

FORMALIZATION OF CONCURRENCY
CONTROL IN DISTRIBUTED
DATA SYSTEMS

Serge M. MIRANDA.

BIBLIOTHEQUE DU CERIST

IST
1139

FORMALIZATION OF CONCURRENCY CONTROL IN DISTRIBUTED

DATA SYSTEMS (1)

Serge M. MIRANDA.

CERISS-INRIA.

Université des Sciences Sociales.

Place Anatole France.

31040 TOULOUSE Cedex.

FRANCE.

Tel (61) 23 11 45 ext 395



A_B_S_T_R_A_C_T

Synchronization issues in distributed data bases were heavily invested in recent literature and many synchronization protocols have been designed.

However much work is still to be done in the areas of ROBUSTNESS and RECOVERY, FORMAL PERFORMANCE ANALYSIS, FORMAL SPECIFICATION AND VALIDATION, RIGOROUS UNIFORMIZATION.

This paper is a contribution to the two latter points; we propose a formal approach based on abstract data types (algebraic methodology) and develop a uniform rigorous framework in which the synchronisation protocols can be specified and validated.

We illustrate our approach on a basic protocol which is representative of a major class of solutions.

KEY-WORDS : protocol formalization (specification and validation), synchronization protocol, distributed data bases, duplicated data, abstract data types.

(1) This research is sponsored by INRIA-ADI (SIRIUS project) under contract # 80003.

C O N T E N T S

1.	Introduction.	P. 1
2.	Integrity concepts.	P. 1
2.1.	Integrity in a centralized DBMS.	P. 1
2.2.	Integrity in a distributed DBMS.	P. 2
3.	Abstract data types.	P. 4
3.1.	Abstract data types concept.	P. 4
3.2.	Algebraic approach.	P. 4
3.3.	OBJ-0.	P. 6
4.	Formalization of a synchronization protocol.	P. 7
4.1.	Global entity states.	P. 8
4.2.	Canonical reduction; serialization.	P. 8
4.3.	Mutual-integrity theorems.	P. 9
5.	Formalization of Thomas' protocol	P. 10
5.1.	Principle of THOMAS' solution.	P. 10
5.2.	THOMAS' protocol formalization.	P. 12
6.	Conclusion.	P. 15
7.	References.	P. 17
ANNEX 1.	Specifications of THOMAS' protocol	P. 22
ANNEX 2.	Analysis of commutations for THOMAS' protocol in a concurrency-conflict environment	P. 25

1. INTRODUCTION

Many researchers have recently presented solutions to concurrency control for distributed data systems. These solutions take form of synchronization protocols which have been designed to coordinate the remote processes in charge of local data (called "controllers") during an update session.

Very few proposals have been made to formally specify and validate ("formalize") these protocols.

This paper is primarily concerned with this crucial aspect; we introduce a formal methodology based on algebraically-specified data types to formalize existing protocols.

This article encompasses two major sections :

- the first one presents a clear definition of mutual integrity which turns out to be the basic requirement which must be verified by a synchronization protocol and introduces our formalism.

This concept is translated in terms of our model through mutual-integrity theorems which are recalled.

- the second section illustrates our approach with a synchronization protocols (for duplicated entities) which has been largely referenced in the literature (namely THOMAS' one).

2. INTEGRITY CONCEPTS

We shall in turn examine the integrity concept in centralized and distributed data base management systems (DBMS).

2.1. INTEGRITY IN A CENTRALIZED DBMS

A data base can be viewed as a collection of entities and constraints whose values define VALID states of the data base. The concept of integrity (consistency) of a DBMS is twofold (MIRA80-b) :

- internal integrity.
- external integrity.

Internal integrity is associated with integrity constraints defined on data to meet real-world restrictions. The concept of transaction (GRAY78) has been introduced in centralized DBMS (and naturally extended to distributed DBMS (GRAY79)) to represent the atomic interaction of the user with the data base which preserves internal integrity.