

3-2

A DATAFLOW DIAGRAM GENERATOR

by

ALICIA ELLEN SPECHT

B. A., University of Georgia, 1977

---

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

1986

Approved by:

  
Major Professor

LD  
2668  
.R4  
1986  
.563  
C.2

CONTENTS

1.	Literature Survey .....	1
1.1	Introduction .....	1
1.2	The Need for Requirements Engineering .....	2
1.3	Characteristics of a Good Requirement Specification .....	4
1.4	Formal Requirement Specifications .....	8
1.5	Types of Requirements Methodologies .....	11
1.6	Dataflow Models .....	14
1.6.1	Dataflow Diagrams .....	14
1.6.2	SADT .....	15
1.6.3	PSL/PSA .....	16
1.7	Future Trends in Requirements Methodologies ...	18
1.8	Conclusion .....	20
2.	Requirements .....	22
2.1	General .....	22
2.2	Input .....	22
2.3	Output .....	23
3.	Design .....	26
3.1	Introduction .....	26
3.2	Overview of System Structure .....	26
3.3	Design of the Dataflow Diagram .....	27
3.4	Detailed Design .....	31
3.4.1	Parse Requirement Spec Module .....	33
3.4.2	Compare Inputs Outputs Module .....	35
3.4.3	Column Marking Module .....	38
3.4.4	Row Marking Module .....	42
3.4.5	Row Spacing Module .....	44
3.4.6	Create Graph Module .....	45
3.5	Implementation and Testing .....	52
4.	Conclusions and Extensions .....	54
4.1	Conclusions .....	54
4.2	Extensions .....	54
	References .....	56
	Appendix A - BNF Description of ERA Specification .....	59
	Appendix B - Module Specifications .....	62
	Appendix C - Error Messages .....	72
	Appendix D - User Guide .....	76
	Appendix E - C Language Code for DFD Generator .....	77
	Appendix F - Sample Output from DFD Generator .....	111

## LIST OF FIGURES

Figure 1.	Format of dataflow diagram .....	24
Figure 2.	Dataflow diagram design .....	29
Figure 3.	Hierarchy diagram of main modules .....	32
Figure 4.	Parse_requirements_specs .....	34
Figure 5.	Compare_inputs_outputs dataflow data structure .....	37
Figure 6.	Column_marking .....	39
Figure 7.	Row_marking - column 1 .....	43
Figure 8.	Row_marking - columns 2 through n .....	44
Figure 9.	Dataflow diagram design .....	47
Figure 10.	Reserved lines data structure .....	50

## Chapter 1 - Introduction

### 1.1 Overview

This report describes a tool that produces a dataflow diagram from a requirements specification in textual form. This project is one tool in a prototype system for an advanced software development environment. The dataflow diagram produced by this tool should be useful in the requirements analysis phase of the software development cycle.

The need for adequate requirements analysis in the beginning of the software development process has received considerable attention in recent years [YE82] [R085a] [R077b] [TE77]. This attention has arisen in response to the growing concern of software developers over the inability to develop software systems in a predictable time frame that accurately perform the intended functions of the system. It has been noted that problems in system development have become less traceable to the hardware or the programming [R077b], inferring that the problems resulted from inaccurate requirement specifications.

This paper surveys the literature on requirements engineering with particular emphasis on dataflow analysis as a tool in requirements analysis. This paper discusses:

1) the need for adequate requirements engineering 2) the characteristics of a good requirements specification 3) the benefits of formal requirements specification languages 4) a survey of existing types of requirements modeling methodologies 5) a description of specific dataflow model requirements specification techniques and 6) future goals for requirements specification methodologies.

### 1.2 The Need for Requirements Engineering

Requirements engineering is an essential step in the software development cycle. Requirements engineering is the iterative process of analyzing the problem to be solved by the proposed software system, documenting the resulting requirements, and checking the documents for accuracy [RZ85]. The resulting requirement specification is then used as a guideline in the software design process. A requirement specification states the intended functional and performance attributes of the system without specifying an implementation plan. Without an accurate requirements specification, the software designer may solve the wrong problem, resulting in a system that does not satisfy the user's needs [RO85a].

Discrepancies between the system requirements and its implementation can result in significant cost overruns and