppose we have a personnel structure, based on only two characteristics: grade and grade-seniority. A user wants to investigate, if an arisen bottleneck can be solved by joining the two specific grades. Then it might not be necessary, that he has to specify the new personnel structure entirely. This relatively simple change in the model structure, already requires the specification of a lot of new data:

- the number of grade-seniorities in the new grade;
- the redistribution of the employees from the two old grades into the new grade;
- the transition possibilities from and to the various seniorities in the new grade with the transition fractions;
- the recruitment possibilities in the new grade and the recruitment numbers;
- the wastage fractions.

It is not desirable that the user has to create the entire new model specification (except of course for the case he wants to do so). It is even likely that the user does not know how to choose all the new parameters. Besides, most of these properties can be derived within reasonable approximation from the properties of the old structure. For instance, when it is known which seniorities of the two grades will coincide, a proposal can be derived for the specification of the new model properties from the data in the old structure. The way these seniorities will coincide might be estimated from the common transitions between these two grades and the other grades in the structure, or eventually from a specification of the user. So a proposal can be made for the new structure, which will describe the new situation or the intentions of the user within a reasonable approximation. Of course, the user may introduce changes. Thus enough knowledge is available in the system to execute this transformation, without too much assistance of the user.

5.2 Global framework

We attempt to design some extensions to the system Formasy, in a way as described in section 2. The resulting system will consists of data and models, sets of algorithms and some reasoning mechanisms, which manipulate the data, models and algorithms in an
intelligent way. The reasoning mechanisms must be able to manipulate and combine the algorithms in a proper way, in order to generate a solution. They must determine what questions will be asked to the user, in case not enough data are available to execute the solution process. The assistance of the user also has to be called in, when critical choices or decisions have to be made. Thus, the reasoning mechanisms will operate as a model manipulator, as well as an intermediary between the user and the underlying algorithms.

As mentioned before, the set of mathematical models can be divided naturally into two subsets (see the sections 2 and 4.2). The first will contain the algorithms, that manipulate the personnel structure. The second subset will contain the algorithms to compute future scenarios, given a steady personnel structure, and can be subdivided further in e.g. push and pull models. The solution finding process also takes place in two stages. When the requests of the user give rise to changes in the personnel structure, first this transformation will occur. Next the system will try to find a solution, based on this structure. The structure of this process is illustrated in figure 2.

![Diagram](image.png)

*Fig.2: The structure of the solution finding process.*

We can distinguish three cases where a reasoning mechanism is desirable. The first one deals with the translation from the questions of the user into the terms of the underlying mathematical models. When this is done, it must be clear in which subset of algorithms, which problems have to be solved. The second mechanism will refer to the subset of models that provides the possibility of alteration of the personnel structure. The third case will refer to the subset of models concerning a steady personnel structure. We will consider this second mechanism in a global way.

5.3 Transformation of the personnel structure

5.3.1 Introduction

The extension of Formasy we consider provides the possibility to change the personnel structure. We assume that the personnel structure only will be changed on demand of the user and that he is able to specify the network structure with respect to the grades (that is the grades and the transitions between grades). Further we assume that the personnel flows in both structures roughly show the same behavior. When other personnel
flows are desired, the adaptation of the model properties, on basis of this new structure, will next be computed via the subset of algorithms concerning a steady personnel structure.

A lot of different transformations of the personnel structure are imaginable. They can all be seen as a series of so called basic transformations, that provide elementary changes of the structure of a model instance. So, this extension will consist of a subset of algorithms, which contains a complete set of basic transformations, and a mechanism to manipulate them. We will discuss which basic transformations have to be incorporated, and the way the reasoning mechanism will manipulate and manage them. In this case, the reasoning mechanism will work rather linear. The basic transformations will roughly be arranged in a fixed order. During the reasoning process, a continual evaluation takes place, to decide which basic transformation has to be selected next, or whether the assistance of the user must be called in. It is obvious that while manipulating the basic transformations, the assistance of the user may be necessary. Sometimes the execution of the basic transformations themselves will also require the assistance of the user, but we will pay no attention to that. So this reasoning mechanism has to be able to evaluate the phase the transformation has arrived, to reason out which new data must be asked and to execute basic transformations in the proper order.

We will explain according to which pattern the reasoning mechanism will operate. This illustrates what kinds of basic transformations are needed and what kinds of data the user has to specify. Finally, we will illustrate this process with the help of a fictive example.

5.3.2 Description of the transformation process

To describe the different stages of the transformation of a personnel structure, we assume that this structure is based on the characteristics grade and grade seniority. First the user must tell the system
- that he wants a transformation of the present personnel structure;
- which grades in the present structure are concerned;
- by which grade structure, this part of the present structure has to be replaced.

After this, the system checks if the structure desired by the user is allowed (i.e. no cycles may occur). The part of the existing structure that is concerned with the transformation consists of the grades given by the user, including the grades that are directly connected to these grades via possible transitions. This part of the existing structure we will call structure I. Structure II is defined as the final transformed version of structure I. At this stage, structure I is completely specified and of structure II the network structure with respect to the grades is known to the system.

To meet as much as possible the intentions of the user, it is also necessary to ask him to make a mapping from the grades in structure I, to the grades in structure II. If there are some grades in structure I, that maps on more than one grade in structure II (the mapping is not a function), the user has to specify for those grades in structure I on which grounds these grades will be subdivided over the regarding grades in structure I. This subdivision can happen on ground of already existing characteristics, as well as on
ground of new characteristics. It is also conceivable, that all the categories in a grade will be divided over the new grades by a fixed fraction.

Now the system is able to select and execute the proper basic transformations. The relevant grades in structure I are divided, as indicated by the user. The resulting structure we will call structure I'. Next a function is made to map the grades in structure I' to the grades in structure II. The corresponding transitions between the grades in structure II follow from this function and the transitions in structure I'. The network structure with respect to the grades obtained in this way often will be different from the network structure desired by the user. Some transitions between the grades in structure I' have to be removed, other transitions have to be added. The system will add and remove these corresponding transitions in the completely specified structure I'.

Finally, structure II will be constructed grade by grade. Those grades in structure I', that are mapped on the same grades in structure II, are grouped. The order is determined by the hierarchy of the grades in the structure. The grades are constructed from bottom to top.

We will now summarize which basic transformations and evaluation possibilities must be provided and which questions have to be asked to the user. The basic transformations that are necessary for this process are:

- check on correctness of the desired structure;
- divide or join (groups of) categories, e.g. grades;
- add or remove (groups of) categories, e.g. increase or decrease the number of seniorities in a grade, or add or remove a new characteristic;
- add or remove transitions between (groups of) categories.

The user has to specify:

- the network structure with respect to the grades;
- the characteristics, on basis of which existing grades will be divided over the new grades;
- sometimes, when in the existing structure new transitions must be added, the user has to specify them.

Resources, that assist in choosing the next action during the process are:

- the evaluation possibility, to examine if a mapping between the grades of two structures is a function, and the possibility to create this function with the help of the user;
- the possibility to a continual interaction between two structures via a function.

5.3.3 Example.

Again we suppose that the personnel structure is based on the characteristics grade and grade seniority and that the existing structure is completely specified. In figure 3, the network structure with respect to all the categories is given. A user wants to transform a part of the model structure. The system will ask him to specify the desired network structure with respect to the grades. The desired grade structure is shown in figure 4, together with the existing grade structure. This structure is allowed (there are no cycles). The user will also have to specify the mapping from the grades in structure I (the existing
Fig. 3: The network of the personnel structure of a fictive organization. The categories are specified by the characteristics grade and grade seniority.

structure) to the grades in structure II (the desired structure). The mapping given by the user is the following:
1 → a and c
2 → b and d
3 → c and e
4 → d and f.
Evidently grade 5 will be mapped on itself.

Fig. 4: The network structures with respect to the grades of the existing structure and the desired, new structure.
The mapping of grades is not a function, so the user has to specify in what way old grades have to be divided over the new grades. In this example, the subdivision will happen on ground of the grade seniority. For grade 1, the dividing line lies between the seniorities 4 and 5, for the grades 2 and 3 between the seniorities 3 and 4, and for grade 4 between the seniorities 2 and 3. Structure I is transformed to structure I', by splitting these four grades, according to the given subdivision. Grade 1 is divided in the grades 1a and 1c, grade 2 in the grades 2b and 2d, grade 3 in the grades 3c and 3e and grade 4 in the grades 4d and 4f. The network with respect to the grades of structure I' is shown in figure 5.

Fig.5: The network structure with respect to the grades of structure I', generated from structure I by dividing some grades.

Fig.6: The network structure, which should arise when forming the grades c and d out of the grades 1c and 3c, resp. 2d and 4d, in structure I'.

It is obvious now, how the function between the grades of structure I' and structure II looks like (1a → a, etc.). The system examines, which transitions between the grades in structure II would follow from this function and the transitions in structure I'. The network of structure II, that is generated in this way, is shown in figure 6. This network structure is not in accordance with the intentions of the user. The transitions a-d, c-d and d-c have to be removed. The transition c-f has to be added. This means in structure I' that the transitions 1c-4d, 2d-3c and 1a-4d have to be removed and the transition 3c-4f has to be added. In figure 7 we show the new formed structure I''. Finally structure II can be constructed grade by grade:

- the grades 1a and 2b become the grades a and b respectively (perhaps after changing the number of seniorities, on demand of the user);
- grade c is constructed by joining the grades 1c and 3c;
- grade d is constructed by joining the grades 2d and 4d;
Fig. 7: Grade structure I", generated from structure I', by changing some transition possibilities.

- the grades 3e, 4f and 5 become the grades e,f and 5 respectively (eventually after changing the number of seniorities, on demand of the user).

6 Final remarks

The object is to develop a DSS that contains sets of computational and transformation algorithms and mechanisms that can handle these algorithms and the available data in an flexible way. In this way a system will be constructed to support transformations of a given model structure. More conventional systems only provide support for the generation and evaluation of variants of models having the same structure. We attempt to realize additional facilities by extending the DSS with a reasoning mechanism that translates requests from a user to a series of elementary model transformations. This paper discusses a first investigation of the reasoning process, and of the elementary model transformations that will be involved.

The development is based on an existing DSS for manpower planning, Formasy. In a global way, we have explained what kinds of extensions are needful and how they roughly would look like. A prototype, in which these ideas are implemented, is being constructed.

References


