Systematic modeling and model handling for manpower planning systems

Citation for published version (APA):

Venema, W. Z., & Wessels, J. (1988). *Systematic modeling and model handling for manpower planning systems*. (Designing decision support systems notes; Vol. 8807). Technische Universiteit Eindhoven.

Document status and date:

Published: 01/01/1988

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

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

**SYSTEMATIC MODELING AND
MODEL HANDLING FOR MANPOWER
PLANNING SYSTEMS**

by

W.Z. Venema
J. Wessels

NFI 11.88/07

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

December 1988

Systematic modeling and model handling for manpower planning systems

*Wietse Z. Venema and Jaap Wessels
Technical University Eindhoven,
Faculty of Mathematics and Computing Science,
P.O. Box 513, 5600MB Eindhoven, The Netherlands*

ABSTRACT

The primary purpose of this paper is to show an efficient way of handling models and model data in a decision support system, in which it is usual to consider several variants of a model. The model data primarily consist of model-defining data, but the same approach may be used for the generated results as well. By efficient handling is meant the handling by the computer as well as by the user. For the user it is particularly important that new models can be conveniently defined as variants of existing models.

The approach is introduced within the context of a decision support system for manpower planning based on Markov models. In the meantime the same approach has been used for the implementation of other decision support systems and has been found to be more generally applicable.

Keywords: Decision support systems, Manpower planning, Push models, Pull models, Model generation, Model data handling.

1. Introduction

A frequently practised way of supporting decision making is by evaluating different scenarios for the future. It is quite usual that these evaluations lead to the formulation of new scenarios, and so on. Therefore, it is important that decision support systems are designed such that users can easily build new scenarios, particularly as variants of older ones. Also, it is useful if a system can provide an overview of forecasts generated from different scenarios.

The evaluation of a scenario is quite often executed via the translation of that scenario into a mathematical model, which is further evaluated by simulation or mathematical analysis. Therefore, it is also important that one can

easily generate a new model from a new scenario.

The primary purpose, of course, of tools for the formulation of scenarios and models is to make life easier for the users. However, the efficiency aspects of data handling and storage cannot be neglected in view of the potentially large number of scenarios, models and forecasts.

In the present paper we will consider the aforementioned problems within the context of medium- and long-term manpower planning. We will start (in section 2) with a short introduction to manpower planning and the underlying models that can be used. In section 3 we give a description of the modeling as used in the decision support system for manpower planning FORMASY. Finally, in section 4, we treat the way in which model data are handled inside FORMASY. In fact, the techniques for handling of model data are independent of the application and have been used for other decision support systems as well.

The underlying idea is to introduce modularity into the model-data handling of closely-related models, largely in the same way as modularity is used in a family of closely-related products. In fact, a similar approach has been used in several areas among which software engineering. Presently, the most common approach to model-data handling is via relational data bases. This often leads to complex data bases that are hard to maintain with conventional data base management tools. Although a modular approach to the building of single models has been used before, our approach to handle the data for families of models is new to our best knowledge.

2. Manpower planning

In the operations-research literature, the term "manpower planning" often refers to *operational* planning with typical concerns such as crew scheduling. In the present paper we are rather concerned with the longer-term *tactical* and *strategical* aspects. Such types of problems emerge whenever decisions have long-lasting effects. One reason can be that certain functions require a long training within the organisation. Long-term decision making may also become necessary when constraints have to be taken into account. For example, employees can hardly be fired, or they even are entitled to some sort of career. Also, scarcity on the labour market can be a reason for long-term decision making.

The central problem in tactical and strategic manpower planning is to adjust the future need for and supply of manpower to each other. The need is determined by the development of the activities of the organisation, both quantitatively and qualitatively. The supply is determined by the manpower

policy with regard to recruitment, training, promotion and job rotation. Finally, the supply of manpower is constrained by the situation on the labour market.

Recommended introductions to this problem area and the techniques involved in its solutions are [2], [3] and [5].

A well-accepted approach to the modeling of the supply of manpower is by using Markov models [1]. The idea is that an employee can be characterised as belonging to one of a finite number of categories, and that transitions from one category to another are governed by a Markov chain.

There are two aspects typical of this way of modeling. The first is the division into categories, which should be such that it accounts for the features of interest as well as for the properties that determine the transitions. The second is the autonomous character of the Markov-chain transitions. Consequently, this type of model (also called push model) corresponds to a point of view in which general policies determine the transitions.

This latter aspect might represent a drawback if it would not be possible to implement constraints which force the flow of manpower within given bounds. So-called renewal or pull models are an extreme consequence (cf. [1]): in these models only the loss of manpower is autonomous. However, it is quite well possible to construct models that combine the push and pull-type behaviour in a realistic way, as will be discussed in the next section.

3. The manpower-planning system FORMASY

The heart of the manpower-planning system FORMASY (short for FOrcasting and Recruitment in MAnpower SYstems) is the generic model which provides a framework for the construction of different types of tools for efficient model handling and friendly user interaction (cf. [4-8]).

The basis for the generic model of FORMASY is the way of defining the categories of employees. This is done with the following criteria:

- a. grade (function type)
- b. grade age
- c. age
- d. extra characteristics

Typical of FORMASY is the introduction of *grade age* as the category-defining criterion. In principle, grade age represents the amount of time a person occupies the current grade. If an employee stays within the same grade, the grade age is incremented each year. Usually, after a promotion to another grade the grade age is set equal to zero.

The main reason for the introduction of the criterion *grade age* is to have a

sensible (statistical) indicator for promotability (see [4], [5], [6] for extensive explanation and illustration). However, recent versions of FORMASY support a more flexible use of the criterion *grade age*, in the sense that one does not necessarily enter a new grade at grade age zero, but at a grade age determined by a probability distribution which depends on the former grade and grade age. Also, it has been made possible to let employees skip one or more grade ages. These extensions have been implemented in order to comply with the situation, common in the Dutch civil service, that there is a formal indicator for the salary of an employee. This indicator is set to some value when a person enters a new grade and increases by at least one each year (up to some maximum value). The two category indicators *age* and *grade age* are typical in the sense that they follow a linear scale with (usually) increments by one.

For *grade* and *extra characteristics* more general transition structures are allowed (for an example, see fig. 1). For *extra characteristics* we discern the following types:

- a. not changing, not influencing
- b. not changing, influencing
- c. changing, influencing

The type *not changing, not influencing* is the simplest one. It is used if some invariant characteristic of persons has no influence on transitions. This is typically done in order to monitor the distribution of the characteristic over the employees. *Sex* could be an example.

The type *not changing, influencing* stands for an invariant characteristic which has at least some influence on some transitions. Again, *sex* can be an example. Another example could be *type of education*. The latter characteristic can play a role when some functions are reserved for persons with a specific type of education.

The third type (*changing, influencing*) can change over time and also has influence on the transitions. One might think here of some quality indicator (see fig. 2), but also of more objective indicators such as *location*, *type of contract*, or *elapsed time within the organisation*. The *level of education* or *experience* could be other examples of changing characteristics with influence on transitions.

For the evaluation of scenarios it is particularly important to know how transitions are induced. For simplicity we will first introduce a pure push mechanism and later add the pull features.

Changes in the occupation of the categories originate from:

- a. hiring and firing
- b. natural losses
- c. transitions between categories

In the push model *hiring and firing* is considered as an influence from outside. For now, we will not consider such external influences. *Natural losses* consist of some well-defined processes, such as

- 1. retirements at the standard retirement age
- 2. early retirements according to special early-retirement schemes
- 3. retirements based on medical rejection
- 4. resignations

Transitions between categories involve promotion in grade, changes with respect to the *extra characteristics* and changes with respect to age and grade age. Assuming category-dependent probabilities for the different types of natural losses and transitions between categories, it is possible to make forecasts for the number of persons in the different categories. Mathematically speaking, there is a matrix P of transition probabilities $p(i,j)_{i,j \in C}$, where C is the set of allowed categories and $p(i,j)$ denotes the probability for a person to belong to category j in the following year, given that the current category is i . Then the forecasted occupation of category j becomes

$$\sum_{i \in C} n_i p(i,j)$$

if n_i denotes the current number of employees belonging to category i . Let $n(t) = n_i(t)_{i \in C}$ denote a row-vector of forecasted occupation numbers for the categories in year t , then

$$n(t) = n(t-1)P \quad \text{for } t = 1, 2, \dots$$

$$n(0) = n$$

Hiring (and firing) can be taken into account easily and, hence, forecasts for occupations of the categories are obtained in a rather straightforward way. Forecasts for individual categories may not be too interesting. However, by aggregation one can obtain forecasts for grade occupation, for age distributions etc. Also, salary forecasts can be produced easily with the same material. Because of the large number of categories it is necessary to exploit the particular structure of the matrix P in the computations, but we will not go into that detail.

Using the techniques described here it is possible to evaluate a scenario for - say - the next five years, if the scenario prescribes the number and types of people to be hired, the promotion policy and the loss expectations. However, promotion policy and hiring policy are not always fixed. It may be more important to have the opportunities to adapt these policies to the actual situation.

Therefore, we introduce the possibility to formulate constraints on the manpower distribution for the coming years. We do this by partitioning the set of categories into clusters (not with respect to ages). For each cluster a target occupation can be formulated, for each year. A target occupation can act as an upper bound, as a lower bound or, as a combined upper and lower bound on the occupation of a cluster. There are different forms of rigidity for the realisation of the target values. For a detailed description see Bens [9]. The main question, of course, is how to use the clusters and their target values for the control of promotions and recruitments. In fact, the division into clusters provides a way to formulate a mixture of push models (based on career policy) and pull models (where promotions only take place in case of vacancies). We will consider an example rather than treating all practical possibilities of the system.

Consider the university example of fig. 1 and table 1. Suppose that there is a maximum constraint on the number of persons in grade *PROFA* and *PROFB* for each year. The current career policy, which is applied for the computations of table 1, leads to over-occupation of the higher grades. In the refined system we introduce two clusters:

cluster *D* consisting of grades *UD* and *UHD*

cluster *PROF* consisting of grades *PROFA* and *PROFB*

As policy we formulate that the occupation targets for cluster *PROF* are strict upper bounds, whereas the targets for the *total* occupation are strict upper and soft lower bounds. No people will be fired if an upper bound is exceeded. Moreover, it is formulated as a goal to hire new members of cluster *PROF* in equal amounts from outside and inside the organisation (from cluster *D*). With such a policy the forecasts can be made similarly as in the pure push model, if only the cluster occupations are computed in a top-down order.

For an example see table 2. Here, one sees the effects of a policy choice. Of course, if one does not like the result, one changes the policy and evaluates the consequences. There are different types of consequences. One type concerns the resulting manpower flows: the number of externally hired members of *PROF* in the forecasting period, or the number of *UHD* who are

not promoted to *PROF* but would be candidates according to the career policy. Information about such consequences also becomes available as a result of the computations.

A necessary class of tools in the system is related with translation algorithms that convert policies, formulated in terminology familiar to the user, into data for mathematical models. However, we will not touch upon this topic in the present paper. In the final section we will put emphasis on the structure of model-data storage in order to support efficient and effective modeling.

4. Structuring model data

As mentioned in the introduction, the primary task for a decision support system is to support the evaluation of alternative scenarios for the future. This is the more important since the evaluation of one scenario will usually lead to the generation of other scenarios. Thus, a decision support system should provide a convenient environment for the construction of new scenarios from previous ones, and at the same time provide an efficient way for storing and accessing model-related data.

The modeling as described in the previous section already requires lots of data for the definition of a pure push model. The combined push-pull model (with clusters) as mentioned at the end of that section involves even more data. The data that define a particular model consist of closely-related groups, each having its own structure. For instance, we have the set of grades (a finite and ordered set of names), the range of grade ages for each grade (a finite and ordered set of natural numbers having the same cardinality as the set of grades), the retirement ages per grade (a set with the same structure as previous one), the possibilities for early-retirement (a finite and ordered set of the same cardinality as the previous one, with for each element an ordered subset of the natural numbers), the inclinations towards early retirement, etc. etc.

By considering the data that define a particular model as a collection of separate building blocks it becomes relatively straightforward to support several variants of one or more models, as well as to derive new models from existing material. Instead of concentrating on complete model instances, the modeling support is concerned with different versions of building blocks. The building blocks constitute the main (private) data base of the dss. They are usually created either by extraction from (external) organisational data bases or by modifying a copy of an already available building block.

Thus, modeling support has become a matter of maintaining appropriate sets of building blocks (see table. 3). In general, a particular version of a building

block will be referenced by several different models. The building-block approach is efficient with respect to storage requirements because many alternative models can be maintained with a small collection of building blocks. It is also convenient for the user since it is relatively easy to incorporate model-defining data from previous models into a new scenario (table 4).

As mentioned before, the evaluation of one scenario often leads to the generation of other, similar, scenarios. Practically, a new model differs from the previous one by only one building block; the new model is created by adding a modified copy of that building block to the dss data base and by inheriting the remaining building blocks from the previous model. The building-block administration can remain completely invisible to the user.

In the same vein, one can attempt to account for the results produced when a scenario is evaluated. Often, the results of such an evaluation can be subdivided into groups of closely related data, analogous to the building blocks mentioned earlier. By recording which particular building blocks were actually used for producing a particular (partial) result, the system can make that result available for all models that refer to that same particular subset of building blocks. Clearly, this approach is both efficient with respect to storage and processing requirements, since a particular result is never computed more than once. The handling of (partial) results can also remain invisible for users; their primary benefit is a quicker response from the dss. A prototype system that implements this feature is currently under construction (cf. [10], [11]).

The modeling techniques presented in this paper should be applicable to classes of problems other than manpower planning. They provide a framework for convenient and efficient handling of models and variants of models.

Acknowledgement:

We acknowledge the great influence of many colleagues who participated in our effort to make better manpower planning systems. For the topic of the present paper, the influence of Ruud Zwart should be acknowledged in particular. The manpower planning system FORMASY has been developed in several versions over the past fifteen years (cf. references [8], [4], [5]). In the present paper we primarily refer to the version which is documented in [7] and which was built by a group in which R. Zwart was the principal engineer. The so-called pull aspects have been designed and implemented by W. Bens (cf. [9]). Application-independent tools for handling model-related data were designed and implemented by A. Driessens and W.

Venema (cf. [10]).

References

- [1] Bartholomew, D.J., *Stochastic models for social processes*, John Wiley, London (2nd. ed., 1973).
- [2] Bartholomew, D.J. (ed.), *Manpower planning*, Penguin Books, Harmondsworth (1976).
- [3] Bryant, D.T., Niehaus, R.J. (eds.), *Manpower planning and organization design*, Plenum Press (NATO Conference Series II: System Science, vol. 7), New York (1978).
- [4] Van Nunen, J.A.E.E., Wessels, J., *Forecasting and recruitment in graded manpower systems*, pp 353-364 in [3].
- [5] Verhoeven, C.J., *Techniques in corporate manpower planning; methods and applications*, Kluwer/Nijhoff Publishing, Boston (1982).
- [6] Verhoeven, C.J., Wessels, J., Wijgaard, J., *Computer-aided design of manpower policies*, Technical University Eindhoven (Manpower planning report no. 16), Eindhoven (1979).
- [7] Zwart, R. (ed.), *Technical documentation of FORMASY and PERFORM*, Technical University Eindhoven, Eindhoven (1984), in Dutch.
- [8] Wessels, J., Van Nunen, J.A.E.E., *FORMASY, FOrcasting and Recruitment in MAnpower SYstems*, Statistica Neerlandica 30(1976)173-193.
- [9] Bens, W.E.J.M., Design and implementation of a push-pull algorithm for manpower planning, Technical University Eindhoven, Eindhoven (1988).
- [10] Driessens, A.J.M., Venema, W.Z., *Implementation notes for the Markov information system*, Technical University Eindhoven, Eindhoven (1988).
- [11] Venema, W.Z., Automatic generation of standard operations on data structures, Technical University Eindhoven, Eindhoven (1988).

Fig. 1. Grade transition structure for faculty in Dutch universities.

Fig. 2. Transition structure for the combination of grade (vertical) and a quality indicator (horizontal) for a group of engineers in Dutch civil service.

	1988	1989	1990	1991	1992	1993
UD	67	64	62	59	56	52
UHD	33	34	34	33	34	33
PROFA	14	13	14	15	14	13
PROFB	16	15	13	12	8	7
Total	130	126	123	119	112	105

Table 1. Forecasts for the numbers of people in different grades in a faculty of a dutch university in five subsequent years, based on the situation in 1988.

		1988	1989	1990	1991	1992	1993
UD		67	63	60	57	53	49
UHD		33	35	36	38	40	40
PROFA		14	13	13	12	10	9
PROFB		16	14	13	12	8	7
PROF	actual	30	27	26	24	18	16
	target		27	24	21	18	15
Total	actual	130	126	123	119	112	105
	target		125	120	115	110	105

Table 2. Forecasts for the numbers of people in different grades in a faculty of a Dutch university in five subsequent years, based on the situation in 1988 with use of targets for the occupation of cluster *PROF* and the total occupation.

building- block type	available versions
<i>A</i>	a_1, a_2, a_3
<i>B</i>	b_1
<i>C</i>	c_1, c_2

Table. 3. Modeling with building blocks. The table shows a hypothetical dss data base with building block types *A*, *B* and *C* which are represented by versions { a_1, a_2, a_3 }, { b_1 } and { c_1, c_2 } respectively.

Model number	building-block versions
M_1	a_1, b_1, c_2
M_2	a_2, b_1, c_2

Table. 4. Modeling with building blocks from the hypothetical data base in table 3. The table shows that model M_1 consists of building-block versions a_1 of type A , b_1 of type B and c_2 of type C , and that model M_2 differs from model M_1 only in its version of the type A building block.