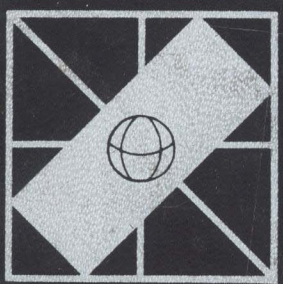
formal models and practical tools for information systems design

edited by
h.-j. schneider



IFIP

north-holland

FORMAL MODELS AND PRACTICAL TOOLS FOR INFORMATION SYSTEMS DESIGN

Proceedings of the IFIP TC-8 Working Conference on
Formal Models and Practical Tools for Information Systems Design
Oxford, U.K., April 17-20, 1979

edited by

Hans-Jochen SCHNEIDER
Technische Universität Berlin
F.R.G.

1979

NORTH-HOLLAND PUBLISHING COMPANY
AMSTERDAM • NEW YORK • OXFORD

ISBN: 0 444 85394 4

Published by:
NORTH-HOLLAND PUBLISHING COMPANY – AMSTERDAM • NEW YORK • OXFORD

Sole distributors for the U.S.A. and Canada
ELSEVIER NORTH-HOLLAND, INC.
52 Vanderbilt Avenue
New York, N.Y. 10017

Library of Congress Cataloging in Publication Data

IFIP TC-8 Working Conference on Formal Models and Practical Tools for Information Systems Design, Oxford, Eng., 1979.
Formal models and practical tools for information symtems design.

"Organized by IFIP Technical Committee 8, Informations Systems, International Federation for Information Processing."
1. Information storage and retrieval systems--Congresses. 2. Formal languages--Congresses. 3. Electronic digital computers--Programming--Congresses. I. Schneider, Hans-Jochen. II. International Federation for Information Processing. Committee on Information Systems. III. Title.

Z699.A1I16 1979 029 79-19860

PRINTED IN THE NETHERLANDS

FORMAL MODELS AND PRACTICAL TOOLS FOR INFORMATION SYSTEMS DESIGN

IFIP Working Conference on
Formal Models and Practical Tools
for Information Systems Design
Oxford, U.K., April 17-20, 1979

organized by
IFIP Technical Committee 8,
Information Systems
International Federation for Information Processing

NORTH-HOLLAND PUBLISHING COMPANY
AMSTERDAM • NEW YORK • OXFORD

PREFACE

In 1975, the general assembly of IFIP (International Federation of Information Processing) initiated a new Technical Committee TC 8 with the name "Information Systems". In 1976, the Working Group WG 8.1 on "Design and Evaluation of Information Systems" started work and established the following scope and aims

Scope

The scope of the working group is the development of approaches for the analysis, design, specification and evaluation of computer-assisted information systems.

Aims

The aims of the working group are:

1) to identify concepts and to develop theories relevant to the design of information systems;
2) to develop methods and tools for applying these theories to the design process;
3) to develop methods for the specification of information needs within an enterprise, with emphasis on interface aspects;
4) to develop methodologies for evaluating proposals for information systems;
5) to develop methodologies for evaluating the operational effectiveness of information systems.

In 1977 the Working Group WG 8.2 on "The Interaction of Information Systems and the Organization" was officially approved.

In Oxford in April 1979, WG 8.1 held his first Working Conference. The title of this conference was "Formal models and practical tools for information systems design". The aim was to address special research and development problems within the scope and aims of the Working Group WG 8.1.

The conference concerned itself specifically with that branch of information systems which deals with the development of formal models and practical tools for information systems analysis, design, realization and evaluation. Emphasis was placed on

Requirement Definition

- Involving the end-user in the requirement definition.
- Formal models and formal languages for stating requirements.
- Transparency of the requirement specification. Documentation and representation problems.
- Tools for analysis and evaluation of requirement specifications.
- Methods for handling systems complexity in information systems analysis and design.
- The role of data semantics in requirement definition.

- Methods for integrating different users views of the information system.
- The use of system prototypes, based on simplifying assumptions about the system, for development of precise requirement specifications.
- (Any) special problems in modelling highly interactive systems?
- The role of requirement specifications in making decisions on the structure of the application environment.
- Involving the end-user in the development of priorities and planning horizons of the information system development project.

Requirements and Software Systems

- Comprehensive methodology for information system design, development and operation.
- Methods for controlling that a proposed software system design meets the stated requirements.
- (How) should changes in the software system be reflected in the requirement specification during debugging and maintenance?
- Methods for utilizing requirement specifications in software systems design, e.g. for software production, consistency testing.
- Methods for testing that software system changes are consistent with the stated requirements.
- Methods for controlling evolution in the requirement specifications and the corresponding implementation of software extensions and software changes, i.e. methods for enhancing systems flexibility.
- Automatic software generation based on direct use of the requirement specifications.

Experimental Evaluation of Methods and Tools of the Systems Analyst and Designer

- Comparative studies of methods and tools. Studies concerning method proposals and studies concerning well established methods and tools are of equal importance.
- The relative importance of new methods and tools on the overall economy of the development, operation and maintenance of an information system.

Computer-assisted information systems, and their components, are becoming of ever increasing interest in our society. The data processing community has evolved towards a recognition of the importance of tools for handling large masses of data. However, the introduction and utilization of such tools for building information systems is not the end of the story. Users are now realising a need for tools which support the complete problem definition and solution process. Specifically such tools are needed when designing and installing information systems in industry, civil service, and in all other environments where such systems are used.

The fourteen papers presented represent a significant contribution to the scope and aims of WG 8.1 as listed above. Some ninety top-ranking professionals from 15 countries and from various groups such as industry, computer manufacturers, research institutes and universities were invited to participate in the conference. Presentations were given by well-known experts from main organizations and Universities in Europe and America.

The 36 papers submitted were reviewed by the Program Committee members

Chairman	A. Sølvberg	Norway
Members	F. Bodart	Belgium
	G. Bracchi	Italy
	M. Klein	France
	P. Kovaćs	Hungary
	F.F. Land	U.K.
	C. McGowan	U.S.A.
	H.-J. Schneider	Fed.Rep.Germany
	H.G. Sol	Holland
	F.Stella de R. Germano	Brazil
	A.I. Wasserman	U.S.A.

who together with the Organizing Committee members

Chairman	R.K. Stamper	U.K.
Members	P. Atkinson	U.K.
	H. Ellis	U.K.
	C. Marks	U.K.
	W. Olle	U.K.

prepared the conference with the general chairmanship of Prof. A.S. Douglas. We are also indebted to the Principal and Fellows of St. Edmond Hall for their hospitality and to the London School of Economics for its valuable contribution. The greatest part of the administrative load was taken by Mary Eades and Buffy Gibons.

The help of a member of very well-known specialist provided the basis for the success of this conference. I would like to thank all of them very much for their contributions.

I hope that this book provides a challenge to the information system community and a stimulus to researchers, implementers and users.

June 1979

Hans-Jochen Schneider

Technische Universität Berlin

TABLE OF CONTENTS

Formal Models and Practical Tools
for Information Systems Design
H.-J. Schneider, Editor
North-Holland Publishing Company

TOOLS FOR INFORMATION SYSTEM DESIGN AND REALIZATION

B.E.Meyer, H.-J. Schneider

Institut für Angewandte Informatik, Fachbereich Informatik,
Technische Universität Berlin, Otto-Suhr-Allee 18/20
D-1000 Berlin 10

The paper is concerned with requirements for an information system in a multi-user environment of an industrial or administrational institution and with an approach to meet these requirements by a method and model base system called BOSS. The system presented consists of the following basic components: a method base system, a model base system, a data base system, interfaces to modules of a program library and to external programs, a dialog system as a very comfortable user interface, and a monitor system for the control and communication between these components.

1. INTRODUCTION

Today we can recognize two essential trends of information system developments: The so-called application based information systems and the generalized information systems. The application based systems disclaim universality, they are systems for specific areas, e. g. medicine, building industry, jurisprudence, special branch systems, etc. As a part of generalized information systems the data base systems are well established today, but the development of methodologies and concepts for method or model base systems is just at the beginning. This paper will deal with the latter systems.

Data processing in the past was determined by the fact that different programs written in different programming languages had been developed for one application environment (e.g. an industry firm or an administrational institution) and that within one computer system different data bases, model bases or method bases together with specific user interfaces had been established. Therefore the need for so-called 'integrated solutions', i.e. the creation of generalized information systems, is recognized. In this context tools have to be provided to support the integration and to unify the handling of the different system components.

Looking some years ahead one can have complete problem solution support systems (PSSS) which are tools supporting the whole problem solution process just from the beginning. First you will define your terminology precisely, then you will fix which words in your application field are of interest and which are not (semi-automatic thesaurus construction). After this you will define the important relations in your problem area. Which relations should be activities and which informations, personal, material is needed to run an activity and which information, material is produced by an activity (precedence structure). This will be described on several levels, the problem will be broken down to a program-specification level. Now you will be able to define for restricted problem classes all infor-

mations you need to specify what a program has to do. You do not have to say how this has to be done, you can remain on a non-procedural or descriptive level. Generator programs create the source code in a procedural language (such as ALGOL, BASIC, COBOL, FORTRAN, PL/1, RPG II) and on this level special supplements can be made in a computer-assisted manner.

With such systems the whole problem solution process is computer-assisted and documented. You will have simultaneously interfaces generated to file or data base systems, to data base design aid systems and data dictionaries. The organization, the project-management and the data processing tasks are described in a common language, those will be no problem in the communications of data processing and application people. Consistency checks on each level of refinement will provide a faster and better problem solution.

The terms model and method base are used in the literature more and more frequently. (cf. [4], [10], [11], [12], [21]). There are many definitions of these two concepts. As we are in a very new research field it is very difficult to define these concepts precisely. Nevertheless we will try to give a short description: A method and model base system is a system which enables the casual user (cf. chapter 3) in a multi-user environment to solve his problems easily and fast with a minimum of programming efforts and a maximum of system support.

The terms method and model have a very general meaning, therefore it is not possible to define these terms in such a manner that all meanings are covered. We will use the term method in the sense of basic algorithms and procedures which are utilized in interdisciplinary subject areas. Examples of such methods are basic algorithms in statistics, mathematics, economy. Models we will use in the sence of mappings of the real world which can be described by equations (linear, non-linear equations, difference and differential equations) or graphs for example.

This paper is concerned with the requirements for such generalized systems within an industrial or administrational institution and with an approach to meet these requirements by the method and model base system BOSS (Berlin method and mOdel baSe System). The design concepts and the realization are discussed.

First we will give a historical overview to the way of development starting with classical data processing via file processing up to data base, method and model base processing. The method and model base system presented consists of the following components: a method base, a model base, a data base, interfaces to a program library and to external program, a user interface and a monitor system for the control and communication between these components. The requirements for such a system with respect to different user groups (e.g. method and model base administrator, application programmer or the casual user) are discussed (cf. chapter 3), the system architecture, the concepts and properties of single components are presented (cf. chapter 4). How to meet user requirements for very comfortable dialog systems with respect to method and model base systems and to different user classes is shown by the realization of our dialog interface (cf. chapter 5).

2. REMARKS ON THE HISTORY OF THE DEVELOPMENT OF DATA BASE, METHOD BASE AND MODEL BASE SYSTEMS

The following pictures represent only very rough the functional interrelationship between the corresponding components. In figure 1 the situation at the beginning of data processing in the mid-fifties is shown. Data descriptions, methods and eventually models were all integrated in the user program. The number of data was very limited, each user program was running seperately. The data were used for one user and one program at a time.

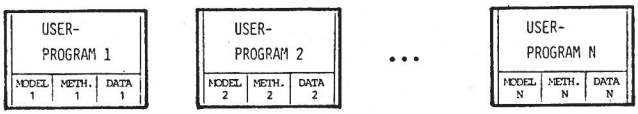


Fig. 1: Classical data processing

Then, in the beginning of the sixties, when monitor systems were first installed, user programs could be run with several data sets. The data were separated from the program but the physical data description remained in the program.

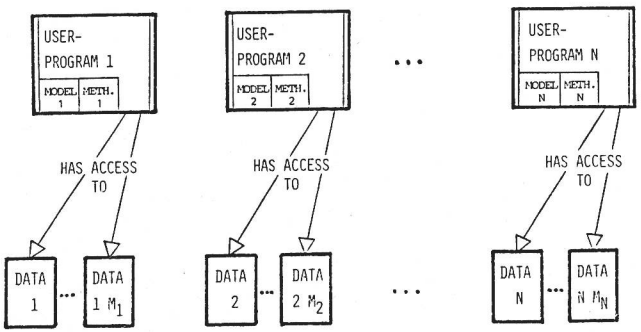


Fig. 2: Data processing with monitor systems

The number of persons engaged in data processing grew fast. The amount of data stored in the computer was increasing super-exponentially. Consequently data could not be used economically by one person alone. Data were used simultaneously by several users. To be able to handle these kinds of problems file management systems were introduced. The system architecture is described in figure 3. Several users are sharing the different data files. The physical data record description is still needed in the user programs.

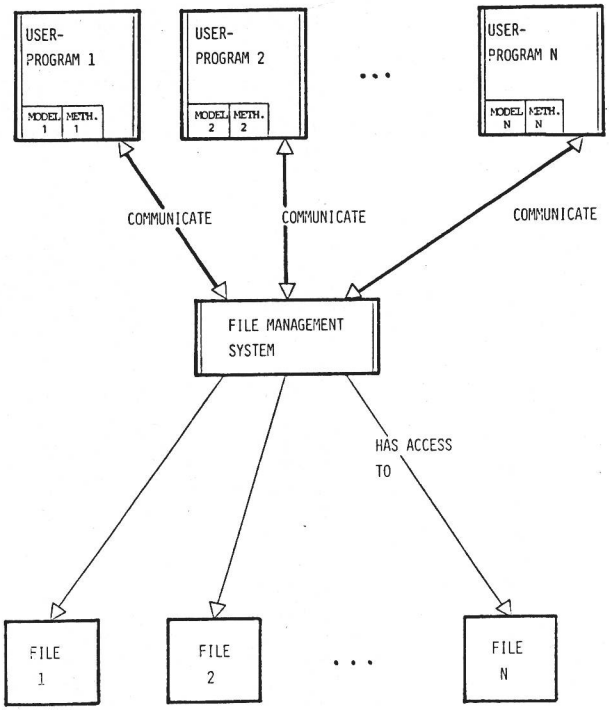


Fig. 3: File management system

A disadvantage of file management systems is on the one hand the redundancy and on the other - and this is more serious - the inconsistency of data, i.e. there is no support to keep the data consistent. By updating, the updating status of duplicates may be divergent. To guarantee the semantic consistency of data, data base systems were introduced (see figure 4). Same as in file management systems the user has no direct access to data, he has to communicate with the data base system which then delivers data to him or makes the required updates. The data base system is fully responsible for the consistency of the data base. In data base systems one gains the so-called data independence which means that user programs contain only the logical description of data structures and the corresponding operations running on these data structures. Only the data base system knows the physical description of the data structures and the corresponding access methods. The data base can be re-structured without influencing the user programs.

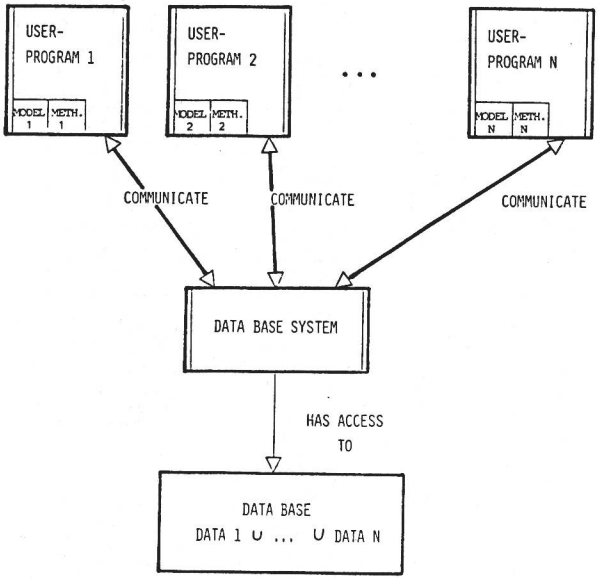


Fig. 4: Data base system architecture

The data base systems have helped us to get support in getting data independence which is very important in developing organisations. In addition to this data dictionaries within data base systems help us to understand the data base schema and the operations allowed.

With method base systems we will get support in handling methods consistently in a multi-user environment. The user will be guided by the system to his problem solution. The user need not have expert knowledge in every subject field. A rough description of the structure of method bases is given in figure 5.

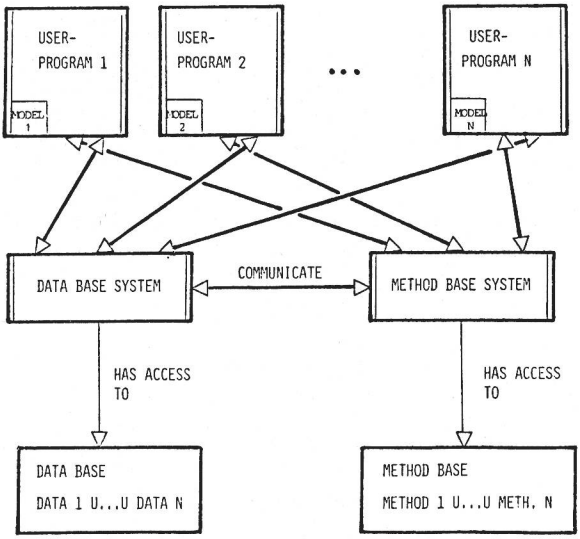


Fig. 5: Method base system architecture

The next step could be foreseen now. The programs will be independent of special models in a method and model base system environment.

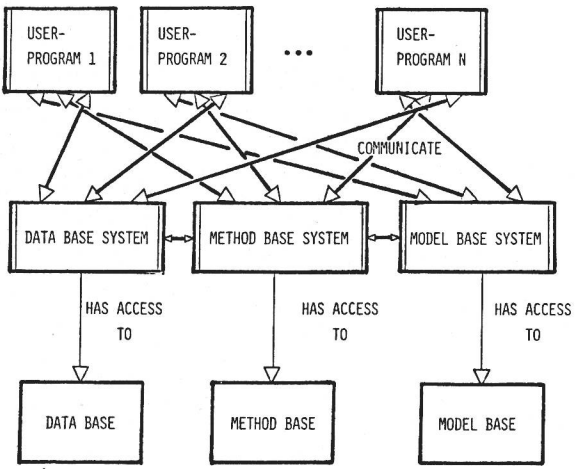


Fig. 6: Method and model base system architecture

We have gained independence, flexibility and multi-user access to data, method and model bases on the one hand, but on the other hand it is obvious that such software systems are very complex and difficult to handle. This complexity is the reason for the development of new ideas and methods for the planning and realization of such systems. To shorten the mass of concepts, of functions and the mass of software systems, it is necessary to come to an integration of these systems on the following levels:

- on the functional level,
- on a language level and
- on a system architecture level.

The following approach can be viewed as an attempt to satisfy the corresponding requirements and to solve the problems mentioned. Further possibilities to handle such complex systems are given in chapter 4.

3. REQUIREMENTS FOR A METHOD AND MODEL BASE SYSTEM

The necessity for a method and model base system is easily made plausible by means of a specific application example. If we constitute e.g. an information system for planning reasons within an operational environment of an industrial firm and if we want to run this system in the daily operation, we have to provide - among other things - tools for input, for validation, for manipulating the data of the firm etc. (cf. for example BLUMENTHAL [2]).

If we want to establish organizational models for planning we have to make available within the system mathematical methods, software-engineering instruments, simulation methods, possibilities for the verification, calibration, sensitivity analysis and validation of these models. Further tools are necessary which help to design and to maintain the layout of reports, and which generate reports automatically. To compare the real data with the planned data we need statistical methods, comparison measures etc.

Many authors have been concerned with the definition of requirements for such method and model base systems and with the motivation for these systems (cf. [4], [10],[11], [12], [18], [21]). In many papers these problems have been discussed very generally. We will focus on - in our opinion - most important requirements and we will try to show how to meet these requirements by practical tools.

The treatment of method and model base systems could not be performed without looking at the user who handles and utilizes these systems.

For the following we will distinguish between 3 user classes

1. the method and model base administrator,
2. the application programmer and system specialist,
3. the expert with respect to various application fields (e.g. the statistician, the manager, the engineer etc.) whom we will call the 'casual user'.

All these user classes are concerned with the system in some way and for all these users tools have to make available which allows these users to handle the system in an user-oriented and adequate way with

respect to their knowledge.

In particular we would like to deal with the last-mentioned user class, the casual user, not only because this is the biggest user group, but also because the motivation for the establishment of method and model base systems is derived from this user environment. This user class within a multi-user environment should have the possibility to solve problems by a minimum expenditure of time and money and by a maximum system support in a very quick and easy way.

This user is not at the same time a specialist in matrices, in statistics, in investment models and data base systems, but he is a user who has heard about these things during his education and wants to have an adequate expert support by the system. The main task of these users is not the development of large programs but the solution of problems. This problem solving support should be guaranteed in an interactive way. A very important tool in this context is a comfortable dialog instrument. In addition this instrument can help the user in structuring his problems and generating models (see chapter 5).

These method and model base systems in particular are adequate tools for the analyst, the system designer, the planner, the scientist, the manager, the man of practice, the application-oriented man, the system trainer, the system evaluator.

Some years ago the development of data base systems was characterized by the fruitless discussion which system is the most user-friendly and best. After these discussions the data base people agreed upon what they called the 'coexistence view' (cf. [1], [15]). The essential point of this view is the attempt to have for the user views as much freedom as possible. Our opinion is that we should transfer these ideas also to method and model base systems, i.e. we should try to give this freedom to the user. Therefore we need methods for integrating different user views of information systems. We should diminish the conceptual distance between the user views and system's view.

This idea fits some of the requirements of WILL (cf.[21]) who stated that models should easily be adaptable to the changing needs of the users and that these models must reflect their problem view. In addition the method and model base system should provide the various users with the possibility for applying and creating models with respect to their tasks and their knowledge.

Further requirements to such systems are derived from the need to be supported by the system to come to a better problem solution. This could be satisfied by

- concepts and language elements which are easy to learn,
- comfortable fault detecting routines,
- comfortable consistency checks,
- improvement of the communication interface between user and machine and
- improved graphical interfaces.