# Advances of DSS Development

## —— General Systems Theory Approach

Yasuhiko Takahara
Chen Xiaohong
Naoki Shiba

经济科学出版社
Economic Science Press

# Advances of DSS Development

## ——General Systems Theory Approach

*Yasuhiko Takahara*

*Chen Xiaohong*

*Naoki Shiba*



经 济 科 学 出 版 社

**Economic Science Press**

责任编辑：齐伟娜
责任校对：徐领弟　董蔚挺
版式设计：代小卫
技术编辑：邱　天

Advances of DSS Development
——General Systems Theory Approach

Yasuhiko Takahara, Chen Xiaohong, Naoki Shiba 著

经济科学出版社出版、发行　新华书店经销
社址：北京海淀区阜成路甲 28 号　邮编：100036
总编室电话：88191217　发行部电话：88191540
网址：www.esp.com.cn
电子邮件：esp@esp.com.cn
高等教育出版社印刷厂印装
690×990　16 开　20 印张　330000 字
2003 年 6 月第一版　2003 年 6 月第一次印刷
印数：0001—2000 册
ISBN 7-5058-3585-8/F·2893　定价：36.00 元
（图书出现印装问题，本社负责调换）

# About the Authors

## *Yasuhiko Takahara* (高原康彦)

***Chiba Institute of Technology, Japan***

Yasuhiko Takahara received his B.S from the University of Tokyo in 1959, his MS. and his Ph.D. from Case Institute of Technology in 1964 and in 1967, respectively. From 1967 to 1971 he served as a research associate and an assistant professor at Case Western Reserve University. From 1971 to 1979 he was an associate professor and from 1979 to 1995 was a professor of the Department of Industrial Engineering and Management, Tokyo Institute of Technology. From 1995, he moved to Chiba Institute of Technology and since then has been a professor of Department of Management Information Science. He is a professor emeritus of Tokyo Institute of Technology. His research interest is primarily in general systems theory, cybernetic systems and management information system.

## *Chen Xiaohong* (陈晓红)

***Central South University, PRC***

Chen Xiaohong received her B.S and M.S from the Central South University of Technology (which has been renamed as Central South University) in 1983 and in 1986, respectively, her Ph.D. from Tokyo Institute of Technology in 1999. From 1991 to 1994 she was an associate professor and since 1994 she has been a professor of the Department of information decision, Central South University of Technology. She is the winner of "State Science Fund for Outstanding Youth","National Excellent Teachers". She won the" Awards for Young Teachers of Institutions of

High Learning" and "National Excellent Young Teachers (research field) under the Education Fund Association of Huo Yingdong". She has been elected as the academy and technology leader of trans-century at the lever of "hundred" & "thousand" layer in "National Hundred, Thousand, Ten Thousand Talents Engineering Project" by State Personal Ministry, She is as well the prizewinner of "the Top Ten Outstanding Youth of Hunan Province" in 1998, "the Top Ten News Personage of Hunan Province" in 2000. She gained more than 10 items of rewards given by Ministry and Province including the first prize of science and technology progress of Hunan and the first prize of science and technology of Nonferrous Metal Industry Company (ministry rank) . Her research interest is primarily in theory and practice of Information System and Enterprises Financing, Enterprise' s Strategies and Capital Operation.

## *Naoki Shiba* (柴直树)

***Chiba Institute of Technology, Japan***

Naoki Shiba received his B.S.MS. and Ph.D. from Tokyo Institute of Technology in 1986, 1988 and 1994, respectively. From 1988 to 1990 he worked for NTT (Nippon Telegraph and Telephone Co.) Software Research Laboratory in Tokyo as a researcher. From 1990 to 2001 he was an assistant professor of the Department of Industrial Engineering and Management, Tokyo Institute of Technology. Since 2001 he has been an associate professor of Department of Management Information Science, Chiba Institute of Technology. His major interests are systems theory and decision support systems.

# Preface

Our research on DSS started from a joint project with Prof. M. D. Mesarovic, Case Western Reserve University, the United States, about 15 years ago. At that time his group had a DSS generator called Aristotle. The system was originally designed to support his consulting work for developing countries. He was a member of Club of Rome and had already developed a computer system before Aristotle in order to prepare the second report of Club of Rome, "Mankind at a Turning Point". Aristotle was designed based on experiences of the first system.

Since Prof. Mesarovic is a systems scientist, the design of Aristotle was strongly influenced by the systems theory and his General Systems Theory. In particular, it had a model building language which was quite strong in describing a dynamic system. It could handle a vector valued variable easily, which differed it from the System Dynamics of Prof. Forrestor' s.Aristotle was also running on a UNIX workstation and has a graphical user interface, which was a rare case at the time.

Since application area of Aristotle was gradually expanded and problem solving in business was included, Prof. Mesarovic proposed to one of us a joint project to develop an intelligent DSS. Ultimately, the project started involving a system developer, Nomura Computer Systems (the current name is Nomura Research Institute), Japan and our DSS research started using Aristotle at TIT (Tokyo Institute of Technology which we belonged to at the time) as well as in the US.

At the initial stage our group at TIT developed a spreadsheet and an expert system generator for Aristotle for the sake of easy construction of intelligent systems. The spreadsheet worked as an operating system as well as a convenient UI (user interface) . A numerical model and a symbolic model were built in the model description language of Aristotle and the expert system generator, respectively. Model call commands could be stored in cells of the spreadsheet. Data and data call commands were also stored in cells. (Aristotle had a small data base system) . Since a user could arrange these commands and data in an organized form on the spread sheet and their execution control programs were also stored in cells, the spread sheet worked as an OS of the DSS, which had turned it into a dominant component of the system.

Applications like a portfolio decision making support system were developed on the system.

It was immediately recognized that the expert system generator of ours was not so convenient because variables could not used in a model. It did not support matching function of variables. In order to overcome the difficulty Prolog was introduced.

Since a Prolog interpreter was not available, the TIT group developed a Prolog interpreter. It was connected with Aristotle and the spreadsheet. Even if a Prolog interpreter had been available, the group could not have helped developing its own interpreter because of necessity of the connectivity.

At first, the Prolog interpreter was used as a flexible expert system generator. A Prolog model was embedded into a cell. Its execution was controlled by the spreadsheet. But it was soon found that the functions of the spreadsheet, OS and UI, could be realized by a Prolog program in a more convenient and powerful way.

A window system was designed for the Prolog interpreter using the X window system. The data structure of the spreadsheet was integrated into that of the Prolog interpreter. Then, the Prolog system became able to control the spreadsheet. A spreadsheet could be opened on a window and its cells could be accessed and controlled by Prolog predicates. The Prolog interpreter has become a dominant component of our system.

The model describing language of Aristotle was designed on Fortran language. Since it was not compatible with the Prolog interpreter, a new language was designed for model building, namely MDL (model description language) . It inherited some of the basic features of the language of Aristotle. For instance, MDL is a functional language. It is strong in describing a dynamic system. A variable can be a multi-dimensional. Since the internal representation of a model in MDL is a Prolog program, it can completely fit the interpreter.

As MDL is a functional language, capsulation of a model by its IORep (input output representation) was naturally conceptualized. The capsulation leads to concepts of a model space, whose components are IOReps, and model integration which is easily implemented due to the typeless property of Prolog.

Around this time, 1995, one of the authors of this book joined the TIT group from China as a research scholar to stay for half a year. Before coming to Japan, she had already completed practical projects of DSS development. The most serious problem of the TIT group was deficiency of experiences of practical systems development which are indispensable to DSS research. Her joining in the group greatly mended the problem. The system development method or the skeleton approach of this book came from combination of her experiences with a complex hierarchical model of the general systems theory.

Our system was originally implemented on UNIX workstations, first on a Masscomp work station and later on SUN work stations. Both were expensive. However, due to luck of the rapid progress of the PC hard ware, the PC UNIX technologies and open soft wares, in particular, Linux and postgreSQL, we could shift our hard ware base from expensive work stations to cheap PCs. We may be one of the earliest users of Linux. This shift opened a door to research on EUD (end user development) . It is obvious that EUD is an ideal style for a problem solving DSS development because a problem formulation is the most crucial stage of the system development and it is an end user that knows and controls a problem best.

This book is organized according to a layer model, which consists of four layers, i.e. application layer, task layer, logical layer and physical layer. It presents a comprehensive view of DSS. As the above history shows, the joint work with Prof. Mesarovic contributed to development of the task layer and the logical layer. The joining of one of the authors contributed to development of the application layer and the progress of PC technologies contributed to development of the physical layer.

We are still moving.

Quite a few books have been published about DSS. Many of them are written by management scientists. This book may be an only book which attacks DSS from formal systems theoretic view point. DSS is certainly a system. There must be a systems approach to develop it.

This book uses Prolog as an implementation language although it is not a so popular language. The reason we adhere to it comes from the character of the book, which emphasizes the systems approach and the formal approach. We believe in a bright future of Prolog due to the remarkable progress of hard wares. As far as we can see, execution speed is practically the only problem with Prolog for problem solving systems development.

This book is a summary of contributions of many friends, colleagues and former students of ours. If the book is worthy of reading, it is due to their contributions. In particular, we must express the deepest thanks to Nomura Research Institute and Mr. Toda who was Vice President of the Institute, for the financial and personnel support for the project. But for the help the project and hence this book could not exist.

As the above history indicates, our research owes Prof. Mesarovic a huge amount of help. His imagination, encouragement and guidance were indispensable to our research. Originally, he was supposed to be one of the authors of this book. It is a pity that he could not join us due to his time constraints.

The final statement is an obvious one that if the book cannot be appreciated well, it is due to our faults.

fall, 2002

Y. Takahara

X. Chen

N. Shiba

# Contents

## Part Ⅲ: APPLICATION

## Chapter 8



## Chapter 9

# Part I

# CONCEPTUAL FOUNDATION

# Chapter 1

# Basic Concepts

This book may not be considered as an introductory textbook for a DSS (decision support system). We want to develop a comprehensive view on a DSS from the so-called specific DSS to a DSS generator and its physical realization from a theoretical basis to real application. We want to discuss the topics as deeply as possible. In order to avoid superficial arguments we will use our real working system, actDSS, as a reference model of this book. This trial, however, has a danger to make the book lead to a "specialized meaningless". In order to avoid the danger the book adopts the strategy that systems, whatever they may be, are formalized in the spirit of the MGST (mathematical general systems theory) or by GST (general system theory) concepts using the set theory [1,2] and their implementations are examined using a logic programming language, Prolog. Prolog can accept a structural formulation by the set theory and execute it on a computer system [3]. The working computer system will be discussed in Chap. 5.

It can be said, then, that an official description language of this book is the GST on the set theory. Its implementation language is the extProlog (extended Prolog) and every argument is supported by a real working implementation. The GST and the Prolog must be considered as general tools.

This book has another feature that it emphasizes EUD (end user development) as one of the most important application field of a DSS generator. As Sec. 1.5 will discuss, EUD must be an important issue of the current information technology. A DSS generator can make a significant contribution to implementation of the EUD concept. One of the objectives of this book is to demonstrate the claim is true.

## 1.1 MGST (*Mathematical General Systems Theory*)

This section will explain why MGST can be used for the study of information systems.

It is obvious that an information system is a system. Then, an information system must be a legitimate target of the MGST. We have made the following statements about the MGST in Ref. [1].

"Systems theory is a scientific discipline concerned with the explanations of various phenomena, regardless of their specific nature, in terms of the formal relationships between the factors involved and the ways they are transformed under different conditions; the observations are explained in terms of the relationships between the components, i.e., in reference to the organization and function rather than with an explicit reference to the nature of the mechanisms involved (e.g., physical, biological, social, or even purely conceptual). The subject of study in systems theory is not a physical object, 'a chemical or social phenomenon, for example, but a system': A formal relationship between observed features or attributes. For conceptual reasons, the language used in describing the behavior of systems is that of **information processing and goal seeking** (decision making, control)."

Furthermore, the reference book says that "General systems theory deals with the most fundamental concepts and aspects of systems. Many theories dealing with more specific types of systems (e.g., dynamical systems, automata, control systems, game-theoretic systems, among many others) have been under development for quite some time. General systems theory is concerned with the basic issues common to all of these specialized treatments. Also, for truly complex phenomena, such as those found predominantly in the social and biological sciences, the specialized descriptions used in classical theories (which are based on special mathematical structures such as differential or difference equations, numerical or abstract alge-