Automated Collision Diagram Production

WA-RD 180.1

Final Report
June 1989

Washington State Department of Transportation
Planning, Research and Public Transportation Division

in cooperation with the
United States Department of Transportation
Federal Highway Administration

Final Report

Research Project GC 8286, Task 11
Collision Diagram Automation

AUTOMATED COLLISION DIAGRAM PRODUCTION

by

Timothy L. Nyerges
Assistant Professor
Department of Geography
University of Washington

Ron F. Cihon
Mapping Supervisor
Washington State Department
of Transportation
and
Department of Geography
University of Washington

Washington State Transportation Center (TRAC)
University of Washington
The Corbet Building, Suite 204
4507 University Way N.E.
Seattle, Washington 98105

Washington State Department of Transportation
Technical Monitor
Paul Moody
Manager, Geographic Services

Prepared for

Washington State Transportation Commission
Department of Transportation
and in cooperation with
U.S. Department of Transportation
Federal Highway Administration

June 14, 1989

WASHINGTON STATE DEPARTMENT OF TRANSPORTATION

TECHNICAL REPORT STANDARD TITLE PAGE

1. REPORT NO.	2. GOVERNMENT ACCESSION NO.	3. RECIPIENT'S CATALOG NO.
WA-RD 180.1		

4. TITLE AND SUBTITLE	5. REPORT DATE
AUTOMATED COLLISION DIAGRAM PRODUCTION	June 1989
	6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S)	8. PERFORMING ORGANIZATION REPORT NO.
Timothy L. Nyerges and Ron F. Cihon	
9. PERFORMING ORGANIZATION NAME AND ADDRESS	10. WORK UNIT NO.
Washington State Transportation Center (TRAC) University of Washington, JE-10 The Corbet Building, Suite 204; 4507 University Way N.E. Seattle, Washington 98105	11. CONTRACT OR GRANT NO. GC8286, Task 11
12. SPONSORING AGENCY NAME AND ADDRESS	13. TYPE OF REPORT AND PERIOD COVERED
Washington State Department of Transportation Transportation Building, KF-10 Olympia, Washington 98504	Final Report
	14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES

This study was conducted in cooperation with the U.S. Department of Transportation, Federal Highway Administration.

16. ABSTRACT

Collision diagrams are graphic representations of intersections, ramps, or sections of roadway that provide detailed information about accidents at a specific location. The main purpose of collision digrams is to serve as depictions of accident patterns, and they are used to evaluate specific sites for possible causes of accidents. Currently, the Washington State Department of Transportation manually constructs these diagrams. However, this study found collision diagrams could be produced with automated technology without changing the data structure of WSDOT's accident data files.

This study examined literature and existing software that other transportation agencies or vendors use to construct collision diagrams. This examination determined that computer generated diagrams created with that software lack graphic detail of individual intersections, are limited to a specific data structure, and are limited to two or three generic intersection depictions. Several alternative means to construct enhanced diagrams were examined, including artificial intelligence (AI) techniques, to find a way to offer a more robust assortment of graphics.

A prototype system for producing automated collision diagrams was developed. This Washington Automated Collision Diagramming System (WACDS) has the following advantages over previous systems: (1) WACDS integrates the technologies of computer-aided drafting (CAD) and database management to provide improved graphic detail and ease of utility. (2) WACDS uses WSDOT's current accident file data structures. (3) WACDS is modular and can be modified, expanded, or shared without major software revision.

The report discusses a strategy for fully implementing WACDS at WSDOT, in addition to a schedule for implementation.

17. KEY WORDS	18. DISTRIBUTION STATEMENT
collision diagrams, computer-aided drafting (CAD), computer-assisted mapping	No restrictions. This document is available to the public through the National Technical Information Service, Springfield, VA 22616

19. SECURITY CLASSIF. (of this report)	20. SECURITY CLASSIF. (of this page)	21. NO. OF PAGES	22. PRICE
None	None	60	

DOT FORM 310-022 (11/86) -1216-

DISCLAIMER

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the Washington State Transportation Commission, Department of Transportation, or the Federal Highway Administration. This report does not constitute a standard, specification, or regulation.

TABLE OF CONTENTS

Section	Page
Study Summary	**ix**
Conclusions and Recommendations	**xi**
Introduction	**1**
Background	1
Problem Statement	3
Research Approach	4
Review of Existing Systems	**7**
General Literature Review	7
Review of Acquired Software	9
SCCOLD -- Small Computer Collision Diagram	9
ACCIDIAGRAM	10
Oregon Department of Transportation (ODOT) ACDS	13
Overview	13
Strengths and Limitations of the ACDS	16
Summary Statements	18
Conventional Software	18
Knowledge-based Approach	19
Prototype WACDS	**23**
Goals of WACDS	23
System Overview	24
Mainframe Connection	25
PC-DOS Environment -- Filebuilder	25
Selection of the Base File	28
Interpretation of Accident, .TXT File Construction	33
PC CADD Environment	34
Accident Symbol Library	34
Plotting Environment	35
Implementation	**37**
Developing a Production Level System	37
Implementation	39
Future Enhancements	41
Data Capture	41
Data Presentation	43
Data Integration	44
References	**45**
Acknowledgments	**47**

LIST OF EXHIBITS

Exhibit		Page
1.	Collision Diagram Example from SCCOLD Showing All Types of Accident Representations	11
2.	Collision Diagram Produced by the ACCIDIAGRAM System	12
3.	The Oregon ACDS System Overview	15
4.	Collision Diagram Example Produced by ACDS	17
5.	The WACDS System Overview	26
6.	Flowchart Depicting Processes in Filebuilder	27
7.	Impact Location Positions for a Simple Cross-type Intersection	29
8.	Rotation and Coordinate Values for Impact Location of Cross-type Intersection	30
9.	Accident Placement Locations for Cross-type Intersection	31
10.	Graphic Depiction of the Impact Position Library and Corresponding Type File	32
11.	Sample WACDS Collision Diagram	36
12.	Graphic Depicting Current and Proposed Enhancements of WACDS	42

STUDY SUMMARY

Collision diagrams are graphic representations of intersections, ramps, or sections of roadway that provide detailed information about accidents at a specific location. The main purpose of collision diagrams is to serve as depictions of accident patterns, and they are used to evaluate specific sites for possible causes of accidents. Currently, the Washington State Department of Transportation manually constructs these diagrams. However, this study found collision diagrams could be produced with automated technology without changing the data structure of WSDOT's accident data files.

This study examined literature and existing software that other transportation agencies or vendors use to construct collision diagrams. This examination determined that computer generated diagrams created with that software lack graphic detail of individual intersections, are limited to a specific data structure, and are limited to two or three generic intersection depictions. Several alternative means to construct enhanced diagrams were examined, including artificial intelligence (AI) techniques, to find a way to offer a more robust assortment of graphics.

A prototype system for producing automated collision diagrams was developed. This Washington Automated Collision Diagramming System (WACDS) has the following advantages over previous systems:

1. WACDS integrates the technologies of computer-aided drafting (CAD) and database management to provide improved graphic detail and ease of utility.
2. WACDS uses WSDOT's current accident file data structures.
3. WACDS is modular and can be modified, expanded, or shared without major software revision.

The report discusses a strategy for fully implementing WACDS at WSDOT, in addition to a schedule for implementation.

CONCLUSIONS AND RECOMMENDATIONS

This research had three major tasks:

1. investigate the literature, and contact individuals or agencies to assess the current state of constructing collision diagrams by automated methods,
2. investigate the possibility of applying AI techniques to assist in the construction of collision diagrams, and
3. develop a prototype system based on the findings of the above two tasks.

The original proposal suggested that a fully implemented production level system may be developed as an outcome of the research. This was based on the possibility that existing software could be adapted to accommodate the WSDOT coding scheme without modification. However, through a review of existing systems, the investigators determined that no software currently available for construction of collision diagrams could be adapted to work with the WSDOT accident codes within timely and economical constraints. A completely new set of programs need to be developed to interpret the WSDOT accident data. The WACDS prototype provides the groundwork for these programs.

The three broad tasks set forth in the proposal were successfully completed. Through an investigation of literature and systems that are currently in use, we have concluded that these systems do not have the ability to construct more than a few, generic intersection types. A system is needed which would portray significantly more detail of unique intersections.

In an attempt to meet this need we investigated the use of AI techniques to assist in the actual construction of collision diagrams. We have concluded that these new techniques may be useful only if a significant amount of additional research is undertaken to apply AI techniques to the specific problem of graphic presentation.

We feel more appropriate areas of application are in accident investigation, recording, and development of countermeasures.

A prototype has been developed that significantly enhances the presentation of intersections. WACDS uses individual graphic depictions that must be manually digitized and stored in "libraries." The system is based on concepts developed at Oregon DOT. WACDS uses conventional programming and off-the-shelf CADD software. The researchers suggest that, if implemented, WACDS can construct collision diagrams more efficiently than by manual methods. WACDS, because it is based on a coordinate system and uses standard software, can be adapted as technology changes. The data stored as graphic representations or impact position locations can be adapted by various systems, and integrated into a much larger safety analysis system.

INTRODUCTION

BACKGROUND

Several descriptions of computer graphic systems used for the display of transportation-related information, such as traffic safety and accidents, appear in the literature. These graphic systems tend to focus on small scale (1:500,000 to 1:24,000) base maps that offer a "system overview" generally preferred by planners and administrators. Similarly at the Washington State Department of Transportation (WSDOT), the Mapping and Display of Geographic (MADOG) Information System currently uses two map scales: 1:500,000 and 1:24,000 as a framework upon which statistical information will be presented. For large scale analysis of spatially related problems, such as accidents, these map scales are inappropriate. There presently is a need for the automated construction of large scale maps (referred to as Collision Diagrams) which are used primarily for engineering analysis.

The construction and use of large scale diagrams for the presentation and analysis of accidents is documented in WSDOT manuals. This assumes preparation by manual means, but does not restrict preparation to manual means. The construction of collision diagrams using automated techniques is less common than preparation by manual means in most Departments of Transportation because software is not readily available or lack of technical expertise limits the implementation of such software.

Portions of some transportation planning and engineering problems require a high degree of flexibility in processing data depending on the nature of the data or problem being solved. Yeh, Ritchie and Schneider (1) describe such problems in transportation planning and engineering as being those that might make use of knowledge-based expert system programming techniques. Such techniques allow for

easier handling of special considerations to certain situations in formulating the solution to a problem. This has often been called exception handling, where exceptions might take the form of certain decisions being made based on the mutual validity of two or more data values in the file. Yeh, Ritchie and Schneider identify traffic safety applications, e.g. roadway geometry analysis, as being one group of applications that might benefit from knowledge-based expert system techniques.

Most of the literature describing automated techniques for the preparation of collision diagrams appears in the middle and late 1970s. The lack of published reports in the 1980s is somewhat perplexing, considering the apparent usefulness of this graphic tool and the newly acclaimed efficiency of construction. In order to obtain additional information on the "state-of-the-art" in automating the construction of collision diagrams, an informal survey of known users of this type of system was conducted for this project. The survey consisted of correspondence with mostly State DOT individuals such as analysts, administrators, and cartographers concerning the usefulness, preparation, and effectiveness of these large scale graphics. The respondents were asked if they knew other agencies, consultants or individuals who have used or participated in the development of automated collision diagrams. This survey was undertaken to sample the "state-of-the-art" issues.

Although the survey was neither scientific nor complete, it did provide some insight into users' opinions and plans for future enhancements. The following comments summarize the status of automated collision diagram production:

1. Most systems were developed in the early and mid-1970s on large mainframe computers. The programming was usually done in FORTRAN or COBOL.
2. The programming was "system specific," and required that the data be collected and formatted in a particular manner.
3. At most agencies, data collection techniques and record formats changed periodically, rendering the initial programming for collision

diagrams useless. Money was not available to make the adjustments, so the system died before its utility ever became fully realized.

4. The cost of running the packages on the mainframe discouraged many users, and the systems were generally too cumbersome.
5. A few researchers are trying to rewrite some of the application programs to the PC level, feeling the number of users would increase. Some programs are starting to appear.
6. Most people thought collision diagrams produced by computer methods would be a useful tool in the identification of hazardous locations if the system was transferable and operated on the microcomputer level, having capabilities of handling large amounts of data inexpensively.
7. The system must be easy to operate in order to be utilized by the greatest number of personnel.
8. The system must have greater graphic capabilities.

This project attempts to address most of those issues in the automation of collision diagrams in a microcomputer environment.

PROBLEM STATEMENT

Collision diagrams are graphic representations of intersections, ramps or sections of roadway that provide detailed information about vehicular accidents at a specific location. Each diagram contains numerous graphic symbols that depict characteristics of each accident. They are used by individuals such as engineers, designers, and planners as a means of evaluating specific sites for the possible causes of accidents.

Presently at WSDOT, collision diagrams are manually produced. The process can be laborious and lengthy because data is extracted from files and then

compiled manually into graphics. Because of this, results are not always standard, and only a limited number of areas can be represented accurately.

There is a need for an automated means of producing these collision diagrams. The premise in this study is that automation of collision diagrams would provide a quicker, standardized and more widely used tool for the analysis of problem areas.

RESEARCH APPROACH

The research was divided into three tasks. The first task evaluated current conventional computer programming techniques and languages to automate the construction of collision diagrams. A literature review was conducted to identify automated collision diagramming systems applicable to WSDOT's needs. Inexpensive or public domain software was acquired from several sources to determine the suitability for implementation or modification.

The second task was intended to investigate which of those processes might best be enhanced/supported using knowledge-based expert systems techniques. This second task was intended as a functional enhancement of the basic system, and assumed that additional software functionality would be more important than a software redevelopment effort. Information for this second task was obtained by interviewing WSDOT personnel who have responsibility for creating and analyzing collision diagrams at the district level.

The third task involved the development of a prototype system for use at WSDOT, using conventional or knowledge-based approaches. This system, if adopted, is intended to automate and simplify the processes of

1. extracting data from the appropriate database,
2. preparing or retrieving large scale, intersection, digital base maps for "generic" and specific locations, and

3. providing accurate placement of symbology to represent characteristics of each accident according to WSDOT coding conventions.

This prototype system is intended only as a starting point for the development of a fully operational, automated diagramming system at WSDOT. Recommendations for system enhancement and the integration with other Departmental systems is included.

REVIEW OF EXISTING SYSTEMS

GENERAL LITERATURE REVIEW

Wandsnider (2) discusses the capabilities of the Geographic Road Network Data Base (GRNDB) used by New Mexico DOT. Using 1:500,000 scale base maps, they have plotted accident information for the entire state route system, and have used these maps to identify traffic safety problem areas in an efficient manner.

Moellering (3) describes the Automap system for the automated mapping of traffic crashes. This system uses a base map or "geographic background" having a scale of 1:62,500. Accident information is portrayed using graphic symbols according to one of the following subsets: subject matter, point, linear, areal, or any combination. Moellering agrees that one of the applications of this tool is "to help recognize spatial patterns." (p. 474)

At WSDOT, the Mapping and Display of Geographic (MADOG) Information System is being developed. This system currently uses two base map scales: 1:500,000 and 1:24,000. In the future, two additional scales will be incorporated. These small and medium scale basemaps will serve as a framework upon which statistical information will be presented. They will contribute to the analysis of spatially related problems, such as accidents.

The use of large scale graphics for the presentation of accident data has also been documented. Graphics used in large scale accident analysis are either spot maps or collision diagrams. As described in a National Safety Council Bulletin (4) the primary purpose of a spot map is "to aid in identifying high accident locations and areas." Baerwald (5) suggests that a manually produced spot map furnishes a quick visual index of the location of accidents by pin, pasted spots, or symbols on the map at the location of the occurrence. Although current documentation involving spot maps and collision diagrams are primarily concerned with the **manual**

preparation of these graphics, the need and utility of this graphic tool is clearly emphasized by Box (6).

Collision diagrams are graphic representations of intersections, ramps, or sections of roadway that provide detailed information about vehicular accidents at a specific location. Each diagram contains numerous symbols which depict attributes and characteristics of each accident. These are used by engineers, designers, and planners as a means of locating and evaluating specific sites for possible errors in channelization, signalling, sign placement, and so forth. At most state and local transportation related agencies, these diagrams are produced manually. The process is laborious and time-consuming, whereas traffic data are individually extracted from files, and then compiled into the appropriate graphics. Because of this, results are not always standard. Only a limited number of areas can be represented accurately and effectively, which leads to under-utilization of this important graphic tool.

Box and others describe step-by-step the collection of data, methods of presentation, and the myriad of applications of this large scale graphic. Baker (7) and Homburger (8) each illustrate the utility and importance of spot maps and collision diagrams in their handbooks on transportation engineering methodology. Homburger views the greatest asset of a collision diagram as its usefulness for intersection analysis once hazardous locations have been identified by some other means. He feels these maps should be used for analysis **within** a single intersection.

The construction of collision diagrams using automated methods is less common. Hagmann (9) has briefly outlined a program developed by the Oklahoma Department of Highways. These automated diagrams consist of a heading, outline of a typical intersection design, and at least one collision symbol. Two intersection templates are used. Coded symbols, representing the accidents, are automatically paced in any one of four quadrants. When accidents overlap, or groups become extremely close, presentations of a group of accidents at a single point is made by

adjusting a single graphic symbol to reflect this situation. Litvin and Datta (**10**) feel the presentation of a single symbol to represent multiple accidents at one location lacks visual impact. Therefore, they have developed the Automated Collision Diagram System (ACDS) to represent each accident by a single symbol. Their system is composed of two programs. The first performs a search of the accident file to extract appropriate records. The second, (a) checking data consistency, (b) plotting intersections or roadway segments, (c) drawing collision symbols, and (d) producing associated reports.

REVIEW OF ACQUIRED SOFTWARE

SCCOLD -- Small Computer Collision Diagram

SCCOLD was developed by the University of Florida Transportation Research Center in cooperation with the State of Florida Community Affairs, Bureau of Public Safety Management. It is readily available from McTrans at nominal cost. SCCOLD, written in BASIC, runs on any IBM-compatible with CGA graphics. The intent of SCCOLD is to present a graphics display which can easily be inserted into reports, of the general accident pattern at an intersection. The program was purchased and tested, in order to evaluate the data entry, data synthesis, and display techniques.

Upon execution of the SCCOLD.EXE program, a main data entry screen appears. This first screen contains a data entry box, accident description box, and an editing box. The user manually inserts codes for accident type and direction, and has the option of adding reference information (date, number, comments, etc.) for each accident. A maximum of 180 accidents can be manually inserted or read from existing files. Control of the data entry is accomplished by cursor movement from field to field. Once the desired number of accidents are entered, the user interacts with an options box that allows for the current data to be saved or plotted, and the addition of title, date, and other descriptive information. A "Plot" selection will

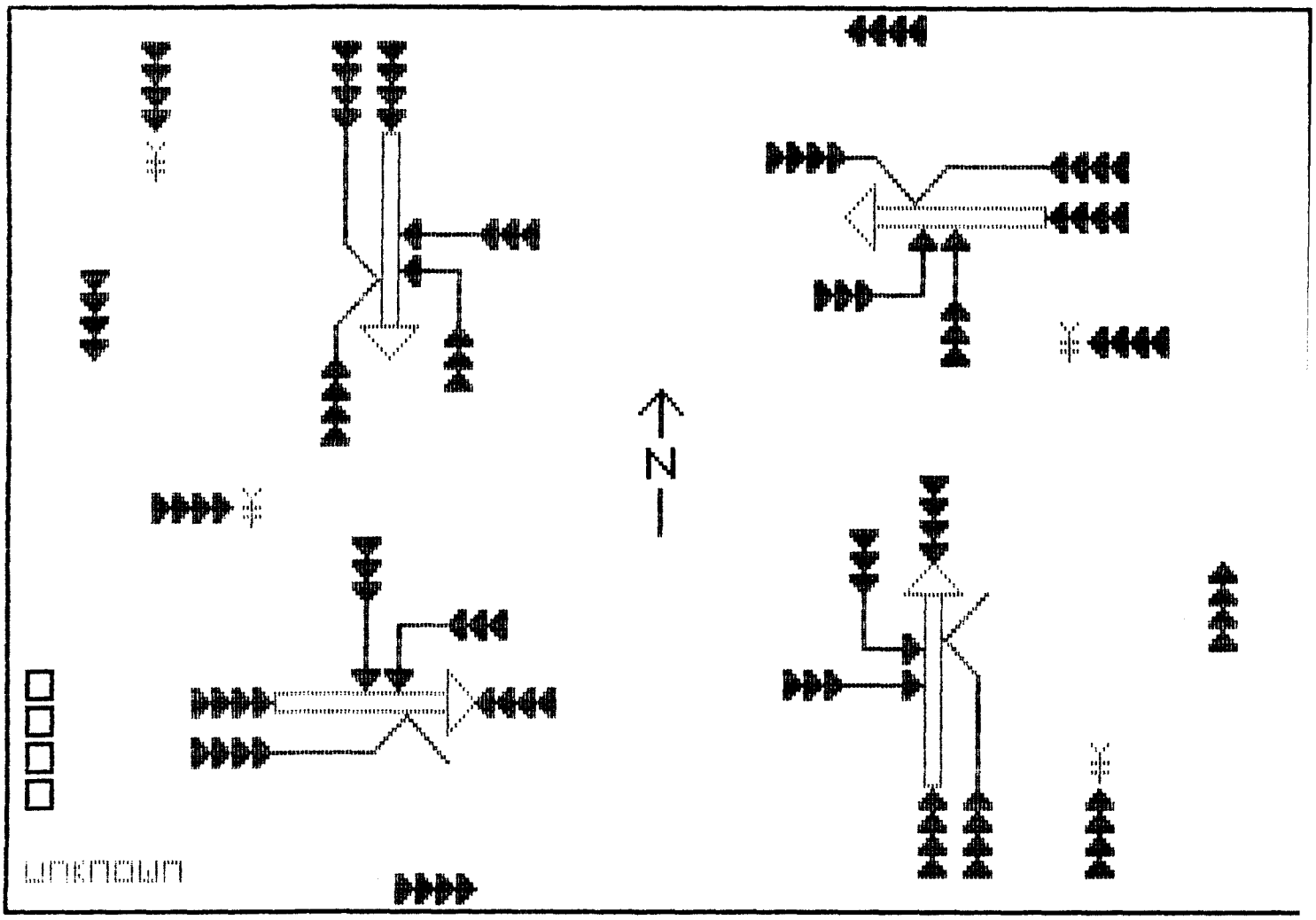
display the collision diagram on a graphics screen. A "Print Screen" command will plot a diagram and summary information on a line printer.

Each accident entered is depicted by a symbol on the collision diagram. The symbols are representations of one of the seven accident types that appeared on the data entry screen. A small arrowhead will appear at the rear of each symbol indicating the number of times an accident of this type occurred. The final report is a one page print that contains a plotted 5" x 8" graphic representation of all of the accidents and textual information including title, user-defined statistics, and summary information of all the accident types (Exhibit 1).

The SCCOLD diagram departs significantly from conventional, manually drawn diagrams. The most obvious difference is the lack of easily identifiable spatial cues. Familiarity with the SCCOLD accident type taxonomy is necessary to understand the intersection configuration. A low level of detail on a computer-generated illustration. For example, multiple vehicle events must be shown as separate incidents. Temporal, injury, and fatality information is not presented, and the only reference to this data is in the user-defined summary statistics. Because of the obvious lack of detail about each accident, more accidents can be presented efficiently. The most pleasing characteristic of the system is the ease with which data can be entered, and the simplicity of commands for moving from screen to screen. A user could easily become productive with the SCCOLD system to produce collision diagrams in about one hour.

ACCIDIAGRAM

ACCIDIAGRAM was originally written in BASIC for IBM-compatible PCs. It is available through D.K. Graphics. A sample accident diagram and corresponding summary statistics were obtained from the company. The software was not purchased for this research. A complete assessment of the program such as ease of operations and options for graphic and statistical presentation was not



COLLISION DIAGRAM

ALL TYPES ... ALL DIRECTIONS ... ONE OF EACH

FROM: 01-01-88 TO: 12-31-88
VEHICLES: 136 DAMAGE: $ 39999
INJURED: 22 KILLED: 1

ACCIDENT TYPE	NUMBER
RIGHT ANGLE	16
HEAD ON	8
REAR END	16
SIDE SWIPE	24
LEFT TURN WITH THROUGH	16
MOTOR VEH VS. NON MOTOR VEH	16
OTHER	16
UNKNOWN	4
TOTAL	116

Exhibit 1. Collision Diagram Example from SCCOLD Showing All Types of Accident Representations

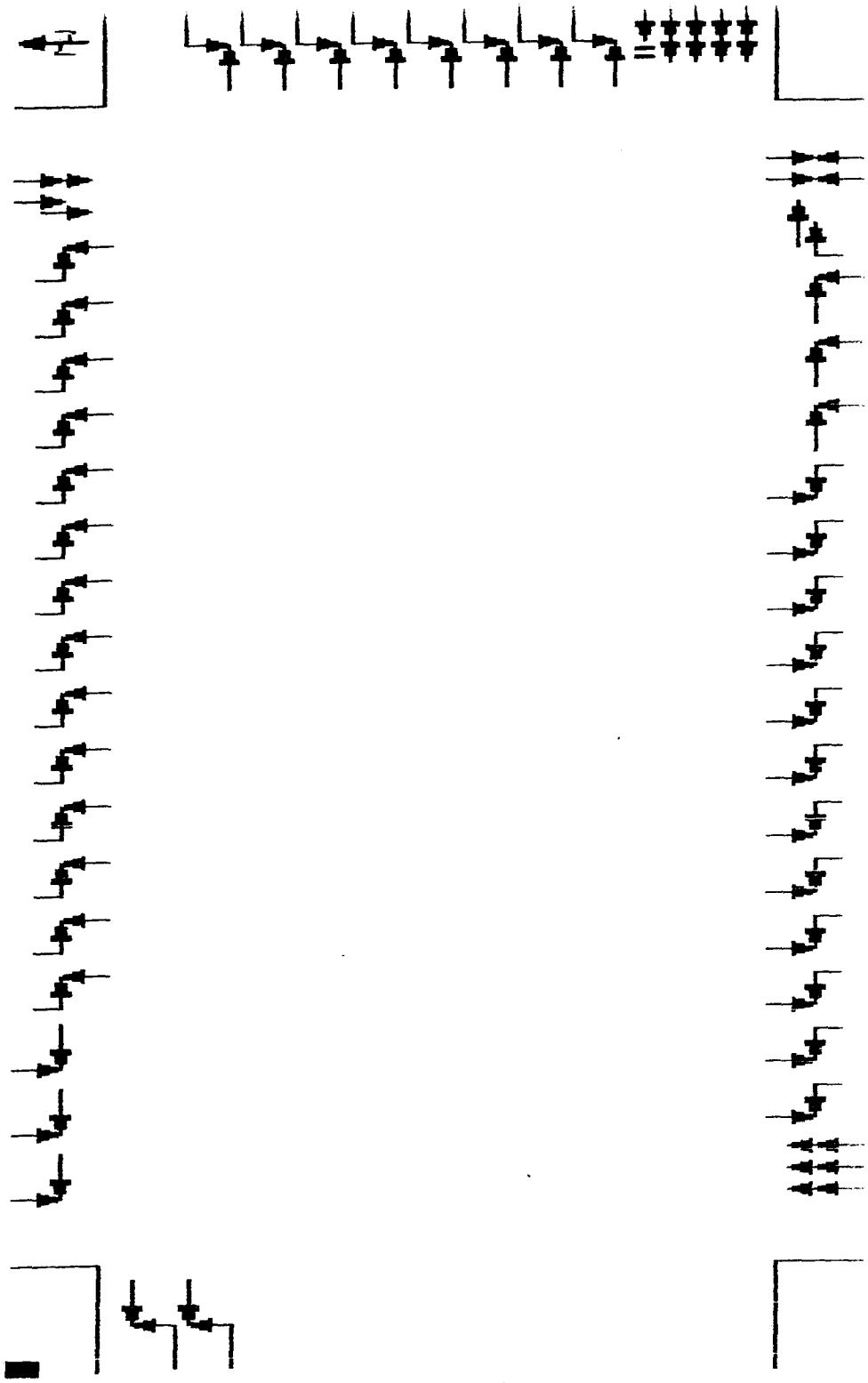


Exhibit 2. Collision Diagram Produced by the ACCIDIAGRAM System

completed, because it was not the intent of this research. However, the output that was provided could be examined in order to assess its utility at Washington State DOT.

Unlike the SCCOLD program, the ACCIDIAGRAM software does provide a basic spatial cue in terms of a simplified intersection. All accidents are depicted using individual symbols, even if the accident type is repeated several times (Exhibit 2). Similar to SCCOLD, no text is placed on the diagram itself, but summary pages of information are provided for further evaluation, or to be inserted into a report.

Since the software was not purchased, the user-friendliness of data input, the ability to read existing formats, and the range of presentation options are not known. From simple cursory examination of the products received, it seems clear that the diagram itself would have limited acceptance at WSDOT because of the disparity with existing techniques. Additional graphic detail, such as lanes, channelization, turning lanes, etc. would be needed. Further, it would be desirable to have more complete symbology describing additional accident types, temporal, and severity information. The addition of text on the diagram itself would be necessary if the diagram was to be used as a "stand-alone" product.

Oregon Department of Transportation (ODOT) ACDS

Overview

A series of programs, written in Microbasic 3.0, and graphic files were obtained from the Oregon Department of Transportation (ODOT). These programs and files contain the decision making routines and the graphic libraries for the drawing portion of their Automated Collision Diagramming System (ACDS). This system was demonstrated at ODOT for project staff on several occasions. ACDS is an ongoing project at Oregon and has been under way for over three years.

The following paragraphs present an overview of the ACDS. No documentation has been prepared for ACDS.

ACDS is actually two separate systems. The first system, much of which is written in COBOL is resident on the ODOT mainframe and has a series of nine steps. These nine steps are intended to extract data from the main departmental accident database to produce an accident listing. This listing contains the necessary descriptive information to determine the correct symbology to represent each accident. Commands are written in COBOL to extract the desired data from the main database and to reformat this data into smaller files (see Exhibit 3).

The resultant data file is then downloaded from the mainframe to the microcomputer and stored in a Colldiag subdirectory. This file contains the essential traffic accident data that is to be translated into graphic representations. The process to download the file is accomplished through IBM 3278 terminal emulation and the results (data) are placed in a file called ACCFILE.DAT.

A user begins the process at the microcomputer level by first invoking the COLLBEGN.EXE program. Interacting with screen tutorials, the user is immediately asked to choose one of three types of diagrams. The three types are (1) a cross type intersection, (2) a tee type intersection, or (3) a straight line section of roadway. These diagrams are simplistic and do not contain information on lanes, crosswalk, etc. The program is not capable of automatically determining the intersection type from the data, the user must provide this specific information prior to any subsequent action.

Once the intersection type is selected, the user provides additional textual information such as city, county, highway, or street names and the program then branches to the appropriate file to "retrieve" the corresponding drawing instructions for that particular intersection.

The drawing "instructions" are contained in AutoCAD drawing interchange files (.DXF), which are standard ASCII text files. The .DXF files can be easily

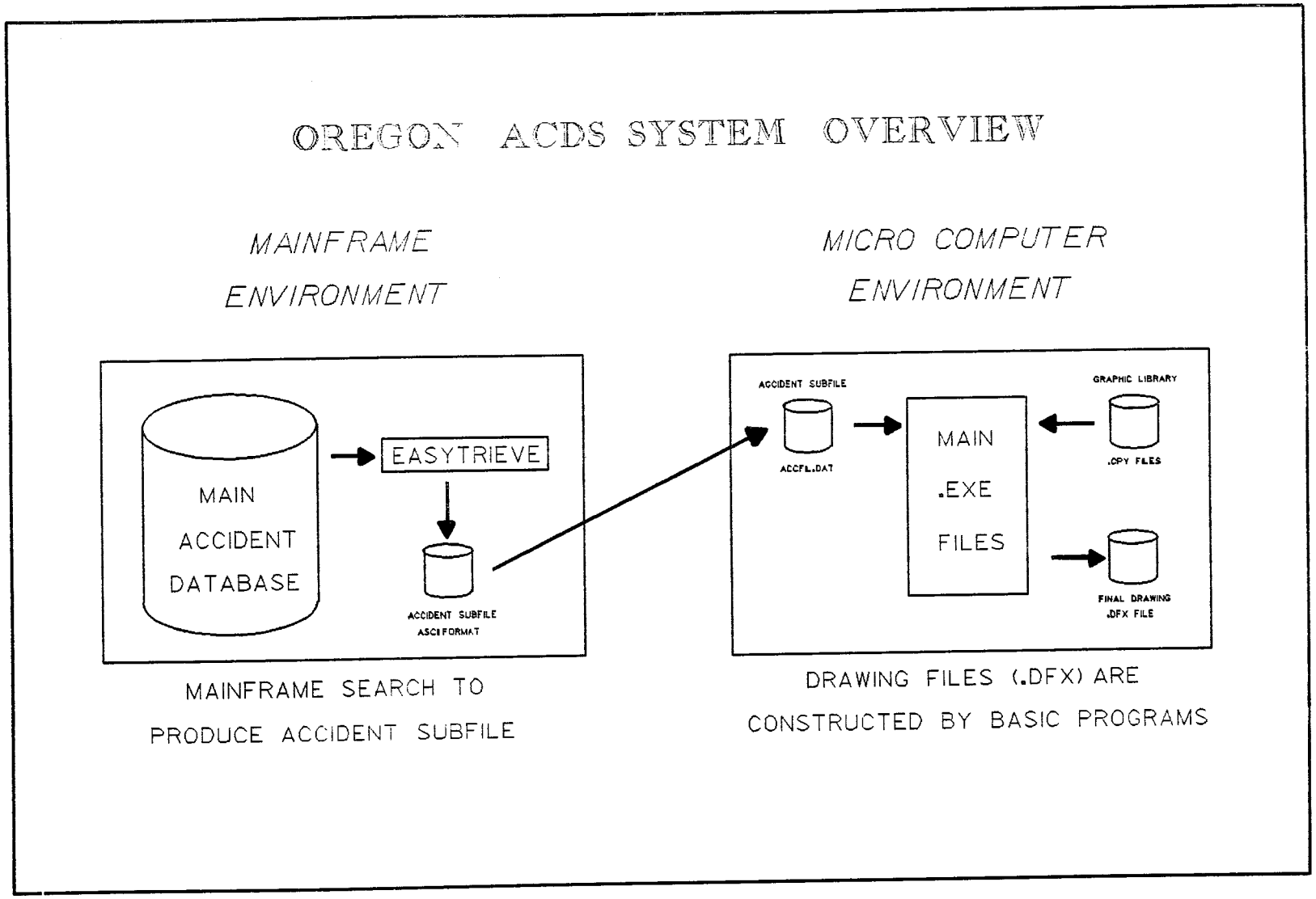


Exhibit 3. The Oregon ACDS System Overview

translated into the formats of other CAD systems. These files contain the "settings" for variables associated with the drawing (scale, units of precision, drawing limits, etc.) and the definitions of the entities that make up the drawing (line, ellipse, arc, etc.). For each intersection type, ACDS has two corresponding .CPY files, one for a 8 1/2 inch by 11 inch paper size and the other for a 11 inch by 17 inch paper size. It also uses over 100 .CPY files for individual accident symbolization.

Once the intersection type drawing information is received, the main interpretive program, COLLDIAG.EXE is involved. COLLDIAG interprets each record of the ACCFILE.DAT file and determines which graphic symbology from the "library" would be most appropriate to represent each accident type. These graphic depictions are stored as CAD "cells" or "components" in .CPY files. The placement of each accident symbol in the appropriate location within the corresponding intersection drawing is based on the ability of COLLDIAG to interpret the ODOT's coding instructions for accident type, location of impact, and type of collision. A sample plot is provided which depicts the results (Exhibit 4).

Strengths and Limitations of the ACDS

The ACDS is superior to the other automated collision diagramming systems that were examined for this project. ACDS provides spatial cues in terms of stylized intersections (although only three varieties). The graphic symbolization of the individual accidents provide much more detail about the accident, due to more (over 100) symbolization possibilities. These symbols are placed much more accurately within the intersection diagram based on the actual location of impact. Thus, the person viewing the diagram will obtain a more locationally accurate and descriptive graphic representation of the accidents at a particular location.

There are, however, limitations to ACDS. The system is complicated, containing over 20 programs and 150 drawing (.CPY) files. ACDS requires communications and compatibility between the mainframe and the microcomputer and the transfer of files is a batch process. Pulling the data from the mainframe

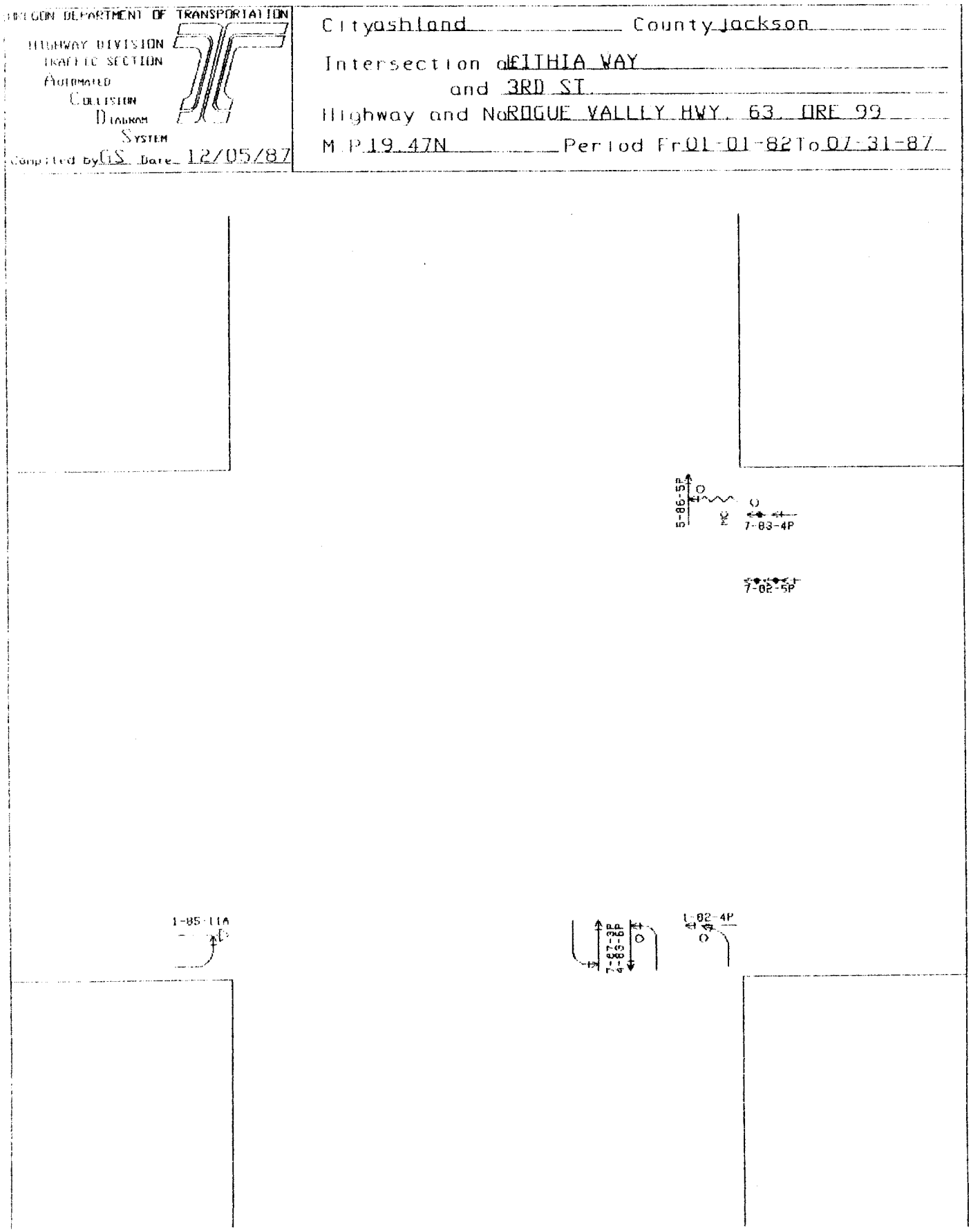


Exhibit 4. Collision Diagram Example Produced by ACDS

database to the microcomputer can take up to four hours, in which case it would be more efficient to produce the diagram manually.

The current status of the system allows for diagramming of three intersection types. This fulfills less than 50 percent of user requests. Unique intersections and detailed description of lane structure cannot be depicted.

The major obstacle is that the coding scheme is very specific. In order to use ACDS, the ODOT coding techniques must be applied. Definition of accidents, characteristics of the road, description of intersection type, location of impact, and type of collision all must be recorded according to the ODOT motor vehicle traffic accident coding instructions.

SUMMARY STATEMENTS

Conventional Software

Based on the literature review and the examination of several existing microcomputer automated collision diagramming techniques, current systems have three major weaknesses. These shortcomings are listed below:

1. Commercially available software offers less detail graphically about each accident than current WSDOT manual diagrams. The graphics are stylized to a point that minimal information is gained through visual inspection. The symbols representing accidents are dissimilar to WSDOT conventions.
2. In order for any existing system to be utilized, the accident data must be reformatted. Data storage formats for accidents would have to be revamped or programs could be written for the translation into the necessary formats. Current systems can only be used for exactly that set of data for which it was developed.
3. There are a limited number of intersection types available to place the accident symbols. Most systems offer only the cross-type

intersection, or other simple, generic types. The ability to portray **specific** ramp configurations, intersections, straight line segments, or particular sections of roadway, is limited. Other locational cues, such as lane channels, crosswalks, or signals, are not portrayed.

The next section of the report addresses the possibility of using a knowledge-based approach to enhance existing systems in order to eliminate some of the shortcomings.

Knowledge-based Approach

During the literature review four areas were identified where Artificial Intelligence (AI) techniques might be helpful:

1. selecting and preparing an appropriate graphics base,
2. extracting and interpreting correct data from files,
3. presentation of symbols representing collisions, and
4. interpretation of information.

Although work in each of the four areas is possible, time and budget constraints permitted examination of only one of the tasks, and did not permit the development of any software. Consequently, this section outlines the results of a requirements analysis interview in relation to the importance of these tasks.

In an interview with WSDOT District 1 Traffic Safety personnel, the prospect of providing a software program that would help automate collision diagram production was accepted favorably. However, this was assumed to be true, and the reason for the interview was to determine if the above tasks would be beneficial if they were automated using AI techniques.

Since manual techniques are used currently to produce diagrams, it seemed reasonable to focus on functionality that would complement an ACDS program rather than rework one using another programming language.

Of the four tasks, Task 2 was identified as perhaps the most beneficial area of focus. When constructing collision diagrams by manual techniques sometimes errors are discovered in the collision codes at the time symbols are drafted. These errors inadvertently enter the database either at the time a collision is reported or during data entry of the codes. Regardless of when and where it occurs, a processing of collision data are needed to insure that the data are logically correct before they are symbolized using an automated technique.

It would be possible to work with a traffic analyst to compile a set of rules that determine the logical consistency of collision codes as they are retrieved from the master database. These rules could be automated using an AI programming environment that expresses rules in terms of:

IF (premise is true)

THEN (conclusion is true -- within some certainty)

Several commercial packages exist with IF-THEN knowledge representation constructs being the foundation of the implementation. These rules are stated in the program in such a fashion that they can be combined to determine the logical consistency of the data submitted to the program for processing. An interactive program takes input in the order of the rules. A batch program can read input as needed.

The easiest AI programming environments to use involve expert system shells. An AI shell is a software package that implements knowledge representation constructs in a manner such that lengthy programs do not have to be written. (In some sense this is like a high level database language.)

Conventional programming languages are very useful for algorithmic problems, i.e. given a finite sequence of steps a problem can be automated. Knowledge-based programming languages are very useful for decision making problems that do not have a set pattern of steps. Knowledge-based programming techniques are perhaps more useful than conventional programming techniques

when domain dependent problems need automation because decision rules are easier to construct and update using knowledge-based languages. This is especially true when a programming shell is being used.

When working in a computer-assisted production environment an analyst relies on a software program to handle the routine and tedious decision making processes for plotting displays. When data are read from a file by ACDS software they are assumed to be correct, e.g., logically consistent in terms of vehicular turning patterns. A procedure to check the logical consistency of the data is required with automated techniques because a human who is responsible for drafting symbols has a chance to mentally preprocess the placement of symbols, whereas a computer will do exactly as instructed by symbolizing whatever data are in the files. Consequently, the routine task of plotting a diagram using manual techniques also incorporates a task to check the logical consistency of the data. Without the logical consistency check on the data, errors can be plotted easily that might not come to the attention of an analyst very readily. Although some errors can be eliminated upon visual inspection of the diagrams, this adds an extra step which should not be necessary.

PROTOTYPE WACDS

GOALS OF WACDS

Existing, inexpensive, or public domain systems for the automated construction of collision diagrams cannot be used by WSDOT, because of the file format of the accident files. Again, major problems with the ACDS are (1) it is too simplistic and does not offer the viewer any clear indication of precise spatial locations within the intersections or adequate thematic information about the accidents, and (2) all programs require data that are in a specific format. Thus, major adjustments to existing accident coding schemes would have to be made. A completely new automated, prototype system was developed for use at WSDOT based on the following broad goals aimed at eliminating the two main obstacles:

1. The system should depict the accident information similarly to the current, manual method in a manner that is simplistic and inexpensive.
2. The Washington Automated Collision Diagramming System (WACDS) should use existing coding schemes and file structures for locating the accident, type of impact, impact position, etc. The coding scheme should be in agreement with the coding schemes found in the current WSDOT traffic accident analysis and location manual.
3. The system should provide for the display of as many spatial cues as possible. Current systems are restricted to one, two, or three generic intersection types. A system should be developed which depicts each intersection as close to reality as possible. This includes accurate street and lane configuration.
4. The system should be integrated with other department spatial information systems such as maps, video logs, photogrammetric

techniques, etc. The sharing of data among systems should remain invisible to the user. It would be desirable that data collected or displayed and one scale -- large scale intersection diagrams -- have the capability of undergoing an aggregation process up to smaller scales for utilization in other display techniques or analysis software.

With these four broad goals as a guide, a WACDS prototype was designed. This prototype is only the initial steps for future development. Major enhancements and recommendations for improving the basic system are made in Chapter IV.

SYSTEM OVERVIEW

Existing automated collision diagramming systems require an interpretation of coded data stored in one main accident database. The accident codes used by systems examined during this research project were different than those currently in use by WSDOT. In order to use existing systems, the accident codes would have to be changed, or programs would have to be modified to work with WSDOT formats. In most cases, original ACDS programs were not available for modification.

Programs written in QuickBasic 3.0 were obtained from ODOT. At first, the plan was to modify these programs to work with WSDOT accident codes to produce collision diagrams similar to ODOT diagrams. However, several hindrances became apparent:

1. All available QuickBasic compilers at WSDOT or for purchase were in version 4.0. A conversion from 3.0 to 4.0 was attempted but this became laborious, time consuming, and finally was abandoned.
2. The accident codes used by ODOT are somewhat different than that used by WSDOT. Major modification of programs became necessary to accept WSDOT schemas.

3. The ODOT ACDS program was limited to only three intersection types that lacked significant detail. A more robust system that used specific intersections or interchanges was desired.

Given these overwhelming encumbrances, it was decided to salvage only those portions of Oregon's ACDS that were applicable to WSDOT needs. It was decided to build the decision making portion of the system from scratch.

The basic core of the WACDS is depicted in Exhibit 5, and consists of four environments in which the user will work to obtain the final, plotted collision diagram. Depending on how the user's microsystem is configured, each environment can be accessed from a single microcomputer screen. The four environments are explained below.

Mainframe Connection

The first step, the mainframe search, is exactly the same as a mainframe search when manual collision diagrams are produced. The WSDOT's Problem Identification and Course Analysis (PICA) system format is used in WACDS. It is a subfile system that extracts data from the Master Accident Report System (MARS). Details for the creation of subfiles are not presented in this report and the assumption is made that WACDS will always be using the PICA-MARS. It is assumed that one will have a working familiarity with obtaining PICA-MARS reports. Specifically, WACDS uses columns 59-68, the diagram analysis data. The data in these columns is used in the decision making process to select accident symbology and to place appropriate symbols in an intersection. The file is named by the user.

PC-DOS Environment -- Filebuilder

Once the PICA-MARS formatted file has been downloaded to a PC, the main WACDS Filebuilder program is invoked. The process within the Filebuilder

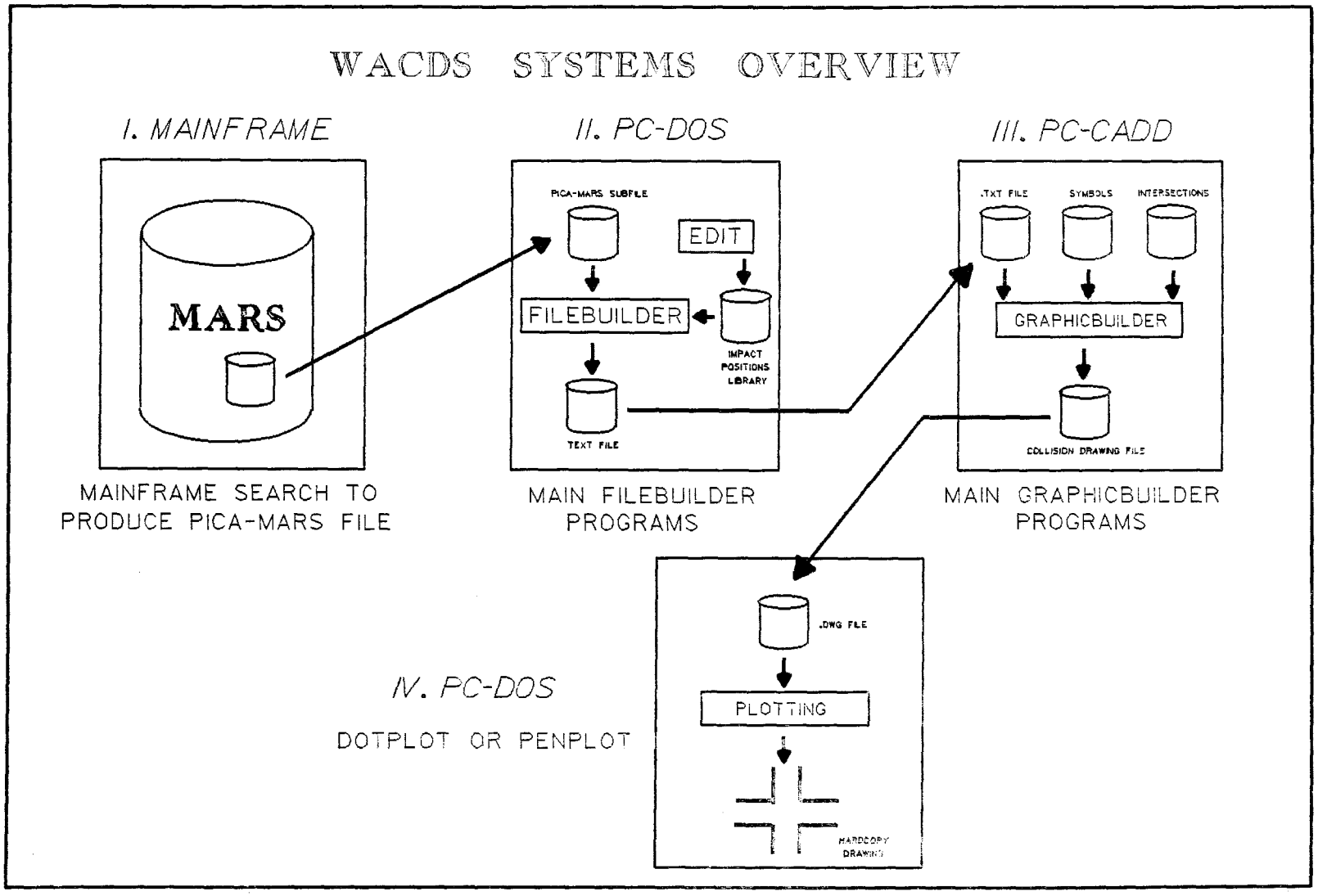


Exhibit 5. The WACDS System Overview

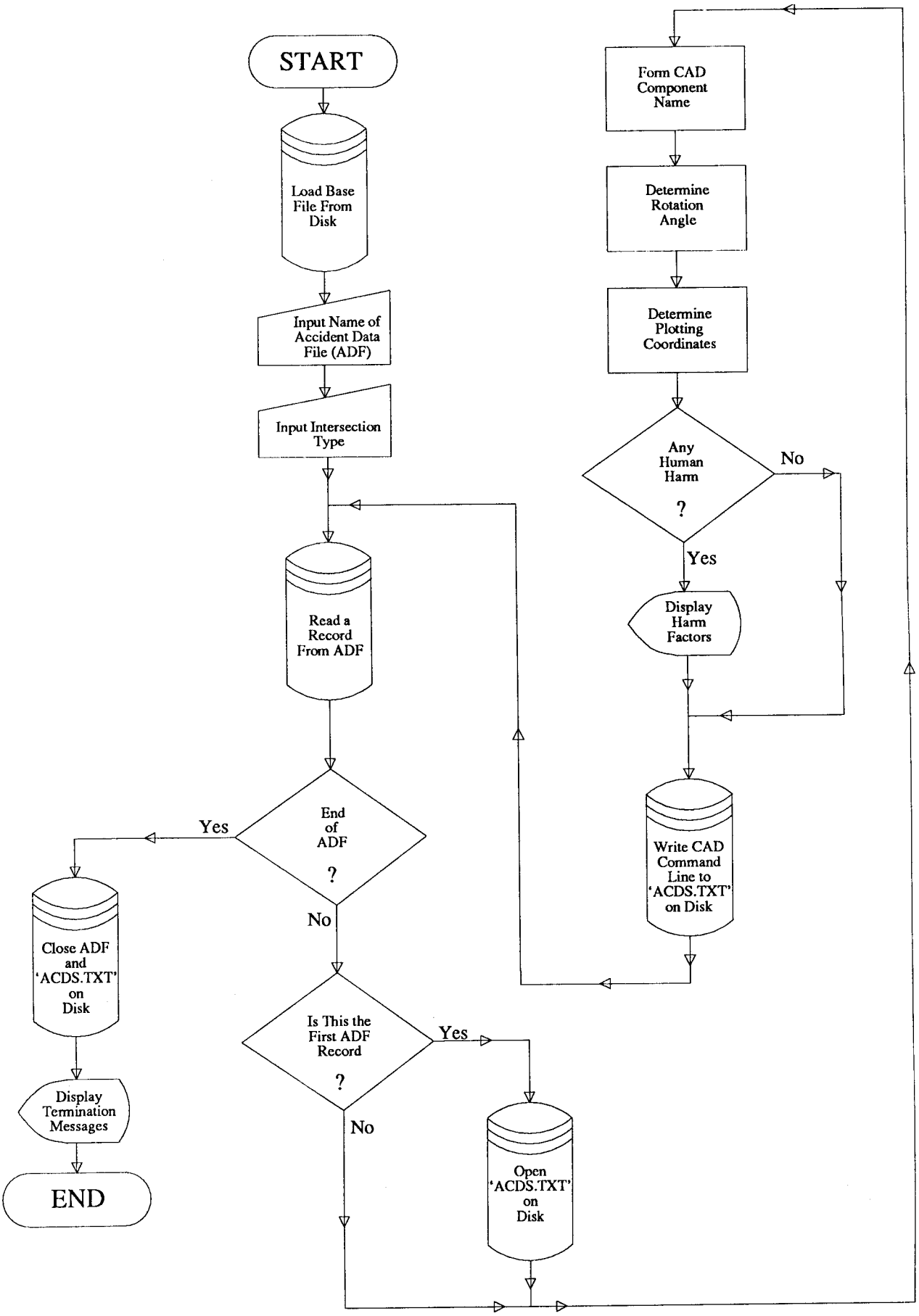


Exhibit 6. Flowchart Depicting Processes in Filebuilder

program and the components that are used by the program are outlined in Exhibit 6.

Selection of the Base File

The program queries the user to input the type or name of the intersection, interchange, or straight line segment of roadway. Filebuilder then will use the name to locate the corresponding base information from a library of base files.

These base or initialization files (.INI) contain data about the impact position based on the WSDOT code location scheme. For example, on a simplistic, cross type intersection with the increasing direction, North, the corresponding code locations are depicted in Exhibit 7. By placing the intersection over a grid with a known 0,0 location, each of the code location zones and areas within each zone are subsequently identified in terms of an X,Y coordinate. Exhibit 8 depicts the code location zones with corresponding coordinate values for the beginning of each zone. Also included are the rotation angles from 0 degrees (right reading) of each zone.

As the PICA-MARS accident record is read, the proper location zone is identified and the accident symbol is placed within each appropriate zone. Each zone can depict a total of 12 accidents. Thus, for the cross type example, the first symbol representing an accident in A-1 will be placed at coordinate 5.5, 3.5 on a grid.

If a second accident occurred in A-1, the symbol will be placed at 5.0, 3.5. Subsequent A-1 symbols will be positioned accordingly until all of the available slots have been used. The complete set of positions for symbol placement in a single cross type intersection are depicted in Exhibit 9.

The base or initialization files consist of a library of intersection or interchange types that have stored intersection name, rotation angle, and corresponding coordinate values for each impact zone. Exhibit 10 demonstrates this impact position library concept. This library is intended to be modular and files are independent of one another. Using the impact position library concept, it is

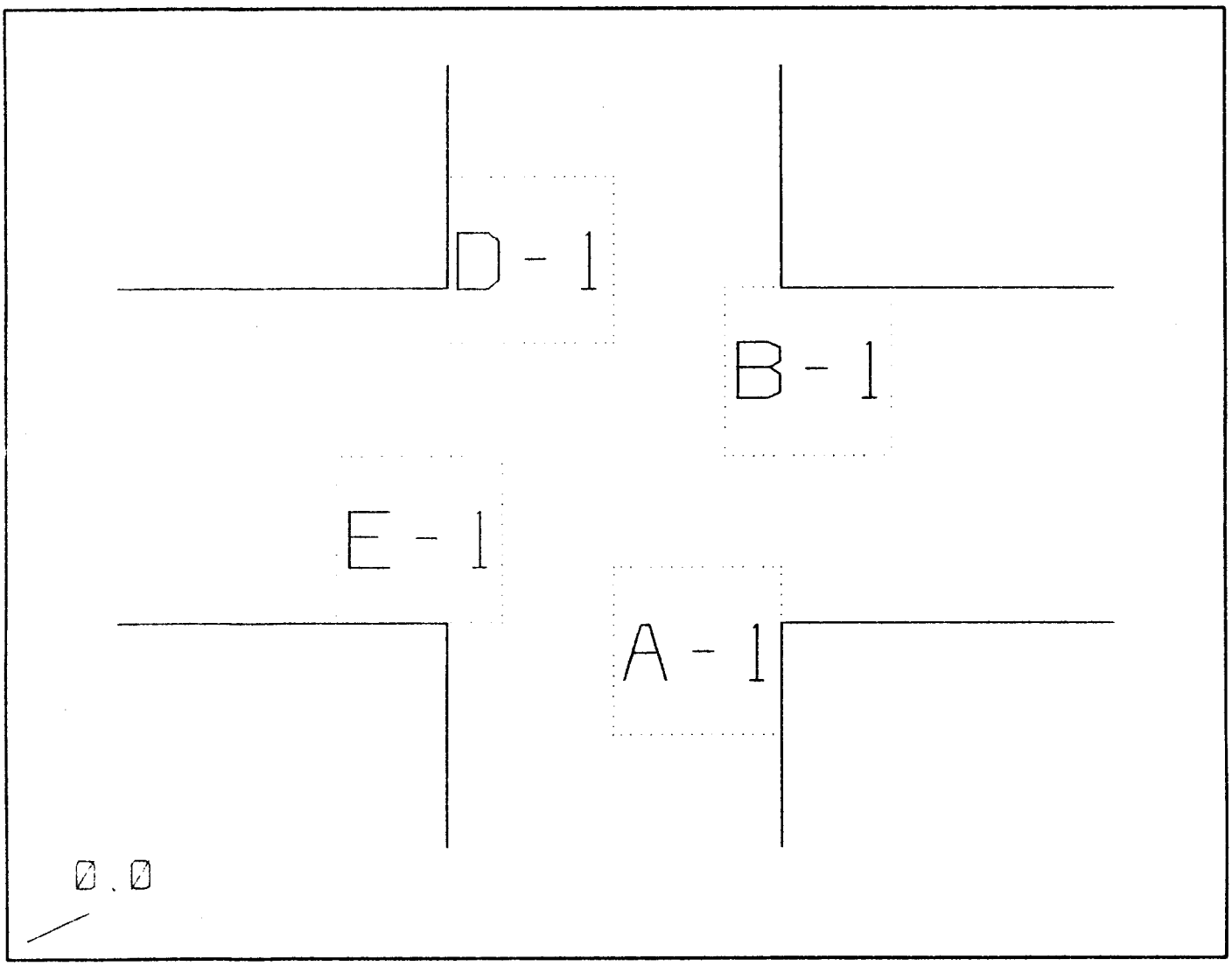


Exhibit 7. Impact Location Positions for a Simple Cross-type Intersection

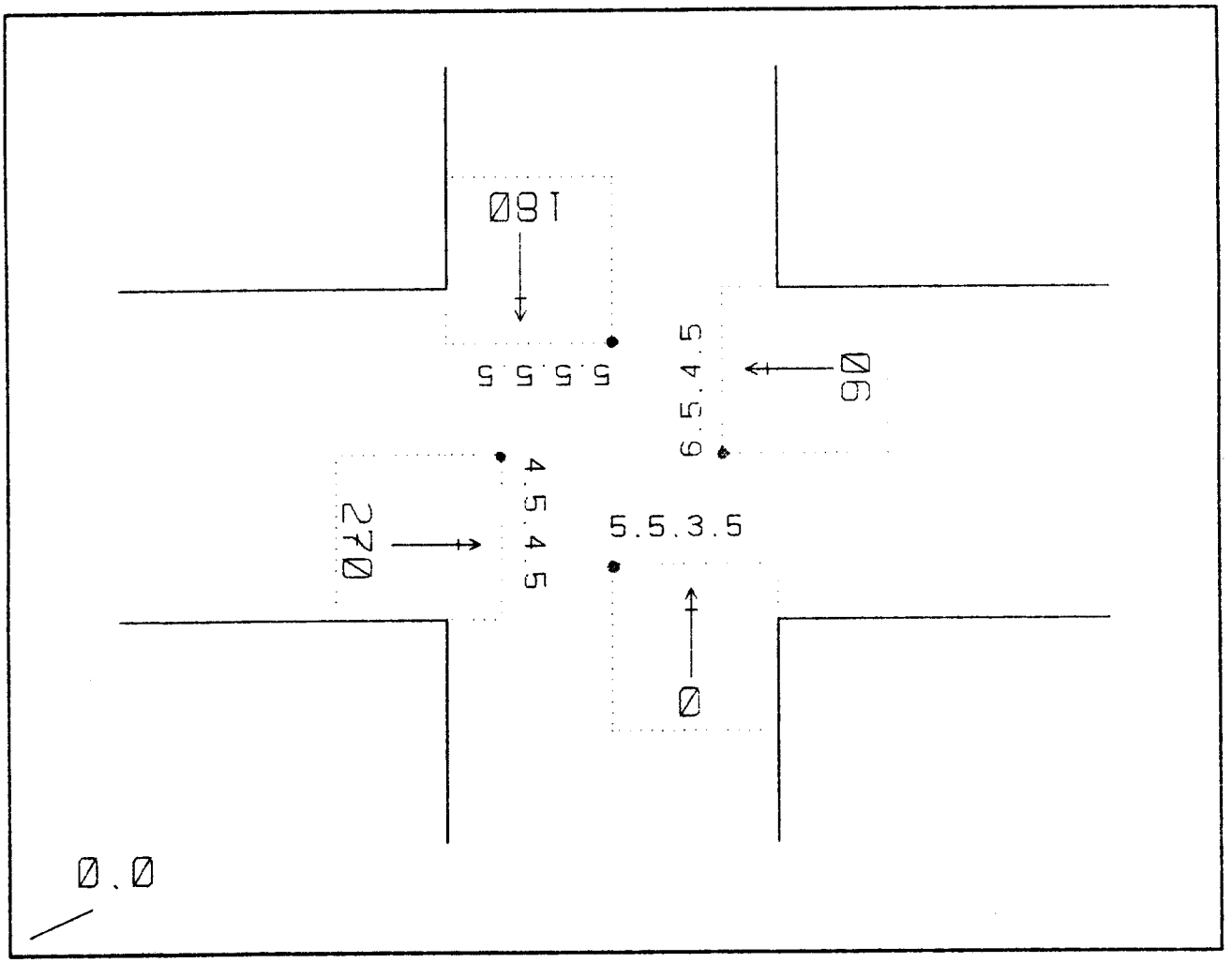


Exhibit 8. Rotation and Coordinate Values for Impact Location of Cross-type Intersection

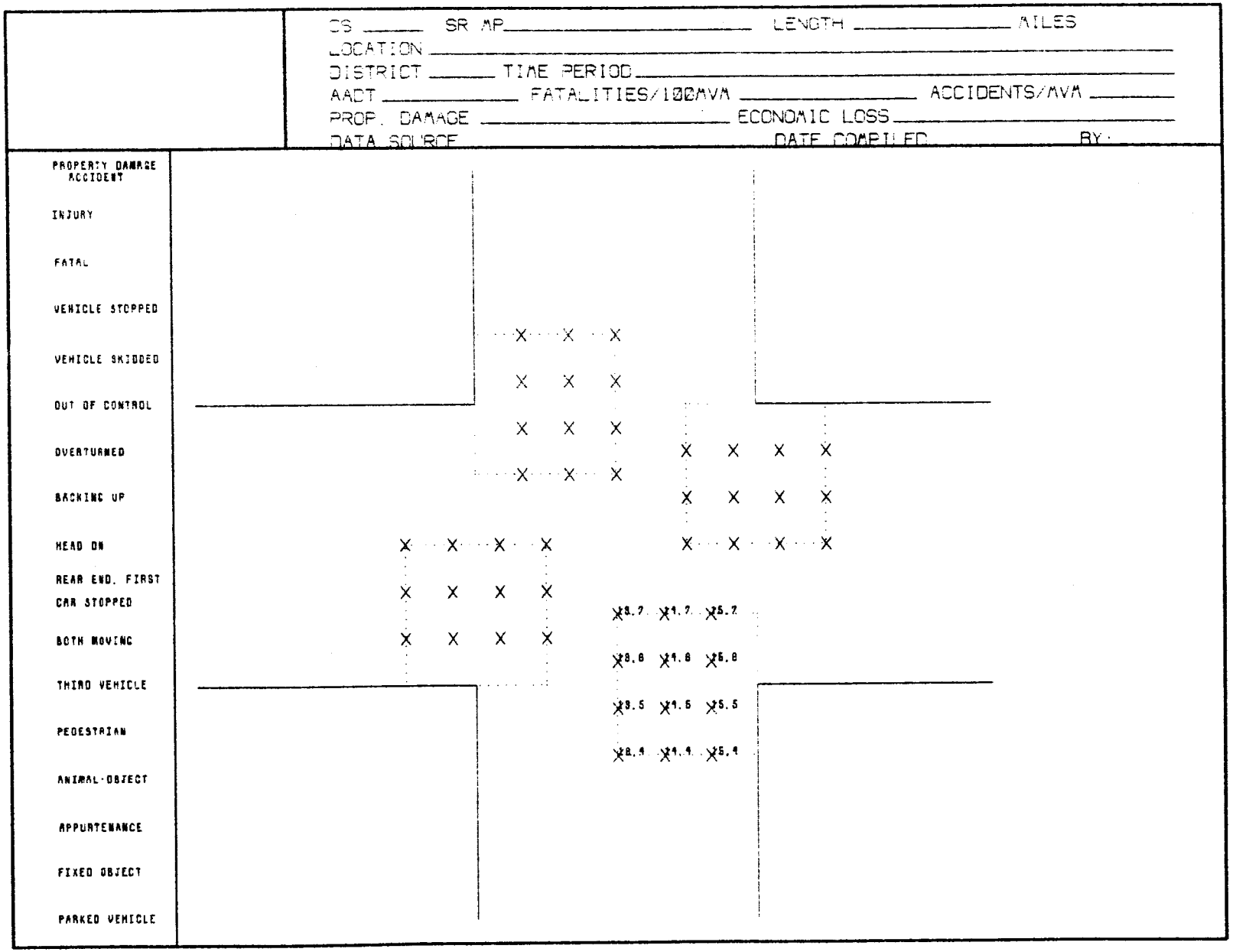


Exhibit 9. Accident Placement Locations for Cross-type Intersection

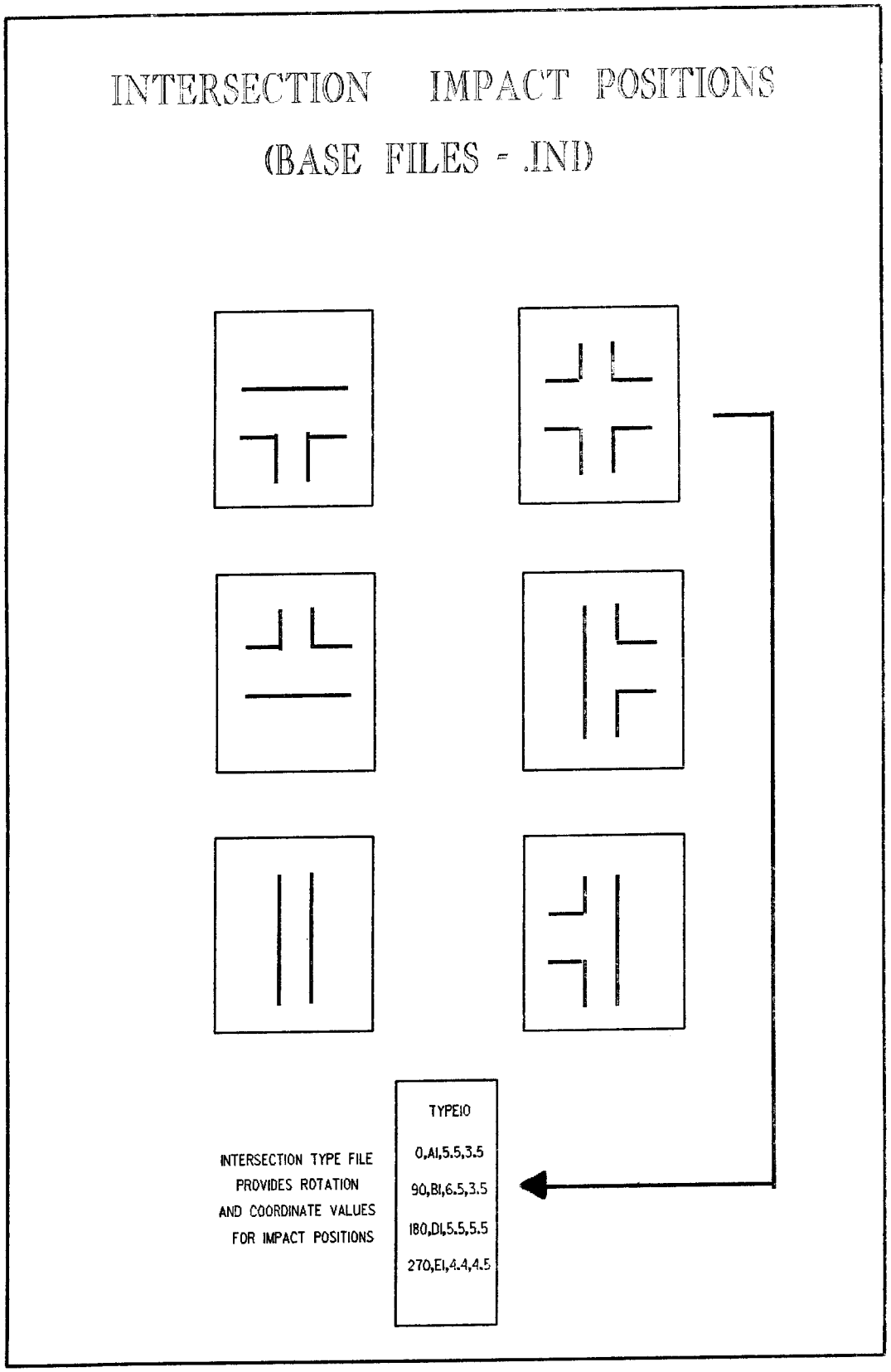


Exhibit 10. Graphic Depiction of the Impact Position Library and Corresponding Type File

possible to insert unique intersections as long as the impact position zone and corresponding coordinate values have been previously identified. User interface programs (Editor) are planned, which would aid the construction of these files and placement into the library. It is necessary that intersections be "built" only once and then stored in the library to be used by Filebuilder, repeatedly.

Interpretation of Accident, .TXT File Construction

Once Filebuilder has retrieved the appropriate base file from the library, it reads the PICA-MARS file that was downloaded from the mainframe. For demonstration purposes, a text file was created for an intersection depicted in the Handbook of Traffic Engineering and in the SCCOLD example. A corresponding data file for these accidents in this example was established using the proper accident codes in the appropriate format. That is, the manually drawn diagram has been coded in terms of the WSDOT coding scheme, and the data for collision diagramming has been inserted in a file (ACC.DAT) for integration by Filebuilder.

Filebuilder reads columns 59 through 68 of each record of the .DAT file and determines vehicle direction, collision type, and input location. It then determines an accident "name." Using the initial information from the base file, text strings are built that will be used in the CADD environment of the WACDS to actually plot the symbology. This subsequent text file (ACDS.TXT) consists of the following data for each accident record:

- scale factor for the symbol,
- rotation angle for each symbol,
- the symbol name, and
- the X-Y coordinate for placement of each symbol.

The text strings are written in a CADD command language for the placement of graphic components in a drawing file. Once this text file is created, Filebuilder is terminated and the user enters a CADD drawing environment.

PC CADD Environment

The CADD package that is used in the WACDS is Generic CADD Version 3.0. It is not necessary that the user have a working knowledge of Generic CADD in order to construct the collision diagram, several two-letter key entries will invoke the necessary files.

The user enters a CADD drawing file in which all of the drawing parameters (scale, text, line widths, coordinate systems, etc.) have been previously established. The first task for a user is to type the name of an intersection on which accidents will be plotted. This intersection is drawn on the screen. The drawing instructions for each intersection are located in an intersection graphic library. When the user types in an intersection name, a search of the library is made until there is a "match." There is one file per intersection, containing previously compiled graphics.

The users can also pick the intersection type from a menu. A menu is provided on a digitizing tablet and contains icons of the more common, or custom intersections that are drawn by individuals using the system.

Once the intersection graphic is placed on the screen, the user can show a "frame" around it. The frame consists of a WSDOT collision diagram form that contains a legend of accident types and slots to provide necessary information about the diagram. The frame is placed by simply typing in "CP-FRAME."

The placement of individual accident symbols is initiated by typing in the name of the .TXT file that was created by Filebuilder. The .TXT file is actually a batch file that contains Generic CADD graphic commands for the placing of components (graphic cells) at desired locations.

Accident Symbol Library

A symbol library of accident representations was constructed for the WACDS. The CADD instructions for drawing each graphic cell in the library are located in component (.CMP) files and there is one file per accident type. Thus, when the batch file for component placement is read, there must be a corresponding

.CMP file that matches the accident type name indicated in the batch file. These files represent the set of graphics that will be used on representations of individual accidents. The library is not complete, but is intended to grow and be modified as required.

To date, there is no capability for the automatic placement of text next to the individual graphic symbol. However, by interacting with the menu, the user can manually place text on the drawing. This is accomplished by selecting from the "text placement" boxes on the menu and working in the "active area" of the menu. Future enhancements of WACDS will provide automatic text placement.

Plotting Environment

A collision diagram displays in a CADD environment as a user sits in front of the screen. Once the diagram is complete, the user can manually enhance it, placing additional graphics or text. The user then saves the finished diagram, closing the file, and returning to the PC-DOS environment to make a plot.

Plotting is archived through the DOTPLOT program for the Generic CADD software. Crisp high resolution drawings are produced on the DOT matrix printer. The size limitation of the final plot is 14 inches wide by as much as 96 inches long. The user simply follows the DOTPLOT prompts displayed by the program (Exhibit 11).

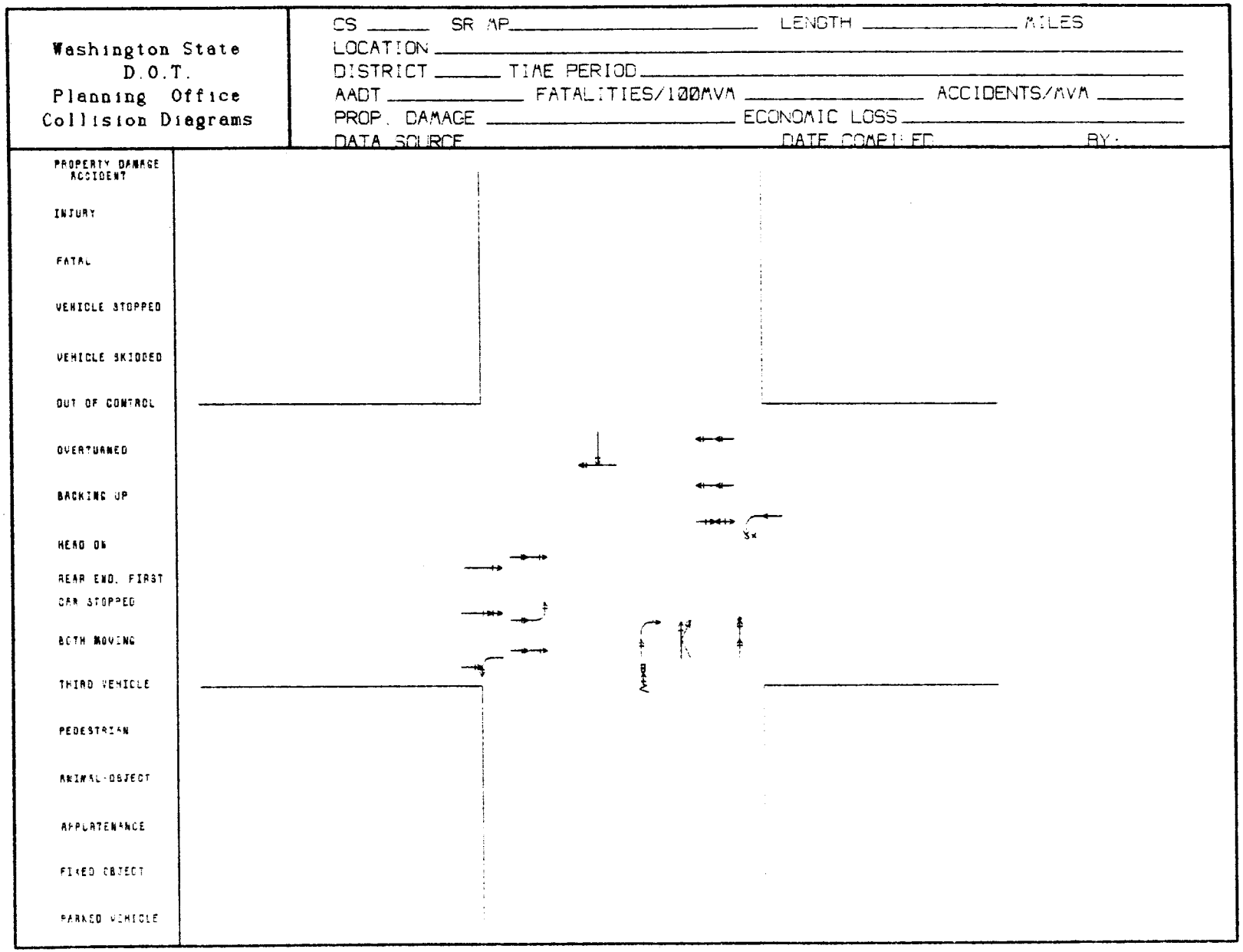


Exhibit 11. Sample WACDS Collision Diagram

IMPLEMENTATION

DEVELOPING A PRODUCTION LEVEL SYSTEM

This research has provided a prototype for a WSDOT automated collision diagramming system. This prototype uses conventional programming techniques and "off-the-shelf" CADD software. The WACDS uses the following libraries of data:

- **Impact Positions**. Reflect the WSDOT coding schemes and contains information about the placement of symbols within each intersection, the rotation angle of text and symbols, and the design plane coordinates.
- **Accident Symbols**. Contain the accident symbols which are used by ODOT, and several new ones developed for WACDS.
- **Intersection Types**. Only three generic types of intersections are available in this library: straight line, cross, and "T."

The Filebuilder and Graphicbuilder programs use data from each of these libraries to construct the final graphic. For each entry in the accident code library, there must be a corresponding entry in the intersection type library. Filebuilder is currently limited to a handful of accident interpretations and three intersection types. However, the prototype clearly demonstrates that any collision diagram can be constructed with computer assistance, using WSDOT formatted data as input. The resulting graphic representations are drawn more efficiently than, and the quality is comparable to, those diagrams which are produced manually.

The prototype provides a framework for the inclusion of more complex drawings. With enhancements, WACDS would be capable of drawing collision diagrams for complicated, and be utilized on a daily basis. The steps necessary to bring WACDS to full production are:

1. The accident symbol, impact position, and intersection type libraries need to be expanded. This research has found that AI techniques for the construction of intersection graphics are not practical at this time. Drawing instructions for each intersection, interchange, or road segment, must be on file prior to the construction of the collision diagram. These graphic files are created by the manual digitizing of hardcopy representations of intersections. In addition, each graphic in the intersection type library must have a corresponding impact position file. The identification of the impact positions and the proper coordinate values for each intersection is a manual process, but this process needs to be implemented only once.

   The library containing the accident symbols is adequate for representation in a similar manner to manual methods. The symbol library obtained from Oregon DOT must be further adapted to work in WACDS. Basically, this involves renaming each file to reflect the WSDOT coding scheme corresponding to text strings which are constructed in Filebuilder.

2. Filebuilder must be enhanced to interpret all of the accident possibilities list in the WSDOT Traffic Accident Analysis and Location Manual. This program is modular so that each change, edit, or addition does not require a re-write.

3. The task of constructing graphic intersection types, identifying impact positions, and subsequently adding to the libraries, must be simplified. To accomplish this, a series of programs should be written which provide screen menus, icons, or prompts, in a manner which is simple and understandable to the user. Specialized menus need to be developed in Graphicbuilder to simplify the process of loading

specific intersection types, and aid in the manual placement of symbology or other accident related graphics.

4. WACDS should be improved to include textual information at each accident location, or wherever text is necessary. Currently all text is placed manually.

5. There is a need for the capability of placing more than 12 accident symbols at any one impact position. If it becomes physically impossible to place numerous accident symbols, then some form of graphic representation must be developed to convey to the viewer this quantitative increase. This may be accomplished through color, shading, or adjusting the size of the symbol to reflect multiple occurrences of similar accidents. Overflow might also be handled by listing accidents in the legend area when there is no room for symbol placement.

IMPLEMENTATION

Assuming the necessary enhancements will be completed, the WACDS can effectively replace manual methods for constructing collision diagrams. However, the system is not fully automated. It is unlikely that WACDS will be completely automatic. Significant user interaction with the system will be necessary. But extensive user training will not be required, and it is anticipated that one could become fully productive in about one-half day, once the proper interface screens are in place. To implement WACDS at WSDOT, the following strategy is recommended:

- Someone from the Cartography branch who is familiar with Turbo C programming should work with a staff member at the Annex to complete each of the three libraries and expand the Filebuilder program. The staff member from the Annex would identify those

intersections which would be placed in the system, and locate corresponding impact positions. The cartographer would then manually digitize the intersection, and identify coordinate values, using one of the available CADD systems. Existing software would translate the files into the appropriate library. An estimate for bringing the libraries to a production level capacity, is roughly 0.25 FTE in Cartography and the Annex. No additional funding is necessary for purchasing software or hardware. The cartographic supervisor would supervise this stage of data capture and participate in writing Filebuilder enhancements and user interface screens.

- Once a production level system is operational, complete documentation would be prepared by cartography staff. The WACDS would be demonstrated at each transportation district, and users would be trained in techniques for customizing the system for that particular district. Districts would request from the Cartographic Section intersections to be digitized for input into the district files.
- The Cartographic Section would be responsible for the distribution of digitized intersections and any enhancements made to the Filebuilder programs or the accident symbol library.

It should be realized that every system requires maintenance. Intersections would be identified within, inserted into, or deleted from, the WACDS as needed. Enhancements to the Filebuilder and user interface for input and query would evolve. As an example, the Oregon ACDS has been under development for approximately two years with numerous modifications which reflect software upgrades or personnel changes. Management must come to the realization that no complete off-the-shelf software is available that would use a single organization's data files without major reformatting. As pointed out in the next section, it is

preferable that the WACDS be developed within WSDOT rather than contracting the project to a vendor.

FUTURE ENHANCEMENTS

This research has dealt with the preparation of collision diagrams by automated methods. The basic issues involve problems of converting the WSDOT coded data into a graphic presentation using specific symbols to represent individual accidents. Symbols are placed on top of a graphic depiction of an intersection or section of roadway at diagram production time.

Although remarks concerning future enhancements are limited to this central theme of graphic presentation of accident data, other issues germane to the future enhancement of WACDS are data capture, presentation, and integration. These three issues are briefly discussed below. Proposed future enhancements to WACDS are depicted in Exhibit 12. For discussion about the broader aspects of traffic safety, such as data collection procedures, identifying hazardous areas or analysis techniques at WSDOT, the reader is referred to "Traffic and Safety Procedure: Project Report on the Safety and Accident Procedures."

Data Capture

Data capture refers to the digital representation of the roadway for graphic presentation. Currently, the intersection diagrams are stored in the library after the manual conversion process. Each intersection is manually digitized.

The Cartographic Section is currently experimenting with using scanned images as part of the graphic data file set. Scanning technology allows for a faster, more detailed conversion of hardcopy image. It is recommended that the use of this technology be investigated for the conversion of plan sheets or existing large scale prints of appropriate intersections. This would add detailed intersection types to the library more efficiently and at less cost.

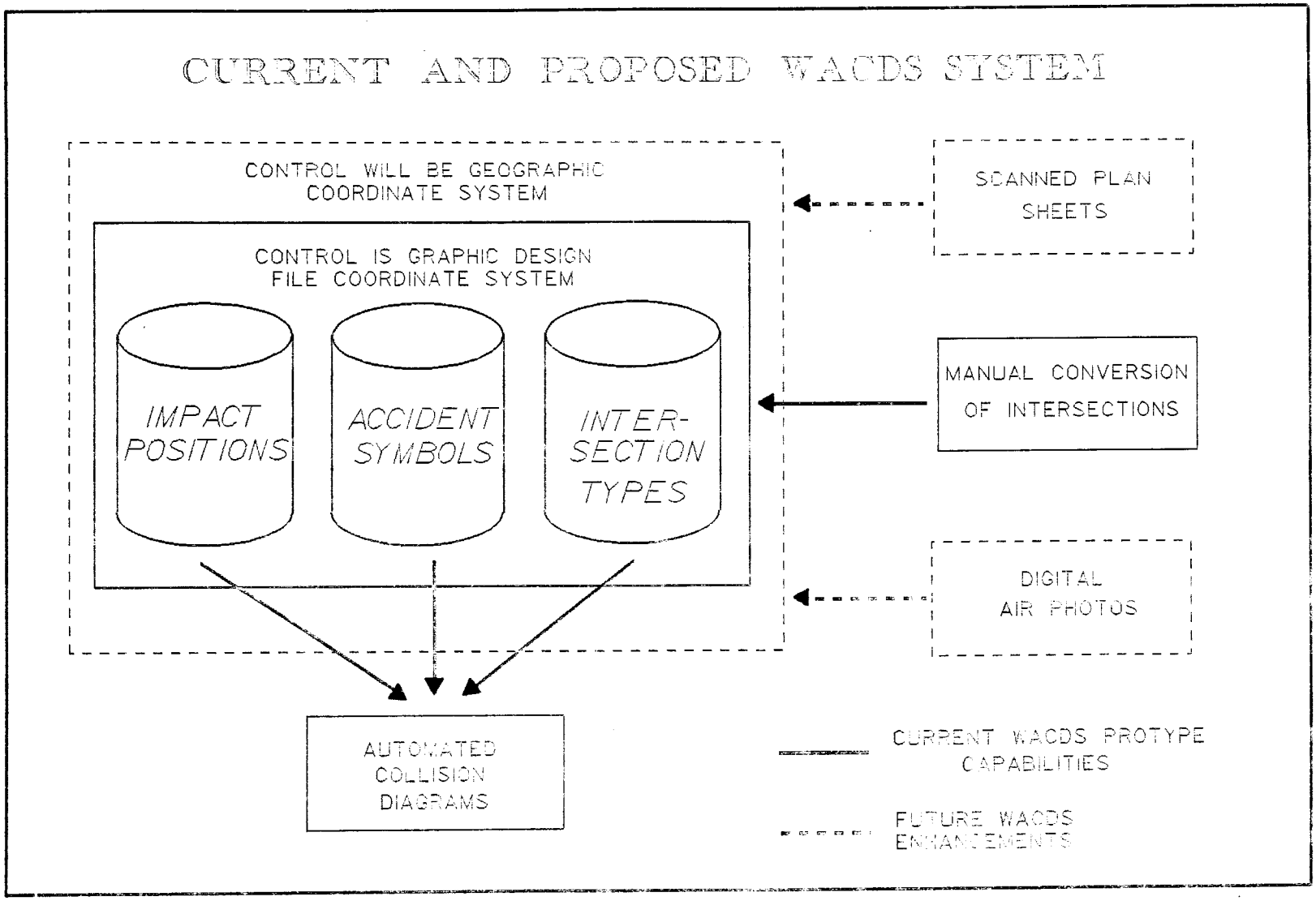


Exhibit 12. Graphic Depicting Current and Proposed Enhancements of WACDS

Many large scale drawings have already been converted into digital images. These drawings have been prepared by Cartography, Photogrammetry, Design, and Maintenance Divisions of WSDOT. There should be an attempt to locate these digital files on the CADD Intergraph system for use in the WACDS. This is one advantage of WACDS using a CADD package for image construction rather than spreadsheet software as some vendors have suggested.

Data Presentation

Current input and presentation in WACDS is limited to vector drawings (drafted line images depicting each intersection). Technology is available that would allow for more detailed "life-like" representations of intersections. The Cartography Unit is attempting to procure an image processing system which integrates raster (pixel) imagery and vector (line) data. This image processing system would serve two functions as applied to collision diagrams. First, air photos could act as the base from which the intersection diagrams are constructed. Essentially, the vector map would be "drawn" directly over the pictorial representation. This would allow for a more detailed description; one closer to reality. Further, by using air photos, it would be easier to keep the base maps (diagrams) up-to-date.

Secondly, the raster image (air photo) can serve as a base for the placement of the accident symbology. These symbol (vector) images would be superimposed on the air photo and the viewer would be able to make a quicker interpretation of events because of the significant amount of added detail offered through the photograph.

Further research is necessary for incorporating and using AI techniques into the graphic presentation of accidents. The design techniques of WACDS are based on manual methods, and the effectiveness of these symbols to convey the desired

information is not known. AI techniques may be used for implementing criteria for the utilization and selection of appropriate graphics.

Data Integration

WACDS is only one of a number of automated systems throughout WSDOT that are used by the traffic engineer to take corrective measures of a hazardous intersection. Steps should be taken that WACDS is integrated or compatible with these other systems.

One method for assuring compatibility among the various storage, analysis, and display systems is through the use of a common geo-referencing system. The primary locational-referencing system that is currently being utilized at WSDOT for analysis is the State Route Milepost (SRMP). The conversion of SRMPs for use in analysis and display is a cumbersome task. This type of control, which is basically linear, is subjected to several additional processing steps before it can be utilized for graphic display.

It is necessary to investigate the use of a geographic coordinate system as a referencing mechanism for the storage and retrieval of accident data. With the advent of lower cost stationary and kinematic Global Positioning Systems (GPS), latitude/longitude and state plane coordinates will become the controlling agent for all locationally related transportation data in the future. By using coordinate based CADD software such as Generic CADD, Intergraph IGDS, or AutoCAD, the WACDS can be easily adapted to accommodate most geographic referencing systems that will be utilized in the future. By having "true" geographic coordinates, it becomes necessary to move away from "stylized" or generic types of intersections used in most current collision diagramming systems, and develop a system that identifies the actual location of the accident as accurately as possible. WACDS has these abilities.

REFERENCES

1. Yeh, C.-I, Ritchie, S. G. and Schneider, J.B., "Potential Applications of Knowledge-Based Expert Systems in Transportation Planning and Engineering," Transportation Research Record 1076, Dec. 1986, pp. 43-50.

2. Wandsnider, LuAnn, Laurence Spear, Steven Flint, and Robert Anderson, "Transportation Applications of Computer Mapping in New Mexico," Division of Government Research Institute for Applied Research Services, University of New Mexico, unpublished paper, 1986.

3. Moellering, Harold, "The Automatic Mapping of Traffic Crashes," **Surveying Mapping**, Dec. 1973, pp. 467-76.

4. "Traffic Accident Spot Maps," National Safety Council Bulletin T-075-80, (no author cited)

5. Baerwald, Jon ed., **Traffic Engineering Handbook**, 3rd ed., Institute of Traffic Engineers, 1965, pp. 230-5.

6. Box, Paul C., "Preparation of Collision Diagrams," **Public Safety Systems**, July/August 1970, pp. 13-16.

7. Baker, J. Stannard and Lynn B. Fricke, **The Traffic-Accident Investigation Manual: At Scene Investigation and Technical Follow-up**, Northwestern University Traffic Institute, Evanston, IL, 1986.

8. Homburger, Wolfgang S., **Transportation and Traffic Engineering Handbook** 2nd ed., Institute of Traffic Engineers, 1982.

9. Hagmann, Marvin L., "A Quick and Simple Way to Prepare Collision Diagrams," **Traffic Engineering**, Dec. 1974, pp. 26-30.

10. Litvin, David M. and Tapan K. Datta, "Automated Collision Diagrams," Transportation Research Record 706, 1979.

ACKNOWLEDGMENTS

Ted Bell, WSDOT District 1

Loretta Lew, WSDOT District 1

Sandy Turner, WSDOT

Brian Limotti, WSDOT

David Trestor, WSDOT PR&PT, for C programming expertise.

George Sackinger, ODOT

Bill Storms, WSDOT MIS