Interactive Systems

Interactive Systems

ISBN 0 903796 06 6

Published by Online Conferences Limited, Uxbridge, England

Printed by Paca-Press & Co. Limited, West Drayton, England

Introduction

By D W Lewin, Professor of Digital Systems, Brunel University and Chairman of The European Computing Conference on Interactive Systems.

Interactive computer systems is a subject of great, and rapidly increasing, importance which is becoming relevant to all fields of technology but especially in the area of engineering design. The term interactive systems is used generally to describe computer architectures (comprising both hardware and software components) which enable the user to interact directly with the system and application software to observe, control and optimise its performance.

Though many interactive systems have been proposed in the literature, especially in the area of computer-aided design, the concept has still not been fully exploited and a large number of problems remain to be solved. Many of these problems are fundamental to the design of any large software complex, in addition however interactive systems present their own special difficulties. In particular, the areas of man-machine communications, high-level descriptive languages, fast computational techniques, data-base management and the use of graphic techniques, require special consideration.

The papers presented in this volume have been broadly classified into two main areas consisting of fundamental aspects of system design and application studies. The section on fundamental techniques comprises papers on software tools, such as debugging systems and program generators, programming languages, graphic languages and systems, man-machine interaction and theoretical methods. Among these papers it is worth noting the requirement for special languages for interactive use, and in particular, the question-answering philosophy described by Konopasek. Also indicative of possible future trends is the use of fuzzy set theory by Ishikawa and Mieno in the design of video information systems and the application of associative storage methods in data-base manipulation as described by Amirchahy and Neel. On the hardware side the use of device independent graphics in a time-shared mode, as discussed by Bruce, promises to become an established technique.

The application papers range over a very wide area, from medicine to cartography, indicating the extent to which interactive systems are rapidly becoming an established design tool. This section also includes a new and exciting application area for interactive systems, that of digital image processing. The technique, reviewed definitively by Billingsley, allows the digital reconstruction and evaluation of pictorial information under direct user control.

Though the application studies presented demonstrate the utility and versatility of interactive computing they nevertheless expose some of the problems current in this field. For example, the lack of standardisation in application languages, the need for common and portable data-bases, the requirement in C.A.D. for three-dimensional displays etc. It is hoped that the presentation of these papers will help to highlight some of these problem areas and encourage others to direct their energies to effect a solution.

Douglas Lewin

Preface

This volume comprises papers presented at The European Computing Conference on Interactive Systems — London, September 1975. To these have been added a number of other relevant papers for which there was not time for presentation at the Conference.

In order to ensure that the technical information is as up-to-date as possible, we have opted for direct reproduction of typed masters. For the same reason, these appear in random order throughout the book and are logically sequenced via the index.

Online wishes to thank Douglas Lewin, the General Conference Chairman, and all the undermentioned Session Chairmen for their diligent assistance throughout.

L Bolliet	Professor, Institut de Mathematique et Informatique, France
J Encarnacao	Doctor, University des Saariandes, Federal Republic of Germany
C Evans	Doctor, Head of Man-Computer Interaction Group, National Physical Laboratory, UK
B Higman	Professor, University of Lanchester, UK
F R A Hopgood	Head of Basic Software Group, Atlas Computer Laboratory, UK
A I Llewelyn	Director, Computer Aided Design Centre, Department of Industry, UK
M Maison	Technical Orientation Manager, Compagnie Internationale pour l'Informatique, France
J J Quann	Chief, Information Extraction Division, National Aeronautics and Space Administration, USA
G Valle	Professor, Universita di Bologna, Italy

This Conference was part of The European Computing Congress 1975 — EUROCOMP. Other conferences in this series investigate:

Business Data Processing
Communications Networks
Systems Design and Implementation
Systems Evaluation

Online is Europe's leading specialist in the design and organisation of international congresses and conferences on advanced computing technology. Major regular events include: Eurocomp — The European Computing Congress, Medcomp — International Congress on Medical Computing, and Minicomputer Forum.

Online (Head Office)
Cleveland Road
Uxbridge, Middlesex
England
Telephone: Uxbridge 39262

Online Nederland
Regentesselaan 182
The Hague
The Netherlands
Telephone: The Hague 336004

Index

Software Engineering

Programming Languages

Graphics Languages

Graphics Systems

Man-Machine Interaction

Theory and Techniques

Digital Image Processing

Applications

Authors

Build program technique: objectives, processes, and practices

John G. Rice
Research Analyst
Olson Research Associates, Inc.
USA

ABSTRACT

The Build Program Technique (BPT) is an approach that increases the reliability of software. BPT shifts the emphasis from highly specialized programming techniques towards a more thorough analysis of information requirements.

Fundamental to the process and its objectives are (1) the automatic construction of customized modules vis-a-vis the requirements specified by the problem-definer and (2) the use of standardized subassemblies called "build-programs" to construct the modules. Specifications are communicated to the build-programs by a problem-statement language.

BPT has increased software production efficiency and provided for ease of maintenance and change. A practical application of BPT for a financial planning system is presented for tutorial purposes.

INTRODUCTION

The Build Program Technique (BPT) is a method for increasing the reliability of software by establishing control over the design, development and construction of the software product.

This paper presents the concepts and processes employed by BPT to automatically construct a time-sharing software product. These concepts and processes are discussed in light of some of the current trends in software architecture: composite design, structured programming and top-down development. A practical application which enjoys the benefits of BPT is discussed to illustrate the advantages and limitations of BPT.

The process involves construction of customized modules vis-a-vis the requirements specified by the problem-definer and the use of standardized subassemblies called "build-programs" to construct these modules. Specifications are communicated to the build-programs by a problem-statement language, hereafter, denoted as the Build Definition Language (BDL). BDL is developed for each application of the build-programs. It consists of verbs and descriptors that are familiar to the problem-definer.

The increase in the complexity of hardware and software, lack of programming and documentation standards, and high personnel turnover make software development overly dependent on the so called "professional". The existence of a "professional" staff to support the product is a function of availability and affordability.

Functionally, programs may be similar but they are usually coded differently according to highly individualized programming techniques. Hence, software tends to vary both with each programmer as well as with the quantity and quality of programming staff. In addition, cost to maintain and support the product increases with every change, and reliability of the product decreases with each change.

BPT shifts the emphasis from highly individualized programming techniques towards a more thorough analysis of product requirements. It demands a specific definition of requirements which is transformed into modular structures. The structures have the product's design objectives as their foundation. This insures that the development process incorporates the definition of requirements into the solution. Hence, BPT produces a standard and reliable product which is easy to maintain or change.

BPT is the result of an experiment aimed at improving the construction process for a time-sharing software product. Experiments were conducted for two basic data processing functions; namely, the preparation of output and non-numeric and numeric computations. Work with BFT has progressed from experimentation to routine use of BPT for constructing a software system. Future experimentation will be directed to the preparation of input using BPT.

THE BPT PROCESS

The application of BPT requires the completion of a series of tasks. These tasks can be scheduled and progress can be measured. Six tasks currently used are:

1. Describe the logical and physical relationships of the data that are known by the user.
2. Classify known business data functions.
3. Classify the degree of similarity of problems and information needs of one user versus another into programmable structures.
4. Develop a problem-statement language to communicate the user requirements into machine processable form.
5. Develop a set of build-programs to structure and generate programs that are machine processable and satisfy the problem-statement objectives.
6. Validate the user functions communicated by the problem-statement and the structures generated by the build-programs against existing applications.

Data structure. In the first task, all of the entities of the data base known to the user are identified and described. Essential to the success of BPT is the clear understanding of the data base structure.

The data base is conceived as the fundamental repository of relevant data not only for the retrieval of information, but also for the planning and control of the operating unit (corporation, bank, division, etc.). To effectively serve these fundamental needs, it is essential that a common and controlled approach by used to identify, describe and record the logical and physical relationships of the user known entities that characteristically reflect the structure of the data base. (1) These entities reflect the user's knowledge about the data that should exist in his system. Experience to this date has been in accessing data from a File System, where the user is aware of the presence and structure of files.

Business data functions. The second task is to identify and classify all of the business data functions known by the user. The business data functions are basically designed to facilitate the interface between the user and programmer and/or the programmer and the machine. These functions can be classified into one of the following types of data processing functions:

1. preparation of output
2. acceptance of input
3. preparation of input
4. updating and maintenance of files
5. numeric and non-numeric computations

The set of functions selected in this task should be complete enough to cover all given data processing functions. Also, it should be possible to split any given problem into a set of functions with no missing pieces. Furthermore, the set of functions should be orthogonal, i.e., what is accomplished in one should not be duplicated in any of the others.

There are essentially three methods used to classify the set of functions that describe the software system.

1. Examine the existing programs which are currently a part of the software product and attempt to determine an optimum set of functions.
2. Analyze the existing requirement for processing the transactions of an organization that will use the software product and define a set of functions that will accomplish the required processing.
3. Develop a set of general functions based on theoretical considerations of future requirements in anticipation that they will serve the same purpose as a high-level programming language.

Initially, the first and second methods are used to fulfill the second task of the BPT process. Ultimately, the third method will evolve from the iterative processes and feedback from the user's evolving requirements and the programming staff. To this end, the stated methods are a starting point for the analyst.

Degree of similarity. The third task is to classify the degree of similarity of problems and information needs among users into programmable structures. This task compliments the previous two tasks by structuring program logic and code to serve the needs of each user. Each program will contain the following elements:

1. inputs to be processed
2. operations to be performed on these inputs
3. outputs to be produced

In comparing two problems, the degree of similarity may be categorized as identical, similar or dissimilar. Regarding the data to be processed (i.e., those entities identified and classified in the first task) the comparison is concerned with:

1. number of fields
2. their significance
3. formats to be used

In comparing the operations to be performed (i.e., those business data functions known by the user) one must not only compare the operations themselves but also their logical sequence. Phillips proposes that these classifications can be represented by the following table. (2)

Comparison of two or more problems		data: identical	data: similar	data: dissimilar
operations	identical	1	3	5
	similar	2	4	6
	dissimilar	7	8	9

Out of the nine possibilities only 1 through 4 are of practical value. The significance of the possibilities 1, 2, 3 and 4 are shown in the following table.

Comparison of two or more problems		data	
		identical	similar
operations	identical	1 fixed subroutine	3 customized sub-routine with para-meters for data
	similar	2 customized sub-routine to handle unique operations	4 customized sub-routine with para-meters for data & to handle unique operations

Experience to date has been focused on possibilities 1, 2, and 3. The last possibility is currently under experimentation for other applications of BPT.

By making the above comparison, each problem and its inputs, operations and outputs will be fragmented into a collection of basic modules. The function of the module is a description of the transformation (input to output) that occurs when the module is called. It is important to note that the function is related not only to the operations performed in that module, but also the functions of any module called by that module.

The motivation behind the concept is clarity. The primary design goal is to restructure existing programs or structure new programs into highly independent modules. This can be accomplished by:

1. minimizing the relationships among modules called "module coupling" and
2. maximizing module strength by maximizing the relationships among the parts of each module.

Each module, from the highest level to the most basic, will have a logical structure. The concept is to force each module's logic into a sequential, top-down structure.