

DISPROS - A Distributed Blackboard Architecture

By

Manoj Kumar Saxena

*A thesis submitted in partial
fulfillment of the requirements
for the degree of*

DOCTOR OF PHILOSOPHY

Department of Computer Science and Engineering

INDIAN INSTITUTE OF TECHNOLOGY

NEW DELHI - 110016

INDIA

JUNE 1991

Certificate

This is to certify that the thesis entitled **DISPROS - A Distributed Blackboard Architecture** submitted by Mr. Manoj Kumar Saxena to the Department of Computer Science and Engineering, Indian Institute of Technology, Delhi for the award of the degree of Doctor of Philosophy is a record of the original bonafide research work carried out by him under our guidance and supervision.

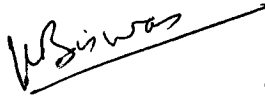
The thesis or any part thereof has not been submitted to any other university/institution for award of any degree or diploma.



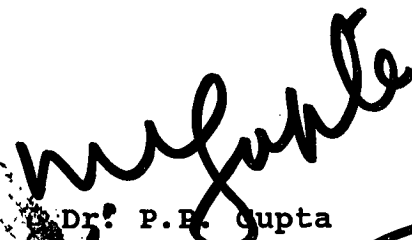
Prof. P.C.R. Bhatt

Dept. of Computer Science and Engineering

I.I.T. Delhi



Prof. K.K. Biswas



Dr. P.B. Gupta

CMC Ltd.

Acknowledgements

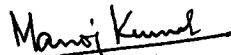
I would first like to thank my supervisors Prof. P.C.P Bhatt, Prof. K.K. Biswas and Dr. P.P. Gupta for their guidance, support and encouragement. This work could not have been accomplished without their assistance. Prof. Bhatt brought the subject to my attention and always made himself available for discussions and advice. Many of the ideas presented in this thesis originated during these discussions and have been benefited from his comments. Prof. Biswas has been an excellent sounding board for ideas. He greatly helped me in keeping the thesis on solid ground by questioning the assumptions and giving sound ideas for improvement. The thesis could not have been in the present form without his help. Dr. Gupta has been a constant source of inspiration and encouragement throughout the period of this work.

I am also indebted to Prof. S. N. Maheshwari, Dr. Sashi Kumar and Dr. S. K. Gupta for useful discussions and advice. Mr. N. C. Kalra and staff members of the computer laboratory deserve a special mention for their assistance and cooperation in porting ISIS to the local area network in the department.

I am grateful to my friends and colleagues for their patience, cooperation and encouragement. My special thanks are due to Mr A. K. Gangal without whose assistance and time, it would have been difficult to implement Field Engineering Planner application.

I would also like to express my gratitude to Prof. K. P. Birman for useful discussions on virtual synchronous environment and ISIS toolkit. Finally, I would like to thank my parents and Deepika, my wife for their immeasurable love, patience, encouragement and support for my educational endeavours.

Indian Institute of Technology
June 1991


Manoj Kumar Saxena

Preface

In recent past, there has been an upsurge of interest in Distributed Artificial Intelligence (DAI), with a recognition that the core problems are quite profound and impact many areas interesting to others both in Artificial Intelligence (AI) and its related disciplines. DAI provides a means for interconnecting multiple expert systems that have different but possibly overlapping expertise to enable the solution of problems whose domains are outside that of any one expert system. DAI is the most appropriate solution when the problem itself is inherently distributed, such as in distributed sensors net and specialized medical-diagnosis systems. The approach potentially has the additional advantages of modularity, speed, reliability, knowledge acquisition, and reusability.

When we started our work in 1987 there were few paradigms or models for building distributed intelligent applications. The principal paradigms that had been employed derived from *expert systems*, *blackboard systems* and *object-oriented systems*. Each of these approaches to system development consisted of an underlying computational model, a method for organising knowledge and other computational elements and a control regime for determining the sequence of operations. We felt a need to combine features of these paradigms into a unified paradigm for distributed problem solving, which would be *highly modular, flexible and support diverse knowledge representation and reasoning mechanisms*.

In this thesis we describe our research efforts to meet the above mentioned objectives and the proposed **DISPROS - DIStributed PROblem Solver** architecture, which is the result of those efforts. We have described the concepts and terminology used in this thesis in Glossary at the end of the thesis. A list of abbreviations used in the thesis is also included.

Abstract

At the current level of evolution, the computers are increasingly being employed to solve complex problems and exhibit intelligent behaviour. Lately, the blackboard architecture and its variations [Nii 86a, 86b, Engelmores 88] are being explored to achieve efficient and reliable problem solving paradigms. It has been viewed as particularly appropriate for parallel and distributed hardware architectures, as blackboard metaphor entails multiple, independent sources of knowledge trying to solve problem collectively. Yet, most of the blackboard systems have been implemented on conventional hardware and are primarily executed serially. So far very few attempts have been made to exploit the concurrency. This thesis describes our efforts to exploit the inherent concurrency offered by blackboard architecture, by distributing it over a network of heterogeneous computers.

To achieve any degree of success in supporting complex, distributed knowledge based problem solving, the system must address the issues of expressive adequacy and notational efficacy of the knowledge representation scheme. The system should be able to reason with incomplete and not necessarily consistent knowledge. The control mechanism has to be intelligent enough to achieve coherence and consistency among the concurrent problem solving activities of various experts, with distribution of problem solving over a network, there is an additional need to have fault-tolerance mechanism to withstand various faults. It is also desirable to make the system modular, flexible and widely distributable.

To address these issues/ requirements, we have proposed **DISPROS - DISTRibuted PROblem Solver**, as a framework for the implementation of distributed problem solving systems in loosely coupled, multi computer environment. DISPROS allows a knowledge based system to be described as a collection of cooperating, distributed, intelligent, encapsulated agents called Level Managers (LMs). Each LM is based on blackboard model and is an active structure consisting of the knowledge base, the behaviour part, the local controller, the monitor and communication controller. LMs execute entirely

asynchronously, in parallel and communication by passing messages. In this thesis, we describe the underlying computational and control models to support the problem solving using DISPROS, and the strategy adopted to withstand various system and communication related faults.

DKRL - **D**istributed **K**nowledge **R**epresentation **L**anguage, the knowledge representation paradigm developed for DISPROS considers LM as an object, and allows object-oriented programming techniques to be used for application development. We have proposed **DIAM** - **D**istributed **I**ntelligent **A**gent **M**ethodology, to develop distributed problem solving using DISPROS. It allows a problem solving system to be specified as a collection of intelligent agents. These agents can be directly mapped to LMs in DISPROS and implemented using DKRL.

To demonstrate the applicability of the proposed paradigm two systems were developed. The ASSIGN [Saxena 90b] system was developed to illustrate the expressive power of DKRL and local control component of the LM. The FEP - Field Engineering Planner, system has been developed to establish the viability of collection of LMs as a distributed problem solver. It also demonstrated the global control mechanism and the viability of message passing as a means of communication and cooperation amongst LMs.

Table of Contents

Acknowledgements	i
Preface	ii
Abstract	iii
Chapter 1. Introduction	1
1.1. Introduction	1
1.2. Research Issues	3
1.3. Related Work	6
1.4. The Proposed Architecture	10
1.5. Preview to Contributions	11
1.6. Thesis Outline	12
Chapter 2. Blackboard Systems	14
2.1. Blackboard Systems	14
2.1.1. Blackboard Metaphor	15
2.1.2. The Implementation	16
2.1.3. The Model and Concurrency	18
2.2. Distributed Problem Solving with Blackboard	20
2.2.1. Design Alternatives	20
2.2.2. Fault-Tolerance	23
Chapter 3. DISPROS Architecture	26
3.1. Structure of DISPROS	26
3.1.1. The Level Manager	27
3.1.2. The Communication Level Manager	30
3.1.3. The Recovery Manager	31
3.2. Organisational Structure and Communication	31
3.3. Messages	33

3.4. Control Model of DISPROS	34
3.4.1. Local Control	35
3.4.2. Global Control	37
3.5. Computational Model of DISPROS	39
3.6. Concurrency in DISPROS	40
3.7. Fault-tolerance in DISPROS	43
3.8. Review of DISPROS	48
Chapter 4. Knowledge Representation	51
4.1. Knowledge Representation Requirements for Distributed Problem Solving	51
4.1.1. Knowledge Representation Approaches	53
4.2. Knowledge Modelling	54
4.3. Knowledge Representation Scheme Design	57
4.4. DKRL - Distributed Knowledge Representation Language	60
4.5. KDL - Knowledge Definition Language	63
4.6. KML - Knowledge Manipulation Language	68
4.7. HLI - Host Language Interface	71
4.8. Semantics of DKRL	73
4.9. An Example of Knowledge Representation with DKRL	74
4.10. ASSIGN - A Case Study	76
Chapter 5. Software Engineering Aspects of DISPROS	83
5.1. Implementation of DISPROS	84
5.1.1. DKRL Interpreter	84
5.1.2. Local Controller	88
5.1.3. User Interface	91
5.1.4. Communication System	92

5.2. ISIS Toolkit and its Integration	94
5.2.1. ISIS Overview	94
5.2.2. Virtual Synchronous Environment	96
5.2.3. Modification and Integration of ISIS with DISPROS	97
5.3. DIAM - An Application Development Methodology for DISPROS	99
Chapter 6. Field Engineering Planner - A Case Study	103
6.1. The Application Domain	103
6.2. Design of Field Engineering Planner	104
6.2.1. Organisational Hierarchy	104
6.2.2. The Agents	106
6.2.3. Interaction Among Agents	106
6.2.4. Hierarchy Relationship Among Agents	107
6.3. Implementation of FEP	107
6.3.1. Knowledge Sources	108
6.3.2. Knowledge Bases	110
6.3.3. Local Control	113
6.3.4. User Interface	114
6.3.5. Fault-Tolerance	115
6.4. Problem Solving Process in FEP	116
6.5. Conclusion	117
Chapter 7. Conclusion	120
7.1. DISPROS Vis-a-Vis other Recent Approaches	120
7.2. Our Contributions	123
7.3. Direction for Further Work	125
Appendix A. Syntax of DKRL	127
A.1. Knowledge Definition Language (KDL)	127
A.2. Knowledge Manipulation Language (KML)	131

Appendix B. Semantics of DKRL	134
B.1. Instruction Set of Abstract Machine	134
B.2. Semantic Domains	137
B.3. Knowledge Definition Language (KDL)	139
B.4. Knowledge Manipulation Language (KML)	141
Appendix C. FEP Knowledge Sources and Traces	147
C.1. Knowledge Sources	147
C.2. Application Traces	156
Bibliography	164
Glossary	175
Abbreviations	178