20-

A TOOL FOR USE IN ON-LINE DATA DICTIONARY CREATION

by

CHINHO LAI

B. S., Soochow University, 1975

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

1983

Approved by:

S. Hunger

Major Professor

TABLE OF CONTENTS

LIST OF FIGURES PAGE

ACKNOWLEDGEMENTS

I would like to thank Dr. Elizabeth Unger for her constant guidance, suggestions, and tireless editing. I would like to thank Drs. David Gustafson and Rod Bates for serving as committee members. Thanks also to my fiancee, Esther, for her enccouragement.

CHAPTER 1. INTRODUCTION

As the eighties are upon us, one fact stands out clearly: past practices and procedures in the data processing (DP) field are inadequate to meet the challenges of the future. There is a need to increase the productivity of an enterprise, to stabilize software costs, and to plan and manage the most critical resource of the eighties, the information resource, in a more efffective fashion. In this report, we address a suggested partial solution to these needs, the data dictionary. The focus is the creation of a dictionary of data about data, i.e., metadata, by the ultimate user of the data.

1.1 Problems

Conventional data definition has been inadequate. The manner in which data has been defined in most of today's application systems has been a function of (1) the programming language used, (2) the characteristics of the programming language used, and (3) the habits of the individual programmers who have written the programs. Some of today's programming languages have the characteristics that data definitions can be embedded in the executable statements. If data definitions are only used in one program, no data administration problem is raised. An advance toward disciplined programming occurred when the data division was defined in the COBOL language. In this division, all of the data definitions used by a program

are stated explicitly. This made the task of retrieving all data definitions much easier than in languages which allow definitions to be scattered throughout the programs. COBOL did not fully solve the problem of controlled data definitons. If the same data file in used by five programs, each program would include the data definitons for that file. If changes are made in the data definitons for file, it is necessary that all programs that use the file be identified and their data divisions changed.

Many user organizations have ended up with multiple data files in which the same data item has been defined differently and or with a different name, different field length, different updating cycle, etc.(semantic redundancy). Additionally, different data items have ended up with the same name (syntactic redundancy). Sometimes the same data item has been defined "well" in certain instances and "poorly" in others, to the point where the same data item in two files cannot compared. Other difficulties from conflicting data definition have included (1) the need for special programs to convert 'division' data to conform with the 'corporate' data definitions, (2) loss of data integrity, due to mistakes because of the multiple definitions of the same data item, (3) confusion in interpretation, with consequent mistakes in decision making, due to the multiple definitions, (4) loss of opportunity for using the data for additional purposes, because