DISTRIBUTED COMPUTER CONTROL SYSTEMS 1982

Edited by
R. W. GELLIE & R.-R. TAVAST

DISTRIBUTED COMPUTER CONTROL SYSTEMS 1982

Proceedings of the Fourth IFAC Workshop
Tallinn, U.S.S.R., 24-26 May 1982

Edited by

R. W. GELLIE

Commonwealth Scientific & Industrial Research Organization
Division of Manufacturing Technology
Fitzroy, Australia

and

R.-R. TAVAST

Academy of Sciences of Estonian S.S.R.
Institute of Cybernetics
Tallinn, U.S.S.R.

Published for the

INTERNATIONAL FEDERATION OF AUTOMATIC CONTROL

by

PERGAMON PRESS

OXFORD · NEW YORK · TORONTO · SYDNEY · PARIS · FRANKFURT

U.K. Pergamon Press Ltd., Headington Hill Hall,
Oxford OX3 0BW, England

U.S.A. Pergamon Press Inc., Maxwell House, Fairview Park,
Elmsford, New York 10523, U.S.A.

CANADA Pergamon Press Canada Ltd., Suite 104,
150 Consumers Road, Willowdale, Ontario M2J 1P9, Canada

AUSTRALIA Pergamon Press (Aust.) Pty. Ltd., P.O. Box 544,
Potts Point, N.S.W. 2011, Australia

FRANCE Pergamon Press SARL, 24 rue des Ecoles,
75240 Paris, Cedex 05, France

FEDERAL REPUBLIC OF GERMANY Pergamon Press GmbH, Hammerweg 6,
D-6242 Kronberg-Taunus, Federal Republic of Germany

First edition 1983

Library of Congress Cataloging in Publication Data

IFAC Workshop on Distributed Computer Control
Systems (4th: 1982: Tallinn, Estonia)
Distributed computer control systems 1982.
(IFAC proceedings)
1. Automatic control--Data processing—Congresses.
2. Electronic data processing—Distributed processing—
Congresses. I. Gellie, R.W. II. Tavast, R.-R.
(Raul-R.) III. International Federation of Automatic
Control.
TJ212.2.134 1982 629.8'95 83-2388

British Library Cataloguing in Publication Data

IFAC Workshop DCCS: (*4th 1982: Tallinn)*
Distributed computer control systems 1982.—
(IFAC Proceedings)
1. Automatic control - Data processing—
Congresses 2. Electronic data processing—
Distribution processing—Congresses
I. Title. II. International Federation of
Automatic Control. III. Gellie, R.W.
IV. Tavast, R.R. V. Series
629.8'95 TJ212

ISBN 0-08-028675-5

In order to make this volume available as economically and as rapidly as possible the authors' typescripts have been reproduced in their original forms. This method unfortunately has its typographical limitations but it is hoped that they in no way distract the reader.

Printed in Great Britain by A. Wheaton & Co. Ltd., Exeter

FOURTH IFAC WORKSHOP ON DISTRIBUTED COMPUTER CONTROL SYSTEMS

Organized by

U.S.S.R. National Committee on Automatic Control
Institute of Cybernetics, Academy of Sciences of the Estonian S.S.R.
Tallinn Technical University

Sponsored by

IFAC Technical Committee on Computers

Co-sponsored by

IFAC Technical Committee on Education

Workshop Chairman

Academician B. Tamm

International Program Committee

R.-R. Tavast, U.S.S.R. (Chairman)
V. G. Bochmann, Canada
R. W. Gellie, Australia
J. Gertler, Hungary
T. J. Harrison, U.S.A.
Th. Lalive d'Epinary, Switzerland
L. Motus, U.S.S.R.
K. D. Müller, F.R.G.
S. Narita, Japan
E. A. Trachtengerts, U.S.S.R.

PREFACE

IFAC Workshops have proven to be a popular and effective forum for presentation and in-depth discussion of ideas by competent experts in emerging areas of automatic control. Indeed, the Workshops on Real Time Programming, which date back to 1971 and have been held annually since that time, have proved so successful that IFAC has published a special booklet "Guidelines for Organizers of IFAC Sponsored Workshops" to assist and encourage more events of this type.

During his term as Chairman of the IFAC Computers Committee, Mr. Charles Doolittle established the Workshops on Distributed Computer Control Systems in recognition of the great interest and activity in this area. The first event in this series was held in Tampa, Florida (1979) with subsequent events in Ste. Adele, Quebec (1980), Beijing, China (1981) and now Tallinn, Estonia in 1982. Every event in this series has been very successful in terms of quality of papers, numbers of participants, and the degree to which the attendees have contributed to lively discussson and exchange of ideas.

As can be noted from the list of participants the Tallinn Workshop was attended by 75 experts from 14 countries. The quality of the papers presented and the discussions which followed may be judged by the reader.

In these proceedings the papers are published in the order in which they were presented. The discussions which took place were recorded and subsequently transcribed. Some editing was done to improve clarity and avoid repetition but it is hoped that the text retains a sense of spontaneity. The Workshop program concluded with a panel session. The initial presentations by the panel members and the ensuing discussion are also included.

The two papers by H.G. Mendelbaum, G. de Sablet., and Wu Zhimei, Zhang Wenkuan, Zhang Yingzhong, Cheng Yunyi were not presented at the Workshop because the authors were unable to attend. However we felt that their inclusion would add to the value of this volume.

I wish to gratefully acknowledge the part played by my co-editor, Raul Tavast, who performed the difficult task of transcribing and editing the discussions.

R.W. Gellie.
November 23, 1982.
Fitzroy, Victoria, Australia.

LIST OF PARTICIPANTS

IFAC 4th WORKSHOP ON DISTRIBUTED COMPUTER CONTROL SYSTEMS DCCS-82

Olympic Yachting Centre, Tallinn, Estonian S.S.R., U.S.S.R., 24-26 May 1982

O. Aarna
Tallinn Technical University
Ehitajate tee 5
Tallinn 200026 USSR

H. Aben
Institute of Cybernetics
Akadeemia tee 21
Tallinn 200026 USSR

A. Abreu
Central Institute of Digital Research
198 No. 1713 Cubanacan
Havana CUBA

Z. Apostolova
State Committee of Science and Technology,
ul. Slavjanska 8
Sofia BULGARIA

A. Ariste
Institute of Cybernetics
Akadeemia tee 21
Tallinn 200026 USSR

S.I. Baranov
Institute for Analytical Instrumentation
Sci. & Techn. Corp.,
USSR Academy of Sciences
Prospect Ogorodnikova 26
198103 Leningrad USSR

B. Becski
Technical University of Budapest
Visepradi u.30
1132 Budapest HUNGARY

G. Bingzhen
Research Institute of Electronical Technical Applications
Beijing CHINA

J. Davidson
Tecsult International Limité
85, rue Ste-Catherine Ouest,
Montreal, Quebec H2X 3T4 CANADA

C. Dimitrov
Institute for Scientific Research in Telecommunications
Haidushka Poljana str. 8
Sofia 1612 BULGARIA

D.G. Dimmler
Senior Scientist
Brookhaven National Laboratory
Upton.
New York 11973 USA

A. Divitakov
ZNIKA
str. I. Vishovgradsky 46,
Room 608 Sofia BULGARIA

V.N. Dragunova
INEUM
Moscow V-334 117812 GSP USSR

T.O. Dzjubek
Institute of Cybernetics
Academy of Sciences
Kiev USSR

J. Ehrlich
Technische Hochschule Leipzig
Karl-Liebknecht-Strasse 132
DDR-703 LEIPZIG

W. Engmann
Technische Hochschule Ilmenau
DDR-63 Ulmenau-Ehrenbert

W. Enkelmann
Zentralinstitut für Hernforschung, Rossendorf
8051 Dresden, P.O. 19
DDR

G. Evstratov
Politechnical Institute Kharkov
Frunze str. 21.
310002 Kharkov USSR

A. Gościński
Institute of Computer Science
Stanislaw Staszik
University of Mining & Metallurgy
al. Mickiewicza 30
30-059 Krakow POLAND

*,** T.J. Harrison
IBM Corporation
P.O. Box 1328 Boca Raton
Florida 33432 USA

H. Hetzheim
Academy of Sciences
GDR

I.G. Ilzinya
Institute of Electronics &
Computer Systems
Akademijas 14
Riga 6 226006 USSR

* A. Inamoto
Computer Systems Works
325 Kamimachiya
Kamakura City
Kanagawa Prefecture
JAPAN 247

U. Jaaksoo
Institute of Cybernetics
Akadeemia tee 21
Tallinn 200026 USSR

H. Jinwei
System Development Division of China
Computer Technical Service Corporation
Beijing CHINA

K.A. Joudu
Moscow Institute of Avionics
Volokolamskoe 4
Moscow GSP 125871 USSR

** A. Keevallik
Tallinn Technical University
Ehitajate tee 5
Tallinn 200026 USSR

B. Kovacs
Computer & Automation Institute
Hungarian Academy of Sciences
Kende u. 13/17
1111 Budapest HUNGARY

K. Kralev
State Committee of Science &
Technology
ul. Slavjanska 8
Sofia BULGARIA

V. Krüger
Academy of Sciences
GDR

*,** K. Kääramess
Institute of Cybernetics
Akadeemia tee 21
Tallinn 200026 USSR

* J. Lan
Department of Computer Engineering &
Science
Qinghua University
Beijing CHINA

J. Lukács
Central Research Institute for Physic
P.O. Box 49,
H-1525 Budapest HUNGARY

U. Luoto
Ekono Oy
P.O. Box 27
SF-00131 Helsinki FINLAND

G.G. Mask
Central Bureau of Statistics
Endla 15
Tallinn 200105 USSR

M. Mantseva
State Committee of Science and
Technology
ul. Slavjanska 8
Sofia BULGARIA

M. Martin
Zentralinstitut für Kybernetik
und Informationsprozesse
Dresden 8027 Dresden
Maeskelstr. 20 DDR

M. Maxwell
Manager of Control Systems
Colgata-Palmolive Co
105 Hudson St.
Jersey City
New Jersey USA

I. Meiszterics
Technical University of Budapest
Vérhalom u.29
1025 Budapest HUNGARY

*,** L. Motus
Institute of Cybernetics
Akadeemia tee 21
Tallinn 200026 USSR

*,** S. Narita
Electrical Engineering Dept.
Waseda University
3-4-1 Okubo, Shinjuku-ku
Tokyo 160 JAPAN

L.F. Natiello
Exxon Research & Engineering Company
P.O. Box 101
Florham Park
New Jersey 07932 USA

D. Nedo
Central Institute of Cybernetics &
Information Processing
Kurstr. 33
1086 Berlin GDR

* M. Ollus
Technical Research Centre
Electrical Engineering Lab.
VTT/SAH
SF-02150 Espoo 15 FINLAND

Z. Pengzu
Lab. North-China Institute of
Computing Technology
Beijing CHINA

D. Penkin
Soviet-Bulgar Institute
Stambulinskaja 62-64
Sofia BULGARIA

V.L. Pertchuk
Institute of Automation and
Control Processes
Suhhanova 5A
Vladivostok 690600 USSR

B. Petkov
Ministry of Communication
Computing Centre
Sofia BULGARIA

K. Petrov
State Committee of Science &
Technology
ul. Slavjanska 8
Sofia BULGARIA

J. Pino
Central Institute of Digital Research
198 No. 1713 Cubanacan
Havana CUBA

W. Qinsheng
Computer Industry of the Ministry of
Electronical Industry
Beijing CHINA

* M.G. Rodd
University of the Witwatersrand
1 Jan Smuts Avenue
Johannesburg 2000 SOUTH AFRICA

L. Rozsa
Computer & Automation Institute
Hungarian Academy of Science
Kendue u. 13
Budapest 1111 HUNGARY

* S.M. Rotanov
Institute of Electronics & Computer
Systems
Akademijas 14
Riga 6 226006 USSR

I.M. Shenbrot
Central Institute of Complex
Automation
Olhovskaya 25
Moscow B-66 107816 USSR

A.V. Shileiko
Moscow Institute of Railway Engineers
F. Engels 36-25
Moscow B-5 107005 USSR

B.T. Shreiber
Central Institute of Complex
Automation
Olhovskaya 25
Moscow B-66 107816 USSR

V. Slivinskas
Institute of Mathematics & Cybernetics
K. Pozelos 54
Vilnius 232600 USSR

A.A. Sternberg
INEUM
Moscow V-334 117812 GSP USSR

* G.J. Suski
Lawrence Livermore National Laboratory
P.O. Box 808,
Livermore
California 94550 USA

* K. Takezawa
Toshiba Fuchu Works - Toshiba Corp.
1, Toshiba-cho, Fuchu
Tokyo 183 TOKYO

** B. Tamm
Tallinn Technical University
Ehitajate tee 5
Tallinn 200026 USSR

H. Tani
Institute of Cybernetics
Akadeemia tee 21
Tallinn 200026 USSR

** R. Tavast
Institute of Cybernetics
Akadeemia tee 21
Tallinn 200026 USSR

I. Tepandi
Tallinn Technical University
Ehitajate tee 5
Tallinn 200026 USSR

P. Tianchian
Research Institute of Metallurgical
Automation
Beijing CHINA

*,** E.A. Trakhtengerts
Institute of Control Problems
Profsoyuznaya 81
Moscow V-279 117806 USSR

I.S. Ukolov
Scientific Council of Cybernetics USSR
Academy of Sciences
40, Vavilov str.
117333 Moscow USSR

V.I. Utkin
Institute of Control Problems
Profsoyuznaya 81
Moscow V-279 117806 USSR

T. Vamos
Computer & Automation Institute
Hungarian of Academy of Sciencies
P.O. Box 63
Budapest 112 HUNGARY

L. Vohandu
Tallinn Technical University
Ehitajate tee 5
Tallinn 200026 USSR

B. Werner
VEB PCK Schwedt
DDR-1330 Schwedt DDR

* G.G. Wood
Foxboro Yoxall
Redhill
Surrey RH1 2HL UK

** A. Work
Institute of Cybernetics
Akadeemia tee 21
Tallinn 200026 USSR

K. Zieliński
Academy of Mining & Metallurgy
Institute of Computer Science
al. Mickiewicza 30
30-059 Krakow POLAND

* Author

** Program/Organizing Committee

WELCOME ADDRESS

T. Vamos
IFAC President

On behalf of IFAC Council I am happy to address this meeting not only because it is my duty and privilege but also because I can express my satisfaction with IFAC's directions of interest. To tell the truth, we would like to organize not only fashion directed meetings, but such ones which go somehow before the general fashion, which predict from the well-estimated trends, stimulate exchange experiences and after that summarise in order to provide something relevant to our control community.

If this symposium were the first of the distributed control series I would say did we sleep before? 1979 was just the right time to begin and, coming here, it is appropriate to speak about experiences about unsolved problems in more depth.

Let me tell you some personal views on the topic which I consider to be even more revolutionary than is generally advertized. Distributed control is much more significant in control philosophy than decentralized control was after the hierarchical centralized ones.

Decentralized control is mostly a system which is really de-centralized: that means the system as a whole, handled as a black box, is a resolved and reorganized centralized one, not losing the strong central control but only delegating some tasks and some information to the lower subjugated levels.

Distributed control is a much more liberal solution indicating a highly cooperative philosophy: a coalition of components arranged not hierarchically but in a very democratic coordinative way. Any centralization loses its rationality as we consider larger and larger systems, systems which have no rigorous physical limitations but can be augmented or dissolved adaptively. A very big system is even theoretically uncontrollable in the old sense, due to combinational explosion and due to randomness of parameters. It's nevertheless controllable as a coalition of autonomous system partners, which cooperate through a well-defined network of information and flow. Flow is understood as anything different from information, e.g. flow of energy, fluids, goods, people, etc. High information transmission technology, high level exchange protocols, reliable well-organized system components are basic requirements and that's the reason why we had to wait until now for this revolution. Several earlier systems like international telephone exchange and some power systems started to realize these ideas long before they were formulated. It's my firm belief that for the future the idea of distributed cooperative systems will be the dominating control principle, for every large-scale system which intends to survive and develop.

IFAC itself is a cooperative system and it is appropriate to express our gratitude to those who cooperate in IFAC. Distribution of information, call for papers and people are organized by our forty national organizations by voluntary, non-profit groups. Participation is free, without any kind of discrimination, independent of the transients of government policies. Our goals and aims are more global and more stable.

Thanks to the International Program Committee of this Workshop that shaped the program which is of interest not only for those who could come here at this time, but for all people who look at IFAC's events for information and stimulation.

Thanks to the local organizers headed by Vice-President of IFAC, Chairman of our Technical Board, Academician Boris Tamm.

Soovin koige paremat. Soovin edu.
Best wishes and big success.

WELCOME ADDRESS

B. G. Tamm

Workshop Chairman, Tallinn Technical University

On behalf of the National Organizing Committee and International Program Committee I am glad to extend a sincere welcome to everybody participating in the 4th Workshop on Distributed Computer Control Systems here in old Tallinn.

Just recently I received two telegrams, one from Dr. W. Gellie, Chairman of the sponsoring committee of this workshop, IFAC TC on Computers, the other from Professor P. Larsen, Chairman of the co-sponsoring committee, IFAC TC on Education both sending their best regards for the success of our workshop and asking me to transfer that to you.

Our workshop is already the fourth in the series of Distributed Computer Control after those held in Tampa (1979, USA) Ste. Adele (1980, Canada) and Beijing (1981, PRC), so we ought to have some kind of experience.

Nevertheless the topic of our interest is developing dramatically and I know the hard efforts of the members of the International Program Committee in selecting the best papers from among those submitted. I should like to thank everybody who sent contributions and congratulate the authors of the papers selected for the Final Technical Program.

As you know IFAC is a society of volunteers consisting of specialists in automatic control who are ready to undertake personal efforts besides their everyday jobs, in order to promote science and technology. In this respect I should like to thank every member of IFAC, especially the Chairman, Raul Tavast, Tom Harrison, who has had a hand in all four of the workshops, and L. Motus and E. Trachtengerts, members from the USSR NMO, for their outstanding contribution to this IFAC event.

Dear guests, I should also like to assure you that the volunteers from the Institute of Cybernetics, Academy of Sciences of the Estonian SSR and Tallinn Technical University have done their best to create a fruitful professional atmosphere as well as to ensure your joyful stay in Tallinn.

CONTENTS

Welcome address - T. Vamos, President of IFAC xvii

Welcome address - B. Tamm, Workshop Chairman xix

Session 1. R.-R. Tavast, Chairman

Software Design Specifics for Multiprocessor Control Systems
E.A. Trakhtengerts, Yu.M. Shuraits 1

Discussion 11

Session 2. L.F. Natiello, Chairman

IEEE Project 802: Local Area Network Standard
A March 1982 status report
T.J. Harrison 13

Discussion 25

Standardization Work for Communication Among Distributed
Computer Control Systems
G.G. Wood 27

Discussion 31

Session 3. L. Rosza, Chairman

Development and Quantitative Evaluation of Distributed
Sensor Based Management System
T. Muto, C. Inamochi, A. Inamoto, S. Kato 33

Discussion 43

Nova Control System - Goals, Architecture and System Design
G.J. Suski, J.M. Duffy, D.G. Britton, F.W. Halloway,
I.R. Krammen, R.G. Ozarski, J.R. Sevevyn, P.J. Van Arsdall 45

Discussion 57

Session 4. A. Work, Chairman

A Memory Intensive Functional Architecture for Distributed
Computer Control Systems
D.G. Dimmler 59

Discussion 73

Implementation and Performance Evaluation of a Distributed
Splitted-Bus Multiple Computer System
Lan Jin, Wei-min Zheng, Ding-xing Wang, Mei-ming Sheng 75

Discussion 81

Session 5. M. Maxwell, Chairman

On the Design of Hierarchical Process Control Computer Systems
J. Davidson, J.L. Houle 83

Discussion 91

A Model Based Design of Distributed Control Systems Software
L. Motus, K. Kääramees 93

Discussion 101

Session 6. M.G. Rodd, Chairman

Parallel Processing for Real-Time Control and Simulation of DCCS
H. Kasahara, S. Narita 103

Discussion 115

On the Distribution of Tasks in Automation Systems
M. Ollus, B. Wahlström 117

Discussion 127

Session 7. Lan Jin, Chairman

Software Test Facilities with Distributed Architecture
K. Takezawa 129

Discussion 139

A Communication System for Use in an Industrial Distributed Control System
M.G. Rodd, N.J. Peberdy, H.F. Weehuizen, D.P.A. Bean 141

Discussion 151

Session 8. G.G. Wood, Chairman

Development and Analysis of Protocols for Distributed Computer Networks
E.A. Yakubaitis, Ya.A. Kikuts, S.V. Rotanov 153

Discussion 165

Ring Computer Networks for Real Time Process Control
A. Gościński, T. Walasek, K. Zieliński 167

Discussion 179

Session 9. T.J. Harrison, Chairman

Panel Discussion

Bottlenecks in the Design and Implementation of DCCS and the Ways to Fight Them.
T.J. Harrison 181

Introductory Statements:
S. Narita 183
M. Maxwell 189
G.G. Wood 191
R.-R. Tavast 193

Discussion 195

Additional Papers

EMUNET-Emulator of Network Systems
H.G. Mendelbaum, G. de Sablet 201

The Communication Software on a Node of the RDC Network
Wu Zhimei, Zhang Wenkuan, Zhang Yingzhong, Cheng Yunyi 211

Author Index 217

Copyright © IFAC Distributed Computer Control Systems
Tallinn, U.S.S.R. 1982

SESSION 1
Chair: R.-R. Tavast

SOFTWARE DESIGN FOR MULTIPROCESSOR SYSTEMS COMPUTER CONTROL

E. A. Trakhtengerts and Yu. M. Shuraits

Institute of Control Sciences, Moscow, USSR

Abstract. The paper treats the specifics of the multiprocessor computer systems software design, namely the design of language means, translation and program parallelization means, operation systems and debug systems. Feasibility of parallel and serial program execution, the influence of the execution mode on the useful performance of a computer system and the reduction of execution time for a group of tasks or mean service time for a flow of queries are considered.

Keywords. Multiprocessor computer system; parallel computations; automatic parallelization; program branch; usefulness of parallelization; multiprocessor computer system performance.

INTRODUCTION

The advent of multiprocessor computer systems capable of performing parallel computations put forward new requirements to the software which realizes control algorithms. These requirements may be verbalized in a single phrase: parallel performance of the computer process.

The process of parallelization imposes specific requirements on
-language means of programming for multiprocessor systems which should make parallel computations possible and provide their timing;
- means of translation which should provide automatic parallelization of the object program;
- means of debug controlling asynchronous performance of certain tasks and localizing errors in running the program on certain processors;
- operation system distributing computational resources and providing its own parallel operation;
- design of algorithms taking full advantage of parallel computations;

The first part of the paper treats various ways to realize the first four requirements.

The second part discusses rational combination of the parallel and serial forms of program execution, the effect of program parallelization upon the useful capability of a multiprocessor system and some problems pertaining to the design of parallel algorithms.

PART I

1. Language means

The language means in multiprocessor systems are intended for the organization of serial - parallel computations. They differ from the "traditional" programming languages in that they incorporate some additional units providing parallel realization of program fragments and its timing.

These include
- the introduction of vector and matrix operations and means of masking the operations on elements of the vectors;
- the apparatus for the creation of sections of parallel program execution which are further referred to as branches, and
- the apparatus for branch synchronization.

The expressions on arrays (vectors, matrices) generally employ the same operations as scalar expressions. Usually subarrays of various kinds and operations on them are specified. A logical conditional statement permits only those operations with the elements of the arrays to be performed which correspond to the TRUE value of the logical expression for conditional statement. Thus masking of the operations with vectors is performed which is generally done by hardware.

Special statements for branch description and initialization are introduced for parallel branch operations. In any point of the program one or several branches may be initialized for parallel execution. Usually static and dynamic definitions of parallel branches are given. In the latter case the number of branches in a given point is dynamically obtained in the course of the program execution.

The body of a branch is specified statically i.e. in the process of translation and, generally, may not be formed dynamically. The begining and end of each branch are specified by special statements.

The synchronization of the computing process was attained through the introduction of variables or arrays of the "event" type and wait and event termination statements. The operands for these statements were the variables or arrays of the "event" type. To provide processing of the same data array by several branches statements of the "semaphore" type were introduced. To raise the level of synchronizing primitives the mechanisms of the conditional critical intervals [4], monitors [2,3,5], sentinels [6], control expressions [1] and rendezvous (in the Ada language) were created. One should note that the above rather complicated synchronization designs may be expressed through the semaphores as well, but the use of these designs increases software "reliability" and lowers the probability of errors in the program when complex interactions are described.

2. Translation means for parallelization

Program analysis and its parallelization in the course of the translation may be carried out as follows.

At the first step linear sections and simple cycles are isolated.

A linear section of the program is a part of it whose statements are executed in the natural order sequentially or in the order determined by unconditional branch instructions. A linear section is limited by the start and end statements.

A start statement of the linear section is the statement for which at least one of the following requirements is satisfied: either it has more than one direct predecessor or its direct predecessor has more than one direct follower.

Using these definitions one may easily construct the linear section search algorithm.

A simple cycle is a fragment of the program consisting of one or several cycle and cycle body statements not containing transfer-of-control beyond the cycle boundaries. These boundaries are found by formed indicators of cycle description used in the corresponding programming languages.

This terminates the process of the analysis of a program graph.

Parallelization inside linear sectors, design of ordered linear sequences (OLS) and parallelization of simple cycles may be executed in parallel.

2.1. Parallelization of linear sections

Parallelization inside a linear section is carried out by statements; inside the statements parallel execution of arithmetic expressions is possible.

The variables processed by the statements P_i of the linear section may be categorized in four groups:
1. Read-only denoted as W_i; 2. Write-only denoted as X_i; 3. Write-after-read denoted as Y_i; and 4. Read-after-write denoted as Z_i. When two statements P_1 and P_2 work in parallel memory cells or, which in this case is the same, identifies of the variables read by statement P_1 should not be affected by writing into them statement P_2. Thus $(W_1 \cup Y_1 \cup Z_1) \cap (X_2 \cup Y_2 \cup Z_2) = \phi$.
Changing P_1 and P_2 symmetrically we obtain

$$(X_1 \cup Y_1 \cup Z_1) \cap (W_2 \cup Y_2 \cup Z_2) = \phi$$

If I_i denotes input data (i.e. the variables and constants of the righthand part of the i-th asignment statement) and O_i, output data (i.e. the variables in the lefthand part of the i-th asignment statement) the above requirements of informational independence of statement i, j may be written as

$$I_j \cap O_i = \phi,\ I_i \cap O_j = \phi,\ O_i \cap O_j = \phi$$

Proceeding from these necessary conditions various algorithms for parallelization of linear sections were suggested. The transformation of scalar arithmetic expressions for parallel computation is in reducing the number of steps necessary to compute the arithmetic expression. For example the computation of the expression $a + b * c + d$ requires two steps. At the first step $(b * c)$ and $(a + d)$ are computed. At the second step the results obtained at the first step are summed up.

This parallelization of arithmetic expressions and parallel execution of information-uncorrelated linear section statements is possible in the cases when the computer system permits realization of pipeline processing and/or is provided with special arithmetical-logical devices for tracking, multiplication, shifting etc. Thus the computing process may be significantly speeded up.

2.2. Parallelization of cycles in the process of translation

For computer systems employing vector registers or sets of processor elements vector computations prove highly effective. The operations with the elements of vectors in such computations are performed one order faster than the same operations with scalars. Consequently the transformation of cycle bodies of sequential programs into vector operations may make the program execution essentially faster.

In the course of transformation of the cycle body of a sequential program into a vector operation the latter should be executed (resulting in parallel computations) on those elements of the vector the coordinates of all points of which are parallel to some plane. For instance, such that the condition $\Sigma a_i I_i =$ const holds. The value of the constant should change after each execution of the cycle body until all points of the cycle are not looked through.

Lending themselves for parallelization are normally not all the cycles but only those which satisfy some restrictions. Usually the following restrictions are imposed upon the body of the cycle:

A. It should not contain any input/output statements.

B. It should not contain any transfer-of-control outside the cycle.

C. It should not contain any references to the subprograms and functions whose parameters are generated variables.

D. It should satisfy certain restrictions on the form and order of index expressions.

The structure of the computer system greatly effects the cycle parallelization technique. Thus for systems of the ILLIAK-IV type one may employ the reference technique [8], for systems with a set of asynchronously operating processors, the hyperplane technique [8] or the method of parallelipipeds [14] etc. These methods differ both in the technique of parallelization and in the strength of restictions imposed upon the cycles to be transformed.

In a parallel cycle body execution one should determine the range of feasible values for each index variable in which the vector operation may be executed. In doing so one should provide equivalency of the vector operation to the initial cycle. The solution to this problem is generally that of system of integer equations and unequalities [9]. Therefore the parameters the cycles to be transformed should be specified in terms of constants rather than variables. Then the entire preparation to parallelization should be carried out in the process translation rather than in the course of the program's execution.

It should be noted that the analysis of cycles used in FORTRAN programs has shown that depending on the structure of the computer system and, consequently, on the parallelization technique used, 30 to 60 percent of cycles in these programs lent themselves to automatic parallelization.

3. Determination of branches in programs

A way to reduce program execution time is parallelization of it in branches i.e. revealing such sections of it which may be executed simultaneously and independently. Program branch initialization takes a great deal of time therefore rational branching implies that individual branches be executed long enough. When greater program units than linear sections or simple cycles are analyzed for possible parallelization, the initial program is presented in the form of linear structures consisting of single-input-single-output nodes. In the