

/THE IMPLEMENTATION OF  
A SUBSET DATA DICTIONARY VERIFIER/

12

by

JACQUELYN FERN CLINE

B.S., Central State University, Edmond, Oklahoma, 1974

---

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY  
Manhattan, Kansas

1985

Approved by:



Dr. David A. Gustafson

LD  
2668  
.R4  
1985  
C52  
C.2  
CHPT

TABLE OF CONTENTS

A11202 996185

CHPT	TITLE	PAGE
1	The Role of Data Dictionaries and a Verification Implementation	1
	1.0 Introduction	
	1.1 Software design crisis	
	1.2 Data definition and management	
	1.3 The concept behind data dictionaries	
	1.4 The role of data dictionary systems	
	1.5 The role of data dictionaries in software development	
	1.6 The benefits of data dictionaries	
	1.7 The role of the Subset Data Dictionary Verifier	
	1.8 The benefits of the Subset Data Dictionary Verifier	
2	The Subset Data Dictionary Verifier Requirements	13
3	The Subset Data Dictionary Verifier Design	17
4	The Subset Data Dictionary Verifier Implementation	22
5	Conclusions and Extensions	27
	References	29
	Appendices	
	Detailed Specifications in BNF format	30
	User's Manual	34
	Source Code	35
	Detailed Module Specifications	56

**THIS BOOK  
CONTAINS  
NUMEROUS PAGES  
WITH DIAGRAMS  
THAT ARE CROOKED  
COMPARED TO THE  
REST OF THE  
INFORMATION ON  
THE PAGE.**

**THIS IS AS  
RECEIVED FROM  
CUSTOMER.**

## CHAPTER 1

### THE ROLE OF DATA DICTIONARIES AND A VERIFICATION IMPLEMENTATION

#### 1. Introduction

This paper introduces the concepts of data dictionaries and a verification process called a Subset Data Dictionary Verifier. The Subset Data Dictionary Verifier is developed to verify the frames in a data dictionary in relation to a master data dictionary. Both dictionaries used in the verification process are derived from the Entity Relationship Attribute (ERA) Specification. The verifier will show the discrepancies between the subset data dictionary and the master data dictionary.

#### 1.1 Software Design Crisis

It has been suggested that we are in a software design crisis resulting from cheaper hardware technology and increasing software design costs. "The crisis is created by the labor-intensive nature of design methods..." (1) The hardware costs are decreasing rapidly; in fact, those costs are being cut to the bone. However, hardware only constitutes approximately 30% of the total cost of a system. The software costs, likewise, need to be cut to a minimal figure. No longer can we continue to support "made to order" applications. Technology is continuing to strive towards better firmware and more generalized hardware. Software, on the other hand, is continuing to fall farther and farther behind.

This crisis is ameliorated if the computer technology implements the philosophies and structures of: automated programming and design aids, such as data dictionaries and data base management systems; direct access to computers by end users, such as user-friendly and user-maintainable query systems; and project management techniques and tools for software development projects.

We are quickly approaching the need for knowledge information processing systems (KIPS) using data inference and artificial intelligence. The Japanese are in a fast paced race to complete such a task by 1991. "That means all the vigorous hand cranking, throttling, and wrenching a pioneer now accepts as the inevitable price of using the machine -- the difficult programming languages, the struggle to make different programs compatible, the problems of putting human knowledge into machine form -- are to disappear, eliminated in the new Japanese Fifth Generation of computers." (2)

Each analyst designs systems individually with little regard to the way other analysts design systems; programmers code programs as individually as their personalities and background allow. Technology has been rapidly changing and now we must draw these concepts together into a unified assembly. Company guidelines are issued to analysts and designers to make sure a development effort is flexible, user-friendly, and user-maintainable. Programmers are being given strict guidelines to insure programming products are cohesive, consistent, and understandable. "We have made great gains in the consistency and rigorousness with which we apply these