AN ERGONOMICS DESIGN KNOWLEDGE BASED EXPERT SYSTEM

By
Md. Dan Md. Palil

A Doctoral Thesis
submitted in partial fulfilment of the requirements
for the award of
Degree of Doctor of Philosophy
of the Loughborough University of Technology
February 1996

Department of Manufacturing Engineering
Loughborough University of Technology

BEST COPY AVAILABLE

Variable print quality

ABSTRACT

The research scope and objectives are to investigate the use of '*geometric reasoning*' using the knowledge based techniques established for expert systems. An Expert System is integrated within the SAMMIE (System for Aiding Man–Machine Interaction Evaluation) computer man modelling system and used for vehicle interior design. Vehicle design objectives are related to a rule base determined from national and international standards and legislation.

Malaysia is now progressing towards becoming an Industrialised Country by the year 2020. In mid 1985 the Malaysian Motor Industry produced the Proton Saga which has since been exported to other countries. Although the Standards and Industrial Research Institute of Malaysia (SIRIM) is playing an important role in design activities and provision of standardisation information, some standards and legislation for vehicle interior design are not easily available. There is an important and urgent need for standards and legislation to facilitate vehicle design within Malaysia and Internationally.

A literature survey on the relevance of ergonomics design to standards and legislation for vehicle interior design is presented. Knowledge and expertise required for the knowledge base were elicited from various resources; extracted from journals, research publications and standards reports from various international organisations.

The SAMMIE system was used to develop a prototype design model for the vehicle interior and the KES expert system shell was selected to develop the Ergonomics Design Knowledge Based Expert System (EDKBES). EDKBES has a modular structure for ease of software readability, editing and testing, and to readily facilitate further development. The knowledge base is divided into several sections related to the hierarchical structure of vehicle interior design.

The integration of EDKBES within the SAMMIE system is achieved by the establishment of a communication method through external programs, communication files and command files. Knowledge is represented using the production rule system available within KES. Values for the attributes or classes are provided both by the user and inferred within the system, and are used in either backward or forward chaining to reach the goal. Integration of EDKBES within SAMMIE is described and evaluated for ergonomics design using geometric reasoning based on the wealth of standards and legislation, design specifications, and design working practices.

This provision of an expert system within SAMMIE is of assistance to designers and engineers in their design working practice, as a design tool for monitoring, advising and checking the design specifications and rules to facilitate vehicle interior design.

ACKNOWLEDGEMENTS

I would like to acknowledge my appreciation of the expertise and advice offered to me during my research at Loughborough University of Technology. In particular, to Professor Keith Case, who as a supervisor, has provided his advice, thoughtful criticisms and suggestions which have greatly added to the value of the thesis. His immense expertise in the SAMMIE system and experience in computer man modelling systems and ergonomics design have enabled me to comprehend quickly my research work. Thanks also extended to the director of research Professor Burns, the head of the Department of Manufacturing Engineering, Dr. Backhouse and Mr. Ball who have been very helpful.

Special thanks are also extended to Robert Doyle and David Walters for their valuable computer expertise and Martin Freer for advice on many aspects of SAMMIE. I should also like to give thanks for support from all other staff and colleagues in the Department of Manufacturing Engineering, especially Mr. Downham, Mr. Temple and Mr. Zarifi.

The University of Technology Malaysia and the Malaysian Government are acknowledged for their sponsorship of the research.

Lastly, I am very grateful to my wife and children who have been my main source of inspiration and loving support, and who have persevered and offered their warm encouragement and help during my study.

DEDICATION

This thesis is dedicated to my family and my parents,
who have been my main source of encouragement
inspiration and loving support.

Family
Mrs Rokiah Daud,
Miss Rabiatul Ilyana, Miss Raizatul Dalila,
Mr Iliyas, Mr Muhammad Faiz and Mr Muhammad Ikmal

Parents
Mr Haji Md. Palil Haji Kusin, Mrs Esah Baba,
Late father-in-law Haji Daud Ali,
Mother-in-law Mrs Hajjah Hindon Jamin

ABBREVIATIONS

ADAPS	Anthropometric Design Assessment Program System
ADR	Australian Design Rules
AI	Artificial Intelligence
BOEMAN	BOEing MAN model
BSI	British Standards Institution
BUBBLEMAN	BUBBLE MAN model
BUFORD	BUFORD model
CAD	Computer Aided Design
CADKEY	Computer Aided Design Knowledge Engineering
CADPEOPLE	Computer Aided Design PEOPLE model
CAEDS	Computer Aided Ergonomics Design System
CAR	Crew Assessment of Reach model
CEN	European Committee for Standardisation
COMBIMAN	COMputerized BIOmechanical MAN model
COMONS	COmputerized MOvement Notation System
CREW CHIEF	Computer Aided model of an Aircraft
CYBERMAN	CYBERnetic MAN model
EEC	European Economic Community
EECD	European Economic Community Directives.
EDKBES	An Ergonomics Design Knowledge Based Expert System
EDS	Ergonomics of Display Systems for Rover
ErgoSPACE	Ergonomics SPACE
ErgoSHAPE	Ergonomics SHAPE
ES	Expert Systems
FEA	Finite Element Analysis
FMVSS	Federal Motor Vehicle Safety Standard
H–pt	Heel–Point
ISO	International Standards Organisation
KB	Knowledge Base
KBA	Knowledge Based Author
KBS	Knowledge Base System
KES	Knowledge Engineering System
kesp	KES PS parser
kesr	KES PS runtime system
MINTAC	Man Machine INTerACtion Model
SAE	Society of Automotive Engineers
SAMMIE	System for Aiding Man–Machine Interaction Evaluation
SgRP	Seating Reference Point
TADAPS	Twente Anthropometric Design Assessment Program System
TEMPUS	Articulated Figure Positioning
2D	two–dimensional
3D	three–dimensional

CONTENTS

CHAPTER 8
Testing and Validation

CHAPTER 9
Conclusions, Contribution of Knowledge and Future Work

Appendices

CHAPTER 1

INTRODUCTION

1.1 Background

"Vision 2020" is a national plan to make Malaysia a more fully developed nation by the year 2020 (Mohamed, 1991). Malaysia is now progressing towards becoming an Industrialised Country within the South East Asia region. It has established agriculturally based industries but is moving rapidly towards more serious industrialisation programmes with emphasis being given to the manufacture of all kinds of products. This includes the National Automotive Industry (PROTON) which led to the development and production of the nation's first car, project M–1 Proton Saga in mid 1985. In May 1985 the Proton Saga was first marketed and has since been exported to other countries. A second national car project M–2 under the Sixth Malaysia Plan from 1991–1995, will be in production and available to the market at the end of 1994 (Ibrahim, 1993). In September 1994, the second national car, Produa Kancil with a three–cylinder 660cc engine, already in production and being marketed, will be exported to other countries. Transportation in Malaysia is increasing at 10.5 per cent per annum, much faster than the growth of the national economy.

The Sixth Malaysia Plan (SMP) is a five year plan covering the period 1991–1995, which represents the first phase in the implementation of the Second Outline Perspective Plan (OPP2). The OPP2 is a ten year plan from 1991–2000, which contains the National Development Policy (NDP) and sets the pace towards achieving the status of a fully developed nation in all socio–economic aspects. The NDP is a new policy of basic strategies for restructuring society and the economy. Science and technology are to be used to promote economic growth towards industrialisation, and to achieve this human resources, manpower in industry and education are required. The manufacturing sector is targeted to grow at 10.5 per cent per annum during the OPP2 period toward the year 2020. Hence, the national policy is to strengthen the manufacturing base and diversify the exports of manufactured products such as automobiles, transport equipment and electronics. Automotive exports will face the competitiveness of the world market, and Proton Saga sales statistics produced by the National Automotive Distributor (EON), show that from 1986 to 30 May, 1991, 32,122 units of Proton Saga have been exported to 11 countries, with the United Kingdom taking 26,653 units (82.9 %) and Singapore second at 2,533 units (7.8 %) (EON, 1991). EON estimates sales of the Proton Saga national car to reach 120,000 units in 1995 from 109,006 units in 1994, 94,000 units in 1993, and 65,000 units in 1992 (Business Times, 1994).

The National Product Design Centre (NPDC) under the Standards and Industrial Research Institute of Malaysia (SIRIM) is playing an important role in the increase in design activities and design awareness, quality and standardisation information. It also provides consultancy services in product design, prototyping, product development and product manufacturing. However some standards, information, and local legislation for vehicle interior design are not easily available.

The automotive industry is expanding with the two national car projects and will require more human resources with skilled and experienced engineers, designers, and ergonomists in research and development (R & D). Also as a consequence of the two projects, enhanced R & D in developing and manufacturing a new design of car for local and international markets is available. In designing new vehicles to meet local and international standards, the engineer and designer require access to ergonomics design specifications, standards and legislation as they relate to vehicle interior design. The National Automotive Industry still does not have any information or standards on ergonomics such as Malaysian Anthropometric Data, and standards and legislation to facilitate vehicle design are not available (Tan, 1986, Yong, 1990 and Tharumagnanam, 1994).

In a report of Council for the Coordination and Transfer of Industrial Technology on a National Plan of Action of Industrialisation Master Plan (IMP) Yong stated that "design engineering, legislation, standardisation, and quality control facilities have not been developed adequately" (Yong, 1990). The report also proposed the setting up of a Malaysian Institute of Design and Innovation.

At present around 2,000 standards for various goods have been developed by SIRIM. Malaysia needs about 8,000 standards in next 15 years on the road to 2020. " Industries should start developing standards for themselves under SIRIM's guidance if the goal of reaching developed nation status is to be achieved " stated the Director General of SIRIM (Ali, 1995), and he also urged research institutions and universities to join forces to develop new standards.

1.2 The Research Objectives

The research scope and objectives are:

1) To investigate the use of 'geometric reasoning' using the knowledge based techniques established for Expert Systems.

(Geometric reasoning concerns the application of computer techniques to spatial problems and is discussed in chapter 4).

2) An integration of Expert System knowledge based techniques within the SAMMIE computer aided ergonomics design system for vehicle interior design.

3) To relate vehicle design objectives to a rule base determined from national and international standards and legislation.

1.3 The Research Area

The integration of an expert system within the SAMMIE computer aided ergonomics design man modelling system is investigated through an application in the interior human packaging of cars. This is an area where there is a wealth of legislation, standards, information, and design working practice to be related to the geometric aspects of design.

This new departure of an expert system in SAMMIE is useful to the designer and engineers in assistance in their design working practice, and is used for monitoring, advising and checking of design specifications.

1.4 The Structure of the Thesis

Chapter One briefly described the importance and urgent need for standards, information and legislation to facilitate vehicle design within Malaysia and Internationally.

Chapter Two studies the literature relevant to ergonomics design standards and legislation for vehicle interior design. The principles of standards and legislation in relation to ergonomics design for vehicle interiors are developed.

Chapter Three is a literature survey on man modelling CAD systems. A brief history of the development of computer–aided design (CAD) and man modelling CAD systems is given. The CAD functions of geometric modelling are described together with comparisons between existing man modelling CAD systems. The application of expert systems in CAD and man modelling systems are analysed for the proposed application of expert systems in the SAMMIE computer aided ergonomics design system.

Expert systems tools and applications, and a brief history of the development of artificial intelligence and expert systems are studied in Chapter Four. The definition, features, and a comparison between conventional computer programs and expert systems is given. The basic principles and structure of expert systems consisting of a knowledge base, an inference mechanism/engine and the user interface are described. Knowledge acquisition, and knowledge representation consisting of the formal logic, semantic networks, production rules and frame structures are described. The application of an expert system for the SAMMIE system is described.

The methodology is presented in two sections within chapter five; the first deals with the selection of the SAMMIE computer aided ergonomics design system for designing and constructing three-dimensional prototype models of vehicle interior workplaces. The second considers the selection of the KES expert system for building the knowledge based system. Building the knowledge base consists of knowledge acquisition and domain knowledge (see chapter 2).

Chapter Six concerns the development of production rules and actions based on knowledge extracted from chapter two.

The implementation of the Ergonomics Design Knowledge Based Expert System (EDKBES) and the methods of integration with the SAMMIE computer man modelling system are described in chapter seven. The KES expert system can exchange information (communicate) with the SAMMIE system by using external programs. The external program is a simple method of communicating through read and write communication files and produces the datafile used for "geometric reasoning".

The testing and validation of the knowledge base and the integration of the expert system with the SAMMIE system is described in chapter eight. The integration between the EDKBES and SAMMIE is carried out to achieve the objectives of the research work.

The international standards and legislation, rules and regulations, design working practice for vehicle interior design collected for this research are detailed in chapter eight.

Chapter Nine contains the conclusions, contribution to knowledge and suggestions for further work. An overall discussion of this research is provided and specific and general conclusions are drawn. Further work is detailed so that limitations of the expert system for SAMMIE as currently developed can be overcome, and for the extension of the system to a broader domain of knowledge.

CHAPTER 2

LITERATURE SURVEY: STANDARDS AND LEGISLATION FOR INTERIOR VEHICLE DESIGN AND RELEVANCE TO ERGONOMICS DESIGN

2.1 Introduction

This chapter introduces the background and definition of ergonomics and anthropometry. Standards and legislation for vehicle interior design and their relevance to ergonomics design are described and discussed in relation to their importance in the context of the design process. The application of anthropometry for vehicle interior design is discussed and the data sources are extracted from journals, research publications and standards reports from various international standards organisations such as the British Standards Institution (BSI), the International Standards Organisation (ISO), the Society of Automotive Engineers (SAE), Australian Design Rules (ADR), European Committee for Standardization (CEN), European Economic Community Directives (EECD), Ergonomics of Display Systems for Austin Rover (EDSAR), and the Federal Motor Vehicle Safety Standards (FMVSS).

Standards such as those produced by BSI and SAE are not necessarily enforceable in law but are frequently used as the basis for legislation. For example SAE recommendations for rear view mirrors were defined in SAE J834a (1967) and subsequently adopted in law by the European Economics Community as 71/127/EEC (1971).

The structure of the vehicle interior can be divided into a hierarchy of four sections; the seats, dashboard, primary controls and mirrors which will be discussed in detail with relation to ergonomics design principles and standards and legislation.

Finally, the application of expert systems in ergonomics design is introduced.

2.2 Background of Ergonomics Design

The history of ergonomics and human factors started back in early civilization where humans first needed to use simple tools and utensils. Since the late 1800s and early 1900s industrialisation has increased the significance of the area particularly through expansion in the complexity of machines and working environments. Frank and Lillian Gilbreth began their work in motion study and shop management in industry and pioneered work on the study of skilled performance and fatigue, and the design of workstations and equipment for the handicapped. Their analysis of hospital surgical teams, for example, resulted in a procedure still used today: a surgeon obtains an instrument by calling for it and extending his or her hand to a nurse who places the instrument in the proper orientation. Despite the early contributions of people such as the Gilbreths, the idea of adapting equipment and procedures to people was not immediately exploited (McCormick and Saunders, 1993).

The ergonomics profession was born during the period after the second world war at a meeting held on 12th July 1949. At a later meeting on 16 February 1950, the term *ergonomics* was adopted and the discipline was established (Oborne, 1995). In 1949 the Ergonomics Research Society, (Kimberlie, 1987) was formed in Britain, and in 1957 human factors started in the United States. In 1959 the International Ergonomics Association was formed to link several ergonomics and human factors societies in various countries around the world (McCormick and Saunders, 1993).

From 1960 to 1980 there was rapid growth in the development of ergonomics and human factors. Human factors in the United States was concentrated in the military and industrial application areas, and with the race for manned space flight, ergonomics and human factors assumed a new importance. Also, during this period, ergonomics and human factors in the Europe and United States expanded beyond military and space applications (McCormick and Saunders, 1993).

Today, ergonomics and human factors groups can be found in many companies and institutions including those dealing in computers, automobiles and consumer products.

2.3 Definition of Ergonomics Design

The word "Ergonomics" is used in Europe, and was derived from two Greek words, "ergon" mean work and "nomos" mean "Laws", which together mean "the laws of work" (Galer, 1987). In the United States of America ergonomics is known as "Human Factors" or "Human Factors Engineering".

Pheasant (1986) defined ergonomics as *"the scientific study of human beings in relation to their working environments"*, and further that *"Ergonomics is the application of scientific information about human beings (and scientific methods of applying such information) to the problems of design"*. Pheasant considered that the application of ergonomics is the process of making or changing things for the better whether this is design of a physical object, a working method, an environment or a system.

The physical environment refers to the surrounding environment of human workplaces such as vehicle interiors and equipment including seats, dashboard, mirrors, primary and secondary displays, and primary and secondary controls.

Anthropometry is the branch of ergonomics concerned with the measurement of the human body. Its historical antecedents may be traced back, through the work of Renaissance artists and authors, to classical times. The four books of *Human Proportion* Albrecht Durer (1471–1528) are the beginning of scientific anthropometry. Two thousand years ago anthropometry and design were considered to be related – but the reasoning underlying the relationship is unappealing to the twentieth–century mind. In the present century, the French architect Le Corbusier, (1887–1965), wrote *The Modular–A Harmonious Measure to the Human Scale Universally Applicable to Architecture and Mechanics*, (Le Corbusier, 1961), an obscure work considered by many to be profound. The theories of human proportion have been used from artistic historical standpoints by Pheasant, (1986).

"Ergonomics is the scientific foundation, both in terms of data and methodolgy, for a user–centred approach to design", Pheasant (1986) wrote that the design process is a

unique creative act and ergonomics is also similarly unique. Ergonomics is not only the collection of data; more importantly it provides a way of looking at the design process. Its viewpoint may be characterized as "*from the human user outward*", or a "user-centred" approach to design. A well designed object should be structurally stable, functionally appropriate (e.g., comfort, convenient and safety) and aesthetically pleasing.

Research work by Jiiptner (1990) provides recent anthropometric data and suggests basic considerations to ensure an adequate driver's position. Task analysis and task design in car-driving are presented in the context of reducing traffic accidents by the application of information technology. Remarks are made on different aspects of traffic and transport with respect to the human being, network considerations and scenarios for problem solving in attempts to develop an outline for automobiles of the future.

2.4 Design Principles, Standards and Legislation for Vehicle Interior Design

A variety of vehicles are used today for personal transportation including personal cars (sports cars), family cars and personal business cars. In designing a vehicle an important objective is to create an effective 'occupant envelope/package'. Standard procedures in vehicle design are required to establish the interior workplace including seats, dashboard, displays, and controls, in a manner that is consistent with driver and passenger safety, comfort, convenience and accommodation.

Vehicle occupant packaging design relies on anthropometric data that has been developed over many years by research and practical application (e.g. the studies of Hammond and Roe, (1972), Hammond et al, (1975) and Roe, (1972) which form the basis of SAE J287–Driver Hand Control Reach). Research carried out in the laboratory and real test evaluations on the road has defined locations for the driver's eye and head, hand reach, preferred seat positions, and other workplace related measures. Many of the research studies have involved hundreds of anthropometric measures and have resulted in standardised methods, templates and computer aided design procedures that are used to develop occupant packaging for a combined population of male and female vehicle users. The relevant standards and legislation for vehicle interior design will be described later in this chapter and the application of anthropometry for occupant packaging is described in the next section.

2.5 The Application of Anthropometry for Vehicle Interior Design

Anthropometry is the measurement of human body dimensions where:–

1) Static anthropometry is concerned with the measurement of human subjects in rigid, standardised positions.

2) Dynamic anthropometry is concerned with the measurement of human subjects at work or in motion, e.g. functional arm reach is dependent on the range of movement at the shoulder, elbow, wrist and fingers as well as the dimensions of the limbs.

Anthropometric data for vehicle interior design is applied:–

1) In designing equipment or the workplace where body measurements are major variables, e.g. occupant space, headroom and seat dimensions (Static anthropometry).

2) In establishing information such as:–

i) control and pedal location using reach envelopes for hands and feet

ii) dashboard and display panel location and interior, and exterior mirror visibility using eye–point envelopes

iii) design of head restraints and seat belts using data concerning the arcs described by various parts of the body under acceleration or deceleration (Dynamic anthropometry).

A number of different anthropometric data sources are available. Various human data have been extensively compiled on military populations, public health service workers, etc and supplied through standards such as, British Standards (BS), European Committee for Standardization (CEN), European Economic Community Directives (EECD), Federal Motor Vehicle Safety Standards (FMVSS), International Standards Organisation (ISO), and the Society of Automotive Engineers (SAE), Australian Design Rules (ADR).

2.6 Design Principles for Vehicle Interior Design

ISO 6385, (1981) and SAE J1100, (1990) describe the ergonomics design principles of the interior of vehicles as a combination of human factors, equipment, and activities. To accommodate this complex situation it is necessary to establish a clear information structure that can be used by geometric modelling system and the expert system. Hence a hierarchical structure consisting of four sections; seats, dashboard, primary controls and mirrors has been established (figure 2.1).

Hierarchical Structure of the Vehicle Interior

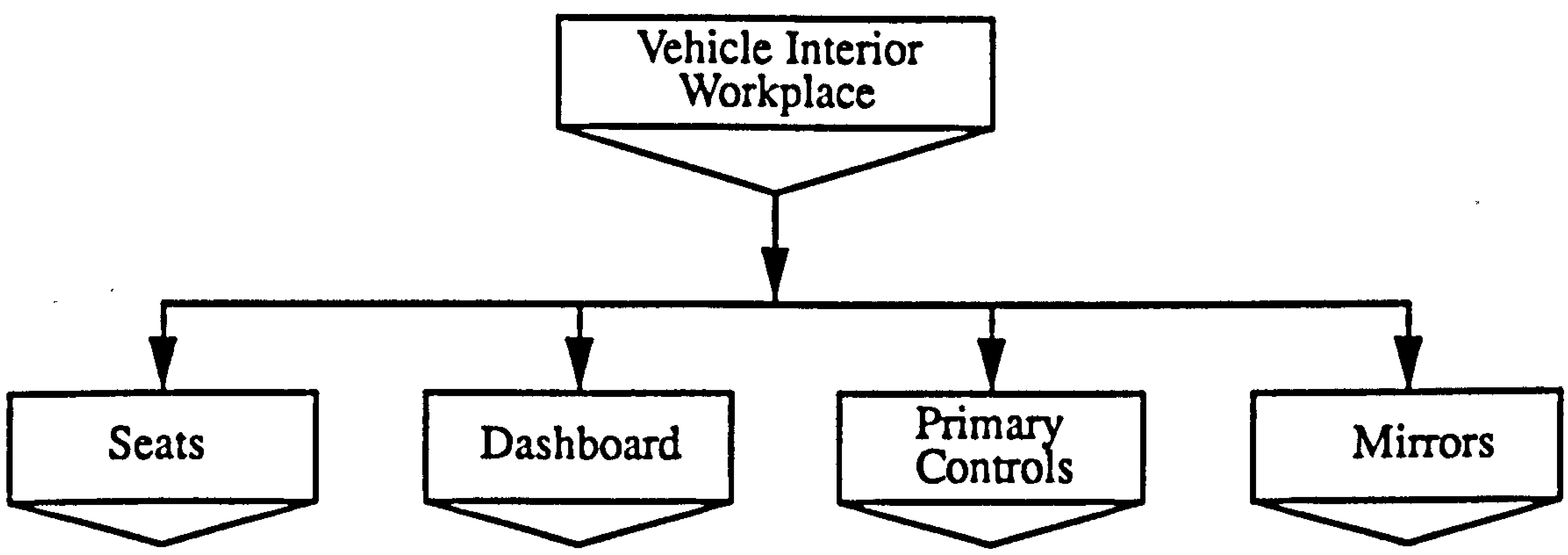


Figure 2.1. Hierarchical structure of the vehicle interior

2.6.1 Design Principles in Relation to Anthropometry

The principal aspects of designing the vehicle interior consist of the dashboard (primary and secondary displays) and the primary controls (gear lever, brake lever, steering wheel, and pedals) and will need take into account constraints imposed by the body dimensions and geometric aspects of the design.

1) The seating height needs to be suited to the body dimensions of the driver and to the kind of driving performed. The seat, dashboard, and primary controls should be designed as a unit to achieve the preferred body posture, force, comfort, and movement of the limbs.

2) The seating arrangements need to be adjustable to the anthropometric and anatomic features of individuals within the defined user population.

3) Sufficient interior workspace should be provided for body motion, particularly for the head, hands, arms, legs and feet.

4) The dashboard layout for the displays and controls, should be within ergonomics range.

5) The brake lever, gear lever, steering wheel and control pedals should suit the functional anatomy of the hand and leg.

2.6.2 Design Principles for Vehicle Seats

Vehicle seats can be divided into two categories–performance or touring. The performance seat is typically stiffer with more contouring (e.g. a bucket type design) with additional adjustable features such as lateral cushion and back bolsters. Touring seats tend to be more comfortable, softer and place greater emphasis on comfort and safety. In designing the seats there is a need to suit styling to the vehicle purpose and function. The geometric features of seat design can be divided into accommodation and comfort requirements (figure 2.2).

1) Accommodation – refers to seat size and adjustment for horizontal or vertical distance from controls, height and back angle.

2) Comfort – refers to stiffness, contour, climate, and vehicle features that promote occupant comfort.

The design principles have the objective of avoiding unnecessary or excessive strain in muscles, joints and ligaments. Body posture, strength exertion and body movement should be in harmony with each other (ISO 6385, 1981):–

1) Seating comfort for driver or passenger.

a) The driver should be comfortable whilst sitting and performing driving tasks, and not be subject to driving fatigue from prolonged static muscular tension.

b) The passenger should be comfortable whilst sitting and not be subject to fatigue from prolonged static muscular tension.

2) Physical capacities/strength of driver or passenger

a) Demands on the driver's strength should be compatible with the physical capacities of hand and leg force.

3) Body movement of driver

a) Driver's task performance; visibility through adjustable steering wheels using eye and head movements.

b) Driver execution and sequencing of movements should be facilitated by primary displays on the dashboard.

Hierarchical Structure of Seating

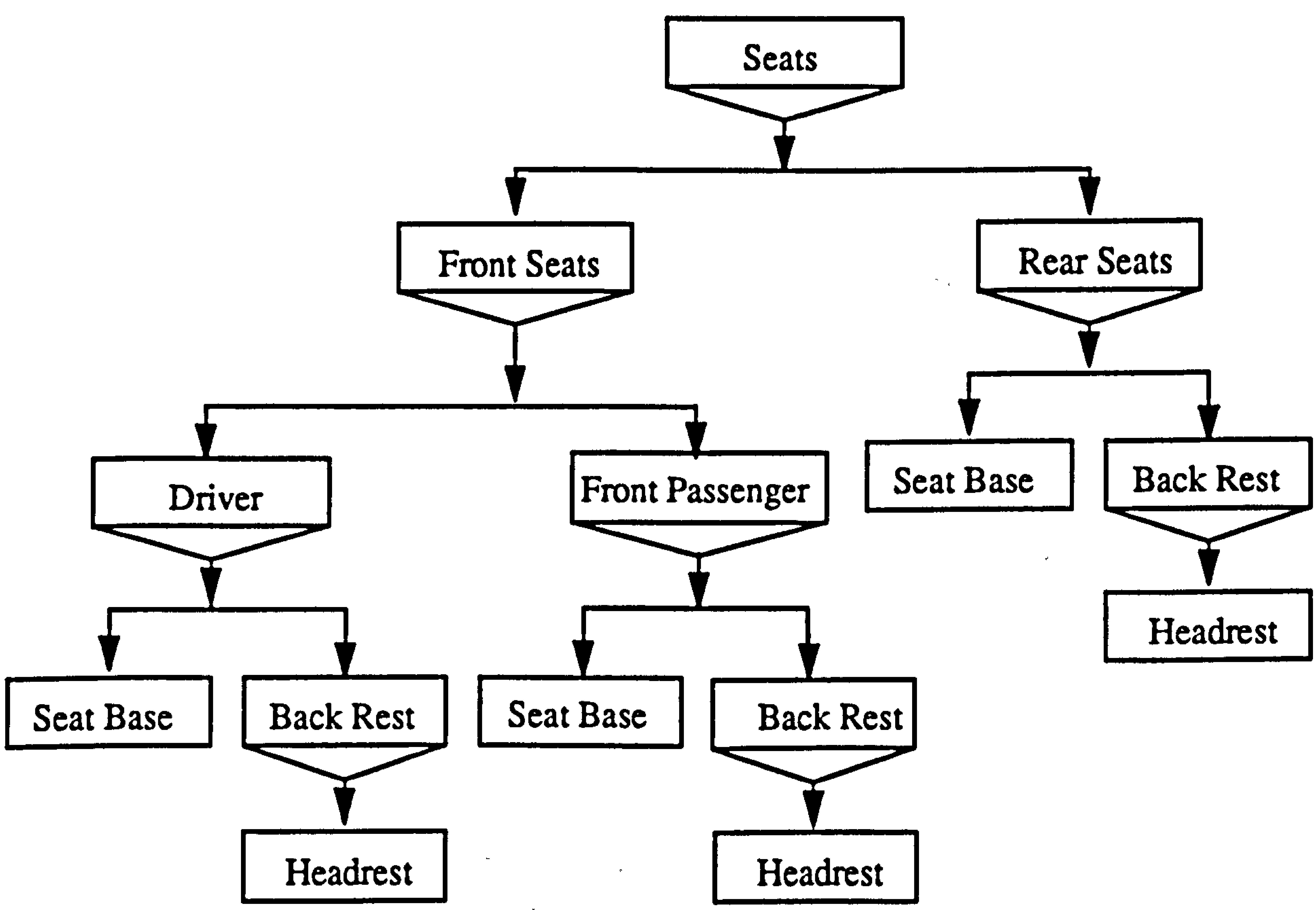


Figure 2.2 Hierarchical structure of seating.

2.6.2.1 Standards and Legislation for Vehicle Seats

Standards and legislation for vehicle seats relates to the three types of different occupant viz: driver, front–seat passenger, and rear–seat passenger. Design principles for driver seats is with reference to selected seat positions as defined in SAE J1517, (1990), SAE J826, (1990), SAE J1100, (1990) and Roe, (1993). Four design principles for the driver's seat are (figures 2.3 and 2.4):–

1) The seat should position the driver with unobstructed vision and within reach of all vehicle controls, displays and panels;

2) The seat must accommodate the driver's size and shape;

3) The seat should be comfortable for extended periods;

4) The seat should provide a safe zone for the driver in an accident (SAE J1517, 1990).

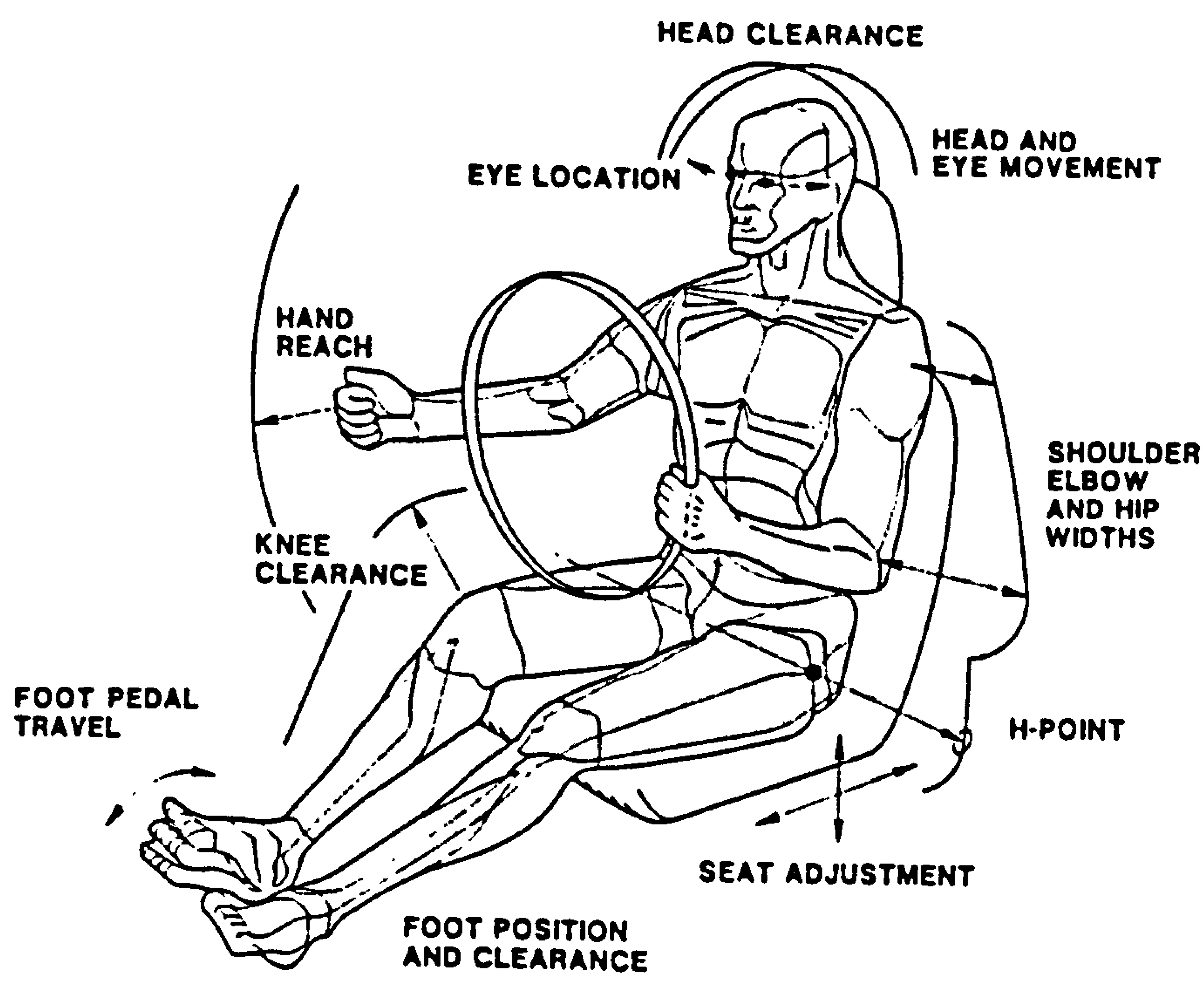


Figure 2.3 SAE J1100 defines important aspects of seat design
(Source: SAE J1100, 1990)

Appendix 1, details the ranges of significant dimensions extracted from BS AU 179, (1981), ISO 4131, (1979), SAE J1100, (1990), Roebuck, (1975) and Roe, (1993) in particular:–

a) The driver adjustable seat positions horizontal distance from SgRP (H–point), front, to accelerator pedal heel point.

b) The driver seat height (H30) ranges between 127 and 177 mm, defined as the vertical distance from the driver's Seating Reference Point – SgRP (H-point) to the floor at the accelerator pedal (Accelerator heel point) (Roe, 1993).

Design criteria for passenger seats in the front and rear are defined with reference to selected seat positions as defined in SAE J1517, (1990) and passenger head positions are defined in SAE J1052, (1990).

1) The seat requires comfortable supporting surfaces for a variety of postures unconstrained by vehicle operation.

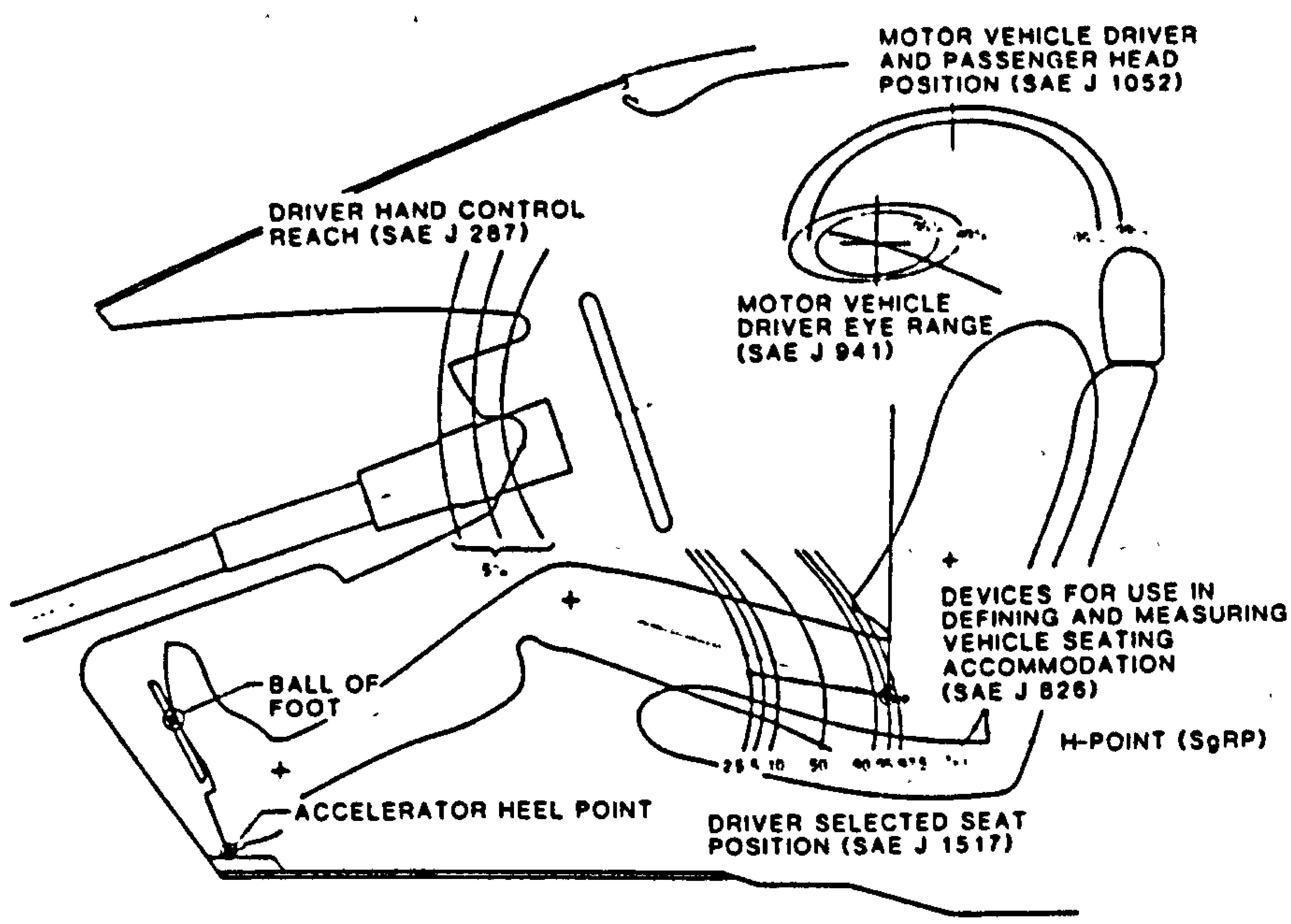


Figure 2.4. Standards for accommodating driver's tasks
(Source: SAE J1100, 1990)

2) The front passenger seat height (refer H30), the vertical dimension from front passenger's Seating Reference Point (SgRP) H-point to floor, should range between 127 and 177 mm.

3) The rear passenger seat height (H31), the vertical dimension from the rear passenger's Seating Reference Point (SgRP) H-point to floor, should range between 200 and 265 mm.

Appendix 1, establishes the ranges of significant dimensions for passenger seats extracted from SAE J1100, (1990) and Roe, (1993).

2.6.3 Design Principles for Dashboards

The design principles for the dashboard can be represented in a hierarchical structure consisting of two sections; primary displays, and secondary controls (figure 2.5). The principles for dashboard design are intended to create a layout that is compatible with the characteristics of human perception and vision as follows:-

1) Dashboard – primary and secondary displays

The dashboard runs the entire width of the interior front of the vehicle. The standard orientation is at an angle of 15 degrees away from the driver. This ensures that the displays are presented perpendicular to the preferred line of sight to minimise distortion (Woodson, 1964).

i) The information on the display panel, should be compatible with the change of the primary source of the information (e.g. signal and direction indicators).

ii) The nature and design of signals and displays should ensure unambiguous perception taking account of size, shape, contrast and intensity. This applies especially to danger or maximum signal indicators.

iii) The nature and number of signals and displays should be compatible with the characteristics of the information.

iv) Clear identification of information is necessary where displays are numerous. Arrangement and grouping should be by functional process.

2) Dashboard – secondary controls

The key ignition switch – should be located next to the steering wheel on the driver's right side (for right hand drive cars), be easy to operate and located for easy access and operation (SAE J1138, 1977).

The turn signal, headlamp and wash–wipe controls should be a group of three switches on the stalk controls, and mounted on the left of the steering column (SAE J1138, 1977).

Hierarchical Structure of the Dashboard

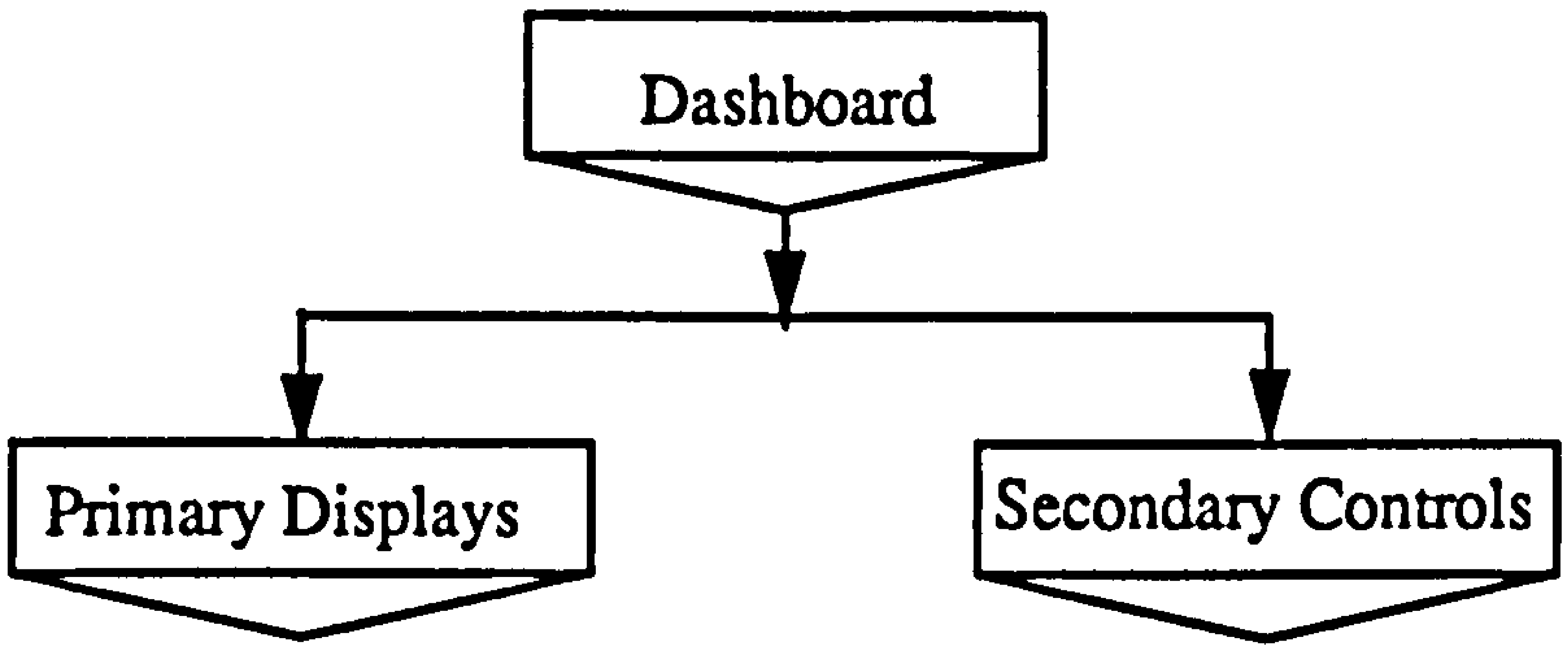


Figure 2.5 Hierarchical structure of the dashboard

2.6.3.1 Standards and Legislation for Dashboard

There are many types of displays, controls, instruments and indicators which are of different shapes and sizes and serve a variety of functions and purposes. Also, there are ergonomics standards and legislative requirements in the form of constraints on the shape, size or position of the displays, controls, instruments and indicators. However, other related design constraints, such as aesthetics, styling, manufacturability, economics and marketing would need to be considered.

Standards and legislation for primary and secondary displays are defined in ISO 4040, (1983) and BS AU 199, (1984).

Primary displays are instruments that give essential information to the driver and consist of speedometer (odometer, trip–odometer and oil pressure meter), fuel indicator, temperature indicator, door–open indicator, turn–signal indicators and headlamp (main beam/flasher indicator).

1) Standards and legislation for the selection, design and layout of primary displays are intended to meet the objective of compatibility with the characteristics of human perception as follows (ISO 4040, 1983 and BS AU 199, 1984):–

i) The nature and number of signals and displays shall be compatible with the characteristics of the information.

ii) Clear identification of information is necessary where primary displays are numerous. The layout should be simple, spacious and arranged so as to promote clear and rapid orientation.

iii) Primary displays should be designed for clear visibility and good visual perception. Account shall be taken, for instance, of intensity, shape, size and contrast.

iv) Rate and direction of change of display information should be compatible with rate and direction of change of the primary source of that information.

v) The function of primary displays shall be identifiable to avoid confusion.

vi) Displays and control reach/movements, equipment response, and display information should be mutually compatible.

vii) Where controls are numerous there is a need to ensure safe, unambiguous and quick operation. The displays and signals should be grouped according to their functions.

2) ISO and BSI define various zones for the location of primary displays (ISO 4040, 1983, BS AU 199, 1984).

i) Zone one is located on the right side of the display panel and should contain the tachometer, economy–mpg meter, fuel indicator and five indicators which should be square in shape and of minimum dimensions 18x18 mm (parking–brake indicator, hazard–warning indicator, battery condition indicator, supplementary restraint system (SRS–D), and seat–belt indicator).

ii) Zone two is located on the centre of the display panel and should contain door–open indicators, turn–signals, and headlamp indicator – upper/lower beam. The remaining parts of the display area shall also be visible with head movement as required.

iii) Zone three is located on the left side of the display panel and should contain the speedometer, odometer, trip–odometer, oil pressure meter, temperature gauge, and five indicators (for brake-failure, service-warning, heater controls, supplementary restraint system (SRS-P) and fog-light) which should be square and of minimum dimensions 18x18 mm.

2.6.3.2 Standards and Legislation for Display Information

Standards for display information are specified in SAE SP-576 Ergonomics Aspects of Electronic Instrumentation (SAE SP-576, 1984 and Galer, 1985). Display information is intended to make the driver aware of the current status during the operation of controls, and is of four basic types:-

1) **Warning** – Warning information is very important to the safe running of the vehicle. Red is used as a "warning" indicator, e.g. for brake failure or brake engaged signals.

a) The driver's attention should be attracted to the warning; the significance of the warning must be apparent through the red colour.

2) **Advisory** – Advisory information is very useful to the safe running of the vehicle, and is also used to convey vehicle state information, e.g. headlight main beam "ON", "FASTEN SEAT BELT".

a) The driver's attention should be attracted to the information but it should not distract him from the driving task. This covers a wide range of information devices from simple tell tales and indicators, to trip odometers/computers.

3) **Diagnostic** – Diagnostic information concerns the condition of the vehicle for maintenance purposes, e.g. warning light for battery charging, and "service" indicator.

a) The driver should be able to choose the appropriate opportunity to assimilate or take action on such diagnostics.

4) **Entertainment** – In some vehicles related information is available via the entertainment facilities, e.g. traffic bulletins transmitted by radio.

a) It is necessary to make sure that other audible forms of information presentation are not masked by the entertainment system.

2.6.3.3 Standards and Legislation for Displays

Standards and legislation for displays are defined in SAE SP–576, Ergonomics Aspects of Electronic Instrumentation (SAE SP–576, 1984, and Galer, 1985). Displays are intended to show information to the driver to confirm correct function during the operation of controls.

1) **Analogue Displays** – Analogue displays typically use a needle pointer on a scale to show the value represented. Often these are used to convey qualitative information, and can be enhanced by a red portion of the scale to signify danger. Types of analogue displays include circular dials, linear scales and curvilinear combinations, and typical applications would be the tachometer and fuel gauge. Analogue displays are generally better than small digital equivalents for quick check reading, and for rate of change and direction information.

2) **Discrete Displays** – Discrete displays are also analogue displays but the markings of the scale are discrete rather continuous. An example is an 8–segment or discrete sections fuel gauge, providing quantity information but without the detail or accuracy of scalar displays.

3) **Digital Displays** – With digital displays the information is presented directly as a number. A good example is the odometer. Digital displays are better than analogue displays where precise readings and perfect indications are required.

4) **Alphanumeric Displays** – Alphanumeric displays present information as textual messages in full or abbreviated form e.g. "FASTEN SEAT BELT".

5) **Representational Displays** – Representational displays present information as graphic diagrams or working models, such as the plan drawing of the car used as a door–open indicator. The graphic diagrams enable the user to observe the function of items such as doors, bonnet and boot in relation to the whole, and to locate faults quickly, and can for example be used for vehicle diagnostics.

2.6.4 Design Principles for Display Panel Layout

Design principles for the layout of a display panel have been specified by ISO 4513, (1978), BS AU 176, (1980), SAE SP–576, (1984) and Galer, (1985) and include:–

1) Layout for good visibility

i) Visibility – The driver should be able to see all displays on the panel from the normal driving position.

ii) The plane in which the displays lie should be perpendicular to the line of the sight.

iii) The driver's view of displays should be unobstructed by the steering wheel, parts of his own body, and windscreen reflections.

iv) The distance between displays should be minimised to reduce eye and head movement. However, it is also useful to spatially separate displays to avoid confusion when reading them quickly.

2) Layout for good design and identification

i) All displays should be clearly labelled, easy to find and identify.

ii) Location and separation – Good layout on the panel is one of the best aids to identification. Primary instruments, e.g. speedometer, tachometer and warning indicators should be located in primary display space.

iii) Functional grouping – Displays should be grouped in terms of functional use. This reduces the area over which the driver has to search for a particular display.

iv) Standardised location – If possible standardise the location of displays or functional groups of displays.

v) Associations – Displays should be arranged to be compatible with the controls to which they are related.

2.6.4.1 Standards and Legislation for Display Instruments

Standards and legislation for display instruments have been specified by ISO 4513, (1978), BS AU 176, (1980), SAE SP–576, (1984) and Galer, (1985) and include:–

1) Speedometer (Odometer, Trip–odometer, and Oil pressure meter), Tachometer

i) The speedometer and tachometer should be positioned so as to be fully seen by all drivers, without eye or head movements.

ii) A combination analogue–digital/electromechanical instrument for speedometer and tachometer display provides good ergonomics and functionality.

iii) Confusion between the tachometer and the speedometer can be avoided by differentiating by styling, colour, relative brightness, etc.

iv) The tachometer maximum limits to engine speed should be indicated on the scale in red, yellow or orange.

v) Odometers and trip–odometers of a square shape are recommended and not so large as to interfere with speedometer readings (10 mm minimum character size).

vi) Odometer analogue–digital/electromechanical displays are best placed within the speedometer above the trip odometer to avoid clutter and confusion.

vii) The odometer and trip odometer should be less bright than the speedometer and the control knob should be easy to reach.

viii) Trip–odometer analogue–digital displays are best placed within the speedometer below the odometer where they can be easily read by the driver.

1.1 Analogue Speedometer and Tachometer

i) Analogue displays are not as good as digital displays for rate of change information.

ii) The standard shape of circular or semi–circular is recommended for ease of reading.

iii) Standard scale markings for speedometers should use a conventional progression system of 10, 20, 30,.....with detail scale markings at 0, 5, 10, 15, 20. Where appropriate minor markers should be used for individual numbers.

iv) Standard scale markings for tachometers should use a conventional progression system of 1, 2, 3,..... and where appropriate minor markers are used for individual numbers.

v) Needle pointers of whatever form must line up with scale markings and be positioned on the scale to be easily read and so as not to distract the driver.

vi) The full scale should be available to the driver with the current value being indicated by a pointer.

1.2 Digital Speedometer and Tachometer

i) Digital displays are read more quickly and accurately than analogue displays. for information acquisition. This permits drivers to take their eyes off the road for shorter periods.

ii) Digital display values must remain visible long enough to be read accurately, (approximately 500 – 1000 m. seconds).

iii) Characters should be upright rather than slanted and their height should be 15–20 mm.

2 Fuel Indicator

i) The fuel indicator should use an analogue or digital display.

ii) Qualitative markings are required for fuel indicators. Further scale markings such as "E" for empty, 1/2 for half full and "F" for full should be provided.

iii) The low fuel level warning should be coloured red, orange or yellow.

2.6.4.2 Standards and Legislation for Displays: Symbols and Colour

ISO 2575, (1982) and BS AU 143c, (1984) define standards for symbols and colours to be used for controls, indicators and tell–tales for road vehicles.

1. Symbols should be recognizable by the driver from his seated position.

2. Standard colours used on instruments of optical tell-tales have the following meanings:-

a) Red: Danger to persons or very serious damage to equipment immediate or imminent. Also used for "hot" in climate control systems or engine temperature indicators.

b) Yellow: Caution, vehicle system malfunction, danger to vehicle likely, or other conditions which may produce a hazard in the longer term.

c) Green: Safe, normal operation of the vehicle system.

d) Blue: Driving full beam tell-tales only. Also used for "cold" in climate control systems or temperature indicators.

e) White: Other conditions where none of the above colours are appropriate.

2.6.5 Design Principles for Primary Controls

The primary controls consist of the hand levers, steering wheel and foot pedals. Hand levers consist of brake and gear levers (manual or automatic), and the pedals consist of accelerator, brake and clutch. Figure 2.6 shows the hierarchical structure of the primary controls.

1) The major design principle for primary controls is to select components and design the layout in such a way that their operation is compatible with the movements of the part of the body by which they are operated. The driving posture is largely determined by the location of the primary controls in relation to the seat, and clearly reach to controls is an important criterion.

The basic anthropometric data necessary for the functional design of primary controls are the dimensions of the operator/driver population and the speed, precision and force which can be exerted in the various hand, arms, foot, and leg positions. The driver's performance in carrying out tasks such as the handling of the vehicle require that skill, accuracy, speed and strength requirements should be taken into account.

Hierarchical Structure of Primary Controls

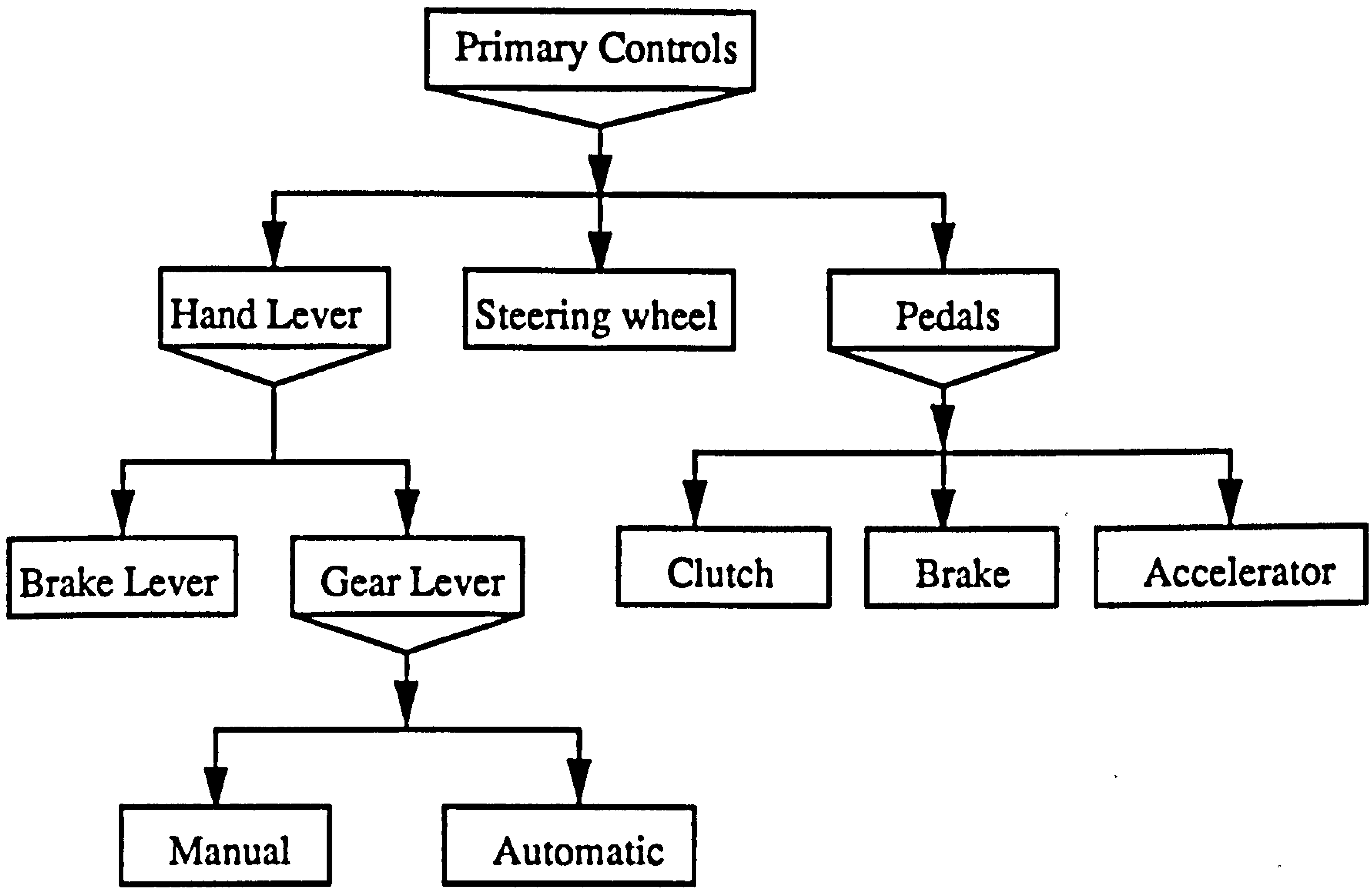


Figure 2.6 Hierarchical structure of primary controls.

i) The type, design and layout of the primary controls should correspond to the control task, taking into account human characteristics, including learned and innate responses.

ii) Control movement, equipment response, and display information should be visibly compatible.

iii) Functions of the controls should be easily identifiable to avoid confusion.

iv) Control layout should ensure safe, unambiguous and quick operation. Similarly, signals should be grouped according to function and in the order in which they are used.

v) Critical controls should be safeguarded against inadvertent operation.

2) Design principles of primary controls.

Design principles for primary controls depend on their suitability for the task to be performed. The first criteria in selecting the type of control are:–

- The function of the control
- The requirements of the task
- The driver's information requirements
- The constraints of the driving workplace including the effects of seat belts.

i) Controls should be located and function in a manner appropriate to the driver or passenger. The position of the display relative to the control and the layout and direction of the display response should be considered.

ii) The direction of movement of the controls and displays should be related to the propose of each control.

iii) Related controls should be functionally grouped to reduce reach movements, and to ease sequential or simultaneous operation. Minimisation of panel space is also an issue.

iv) Controls should be easily identified by location, size, shape, texture, colour, labels, illumination or mode of operation. The primary and hazard/warning controls should be easily identified both visually and by touch.

2.6.6 Design Principles for Primary Controls: Steering Wheel

The design principles for primary controls: visibility through the steering wheel and locations are defined in SAE J941, (1990), SAE J1052, (1990), BS AU 199, (1984), ISO 4040, (1983) and ADR 18/00, (1988).

The definitions and principles for the steering wheel are:–

Steering wheel plane:– The plane passing through the upper surface of the steering wheel rim, as designated by the vehicle manufacturer with the vehicle wheels in the straight ahead position (BS AU 199, 1984).

Steering wheel axis:– A line at right angles to the steering wheel plane, passing through the centre of rotation of the steering wheel rim (BS AU 199, 1984).

The areas visible through the steering wheel can be divided into three zones:–

1) Zones one and three are areas to the left and right respectively:– of the reference plane bounded by the following surfaces (figure 2.7) (BS AU 199, 1984).

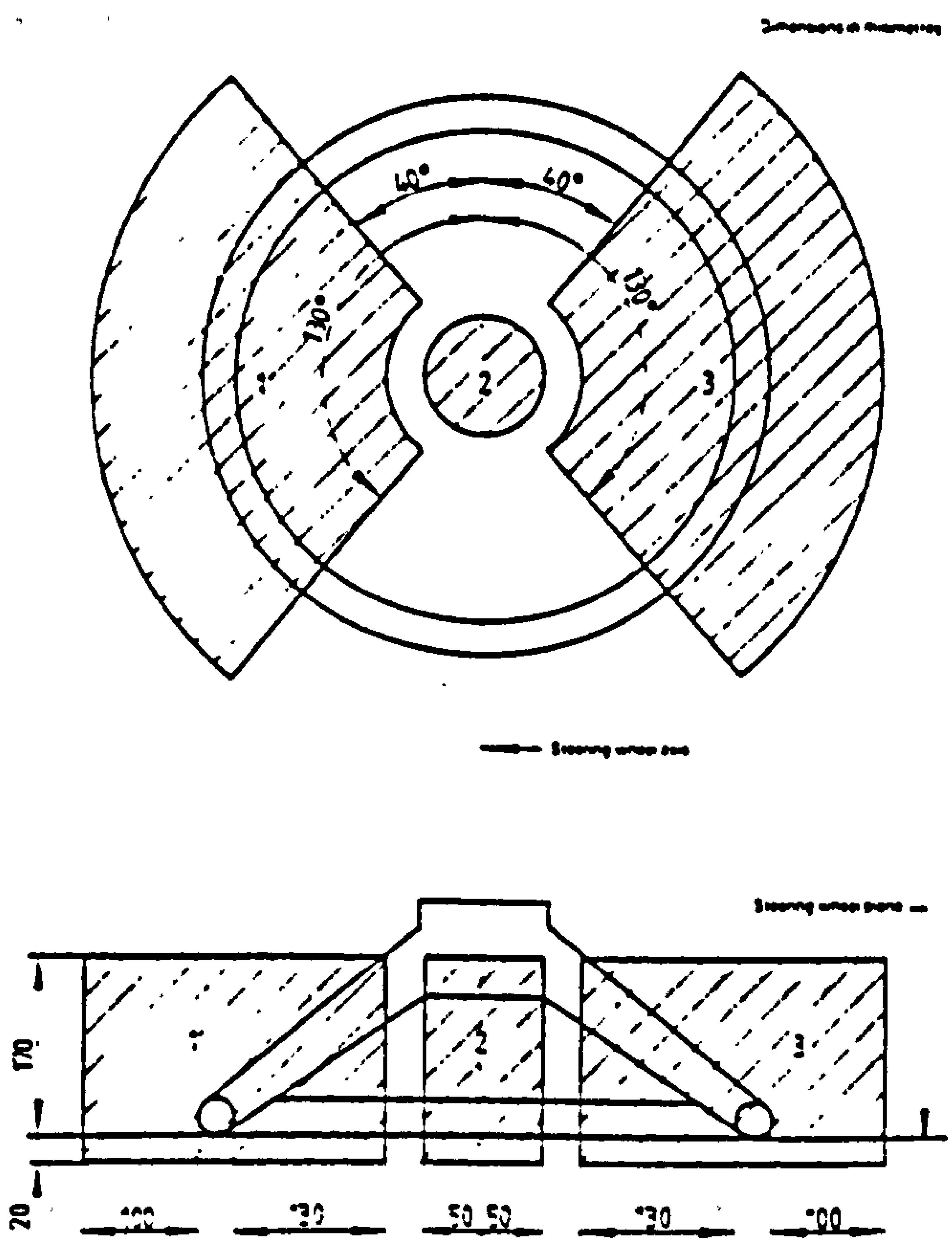


Figure 2.7 Location of zones (Source: BS AU 199, 1984)

i) a plane parallel to the steering wheel plane and 20 mm above it;

ii) a plane parallel to the steering wheel plane and 170 mm below it;

iii) a cylinder which extends 100 mm outside the periphery of the steering wheel rim;

iv) a cylinder which lies 130 mm inside the periphery of the steering wheel rim;

v) two planes which intersect along the steering wheel axis and whose intersections with the steering wheel plane are at 40 and 130 degrees from the reference plane.

2) Zone two:– An area at the centre bounded by the following surfaces.

i) a plane parallel to the steering wheel plane and 20 mm above it;

ii) a plane parallel to the steering wheel plane and 170 mm below it;

iii) a cylinder of 50 mm radius whose axis is on the steering wheel axis.

2.6.6.1 Standards and Legislation for Primary Controls: Steering Wheel

The standards and legislation for steering wheel visibility and location are defined in SAE J985, (1967), SAE J941, (1990), SAE J1052, (1990), BS AU 199, (1984), ISO 4040, (1983) and ADR 18/00, (1988). Established standards and legislation, are described below:–

1) The head contour locator line; is a locus of points used to locate the driver's head position contours in fixed seats with back angles from 5 degrees to 45 degrees.

2) The standard viewing distance or visibility distance from the driver's eye to the primary displays during driving will vary to some small extent due to eye or head movements. The standard viewing distance in cars is about 750 mm. For drivers wearing bifocal lenses, the reading lens is usually focused at 300 mm with the distance lens at infinity.

3) The angle between the line of sight and a perpendicular to the primary display screen is called the viewing angle or acceptable viewing angle. The acceptable viewing angle for legibility of displays is affected by ambient illumination, screen curvature, use of lenses, contrast, resolution and character size. A relatively wide range of acceptable viewing angles that are generally accepted in ergonomics practice are :–

- 15 degrees– comfortable viewing angle.
- 30 degrees– maximum acceptable angle.

Appendix 1, established from the standards and legislation, relates to steering wheel dimensions as defined in ISO 3958, (1977) and SAE J1050, (1990):–

Steering wheel rim diameter for males should be between 300 and 460 mm and 300 mm for females. Resistance to motion of the steering wheel should be between 56 and 90 Newtons. The steering wheel should be angled at 10 to 70 degrees depending on type of vehicle and displacement limited to about 120 degrees with no need to remove hands during turning (Woodson, 1964).

2.6.6.2 Standards and Legislation for Steering Wheel: Visibility

Eye and head movement allowable for vision through the steering wheel and are defined in SAE J985, (1985) (figure 2.8).

1) In the vertical plane, eye movement is comfortable within 15 degrees above or below the horizontal. Although the eye can be rotated up to 45 degrees upward or 65 degrees downward if necessary.

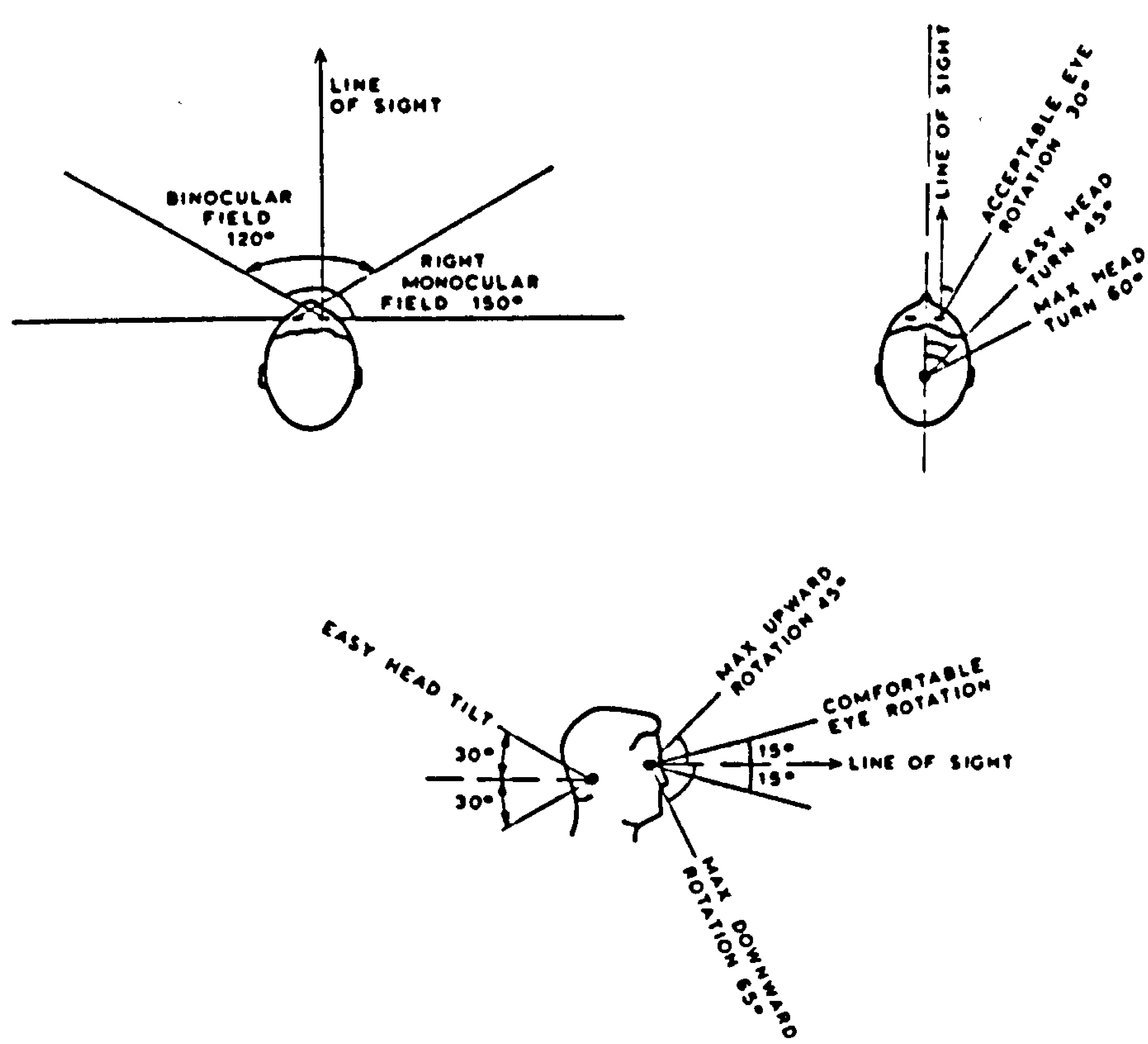


Figure 2.8 Visibility of visual field, eye and head movement
(Source: SAE J985, 1967).

2) In the horizontal plane the binocular field of view extends for some 120 degrees.

3) The eyes generally only turn by about 30 degrees before the head is turned. This can comfortably give a further 45 degrees view to either side.

Visual aspects of vehicle interior design, limits to the visual field, eye movement and head movement are defined in (SAE J985, 1967), and summarised in Appendix 1.

2.6.7 Design Principles for SRS Airbag Systems

Safety and occupant protection has always been a crucial element in the minds of engineers and designers, and many automobile companies have invested heavily in occupant safety.

Supplementary Restraint (SRS) Airbag Systems are a recent innovation in occupant safety. It is estimated that half the cars sold in Europe during 1995 will have airbags fitted. The SRS airbag system is rapidly becoming a standard item of equipment for vehicles. About 90 percentage of all cars made in Japan for the Japanese domestic and USA export markets will have airbags fitted as standard in 1995 (Ford DSE, 199).

Between 1993 and 1994, about 90 percent of all cars sold in North America, were fitted with airbags. Hopefully by 1999 all cars and light commercial vehicles sold in the European region will be required to have both driver and front passenger airbags (Ford DSE, 1994).

By the end of this decade it is expected that there will be standards and legislation for SRS airbags systems in vehicles.

Design principles for SRS airbag systems, safety features, specifications and design practice are described below (Ford DSE, 1994):–

The SRS airbag system, or so called Supplementary Restraint System, is accommodated in the padded boss at the centre of the steering wheel and the driver's seat is provided with anti–submarine ramps together with seat belt grabbers and pretensioners (Restraint System) to provide fast response safety that gives additional protection to the driver's head, face and chest.

The SRS airbag system for the front passenger seat is accommodated in a padded boss at the centre of the dashboard and the seat is similarly provided with anti–submarine ramps and seat belt grabbers and pretensioners.

The design principles for SRS airbag systems are (Ford DSE, 1994):–

1) Firstly, the operational readiness of the Restraint System (airbag) is indicated by the SRS indicator light in the instrument cluster. If the key in the steering lock is turned to position 1 or 2, the indicator light stays on for approximately 4 seconds. Should an impact occur at the steering wheel the airbag will fully inflate the moment a signal is received from the sensor, make contact to protect the driver, then deflate as it absorbs the impact, all in about 50 milliseconds. If the indicator light fails to come on when starting the car or comes on while driving there is a fault in the system. The SRS airbag, however, is not activated by this fault. The airbag is so designed as to be activated only in severe head on collisions. The driver should have fastened his belt as otherwise the airbag cannot provide the envisaged protection.

2) Secondly during the activation of the SRS airbag a small volume of air will be released. Then the seat belt inertia reels are locked and in addition other sensors will have activated clamps which grab and hold the belts to minimise any paying out due to the effect of spooling. In a more severe impact the pretensioners work in conjunction with the grabbers to further enhance seat belt efficiency by pulling the belt buckles downwards to reduce any slack in the diagonal and lap belts.

3) Front and rear seats have anti–submarine ramps so as to reduce sliding forward under the seat belt during impact. In the most severe impact the steering wheel airbag will inflate and deflate as it absorbs the impact.

2.6.7.1 Standards and Legislation for SRS Airbag Systems

Major automobile companies have implemented standards for Supplementary Restraint System (SRS) Airbag Systems (Ford DSE, 1994, and Mercedes–Benz SRS, 1994).

1) The driver's seat and front passenger seat should have an SRS airbag system.

2) Rear passenger seats should have anti–submarine ramps with seat fastener, seat belt grabber and pretensioners,

3) The internal structure including all four doors should incorporate cross-car beams and a safety cell or cage.

4) The standard material for airbags is neoprene lined textile with an internal coating of silicone to protect the inner surface from the gas generated to inflate the airbag (Ford DSE, 1994 and Mercedes SRS, 1994).

5) "SRS-D AIRBAG" should be stamped on the padded boss of the steering wheel and an "SRS-D" indicator light should be provided on the right of the instrument cluster.

6) For the front passenger seat the lettering "SRS-P AIRBAG" is stamped on the dashboard drawer of the front passenger seat and an "SRS-P" indicator light should be provided on the left of the instrument cluster.

2.6.8 Design Principles for Hand Lever Controls

Design principles for hand lever controls for passenger cars are defined in ISO 3958, (1977), ISO 3833, (1977), SAE J287, (1990) and are discussed below (figure 2.9):-

1) Driver hand-reach capability: the maximum reach capability of the driver is defined in a simulated driving situation with the left hand reaching to the gear lever, the right hand reaching to the steering wheel, left foot on the clutch pedal and the right foot on the accelerator pedal.

2) Hand reach task: it should be possible to reach hand controls with the controls held in a three-finger grasp, fingertip, and hand grasp.

3) Control knobs should be reachable with a three-finger grasp.

4) Driver hand brake reach capability: the hand brake is normally located near the floor between the driver's seat and passenger seat. It is operated by reaching down pulling up through an arc. To operate the brake, the driver must reach between the bottom of the handle and the floor (a space of less than 50 millimetres) and pull up.

Hierarchical Structure of Hand Levers

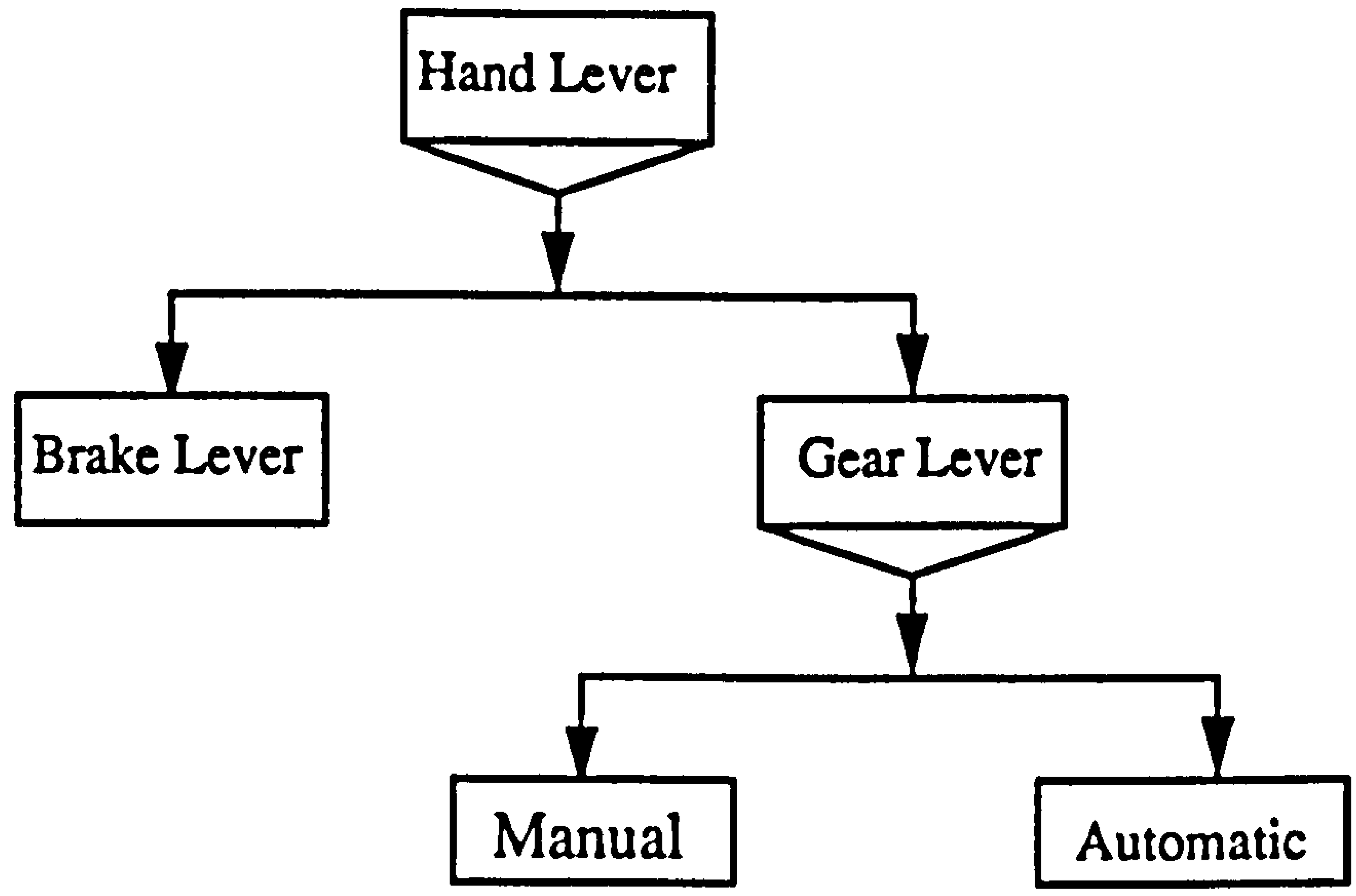


Figure 2.9 Hierarchical structure of primary controls: hand lever controls.

2.6.8.1 Standards and Legislation for Hand Lever Controls

Standards and legislation for hand lever controls, driver hand-reach capability, hand grasp for gear lever and driver hand-brake reach capability are discussed in ISO 3958, (1977), ISO 3833, (1977), SAE J287, (1990), and Dreyfuss, (1960):-

1) The driver should be capable of reaching and grasping the gear lever knob, activating it through an arc of some 15 degrees with the left hand, while the right hand is reaching to the steering wheel and rotating it through an angle of some 20 degrees. At the same time the left foot is on the clutch pedal and the right foot is on the accelerator pedal.

2) Control knob grasp – three-finger grasping reach to a 25 mm. diameter knob should be possible.

3) The driver should be capable of reaching to the hand-brake and pulling it up through an arc of some 30 degrees. The hand brake should have a maximum handrail diameter of 44 mm and a rod diameter between 20 and 38 mm.

2.6.9 Design Principles for Pedal Controls

The design principles for pedals involve the accommodation of large forces and the gross motions and positions of the legs. The driver's seat positioning in relation to the foot pedal controls is dependent on (Roebuck et al, 1975):–

i) The type and size of the driver's seat.

ii) The number, type, size and mode of operation of the pedals.

iii) The relative spatial arrangement of the seat and pedals.

The relationships between driver's seat and the pedals are height and lateral and sagittal distances. Pedals should be located within easy reach of the driver. Adjustability of the driver's seat in height and distance will be required to accommodate operators of different body sizes, especially with different leg lengths.

i) Latitudinal placement of the foot pedals affects the comfort of the operator.

ii) The operator's foot when at rest on the control, should exert between 23 and 45 Newtons force.

iii) If the foot pedal angle is greater than 20 degrees above horizontal then a heel support is required.

iv) The clutch, brake and accelerator pedals should be spaced 50–100 mm apart.

2.6.9.1 Standards and Legislation for Accelerator Pedals

ISO 3958, 1977 defines standards and legislation for accelerator pedals (figure 2.10).

1) The accelerator heel point is the lowest point at the intersection of the driver's heel and the depressed floor covering with the shoe on the unpressed accelerator pedal. The foot angle (L46) should be at a minimum of 87 degrees with the operator heel point at the Seating Reference Point (SgRP). For a vehicle SgRP to heel (H30) dimension greater than 177 mm, the accelerator pedal may be depressed as specified by the manufacturer (Roe, 1993).

Hierarchical Structure of Pedal Controls

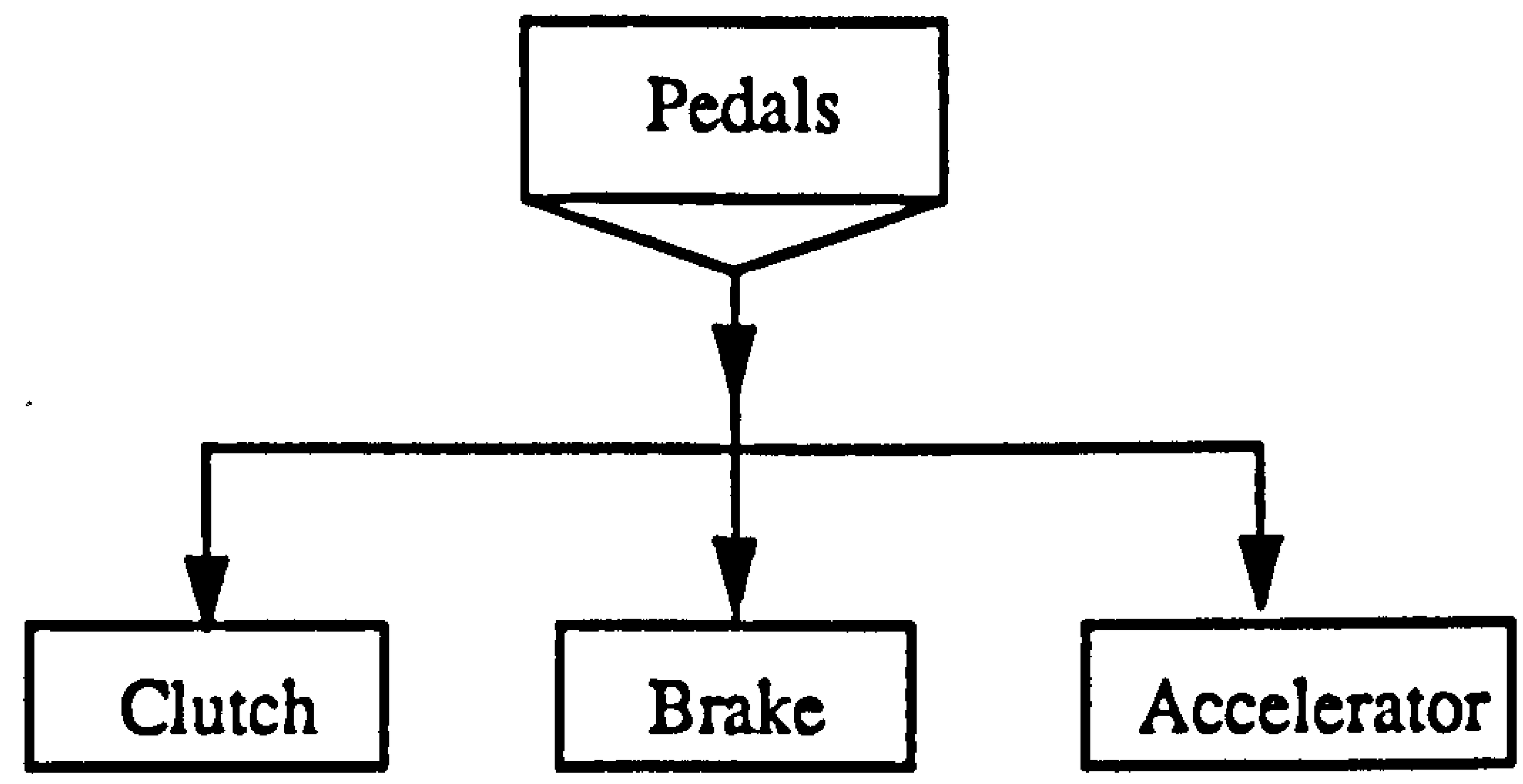


Figure 2.10 Hierarchical structure of primary controls: pedals

2) The foot angle (L46) is the angle between the lower leg centreline (two-dimensional) and the straight line between the ball and heel of the bare foot should be at a minimum of 87 degrees.

3) The accelerator foot plane, is a plane passing through the accelerator heel point and the ball of foot and is normal to the y-plane (side view plane).

2.6.9.2 Standards and Legislation for Brake Pedals

ISO 3958, (1977) and Roe, (1993) defines standards and legislation for brake pedals.

1) The brake pedal hanging at the centre pivot should be on a straight line tangent to the bottom of the driver's shoe parallel to the horizontal plane at a distance of 72 mm from the accelerator heel point.

2) The brake heel point or pedal height should be between 76 and 152 mm, with a travel depth of 51 mm and a maximum force of 34 Newtons. The distance between the brake pedal and the clutch/accelerator pedals should be between 50 and 100 mm.

2.6.9.3 Standards and Legislation for Clutch Pedals

Standards and legislation for clutch pedals are defined in ISO 3958, (1977): –

1) The clutch pedal hanging at the centre pivot should be on a straight line tangential to the bottom of the driver's shoe and parallel to the horizontal plane at a distance of 72 mm from the accelerator heel point.

2) The clutch heel point or pedal height should be between 76 and 152 mm, with a travel of 102 mm and a maximum force of between 180 and 203 Newtons. The distance from clutch pedal to brake pedal should be between 50 and 100 mm.

2.6.10 Design Principles for Mirrors

Standards exist for interior and exterior mirrors (figure 2.11). Exterior mirrors can be for the nearside or offside of the vehicle as described in FMVSS 111, (1990), SAE J941e, (1983), SAE J1050, (1977), ISO 4513, (1978) and EECD 127, (1971). The standards and legislation are:–

Hierarchical Structure of Mirrors

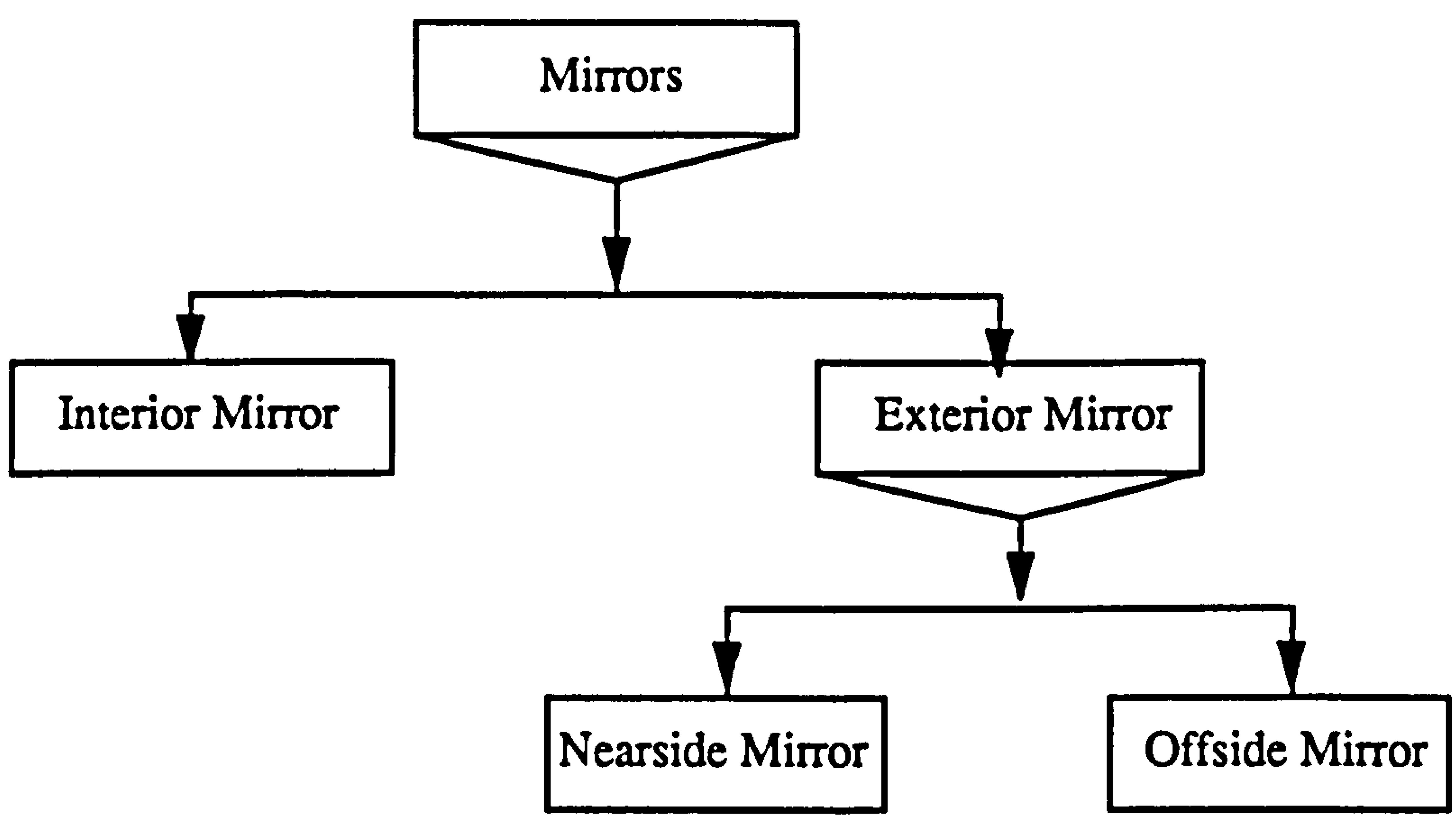


Figure 2.11 Hierarchical structure of mirrors.

1) Define the field of mirror view provided for the driver, establishing any obscurations caused by the body structure and fixtures such as windscreen wipers or by the interior mirrors;

2) Design suitable interior and exterior mirrors, and determine where they are best mounted so as to maximize the field of view while avoiding unnecessary obscuration of the direct view.

2.6.10.1 The Standards and Legislation for Mirrors

The Standards and Legislation for mirrors are defined in SAE J941e, (1983), SAE J985, (1967), SAE J1050, (1977), BS AU 199, (1984), EECD 127, (1971), FMVSS 111, (1990) and Haslegrave, (1993) (figure 2.12).

1) The nearest visible point seen in the interior mirror should not be further than 60 metres to the rear of the driver's eyes (Directive 127/71/EEC) or 200 ft (FMVSS 111) from the rear of the vehicle.

2) The nearside mirror must have sufficient width to give the driver a 20 metres wide view (Directive 127/71/EEC) or 20 degree horizontal angle view (FMVSS 111) beyond the second plane.

3) The nearside rear mirror view is bounded by a plane through the widest point on the side of the vehicle and a second plane 4 metres away from the vehicle.

4) The external mirror on the driver's side, has a required field of view extending rearward from a vertical plane (stretching between ground level and horizon) 10 m behind the driver's eyepoint.

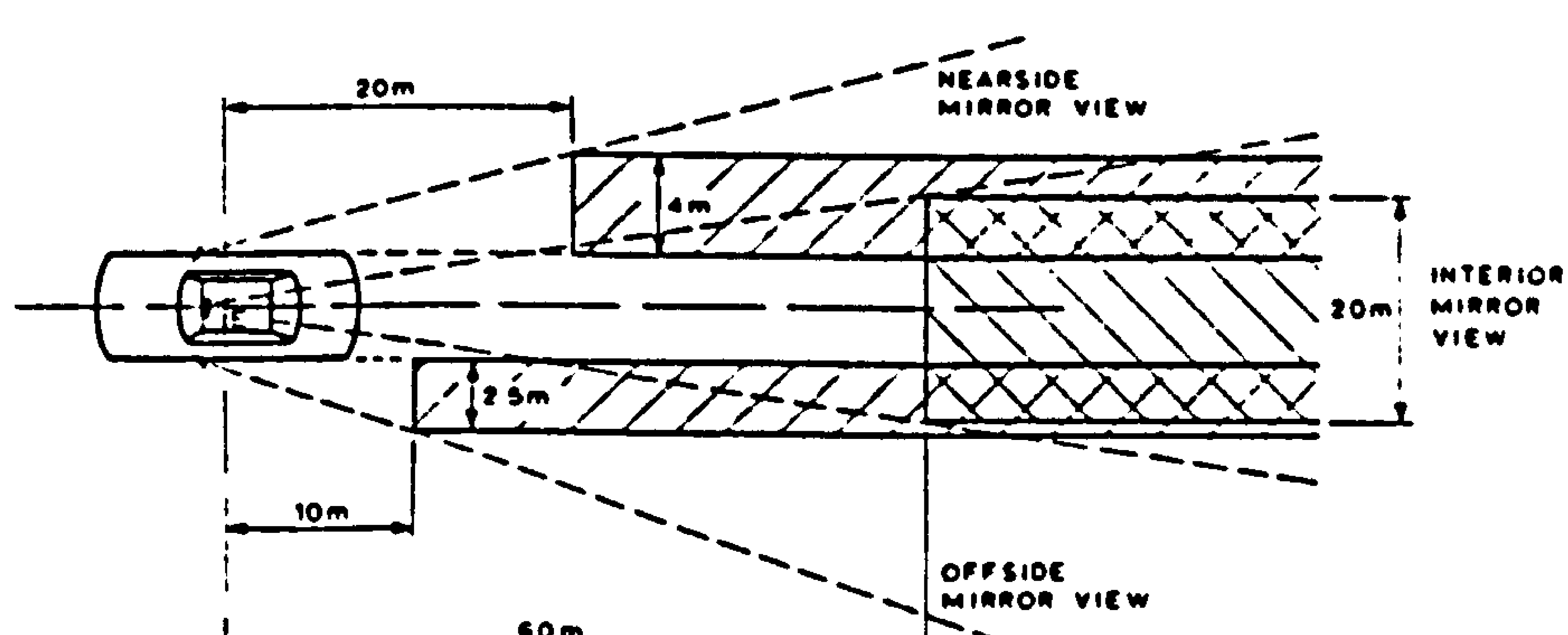


Figure 2.12 Minimum fields of view specified by ground plan
(Source: EEC Directive 127, 1971)

5) The offside rear view mirror has a required field of view bounded by a plane through the widest point of the vehicle and a second plane 2.5 metres away from the vehicle.

2.7 Application of Expert Systems in Ergonomics Design

A considerable amount of knowledge from various international organisations, has been described and illustrated by examples in this chapter. It is difficult to apply this knowledge directly to design problems if conventional design methods are used, due to the quantity of knowledge and conflicts in its content.

It is possible to identify three crucial aspects of the knowledge represented by standards and legislation.

i) It relates in part to variability of anthropometry in human dimensions.

ii) It relates in part to geometric aspects of the design being evaluated.

iii) It is sufficiently complex and interrelated to be unsuitable for straightforward algorithmic implementation in a computer system.

For these reasons it is desirable to combine the techniques of geometric modelling, human modelling and expert systems as described in the next two chapters.

CHAPTER 3

LITERATURE SURVEY: COMPUTER AIDED DESIGN SYSTEMS AND MAN MODELLING CAD SYSTEMS

3.1 Introduction

This chapter introduces the background of computer–aided design (CAD), man modelling CAD systems and geometric modelling.

In describing and classifying systems for workplace design, some twenty seven existing man modelling CAD systems are briefly compared. The SAMMIE Computer Aided Ergonomics Design System is described in more detail as it provides the basis of this research. Finally, the applications of Expert Systems in general and man modelling CAD systems are described.

3.2 Computer Aided Design and Man Modelling CAD Systems

The origins of today's CAD can be traced back to the beginning of civilization, when graphics communication in the form of drawings was used by the engineers of ancient Egypt, Greece, and Rome, and some of the drawings on Egyptian tombs can be considered as technical drawings. Also the work and notes of Leonardo da Vinci show the use of today's graphics conventions such as isometric views and sketching. Orthographic projection, which we practice today, was invented by the French mathematician Gaspard Monge (1746–1818) who was working as a designer, and was only made available for public engineers at the beginning of the nineteenth century after the military kept it a secret for thirty years (Zeid, 1991).

The development of CAD can be traced back to the early 1950's when researchers at the Massachusetts Institute of Technology (MIT) started their work on the SAGE (Semi–automatic ground environment) system producing simple pictures by interfacing a television–like cathode ray tube (CRT) with the Whirlwind computer. Later the development of the Sketchpad system by Ivan Sutherland at MIT, marked a great research achievement and was published as his thesis in 1962 (Zeid, 1991). The Sketchpad system demonstrated the creation of drawings and alteration of objects through an interactive computer graphics system used as an electronic drafting board (Majchrzak et al, 1987). By the mid 1960s, substantial computer graphics research was being undertaken by various groups, and the term "computer aided design", or CAD, started to appear. The term implied computer graphics with the word "design" extending it beyond basic drafting concepts (Zeid, 1991). The CAD functions of two–dimensional (2D) drafting, three–dimensional (3D) drafting, wireframe and sculptured surfaces (Majchrzak et al, 1987) were developed in the 1960s shortly after interactive computer graphics devices became available. CAD grew from the needs of the aerospace and automotive industries in the fifties and sixties.

CAD techniques are becoming very popular with engineers, designers and architects as they provide considerably more flexibility than conventional methods. They are being used increasingly by engineers, designers, ergonomists and stylists to quickly explore and easily determine the feasibility of various solutions to workplace design problems. Since the early 1970s the aerospace industry has been the leader in using CAD as a design analysis tool for the representation of complex three–dimensional shapes and for the production of drawings. Geometric modelling rather than drafting techniques, began to appear in the late 70's and early 80's (Majchrzak et al, 1987). This allowed the automotive industry, for example, to diversify its use of CAD systems to all design work from early stages of conceptual ideas, sketching and styling, presentation, through to detail drawing, assembly, production drawing, geometric modelling and finite element analysis. Similarly the electronic industry has become a leader in the use of CAD for design and development of circuit systems, new products, simulation of operation, production and assembly line design, and testing and evaluation of new products.

3.3 Definition of CAD and Man Modelling CAD Systems

CAD (Computer Aided Design) and Man Modelling CAD systems were created to support and assist in the design process, for the modification, presentation and analysis of designs. The three major functions in the design process are synthesis, analysis and presentation. Synthesis is the initial stage where conceptual ideas are generated using creativity and decision making, and results in engineering drawings or prototype models.

In the second stage the product is analyzed, and evaluated to determine its suitability for its intended function and if necessary it is modified. After several iterations, the final design is determined and prepared for presentation and approval. In the traditional design process, or non–CAD system, these functions are carried out manually with the aid of drafting instruments, sketchpads, handbooks, charts, tables, calculators, etc. At their simplest, computer–aided design systems merely substitute a computer and input/output devices for these traditional aids.

CAD programs are used to exploit the computer in the design of products and structures. The graphical product model existing in the computer's memory can be seen on the terminal display, and can be modified until the desired structure is found. Computer aided modelling spares the user the need for manual drafting and reduces the need for mock–ups and experiments, and facilitates the comparison and modification of design parameters.

Tomiyama and Yoshikawa, (1984) have suggested that intelligence should be the new concept for the future generation of computer aided design systems and information technology. It is expected that this will greatly assist the engineer and designer in saving time whilst solving design problems.

3.4 CAD Functions

CAD functions to provide assistance and support to the designer and engineer in their task of designing, with geometric modelling techniques, 2D or 3D drawings etc. CAD's development has come through the evolution of computer graphics and Computer Aided Drawing to provide quick visualization and verification.

3.4.1 Geometric Modelling

Since the time of Gaspard Monge (1746–1818), geometers have successfully developed techniques to bridge the gap between two and three dimensions through descriptive geometry, a variety of applied Euclidean geometry (Crapo et al, 1989). Details of related geometric modelling techniques can be found in Requicha (1980), Requicha and Voelcker, (1982), and Majchrzak et al, (1987). A designer constructs a geometric model on a CAD computer terminal to describe the shape of a structure to the computer. Complex shapes can be visualised at the design stage, through three–dimensional wireframe, surface or solid models. Once displayed, suitable models may be sectioned, intersected, blended with other shapes, and viewed from any direction. Solid models are a complete geometric description of a part, whereas wireframe and surface models are only a partial representation. The systems usually incorporate extensive colour choice and tone control for improved visualisation of shape and ready identification of components, surfaces and cross–sections (Hawkes and Abinett, 1984).

3.5 Man–Modelling CAD Systems

Man modelling CAD systems are new design tools for designers and engineers, and at present a number of such systems have been developed and made commercially available. These tools provide a quick and accurate means of evaluating the interaction between an operator and their workplace. Without such tools, evaluations will be time consuming, costly and less effective. By using CAD/CAEDS or man modelling systems the designer or engineer can evaluate and improve ergonomics from the beginning of the design process, when modifications of a design concepts can be undertaken easily and quickly.

Hickey et al. (1985) list 77 computer aided design and assessment programs, of which 37 include a man model, wholly or partially. Some programs seem to have disappeared or are no longer available to the general public, while new ones, along with inexpensive yet more powerful hardware support, appear at frequent intervals (Hickey et al, 1985, Hoekstra, 1993). This review had broad terms of reference, and some useful summaries have been presented by Kennedy, (1995).

The survey presented here is restricted to those systems which provide variable man models within a more general workplace modelling capability.

3.5.1 Man Modelling System Classification

Computer man models can be classify into three principal application groups; animation, anthropometry and biodynamics (Rothwell and Hickey, 1986).

Animation programs present the motion of human beings in a stylised form in real time. Natural, sporting or pathological motion sequences are generated using motion control algorithms or alternatively are captured and reviewed for analysis or entertainment/leisure purposes (Thalmann and Thalmann, 1991).

The anthropometric programs are based upon human structural and functional characteristics and are used to evaluate and analyse work tasks performed in some environment. This analysis may be simple or sophisticated and include things such as control of dimensions, body posture, body proportions, joint location, articulation range, grip functions, and field of vision.

Biodynamic programs apply anthropometic data in order to investigate the motion or the potential for motion and are particularly used in crash simulation situations.

In general anthropometric and biodynamic programs evaluate/simulate the interaction of man model with his working environment whereas animation programs consider the human being in isolation, and geometric environments are only background aids for scaling or aesthetic purposes.

3.5.2 Man Modelling System Requirements for Workplace Design

Man models have specific behavioural characteristics created by the disciplines of ergonomics, anthropometry, biomechanics, psychology, physiology and sports science according to their needs and interests.

Designers and engineers will have interest in the application of man modelling for:–

i) Anthropometric considerations – for assessment of reach for a specified range of stature and build, fit and vision.

ii) Biomechanical analysis – evaluating the performance capabilities of the operator in carrying out manual tasks.

iii) Information related workplace design – man model control loops and the potential for overload.

iv) Safety related workplace design – hazard avoidance and conformance to current and pending legislation.

v) Standards and information related workplace design – compliance with legislation, rules and regulations, and design working practice.

Computer aided ergonomics design systems provide the user, quite often an ergonomist, designer, or engineer, with the possibility to visualize on a computer screen a man model, integrated within a relevant workplace design. The flexibility in adapting the design to changes in functional postures of a chosen percentile of a target population is one of the main features of computer man modelling systems. Ease of use of the man model, directing its line of sight and displaying the resulting field of view, are becoming standard facilities. The availability of computer man modeling techniques and adaptive capabilities via libraries of anthropometric data and workplace elements, creates an environment for real–time interaction between the designer and design alternatives, and in this way makes computer aided ergonomics design systems powerful tools in the first stages of human workplace design.

Computer aided ergonomics design is an activity which utilizes a computer to assist in the creation, modification, analysis and presentation of a human related design.

The main features of computer man–modelling systems are:–

1) 2D/3D modelling system for workplace/workspace equipment.

2) A man model containing a range of evaluations.

Typical ergonomics evaluations available in man modelling include:–

1) Reach tests to or across surfaces of items in the workplace model.

2) Fit tests, to see if the operator fits into available working space without interference with machines.

3) Vision tests.

4) Posture assessment in relation to the extent of movement available to the limbs.

5) Balance assessment by locating the centre of gravity of the man and loads being carried.

The first and oldest computer man modelling systems were developed in the late 1960s and included SAMMIE, BUFORD, BOEMAN, COMBIMAN, CREW CHIEF, CYBERMAN, CAR, BUBBLEMAN, TEMPUS and HEINER. An example of a very recent modeller is JACK. Most computer man modelling CAD systems are three–dimensional and run on a computer workstation, but a few can be used on personal computers (eg. MINTAC, COMONS, OSCAR, TADAPS, FRANKY, CADPEOPLE, MAN3D, MANNEQUIN, SAFEWORK, ErgoSPACE, and ErgoSHAPE). Many of these PC–based systems are two–dimensional.

In recent years these older systems have been developed and also new systems have appeared and been widely used to evaluate ergonomics aspects of the human being in workplace design. Descriptions and and comparisons of man modelling CAD systems are given in the following sections.

3.6 Comparison of Man Modelling CAD Systems

There are various existing man modelling CAD systems, and comparison shows that there are considerable differences between them in terms of features such as the complexity of the man model, numbers of joints and segments, the anthropometric databases available, the ability of the modeller to provide functional modelling, the assessment of reach, and their potential applications. The well–established systems are compared with respect to these characteristics in table 3.1.

Table 3.1 Comparison of Man Modelling Program Capabilities

System	Origin	Man–Model					Purpose	Ws/ PC	ES
		Yr.	3D	Jts.	%tile	Database			
ADAPS/ TADAPS	Holland	87	2D	24	50th	Civil/Pilot	automobile evaluation	VAX	?
ANYBODY/ CADKEY	Germany	90	3D	No	50th	DIN33402	design of workplaces	IBM AT	?
BOEMAN	USA	69	3D	23	50th	Cockpit	aeroplane cockpit	?	?
BUBBLEMAN	USA	79	3D	20	Any	Car design	movement visualisation	VAX	?
BUFORD	USA	66	3D	15	50th	Spacesuit	cockpit evaluation	Cptrvis.	?
CADPEOPLE	Holland	91	3D	No	50th	DINED	design of workplaces	PC	?
CAR	USA	76	–	31	Any	Crewstation	reach analysis	PC	?
COMBIMAN	USA	73	3D	35	Any	Craft pilot	aeroplane cockpit	IBM	yes
COMONS	Canada	89	2D	15	Any	Civilians	lifting analysis	PC	?
CREW CHIEF	USAForce	88	3D	12	No	Craft pilot	aeroplane cockpit	Cptrvis.	yes
CYBERMAN	SAE USA	74	3D	15	Any	Automobile	automobile cockpit	CDC	?
ErgoSHAPE	Finland	87	2D	17	50th	Civilians	handling evaluation	PC	?
ErgoSPACE	Finland	87	3D	17	No	Civilians	workplace evaluation	PC	yes
ERGOMAN	France	84	3D	?	?	?	design of workplaces	MATRA	?
FRANKY	Germany	86	3D	No	No	DIN33416	reach analysis	?	?
HEINER	Germany	?	3D	No	50th	DIN33402	design of workplaces	HP 1000	?
ISM	Germany	?	3D	?	?	Simulation	automobile evaluation	?	?
JACK	USA	92	3D	17	Any	Car design	automobile cockpit	SG Ws	?
MADYMO	Holland	?	3D	?	?	Simulation	crash simulation	?	?
MAN3D	France	86	3D	23	Any	Civilians	automobile evaluation	PC MATRA	
MANNEQUIN	USA	89	2D	16	Any	Civilians	design of workplaces	PC	?
MINTAC	Finland	84	3D	6	Any	Civilians	reach, OWAS analysis	Cptrvis.	?
OSCAR	Hungary	86	3D	–	No	DIN33402	design of workplaces	IBM	?
SAFEWORK	Canada	90	3D	14	No	Civilians	design analysis	PC	?
TEMPUS	USA	87	3D	17	No	US Army	cockpit evaluation	Iris 3030	?
WERNER	Germany	87	3D	10	No	DIN43116	ergonomics assessment	Atari ST	?
SAMMIE	UK	69	3D	21	Any	Variety	man–machine interaction	SUN SPARC	yes

ES = Expert System included
Ws = Workstation
PC = Personal Computer

3.6.1 Existing Man Modelling CAD Systems

3.6.1.1 ADAPS/TADAPS

The Twente Anthropometric Design Assessment Program System (TADAPS) was developed in 1987, jointly by the University of Twente (UT) and Delft University of Technology (DUT). It originated from ADAPS (Anthropometric Design Assessment Program System) which was developed in the late 1970s and implemented on PDP–11 computers. (Post and Smeets, 1981, and Westerink et al, 1990).

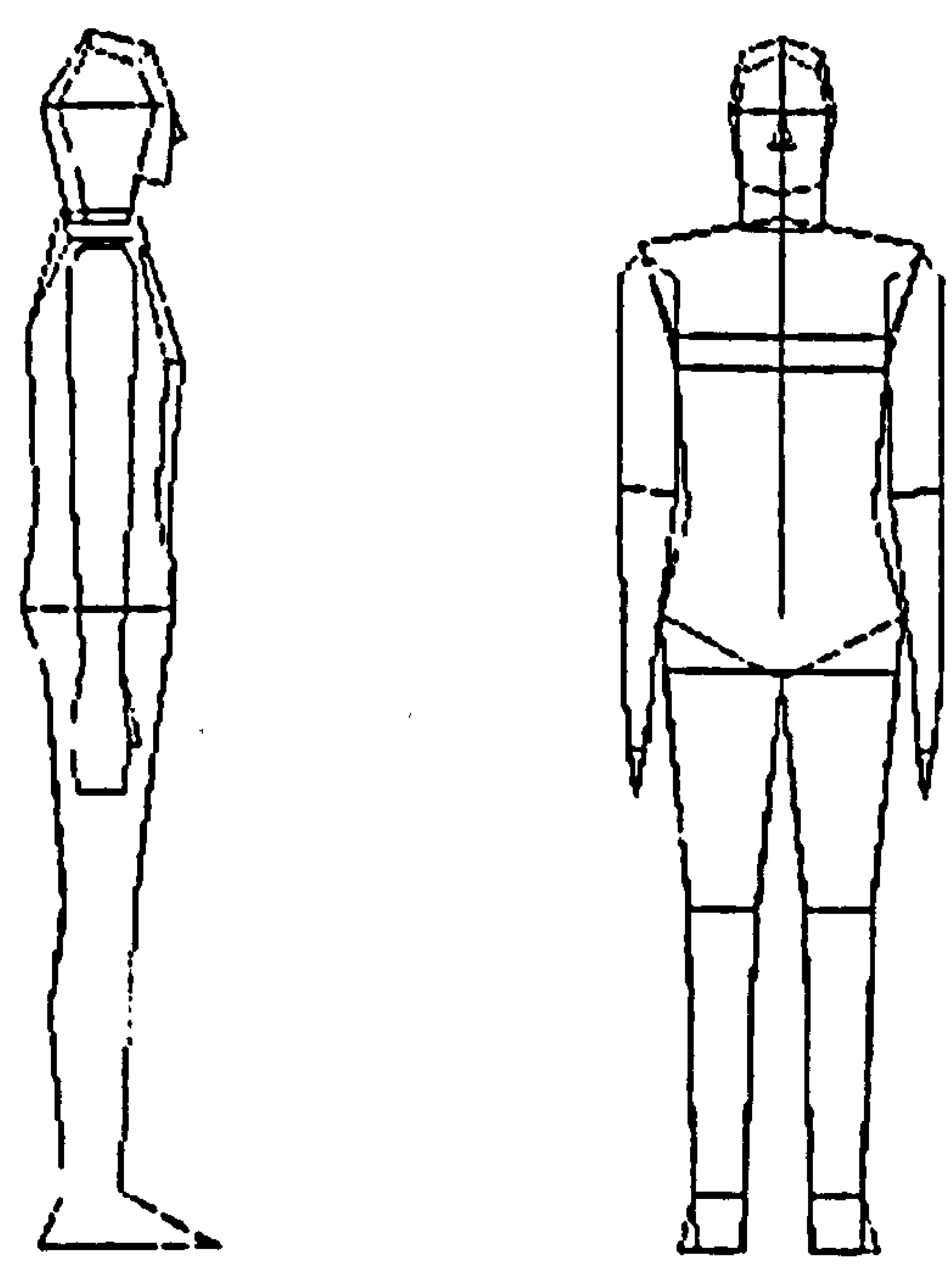


Figure 3.1 (T)ADAPS wireframe man model (Source: Hoekstra, 1993, Westerink et al, 1990)

The TADAPS man model is a three–dimensional solid model used in the design and evaluation of automobiles. It consists of 24 links with a wire frame outline. The system includes its own workplace modeller and the whole system appears to have a very similar suite of facilities to SAMMIE in concept, although it is not as fully developed. The anthropometric database comprises Dutch men, women and 4 year old boys as well as American pilots and it is relatively easy to create models for other populations. All percentiles can be chosen although the man model is linearly scaled out of the 50th percentile proportions and it is not clear whether individual body segments can be set to different percentile values. TADAPS offers a prediction of the compression and shear force of the intervertebral disc L5–S1 for various postures and external loads. TADAPS has no expert system (Westerink et al, 1990, and Porter et al, 1993).

3.6.1.2 ANYBODY/CADKEY

CADKEY is a CAD package for IBM AT PC machines developed in 1990 and marketed by IST GmbH in West Germany. ANYBODY is a three–dimensional template or stencil of the human body that can be used with CADKEY, through use of CADKEY's integrated software modules. The workplace can be modelled and viewed using existing CADKEY facilities, while man model manipulation is provided by an external program. The animation programme ANI operates with a separate databank communicating with the CADKEY databank. This is an interactive system permitting modification of the models by input of data to define percentiles or sizes by proportional alteration of torso, legs, arms, 2D/3D transformation by data input and change of viewpoint, and 2D/3D animation.

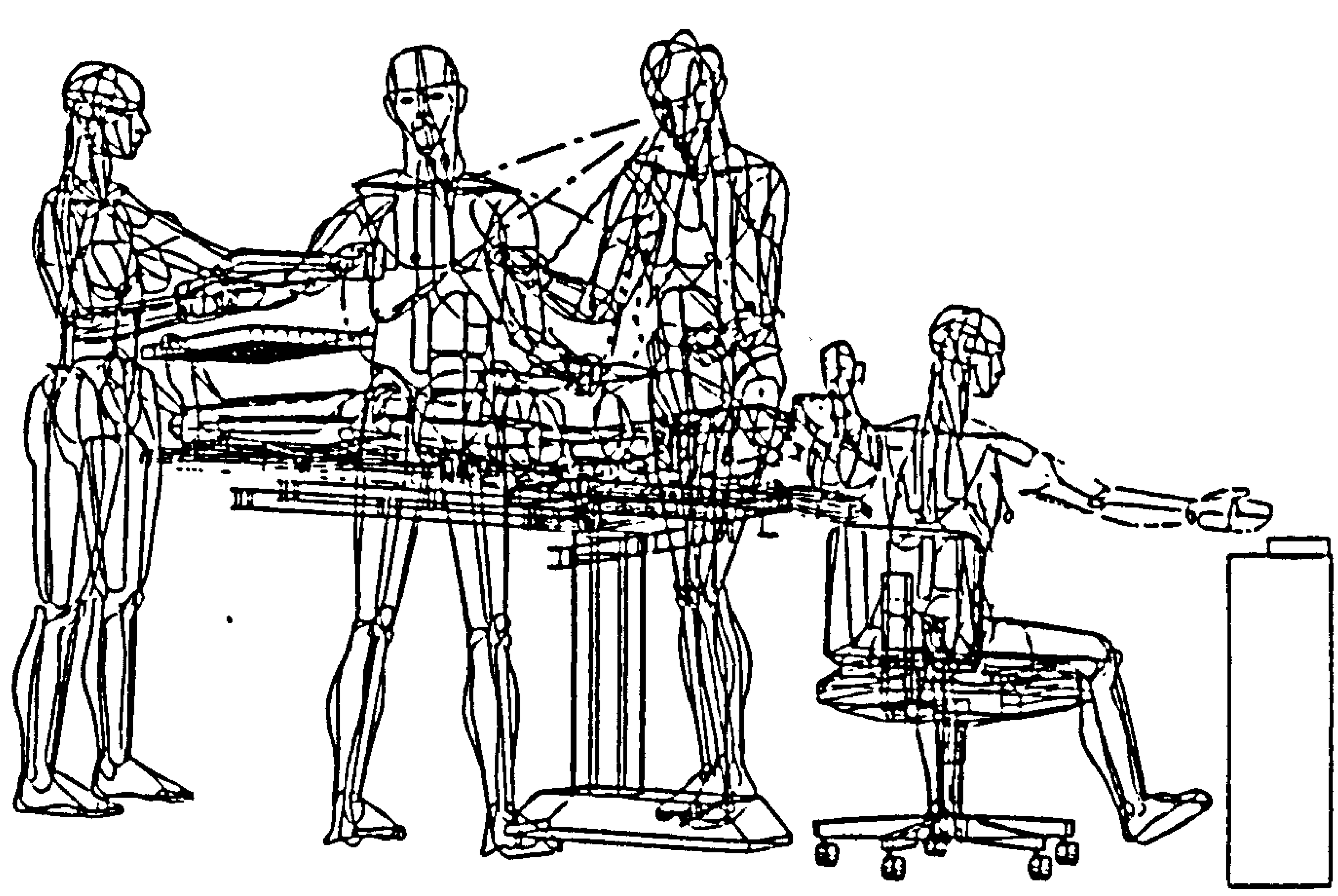


Figure 3.2 ANYBODY/CADKEY (Source: IST GmbH, West Germany, Porter et al, 1993)

ANYBODY provides default male and female models for the 5th, 50th and 95th percentile German population. The anthropometric data is taken from DIN 33402 part 2 and provides a range of different somatotypes (ectomorph, mesomorph and endomorph). The templates can be linearly scaled to represent other body dimensions which may be enhanced using different hand modules and the provision of shoes and safety helmet. No published studies on the development or application are available (Porter et al, 1993).

3.6.1.3 BOEMAN (BOEing MAN Model)

The BOEing MAN Model (**BOEMAN**) is one of the earliest computer man modelling systems and was developed in 1969 by the Boeing Corporation in Seattle, Washington. The man model has been used as a tool to aid cockpit design for military aircraft, such as the A7–E. BOEMAN provides a mathematical computer modelling analysis of the human workplace geometry. It uses a multimethod approach involving experimental laboratory work, computer modelling, flight simulators and actual flight test evaluation.

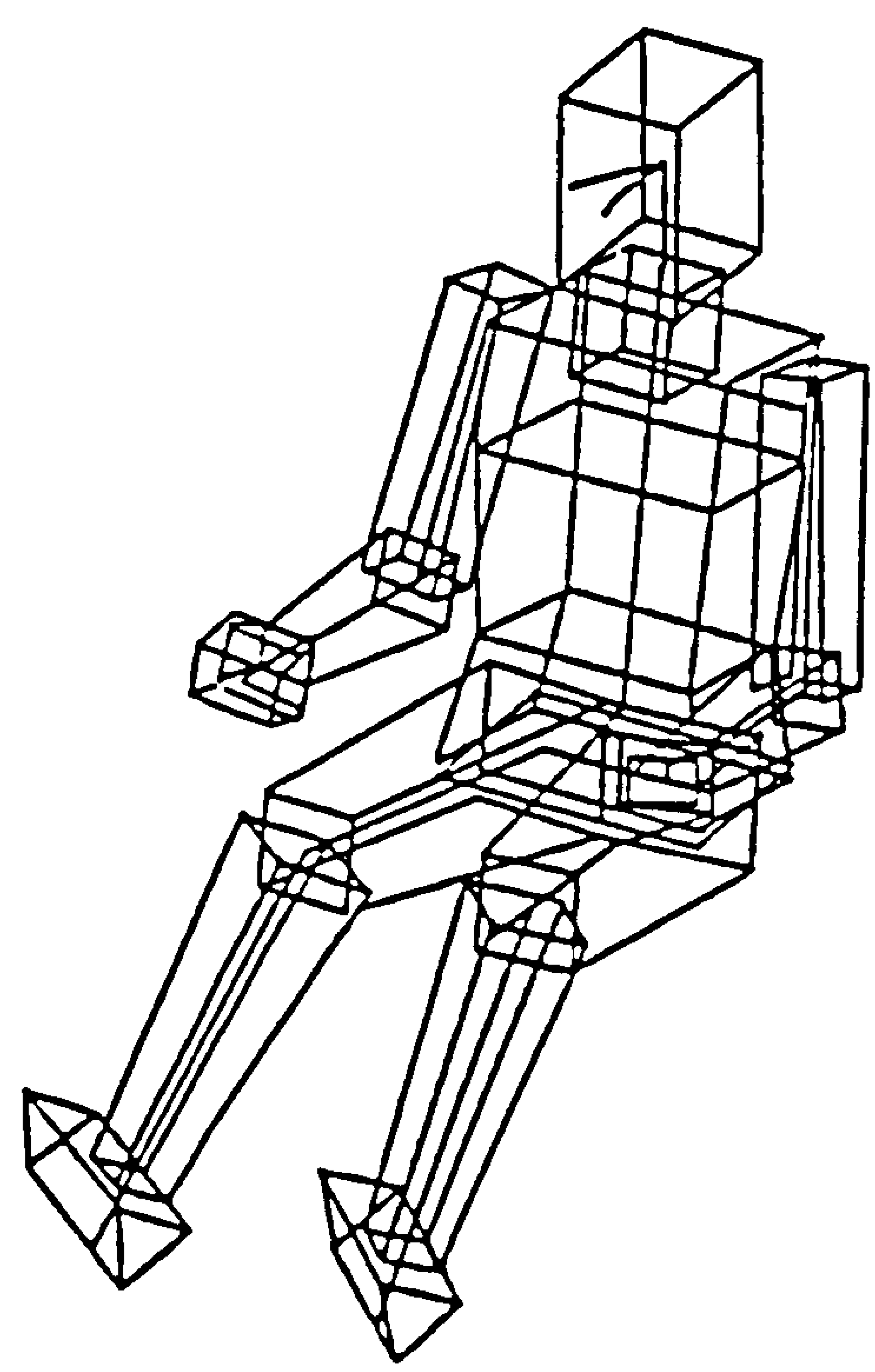


Figure 3.3 BEOMAN (Source: Dooley, 1982)

The man model eyepoint can be depicted, reach evaluated, and physical and visual interferences indicated, and task sequences executed. The initial model used a 'stick–pilot' concept that had 23 joint segments with variable link lengths. Subsequently a 'skin' was provided to produce a three–dimensional outline figure. The system was implemented on IBM mainframe computers and although designed for sophisticated analysis it was non interactive in use as graphic CRT terminals were not commonly in use at that time. The man model is based on the average, 5th, 50th and 95th percentile anthropometric data from Dreyfuss (1966) and Dempster (1955). Work on **BOEMAN** was discontinued in 1970 but it formed the basis for development of COMBIMAN (Fetter, 1980 and 1982, Dooley, 1982, and Okey et al, 1989).

3.6.1.4 BUBBLEMAN

The **BUBBLEMAN** model was developed in 1979 by the Centre for Computer Graphics Research, University of Pennsylvania, USA, using Univac or VAX computers. The program was designed for use in design studies of vehicle interiors and contains a three–dimensional solid volume model that can be used to display the output of an analytical program.

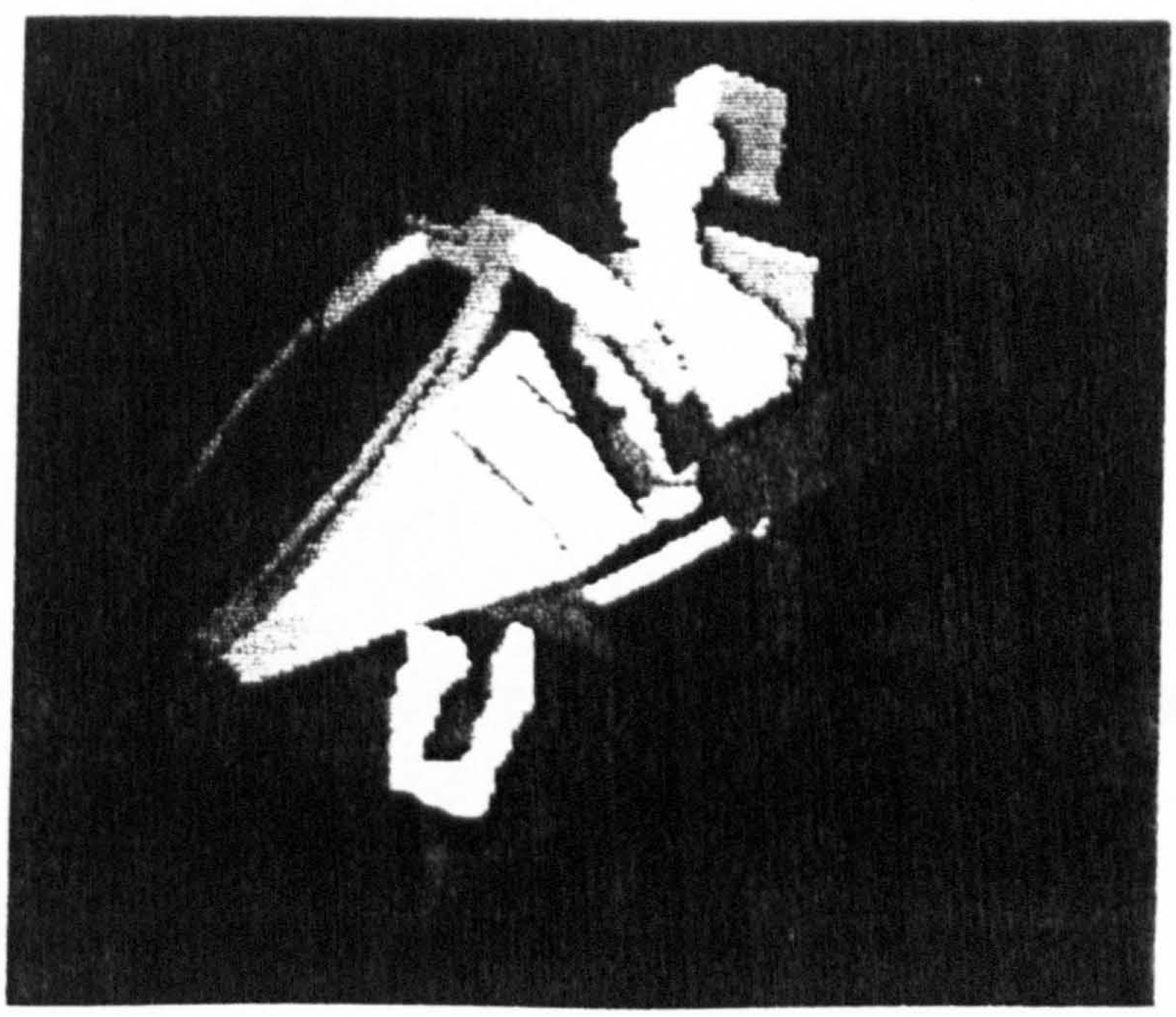


Figure 3.4 **BUBBLEMAN** (Source: Dooley, 1982).

The man model, when viewed within an environment, consists of a number of spheres or bubbles, and an eyepoint view can be superimposed on the image. The body consisted of 20 links, and 19 joints covered by a "spin" of overlapping spheres. Movements can be analysed independently of the entire model, by speeding up computation to permit closer observation.

Clashes or collisions within the model can be detected between the model and its environment, or between models. The man model anatomical structure can be varied from 60 to 600 spheres including a fully articulated hand and palm. The program can also be used for reach analysis, by a three-step reach to a defined point, or for dynamic reach analysis, where the operator specifies timed sequences of point movements or reaches. BUBBLEMAN interfaces with anthropometric analysis are presented via a black and white or colour solid model display (Dooley, 1982).

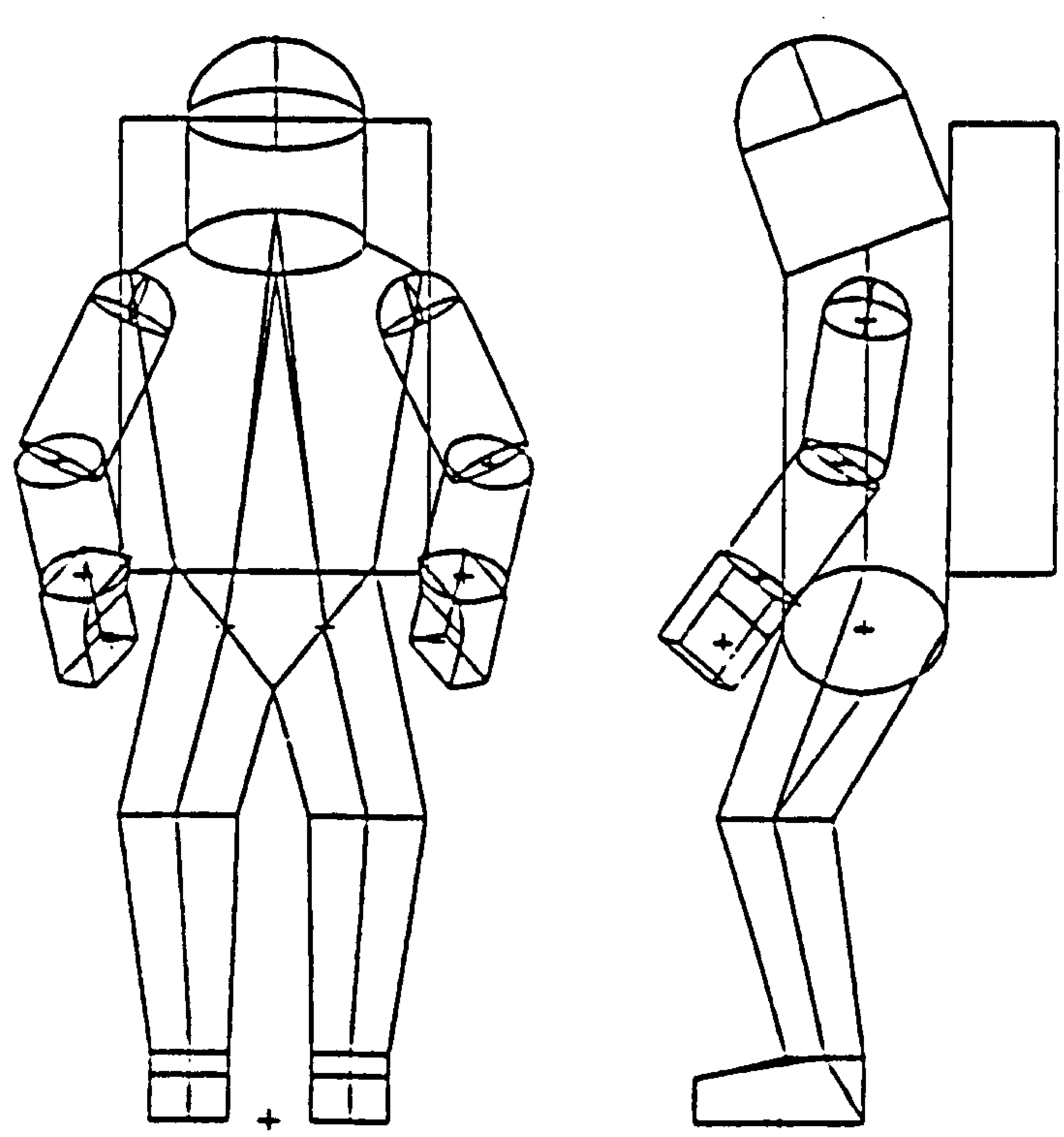


Figure 3.5 BUFORD: A model of an astronaut wearing a spacesuit
(Source: Dooley, 1982).

3.6.1.5 BUFORD

BUFORD was one of the earliest three-dimensional man models developed in 1966, by Rockwell International, Downey, California, U.S.A. using a Computervision turnkey CAD vector graphics system and static display. It is a simple man model of an astronaut wearing a spacesuit. The man model with 15 links is a discrete entity, with body segment enfleshment lines for 5th, 50th and 95th percentile models based on Dreyfuss, (1966), However the anthropometry is scalable to any dimension. The model uses joint movements to graphically depict general reach and reach areas around an operator-positioned model and clearances are visually determined, The system can be used to design an environment around this model and to modify the model's limbs, body positioning, and percentile size (Dooley, 1982).

3.6.1.6 CADPEOPLE

The Computer Aided Design PEOPLE model (CADPEOPLE) was developed in July 1991 jointly by the Dutch ergonomics consultancy ErgoCAD and software consultants "W6". The system uses Autocad and has a range of human models consisting of four males and four females to represent the Dutch population. The anthropometric data was taken from Dutch Standards DINED and Dreyfuss and provides a range of different somatotypes. The CADPEOPLE model is similar to ANYBODY. Man model parts are drawn as three–dimensional vectors making hidden–lines removal, shading or rending impossible (Vellinga, 1992).

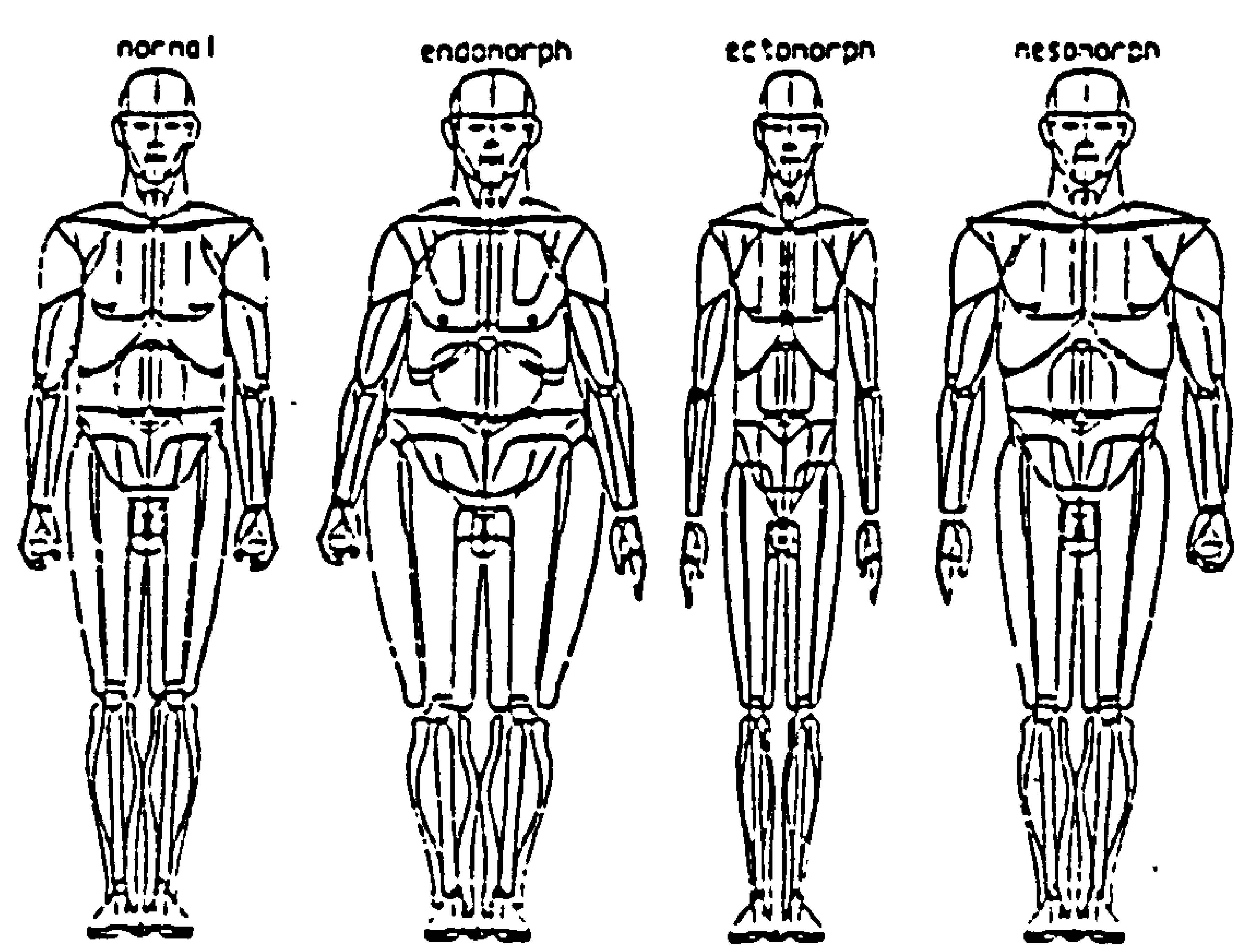


Figure 3.6 CADPEOPLE model (Source Vellinga, 1992).

3.6.1.7 CAR (Crew Assessment of Reach)

The Crew Assessment of Reach model (CAR) was originally developed in 1969 by the Boeing Aerospace Corporation for the Naval Air Development Center, U.S.A., and is based in part on the BOEMAN program. The system has no graphical display and has been designed specifically for the in-house assessment of reach in aircraft crew stations. The original system was modified and refined by Analytics, a company in Willow Grove, Pennsylvania in 1980 to produce CAR-II. The man model has 31 links derived from 12 external anthropometric measurements; linkages for a reach analysis are constructed by adding successive links beginning at the lumbar link and progressing in the direction of each specified control; the location of the lumbar link is fixed, based upon the seat geometry and operator clothing (Dooley, 1982).

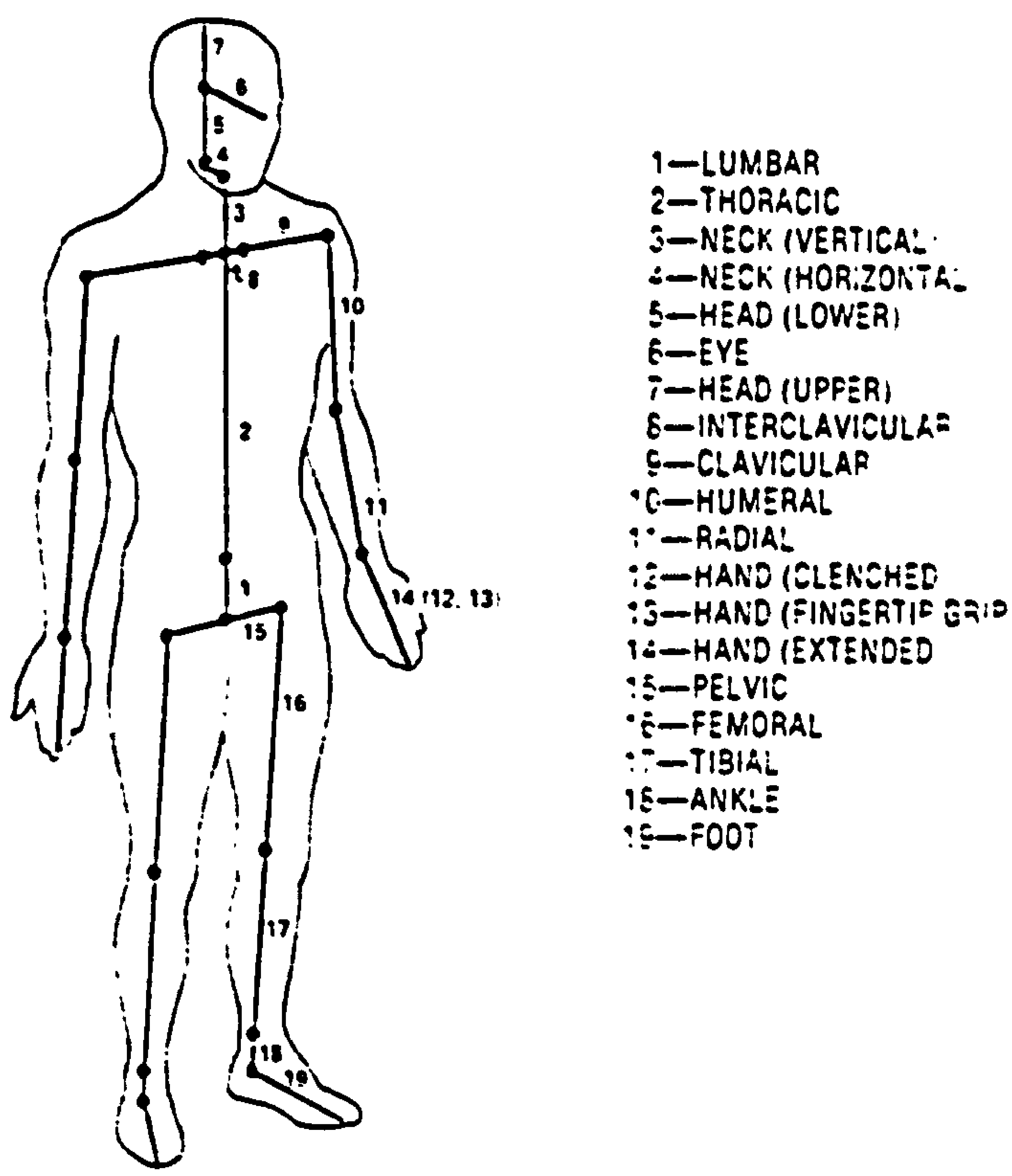


Figure 3.7 CAR (Source: Dooley, 1982).

3.6.1.8 COMBIMAN

COMputerized BIOmechanical MAN model (COMBIMAN) was developed in 1973, jointly by the Armstrong Aerospace Medical Research Laboratory (AAMRL) and the University of Dayton Research Institute. COMBIMAN has been distributed to the major aerospace industries since 1978. COMBIMAN is a three-dimensional interactive computer man model of an aircraft pilot used to evaluate the physical accommodation of the pilot in existing or conceptual crewstation designs. The man model has a 35 internal link-skeletal system derived from 12 readily measurable anthropometric surface dimensions. Body motions are realistically constrained within accurate joint movements, and lap and shoulder harness restraints can be applied. Flight suits and helmets can be included; reach success/failure indicated; clearance is detected visually (Dooley, 1982, and Okey, 1989).

COMBIMAN has two methods for dimensioning the pilot model. The first is to specify one or two critical dimensions, the remainder being generated using appropriate regression equations. The second methods involves the specification of dimensions for a variety of pilot models.

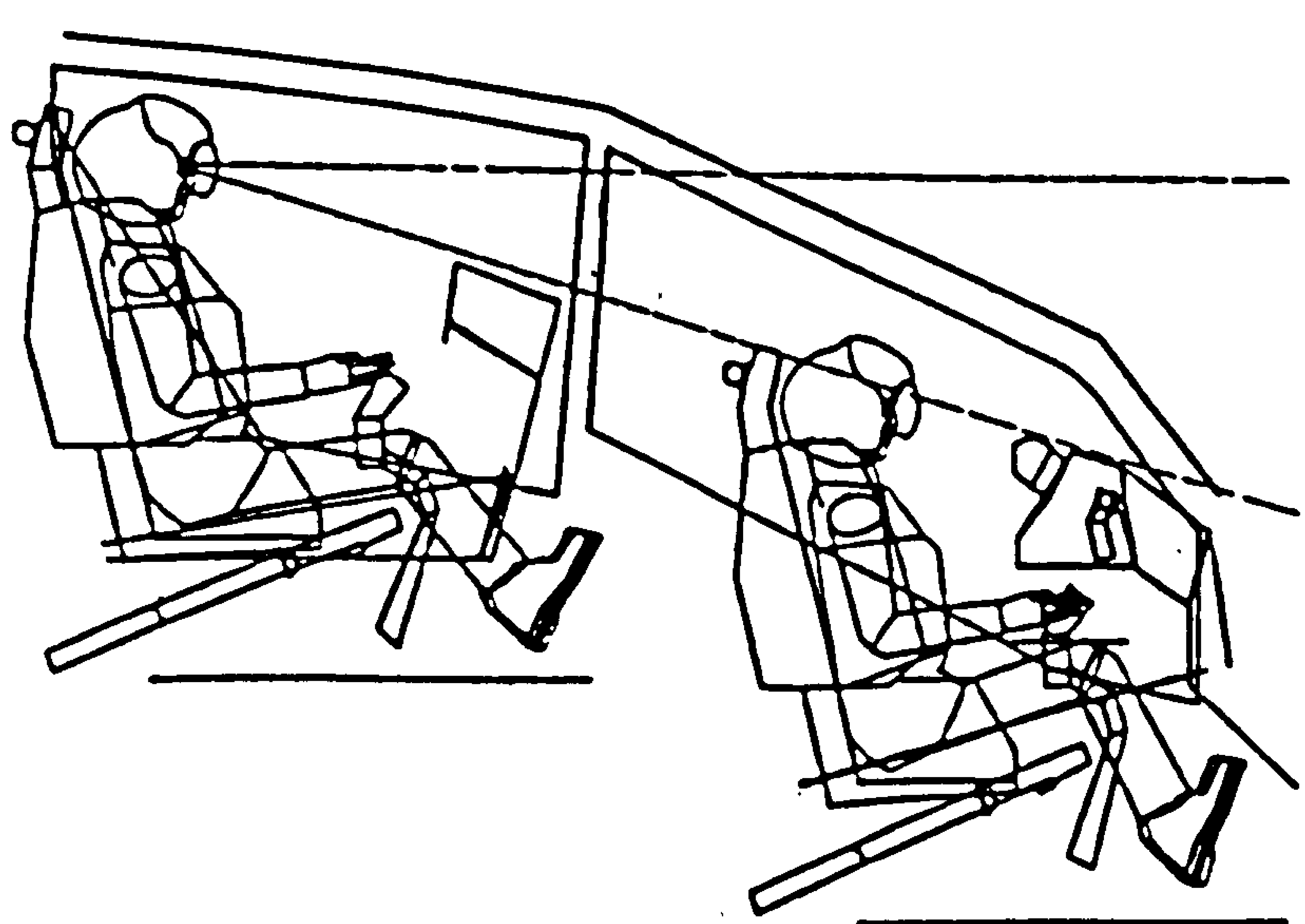


Figure 3.8 COMBIMAN model of pilot in a helicopter crewstation
(Source: McDaniel, 1990)

The latter method is useful in the assessment of multivariate accommodation from the anthropometric database consisting of military pilots and women. The pilot model is constructed using an array of small interconnected triangles, but this complexity can be automatically reduced to just the profile view and any essential features from any viewing angle. Postures can be set by specifying individual joint angles or by using task related commands. Pilot visibility plots can be produced to required military standards (MIL-STD-850), and reach tests conducted for three types controls, six types of clothing, three types of harnessing and seven reaching planes. Strength predictions can be made for seated pilots for both hand foot controls (McDaniel, 1990, and Porter et al, 1993). COMBIMAN represents an expert system that can perform four types of analysis: fit, visibility, reach and strength for operating controls with arms and legs (McDaniel, 1990).

3.6.1.9 COMONS

COmputerized MOvement Notation System (COMONS) was developed in 1989, jointly by the University of Waterloo and the Defence and Civil Institute of Environmental Medicine, Ontario, Canada, as a computer-based movement data management and simulation system. COMONS is a two-dimensional man modelling system developed on an IBM PC compatible running MS-DOS with 512 kilobytes of memory, colour graphics, the multihalo graphics library, and Microsoft C. The user can edit and customize the physical dimensions of the human figure to any person and the model can be drawn with a maximum of 15 body segments. Once the human figure is defined in the sagittal plane, the relative joint angles for each segment are entered, and the absolute position of the head is defined by its x-y coordinates in the workplace.

The program can store specific models for each of 5th, 10th, 50th, and 95th percentile. The COMONS system was intended to provide a cost-effective, standardized, and flexible tool for clinicians and ergonomists in the evaluation of movement. A COMONS option was specially built for the assessment of lifting and gait by the provision of strobing of movement along with relevant information shown on the top of the plots. The postures may then be displayed in strobe mode, according to user defined positions. The gait parameters displayed are calculated for each gait profile (Patla and Eickmeier, 1989).

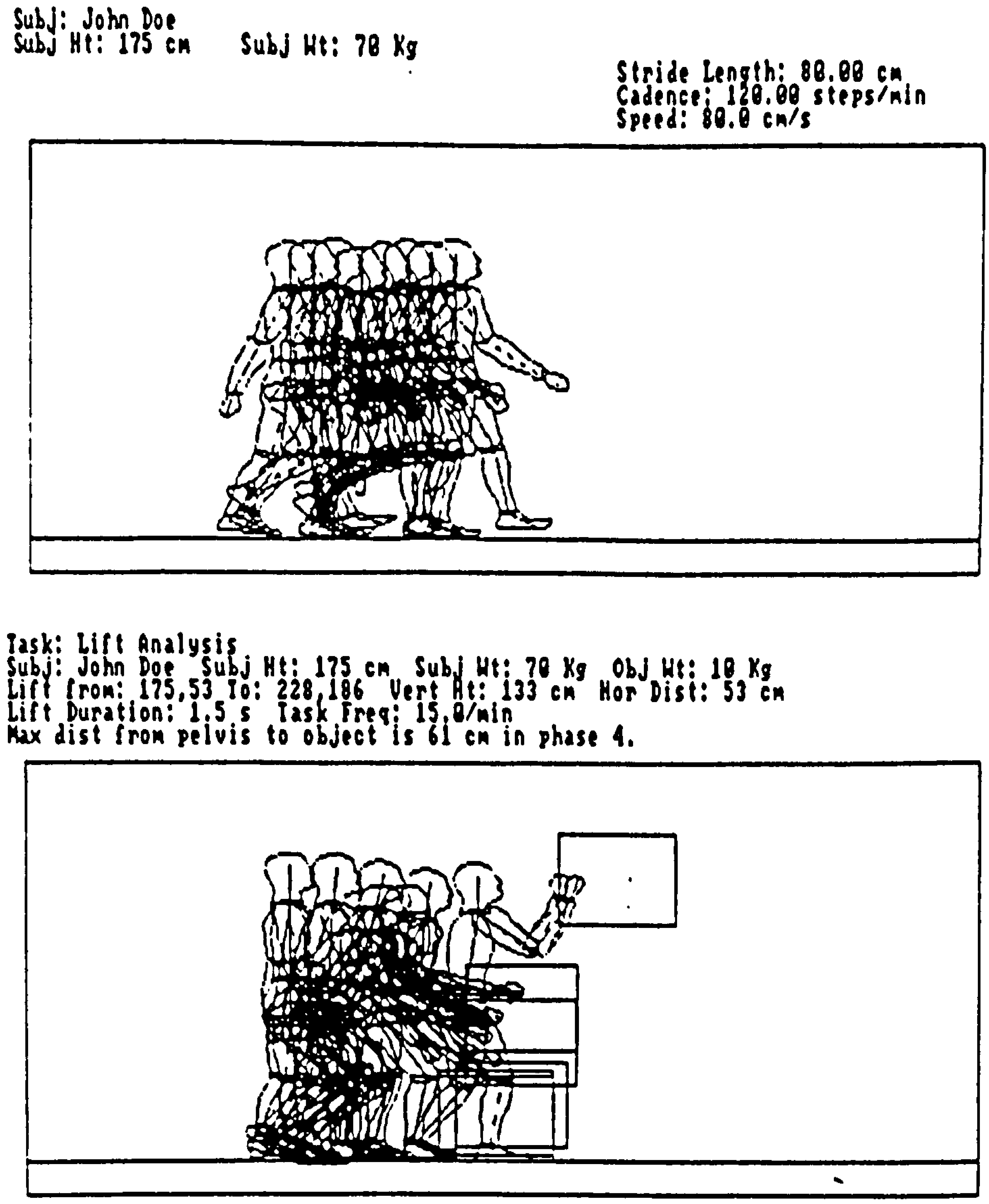


Figure 3.9 COMONS (Source: Patla, and Eickmeier 1989)

3.6.1.10 CREW CHIEF

CREW CHIEF is a computer aided design model of an aircraft maintenance technician and was developed in 1973, jointly by the Armstrong Aerospace Medical Research Laboratory (AAMRL) and the Human Resources Laboratory (HRL). Much of CREW CHIEF's functionality is based upon that incorporated in COMBIMAN. CREW CHIEF can generate 10 sizes of human (five male and five female), has mobility limitations for 12 segments, and evaluates physical access for reach into confined areas (tools and objects), automated obstacle avoidance, visual accessibility and strength analysis (using hand tools and manual materials handling of objects).

CREW CHIEF represents an expert system which lets the designer perform the functions of an expert ergonomist. The man model simulates a maintenance activity on the computer generated image of the design and determines if the required maintenance activities are feasible. (Okey et al, 1989, and McDaniel, 1990).

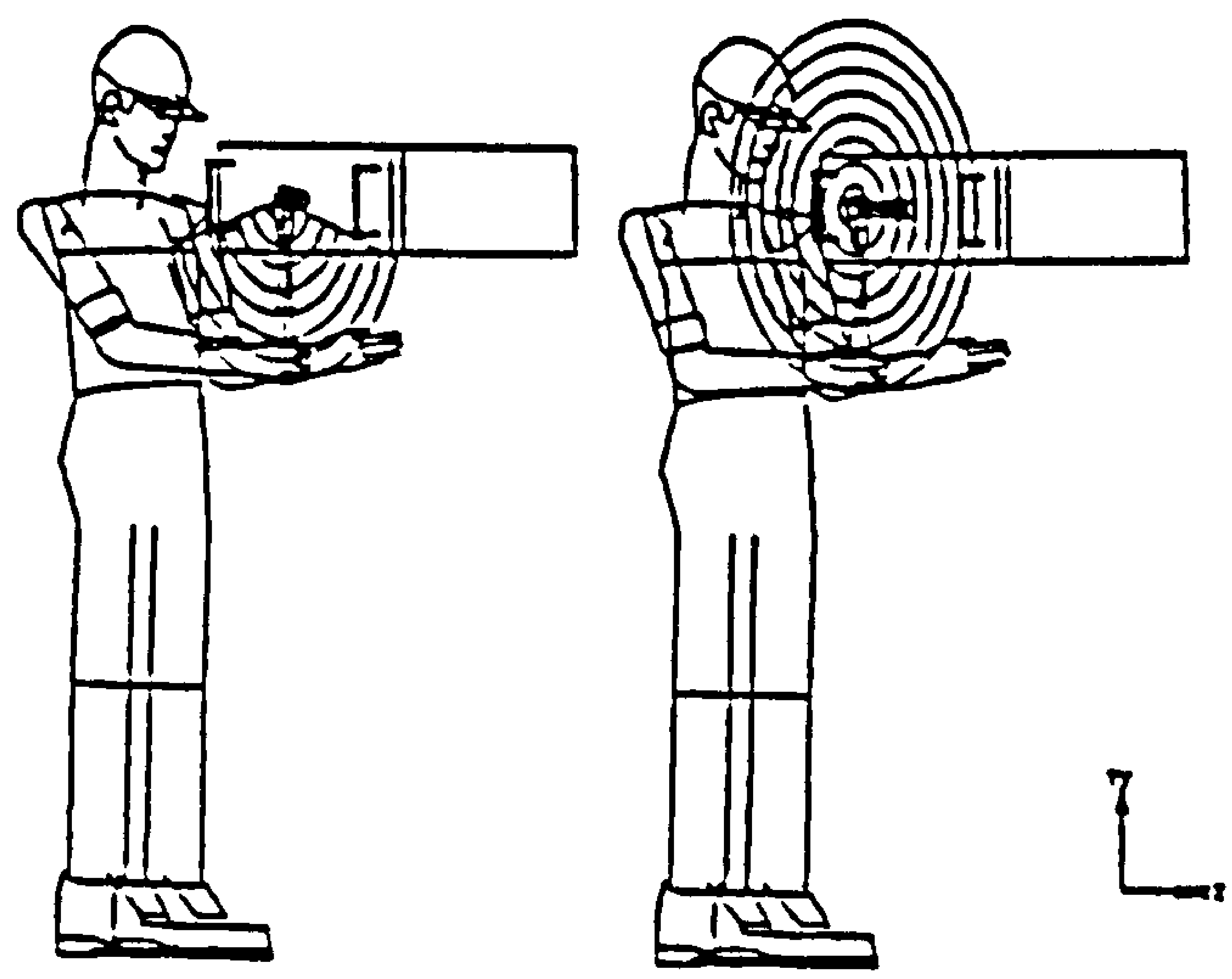


Figure 3.10 CREW CHIEF model simulates the task of rotation of a ratchet wrench (Source: McDaniel, 1990)

3.6.1.11 CYBERMAN

The CYBERnetic MAN model (CYBERMAN) was developed in 1974, by the Chrysler Corporation, for use in the automobile industry in the design and development of cars, to model drivers, passengers, and their activities inside and outside the car. The man model is three-dimensional with a stick structure of 15 links with or without complete wireframe outline. As there are no constraints on the choice of joint angles the man model's usefulness for in-depth ergonomics evaluation is rather limited. The anthropometric data is based on a combination of dimensions from the SAE two-dimensional manikin and the HEW 1960 census (Blakeley, 1980, Dooley, 1982, and Okey, 1989). The system is not generally available (Porter et al, 1993).

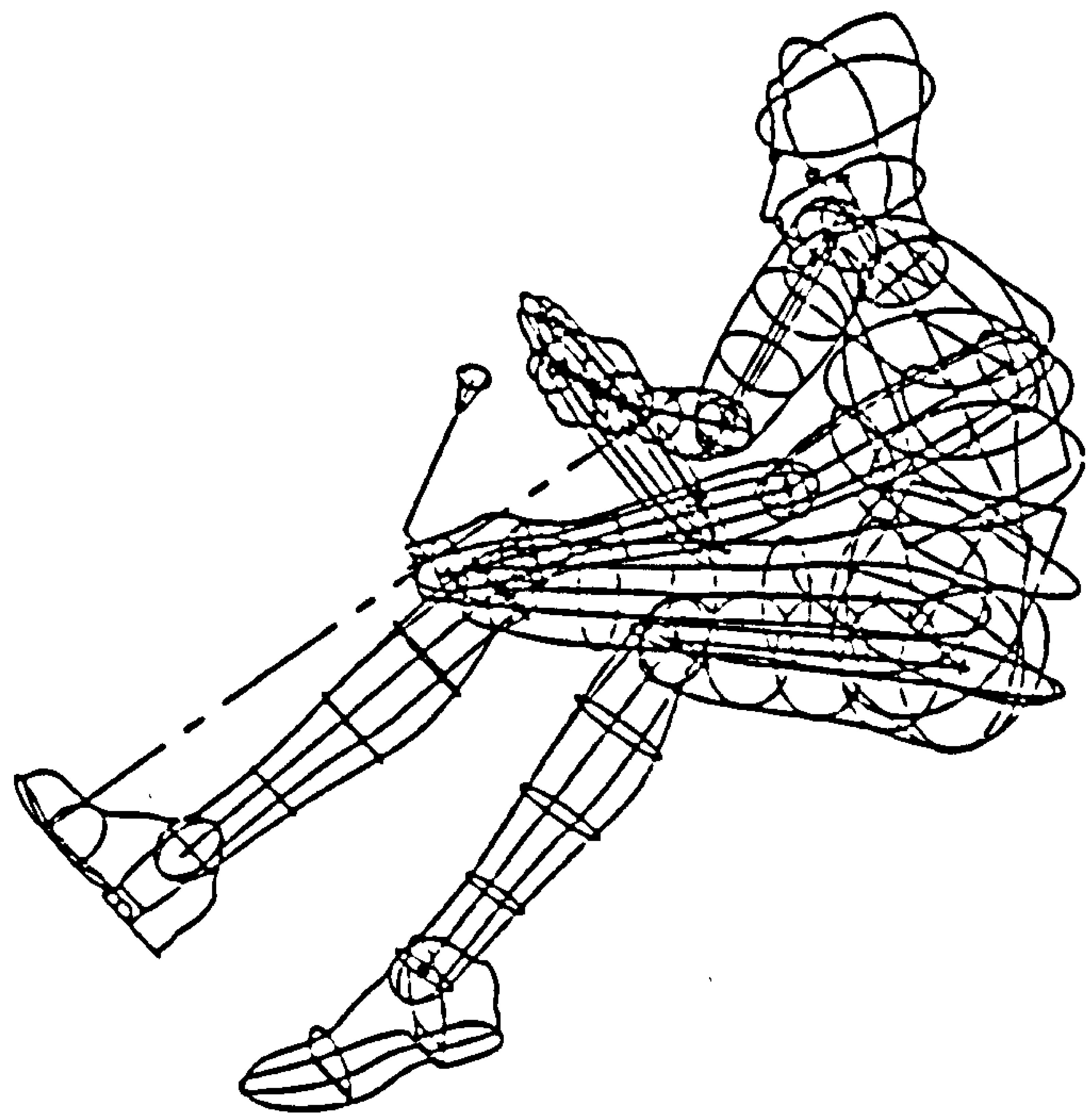


Figure 3.11 CYBERMAN (Source: Dooley, 1982).

3.6.1.12 ErgoSHAPE System

The ErgoSHAPE system was developed in 1987–1988, by the Institute of Occupational Health (Haartmaninkatul), Helsinki, Finland. ErgoSHAPE is a PC–based two–dimensional man modelling system which runs within the Auto CAD design system and has limited scope for man and workplace modelling. The man model consists of 9 joint segments and the basic postures are standing and sitting. Anthropometric man models are available in three sizes, small, average and large (5th, 50th, 95th percentiles) and with a user–specified size, for both male and female. The anthropometric data is based on Finnish, North European and North American populations but the man models can be easily scaled linearly as required. The ErgoSHAPE system consists of three parts which are: man models in a working space that can be fitted to human dimensions, biomechanical calculations, where evaluation of postural stress in manual materials handling or in static postures can be determined, and recommendation charts as design guidelines to specific work situations and workplaces (Launis and Lhetela, 1990).

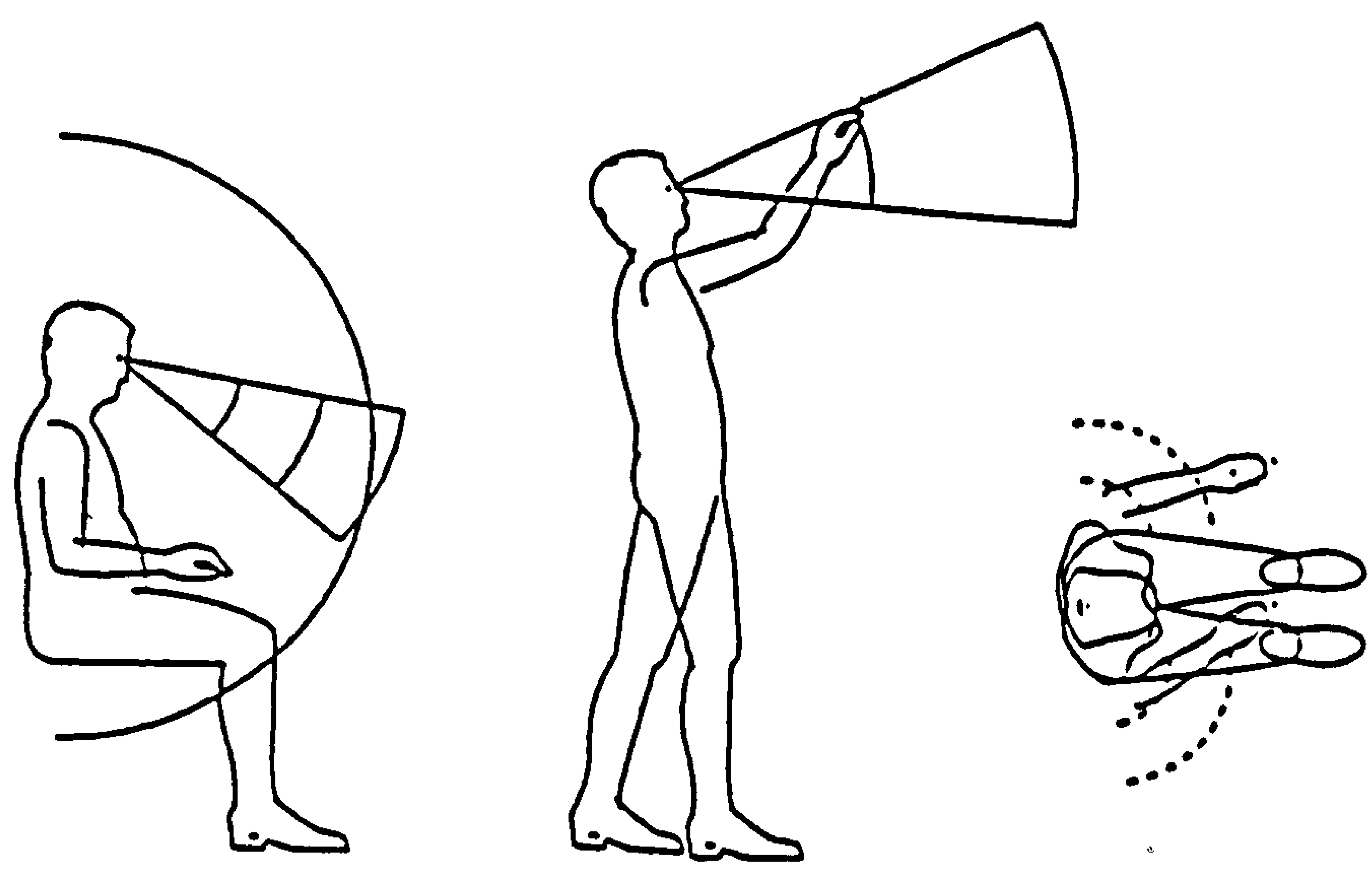


Figure 3.12 ErgoSHAPE two–dimensional man models in various postures
(Source: Launis and Lhetela, 1990)

3.6.1.13 ErgoSPACE System

The ErgoSPACE system was developed in 1987–1988, by the Institute of Occupational Health, (Haartmaninkatul), Helsinki, Finland. ErgoSPACE is a three–dimensional man modelling system, which runs on a PC computer and the AutoCAD system and is programmed in Turbo–Pascal. The system was developed to improve the restrictions of based PC computers and therefore the graphic presentations of the man model and the workplace are very simplified and of rather limited use in designing workplaces because of its simple, unnatural and unclear visual appearance. The workplace is structured as simplified wire models, lines, squares and boxes and hidden lines cannot be removed.

The ErgoSPACE man model has 17 movable joints, can be evaluated within a workplace, and in order to attain a reasonable response time, a stick model 's link structure representation is used for moving the model. The model can be 'filled' to form an ellipsoidal wireframe for evaluations. The man model database is identical to that of ErgoSHAPE. ErgoSPACE data can be saved in the DXF–transfer file format, and so can be used in other CAD programs (Launis and Lhetela, 1990).

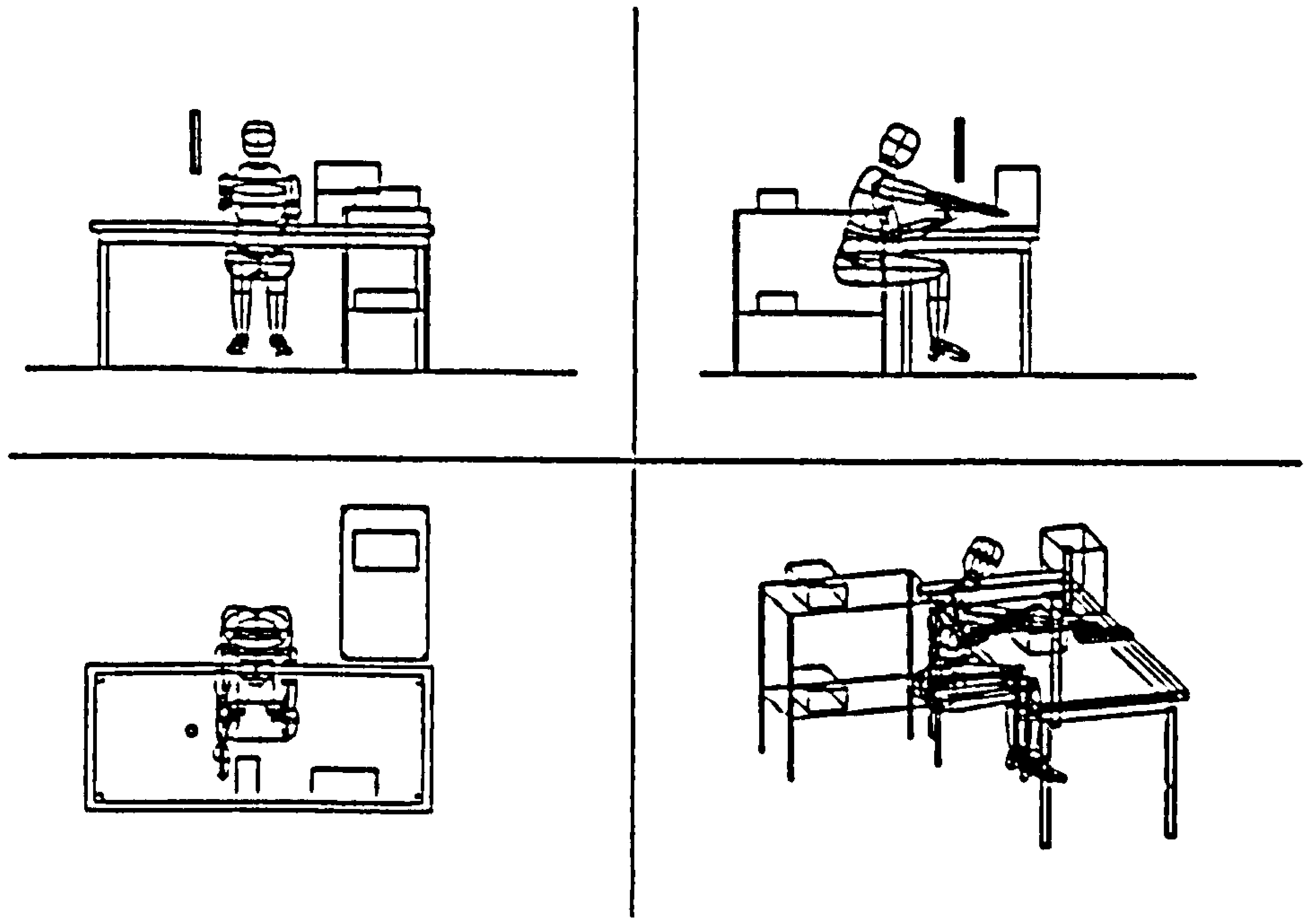


Figure 3.13 ErgoSPACE: four views (front, side, plan and perspective) of a workstation (Source: Launis and Lhetela, 1990).

3.6.1.14 ERGOMAN (ERGOnomics MAN model)

The ERGOnomics MAN model (ERGOMAN) was developed by the Laboratoire d'Anthropologie appliquee et d'Ecologie humaine of the Universite de Paris, France. The system is generated and manipulated interactively using the EUCLID 3D solid modelling system by MATRA datavision. The ERGOMAN model is similar to MAN3D. The ERGOMAN anthropometric data model is based on a huge source consisting of approximately four million individual measurements (Verriest et al, 1991).

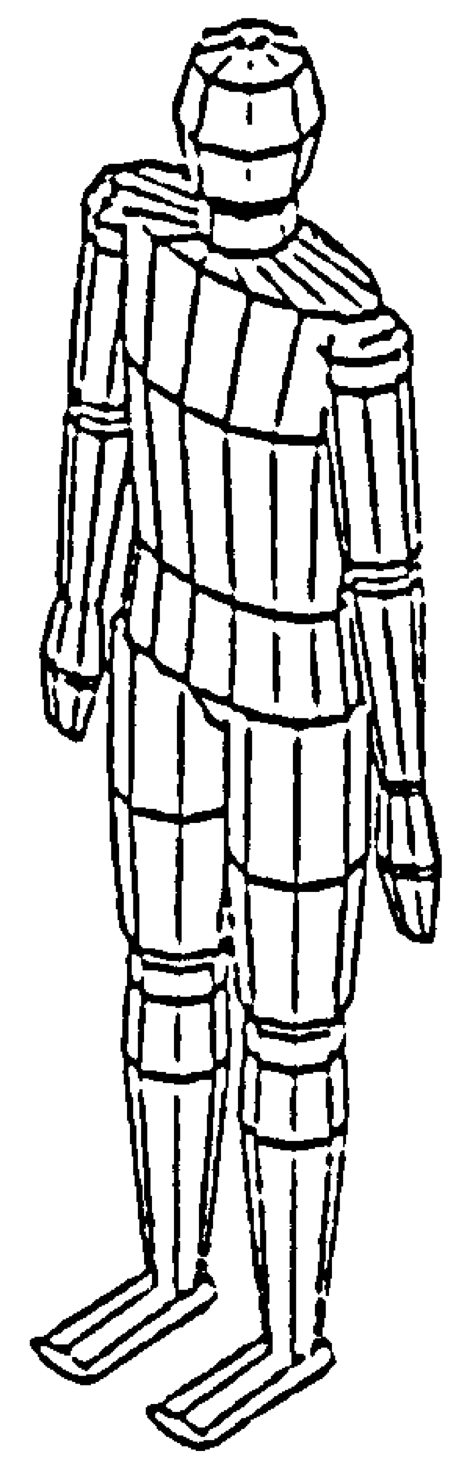


Figure 3.14 ERGOMAN model created using EUCLID CAD
(Source: Verriest et al, 1991).

3.6.1.15 FRANKY

FRANKY was developed in 1986, by Gesellschaft fur Ingenieur–Technik (GIT – Society of Engineering Technology) mbH in Essen, and has a very similar suite of facilities as SAMMIE . The three–dimensional man model has four different sizes, based upon DIN 33416, although its anthropometric dimensions can be easily changed. A detailed hand model has been modelled for close–up views. The CAD package ROMULUS is used to construct the workplace models. FRANKY can be used to assess fit, reach and vision in ways very similar to SAMMIE. The closure of the GIT in 1987 is likely to have limited the development of FRANKY and the software is not presently commercially available in the market (Elias and Lux, 1986, and Porter et al, 1993).

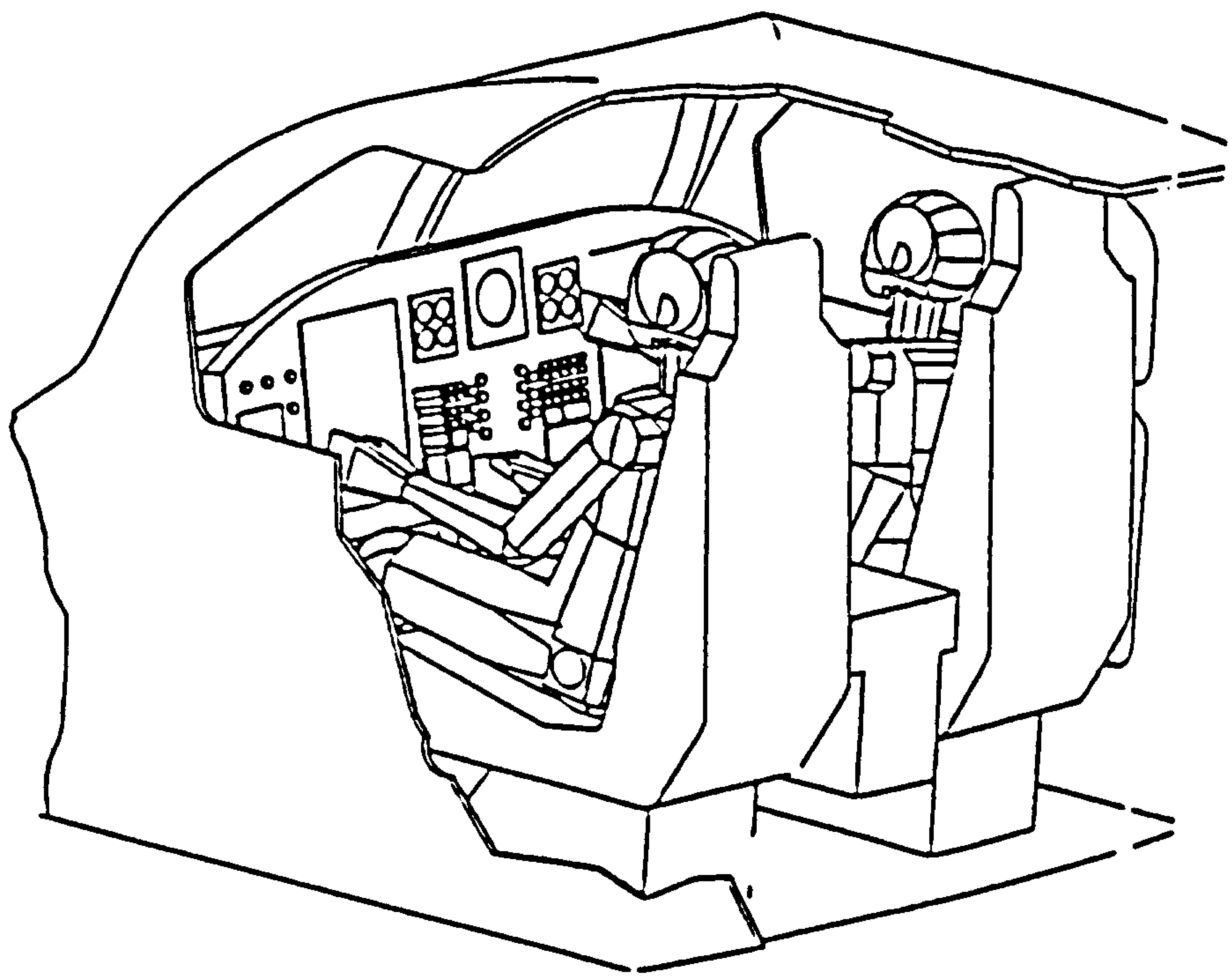


Figure 3.15 FRANKY model of pilot shows a perspective view of a cockpit with hidden lines removed (Source: Elias and Lux, 1986)

3.6.1.16 HEINER

HEINER was developed to support research work using Hewlett Packard workstations (HP 1000) at the Institute for Ergonomics of the Technische Hochschule Darmstadt, West Germany. The HEINER man modelling system is nearly the same as the SAMMIE system in appearance and can be used for reach and fit analysis, sight paths and perspective views from the man model's eyepoints. The system contains a module for representation of workplace elements composed from geometrical primitives. The anthropometric data model is based on German standards DIN33402 and is similar to ANYBODY and OSCAR (Kennedy, 1995).

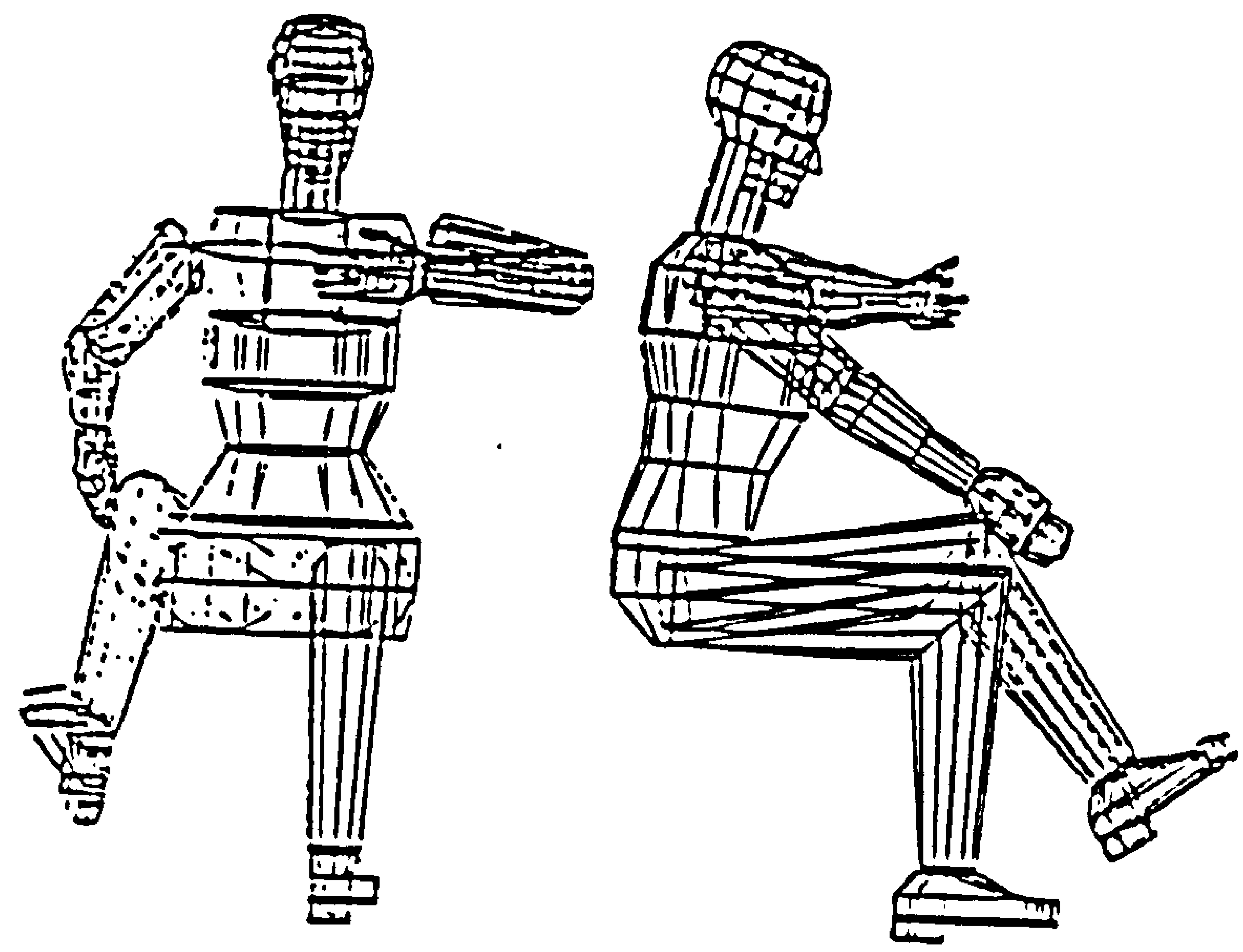


Figure 3.16 HEINER man model (Source: Kennedy, 1995).

3.6.1.17 ISM (Isohuman Simulation Model)

The ISM (Isohuman Simulation Model) was developed by Mercedes/Daimler Benz, West Germany. The system is intended for simulation models applied to car passengers and pedestrians in traffic accidents. The ISM system is not used for human workplace design but like MADYMO is used for simulating dynamic biomechanical models (Jones, 1988).

3.6.1.18 JACK

JACK was developed in 1992, by the Centre for Computer Graphics Research at the University of Pensylvania, USA. The development of JACK was based on earlier work from BUBBLEMAN, and TEMPUS. It is a very recent development and is commercially available using Silicon Graphics workstations. The man model consists of an 88 joint model with a 17 joint spine and a 16 joint hand. It is available with clothing, contoured bodies and facial characteristics. The joint angles of the man structure are calculated by inverse transformation of the kinematic chain. The anthropometric data base consists of both military and civilian populations.

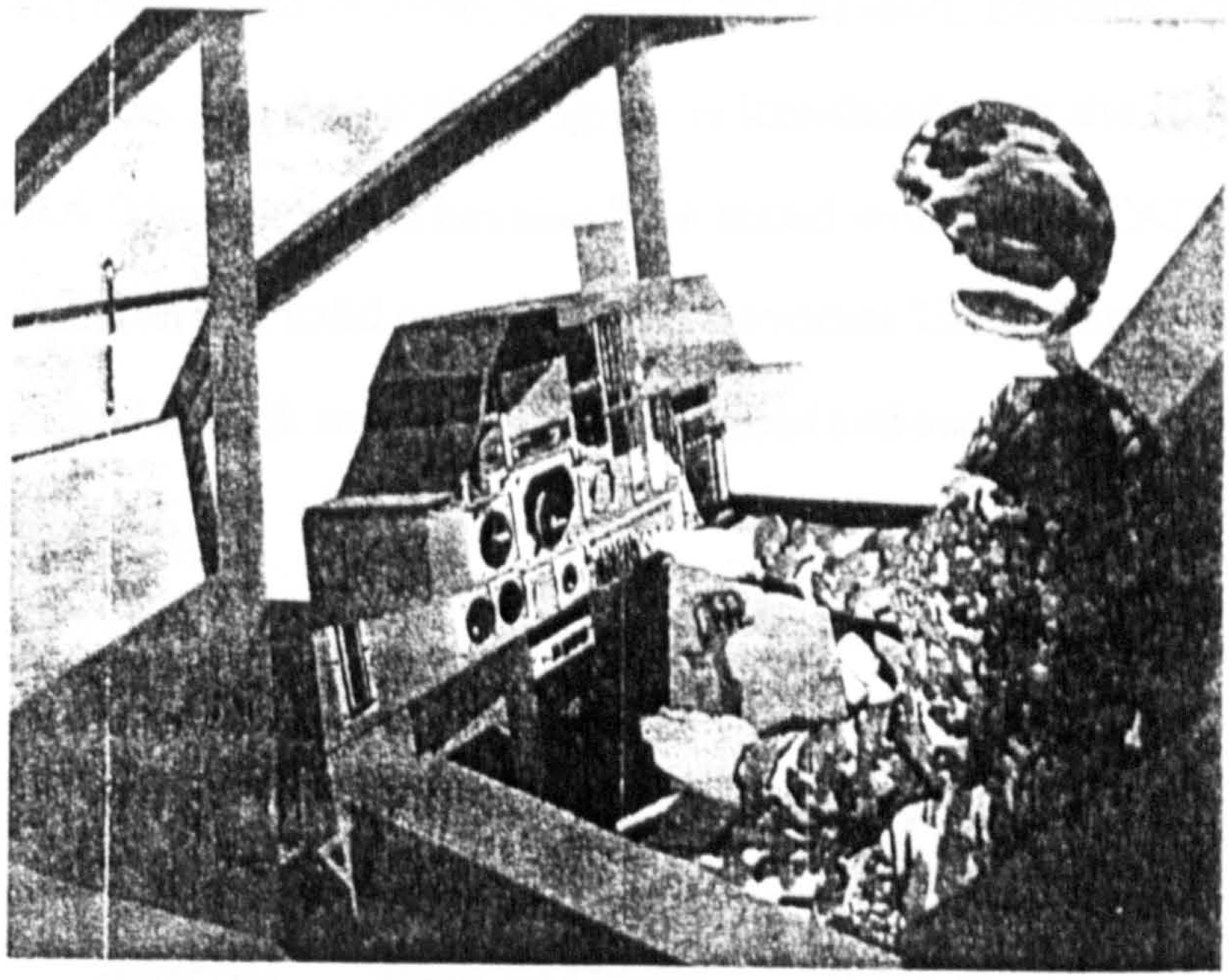


Figure 3.17 JACK man model (Source: Zhao and Badler, 1994).

The man model can be modified by selection of a given percentile stature. The hand is modelled in detail with three predefined hand grips which can be manipulated interactively. The system has facilities for importing the workplace model from other CAD systems. The man model can be manipulated interactively within defined joint mobility constraints and the limbs can be grabbed and moved by a model entity. The system performs intelligent moves of the man's stance to maintain balance, though within a limited range of biomechanical evaluations. Strength capability can be calculated based on the joint excursion and torque generated by vertical loads (Zhao and Badler, 1994).

3.6.1.19 MADYMO

MADYMO was developed by TNO, Holland. The man model is used in the automotive industry for dynamic motion simulation of car crashes and the resulting effects on the driver and passenger within the car. The output data of time based segment location and orientation is postprocessed to produce a sequence of animated pictures, which are used to assess the severity of head and body injuries (Jones, 1988).

3.6.1.20 MAN3D

MAN3D was developed in 1986, by Inrets and Renault, Ergonomics Laboratory of Sante Confort, France. The MAN3D program is interfaced with the EUCLID-IS CAD system of MATRA Datavision and has also been tested with AUTOCAD. The manikin is composed of a skeleton and solid contours which involves 23 rigid articulated links (6 leg links, 8 arm links, and 8 trunk segments including head and neck and 1 line of sight) giving 59 degrees of freedom.

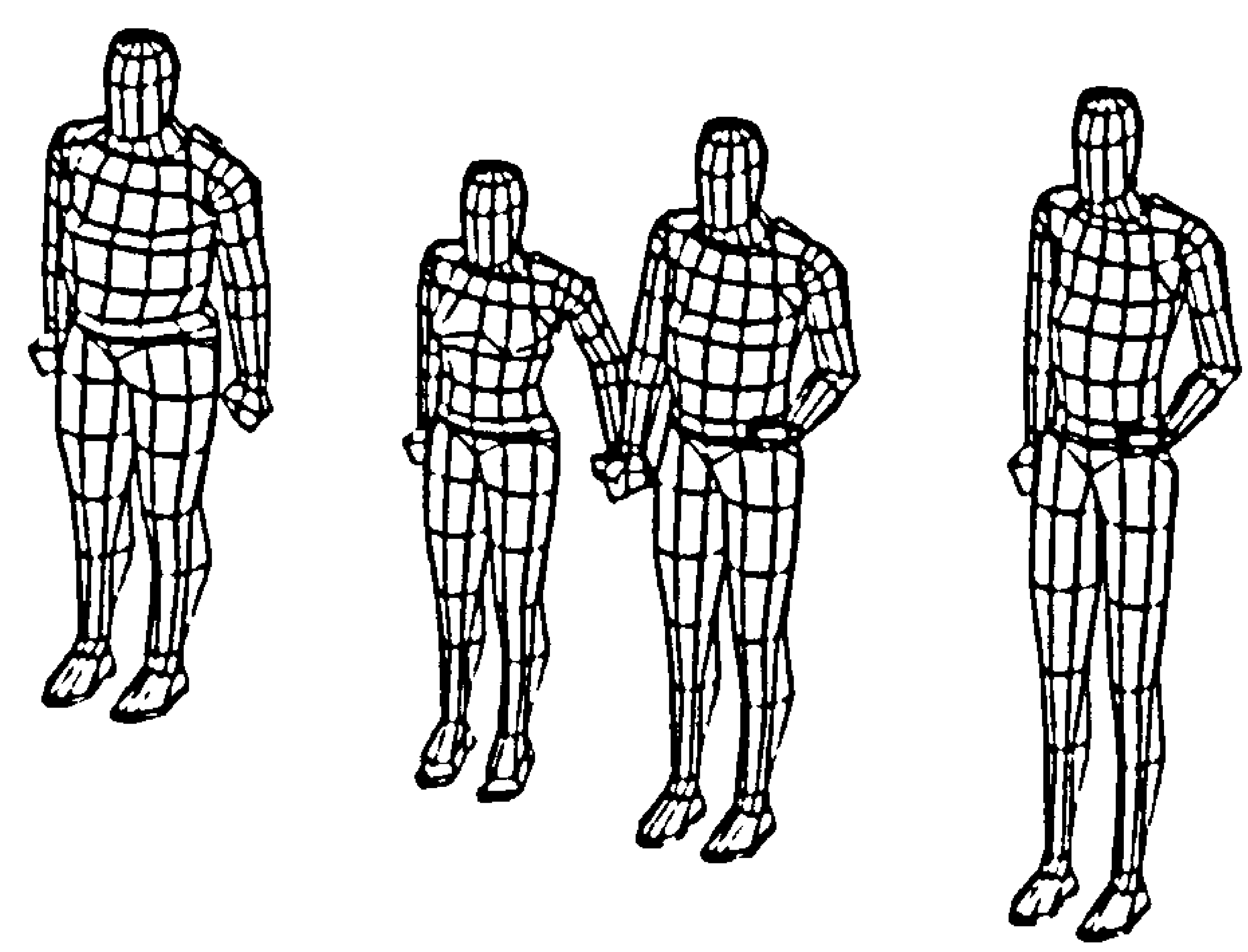


Figure 3.18 MAN3D man model of different sex, stature and weight
(Source: Verriest et al, 1991)

The joints have 2 or 3 rotation axes and a motion limitation cone is defined by spherical polygons. Solid contours are defined by closed polygonal lines, links and quadrangular facets. The anthropometric databases contain 54 basic body dimensions based on stature and weight from French populations. The first version of MAN3D was tested at the Renault technical center in the design of a car interior.

MAN3D is used to design car interior space and to locate various controls for driving. The interface between the occupant and workplace can be analysed for field of vision and control reach. MAN3D has limited functionality and capability, and is not commercially available (Verriest et al, 1991). The MAN3D model is similar to ERGOMAN.

3.6.1.21 MANNEQUIN

MANNEQUIN HUMANCAD was developed in 1989, by Humancad Development Laboratories, Biomechanics Corporation, New York, USA. The three-dimensional man model consists of 16 segments, capable of interactive movement by selecting individual joints. The anthropometric data is available in different sex, body size and nationality for adults and children from 10 countries. The man model can be changed in three degrees of detail from stick figure to a smooth Mannequin and has a full range of human movements. It can walk, bend, sit, reach and grip objects. Views that the man model can see of the environment are also available. The detailed hand model consists for four fingers and a thumb with 3 joints on each with nine predefined finger grips which can be generated interactively. The range of motion is calculated and visually charted for both hand and feet. Biomechanical torque calculations on the major body joints can be performed for any task. The workplace can be created from primitive shapes using the geometrical editor.

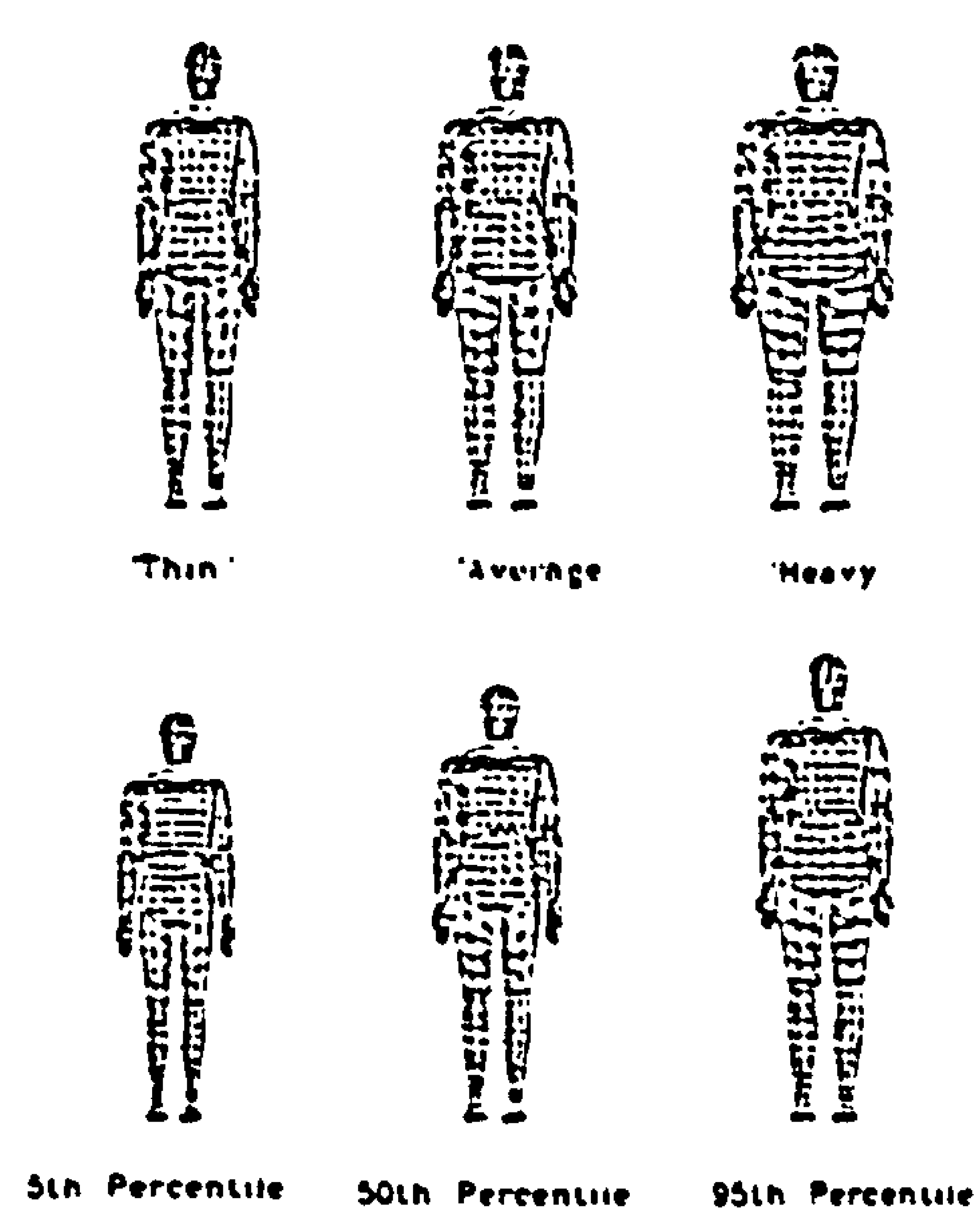


Figure 3.19 MANNEQUIN man model (Source: Walters, 1989)

The system runs on an IBM PC, or any compatible 286, 386 or 486, with a maths coprocessor, and EGA or VGA monitor. Mannequin has 3D drafting capabilities including dimensioning, scaling, shading, hidden line removal, rotation, stretching, and assembly drawing 2D objects into 3D objects. Other capabilities include animation by the creation of frames for use in animation programs such as Autodesk's "Animator" and Macro Mind's "Director". The system can import other workplace models from a range of systems using the Autodesk data exchange format (DXF) or the initial graphic exchange specification (IGES). Models can also be transferred from one system to another for improving visualisation (Walters, 1989 and 1991).

3.6.1.22 MINTAC (Man Machine INTerACtion)

Man Machine INTerACtion model (MINTAC) was developed in 1984–1985, jointly by the Kuopio Regional Institute of Occupational Health, Tampere University of Technology and the University of Oulu, Finland. This program started from the need to evaluate the difficult working postures occurring in Finnish agriculture and forestry.

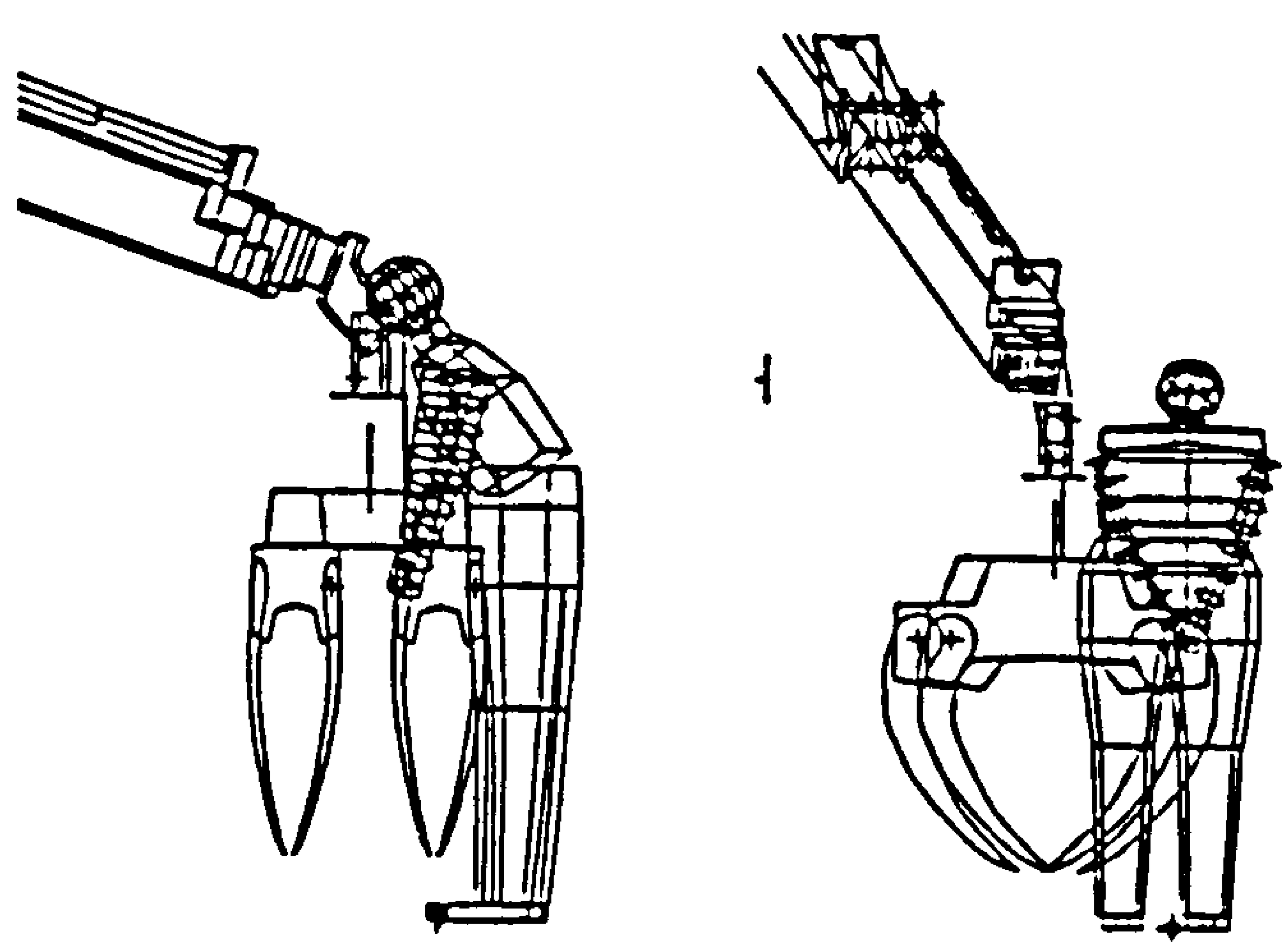


Figure 3.20 MINTAC the operator lubricating grease nipples of a tractor's grabbing mechanism (Source: Kuusisto and Mattila, 1990)

MINTAC was developed for the Computervision CAD/CAM system and consists of the Computervision Designer V hardware and the CADDS4 modelling program. MINTAC is a three-dimensional man model based upon the anthropometric database published by Dreyfuss (1967) for the American civilian population, although the model is adjusted to simulate the wearing of winter clothes in order to evaluate difficult working postures encountered in Finnish agriculture and forestry. The simple man model contains six links, lower links (one rigid block which can be selected from a choice of 13 postures), back, upper arms and forearms. Solid modelling is used for design and evaluation and can be modified and placed in the working environment. MINTAC was designed to be compatible with the OWAS working posture analysis system. MINTAC is not considered appropriate for wide-spread use because of its simplified posture and is suitable only for the analysis of heavy work (Kuusisto and Mattila, 1990, and Kuusisto, 1990).

3.6.1.23 OSCAR

OSCAR was developed in 1986, by the Hungarian Company Szamred and promoted by the Hungarian board for Industrial Design, the Hungarian Design Council in Budapest, Hungary. It has been adapted for use in Germany and Western Europe being distributed by the SOMACAD team of the Fachhochschule Darmstadt.

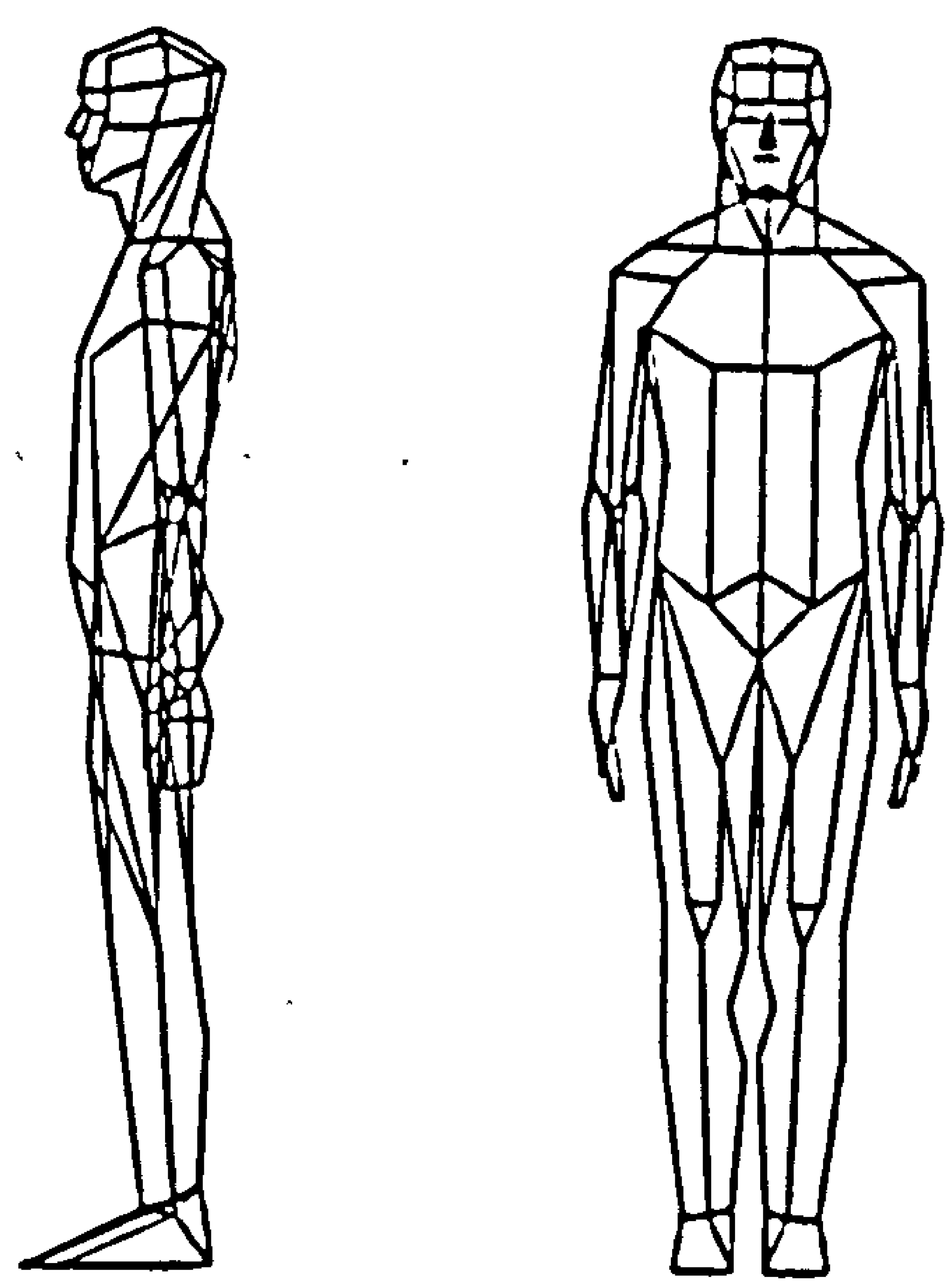


Figure 3.21 OSCAR man model (Source: Case et al., 1990)

Oscar is used on an IBM-PC or compatible machine and it is possible to construct very simple and limited workplace models. The anthropometric data model is based on the German National Standard DIN 33402 part 2 and provides a range of different somatotypes (ectomorph, mesomorph and endomorph) and is similar to

ANYBODY/CADKEY and **HEINER.** The system is widely available commercially (Case et al, 1990).

3.6.1.24 SAFEWORK

SAFEWORK was developed in 1990 jointly by the University of Quebec, Montreal and the Polytechnic School of Montreal. The system is a highly flexible, easy–to–use microcomputer program for the ergonomic analysis and design of workstations, evaluation of human–machine interfaces and the development industrial products.

The SAFEWORK man model was developed on an IBM–AT PC compatible. Three–dimensional biomechanical models with 14 body segments are generated from an anthropometric database. It is a small and transparent system which leads the user, with simple questions, to choose the appropriate model for the situation. Each situation has different elements, e.g. towing or pushing force, use of one or both hands, lifting a weight, slippery floor, etc. (Fortin et al, 1990).

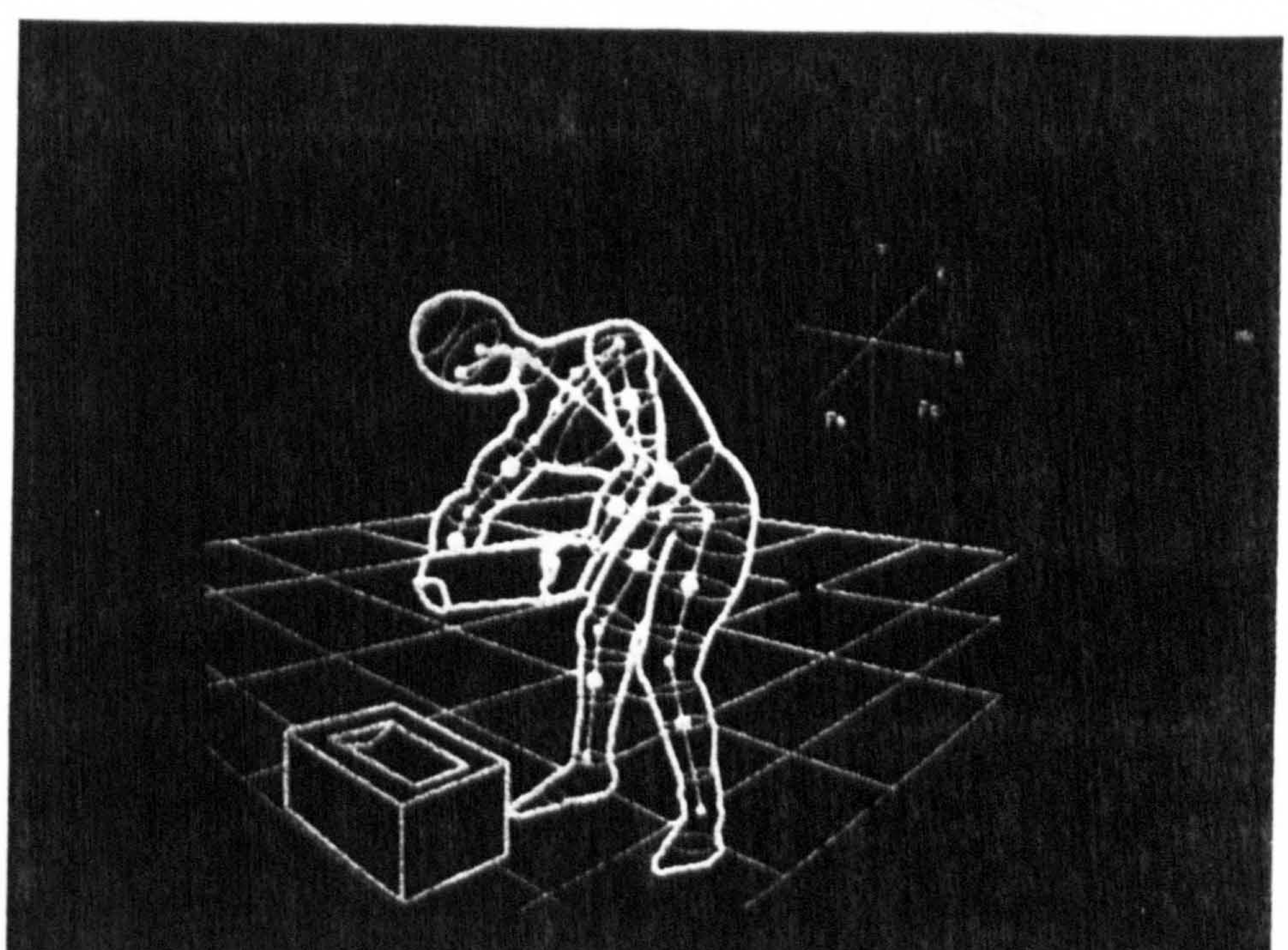


Figure 3.22 SAFEWORK man model in a workstation (Source: Fortin et al, 1990)

3.6.1.25 TEMPUS

TEMPUS was developed in 1987, by NASA, Lockheed Engineering and Management Services, and the US Army Human Engineering Laboratory, USA. TEMPUS was designed and implemented using a system called POSIT to investigate articulated figure positioning by the method of multiple constraints. POSIT was used on a Silicon Graphics Iris 3030 workstation, for real-time visual feedback. For interactive input, two devices are used: a mouse and a six-degree-of-freedom sensor – a 3 SPACE digitizer system (from Polhemus Navigation Sciences Division, McDonnell Douglas Electronics Company), then simply called the Polhemus or Tempus.

The man model consists of 17 joints and 48 degrees of freedom. TEMPUS has six degrees of freedom within a metre region of space at a sample rate of approximately 40 Hz. The man model can be interactive in 3D space in a reasonable kinematic fashion (Badler et al, 1987).

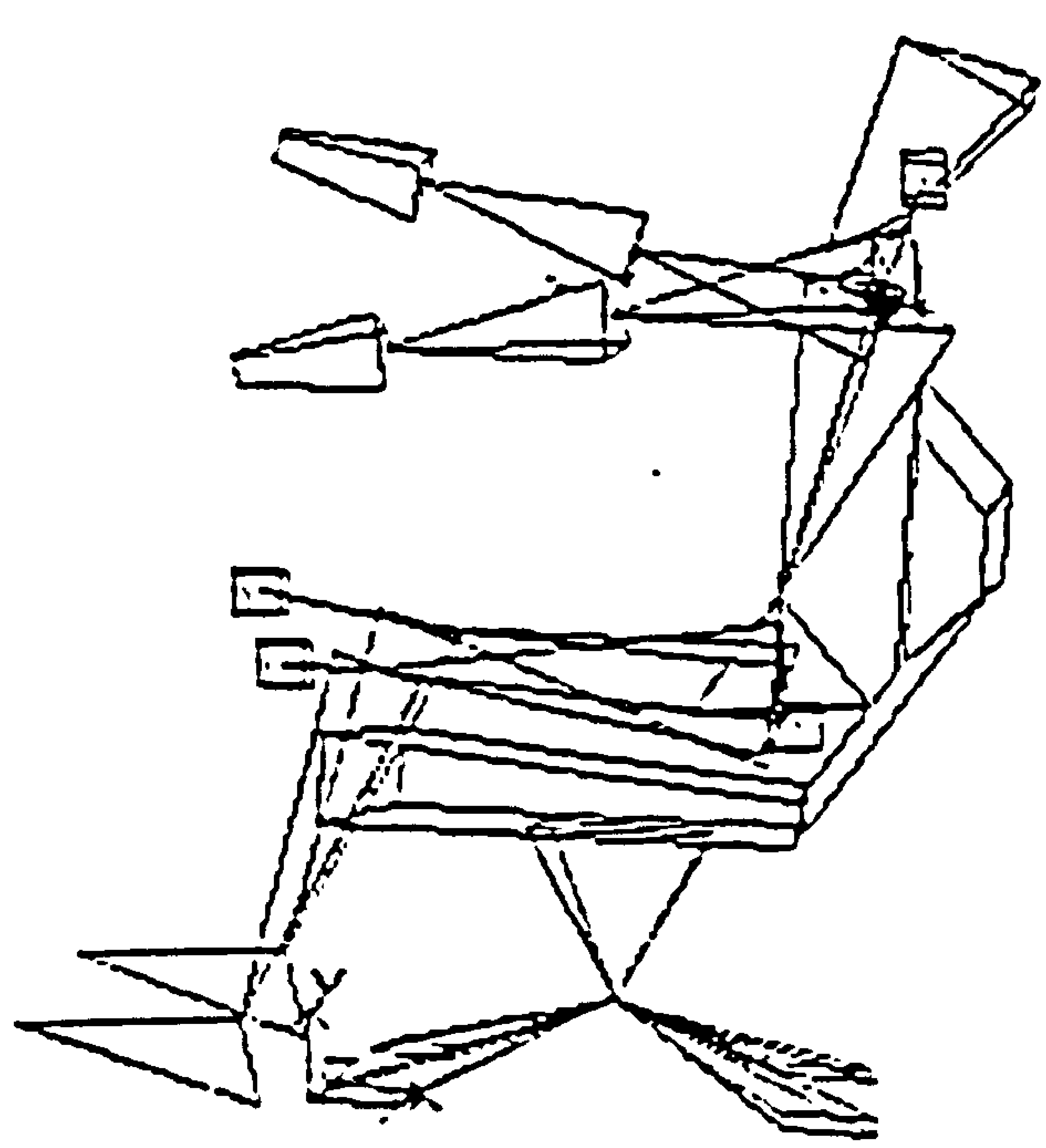


Figure 3.23 TEMPUS man model (Source: Badler et al, 1987)

3.6.1.26 WERNER

WERNER was developed in 1987 by the Institute of Occupational Health at the University of Dortmund, Germany. WERNER was written in the C language, and implemented on an Atari ST as a tool for workplace design on PC computers. The GEM window manager is used for interactive evaluation, where the three–dimensional data processing is done by internal routines. It is also available for the IBM PC and the micro VAX under Ultrix with X Windows. The anthropometric data is based on the German National Standard (DIN 43116, 1985), and figure. 3.24 shows the man model used in the context of the ergonomics assessment of an assembly workplace. It is also available as a conventional template or stencil.

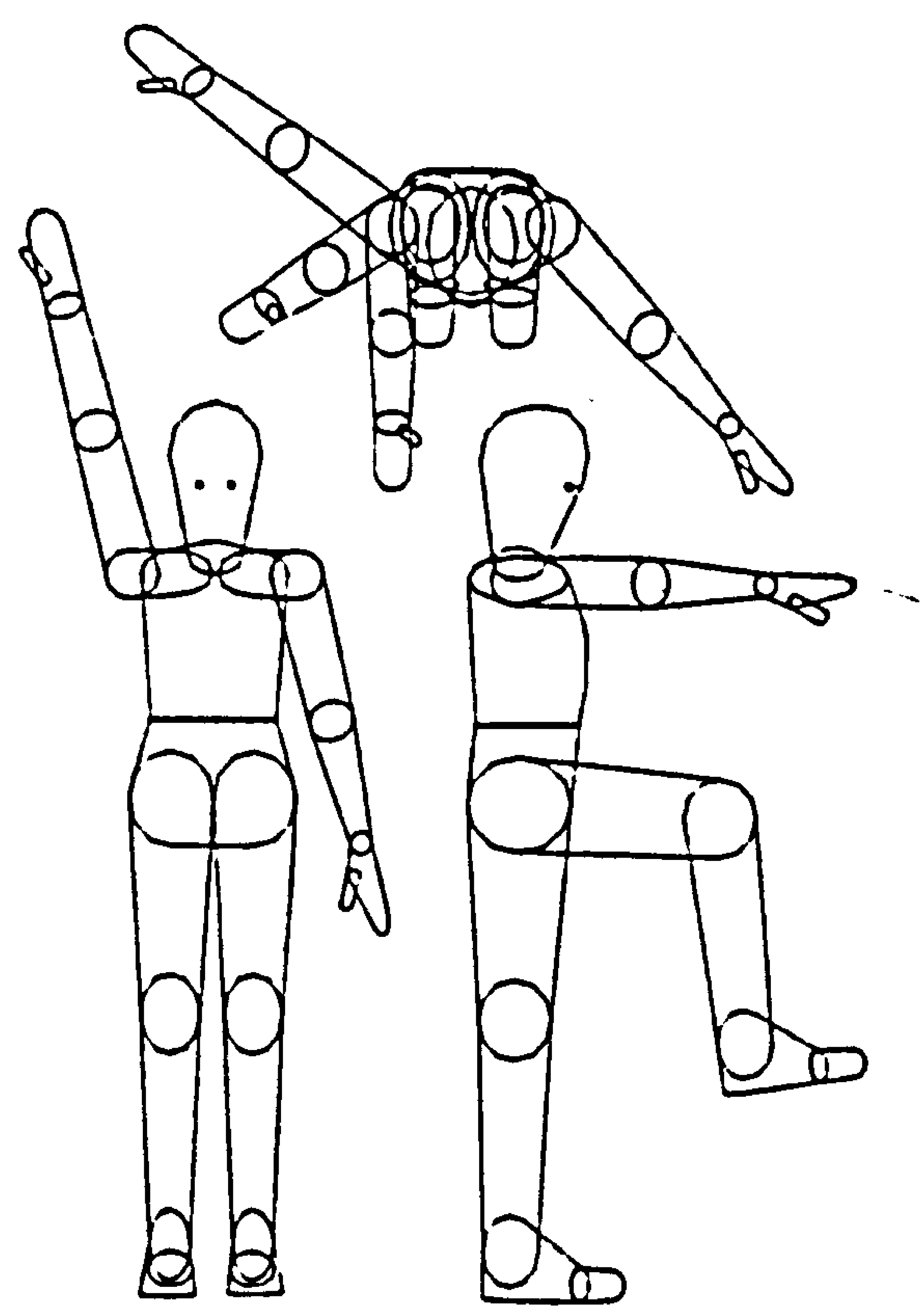


Figure 3.24 WERNER man model (Source: Kloke et al, 1990)

WERNER is a three–dimensional man model based on standard CAD programs and consists of 19 joint segments. Using a rigid skeleton, each segment is defined by some simple solid, mostly ellipsoids, and located in a fixed or changeable position. A convex hull outline is shown as the final projection, so as to define a silhouette of the man model.

WERNER has an external representation language using a Lisp-like format and also an extended Lisp computer language like AutoCAD's AutoLisp, to provide its man modelling and workspace features (Kloke, 1990).

3.6.1.27 SAMMIE (System for Aiding Man-Machine Interaction Evaluation)

SAMMIE (System for Aiding Man-Machine Interaction Evaluation) has been developed since the late 1960's, originally at the University of Nottingham, and more recently at Loughborough University of Technology. It can be considered as the longest established, earliest and most widely used computer man-modelling system on the market. SAMMIE is a computer aided design system that was designed for general computer man modelling to assist in design, analysis and evaluation of human workplaces rather than for specific applications. The man model has a total of 21 rigid links with 17 pin joints, and three-dimensional flesh contours surround this model. Variations in both percentile and flesh contours (thinness, muscularity, fatness) are allowable, using male and female data (Case et al, 1990, and Porter et al, 1993).

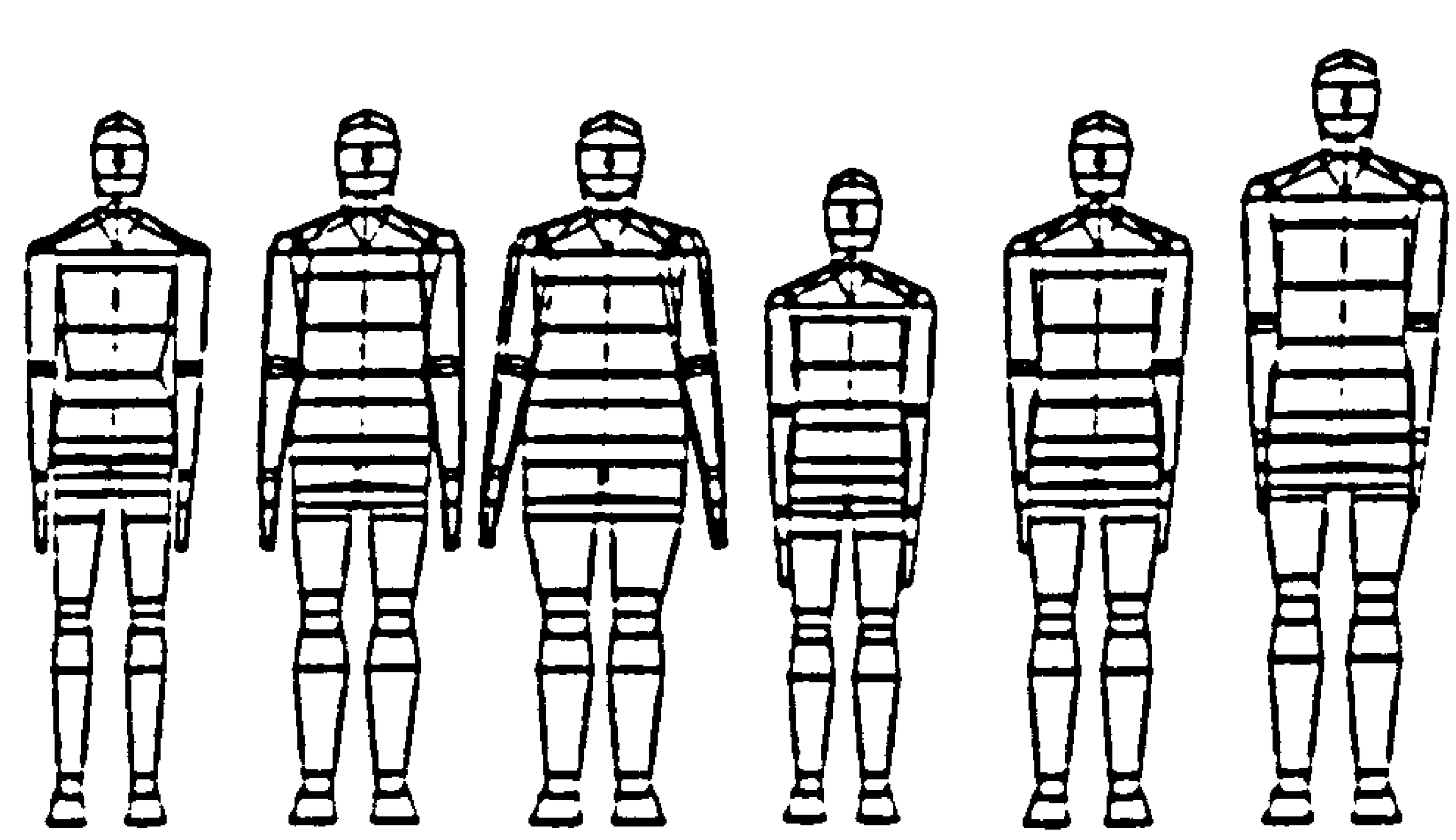


Figure 3.25 SAMMIE man model in a range of somatotype and stature
(Source: Case et al, 1990)

Three-dimensional solid models of items of equipment such as machines and consumer products or workplaces such as the passenger compartments of cars, aircraft cockpits and computer workstations are built using a simple solid modeller which has similar functionality to many other such modellers.

The man model is an integral part of the solid model and thus all the viewing and manipulation facilities available in the solid model also operate on the man model. The man model can be made to represent individuals or a part of a user population by control of its anthropometry (joint–to–joint or link dimensions), somatotype (flesh shape) and mobility (constraints to joint motion) (figure 3.26). Other facilities within the man model are provided as aids to evaluating reach, fit, vision, work tasks etc.

With experience of using SAMMIE the author can comment that the SAMMIE computer man modelling is user friendly in contrast to some other computer aided design programs. SAMMIE is only a tool or an aid, allowing the simulation of man–machine interaction to assess the validity of a design with regard to ergonomics factors (figure 3.26) (Fetter, 1980 and 1982, Dooley, 1982, and Okey, 1989).

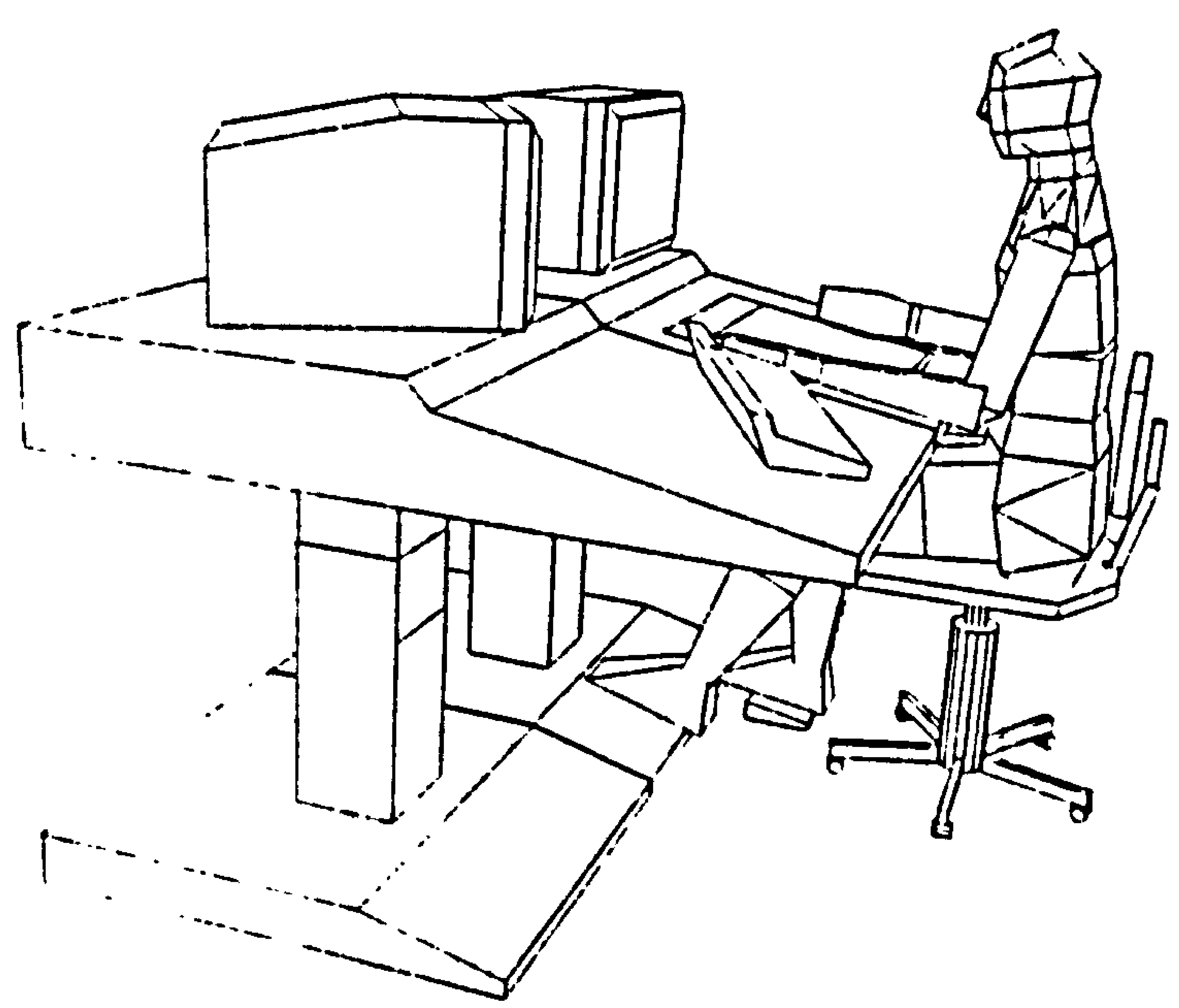


Figure 3.26 SAMMIE man model in the workstation
(Source: Case et al, 1990, and Porter et al, 1993)

3.7 Ergonomics Design Using CAD Systems and Man Modelling Systems

Traditional ergonomics design methods used by ergonomists, engineers, and designers for the specification of measurement and dimensional information for equipment and workstation design include the use of standards and published reports, in-house recommendations and guidelines, anthropometry, two-dimensional plastic manikins (full size or scaled to be used in conjunction with engineering drawings), mock-ups, user questionnaires, and user trials (Porter et al, 1993).

Man modelling CAD systems have proved to be most useful tools for designers, engineers, ergonomists for the physical design of the man-machine interface. With man-modelling CAD systems available today it is quite possible for a design to progress from the concept stage right through to the manufacturing stage without any recourse to full size or even scaled evaluation models. This process helps to reduce the timescale of product design considerably, although questions must be raised concerning the quality of design. (Porter et al, 1993).

Computer software packages are available that are capable of modelling the workplace and operator, whereas traditionally the ergonomics of operator packaging will be performed by drafting methods followed by subjective evaluations on mockups. Man modelling CAD systems do offer significant advantages to the designer with respect to issues such as fit, reach, vision and task related posture.

3.7.1 Computer Aided Ergonomics Design Systems as a Tool for Designers and Engineers

The importance of CAD for industrial design as well as all other design disciplines, is that one is drawing with mathematics rather than with lines on paper. The result of this is that whatever the designer draws is instantly measurable in 2D or 3D, depending on the system used. The designer is also able to scale and layer information so as to concentrate that information into one centre of decision. Apart from dimensional information, the designer may also apply tests to the mathematical data-base with which he is drawing in order to establish physical performance such as testing, validation, and verification. Having created his mathematical file or data file, either two-dimensional or three-dimensional, he may also output drawings of any scale.

CAD techniques are still in their infancy and even if the promise of many of the computer manufacturers is in excess of the performance of the machines that they sell, the implications to the designer of working interactively with a mathematical model of his design are very far reaching.

3.8 Expert Systems in CAD and Man Modelling Systems

This chapter has shown that many efforts have been made to produce CAD man modelling systems in the past few years. These reflect the traditional approach to design and have objectives and methods that are similar to any engineering analysis system. The ergonomics design area is about applying knowledge to the design problem and hence knowledge based expert system techniques and geometric man modelling are appropriate to the objectives of this research. There are very few existing CAD man modelling systems that use knowledge based techniques (those that do include COMBIMAN, CREW CHIEF and ErgoSPACE).

Many of the system described are potentially suitable for carrying out ergonomics evaluations of vehicle interior design. The SAMMIE system was selected on the basis of:–

a) Availability. Not only was the system available, but its active use in a variety of teaching, research, development and consulting situations made a fund of experience available.

b) Geometric Domain. Several of the alternative systems are based on inadequate geometry either through being only two-dimensional or three-dimensional wireframe. The geometric inquiry and manipulation requirements are better met by the solid representation available in SAMMIE.

c) Accessibility. For successful communication with an expert system access into the data structure of the geometric modeller must be available. This is provided in SAMMIE through the use of command processing (effectively running SAMMIE from an external program rather than interactively).

This research attempts to include expert system techniques within SAMMIE to provide a general method of applying knowledge to design problems. The underlying techniques are discussed in the next chapter.

CHAPTER 4

ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEMS

4.1 Introduction

This chapter introduces the background of Artificial Intelligence (AI) and Expert Systems (ES) and definitions of Artificial Intelligence. Artificial Intelligence activities can be divided into the five subsections of natural language, computer vision, improved human interfaces, exploratory programming and expert systems. The background and features of expert systems are described and compared with conventional computer programming methods.

Expert systems are knowledge based systems that contain a knowledge base, an inference mechanism/engine with methods for forward, backward or mixed chaining, and a user interface. Frequently explanation and graphical facilities are also provided. Knowledge acquisition or elicitation, by the knowledge engineer from the domain expert is a crucial aspect to the successful implementation of an expert system. Various knowledge representation methods are available including formal logic, semantic networks, production rules and frames.

This chapter also discusses the motivation behind expert system development and the available expert system languages, shells and tools. The application of expert systems in general is described as an introduction to the proposed expert system for CAD and man modelling (CAED) system.

4.2 Background of Artificial Intelligence and Expert Systems

In the 1650's, the English philosopher Thomas Hobbes put forward the interesting concept that human intelligence consists of symbolic operations and that everything in life can be represented mathematically. These concepts directly led to the notion that a machine capable of performing mathematical operations on symbols could imitate human intelligence. This is the basic concept behind artificial intelligence, and consequently

Hobbes is sometimes referred to as the grandfather of Artificial Intelligence (Badiru, 1992). In more recent times Alan Turing pioneered AI studies in the late 1940's in the United Kingdom (Simons, 1984).

In the late 1950's, the term Artificial Intelligence was coined by McCarthy at MIT, USA. He was involved with the development of the LISP system that laid the foundations for this new research area, which initially was concentrated in the USA (Costed, 1987). In the early 1980's the Alvey directorate attempted to promote research activities and applications in AI technology with a program to revitalise the information technology industry in the United Kingdom. Then later, in 1990, the Japanese researched into fifth generation computer systems, building computing systems that could converse with their users in natural ways, through human language, speech and visual images (O'Shea, 1987). Currently this is the highest goal in the application of AI technology, where the rapidly growing technology promises much for the future.

4.3 Definition of Artificial Intelligence

Artificial Intelligence (AI) is a technology to understand the nature of human intelligence through the construction of computer programs that imitate intelligent behaviour. The computer program solves a problem, or takes a decision on the basis of a description of some situation, meaning that the program itself finds the method to be used in order to solve the problem or to reach the decision, calling on a wide range of embedded "reasoning" processes. The reasoning is captured from human expertise and the machine is used primarily for calculation (Bonnet, 1985).

Artificial intelligence is an exploration into the nature of human intelligence or cognition. The interesting part is how computers can be used to solve specific problems. Figure 4.1 shows some AI research activities conducted during recent years. The five most active areas of AI are (1) natural language, (2) computer vision, (3) improved human interfaces, (4) exploratory programming, and (5) expert systems, and these are described briefly below:–

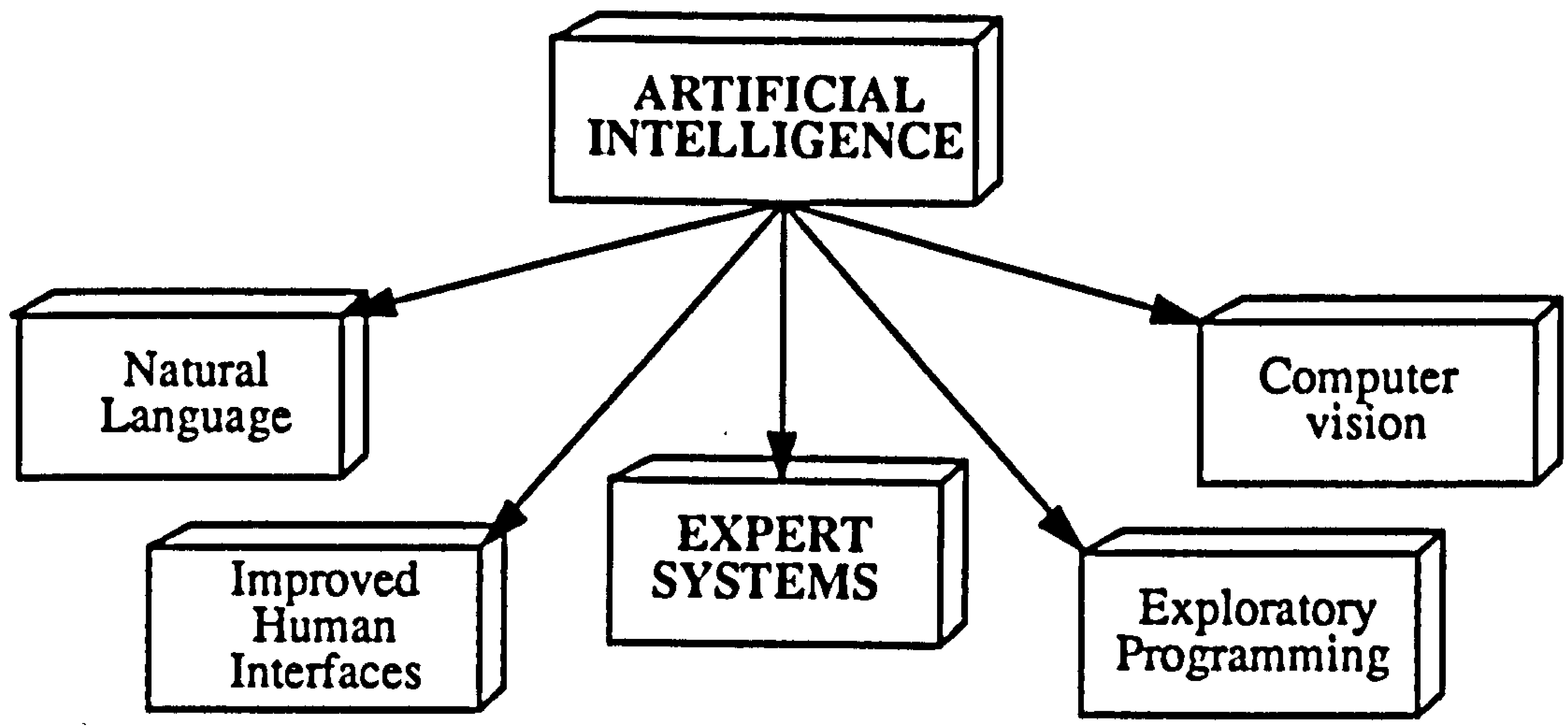


Figure 4.1 The five most active areas in Artificial Intelligence.

4.3.1 Natural Language

A natural language is any language that humans speak, e.g., English, French, or Japanese. Some AI researchers are trying to develop computer hardware and software that will allow computers to interact with people in such a natural language through the development of interfaces for existing data bases (Harmon et al, 1988), and various areas of research such as machine translation, speech analysis, grammar and text generation (Badiru, 1992).

4.3.2 Computer Vision

This deals with research efforts involving scene analysis, image understanding, and motion derivation (Badiru, 1992). Example applications of computer vision are found in pattern recognition (detecting flaws in materials and products) and part orientation for automatic handling.

4.3.3 Improved Human Interfaces.

A third area in which AI concepts and techniques are being actively developed is in improving existing commercial computer practice concerning the design and development of better interfaces. It is also hoped that alternative interface techniques will become more widely understood (Harmon et al, 1988).

4.3.4 Exploratory Programming

Exploratory programming refers to the application of AI concepts and techniques to developing large applications, and includes new programming languages and environments, modularity, and incremental development, which can all be used to increase conventional programmer productivity. Some might include "automatic programming" as using AI techniques by specifying the goals of the program and leaving it up to an automatic programming system to generate most of the specific code for the application (Harmon et al, 1988).

4.3.5 Expert Systems

Expert systems is the branch of Artificial Intelligence that has received most of the attention. These problem solvers are called "expert systems" to suggest that they function as effectively as human experts at their highly specialised tasks (Harmon et al, 1988). It also involves the development of computer software to solve complex decision problems (Badiru, 1992).

An expert system is a program that manifests some combination of concepts, procedures, and techniques. These techniques allow people to design and develop computer systems that use knowledge and inference techniques to analyse and solve problems. More detailed descriptions and discussion of expert systems follow later in this chapter.

4.4 Background of Expert Systems

A variety of definitions for expert systems are given in the AI literature although the meanings are very similar. A typical definition is:

Goodall and Beynon–Davies defined expert systems in the simplest manner possible: *"An expert system is a computer system, that performs functions similar to those normally performed by a human expert"* (Goodall, 1983).

"An expert system is a computer system that uses a representation of human expertise in a particular domain in order to perform functions similar to those normally performed by a human expert in that domain" (Beynon–Davies, 1991).

"An expert system is a computer system that operates by applying an inference mechanism to a body of specialist expertise represented in some knowledge representation formalism" (Beynon–Davies, 1991).

The first definition, however, conveys little of the nature of an expert system. How does an expert system work? The second definition moves us closer to this idea. It defines expert systems in terms of mechanisms for representing human expertise. Here we are moving from a human–centred to a more technical–centred definition for expert systems. The technical direction reaches its conclusion in the third definition. Here we define an expert system in terms of its major technical components: a mechanism to perform inferencing and a mechanism to represent knowledge. In this chapter the detail of the technical aspects of expert systems is expanded.

4.5 Expert Systems Features

An Expert System is a computer program that uses knowledge and reasoning techniques to solve problems that normally require the services of a human expert. An expert system may imitate the external behaviour of an expert (i.e., gathering information and producing solutions to problems), or it may attempt to closely model the internal mental processes of the expert (KES, 1990).

The fundamental feature of an expert system is that it contains large amounts of knowledge or expertise known as the *Domain Knowledge*. However, there is common agreement on the distinctive features which expert systems have including (Forsyth, 1984):–

1) An expert system is limited to a specific domain of expertise. A specific expertise area has to be chosen and all of the knowledge about that area has to be collected. This involves

interviewing experts in the field, collecting and extracting information from standards, technical books, journals, etc.

2) They are typically rule-based. Knowledge elicited from experts in the field or from other sources is coded in to a form of rules. They incorporate *'rules of thumb'* or *'heuristics'*, which are generally true, but perhaps not under all circumstances.

3) They can explain their train of reasoning in a comprehensive way. The computer should be able to answer questions such as **'what'**, **'why'**, **'which'** and **'how'**. For example, 'How have you arrived at that conclusion?'

4) Facts and the inference mechanism/engine are clearly separated. In this way, if the facts need to be changed or added to, it can be done quickly without any need to change the inference mechanism.

5) They are designed to grow incrementally. As more knowledge is gathered, this can be coded and added to the knowledge base. In this way, knowledge about a particular domain can be grown.

6) They deliver advice as their output, not definitive data. They will recommend a certain course of action rather than specific parameters, although parameters may be included in the advice.

The key words of expert systems are knowledge and reasoning. The objective of an intelligent problem-solving system (like a human) is to cut out blind and random searching. The computer system has to exploit the same advantage that the human expert has over the novice – i.e. he processes expertise or organised knowledge. Knowledge is about facts, rules of inference and solution strategies. However, up to now the biggest problem has been getting the knowledge from an expert into a machine.

4.6 Comparison of Conventional Computer Programs and Expert Systems

Expert systems can be distinguished from conventional computer programs as, the former has a reasoning mechanism fed from a knowledge base and its output consists of numbers, possibilities and advice. In contrast a conventional program manipulates logic presented by code to give answers as output data, as shown in figure 4.2. Expert systems can perform no better than the expert that provides the knowledge on which they are based and they can therefore exhibit " human frailties". Expert systems, on the other hand, are 'declarative' – no instructions on how to solve the problem are given. Instead the knowledge is represented as a collection of facts accompanied by a collection of rules for using them. An inference mechanism, or 'inference engine', finds the appropriate solution to the problem from the facts and rules.

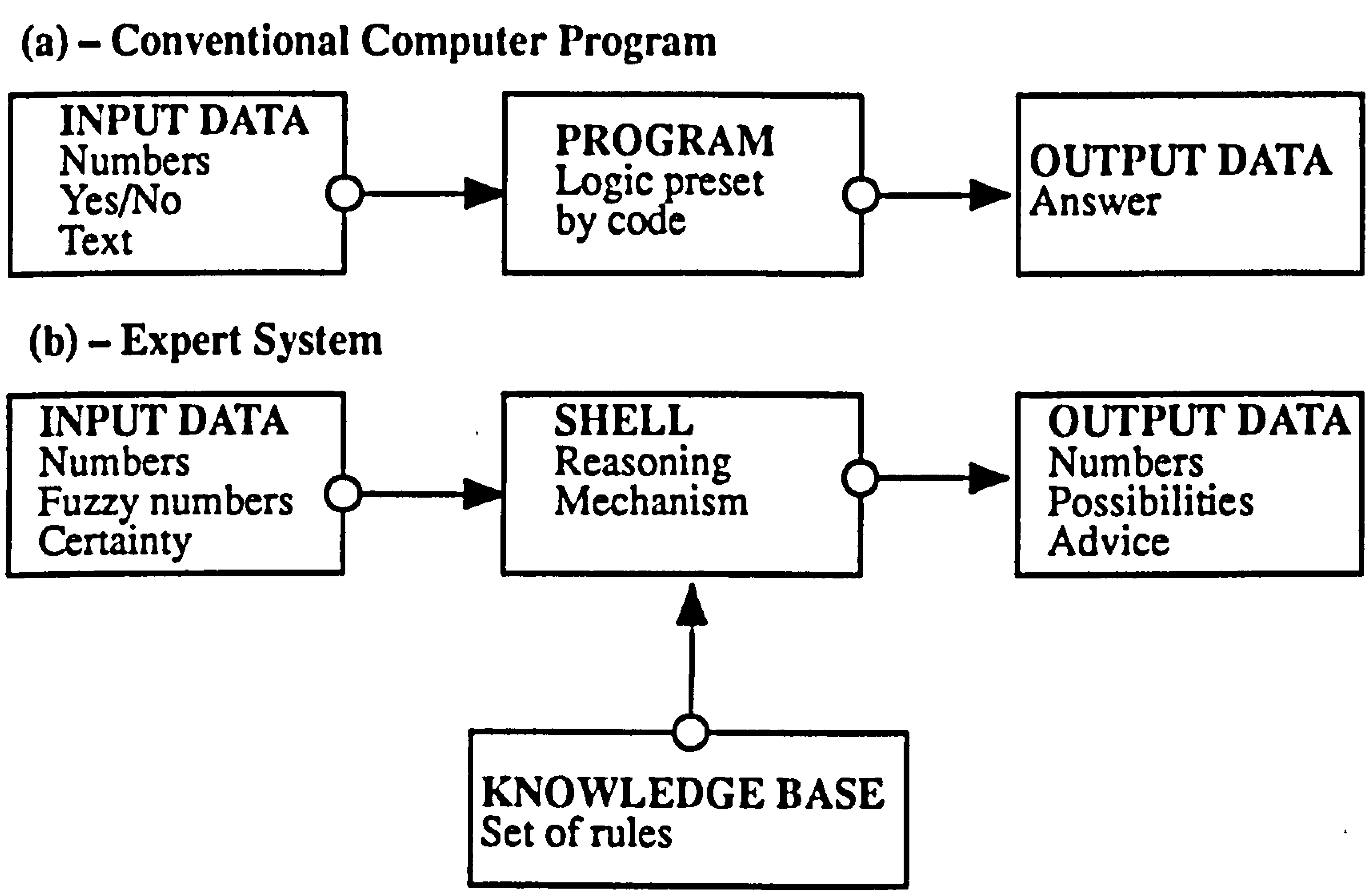


Figure 4.2 – Schematic comparison of conventional computer programs and expert system (Source: Forsyth, 1984).

Conventional computer programme languages such as PASCAL, FORTRAN and C have a rich set of data structures for non-numerical computations, for example, lists, arrays, strings, words, procedures, processes, and so on. Their data structures are type-free. So, for example, lists can contain arbitrary objects, and code can be treated as data. However, AI languages, as well as having some features of conventional languages, have the capability of making decisions, solving problems, and justifying their course of action. They are also embedded in environments (programs) and these language features and their environments together help the programmer to cope with complexity.

The following is a brief description of some characteristics which can be used to compare a conventional computer program or data processing program with expert system software (Forsyth, 1984):–

Conventional Computer Program

1) Deterministic and follows a predetermined sequence for every problem.

2) Constructed primarily of linear relationships.

3) Typical objectives are to compute values, store constants, and retrieve data.

4) Deals with universally accepted processes, for example the value of a log.

5) Follows established mathematical rules.

Expert System Software

1) Creates new decision tree for each new goal.

2) Deals with arbitrary 'symbolic' expressions, for example, conceptual, temporal, and spatial relationships.

3) The objective is to give advice based on the predicted behaviour of observable objects and events.

4) Focuses on empirical associations that describe the behaviour of concepts and events; the information base is derived from engineered knowledge supplied by human experts, or accumulated data.

5) The accumulated expertise is expressed as a series of descriptive attributes and contingency relationships. Thus expert systems rely on symbolic processing based on heuristic reasoning.

The comparison of conventional computer software (1) and expert system software (2) can perhaps be abbreviated as (Forsyth, 1984):–

(1) **Data + Algorithm = Program**
e.g. **IF the input is add-a-record**
THEN execute the add-a-record process.

(2) **Knowledge base + Inference = System**
e.g. **IF it is raining**
THEN use an umbrella.

4.7 Basic Principles of Expert Systems

This section outlines the basic principles and the structure of expert systems. Later discussion concentrates on the common issues in building an expert system, such as the knowledge acquisition and knowledge representation. Expert systems are knowledge based systems that contain a knowledge base, an inference mechanism/engine component and a user interface. The knowledge base contains knowledge that is required to solve problems which usually require a measure of (human) intelligence. Knowledge Engineering is that discipline which acquires the knowledge from an expert, or some other source, and represents this in the same form in the expert system knowledge base. Knowledge acquisition and knowledge representation are at the forefront of these activities. The following sections describe the structure of expert systems, inference mechanisms/engines and user interfaces.

4.7.1 Expert Systems Structure

An expert system basically consists of three components, a knowledge base, an inference mechanism or engine, and a user interface, figure 4.3, shows a typical structure and functional integration of the components. The main components are the knowledge base of facts, data and rules, an inference engine, and a user interface for various tasks including filing of required data or retrieval from a particular file, comparing data and numerical calculation. However, there is an important prior stage in the development of an expert system, i.e. knowledge elicitation or knowledge acquisition. All these components of an expert system will be described and discussed below:-

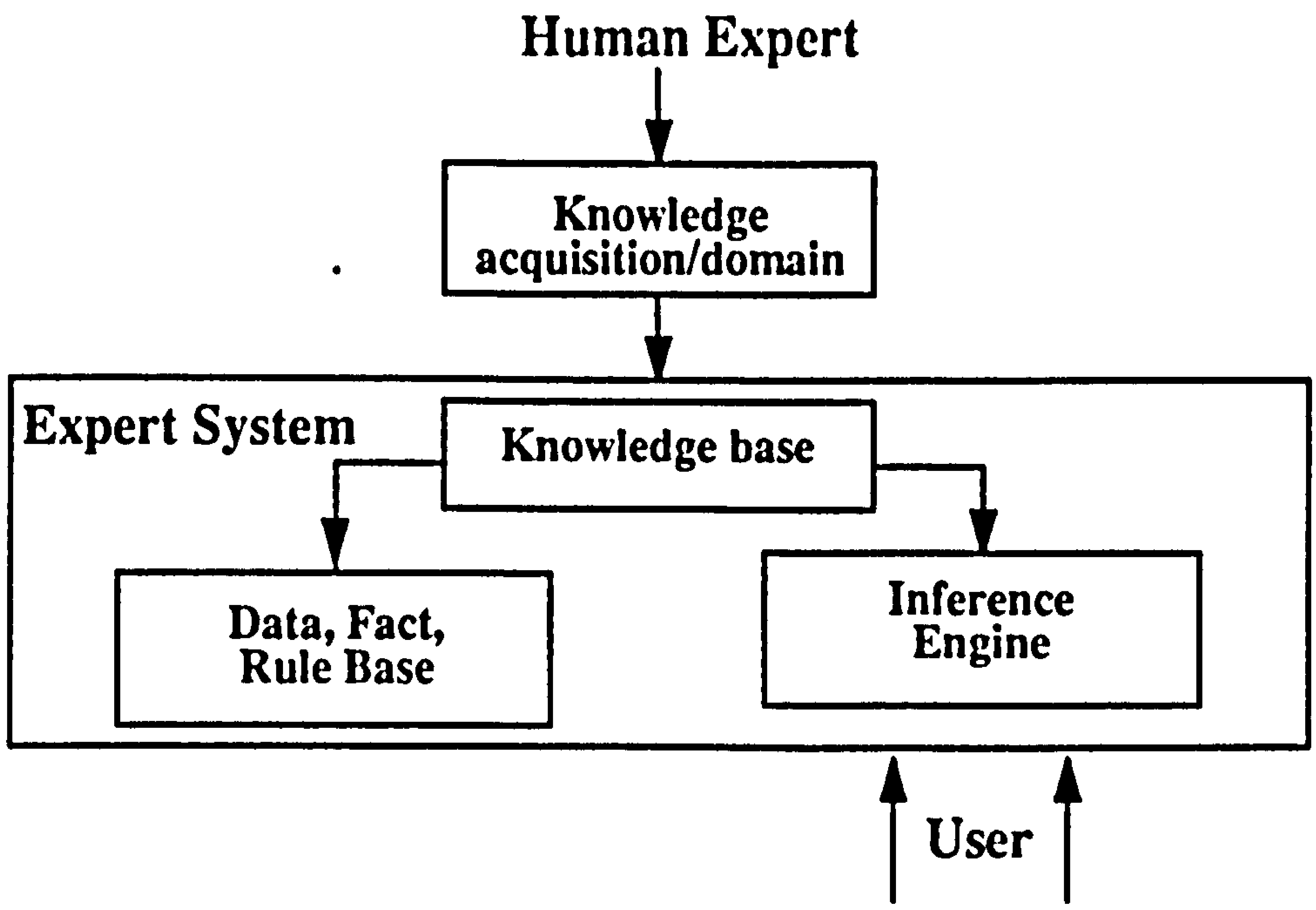


Figure 4.3 The structure of expert systems.

4.7.1.1 The Knowledge Base

The knowledge base is where knowledge is stored as statements about facts and rules and possibly numerical values. As more information becomes available the knowledge base grows. If information about a particular problem is missing from the knowledge base, attempts are made to obtain the missing information, for example, by requiring input from the user. However, the task of the knowledge engineer is to select appropriate means of storing such information symbolically. It is appropriate therefore to examine some of the methods of knowledge representation commonly used (Alty, 1985).

The knowledge base is a store of information collected from the subject domain, typically in the form of facts, data, and rules. The user creates a file that contains statements describing a certain model. These statements are error – checked and compiled. The model may then be executed. The model consists of three sections: hypotheses, findings and decision rules.

Findings are the facts (e.g. symptoms, test results) and hypotheses are the conclusions drawn from these facts and input data. After an assertion about a finding has been made, the corresponding rules are evaluated. Such a system is able to explain its reasoning.

The facts and heuristics are normally represented in sets of rules in the knowledge base, containing the expertise or knowledge of the human expert in the domain of interest. The rules give a picture of how a human expert would actually tackle the problem in his expert domain. Rules are also used to represent certain scientific formula or specific patterns that can be deduced from any standard data or graphical representation of certain empirical relationships. For example, in the production systems approach all rules are written in the form (Alty, 1985):–

Rule 1:	**IF condition**	(conditions or antecedents)
	THEN actions	(conclusions or consequences)

Each rule has two essential parts, the conditions or antecedents and the action (or conclusions or consequences). The conditions may be a single condition, or may be a Boolean expression. Conclusions may also be compounded.

This is important since the inference mechanism actually compares or tries to match the conditions of the rules against the facts in the knowledge base. The **actions** part of the rules resembles the actions that were to be taken by a human expert. Typically, they include actions such as creating new facts from the inference made, adding or deleting unwanted facts from the knowledge base, giving suggestions, etc. The separation of knowledge and control permits the run–time facility of providing on–demand explanation and justification.

The rules set in the knowledge base can be regarded as an open–ended set, meaning the extension of the set is achieved by merely adding new rules (Buchanan and Duda, 1983). The integrity of the rule set must be considered as new rules are added, especially when the knowledge base becomes considerably larger. The following possible conditions that can happen are to be avoided:–

1) Infinite chaining – this occurs when rules have interdependencies that result in a circular sequence of firing

2) Introducing contradictions between two or more rules. This is quite usual for systems with very large rule bases.

Introducing **meta rules** (also known as rules about rules) and the breaking down of a large rule base into smaller bases would reduce the problems outlined above. Meta rules are rules that govern the mechanism upon which rules are selected, fired, terminated, etc. The grouping of rules into smaller rule bases enables fewer rules to be scanned by the inference engine during the reasoning process. The advantage of this is two fold. First, the firing efficiency will be increased since there are fewer rules to search in a particular rule base and secondly, the chance for infinite chaining to occur is reduced considerably, since fewer sub–goals can be set up.

4.7.1.2 Inference Mechanism/Engine

The knowledge base contains no information on how to find the rules that apply and when to evaluate them. However, the inference mechanism is used to ***'fire'*** an individual rule when the rule is selected for evaluation. Such mechanisms for selecting which rules to fire and in what order use two alternative strategies, ***forward chaining*** and ***backward chaining***. Forward chaining is a line of reasoning that starts from known facts or ***"data"*** and fires rules to infer conclusions or ***"hypotheses"***, while backward chaining starts with a conclusion and then fires rules which can establish that conclusion. These mechanisms also have to deal with uncertain data. There are many ways of dealing with uncertainty such as fuzzy logic, Bayesian logic, multi–valued logic and certainty factors (Forsyth, 1984 and Black, 1986). The following set of rules are used to explain both chaining mechanisms.

IF E AND C THEN F
IF A THEN C
IF B THEN D
IF D THEN A

The following subsections will describe the details of these two inference mechanisms.

4.7.1.2.1 Forward Chaining

Forward chaining works by taking all the known facts and examining each rule to see if all the IF parts of the rule are true. With this new extended list of true facts the process is repeated until either a suitable goal has been derived or no more information can be discovered. It is also known as data–driven inferencing since the method will not work unless there is some data available to drive the inference procedure – when there is no (more) data the inference engine stops. In the previous examples:–

Assume the fact that B and E are true.

IF E AND C THEN F	Rule ignored since not all IF part are known
IF A THEN C	Rule ignored since IF part not known
IF B THEN D	Rule is fired since B is true and D is added to the knowledge base
IF D THEN A	Rule fired and A is added to the knowledge base
IF A THEN C	Rule is fired and C is added to the knowledge base
IF E AND C THEN F	Rule is fired. No more rules to prove so the inference engine stops.

4.7.1.2.2 Backward Chaining

Backward chaining starts with a goal expression whose value must be determined, and the mechanism searches the knowledge base for rules that allow it to determine a value for the goal. This then serves as the new goal and the search continues. In the process, the system is using rules and facts from the knowledge base and may also seek the values of certain expressions from the user. This strategy is also called goal–driven inferencing. In the previous example:–

Consider the system is trying to determine or prove F.

The first rule states that F can only be true if both E and C are true. E cannot be proved by using any other rules, so it must be asked directly of the user. The next clause to be proven is C. C can be proven using the second rule. To prove C we need to prove A. To prove A we need to prove D in rule four. This recursive idea of trying to prove one thing by proving others is repeated one last time as B is needed to prove D. There is no rule that can prove B so once again the user is asked for the value of B (Harmond and King, 1985).

4.7.1.3 The User Interface

Another component of an expert system is the ***User Interface,*** that is the communications link between the program and the user. At its most basic level, it is only what the user sees on the video monitor, however its importance should not be underestimated. The interface is particularly important for successful expert systems which require an opinion of what the user sees, hears and feels about the subject in question. The operation of an expert system is extremely important, and poor design of a user interface module has been a recurring criticism of interactive computer systems. An expert system interface has to provide a good dialogue and comprehensive explanation facilities.

An effective human/machine interface is essential in operating an expert system. A survey suggests (Goodall, 1983) that, of a typical expert system's code, 44 percent deals with user input/output. There are two types of facility that are usually provided to a user during consultation with an expert system; explanation and or graphical facilities.

4.7.1.3.1 Explanation Facilities

Expert systems need to be able to explain why certain conclusions were reached.. The degree to which the expert system is accepted by the user will depend significantly on its ability to explain its reasoning. Explanation facilities are particularly important in domains where the expert system is asked to make judgments. The basic explanation facilities provided by an expert system should include why it came to a particular conclusion and how it is going to achieve a particular goal. In the 'why' explanation the rules which have been used to derive the conclusions should be presented to the user, whereas in the 'how' explanation the rules that will be used to achieve the goal should be presented to the user.

Explanations in expert systems typically consist of tracing the rules in the reasoning process. Such tracing only provides the user with a logical proof of the correctness of conclusions made by the system. However, explanation of some of these conclusions or interrogation during consultation may be represented by graphical means (Goodall, 1983).

4.7.1.3.2 Graphical Facilities

Another class of tool is concerned with display of knowledge and inference processes, so that the user can better understand them. Graphical representations may be used to maximise the understanding of questions during interrogation, or the reasoning of a solution. For example, in vehicle interior design, the system may ask the user about the legislation, rules and regulations on which he/she would like to base the design. At the same time ergonomics standards and information on the 'geometric reasoning' aspects of the design working practice may be displayed to facilitate the design.

Several techniques can be used to try to establish a successful user interface:–

1) The programmer must word the questions extremely carefully to help the user to supply just the correct information.

2) Figures and diagrams are an invaluable aid to explanation. Using software drawing packages, the system can be designed to present diagrams to the user for consultation.

3) Comprehensive explanation facilities should be provided for the user. This explanation may consist of the following:–

   i) Comprehensive text explaining each question or rule. This may be retrieved by the user on request.

   ii) Announcement statements notifying the user which parameter is going to be investigated next.

   iii) Clear explanations of how the system arrived at a decision or recommendation.

The 'user–friendly' nature of the user interface is very important, particularly for expert system applications where the user is likely to be more important than usual due to the interactive nature of expert systems.

In order to include the above components in an expert system, the system developer has to use either an AI language or a procedural language, but such programming can be time–consuming and requires extensive programming skills. Another approach is by using an expert system shell, which normally includes the format for the user interface facilities. The flexibility of formulating the user interface using expert system shells is enhanced if these shells are capable of being interfaced or embedded into other systems or software. This is one of the reasons why the KES expert system shell is used in this research.

4.7.2 Knowledge Acquisition/Elicitation

Buchanan and Duda define knowledge acquisition as: *"The transfer and transformation of potential problem–solving expertise from some knowledge source to a program"* (Buchanan and Duda, 1983).

Knowledge acquisition is implemented in multiple phases. The phases involve finding a good knowledge engineer, establishing the characteristic of the knowledge to be acquired, choosing the domain expert, and the transfer/acquisition of knowledge (Badiru, 1992).

Expert systems deal with knowledge and expertise. The problem of how to acquire the required knowledge is among the early issues to be solved in building expert systems. In the AI literature, the study of how to acquire knowledge is classified as ***Knowledge Acquisition.*** Knowledge acquisition is the most time consuming stage in the development of expert systems. There are many ways of acquiring knowledge from experts. Knowledge acquisition is the process of identifying, extracting, documenting, and analysing the knowledge and information processing behaviour of domain experts in order to define an expert system's knowledge base and the requirements of the inference engine (Levine, 1986). The regularity of knowledge or information is very important since it enables us to predict what will happen next or to explain how and why something has happened. The laws of science are the most regular forms of knowledge that one can think of.

There are many different ways for the methods or systems 'author' to acquire or engineer the knowledge. The process of translating knowledge from the source knowledge to the expert system is usually carried out through a dialogue between one or more ***domain experts*** and the knowledge author or ***knowledge engineer.***

4.7.2.1 Domain Expert and Knowledge Engineer

The Domain Expert is a specialist in a particular chosen field who has the ability to apply that knowledge to solve problems and make decisions. Also, Domain Knowledge is the area of expertise that is used to formulate a response to the design task.

Knowledge engineers are first concerned with identifying the specific knowledge that an expert uses in solving a problem. Initially, the knowledge engineer studies human expertise and determines what facts and rules of thumb the expert employs. Then the knowledge engineer determines the inference strategy that the expert uses in an actual problem solving situation. Finally, the knowledge engineer develops a system that uses similar knowledge and inference strategies to simulate the expert 's behaviour.

A knowledge engineer assumes the responsibility of modelling human reasoning and expertise in the form of a computer program. The variety of techniques include written documentation, past examples of human performance, domain experts, and the knowledge engineer's own expertise (Badiru, 1992). The knowledge base engineer uses expert system tools to create an expert system.

Sources of knowledge depend greatly on the accuracy and consistency of any knowledge base and on how knowledge was acquired in the first place. It is therefore important to develop effective methods of knowledge acquisition. Typical sources of knowledge are described in section 4.5.

There are a number of guidelines which assist in knowledge acquisition from domain experts. Olson and Rueter, (1987) have reviewed the literature and reported that there are two classes of knowledge acquisition methods; direct and indirect methods. The ***direct method*** asks the expert to report on knowledge he/she can directly articulate. This set of methods includes interviews, questionnaires, simple observation of the task performance, protocol analysis, interruption analysis, etc. In contrast, ***indirect methods*** do not rely on the expert's abilities to articulate the information that is used; the data sources are extracted from journals, research publications and standards reports from various international standards organisations such as BSI, ISO, SAE, EEC, CEN, ADR, FMVSS etc. magazines, technical books, empirical data and other data or knowledge bases.

4.7.3 Knowledge Representation

Another important issue in expert systems is representing and using knowledge. This terminology has been given the following meaning by Rao and Jain, (1988).

1) Knowledge is a collection of descriptions, assimilation procedures, and problem-solving methods that first serves to organise and summarise observations, and secondly supports a useful degree of problem-solving ability.

2) A symbol stands for something else by reason of association or convention.

3) A representation is a particular kind of symbol, in which the symbol's structure is perceived to correspond in some way with the structure of the thing for which the symbol stands.

4) A representation of knowledge combines data structures and interpretive procedures that, if used correctly in a program, will lead to ***Knowledgeable*** behaviour. There are several data structure classes for storing information in computer programs, and procedures have been developed that enable intelligent manipulation of these data structures to make inferences.

Once knowledge has been acquired, the next stage is to record it. Recording knowledge into machines requires some representation method especially when it is to be used intelligently. The knowledge has to be coded in an organised format, recognised by the inference engine, be easy to debug, etc.

Four of the most important and common knowledge representation techniques are: *formal logic, semantic networks, production systems andframes* (Rich, 1981, Barr and Feigenbaum, 1982).

4.7.3.1 Formal Logic

The knowledge base is a store of information collected from the subject domain. The model consists of three sections: hypotheses, findings and decision rules. Findings are the facts (e.g. symptoms, test results), and hypotheses are the conclusions drawn from these facts and data. After an assertion about a finding has been be made, the corresponding rules are evaluated.

Formal logic was one of the first knowledge representation schemes used in AI. It has two important and interlocking branches. the first is the *axioms*, which are used to represent *what can be said* and the other is the *rules of inference*, which are used to determine *what are the possible inferences*, that can be made if certain axioms are true (Barr and Feigenbaum, 1982).

In formal logic, a properly formed statement is known as a *proposition* and can have a truth value of true or false. An example of a proposition is *"The system is working"*. A proposition is taken as a whole sentence and is not to be broken down into its constitutent parts. Thus, *The system is stable* can be assigned to be true or false and one should not assign separate terms such as *The system* to be true or false since it is not possible to assign such truth values to them.

More complex propositions can be formed from many simple propositions being connected together by *sentential connectives*. The five commonly used connectives are shown here below figure 4.4:-

Connective	Symbol
AND	$\wedge$
OR	$\vee$
NOT	$\neg$
IMPLIES	$\rightarrow$
EQUIVALENT	$\equiv$

Figure 4.4 Connectives used in formal logic.

The use of these connectives brings us to *propositional calculus,* where statements like ***"The book has been borrowed or it has been sold"*** can be expressed.

The use of propositional calculus is too simple where it only deals with true or false propositions. Knowledge in the real world is more complex than just truth values when we need to speak about objects, postulate relationships between them, etc. *Predicate calculus* is an extension of propositional calculus that can be used in order to achieve these objectives (Barr and Feigenbaum, 1981).

4.7.3.2 Semantic Networks

Semantic networks are the most general and, perhaps, the oldest representational structure for expert system knowledge bases. Semantic networks consist of a collection of nodes that are linked to form object relationships. Arcs linking nodes carry notations that indicate the type of relationship (Badiru, 1992).

Semantic networks were developed by Quillian and Shapiro (Barr and Feigenbaum, 1982). It is a representation that makes use of nets which contain nodes and links. A node is used to represent an object, a concept or an event and a link represents a relationship between two nodes.

Hierarchical Structure for Interior Vehicle Design.

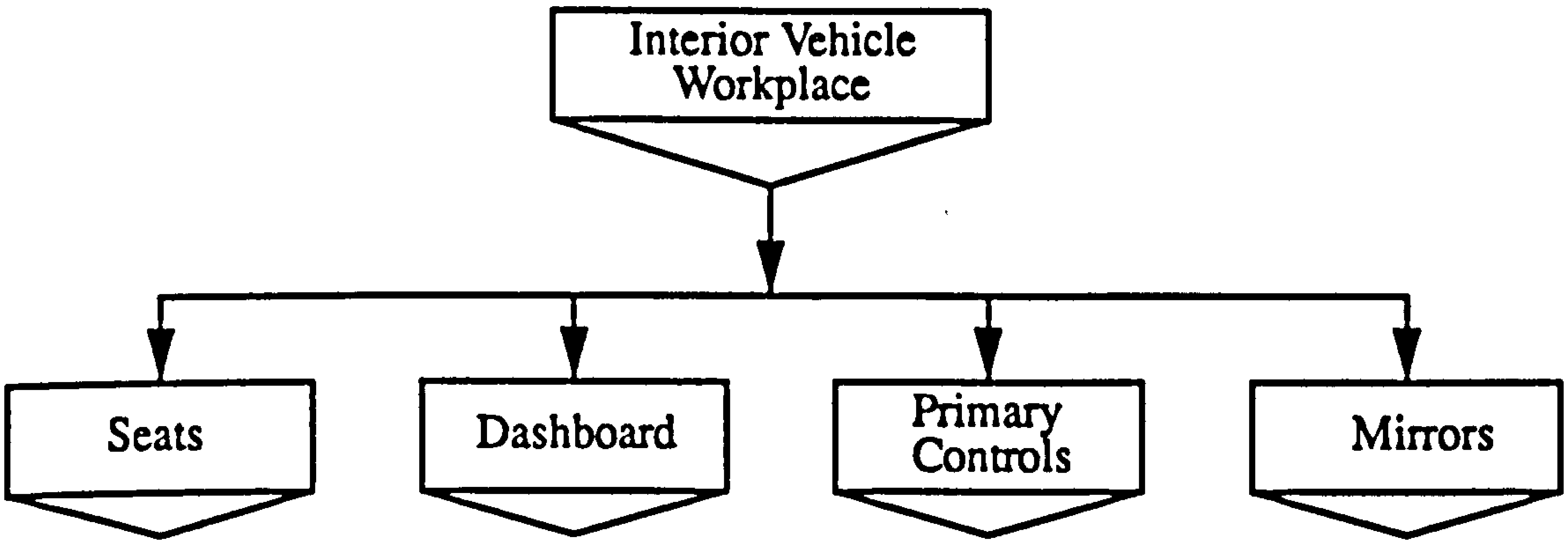


Figure 4.5 Semantic net for interior vehicle design.

Hierarchical Structure of a Pedal Controls.

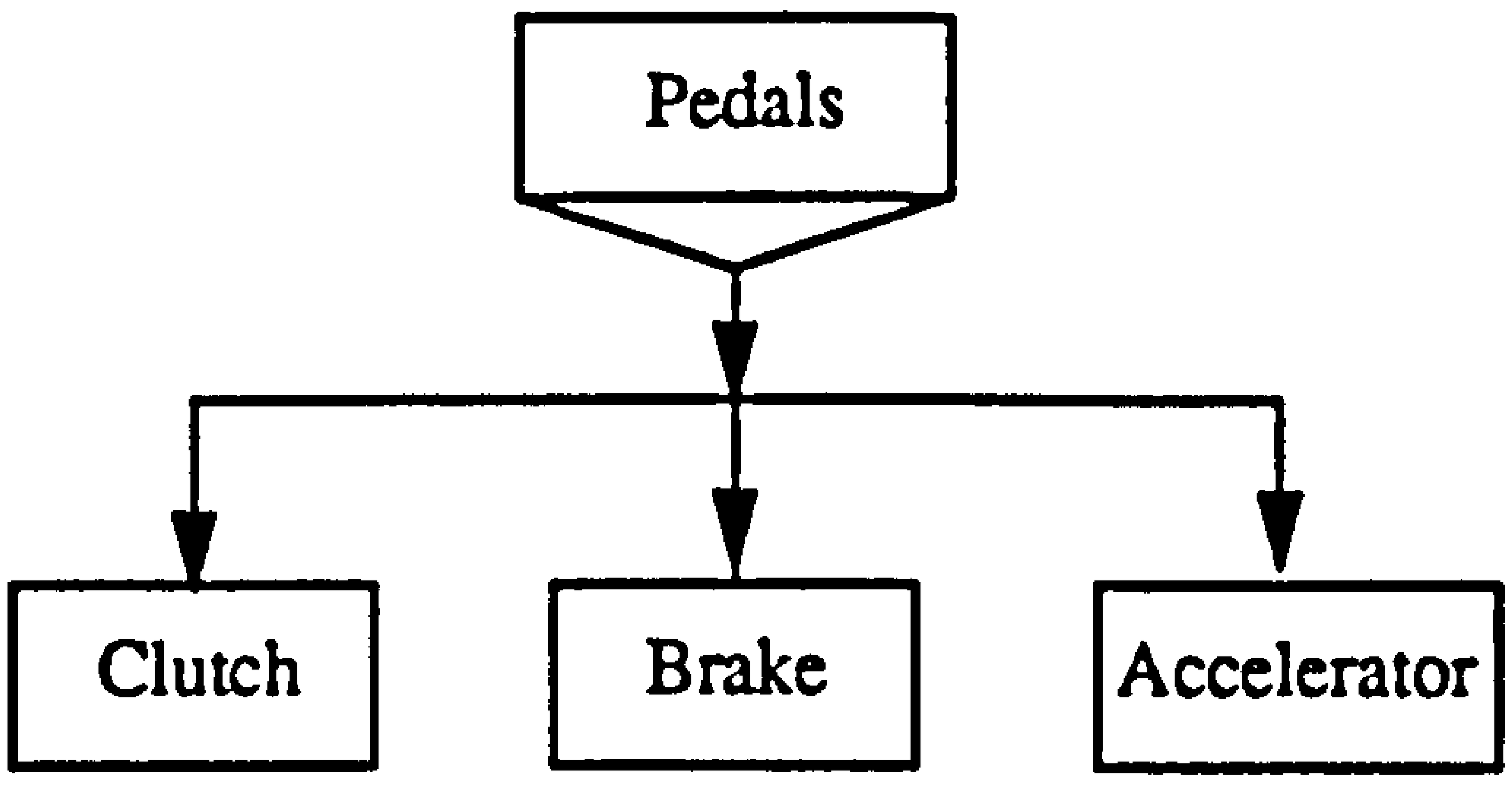


Figure 4.6 Semantic net of pedal controls

A simple example from the research would be that a vehicle has nodes representing seats, dashboard, primary controls and mirrors (figure 4.5), pedals and controls (figure 4.6). The links between the nodes represent relationships such as the ownership relationship that defines the clutch, brake and accelerator as members of the pedals group.

4.7.3.3 Production Rules and Systems

Production rules system architecture is the knowledge representation method that has become popular and versatile in expert systems. Newell and Simon (Barr and Feigenbaum, 1982) advocated this representation scheme as a model of human reasoning. Representing knowledge in this form is a natural way to extract and encode rule-based knowledge in many applications (Barr and Feigenbaum, 1982, and Rao and Jain, 1988).

The underlying idea of a production rules system is the notion of the *condition–action* pairs called ***production rules*** or ***productions*** (Levine, 1986) or ***rules*** (Badiru, 1992). ***Production rules*** provide a formal way of representing recommendations, directives or strategies (Badiru, 1992, and Bonnet, 1985). A system based on production rules consists of the three parts:–

i) A rule base containing a set of production rules.

ii) One or more data structures containing the knowledge relevant to the domain of interest and also some possibly useful definitions. These are often called the facts bases or databases. A database (also called a context) contains facts that describe the status of the system.

iii) An inference engine (also called the interpreter) which makes use of the production rules and the database of facts in order to make inferences, diagnose, suggest actions, etc. A production rule is a statement cast in the form shown below:–

> **"IF this condition(s) holds**
> **THEN do this action(s)"**

An example of a production rule used for design and styling purposes in the vehicle interior design:–

```
if
      displays:speedo>shape = circle and
      displays:tacho>shape = circle
then
      displays:panel>styling = good design | balance.
endif.
```

A styling rule, that says if the shapes of both the "speedometer" and the "tachometer" that belong to the "displays" is "circle" then this panel's styling is "excellent" a good design and "excellent" of balance.

The IF part of the production rule is called the **condition** and describes the required conditions for the particular production rule to be applicable (in this case that both the speedometer and tachometer are circular). The **THEN** part contains the **actions** which describe the actions to be taken if a particular production rule is fired (in this case a value is assigned to 'styling').

Production systems or production rules can represent knowledge in several independent modules, allowing greater modularity in the knowledge base than any other scheme allows. In production systems or production rules, one can add or delete rules without changing other rules. Since the knowledge base is changed very frequently by adding and deleting rules in most applications, this feature enables system maintenance with minimal effort. Furthermore, imprecise rules can be used with some measure of belief or reliability associated with each rule. The inference engine then uses this measure to reach appropriate conclusions.

4.7.3.4 Frames

A frame consists of a collection of slots that contain attributes to describe an object, a class of object, a situation, an action, or an event (Badiru, 1992). A ***frame***, sometimes called a '***schema***', is a generic data structure representation, a stereotyped situation or class of objects having different categories of information which are called '***slots***'.

A frame can be considered as a slot-and-filler structure formed by the following network of nodes and relations (Minsky, 1972, Pang and MacFarlane, 1987 and Dym 1994):-

1) The frame's top level is fixed and represents things that are always true.

2) Lower levels have many slots or terminals that must be filled by specific instances or data (describing aspects of objects). Associated with each slot may be a set of conditions that must be met by any filler for it.

Frames can be linked together to form a hierarchical classification of the domain knowledge, called *frame-systems,* which allows for *inheritance* relationships between them. Each frame has a name which corresponds to one entity. The structure of a frame is shown in figure 4.7.

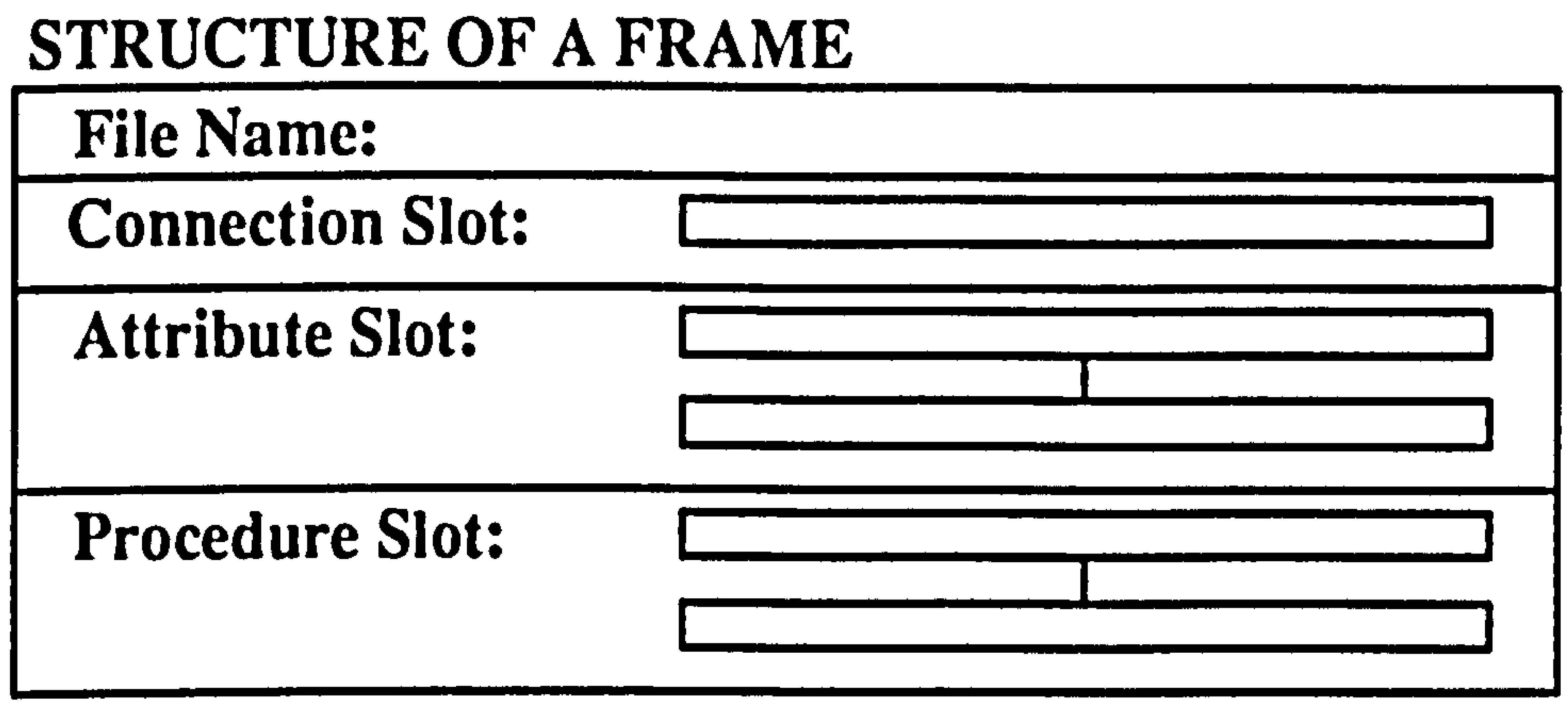


Figure 4.7 Structure of a frame (Source:– Minsky, 1972 and Dym, 1994).

The allowable contents of a slot are shown in figure 4.8. A slot is a component of knowledge which can have a number of entries. Each one can either be the name of another frame (in the case of inheritance) or a primitive, which serves as the basic piece of information for the current domain knowledge. A primitive can be a rule, a datum, a conclusion, etc. Slots can be of three basic types as described below:–

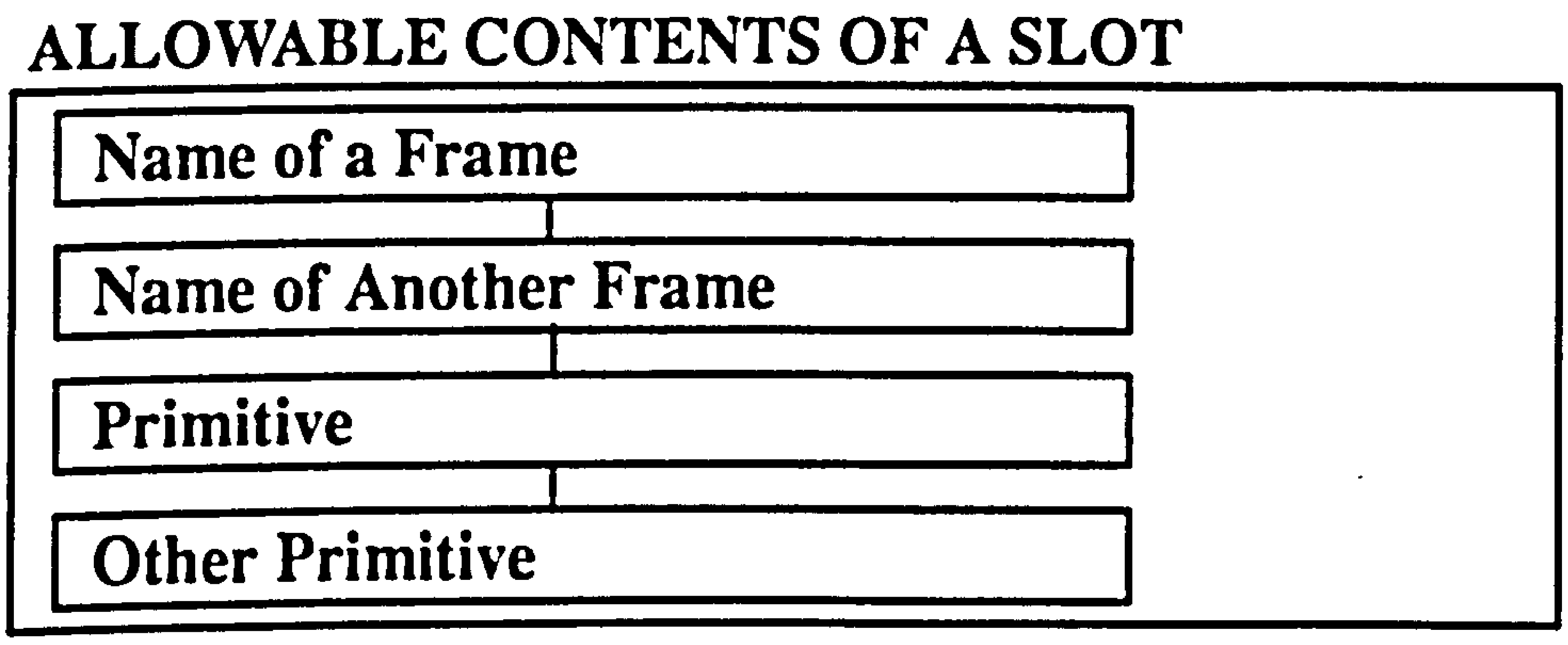


Figure 4.8 Allowable contents of a slot (Source:– Minsky, 1972 and Dym, 1994).

i) Connection slot – this defines the position of the current frame within the knowledge base.

ii) Attributes slot – this describes the information represented in the frame. The number and the types of these slots depend on the domain knowledge.

iii) Procedure slot – this defines how the information required by the frame should be obtained and what are the steps to be taken if that particular frame is invoked (Minsky, 1972, Pang and MacFarlane, 1987). An example of a frame for the ergonomics design knowledge base expert system which is based on KES shell tools is described below:–

1)	**Frame Name:–**		Displays
2)	**Connection slot:–**		Speedometer
3)	**Attribute slot:–**		Styling
		type:–	mlt
		Possible values:–	good ergonomics
			good design
			good styling
			aesthetics
		Constraint:–	None
4)	**Procedure slot:–**	Knowledge sources:–	rule 1
			rule 2
			rule 3
		Demons:–	none
		Attachments:–	none

4.8 Development and Application of Expert Systems

Expert systems have been in existence for over 15 years, but their early applications were basically limited to research and development. These early systems required large resources to model decision making and thus could be afforded by only a few large companies and government departments (Barr and Feigenbaum, 1982, Buchanan and Duda, 1983). Expert systems are becoming more widely used both in the financial and manufacturing industries whereas earlier applications were limited to consultative or diagnostic activities involving no immediate external feedback into the expert systems to affect their inferencing (Beynon–Davies, 1991).

Today expert systems are of interest in a wide range of areas, as indicated in figure 4.9 Badiru, (1992) has stated that currently more than 50 companies are involved in development of expert systems or so called knowledge based systems.

Other application areas are found to be as follows (Harmon et al, 1988, Beynon–Davies, 1991, and Badiru, 1992).

1) Science and Engineering – expert systems are used to organise and manipulate large bodies of information and analysis processes used in mass spectrometry analysis, biological classifications, metallurgy, mathematics (MACSYMA) (Beynon–Davies, 1991), engineering structure calculation (SACOM) (Badiru, 1992).

2) Design and fault diagnosis – XCON/R1 (a configurer for VAX computers) (Harmon et al, 1988), faults diagnosis system (FALOSY), fault diagnosis for computer hardware and software (CRIB) (Beynon–Davies, 1991, and Badiru, 1992).

Category	Problem Addressed
Interpretation	Inferring situation descriptions from sensor data.
Prediction	Inferring likely consequences of given situations.
Diagnosis	Inferring system malfunctions from observations.
Design	Configuring objects under constraints
Planning	Designing actions.
Monitoring	Comparing observations to plan vulnerabilities
Debugging	Prescribing remedies for malfunctions.
Repair	Executing a plan to administer a prescribed remedy.
Instruction	Diagnosing, Debugging, and monitoring student behaviour
Control	Interpreting, predicting, and monitoring system behaviour.

Figure 4.9 Generic categories of knowledge engineering applications. (Source: Hayes–Roth, 1983).

The application of expert systems in industry, particularly in design and manufacturing, is an emerging technology for industrial problems. Examples of these industrial applications include (Schoen and Sykes, 1987):–

- CAD/CAM
- Production planning and scheduling

- Group technology
- Inspection
- Quality control and analysis
- Tuning of closed–loop control systems
- Flexible machining systems, etc.

More recently, the potential for expert system application to Computer–aided Design Systems and Man Modelling CAD systems or Computer Aided Ergonomics Design Systems (CAEDS) has been proposed and is discussed below.

4.9 Application Expert Systems in CAD and CAEDS Systems

The possibility exists of interfacing to the design process by the application of Artificial Intelligence and Expert System technology in CAD. Although the creativity of the human can not be substituted for, at least in the near future, AI can assist human designers and engineers in improving design and can ease the modelling task. One can expect to see more and more involvement of CAD systems in the design process. The potential application of expert systems in computer aided design (CAD) technology is likely to influence (Kloke, 1990):–

1) The graphics of man modelling systems;
2) Ergonomics design assessment and model evaluation methods;
3) Implementation using external program/language;
4) Reduction in total time of the design process.

There are several factors which must be considered in the determination of whether a particular CAD problem area can be solved using the expert systems technique (Costed, 1987). Figure 4.10 shows a computer–integrated engineering system with artificial intelligence. Expert systems could be involved in three phases:

4.9.1 Problem Definition

The problem definition phase; the problem associated with the design process and CAD system is presented.

4.9.1.1 Geometric Reasoning

Geometric reasoning can be understood at three levels, firstly the knowledge base techniques for vehicle interior design requires a geometric reasoner; secondly as a means for the abstraction of knowledge sources and construction of the knowledge base required for the evaluations; thirdly for the implementation of the evaluation.

Bonney defined geometric reasoning in a straightforward manner as "*the application of computer techniques to spatial problems so that deductions can be made from the geometry*" (Bonney et al, 1989). The essential aspect of this definition is that geometric data structures contain 'hidden' information that can only be used in applications by the use of deductive processes. Determination of the volume of a geometric component provides a simple example as most geometric modellers would not maintain this information as an explicit item within the data structure. Instead, the volume would be determined as required by the application of numeric algorithms.

However, Bonney goes on to say that "our aim is to apply geometric reasoning to geometrically complex domains in a multivariate decision environment". Hence the phrase "geometric reasoning" is usually reserved for situations where the deductive processes need to be "intelligent" as they require some knowledge of the environment within which the geometry is defined and the objectives of its application. Thus geometric reasoning is typically used in complex situations where there are a number of alternatives which have no direct algorithmic solution based on geometry alone. Contributions to Woodwark's "Geometric Reasoning" (1989) illustrate the variety of possible applications including robot motion planning (Davenport, 1989), where a robot could adopt a variety of strategies to avoid obstacles in the workplace, feature recognition (Jared, 1989), where geometric entities constituting the model of a component are grouped into higher entities useful for a manufacturing application and the operational planning of manufacturing cells (Stobart and Williams, 1989).

One method of supplying the intelligence necessary for geometric reasoning is through the use expert systems and this is the approach that has been in this research. Other alternative methods from the field of artificial intelligence have been briefly reviewed in this chapter.

The use of expert system for this research, and the application area of ergonomics design, has its roots in Bonney's paper and the subsequent work Azarkadeh (1988). Bonney's ALFIE system demonstrated the use of expert systems for heat stress and strength analysis (Bonney et al, 1989) where the knowledge base required geometric information from the SAMMIE system in order to reach conclusions on the suitability of proposed designs. In this way the computer aided design system and the knowledge-based system were in a one-way communication. i.e. the expert system made judgements on the basis of the supplied geometric information, but did not provide recommendations as to how geometry could be changed to improve the design. Azarkadeh adopted a very similar approach, but based on the legislative requirements of the layout of instruments on car dashboards.

This research has been based on this earlier work, but has extended the method beyond evaluation and towards the provision of design change recommendations. A two-way communication is established between the computer aided design and expert system such that recommended changes can be implemented in the geometric database. In this way geometric reasoning is used to not only understand the consequences of a particular geometric configuration, but also to modify geometry to more closely match desired design criteria.

4.9.1.2 Presentation

Presentation is a recommendation phase where recommendations or established legislation and design working practice, can be graphically presented to the user.

Design has not been given the attention that other areas have. Although there has been a great deal of interest in design and manufacturing problems, there has been little work done from the standpoint of using rule-based techniques to address this problem.

To build an expert system for computer–aided design systems, the designers' knowledge of design rules, drawing and problem solving methods are encoded into a knowledge base (figure 4.10).

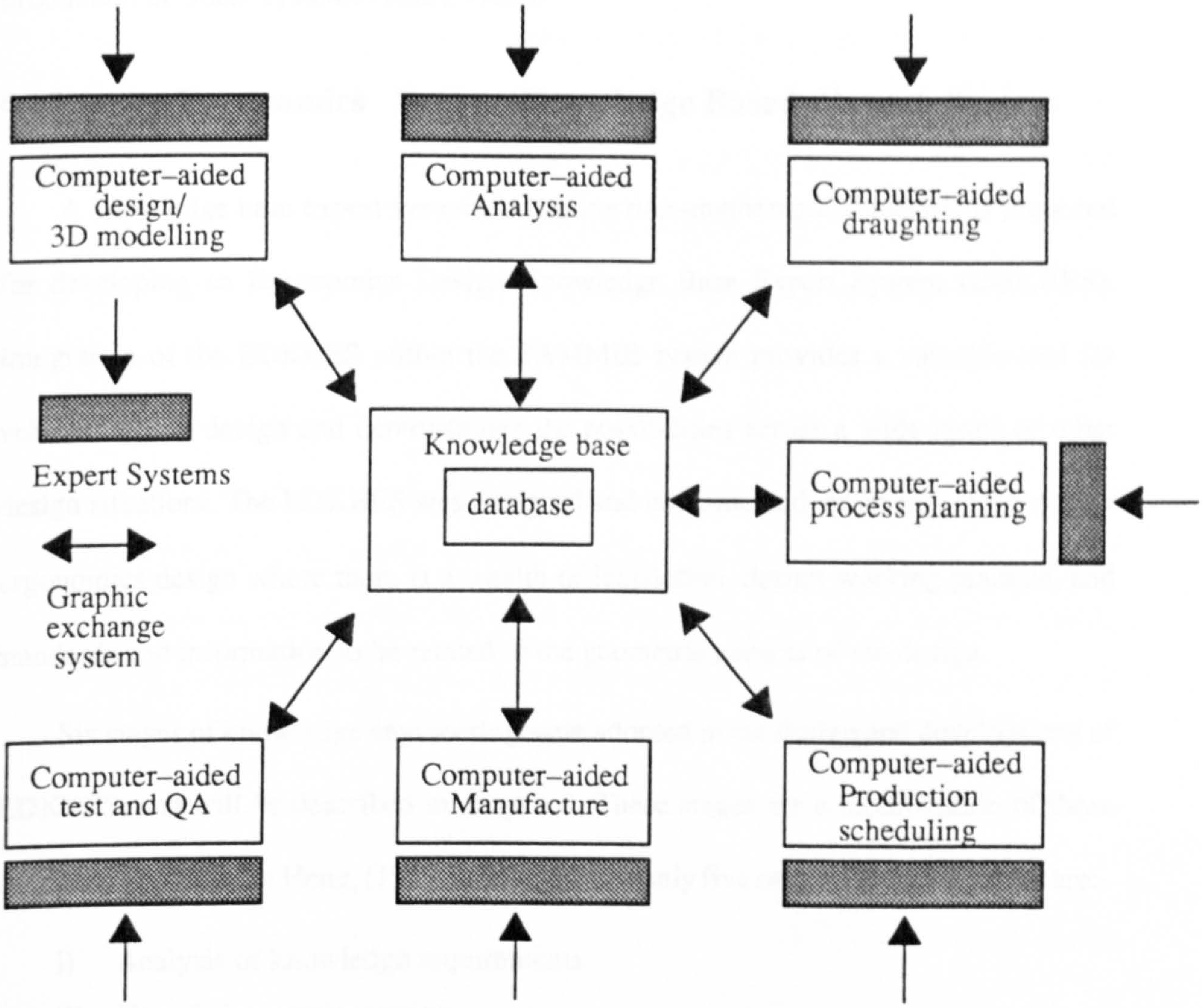


Figure 4.10 A computer aided engineering system integrated with artificial intelligence (Source: Austen, 1985)

Judgmental knowledge consists of rules–of–thumb and is involved in all phases of the design process, from conceptualization, methodology, implementation, and test and integration. It is the most difficult type of design knowledge to capture in an expert system.

Finally, control knowledge directs the interface into the database where "geometric reasoning" is applied. It decides which facts, data and rules to apply. It controls and coordinates other knowledge in expert systems.

The development of expert or intelligent knowledge-based systems is raising some new and very important human factors issues in the field of man-computer interaction. With the advent of thinking and reasoning systems, the cognitive characteristics of the human mind must become a critical design constraint in the production of such systems (Kidd, 1983).

4.10 An Ergonomics Design Knowledge Based Expert System

A knowledge base expert system containing non-mathematical models is proposed for developing an Ergonomics Design Knowledge Base Expert System (EDKBES). Integration of the EDKBES within the SAMMIE system provides a valuable tool for vehicle interior design and demonstrates the possibilities across a wide range of other design situations. The EDKBES was designed and implemented as an expert system for ergonomics design where there is a wealth of legislation, design working practice, and standards and information to be related to the geometric aspects of the design.

Six stages of knowledge engineering were adopted in the design and development of EDKBES, and will be described in chapter 5. These stages are a modification of those originally proposed by Heng, (1987) who suggested only five stages. These six stages are:-

i) Analysis of knowledge requirements
ii) Knowledge representation
iii) Structuring the knowledge
iv) Formulation of the knowledge
v) Implementation of the knowledge
vi) Integration within an application system

Stages (i) to (iii) have been discussed in this chapter, stage (iv) is discussed in chapter 5, and stages (v) and (vi) are discussed in chapters 7 and 8.

CHAPTER 5

METHODOLOGY

5.1 Introduction

This chapter introduces the general methods used in building an integrated CAD and knowledge-based system for vehicle interior design. The selection and use of the SAMMIE System for vehicle interior design is discussed as is the selection of the KES Expert System for building the Ergonomics Design Knowledge Based Expert System (EDKBES). The four related areas discussed are the determination of the knowledge base, the selection of domain knowledge, examples of the domain knowledge and the selection of a method of knowledge representation.

The important features and applications of computer aided design (CAD) and computer aided ergonomics design systems (CAEDS) are described. The approaches and tools that are available to develop CAD and SAMMIE CAEDS are also discussed. Finally, the use of the SAMMIE computer man modelling as a tool in the development of vehicle interior design is discussed.

5.2 General Methods and Preparation

There are a number of preparatory stages required before the complete system can be built.

The initial stage is the design and construction of a three-dimensional prototype model of the interior of a car using the SAMMIE computer man modelling system. It was necessary to build the model paying particular concern to its logical structure so that it was suitable for communication with the expert system.

The second stage was concerned with the development of an expert system using the method promoted by Hayes-Roth et al, (1983) and Heng, (1987) and six steps were used as shown in figure 5.1. The development involved the creation of an ergonomics design knowledge based expert system (EDKBES) using the KES expert system shell, and the development of external programs using communication files and command files and these will be discussed in chapter seven.

The third stage of the research work was the incorporation of the EDKBES within the SAMMIE system by using external program communication files for evaluation using geometric reasoning.

5.3 Integration of an Ergonomics Design Knowledge Base in the SAMMIE System

There are three reasons for integration of an ergonomics design knowledge based expert system in the SAMMIE system:–

1) As a "design assistance tool" to allow a designer or engineer to monitor and modify his design repeatedly and observe the results of different designs and specifications.

2) As a "decision assistance tool" to provide a high level design environment helpful to both the expert and user. Expert systems have been found to be useful in judgmental and problem solving areas, where constraints may not be uniquely resolved by theory or algorithm.

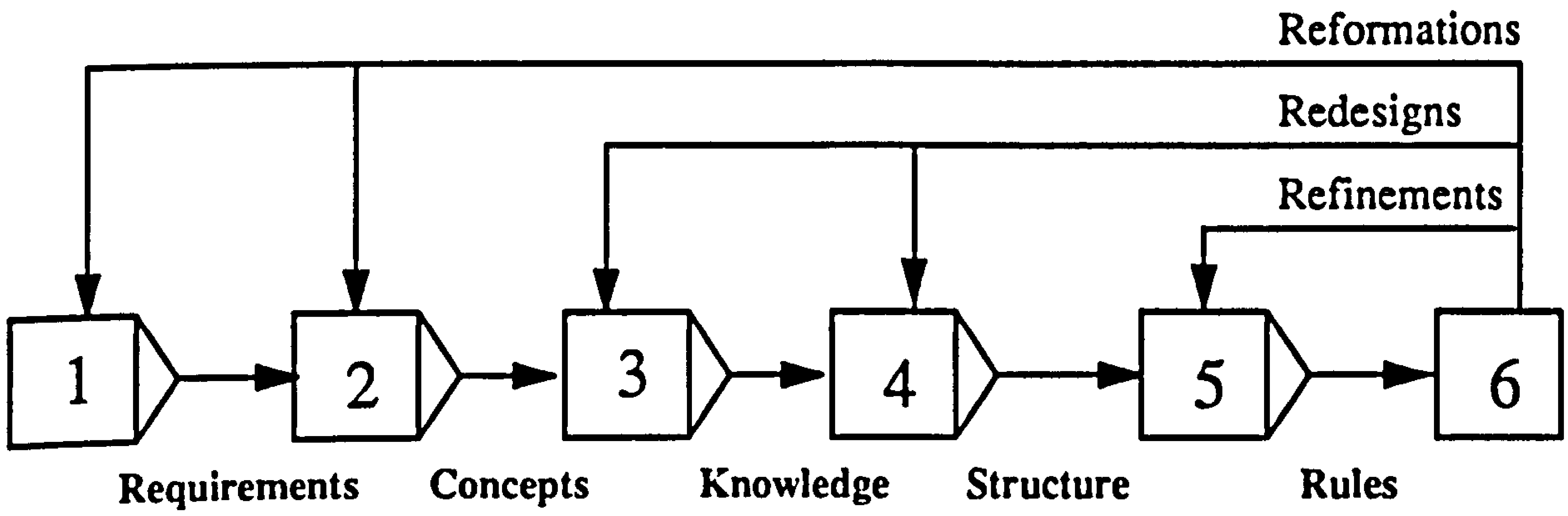
An Ergonomics Design Knowledge Base Expert System (EDKBES) can help the designer and engineer to solve the problem parts of the design process.

3) As a "Knowledge Based tool" incorporated in the the SAMMIE system. The domain knowledge is seen as the most important component of an expert system and the SAMMIE system will require an extensive application dependent knowledge base.

Two methods for developing intelligent systems have been clearly identified (Gero, 1987 and 1990). The first method was through dedicated expert systems for particular design applications to give power to the problem solving activity, while the second emphasises attempts to understand the designer's or engineer's capabilities in order to assist their expertise or intellectual capability. Elements of both approaches have been adopted in this research with an emphasis on the first.

Expert System development faces similar problems to those that occur in conventional software systems. In addition, one has to cope with the question of extracting knowledge from domain experts and structuring this knowledge in a form suitable for design working practice.

The process of developing a functional EDKBES is based on a Systematic Design approach of development in six stages. Figure 5.1 shows the methodology and is a modification of the model developed by Hayes–Roth et al, (1983), and Heng, (1987). The methodology stages are briefly described in subsequent sections.



1. Identification – identify problem characteristics, (results in Requirements)
2. Collection – collection, extraction of knowledge, (results in Concepts)
3. Knowledge – knowledge acquisition and elicitation, (results in Knowledge)
4. Methodology – design structure, organise knowledge, (results in Structure)
5. Implementation – constructing knowledge base, rules (results in Rules)
6. Test and Validation – test and validate the systems.

Figure 5.1 A methodology for the development of an Expert System

During the building of an expert system (Heng, 1987) has reported that there are four crucial problems:–

i) defining system requirements,

ii) extracting expert knowledge,

iii) organising and structuring expert knowledge for machine manipulation,

iv) maintaining the interest and enthusiasm of the domain expert.

The fourth stage is the representation of knowledge, in which knowledge has to be acquired and represented, i.e., formalised, implemented, tested and validated as shown in figure 5.1. Here there are a few possible approaches that are normally considered:–

i) The author acts as both knowledge base engineer and domain expert, using a tool to encode knowledge he already possesses.

ii) The author acts as the knowledge engineer and must elicit the domain knowledge from another source, such as another person, written material, or a data base.

iii) The author is the domain expert and acts as the knowledge engineer, but requires the services of a software engineer to build an interface between the expert system and other applications.

iv) The author is the domain expert and someone else acts as the knowledge engineer because the author lacks the time and resources.

For the design procedure expert system developed, the author adopted approaches (i),(ii) and (iii) since at the commencement of the research he was neither an ergonomics expert nor a software engineer. The knowledge acquisition carried out by the author is described in chapter two and the building of the knowledge base in this chapter.

5.3.1 Requirements Analysis

The first step in the design of the ergonomics design knowledge based expert system was to analyse the requirements of the system. A complete understanding of the requirements was necessary to guide and support the design activities. To establish an understanding of requirements it was necessary to:–

1) Identify the external requirements of the system (e.g. information requirements of the EDKBES from the relevant standards, legislation and design working practice).

2) Determine the available domain resources (e.g. experts, books, magazines, standards, reports, journals, publications, etc.).

3) Characterise the end users (their familiarity with expert systems, and computer man modelling systems). Normally the Ergonomics Design Knowledge Based Expert System (EDKBES) runs invisibly in conjunction with SAMMIE but it may be desirable for the user to obtain explanation and justification of the decisions made by the system. It is assumed that a designer/engineer with some knowledge of the KES expert system shell will operate the **EDKBES**, although 'help' facilities are available.

5.3.2 Knowledge Representation Concepts

Knowledge representation is accomplished by using the methods described in chapter 4 and implemented in the KES expert system shell, and the geometric knowledge is handled using the data structuring techniques available within the SAMMIE system.

5.3.2.1 Knowledge Acquisition and Formulation

Knowledge representation is the most crucial task in the development of the EDKBES, because it determines its inferential capabilities. During this task, knowledge extraction has been undertaken via books, research publications, journals, magazines, reports, and standards etc. This chapter discusses how such knowledge was extracted in order to find a general solution to the selected aspects of the design problem.

Domain knowledge consists of facts and data about the domain and relationships between these facts and rules. For **EDKBES**, these facts were first structured in a modular form and then built as an attribute hierarchy so that the relationships between them could be realised. This eased the formulation of the knowledge (facts) in the form of rules and the inter-relationships between them. The KES expert system shell was used for formulation and implementation of the heuristic rules. Later in this chapter the formulation of knowledge elicited and implemented in the KES shell is discussed.

5.3.3 Implemented Knowledge

The formulated rules are implemented using a text editor (SUN window) and EDKBES (using the kesp production rule inference engine) to represent the expert system.

The software was produced as a set of modules, and these were tested to verify that each module meets its specification as described in chapter seven.

5.3.4 Testing and Validation of Knowledge

Tests were carried out to show the validity of incorporating EDKBES in SAMMIE for vehicle interior design. The validation involved selecting and running different test cases to show the ability of the system in handling typical inference and end user scenarios. In this way the 'Window' and 'linear relationship' techniques were evaluated. The run–time efficiency of incorporating EDKBES in SAMMIE during the evaluation task, together with refinements, redesign, reformulation and modification have been evaluated as described in chapter eight.

5.4 The SAMMIE System for Developing a Model of Vehicle Interiors

The technical reasons for selection of the SAMMIE System have been discussed in chapter three, and the opportunities for enhancing the systems capabilities are briefly described here.

The SAMMIE Computer Man Modelling System could benefit from the application of Expert Systems technology – to provide design support, and possibly, to assist in the creative and intellectual processes of design.

SAMMIE man modelling gives support to certain parts of the design process, such as early conceptual design activity, by providing facilities for simulation models in three dimensions. However, this support is not intelligent.

Knowledge engineering is concerned with the building of knowledge based programs that can represent and reason with knowledge. The inference engine is a fundamental procedure used for the process of reasoning and concept manipulation where no prior mathematical theory exists. Design activity is a non–numeric and non–algorithmic area of causal and experimental knowledge. Knowledge engineering has the potential to make significant contributions to SAMMIE man modelling through the manipulation of knowledge or expertise in a particular domain and interactive operation with the user.

Design is considered to be a highly intellectual, creative and imaginative activity involving knowledge, experience and the intuition of the designer combined with the design constraints. Usually the designer presents design ideas using a set of conceptual design variables which have been selected as the best alternatives. Finding the optimum selection from the these alternatives is often based on intuition and experience.

The application of expert systems allows the design to be monitored as it progresses. With this intelligent support, the design knowledge base and the goals and the constraints of the domain, the expert system can evaluate the design against those constraints and give explanations for the rejection of certain designs. Recommendations for improvement can be made through a dialogue with the user and the provision of information as required for decision making.

5.4.1 Description of the SAMMIE System

SAMMIE is available on engineering workstations (e.g. SUN SPARC, and Silicon Graphics), and is intended to assist designers and engineers in design and evaluation of human workstation/workplaces, using the following features:–

- A three–dimensional solid modelling system that enables the construction of full size models.
- Three–dimensional models of the human form that can be altered to reflect the variety of shapes and sizes of the population.
- Special facilities to evaluate human capabilities, such as reach, vision and fit.
- A highly interactive man–machine interface, with a user–friendly dialogue.

5.4.2 SAMMIE Equipment and Workplace Modelling

The workplace modelling system is used to generate full size three–dimensional geometric representations of a working environment and specific items of equipment. The modelling scheme is based on the simple boundary representation form of solid modelling

(Requicha, 1982) which enables the system to be highly interactive whilst maintaining a sufficiently accurate three-dimensional model. This method requires that solid shapes are constructed from a description of the location of their vertices, a knowledge of which vertices are joined together to form edges and which edges form plane polygon faces. It allows solid evaluations such as hidden line removal and interference checking without producing unduly complex models (Porter et al, 1993).

Complex models may be created from a range of (simple) primitives – cuboids, prisms, cylinders etc and non regular solids from a description of their vertices, edges and faces as shown in figure 5.2.

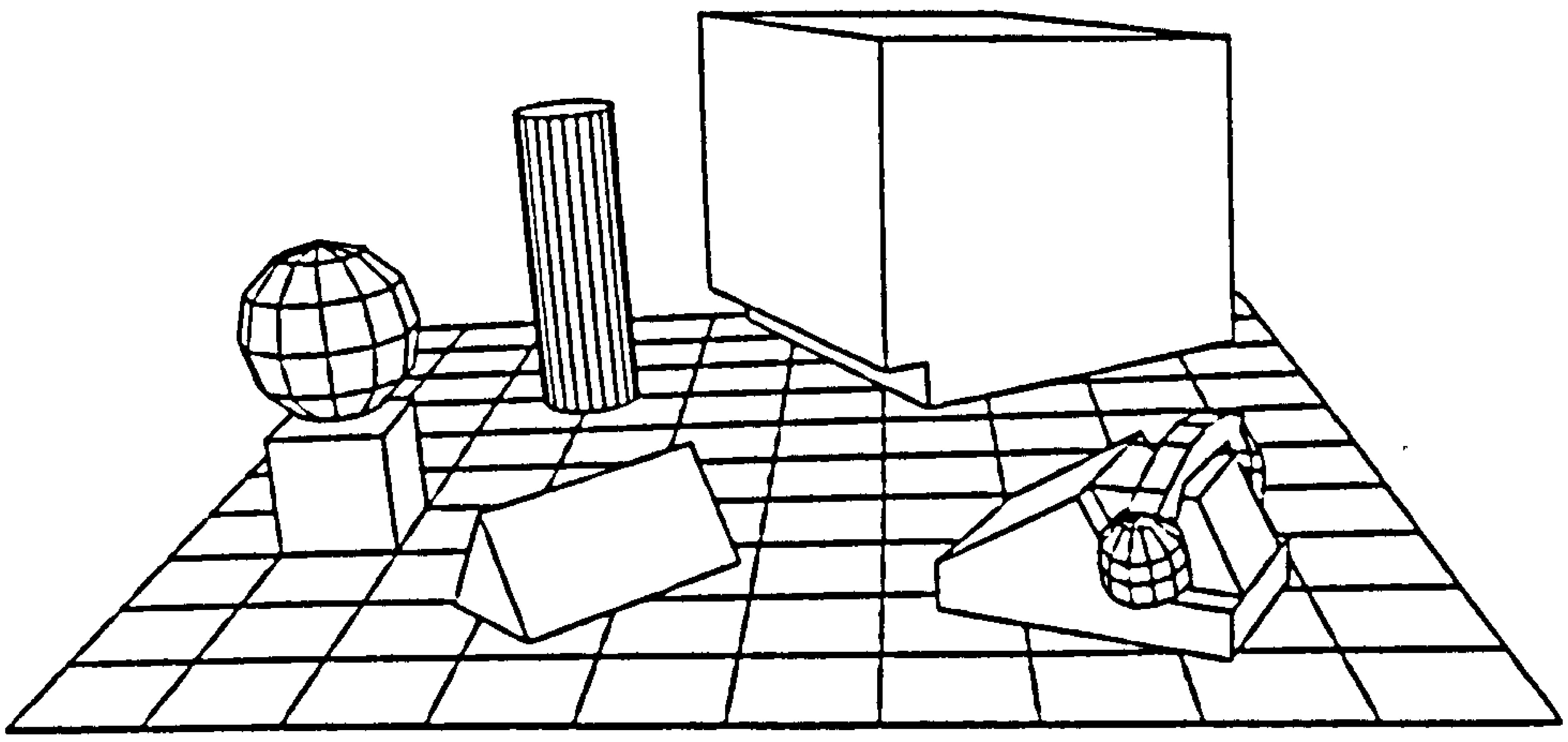


Figure 5.2 Example of simple model types available in the SAMMIE system.
(Source: Case et al, 1990)

The SAMMIE user controls the actions of the man model within the workplace model as an aid to evaluation. Figure 5.3 shows the SAMMIE menu structure, which is described below.

5.4.3. SAMMIE Ergonomics Facilities

The SAMMIE system has several ergonomics facilities to help the user assess and evaluate the ergonomics of a particular design in workplace.

5.4.3.1 Reach Assessment

A major evaluation tool is provided by various assessments of the reach capabilities of the man model. Reach can be assessed simply by positioning the arms or legs so that the hands or feet either contact, or fail to contact, a specified control or point in space (Case et al, 1990).

5.4.3.2 Vision evaluation

Views 'seen' by the man model are fully under the control of the user. For example, the user may select the left, right or a mean eye position, 60 degrees or 120 degrees cones of vision and specify the angle of vision using the eye position and/or head as appropriate. Two-dimensional visibility plots allow vision to be evaluated at any given surface and can be used for example to check vision of a vehicle fascia through the steering wheel. Three-dimensional visibility charts describe all-round visibility and can be used for example to check external visibility from a vehicle through all the windows (Case et al, 1990).

5.4.3.3 Mirrors and Reflections

The mirrors modelling facility can be used to design mirrors for vehicles and to determine whether mirror reflections will be a problem in windscreens or computer screens. The mirror parameters of focal length, convexity/concavity, size and orientation are all variable and can be interactively adjusted to provide the required field of view displayed on the mirror surface as seen by the man model (Case et al, 1990).

5.4.4 User Dialogue/Interacting With SAMMIE

The SAMMIE system is highly interactive and allows designers to proceed through the design process in a manner determined by their own requirements rather than in a predetermined manner. The user communicates with the system through a menu based dialogue using either keyboard or mouse. Its has nearly 30 menus, containing between 10 and 20 separate commands grouped according to their functions (Porter et al, 1993).

The menu structure is shallow being only a maximum of three layers deep. This menu structure works very efficiently as it allows the designer to integrate facilities quickly and it is an asset for both naive and skilled users as all the commands are displayed, acting as a job aid. The main menus are briefly described below:–

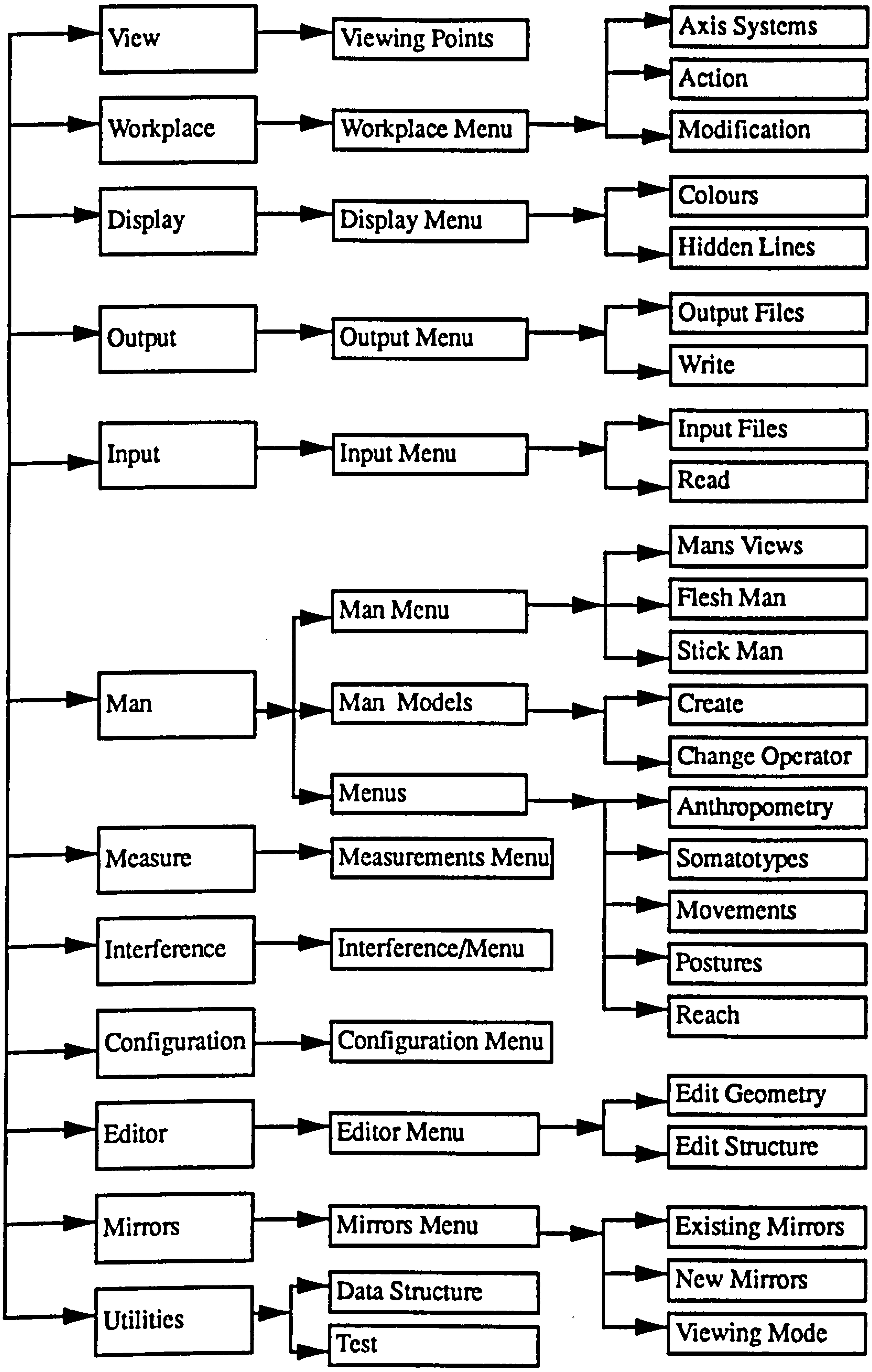


Figure 5.3 The SAMMIE Computer Man Modelling Menu Structure.

VIEW:- The View menu has commands for locating the viewing point, centre of interest, parallel projections, perspective views, changing the scale and acceptance angle.

DISPLAY:- The Display menu enables the various components of the models to be displayed selecting detail only when and where it is required. Colours can be specified for particular model components.

WORKPLACE:-The Workplace menu allows the positioning of the model components in three-dimensional space for moving the objects. Explicit instructions can be inputted or components can be 'dragged' on the screen using the mouse.

MASTER:- The Master menu is the main menu through which access is gained to the remainder of the other menus.

INPUT:- The Input menu enables the interactive construction of model components using a query and response format. Data input from a disk file can also be accepted.

OUTPUT:- The Output menu allows the output of various forms of file, model and plot data.

GEOMETRY EDITOR:- The Geometry editor menu enables the interactive modification of the geometry of model components so that, for example, an object can be made larger or smaller in any of its dimensions (SAMMIE, 1990, and Porter et al, 1993). This menu enables the X, Y or Z dimensions (width, depth and height) to be modified in isolation or concert. This feature is very useful at the concept stage because simple representations can be built and subsequently interactively edited to form more complex objects (Case et al, 1990).

STRUCTURE EDITOR:- The Structure Editor menu provides interactive control of the functional/logical relationships between model components. For example, this allows the man model to become a subset of a selected equipment model, such as the car seat, enabling both items to be moved in three-dimensional space as one unit when necessary (SAMMIE, 1990).

HIDDEN LINES:- The Hidden lines menu allows the creation of hidden line views, by deleting those lines behind solids, and the presentation of saved hidden line views

(SAMMIE, 1990). Models are usually displayed on the graphics screen in wire frame form. This type of display is easily interpreted by an experienced user although, for extra clarity or presentations, the 'hidden lines' can be automatically removed (Case et al, 1990).

MODIFICATIONS:– The Modification menu allows orientation of model components so that seats, doors, pedals etc., can only operate in a realistic manner. This menu also enables the creation of simple functional commands that for example allow a seat to be 'adjustable' forwards and backwards.

MIRRORS:– The Mirror menu allows convex concave or plane mirrors to be created from the chosen face of any solid. Reflected views are displayed on the face of the mirror as seen from the man model's viewpoint (SAMMIE, 1990, Porter et al, 1993).

MEASUREMENTS:–The Measurements menu is a geometric measuring facility which enables the minimum distance between any two solids to be calculated.

INTERFERENCE:– The Interference menu can be used to determine whether any chosen model components are intersecting. Any such solids are highlighted by flashing.

CONFIGURATION:– The Configuration menu allows the user to have control over the format of the screen display and output of messages.

DATA STRUCTURE:– The Data Structure menu allows the user to interrogate and manipulate the data structure of the various models which have been created. Its primary purpose is for debugging.

MAN:– The Man menu is the header menu for a suite of man–model related menus which are described below.

ANTHROPOMETRY:–The Anthropometry menu enables the various link lengths to be independently or collectively altered using either a specified percentile value or actual dimensions. This menu also allows the user to access the various stored databases for different nationalities, age groups, sex etc., in order to specify changes to the man model's anthropometry. These databases are under the user's control and they can be extended by providing the necessary means and standard deviations for the various link lengths.

SOMATOTYPE:- The Somatotype menu provides the user with control over the flesh shape with 76 options available.

JOINT ANGLES:- The Joint Angles menu allows the incremental or absolute setting of the various joint angles using flexion/extension, abduction/adduction and medial/lateral rotation as appropriate. A listing of the current posture is available. Joint constraint data informs the user whether a given joint is within 'normal' limits or within 'maximum' limits. Any request which would place the joint outside the pre-set constraints is not modelled and the user is informed accordingly. The choice of 'normal' and 'maximum' values is at their user's discretion depending upon the task characteristics.

POSTURES:- The Postures menu provides several 'basic' postures, such as standing, sitting, crouching, and crawling, which can then be modified to suit the specified task requirements using the JOINT ANGLES menu or the REACH menu.

MAN'S VIEW:- The Man's view menu provides perspective views from the model's left, right or mean eye-point.

REACH:- The Reach menu provides a variety of automated methods to determine the reach capabilities of the man model for both feet and hands.

CHANGE OPERATOR:- The Change operator menu allows the user to construct additional man models and to select the currently active model for evaluations of reach or vision.

Figure 5.4 shows the prototype model of a vehicle interior using the SAMMIE CAEDS system.

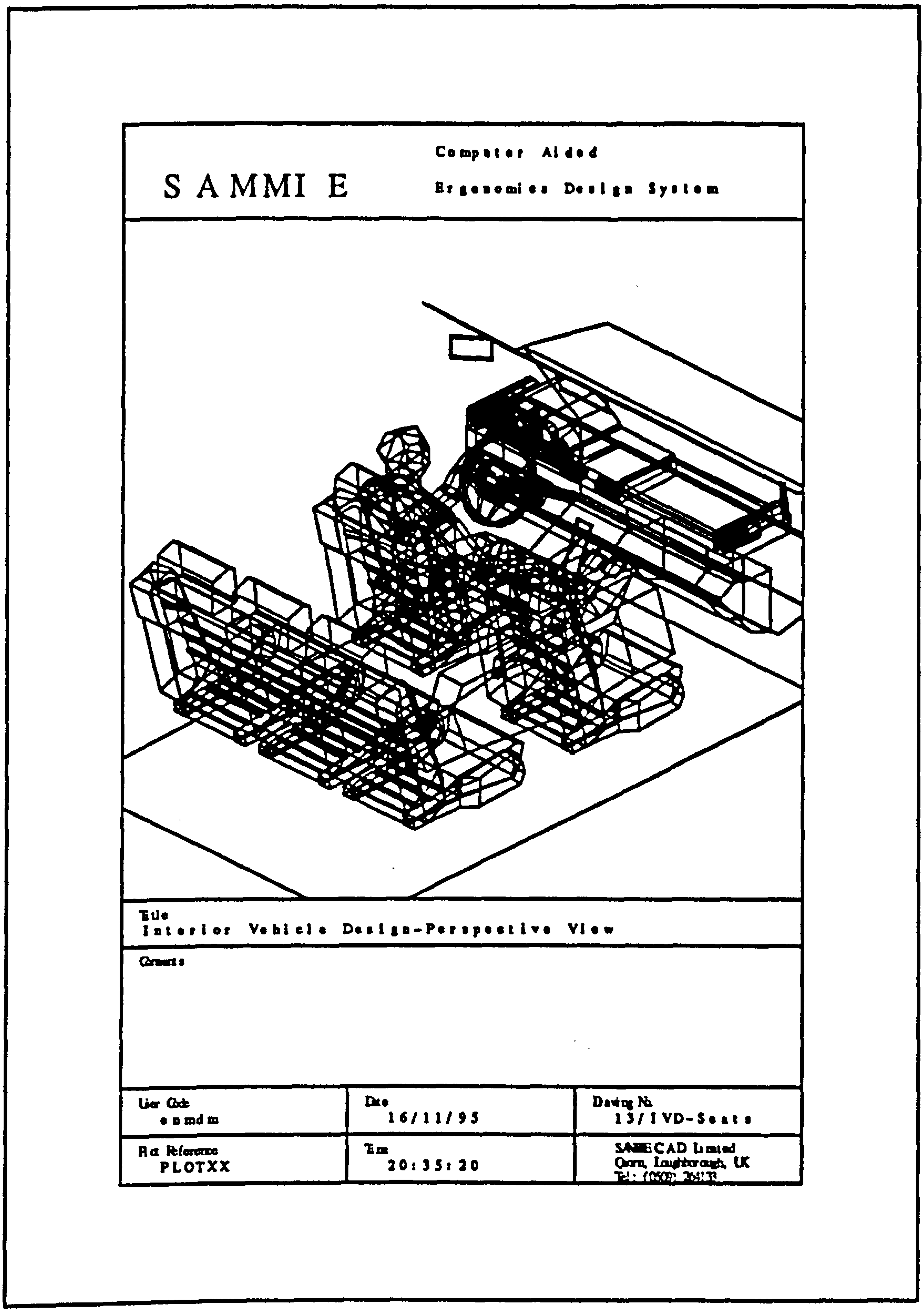


Figure 5.4 The prototype model of a vehicle using the SAMMIE system.

5.5 Selection of KES Expert Systems Shell

KES, Knowledge Engineering System, is a product of Software Architecture and Engineering, Inc., and was used to develop the ergonomics design knowledge base expert system (EDKBES) simply because it was suitable and available. KES is a domain independent expert systems development tool, written in the "C" programming language. The KES structure is made up of many independent, logical units or modules, designed to make building a system easier. KES is an expert system building tool that allows non-programmers to build powerful systems that can help solve problems.

1) KES is designed to aid in the tasks of building, maintaining, and running a knowledge base "parser", three knowledge base run-time systems (a character-based window interface, a bit-mapped window interface, and a scrolling interface) and a fully integrated developer's environment.

2) KES is an expert system development tool that supports the rapid development of prototype and production knowledge-based systems. KES supports an extensive set of knowledge base features including: backward and forward chaining, object oriented data representation, inheritance relationships, consistency (truth) maintenance, certainty factors and integration capabilities.

3) KES provides multiple inference engines to allow the expert system developer to use a reasoning method suited to the problem at hand, as no single approach is well-suited to all expert system applications. The methods provided are Production Rules (PS), Hypothesize and Test (HT) and Statistical Reasoning (Bayes) subsystems (KES, 1990).

5.5.1 Production Rules (PS)

Production rules are a modular knowledge structure representing a single chunk of knowledge in the form of "if then" or condition-conclusion statements. Deductive reasoning is used where conclusions follow from the premises, as shown in figure 5.5.

5.5.2 Hypothesize and Test (HT)

With hypothesize and test reasoning is provided through hypothesis formulation and subsequent verification. Abductive reasoning is used where the conclusion is a likely explanation of the premises, i.e. the domain knowledge is represented in the form of 'descriptions' as shown in figure 5.5.

5.5.3 Statistical Reasoning (Bayes Theorem)

This module performs statistical pattern classification based on Bayes Theorem. This theorem relates the probability of a hypothesis being true before receipt of extra information, to the probability of a hypothesis being true after that information has been received. The Bayes inference engine is well–suited for applications where there is a large body of pre–existing data expressed as probabilities (KES, 1990).

5.6 Inference Engine Selection

The main features of each inference engine are provided in figure 5.5 (KES, 1990). There are two major criteria in selecting the inference engine; the way knowledge is represented, and the way information is processed. Suggestions for the selection of an inference engine for use in particular environments are provided in the KES Knowledge Base Author's Manual (KES, 1990).

The production rule inference engine was selected for vehicle interior design, as this inference engine is appropriate where:–

1) The knowledge base is to be represented by rules, which may also use certainty factors. This is a way of expressing conditional relationships between attributes giving answers with degrees of confidence (See section 5.7.3.3).

2) Good control over inferencing is required. This has been achieved by a combination of controllable forward and backward chaining.

3) Classes and class inheritance may be used. Classes allow reasoning about groups of objects with the same characteristics. Class inheritance expresses hierarchical relationships between classes.

KES / Inference Engine	Production System	Hypothesize and Test	Bayes
Knowledge Representation	Production Rules	Descriptions Tables	N/A
Backward Chaining	Rules	Set Covering	Bayes
Forward Chaining	Demons	Demons	Theorem
Procedural Control	Action, Demons	Actions, Demons	Actions
Reasoning Method	Deductive	Abductive	N/A
Developer's Interface	Yes	Yes	
Incremental Parser	Yes	Yes	N/A
Certainty Factors	Numeric	Symbolic always to never	Probability
Classes With Variables	Yes	N/A	N/A
Class Inheritance	Yes	N/A	N/A
Relation Attributes	Yes	N/A	N/A
List Attribute	Yes	N/A	N/A
Through Function Calls	N/A	N/A	N/A
True Embedability	Yes	Yes	N/A
Externals	Yes	Yes	Yes
Explain Facility	Yes	Yes	Yes
Help Facility	Yes	Yes	Yes
Why Facility	Yes	N/A	N/A
Justify Facility	Yes	N/A	N/A
Trace Facility	Yes	Yes	Yes
Frame–like Representation	Classes	Description	N/A
Handles Unknown Values	Yes	Yes	Yes
Numerical Calculations and Functions	Yes	Yes	Yes
String Pattern Matching	Yes	Yes	Yes
Consistency (truth) Maintenance	Yes	N/A	N/A
Windowed Interface	Yes	Yes	Yes
Local Language Capability	Yes	Yes	Yes

Figure 5.5 – The main features of KES Expert System (KES, 1990).

4) Communication with the outside world is required. KES uses external communication files to call other expert systems, data bases or application software such as the SAMMIE system.

5) Consistency maintenance is required – in this way, the dependent values can be updated (or modified), so that they are consistent with new attribute values.

Expert system shells have significant potential for application in the SAMMIE system environment, for the provision of intelligence for ergonomics evaluation. SAMMIE has been considered as a design tool which provides assistance to the designer, and although it has predictive algorithms, it has no genuine intelligence. The usefulness of expert systems has been shown in research by Azarkadeh (1988) in devising a 'design shell' for control and instrument layout within automobiles. The KES expert system was used for the development of the domain knowledge and integrated with the SAMMIE computer man modelling systems for ergonomics design evaluation.

KES is C based and hence, in a 'C' environment it has the ability to be integrated with external programs, including communication with graphics packages. There are two phases to building a program with an embedded KES:–

1) Build a knowledge base that runs as a stand–alone expert system.
2) Integrate the KES expert system and the SAMMIE system.

5.7 Constructing the Knowledge Base

The knowledge base is a file containing the facts and heuristics that make up an expert's knowledge and the inference engine uses the knowledge base and information provided by the user to solve problems.

A knowledge base is represented in a different manner to a conventional data base. Data bases (and data base management systems) were originally developed to maintain records involving large volumes of data. Sophisticated data base methods allow entities and their relationships to be represented, and so a data base may also be viewed as a model of its domain. Thus, in a data base, knowledge about the domain may be implicity represented by the structure of the database.

The knowledge representation in KES consists of the knowledge information and the knowledge base, structured into the following sections; constants, attributes, classes, rules, actions and others depending on the application.

5.7.2 Knowledge Representation

A knowledge base developed with KES has to follow the conventions provided by the KES expert system shell. A Production System (PS) knowledge base can have up to ten non–mandatory sections, each of which contains and/or manipulates domain knowledge. These sections are as follow:–

1. **constants** 2. **texts** 3. **patterns** 4. **types**
5. **attributes** 6. **classes** 7. **externals** 8. **rules**
9. **demons** 10. **actions.**

A knowledge base can be developed using only the sections that are needed. Further explanation of the detail can be found in the KES Reference Manuals (KES, 1990), and by the examples provided in the knowledge bases in EDKBES.

Knowledge representation is the structuring and organisation of a body of facts into a knowledge base in a form that in some sense reflects the knowledge of an expert. Hence, the knowledge representation used needs to reflect the manipulation techniques available and to be supported by the shell inference engine.

5.7.2 Construction of Domain Knowledge

In constructing the prototype system, some basic knowledge of the domain and methods methods of knowledge acquisition was required. Chapter two identified the sources of knowledge from the technical reports, text books and standards such as BSI, ISO, SAE, FMVSS, ADR and EECD.

5.7.3 Examples of Domain Knowledge

The domain knowledge can be characterised as being in part undeterministic, interrelated and dependent. This includes: geometric data, ergonomics design; design principles, standards, styling and aesthetics, economics and marketing, manufacturing and design working practice.

For example, certain types of vehicle components such as the dashboard, displays, controls (visual indicators), instrumentation, driver's seat and passenger seat are required by law, whereas others may be for ergonomic, marketing, and manufacturing reasons. Some may be there only for stylistic reason for example the aesthetic appearance of the displays.

In designing the dashboard, the designer places most importance on functional, aesthetic and styling aspects which primarily govern the shape and size of the display objects within the satisfaction of basic minimum ergonomics, rules, standards and legislation. For example, conventional circular speedometer dials are adopted because they are more stylish and popular with the public, even through digital displays prove to be a better ergonomics design. Also, for aesthetic reasons, the speedometer may be larger than really necessary for accuracy of reading.

Ergonomics standards and legislation require some displays, controls and instruments to be located so that they are visible through the steering wheel for the 95th percentile driver eye ellipse, with no head movement and with minimum eye movement, since these displays are considered to be vitally important and used with high frequency. Other important displays may be located so that they are visible through the steering wheel but some head movement is permitted.

Manufacturing constraints, as another example, do not allow some displays to be located adjacent to each other because of electrical connections behind the display panel or conversely sometimes they need to be placed as close as possible.

Colours and colour contrasts are seen as important considerations for the elimination of glare, and visibility at night. This is an integration of both ergonomic functions and aesthetic factors.

5.7.3.1 Development of Production Rules

The following production rules are developed according to design principles and the standards and legislation described in Chapter 2. Examples of the production rules generated are described in the following chapter.

5.7.3.2 Attributes

Attributes represent the knowledge of the domain and are grouped in the **attributes** section of a KES knowledge base and contain the declarations of the global attributes represented in the attribute hierarchy, (figure 5.6). An attribute is assigned a value at run time; the **value** holds data specific to a particular 'end–user' (consultation) session with the expert system.

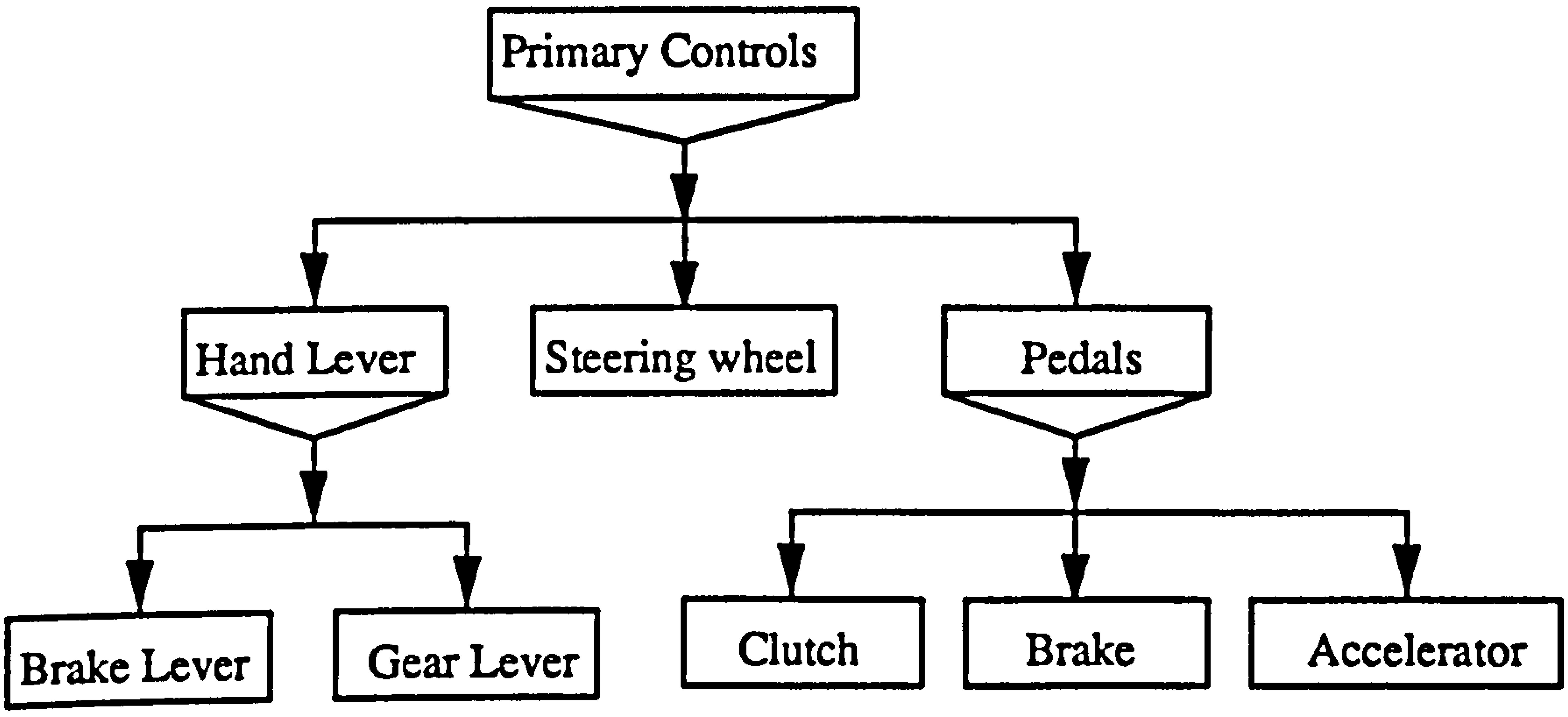


Figure 5.6 The Attributes Hierarchy of Primary Controls.

In the attributes section, there are two kinds of attributes declared. Firstly, the **input attributes**, whose values are determined by:–

i) The end user directly.

ii) Assertions in the actions section

iii) External programs

iv) The read command in the actions section.

Secondly, the inferred attributes, whose values are usually determined by **knowledge sources** such as:–

i) Production rules.

ii) Default Clauses.

iii) External programs

iv) Can also be determined by the source for input attributes.

The attribute values are determined either through question and answer dialogues where the end user answers questions or through the various knowledge sources in the system. The inference engine determines the values of the inferred attributes. An example of a simple attributes section would be:–

```
attributes:
        displays panel: sgl                      / name and type of attributes.
        (speedometer,                            / attributes values........
        tachometer,
        fuel indicator,etc.).
        angle of driver eye movement: int.       / attribute of integer type.
 limits: sgl
        (within legislation standards,
        outside legislation standards)
```

Knowledge sources, parts of the knowledge base, typically express relationships between attributes. Rules are unique to the KES PS subsystem. If the PS inference engine needs an attribute's value and it has not been specified in any of the knowledge sources for the attribute, then the inference engine will ask the end user to provide a value.

If one type of knowledge source cannot provide a value for the attribute, then the inference engine looks for another type. The PS inference engine consults an attribute's knowledge sources in the following order:

i. External programs

ii. Production rules.

iii Default or calculation clauses.

The obtain command, either in the actions section or entered by the end user, activates the PS inference engine. The format of this command is:–

```
(obtain      attribute name,
             attribute name,
             attribute name.)
```

This command provides the inference engine with the goal of finding attribute values for the attributes named in the command line. Detailed descriptions of the knowledge representations in the PS subsystem can be found in the KES Manual (KES, 1988). There are two types of attributes:–

1) Input attributes, whose values are determined by asking the user, or by assertions in the actions section or read from data files;

2) Inferred attributes whose values are usually obtained from knowledge sources – as rules, externals, or defaults and calculation clauses.

For example, the rule:–

```
if
        displays item  = tachometer and angle of driver eye movement gt 20
Then
    limits = outside ergonomics standards
endif.
```

The end user may provide the value of "tachometer", the measurements for input attribute "displays item" and the value of the other input attribute "angle of driver eye movement" may be obtained from a communication file. Subsequently the inferred attribute "limits" will be determined from this rule.

5.7.3.3 Certainty Factors

In many circumstances there is a degree of uncertainty such that it is not possible to assign absolute truth values to attributes or consequents of rules. These situations can be handled by KES through the use of numeric scales or symbols representing measures of belief which are known as ***Certainty Values***. In their numeric form certainty factors take values between 1.0 and –1.0 inclusive, and in symbolic form have the values a, h, m, l and n. The symbolic form is merely a convenient form for certain applications where each symbol represents a particular value so that a (always) = 1.0, h (high) = 0.75, m (medium) = 0.5, l (low) = 0.25 and n (never) = –1.0.

A certainty factor of 1.0 has the meaning ***absolutely certain*** or ***certainty of truth***, whereas a factor of –1.0 has the inverse meaning of ***absolutely certain not*** or ***certainty of not truth***. Certainty factors between 1.0 and –1.0 indicate some degree of uncertainty, including the factor of 0.0 that means that there is no more ***certainty of truth*** than there is ***certainty of not truth*** (KES, 1990).

Certainty factors can appear in expressions entered by the end user and in communication files, for example – an attribute could be assigned a value by:–

Car Colour = Red < 0.4 > | Green < 0.6 >

The vertical bar represents the or (disjunction) relationship and the certainty factors are contained within diamond brackets (< >). In this example there is a lower certainty (0.4) of the car being red and a higher certainty (0.6) of it being green. The sum of the certainty factors is 1.0 implying that all cars are red or green.

In the case of rules the certainty factors are a measure of belief in the consequent if the 'IF' condition is true. Thus for example the rules:–

Engine light rule:

```
if
        engine light on
then
        component to check = brakes < 0.1 >.
endif.
```

Brake light rule:

```
if
        brake light on
then
        component to check = brakes < 0.9 >.
endif.
```

have the IF condition "engine light on" and "brake light on" respectively and the consequent is the assignment of a value to the attribute "component to check". The certainty values of 0.1 and 0.9 are measures of belief in the choice of the most appropriate component.

In addition to user–assignment of certainty factors, KES also calculates default values, validates and re–assigns certainties in complex assignments and calculates values as part of the inferencing process. The complex rules associated with these calculations may be found in the KES Reference Manuals (KES, 1990).

5.7.3.4 Classes

The main body of the system knowledge was statically represented as "frame" (see chapter four, section 4.7.3.4) like structures in the classes section. Classes are used to represent information about groups of objects. The objects are members of a class that share a common set of attributes, with each object having its own attribute values.

The knowledge represented in the form of classes of objects is primarily related to the geometry of the objects and inferences about the properties and relationships between objects that accommodate geometric reasoning.

Classes were constructured to represent hierarchies which describe the relationships between objects – internally or externally, with other objects and with the supporting facts. Classes of knowledge are comprised of two main types; knowledge about individual objects and knowledge about groups of objects. An example of a Classes Section for the display panel and pedal controls is given below.

```
Classes:
    displays: [default: speedo, tacho, fuel]      /class name with default members.
attributes:
    shape: mlt                                    / class attribute and its type.
        (square, circle, triangle, rectangle)     / attributes values
    type: sgl                                     / class attribute and its type.
        (electromechanical, digital,
        curvilinear, discrete, etc.).             / attributes values
```

```
    area: real.
    relative position: mlt                    / class attribute and its type.
        (left-of, right-of, above, below).    / attributes values
    styling: mlt                              / class attribute and its type.
            (aesthetic,
            good ergonomic design,
            harmony,
            unpopular design,
            poor design, etc.).               / attributes values
%
endclass.

Classes:
    pedals: [default: accele, brake, clutch]  /class name with default members.
attributes:
    shape: mlt                                / class attribute and its type.
        (square, rectangle)                   / attributes values
    type: sgl                                 / class attribute and its type.
        (hand control, foot control,
        automatic, manual)                    / attributes values
    area: real.
    relative position: mlt                    / class attribute and its type.
        (left-of, right-of, centre-of,
        above, below, angle-of).              / attributes values
    styling: mlt                              / class attribute and its type.
            (aesthetic,
            good ergonomic design,
            harmony,
            popular design,
            poor design, etc.).               / attributes values
%
endclass.
```

Examples of other classes with default members (objects) are:–

```
    speedo: [default: left-most point, right-most point,
            lowest point, upper-most point]   /class name with default members.
attributes:
        cox: int.                             / attribute of integer type.
        coy: int.                             / attribute of integer type.
        coz: int.                             / attribute of integer type.
%
endclass.

st-wheel:                                     / another class, with members.
attributes:
```

```
        cox: int.                        / attribute of integer type.
        coy: int.                        / attribute of integer type.
        coz: int.                        / attribute of integer type.
%
endclass.
```

In this context expressions of the type,

```
i)      display: speedo> shape = circle
        display: tacho> shape = circle
ii)     speedo:right-most point> cox = 30 and
        tacho:left-most point> cox = 30
```

are used to describe (i) the shape of the speedo in the class of "displays" as being circular and (ii) to define the X coordinate of the right most point of the speedometer and the X coordinate of the left most point of the tachometer.

What is described, in this context, is a semantic representation (see chapter 4, section 4.7.3.2) of knowledge as well as the syntax. For example, the syntactic representation "right-of (speedo, tacho)" (as in Prolog) could eloquently describe the relationship between the objects. These are currently handled by manipulation rules, such as:–

```
    if
        display: speedo>relative position = right-of and
        display: tacho>relative position = left-of
    then
        display>styling = good ergonomics <0.5> | good styling <0.5>
endif.
```

Alternatively,

```
    if
        speedo: right point>cox lt -30
        tacho: left point>cox gt 30
    then
        display>styling = good ergonomics <0.5> | good styling <0.5>
endif.
```

Production rules can be written to be specific about objects or to be general, applying to all objects of a class. This is shown by a production rule on styling.

```
Styling rule:
        if
            displays panel: speedo > shape = circle and
            displays panel: tacho > shape = circle
        then
            displays panel> styling = good design <0.1> | balance <0.9>
    endif.
```

Variables can also be used to generalise a rule to apply to all class members:–

```
General rule:
        if
            displays > area lt 400 and
            displays > area gt 200
        then
            displays > styling = good design <1.0>.
        endif.
```

The styling rule says if the shape of the "speedo" that belongs to the class "displays" is "circle", and the shape of the "tacho" is "circle", then this panel display's styling is good aesthetically. The general rule, however, says if the area of any object in the "displays" is less than 400 square mm or greater than 200 square mm, then the panel display's styling is a good design.

It is possible to organise several classes into hierarchical relationships, referred to as "class inheritance". Class inheritance provides a powerful and flexible knowledge representation and makes it easier to manipulate the knowledge through rules.

5.7.3.5 Calculation Classes

Calculation classes are knowledge that is required in this paradigm, to check a design against the constraints including the visibility of a display through the steering wheel, the degree of eye movement and the (allowable) head movement.

A large number of calculations were required to compare the relative coordinates of various points on a display object with the coordinates of the points of the steering wheel and the coordinates of the eye point of the operator to establish " visibility" attributes. Also calculations were required to determine the amount of adjustment required for each object, if it did not satisfy the specification.

These calculations were represented in the classes section which later would be used as knowledge sources for values of attributes of the objects.

5.8 Conclusion

This chapter has described the methodology used in developing the Ergonomics Design Knowledge Based Expert System. The development of the production rules and the implementation of the two systems will be described in the next two chapters.

CHAPTER 6

DEVELOPMENT OF PRODUCTION RULES

6.1 Introduction

The following production rules are developed according to the design principles and the standards and legislation described in chapter 2.

The production rules are extracted from data sources, such as the British Standards Institution (BSI), the International Standards Organisation (ISO), the Society of Automotive Engineers (SAE), European Economic Community Directives (EECD), and the Federal Motor Vehicle Safety Standards USA (FMVSS).

The various recommendations from these sources can be represented by a collection of attributes (knowledge) embedded in the production rules, as described in the following sections.

6.2 Production Rules

Production rules are the knowledge source that constitute the primary method by which the inference mechanism/engine can determine the values of attributes. Other knowledge sources are also available through "Externals" and by the "default and calculation" classes. The inference engine is directed to use production rules as the main form of knowledge representation as described in chapter 4.

The following are production rules that dynamically manipulate the static knowledge representations in the classes and attributes sections and are used for design and styling purposes in the vehicle interior domain.

6.2.1 Use of Certainty Factors

In building the expert system certainty factors have been used as a means of informing the user of the certainty of the outcome of the rules.

An example of this is the evaluation of speedometer location which is required to be within the appropriate zone defined for example in BS AU 199, (1984).

The rule takes the form:–

```
speedo visibility:
    if
        speedo:tp>coz lt visible:panel>top limit and
        speedo:bp>coz gt visible:panel>bottom limit and
        speedo:lp>cox gt visible:panel>left limit and
        speedo:rp>cox lt visible:panel>right limit
    then
        speedo:area>visibility = visible through steering wheel.
    endif.
```

As all of the geometric attributes in the rule are known with absolute certainty (certainty factors of 1.0 by default) and if the 'IF' condition is found to be true (coordinates of speedometer's location lie within the limits of the zone) then the consequent of the rule will be given with absolute certainty as:–

```
speedo:area>visibility = visible through steering wheel < 1.0 >.
```

This presumes of course that the zone has previously been found to be not obscured by the steering wheel.

Other situations do not lead to such certain conclusions. As an example, a displays instrument styling rule is:–

```
if
    displays panel: speedo > shape = circle and
    displays panel: tacho > shape = circle
then
    displays panel> styling = good design < 0.1 > | balance < 0.9 >.
endif.
```

Certainty factors are used here to convey the strong certainty that providing that both the speedometer and tachometer are circular this leads to a balanced design and a weaker certainty that this also represents good design.

In this research the attributes for the design criteria and their certainty factors have been selected through personal opinion, observation and experience. The attributes chosen are:–

good ergonomics,
good design,
good styling,
popularity,
balance,
aesthetics.

Good ergonomics. The vehicle interior design should comply with accepted ergonomics practice and relevant standards.

Good design. The functional requirements of the design are achieved.

Good styling. Styling relates to the selection of appropriate shapes for objects such as instruments.

Popularity. Popularity refers to commonly expected or stereotyped features, such as the shape and arrangement of foot pedals.

Balance. Conformity in shape and arrangement. For example the consistent use of circular dials.

Aesthetics. This is concerned with form, colour, style and the compatibility of instruments within the overall display environment.

Measures of attaining these criteria are expressed to the uses as certainty factors on the following scale:–

Certainty Factor	Criteria
0.75 – 1.0	"excellent"
0.5 – 0.75	"good"
0.25 – 0.5	"average"
0.0 – 0.25	"low"
–1.0 – 0.0	"inadequate"

6.2.2 Production Rule – Primary Displays

The construction of a production rule is based on the frame system where a class of objects is described by a fixed format containing attributes which differentiate between members of the class.

The frame below is an example of the "displays" class where the connection slot gives class members such as " speedometer" and "tachometer". Attributes such as "shape", "relative position" and "area" define characteristics of particular displays.

SLOT/ATTRIBUTE	VALUE
Name:–	Displays
Slot:–	Speedometer, Tachometer, Panel
Attributes:–	Shape and Styling
Expression:–	mlt
width	200 mm
length	400 mm
shape:–	Square
	Circle
styling:–	good ergonomics
	good design
	good styling
	good aesthetics
	balance
relative position:–	right of
	left of
	above
	below

Production rule 1, shown below, is a styling rule, that says if the shapes of both the "speedometer" and the "tachometer" that belong to the "displays" are "circle" then this panel's styling has good balance. i.e this rule ensures consistency in the shape of the two major displays on the panel. The value of the balance certainty factor is in the category of "excellent". Functional capabilities are only weakly influenced by this consideration and thus the certainty factor for good design has a low value.

Rule 1

```
    if
         displays:speedo> shape = circle and
         displays:tacho> shape = circle
    then
         displays:panel > styling = good design <0.1> | balance <0.9>.
    endif.
```

Similarly, styling rule 2, states that if the position of the "speedometer" is to the "right of" the centre line datum and if the "tachometer" is to the "left of" the same datum then this panel's styling is ergonomically sound (complies with standards) and of good styling. The values of certainty factors are in the category of "average" design.

Rule 2

```
    if
        displays:speedo> relative position = right of and
        displays:tacho> relative position = left of
    then
        displays:panel> styling = good ergonomics <0.5> | good styling <0.5>.
    endif.
```

Rules 3 and 4 are general rules, that say if the area of any object in the "displays" class is more than 400 square mm or less than 200 square mm, then the panel's styling is very good from the point view ergonomics and of good design. i.e. this rule ensures consistency in the size of displays and that they are neither too large nor too small. The value of certainty factors is in the category of "excellent" design and ergonomics.

Rule 3

```
    if
        displays>area lt 400 and
        displays>area gt 200
    then
        displays>styling = good ergonomics <1.0>.
    endif.
```

Rule 4

```
    if
        displays>area le 400 and
        displays>area ge 200
    then
        displays>styling = good design <1.0>.
    endif.
```

Other, similarly constructed production rules are used to evaluate the type of instrument (analogue, digital, etc.) and the use of colour based on the frame below.

SLOT/ATTRIBUTE	VALUE
Name:–	Displays
Slot:–	Speedometer, Tachometer, Panel

Attributes:– Instrument and Colour
Expression:– sgl
Type of instrument:– analogue
discrete
digital

Colour:– Red: danger, or damage to equipment immediate or imminent,
hot in climate control system or engine temperature indicators.
Yellow: caution, vehicle system malfunction, danger to vehicle likely,
or other conditions which may produce hazard in the longer term.
Green: safe, normal operation of the vehicle system,
air–conditioner indicator system functioning.
Blue: driving upper high beam tell–tale indicator on, and cold,
in climate control systems or temperature indicators.
White: other conditions where none of the above colours are appropriate.

Rule 5, is a styling rule, that says if the "type" of the "speedometer" and "tachometer", both members of the "displays" class, are both "analogue" or both "digital" then this panel's styling is a good design and has good balance. i.e. this rule tends to prevent the mixing of different display types on the same panel. The value of certainty factors is in the category of "excellent" design.

Rule 5

if
displays:speedo>type = analogue and
displays:tacho>type = analogue
then
displays:panel>styling = good design <0.1> | balance <0.9>.
endif.

Rule 6, the display's colour rule, says that if the colour of any instrument in the "displays" class is for example assigned the colour "blue" then the message "colour blue, driving upper high beam tell–tales indicator on, and cold, in climate control systems or temperature indicators = blue <1.00>" will be displayed. The value of certainty factors is in category of "excellent".

Rule 6

if
Colour is Red, danger, or damage to equipment immediate or imminent,
hot in climate control system or engine temperature indicators.
Colour Yellow, caution, vehicle system malfunction, danger to vehicle likely,

or other conditions which may produce hazard in the longer term.
Colour Green, safe, normal operation of the vehicle system,
air–condition indicator system functioning.
Colour Blue, driving upper high beam tell–tales indicator on, and cold,
in climate control systems or temperature indicators.
Colour White, other conditions where none of the above colours are appropriate.
status (displays:panel>colour) = known
then
message" combine ("colour is rated to be blue is driving upper high beam tell–tales
only, and cold, in climate control systems or temperature
indicators. displays:panel> colour).
endif.

6.2.3 Production Rule – SRS Airbag

Rules related to the SRS airbag also use the frame method as described below:–

SLOT/ATTRIBUTE	VALUE
Name:–	Steering Wheel
Slot:–	SRS Airbag
Attributes:–	relative position
Expression:–	mlt
relative position:–	centre
	right of
	left of
	angle of
steering wheel impact	50 milliseconds
seat	anti–submarine ramps
seat	safety belt fastener

The class object "steering wheel" has a member called "SRS airbag". Rule 7 states that if the relative position of the "steering wheel" is to the "centre" of the displays and at an "angle of" to the horizontal then this steering wheel's airbag system is good from the point of view of functional design. The value of the certainty factors is in the category of "excellent" design.

Rule 7
if
st–wheel:SRS airbag > relative position = centre of and
st–wheel:SRS airbag > relative position = angle
then

st–wheel > SRS airbag = good design <1.0>.
endif.

6.2.4 Production Rule – Pedals

The pedal controls are represented by the following frame:–

SLOT/ATTRIBUTE	VALUE
Name:–	Pedals
Slot:–	Accelerator, Brake, Clutch, Controls
Attributes:–	Shape and Styling
Expression:–	mlt
width x length	80 – 100 mm (brake and clutch)
width x length	80 – 150 mm (accele)
distance	50 – 100 mm (accele to brake, brake to clutch)
angle	87 degrees
height	152 mm (from floor)
Shape:–	rectangle
Styling:–	good ergonomics
	good design
	good styling
	good aesthetics
	good balance
relative position:–	right of
	left of
	above
	below
	centre of

The styling rule 8, says that if the shape of the "accelerator", "brake" and "clutch" are all "rectangle", then this control pedal's styling is a good design and of good balance. i.e this rule ensures consistency in the shape of the pedals. The value of certainty factors is in the category of "excellent" design.

Rule 8

```
if
    pedals:accele>shape = rectangle and
    pedals:brake>shape = rectangle and
    pedals:clutch>shape = rectangle
then
    pedals:controls>styling = good design <0.1> | balance <0.9>.
endif.
```

Styling rule 9 for the "pedals" class, says that if the position of the "accelerator" is to the "right of" the foot controls centre line, "brake" is to the "centre of" and the "clutch" is to the "left of" then the control pedals' layout represents good ergonomics and good styling. The value of certainty factors is in the category of "excellent" ergonomics.

This rule enforces the layout of the foot controls in the stereotyped order as required by legislation.

Rule 9

```
    if
        pedals:accele>relative position = right of and
        pedals:brake>relative position = centre of and
        pedals:clutch>relative position = left of
    then
        pedals:controls>styling
            = good ergonomics <0.9> | good styling <0.1>.
    endif.
```

Rule 10 is a general rule that says that the distance between two pedals such as the "accele" and the "brake" should be greater than 50 mm and less than 100 mm for good ergonomics.

Rule 10

```
    if
        pedals:accele>distance gt 50 and
        pedals:brake>distance lt 100
    then
        pedals:controls>styling = good ergonomics.
    endif.
```

Rule 11 is a general rule that says that the distance between the two pedals "brake" and "clutch" should be greater than 50 mm and less than 100 mm for good ergonomics.

Rule 11

```
    if
        pedals:brake>distance gt 50 and
        pedals:clutch>distance lt 100
    then
```

pedals:controls>styling = good ergonomics.
endif.

Rules 10 and 11 thus ensure appropriate pedal spacing within the stereotyped layout defined by rule 9.

Rules 12 and 13 are general rules that control the size, position and angle of the foot pedals. The value of the certainty factor for design is in the category of "excellent".

Rule 12

```
if
    accele width > gt 80 and
    accele distance > gt 50 and
    accele angle > lt 87 and
    accele length > gt 150
then
    accele >styling = good aesthetic <0.1> l good design <0.9>.
endif.
```

Rule 13

```
if
    brake width > gt 80 and
    brake distance > gt 50 and
    brake angle > lt 87 and
    brake height > gt 152
then
    brake > styling = good aesthetic <0.1> l good design <0.9>.
endif.
```

6.2.5 Production Rule – Seats

The frame representation for driver's and front passenger seat is shown below:–

SLOT/ATTRIBUTE	VALUE
Name:–	Seat
Slot:–	Driver's Seat and Front passenger Seat
Attributes:–	Distance
Expression:–	mlt
width	380 mm
length	480 mm
back rest height	127–177 mm maximum (adjustment back rest height)
adjustable height	50 mm (vertical height from floor to SgRp H–point)

back rest angle	15–40 degrees maximum.
statement:–	distance is too far
	distance too short
	move forward
	move backward

Rule 14, the seat rule, says that the seat width should be greater than 380 mm, seat length greater than 480 mm, seat back rest height greater than 127 mm, and the seat angle greater than 15 degrees for good ergonomics. The value of certainty factors is in the category of "excellent" ergonomics.

Rule 14

```
    if
        Seat width > gt 380 and
        Seat length > gt 480 and
        Seat back rest height > gt 127 and
        Seat angle > gt 15
    then
        Seat > styling = good ergonomics <1.0>.
    endif.
```

6.2.6 Production Rule –Visibility

The visibility rules are designed to evaluate the visibility of various instruments with ranges of eye and head movement. The frame shown below contains the necessary information.

SLOT/ATTRIBUTE	VALUE
Name:–	Displays – visibility
Slot:–	Speedometer, Tachometer and Panel
Attributes:–	Visible, Eye move, Head move
upper	15 degrees.
lowest	30 degrees.
right	45 degrees.
left	65 degrees.
angle	30 degrees.
real:–	cox
	coy
	coz
geometric position:–	top adjustment: real.

```
bottom adjustment: real.
left adjustment: real.
right adjustment: real.
```

The rule 15, is a visibility rule which in this example governs the locations of the speedometer for visibility.

The rule also provides recommendations for improving visibility by suggesting correction movements in the four directions up, down, left and right relative to the panel.

Rule 15

```
if
    speedo:tp>coz lt visible:panel>top limit and
    speedo:bp>coz gt visible:panel>bottom limit and
    speedo:lp>cox gt visible:panel>left limit and
    speedo:rp>cox lt visible:panel>right limit
then
    speedo:area>visibility = visible through st-wheel.
endif.
```

speedo correcting tp:

```
if
    speedo:area>visibility = not visible through st-wheel and
    speedo:tp>coz gt visible:panel>top limit
then
    sp>top adjustment = vis>move speedo down.
endif.
```

speedo correcting bp:

```
if
    speedo:area>visibility = not visible through st-wheel and
    speedo:bp>coz lt visible:panel>bottom limit
then
    sp>bottom adjustment = vis>move speedo up.
endif.
```

speedo correcting lp:

```
if
    speedo:area>visibility = not visible through st-wheel and
    speedo:lp>cox lt visible:panel>left limit
then
    sp>left adjustment = vis>move speedo to right.
endif.
```

```
speedo correcting rp:
    if
        speedo:area>visibility = not visible through st–wheel and
        speedo:rp>cox gt visible:panel>right limit
    then
        sp>right adjustment = vis>move speedo to left.
    endif.
```

Also, it is possible to have additional information attached to the rules which may be used to inform the user about the reasoning behind the rule.

For an example of additional information about the reasoning behind the above rule the user would have to respond "yes" to the message as follows.

```
    if reason = yes then
        message banner, " ".
        justify speedo:tp>visibility.
        message banner, " ".
    endif.
```

An example of the supporting knowledge sources and the rules is:–

```
groupII head move outside limit:
v_w: view_with
    if
        view_with:head move > horizontal left gt 15 or
        view_with:head move > horizontal right lt –15 or
        view_with:head move > vertical upwards gt 15 or
        view_with:head move > vertical downwards lt –15
    then
        v_w>gII head limit = outside gII head limit.
    endif.

{references: ISO}
{explanation: "The identification and those parts of      ",
              " the display area required to indicate     ",
              "a critical condition shall be visible      ",
              "without head movement.                     ",
              "The remaining parts of the display         ",
              "shall also be 'visible'; for these,        ",
              "head movement is permitted.                 "}.
```

6.3 Actions

The actions section of the knowledge base directs the operations of the expert system and guides it to seek the appropriate data and to inform the user about relevant information generated by the system.

Examples of commands in the actions section include:–

what would you like to start with?
It would be a good idea to go through each stage systematically.
1. Check visibility through the steering wheel
2. Check eye movement within the ergonomics range
3. Check that no head movement is required for visibility
4. Finished with this menu

The expert system attempts to determine the values of one or more inferred attributes. The values are determined through a question and answer dialogue – where the user answers questions that provide the values of input attributes and the inference engine determines the values of inferred attributes. All these are done through a series of commands in the actions section. The dependency hierarchy of the inference mechanism/engine of the EDKBES can be found in Appendix 4.

6.4 Conclusions

This chapter has described the development of production rules for the Ergonomics Design Knowledge Base Expert System, the implementation of which is discussed in the next chapter.

Rules 7 and 11–14 have been defined and the knowledge to support them has been collected. However their implementation into the prototype system has not yet been undertaken.

CHAPTER 7

IMPLEMENTATION

7.1 Introduction

This chapter discusses the implementation of the Ergonomics Design Knowledge Base Expert System (EDKBES) and the integration of the EDKBES in SAMMIE for vehicle interior design. The purpose and approach to the implementation and integration, and the facilities in the KES expert system tools are described, as are organisation of the system, the system's modules and the implementation of overlays.

7.2 Implementation Approach

The implementation approach adopted is to establish communication between the two components, i.e. a model of a vehicle interior in SAMMIE and the EDKBES expert system. The EDKBES expert system menu has commands to open the communication file and also there are facilities in SAMMIE to enter and retrieve various geometric information from the design model. The geometric information consists of the data required by the expert system for generation and manipulation of various coordinates of the vehicle interior design.

Communication is achieved by the expert system through reading of the appropriate data from the communication file. Then, through evaluating this data the faults of the design can be diagnosed, monitored and recommendations made against the legislation, design rules and regulations and ergonomics information and standards.

7.3 Implementation Facilities in Expert System Tools

Implementation can be more or less complicated depending on the tool used (Peterson et al, 1990). The facilities implemented have been described in chapter 4, and can be briefly summarised as:–

1) WHAT and WHICH facilities. The WHAT facility makes it possible for the user to ask the system what question was asked and the WHICH facility makes it possible to ask how the system reached a certain conclusion.

2) Hypothetical reasoning, where the system explains what would have happened if a particular rule had been different or if a different precondition had been true.

3) A graphical presentation of the SAMMIE model as defined by the database.

4) A graphical presentation of the SAMMIE model generated as a result of the reasoning chain that leads to a particular conclusion.

5) A simulation aid which make it possible to test the system response with simulated input and output communication and command files.

6) The possibility of saving and utilizing scenarios or situations.

7.4 The System Structure

The structure of the prototype system is outlined in this section. The system was designed to conform to the requirement of the ergonomics design knowledge base expert system (EDKBES) as described in chapter 5.

An overview of the prototype system is shown schematically in figures 7.2 and 7.3, and describes the relationship among functional modules of the system according to the flow of information within the system. These modules were designed separately in three groups according to their respective functions. The modular system approach to design is adopted as this approach is compatible with future extensions to the knowledge base. A modular system is also desirable with regard to improving or editing and readability of the content of each module.

In the KES shell system,the knowledge base is processed into "parsed knowledge" (figure 7.1). The EDKBES program can be tested and the 'runtime' programs can be executed more quickly if the parsed knowledge base consists of number of small modules rather than a single entry. The EDKBES knowledge base modules are written in KES, and the external program which communicates between the knowledge base modules, the design parameter calculations and reasoning and the communication files are written in the KES shell environment. The external program to communicate between the EDKBES modules and the SAMMIE modules and the command files are written in the SAMMIE command language as are the SAMMIE modules.

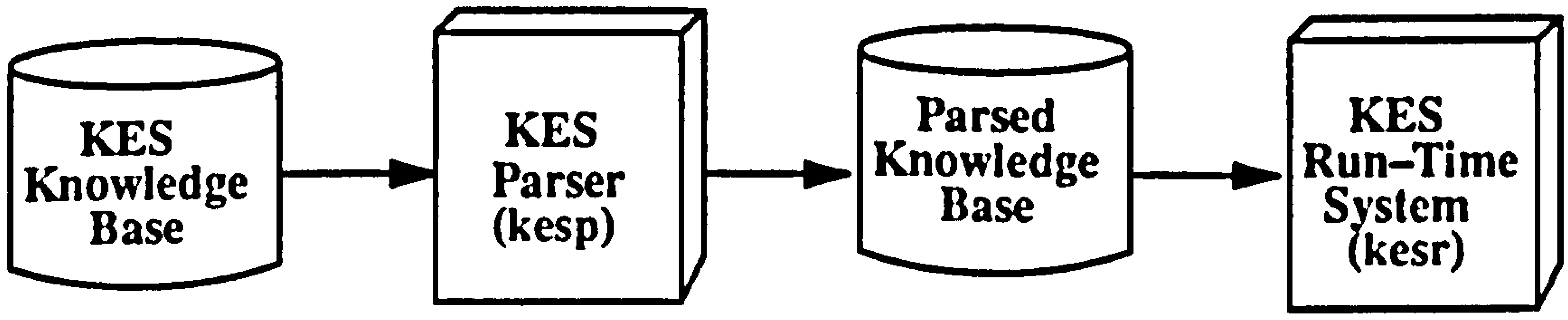


Figure 7.1 Developing and testing the Knowledge Base.

7.4.1 Prototype System Modules

The prototype system contains the following modules:-

Introduction modules which provide for information flow control, the introduction to the system and an explanation of the use of the system.

An **ergonomic design capability module** which informs the user of the capabilities of the system in monitoring and checking the design against the knowledge base.

A **text references module** which provides information on the sources used for the knowledge base.

Design parameters calculation modules, used to obtain geometric information from SAMMIE (e.g speedometer visibility through the steering wheel, views with eye and head movement).

Communication file modules for reading and writing of KES communication files.

Command file modules for executing SAMMIE commands including the reading and writing of KES communication files.

The SAMMIE database modules which are used to generate the prototype model of the vehicle interior design, provide geometric information, and to perform ergonomics evaluations.

The EDKBES Knowledge Base module which controls the interactions between the following EDKBES modules.

The EDKBES information for design module which controls the knowledge base containing information on standards and legislation.

The EDKBES displays; speedometer and tachometer module which contains the production rules for visibility through the steering wheel with eye movement and no head movement.

The EDKBES displays; fuel, temperature and door open module which contains the production rules for visibility with allowable eye and head movement.

The EDKBES pedals; accelerator, brake and clutch module which contains the production rules for reachability using seat adjustment and leg movement.

7.5 The Modular System

This section describes the main controlling program and various part modules of the Ergonomics Design Knowledge Base Expert System. These modules have been briefly described above and have been organised using an overlaying technique controlled by the main program, as described below.

7.5.1 Sectioning the Knowledge Base

The structuring of the knowledge base into sections or modules is to serve several aims:-

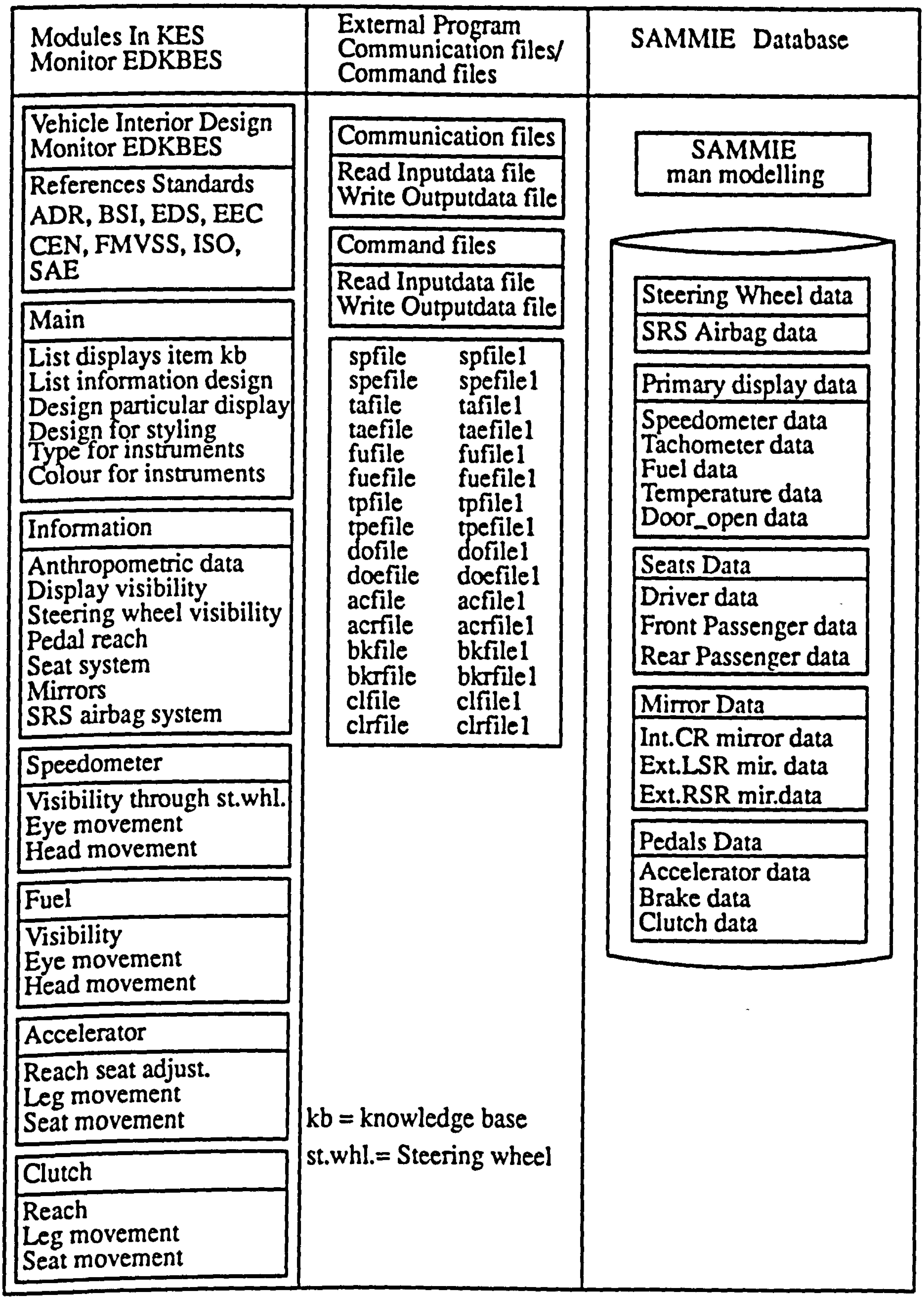


Figure 7.2 Overview of the Structure of the Prototype system.

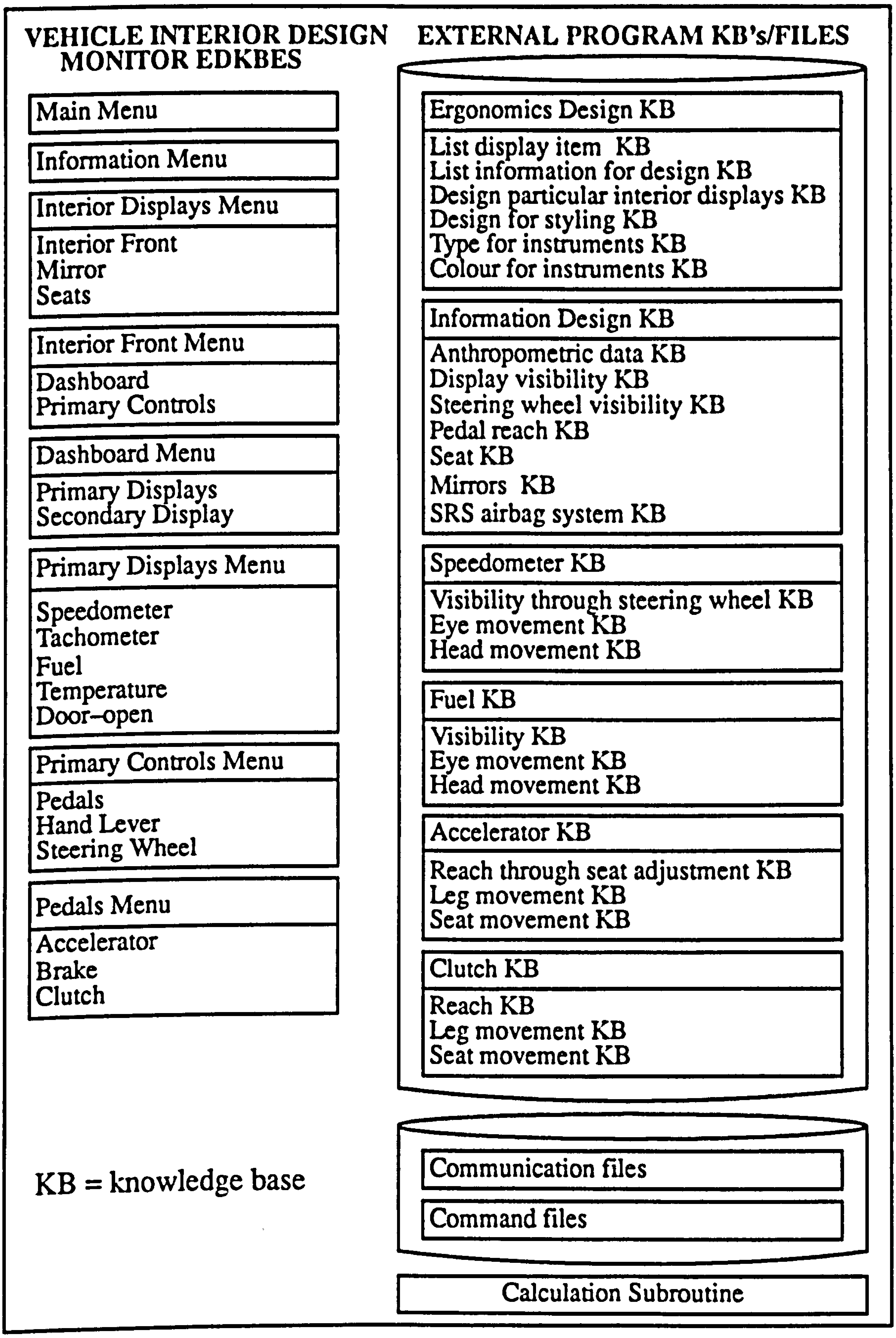


figure 7.3 The Modules of the Prototype Ergonomics Design Knowledge Base Expert System (EDKBES).

1) To minimise the time taken by designers or engineers where the overall design problem can be reduced to a series of smaller sub–problems. In this way each module is designed to tackle a specific task which contributes to the overall design process.

2) To avoid the problem of infinite chaining of the rules in the knowledge base. Sectioning results in each rule base containing fewer rules, each with it own function. Thus, the number of active rules that are chained during operation of the inference engine will be less and this reduces the chance of infinite chaining occurring.

3) To reduce the run time of the program. Without sectioning, all rules will be searched and executed by the inference engine. The searching is sequential from the top of the knowledge base. If the appropriate rule to be fired is situated at the end of the knowledge base this will increase the run time of the searching process.

4) To ease debugging and editing of the rule base, and also to facilitate the future expansion of the knowledge base. Adding new knowledge to a small modular knowledge base is less complicated and faster, than adding to a single large knowledge base.

The body of the EDKBES is sectioned into several groups which follow the hierarchical structure of the vehicle interior. The knowledge bases, are:–

1) The Vehicle Interior Workplace; seats, dashboard, primary controls, and mirrors.
2) The Seats; driver seat and passenger seat.
3) The Dashboard; Primary displays and secondary controls.
4) The Primary Controls; hand lever, steering wheel and pedals.
5) The Pedals; accelerator, brake and clutch.
6) The Mirrors; interior and exterior mirrors.

7.6 System's Message Facilities

Message facilities are normally available on standard software packages and are generally considered essential to the commercial and user acceptability of a package. Some packages have good facilities that provide message information instantaneously wherever it is required. On the other hand some message facilities are less efficient and the message needed is not directly accessible nor always available. In such cases, insufficient messages require the user to refer to the manuals for further information before continuation.

The integration of EDKBES within the SAMMIE system ensures that message facilities are provided throughout the system so that, as far as possible, a novice user will be guided in using the software without referring to the software manuals. These message facilities can be accessed in each of the options available in a particular menu-driven application and all sections of the knowledge base. A full description of message facilities and their use is given if the option 'Return' to proceed, type 'c' to continue, or 's' to stop is selected. "About the System", may be selected from the Main Menu at the commencement of a consultation or whenever the Main Menu is displayed. For a full system description refer to the KES files in appendix 3.

7.7 Requirements for the Integration of EDKBES within the SAMMIE System

In order to achieve the integration of EDKBES within SAMMIE, the following criteria have been identified.

1) The system should have an expert system shell with good integration capability. The KES expert system shell was selected for the reasons discussed in Chapter 5. This chapter discusses the communication capability of the KES shell and the SAMMIE command processing system.

2) The EDKBES – KES expert system should be capable of communicating with other applications using external programs by read/write communication files.

3) The SAMMIE system database for model data storage/retrieval should be easy to use and have the support of a graphical display, whether used directly by designers or external programs.

At present, graphical representation of the vehicle interior design and its dimensions are available in the SAMMIE system, and it is possible to interface the EDKBES with SAMMIE by using external programs and the command processing system.

4) The system should have future expansion capability. This could be achieved using other KES methods of software embedding such as the use of languages such as C and C++.

7.8 Integration of EDKBES within the SAMMIE System

The KES expert system shell was partly chosen because it provides the flexibility for integrating knowledge-based systems. Integration is used here to describe any interaction between the knowledge-based system with other knowledge based or other software applications. Typical interaction includes:–

1) Program calls made through the operating system, or information being passed between two applications through memory, (for example, when two expert systems are run concurrently but with different knowledge bases and exchange information/data or transfer information/data to other external programs.)

2) Program commands in the system to interact with the host operating system to execute other necessary functions or programs.

3) Program commands to retrieve or read information and data from external programs as input to the knowledge base, for example, a database program or a program that contains algorithms to calculate some values required by the knowledge base.

4) An application program that calls an expert system to acquire expert advice or expert decisions.

In this research work the most important aspects of the integration were realised by external programs using KES communication files and SAMMIE command files. This allows SAMMIE, a conventional and non intelligent application, to access expert system techniques in solving a problem or making a decision involving "geometric reasoning".

To implement the system and to establish communications between the two components, a model of the interior of a vehicle was used as an example of the general technique. The two main example models are the dashboard and pedal controls. The dashboard subsection consists of the primary and secondary controls and the primary and secondary displays. The primary displays consist of a variety of instruments such as a speedometer, tachometer, fuel gauge, temperature gauge, door–open indicator, and direction indicators. There are also a number of warning signals and indicators.

With the steering wheel in position and various models of the driver in adjustable seat positions the relative locations of objects in space can be determined and various views from the driver's eye point obtained (figures 7.4 and 7.5).

The ergonomics design knowledge base expert system (EDKBES) menu has commands to open the communications file and there are also facilities in SAMMIE to enter and retrieve various geometric information relating to the vehicle interior. The geometric information consists of the data required by the EDKBES for manipulation and includes various coordinates of the display panels, the instruments, steering wheel, eye position and angles of eye and head movements required to see the displays on the panel.

The interface capability plays an important role where the EDKBES is integrated with the SAMMIE system for evaluation or "geometric reasoning", and for situations that require external communication or exchange of information datafiles. The integration of the EDKBES within SAMMIE system will be described in the next chapter.

The KES shell can communicate with other programs or external files by three methods; 1) software embedding of the KES shell, 2) external functions and 3) read/write message commands as described below.

7.8.1 Embedded Programs

A conventional application accesses the expert system for assistance in solving a problem or making a decision. In the first method (embedded program) a KES expert system becomes part of a single executable program using languages such as 'C' or 'C++'. Such a method is suitable for fully developed and tested applications but is somewhat cumbersome for prototyping new systems.

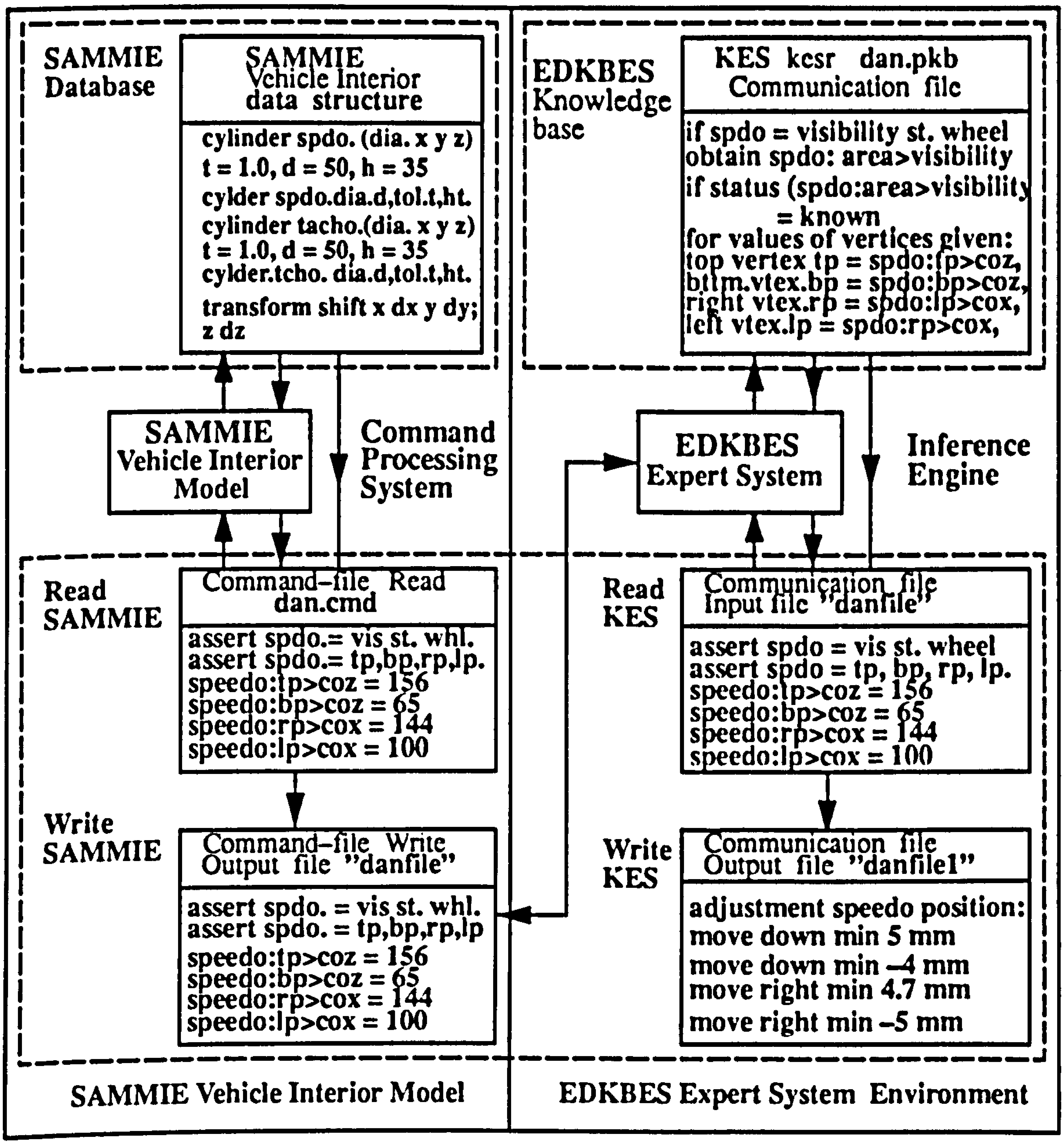


Figure 7.4 External program " displays panel" – integration KES within SAMMIE using Communication files and Command files.

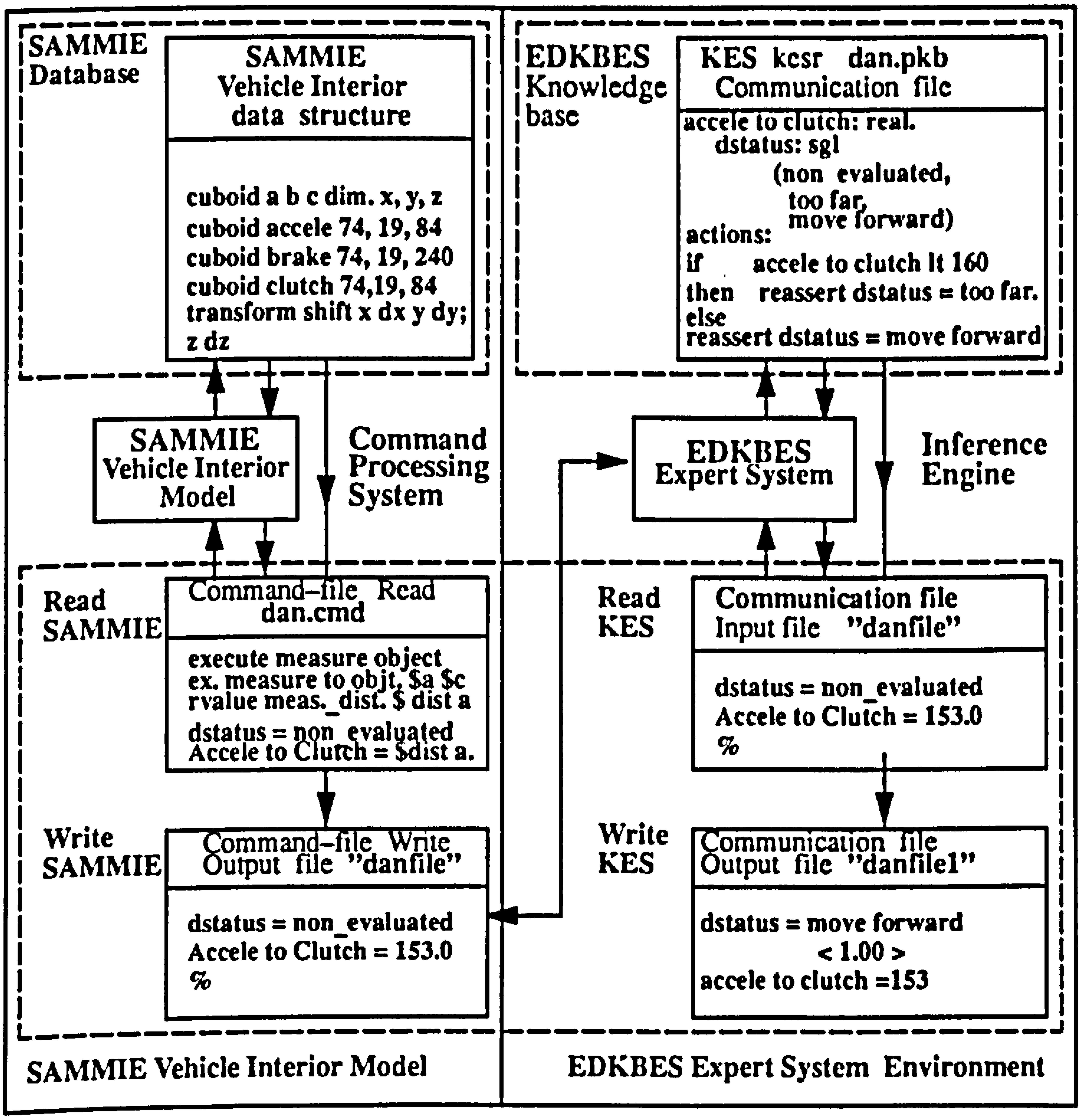


Figure 7.5 External program "pedals control" – integration EDKBES in SAMMIE using Communication files and Command files.

7.8.2 External Program/Function

An alternative method of integrating a KES expert system within another application is to use an external program. External programs are a simple way of communicating with executing programs outside of the expert system environment by sending or receiving information via communication files (Figure 7.7).

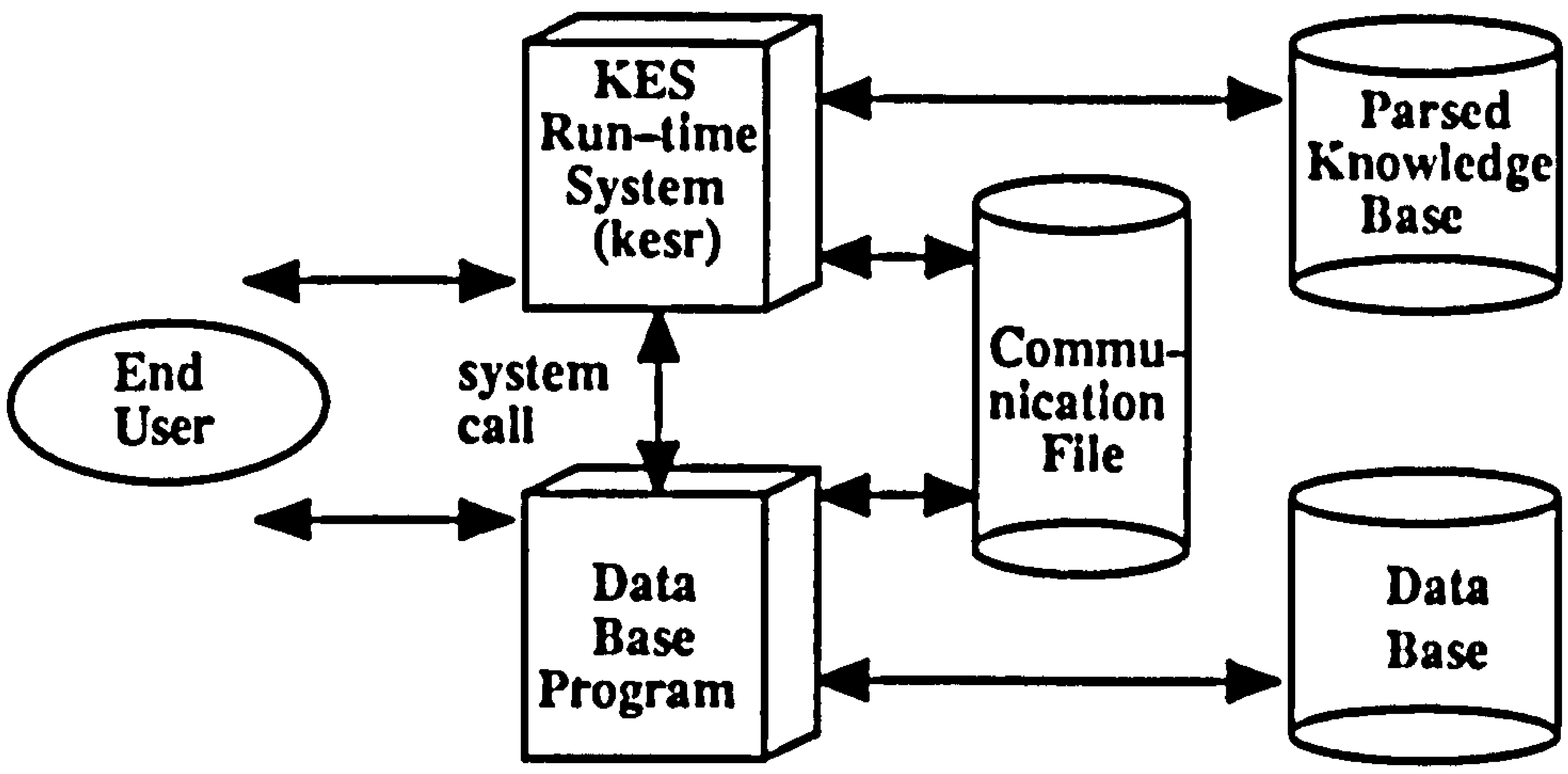


Figure 7.7 External Program KES Expert System.

Communication with an external program in this way is much slower than using an embedded KES expert system because there is no direct link between the knowledge base and the program; instead system calls and communication files are used to exchange data (Figure 7.8)

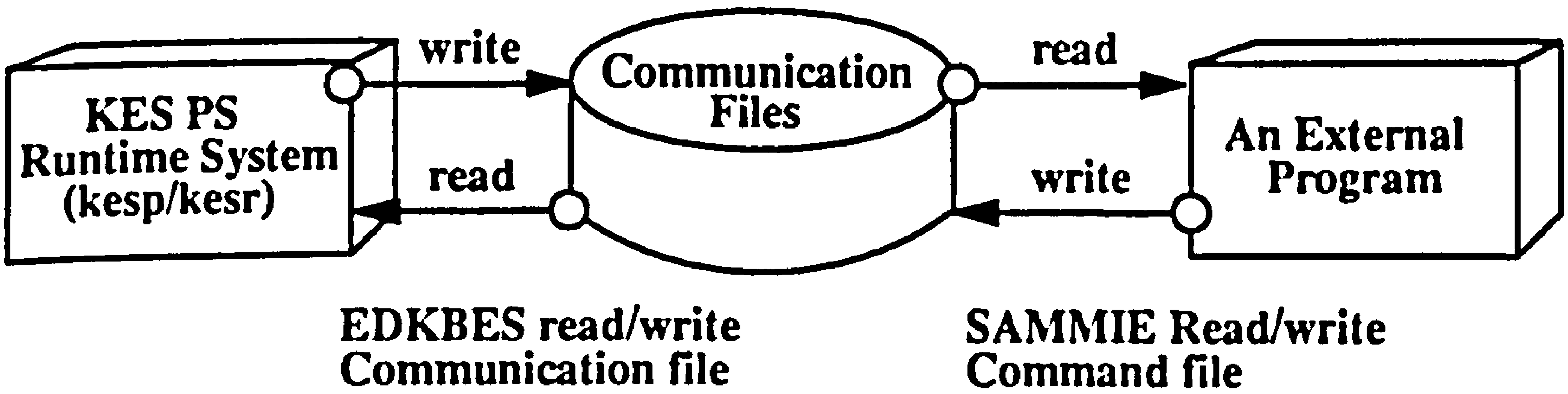


Figure 7.8 Communication between a KES expert system
and an external program

The Actions section of the knowledge base directs the operation of the Expert System and guides it to seek the appropriate data and to inform the user about relevant information generated by the system.

The knowledge base can communicate with an external program or file through one or more of the following methods (figures 7.8 and 7.9): –

i) read, write and message communication files in the actions section or embedded in the rules section of the knowledge base for the KES shell and command files for the SAMMIE system.

ii) interfaces defined in the externals section of the knowledge base.

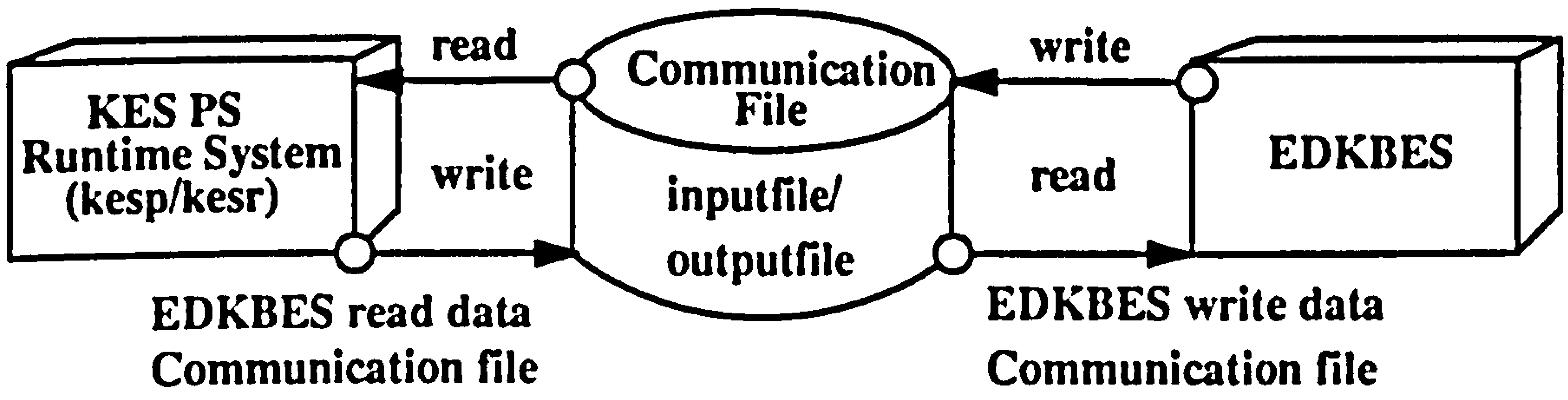


Figure 7.9 Use of read and write commands through a communication file.

In this research work the most important aspects of the integration of EDKBES within SAMMIE are achieved by external programs.

In the integration of EDKBES within the SAMMIE system, external functions are used to interact with the SAMMIE operating system to run the search program (command file) for data retrieval from the SAMMIE database. For example data stored in the SAMMIE database for measured distances are retrieved in the form of read and write SAMMIE command files (figures 7.4 and 7.5).

This is a simple one–way connector in which the SAMMIE system extracts and transfers geometric information from the design models. KES subsequently identifies and utilises the appropriate data for evaluation.

The interaction can also involve program command files in the system which interact with the host operating system to execute other necessary functions or program command files.

Program command files write or read information and data from external programs as input to the knowledge base. For example this technique is used for a database program or a program that contains algorithms to calculate some values required by the knowledge base.

7.8.2 Read, Write and Message Command

There are a number of function commands provided by the KES shell which are used for example, to read or write the attribute values of the coordinates of the speedometer object to object distance parameters to or from a datafile, or the appropriate communication file (figures 7.10 and 7.11).

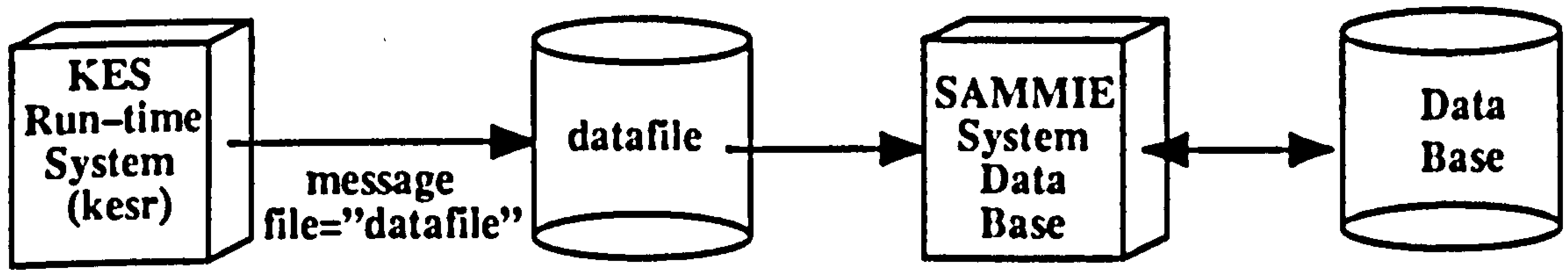


Figure 7.10 Example of Message Command

Communication was achieved by the EDKBES reading the appropriate data from the communication files. Then, through evaluating this data the faults of the design could be diagnosed and recommendations proposed. However, communicated information passed to and from SAMMIE was in command files in a different format to that of the KES expert system (figure 7.12).

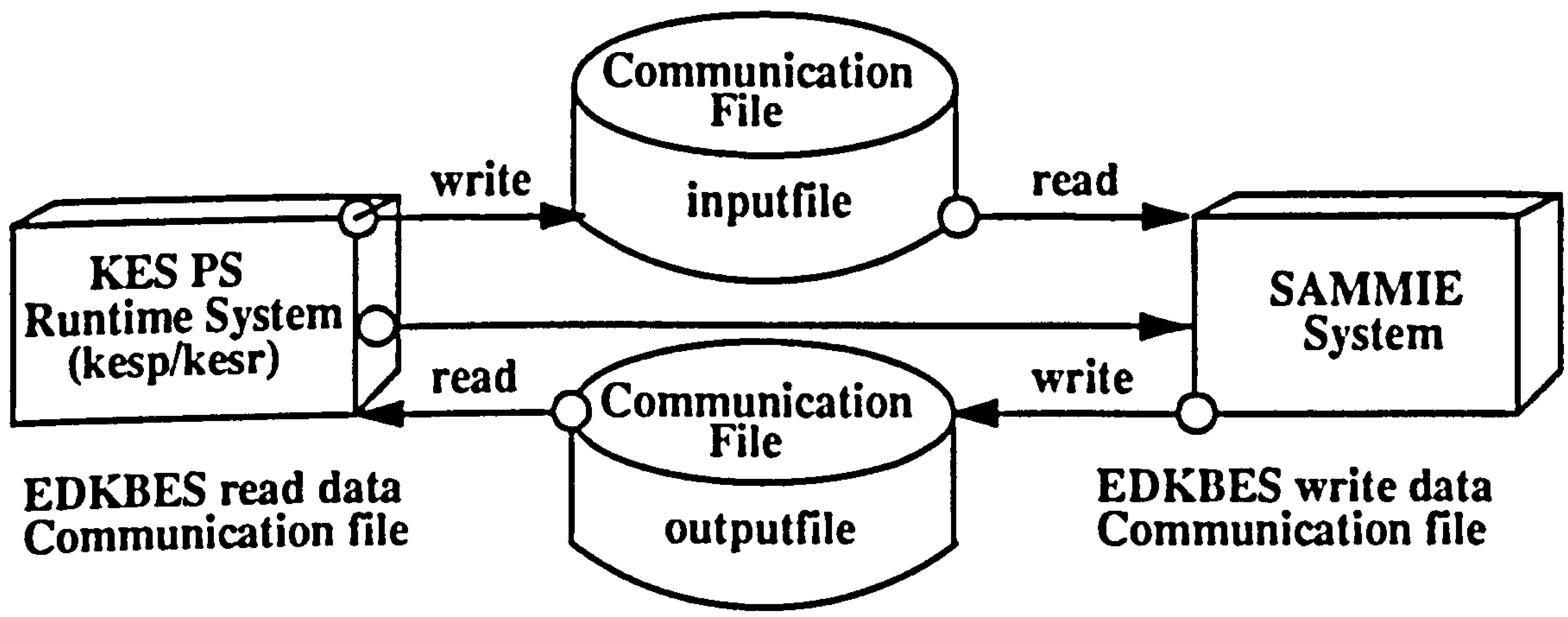


Figure 7.11 Use the "External" function to execute SAMMIE program read and write commands through a command file.

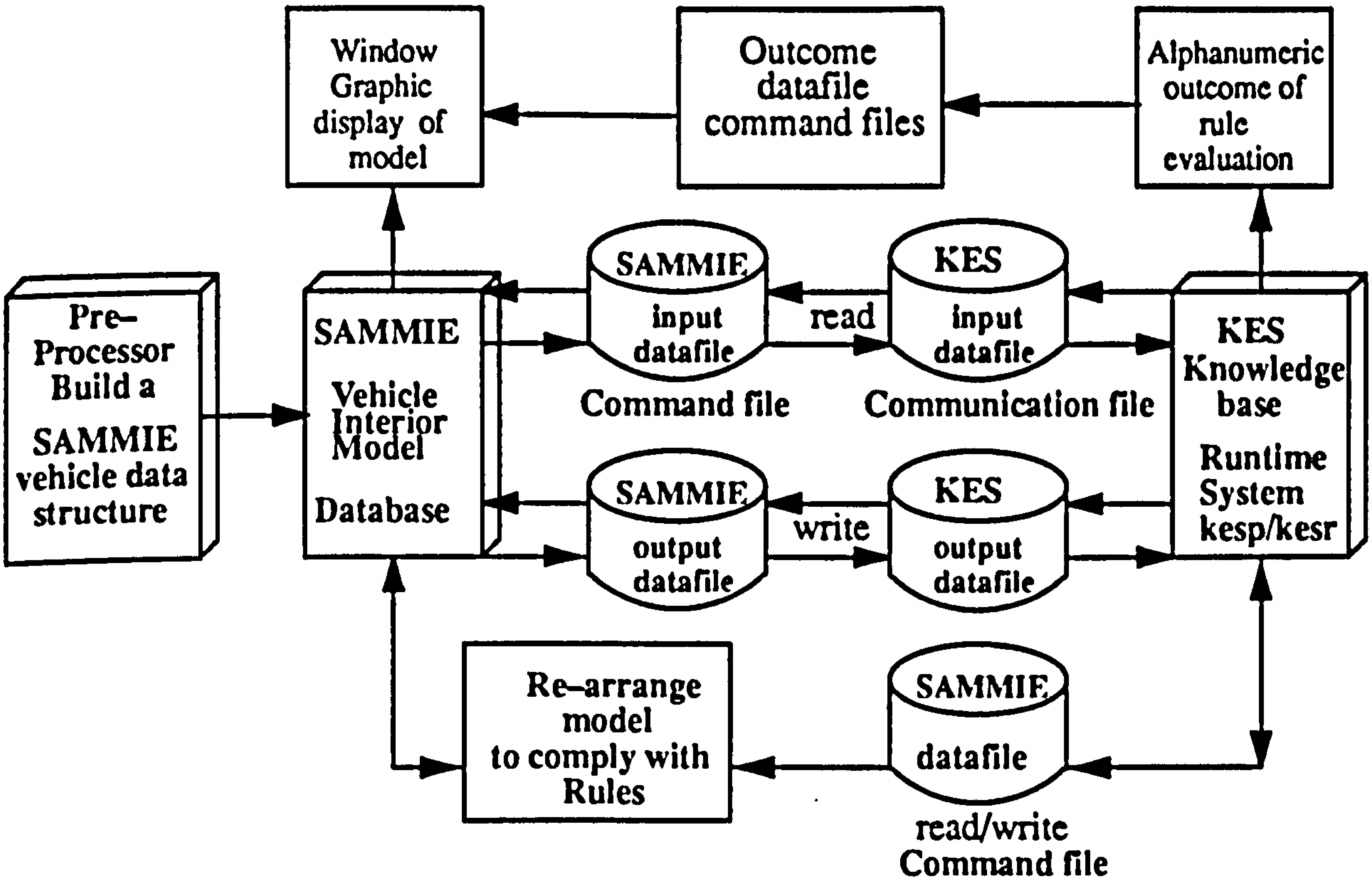


Figure 7.12 External programs – Integration of the EDKBES within the SAMMIE system

Thus, a convention was adopted that ensured that similar names were used in the SAMMIE command files and the expert system – EDKBES. The information in the command files needs to represent the class hierarchical form "of an attribute of an object or class". For example:–

The KES format of the read/write command is:–

read "file name", attribute or class,.........,attribute or class.
write "file name", attribute or class,.........,attribute or class.
(e.g. read "spfile" , speedo, speedo(cox, coy, coz)
(e.g. write "spfile1", speedo, speedo(cox, coy, coz)

In this example important coordinate information relating to the speedometer is transferred between SAMMIE and EDKBES.

Communication files are special files which include a series of assertions. The format used by this command is shown in figures 7.4 and 7.5.

The structure of the modular system is shown in figures 7.2 and 7.3 and includes:–

i) SAMMIE Database modules (parameters file: "interior car.src")

ii) SAMMIE Command files (read file:– "speedo.cmd" etc.)

iii) SAMMIE Command files (write file:– "speedofile1"etc.)

iv) EDKBES Knowledge Base module (program:– " dan.kb")

vi) EDKBES displays; speedometer, tachometer etc. module (program:– "dan.kb")

vii) EDKBES pedals; accelerator, brake and clutch module (program:– "dan.kb")

viii) KES Communication file "read file" (inputdata :– "speedofile" etc.)

viiii) KES Communication file "write file" (outputdata :– "speedofile1" etc.)

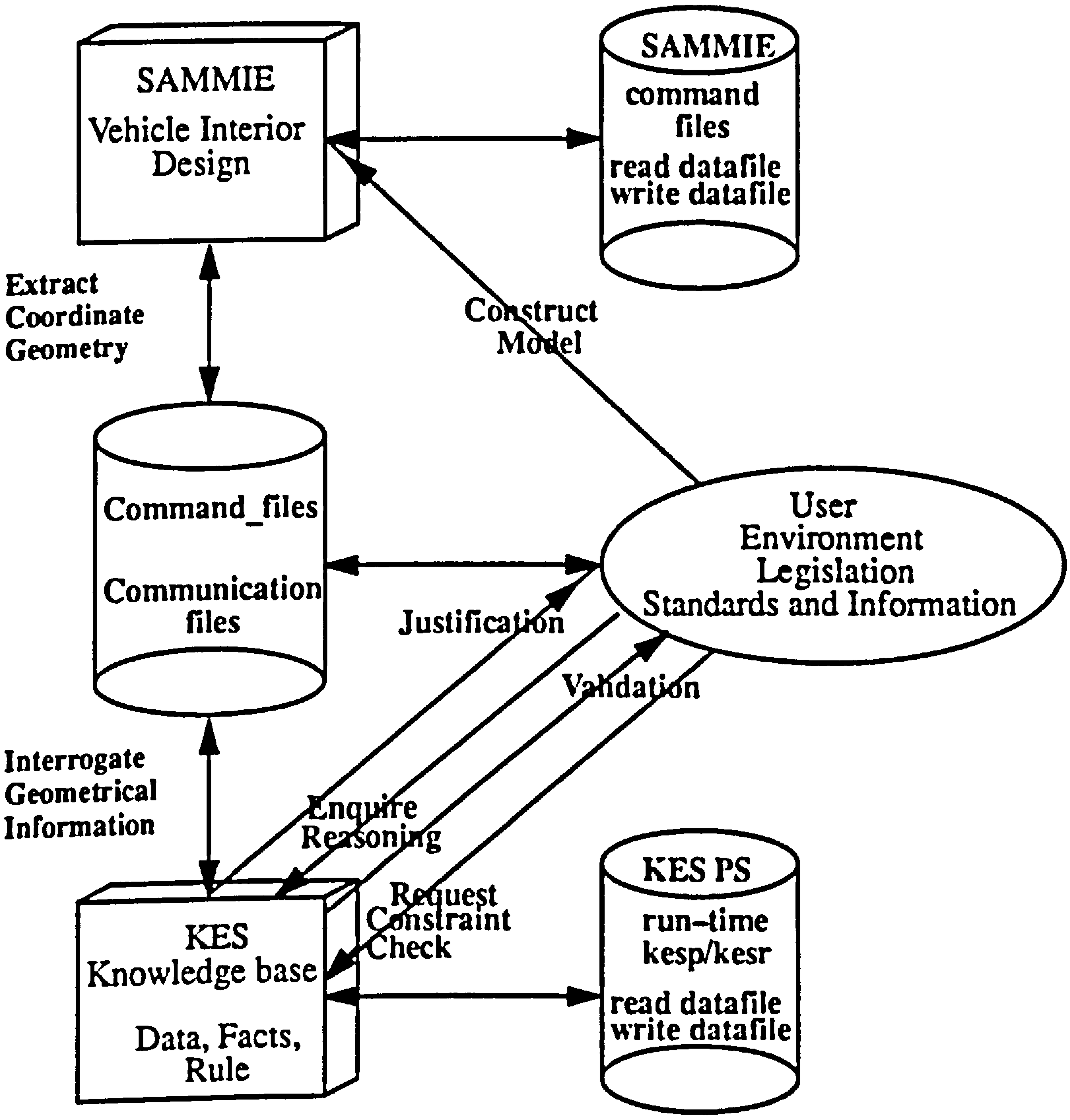


Figure 7.13 Structure of integration EDKBES within SAMMIE System for vehicle interior Design

7.9 Conclusions

This chapter has described the implementation of the Ergonomics Design Knowledge Base and the methods of integration. The interface capability is very important where the EDKBES is integrated with the SAMMIE system for evaluation or where "geometric reasoning", is carried out to determine whether it is functioning and the objectives have been achieved in relation to the hypotheses/findings (figures 7.11, 7.12 and 7.13). This testing and validation of this EDKBES knowledge base is considered in the next chapter.

CHAPTER 8

TESTING AND VALIDATION

8.1 Introduction

This chapter discusses the testing and validation of the knowledge base which is carried out to determine whether the integration between the Ergonomics Design Knowledge Base Expert System (EDKBES) and SAMMIE is functioning and to establish that the objectives have been achieved in making the standards and legislation, rules and regulations, and design working practice available to vehicle interior design.

8.2 Purpose of Testing and Validation

The testing and validation of the EDKBES expert system incorporated within SAMMIE has been divided into two areas:–

1) Firstly the software itself is tested and evaluated to determine whether the expert system is capable of producing the intended output or solutions.

2) Secondly, testing and validation of the integration is carried out to establish that it meets the requirements imposed upon it.

8.3 Testing and validation of EDKBES

In running the EDKBES in the windowed KES PS runtime system information concerning the user intentions is required – what he is going to do in the session. On startup the user has to load the previously parsed knowledge base. The initial dialogue is given below, and full details can be found in Appendix 3.

```
Knowledge Engineering System (KES), Release 3.0.
Copyright 1990, Software Architecture & Engineering, Inc.
Parsing the knowledge base 'dan.kb'.
Saving parsed knowledge base in 'dan.pkb'.
YES>kesr dan1.pkb

Knowledge Engineering System (KES), Release 3.0.
Copyright 1990, Software Architecture & Engineering, Inc.
Loading the knowledge base 'kesr dan.pkb'.
```

****

WELCOME TO THE INTERIOR OF A CAR DESIGN MONITOR EDKBES.

****

This Ergonomics Design Knowledge Base Expert System EDKBES knowledge base attempts to assist, the engineer and designer while designing for driver visibility through steering wheel, viewing with eye movement and no head movement, driver's seat adjustment for pedal locations and reach to accelerator, brake and clutch.

The EDKBES provides the necessary assistance in an area where there is a wealth of legislation. The resources available will be inferred in determining the choice of various entities for standards and legislation, rules and regulations, information on ergonomics standards and specifications, the design working practice to be related to the geometric reasoning aspects of the design process. The design is monitored against these standards and information in the window environment.

PROGRAMMER:-	**MD.DAN BIN MD. PALIL.**
FILE NAME:-	**dan.kb**
ORIGINAL DATE:-	**01 May 1993.**
PURPOSE:-	**Ergonomics Design – Integration EDKBES within SAMMIE system for vehicle interior Design running in SUN SPARC workstation on window.**

****

The Wealth of Legislation on Standard and Legislation(s).
The Rule and Regulation the Interior of a Car(s).
The Information on Ergonomics Design of Standards & Specification.
The Design Working Practice on Aspect of Design Process.

****

What would you like to do?
1. See a list of displays item developed in the knowledge base
2. See a list of some useful information for design
3. Design a particular interior display item
4. Check the design for styling
5. Check the type of instruments
6. Check the colour of instruments
7. Quit

As an example of the approach, item 3 could be selected if the objective were to evaluate visibility of displays through the steering wheel. On selecting the appropriate primary display the following menu appears.

8.3.1 Visibility of Displays – Speedometer

what would you like to start with?
It would be a good idea to go through each stage systematically.
1. Check visibility through the steering wheel
2. Check eye movement within the ergonomics range
3. Check that no head movement is required for visibility
4. Finished with this menu

If item 1 is selected then geometric information will be required from SAMMIE concerning the location of the steering wheel and the chosen display. This information is obtained by executing a SAMMIE command file to create a KES communication file containing entries such as those below:–

```
read "spfile" speedo, speedo(cox, coy, coz),
read "spfile", st_wheel, st_wheel(cox, coy, coz).
read "spfile", visible, visible(cox, coy, coz).

      obtain speedo:area>visibility.
if
      status (speedo:area>visibility) = known
then message
      " ",
      banner,
      combine (" for values of vertices given : "),
      combine (" top vertex tp = ", speedo:tp>coz),
      combine (" bottom vertex bp = ", speedo:bp>coz),
      combine (" right vertex rp = ", speedo:lp>cox),
      combine (" left vertex lp = ", speedo:rp>cox),
      combine (" the speedometer is ", speedo:area>visibility),
      " ",
      banner,
      " ".
endif.
```

In this instance the coordinates of the speedometer, steering wheel and driver's eyepoint are obtained from SAMMIE and used to update attributes in the knowledge base.

A command such as read "spfile", speedo, speedo(cox, coy, coz) instructs KES to access the communication file 'spfile', search for entries related to 'speedo' and extract numeric values for the coordinates cox, coy, coz.

The complete communication file relating to visibility through the steering wheel is shown below. Other, similar files relate to other aspects of the overall design problem.

```
assertclass displays = speedo, st_wheel, visible, view_with.
speedometer menu = Check visibility through the steering wheel.
speedometer menu = Check eye movement within the ergonomics range.
speedometer menu = Check that no head movement required for visibility.
assertclass displays = area, panel.
assertclass speedo = tp, bp, rp, lp, area.
assertclass st_wheel = tp, bp, rp, lp, cg.
assertclass visible = eye, head, panel.
assertclass view_with = eye move, head move.
assertclass speedo = cylinder, diameter tolerance length.
assertclass speedo = rectangle, width length height.
assertclass speedo = tp, an upper most point.
assertclass speedo = bp, a lowest point.
assertclass speedo = rp, a point on the far right side.
assertclass speedo = lp, a point on the far left side.
assertclass speedo = cg, an approximate centre of_gravity.
assertclass st_wheel = cylinder, diameter tolerance length.
```

assertclass st_wheel = rectangle, width length height.
assertclass st_wheel = tp, an upper most point.
assertclass st_wheel = bp, a lowest point.
assertclass st_wheel = rp, a point on the far right side.
assertclass st_wheel = lp, a point on the far left side.
assertclass st_wheel = cg, an approximate centre of_gravity.
assertclass visible = cylinder, diameter tolerance length.
assertclass visible = rectangle, width length height.
assertclass visible = tp, an upper most point.
assertclass visible = bp, a lowest point.
assertclass visible = rp, a point on the far right side.
assertclass visible = lp, a point on the far left side.
assertclass visible = cg, an approximate centre of_gravity.

speedo:tp>cox = 80.
speedo:tp>coy = 15.
speedo:tp>coz = 65.
speedo:bp>cox = 40.
speedo:bp>coy = 50.
speedo:bp>coz = 156.
speedo:lp>cox = 144.
speedo:lp>coy = 25.
speedo:lp>coz = 20.
speedo:rp>cox = 100.
speedo:rp>coy = 30.
speedo:rp>coz = 35.
speedo:area>cox = 150.
speedo:area>coy = 110.
speedo:area>coz = 120.
st_wheel:tp>cox = 125.
st_wheel:tp>coy = 110.
st_wheel:tp>coz = 120.
st_wheel:bp>cox = 105.
st_wheel:bp>coy = 120.
st_wheel:bp>coz = 130.
st_wheel:rp>cox = 125.
st_wheel:rp>coy = 120.
st_wheel:rp>coz = 105.
st_wheel:lp>cox = 125.
st_wheel:lp>coy = 150.
st_wheel:lp>coz = 140.
st_wheel:cg>cox = 125.
st_wheel:cg>coy = 130.
st_wheel:cg>coz = 120.
visible:eye>coz = 100.
visible:eye>coy = 115.
visible:eye>cox = 145.
visible:head>coz = 105.
visible:head>coy = 115.
visible:head>cox = 130.
visible:panel>coy = 125.
visible:panel>cox = 120.
visible:panel>coz = 110.

The result of executing this command is the display of important locations on the speedometer and an assessment of its visibility through the steering wheel.

for values of vertices given:-
top vertex tp = 165
bottom vertex bp = 65
right vertex rp = 144
left vertex lp = 100
the speedometer is not visible through st_wheel <1.00>

An opportunity for an explanation of the visibility problem is then given, and suggestions made for corrective action e.g..

the adjustment for speedometer position is to:-
move down at least by a min 5 mm
move up at least by a min -4 mm
move right at least by a min 4.7 mm
move left at least by a min -5 mm

The extent of recommended movement is determined by calculations embedded in the KES knowledge base as part of the attribute class and simply consists of a difference between coordinates of the speedometer (from SAMMIE) and limits defined in the KES knowledge base. The four rules (one for each limit of the rectangular placement zones) result in four recommendations, two of which are redundant. In the example above 'down by 5 mm' is equivalent to 'up by –4 mm' (within the limitations of the integer arithmetic used by KES).

8.3.1.1 Eye Movement

The objective of the eye movement evaluations is to determine if a particular display can be directly viewed by the driver using allowable eye movement only.

The command (issued by SAMMIE) to determine eye movement is in the form:–

```
read "spcfile", view_with, view_with(horizontal left,
      horizontal right, vertical upwards, vertical downwards).
```

This has the objective of determining extreme angles of eye movement in the horizontal and vertical planes.

For viewing the speedometer through the steering wheel the communication file will have the following format.

```
assertclass displays = speedo, st_wheel, visible, view_with.
speedometer menu = Check eye movement within the ergonomics range.
assertclass speedo = tp, bp, rp, lp, area.
assertclass st_wheel = tp, bp, rp, lp.
assertclass visible = eye, head, panel.
assertclass view_with = eye move, head move.
assertclass speedo = cylinder, tolerance length.
assertclass speedo = rectangle, width length height.
assertclass speedo = tp, an upper most point.
assertclass speedo = bp, a lowest point.
assertclass speedo = rp, a point on the far right side.
assertclass speedo = lp, a point on the far left side.
assertclass speedo = cg, an approximate centre of_gravity.
assertclass st_wheel = cylinder, diameter tolerance length.
assertclass st_wheel = rectangle, width length height.
assertclass st_wheel = tp, an upper most point.
assertclass st_wheel = bp, a lowest point.
assertclass st_wheel = rp, a point on the far right side.
```

```
assertclass st_wheel = lp, a point on the far left side.
assertclass st_wheel = cg, an approximate centre of_gravity.
assertclass visible = cylinder, diameter tolerance length.
assertclass visible = cylinder, diameter tolerance length.
assertclass visible = rectangle, width length height.
assertclass visible = tp, an upper most point.
assertclass visible = bp, a lowest point.
assertclass visible = rp, a point on the far right side.
assertclass visible = lp, a point on the far left side.
assertclass visible = cg, an centre of_gravity.
assertclass view_with = cylinder, diameter tolerance length.
assertclass view_with = rectangle, width length height.
assertclass view_with = tp, an upper most point.
assertclass view_with = bp, a lowest point.
assertclass view_with = rp, a point on the far right side.
assertclass view_with = lp, a point on the far left side.
assertclass view_with = cg, an approximate centre of_gravity.
view_with:eye move>horizontal left =        22.
 view_with:eye move>horizontal right =      20.
view_with:eye move>vertical upwards =       15.
view_with:eye move>vertical downwards =  18.
view_with:head move>horizontal left =       14.
view_with:head move>horizontal right =      11.
view_with:head move>vertical upwards =      14.
view_with:head move>vertical downwards = 11.
```

The extremes of eye movement are determined to be those below:–

```
for angles of eye movement given:–
horizontal left =           22
horizontal right =          20
vertical upwards =          15
vertical downwards =        18
these angles of eye movement are: outside gI eye limit <1.00>
```

These angles are found to be outside group I limits as indicated by the message.

The rule generating this information can be displayed as below:–

```
Name: group I eye move outside limit
Kind of entity: Production Rule
group I eye move outside limit:
v_w: view_with
     if
          view_with:eye move > horizontal left gt 20 or
          view_with:eye move > horizontal right lt -20 or
          view_with:eye move > vertical upwards gt 0 or
          view_with:eye move > vertical downwards lt -35
     then
          v_w>gI eye limit = outside gI eye limit.
     endif
```

The supporting knowledge associated with this rule can also be displayed.

{references: ADR}
{explanation: "AUSTRALIAN DESIGN RULE 18/00 (1988) ",
"All 'Visual Indicators' specified as Group I in Clause 18.2 shall",
"be totally located between 2 vertical planes inclined at 20 deg.",
"left and 20 deg. right of the longitudinal axis of the vehicle ",
"and passing through the far most points of the left and right ",
"95th Percentile Eye Ellipses respectively. Such indicators shall",
"be totally located above a plane inclined downwards at 35 deg. ",
"from the horizontal and including a horizontal transverse line ",
"through the for most points of each of the '95th Percentile Eye ",
"Ellipses' and below a plane tangential to the bottom of the 95th",
"Percentile Eye Ellipses which includes a line at ground level ",
"transverse to the longitudinal axis of the vehicle 11m forward ",
"of the rear most eye ellipse point. "}.

Solutions to the problem are presented in the form of actions to be taken to reduce eye movement.

You can adjust the position of the speedometer so that it lies
within the allowable ergonomics eye movement by:-
move to right to reduce angle of eye movement by: 2 deg
move down to reduce angle of eye movement by: 5 deg

8.3.1.2 Head Movement

Head movement for viewing displays is treated in a similar fashion to eye movement.

The KES command:-

read "spcfile", view_with, view_with (horizontal left,
horizontal right, vertical upwards, vertical downwards).

Gives head movement angles as below:-

for angles of head movement given:-
horizontal left = 14
horizontal right = 11
vertical upwards = 14
vertical downwards = 11
these angles of head movement are: outside gI head limit <1.00>

Again, the rule generating this information, the supporting knowledge and recommendations for change can be displayed.

Name: group I head move outside limit
Kind of entity: Production Rule
groupI head move outside limit:
v_w: view_with

```
if
    view_with:head move > horizontal left gt 0 or
    view_with:head move > horizontal right lt 0 or
    view_with:head move > vertical upwards gt 0 or
    view_with:head move > vertical downwards lt 0
then
    v_w>gI head limit = outside gI head limit.
endif
```

This rule reflects the requirement that displays should be viewable without head movement and the supporting knowledge associated for this can also be displayed.

```
{references: ISO}
   {explanation: "The display area of speedometer shall ",
        "    be visible  without head movement   ",
        "    Group I 'Visual Indicators' shall be  ",
        "    visible without head movement.     "}.
```

Solutions to the problem are presented in the form of actions to be taken to reduce head movement.

```
You can adjust the position of the speedometer so that it  lies within
the allowable ergonomics head movement by:-
move to left to reduce angle of head movement by:        4 deg
move to right to reduce angle of head movement by:      -21 deg
move down to reduce angle of head movement by:          4 deg
move up to reduce angle of head movement by:            -21 deg
```

8.3.2 Reachability of Pedals – Seat Adjustment

```
what would you like to check?
1. Check reachability through seat adjustment
2. Check that leg movement is within the ergonomics range
3. Check the seat movement required for reachability
4. Finished with this menu
```

If item 1 is selected then geometric information will be required from SAMMIE concerning the location of the seat and the chosen display. This information is obtained by executing a SAMMIE command file to create a KES communication file containing entries such as those below:–

```
read "acfile", accele, accele(cox, coy, coz).
read "acfile", seat_adjust, seat_adjust(cox, coy, coz).
read "acfile", reach, reach(cox, coy, coz).
     obtain accele:area>reachability.
if
```

```
            status (accele:area>reachability) = known
      then
            message " ", banner,
            combine (" for values of vertices given : "),
            combine (" top vertex tp = ", accele:tp>coz),
            combine (" bottom vertex bp = ", accele:bp>coz),
            combine (" right vertex rp = ", accele:lp>cox),
            combine (" left vertex lp = ", accele:rp>cox),
            combine (" the accelerator is ", accele:area>reachability),
      banner,
            " ".
endif.
```

In this instance the coordinates of the accelerator, seat and driver's heel point are obtained from SAMMIE and used to update attributes in the knowledge base.

A command such as read "acfile", accele, accele(cox, coy, coz) instructs KES to access the communication file 'acfile', search for entries related to 'accele' and extract numeric values for the coordinates cox, coy, coz.

The complete communication file relating to reachability with seat adjustment is shown below.

```
assertclass pedals = accele, seat_adjust, reach, dist_with.
accelerator menu = Check reachability through the seat adjustment.
accelerator menu = Check leg move within the ergonomics range.
accelerator menu = Check seat move required for reachable.
assertclass accele = tp, bp, rp, lp, area.
assertclass seat_adjust = tp, bp, rp, lp.
assertclass reach = leg, seat, pedals.
assertclass dist_with = leg move, seat move.
assertclass accele = cylinder, tolerance tolerance length.
assertclass accele = rectangle, width, length, height.
assertclass accele = tp, an upper most point.
assertclass accele = bp, a lowest point.
assertclass accele = rp, a point on the far right side.
assertclass accele = lp, a point on the far left side.
assertclass accele = cg, an approximate centre of_gravity.
assertclass seat_adjust = cylinder, diameter tolerance length.
assertclass seat_adjust = rectangle, width length height.
assertclass seat_adjust = tp, an upper most point.
assertclass seat_adjust = bp, a lowest point.
assertclass seat_adjust = rp, a point on the far right side.
assertclass seat_adjust = lp, a point on the far left side.
assertclass seat_adjust = cg, an approximate centre of_gravity.
assertclass reach = cylinder, diameter tolerance length.
assertclass reach = rectangle, width length height.
assertclass reach = tp, an upper most point.
assertclass reach = bp, a lowest point.
assertclass reach = rp, a point on the far right side.
assertclass reach = lp, a point on the far left side.
assertclass reach = cg, an approximate centre of_gravity.
```

accele:tp>cox =	80.	seat_adjust:bp>cox =	105.
accele:tp>coy =	15.	seat_adjust:bp>coy =	120.
accele:tp>coz =	65.	seat_adjust:bp>coz =	115.
accele:bp>cox =	40.	seat_adjust:rp>cox =	120.
accele:bp>coy =	30.	seat_adjust:rp>coy =	125.
accele:bp>coz =	150.	seat_adjust:rp>coz =	100.
accele:lp>cox =	140.	seat_adjust:lp>cox =	110.
accele:lp>coy =	125.	seat_adjust:lp>coy =	135.
accele:lp>coz =	120.	seat_adjust:lp>coz =	140.
accele:rp>cox =	100.	reach:leg>cox =	135.
accele:rp>coy =	130.	reach:leg>coy =	115.
accele:rp>coz =	120.	reach:leg>coz =	100.
accele:area>cox =	115.	reach:seat>cox =	150.
accele:area>coy =	110.	reach:seat>coy =	140.
accele:area>coz =	120.	reach:seat>coz =	130.
seat_adjust:tp>cox =	140	reach:pedals>cox =	120.
seat_adjust:tp>coy =	110.	reach:pedals>coy =	125.
seat_adjust:tp>coz =	120.	reach:pedals>coz =	110.

The result of executing this command is the determination of important locations on the accelerator and an assessment of its reachability through seat adjustment.

for values of vertices given:–
top vertex tp = 156
bottom vertex bp = 65
right vertex rp = 144
left vertex lp = 100
the accelerator is not reached through seat adjustment <1.00>

An opportunity for an explanation of the reachability problem is then given, and suggestions made for corrective actions e.g.

the adjustment for accelerator position is to:–
move down at least by a min 5 mm
move up at least by a min –4 mm
move right at least by a min 5 mm
move left at least by a min –5 mm

8.3.2.1 Leg Movement

The objective of the leg movement evaluations is to determine if a particular pedal can be directly reached by the driver using allowable leg movement only.

The command (issued by SAMMIE) to determine leg movement is in the form:–

```
read "acrfile", dist_with, dist_with(horizontal left,
      horizontal right, vertical upwards, vertical downwards).
```

This has the objective of determining extreme angles of leg movement in the horizontal and vertical planes.

For reaching the accelerator with seat adjustment the communication file will have the following format.

assertclass pedals = accele, seat_adjust, reach, dist_with.
accelerator menu = Check leg move within the ergonomics range.
accelerator menu = Check seat move required for reachable.
assertclass accele = tp, bp, rp, lp, area.
assertclass seat_adjust = tp, bp, rp, lp.
assertclass reach = leg, seat, panel.
assertclass dist_with = leg move, seat move.
assertclass accele = cylinder, diameter tolerance length.
assertclass accele = rectangle, width length height.
assertclass accele = tp, an upper most point.
assertclass accele = bp, a lowest point.
assertclass accele = rp, a point on the far right side.
assertclass accele = lp, a point on the far left side.
assertclass accele = cg, an approximate centre of_gravity.
assertclass seat_adjust = cylinder, diameter tolerance length.
assertclass seat_adjust = rectangle, width length height.
assertclass seat_adjust = tp, an upper most point.
assertclass seat_adjust = bp, a lowest point.
assertclass seat_adjust = rp, a point on the far right side.
assertclass seat_adjust = lp, a point on the far left side.
assertclass seat_adjust = cg, an approximate centre of_gravity.
assertclass reach = cylinder, diameter tolerance length.
assertclass reach = rectangle, width length height.
assertclass reach = tp, an upper most point.
assertclass reach = bp, a lowest point.
assertclass reach = rp, a point on the far right side.
assertclass reach = lp, a point on the far left side.
assertclass reach = cg, an approximate centre of_gravity.
assertclass dist_with = cylinder, diameter tolerance length.
assertclass dist_with = rectangle, width length height.
assertclass dist_with = tp, an upper most point.
assertclass dist_with = bp, a lowest point.
assertclass dist_with = rp, a point on the far right side.
assertclass dist_with = lp, a point on the far left side.
assertclass dist_with = cg, an approximate centre of_gravity.
dist_with:leg move > horizontal left = 21.
dist_with:leg move > horizontal right = 25.
dist_with:leg move > vertical upwards = 9.
dist_with:leg move > vertical downwards = 7.
dist_with:seat move > horizontal left = 14.
dist_with:seat move > horizontal right = 11.
dist_with:seat move > vertical upwards = 13.
dist_with:seat move > vertical downwards = 16.

The extremes of leg movement are determined as being those below:–

for angles of leg movement given:–
horizontal left = 22
horizontal right = 20
vertical upwards = 15
vertical downwards = 18
these angles of leg movement are: outside gl leg limit <1.00>

These angles are found to be outside group I limits as indicated by the message.

The production rule generating this information can be displayed as below:–

```
Name: group I leg move outside limit
Kind of entity: Production Rule
group I leg move outside limit:
d_w: dist_with
    if
        dist_with:leg move > horizontal left gt 20 or
        dist_with:leg move > horizontal right lt -20 or
        dist_with:leg move > vertical upwards gt 0 or
        dist_with:leg move > vertical downwards lt -35
    then
        d_w>gI leg limit = outside gI leg limit.
    endif
```

The supporting knowledge associated with this production rule can also be displayed.

```
{references: ISO}
{explanation: "ROAD VEHICLES PASSENGER CARS DRIVER HAND ",
"CONTROL REACH, ISO 3958-77. Leg_reach H_point of the pedals.   ",
"The following location for leg_reach H-point of_pedal:        ",
"   -   ACCELERATOR PEDAL   -        ",
"The seating reference point is terms of the R-point.    ",
"The pivot centre of the torso line and thigh centreline    ",
"of the two or three-dimensional H-point machine template with ",
"95% leg length used to describe vehicle seating geometry.  ",
"The accelerator heel-point or H-point device with floor    ",
"covering; the foot angle of the device is restricted to not  ",
"less than 87 deg. Horizontal dimensional from the R-point   ",
"to the driver H-point (Hx). The vertical dimensions from    ",
"R-point to driver H-point (Hz).          ",
"   -   BRAKE PEDAL   -        ",
"The brake heel-point or H-point device height 3-10 inches   ",
"max., 2 inches max. travel with 15 lbs.max. force brake.   ",
"The distance between other pedal 2 inches min. gap.     ",
"   -   CLUTCH PEDAL   -        ",
"The clutch heel-point or H-point device height 3-10 inches   ",
"max., 4 inches max. travel with 80-90 lbs.max. force brake.  ",
"The distance between other pedal 2 inches min. gap.     "}.
```

Solutions to the problem are presented in the form of actions to be taken to reduce leg movement.

```
You can adjust the position of the accelerator so that it lies within
the allowable ergonomics leg movement by:–
move to right to reduce angle of leg movement by:      2 deg
move down to reduce angle of leg movement by:          5 deg
```

8.3.2.2 Seat Adjustment

Seat adjustment for reachability of pedals is treated in a similar fashion to leg movement.

The KES command:-

```
read "acrfile", dist_with, dist_with (horizontal left,
          horizontal right, vertical upwards, vertical downwards).
```

Gives seat adjustment angles as below:-

```
for angles of seat adjustment given:-
horizontal left =           14
horizontal right =          11
vertical upwards =          14
vertical downwards =        11
these angles of seat movement are: outside gI seat limit <1.00>
```

Again, the production rule generating this information, the supporting knowledge and recommendations for change can be displayed.

```
Name: group I seat move outside limit
Kind of entity: Production Rule
group I seat move outside limit:
d_w: dist_with       if
          dist_with:seat move > horizontal left gt 0 or
          dist_with:seat move > horizontal right gt 0 or
          dist_with:seat move > vertical upwards gt 0 or
          dist_with:seat move > vertical downwards gt 0
     then
          d_w>gI seat limit = outside gI seat limit.
     endif
```

The supporting knowledge associated with this production rule can also be displayed.

```
{references: ISO}
{explanation:    "The identification and those parts of   ",
          "the pedals area required to indicate    ",
          "a critical condition shall be leg_reach ",
          "without seat movement.                  ",
          "The remaining parts of the pedals       ",
          "shall also be 'reach'; for these,       ",
          "seat movement is permitted.             "}.
```

Solutions to the problem are presented in the form of actions to be taken to reduce seat adjustment.

You can adjust the position of the accelerator so that it
lies within the allowable ergonomics seat adjustment by:-

move to left to reduce angle of seat adjustment by:	4 deg
move to right to reduce angle of seat adjustment by:	-21 deg
move down to reduce angle of seat adjustment by:	4 deg
move up to reduce angle of seat adjustment by:	-21 deg

Running the EDKBES, as described above, results in various recommendations for design changes together with the values of geometric parameters to achieve this. Typical information relates to visibility of the speedometer through steering wheel, corrective eye and head movement, the reachability of pedals and corrective leg movements and seat adjustment.

The next section describes how this information is used to implement the recommendations within SAMMIE.

8.3.3 Measurement Distance

8.3.3.1 Primary Display

The measurement of distances between displays is an example of geometric information acquisition from SAMMIE and a typical situation is shown in figure 8.1. The following SAMMIE command file is used to obtain the necessary information.

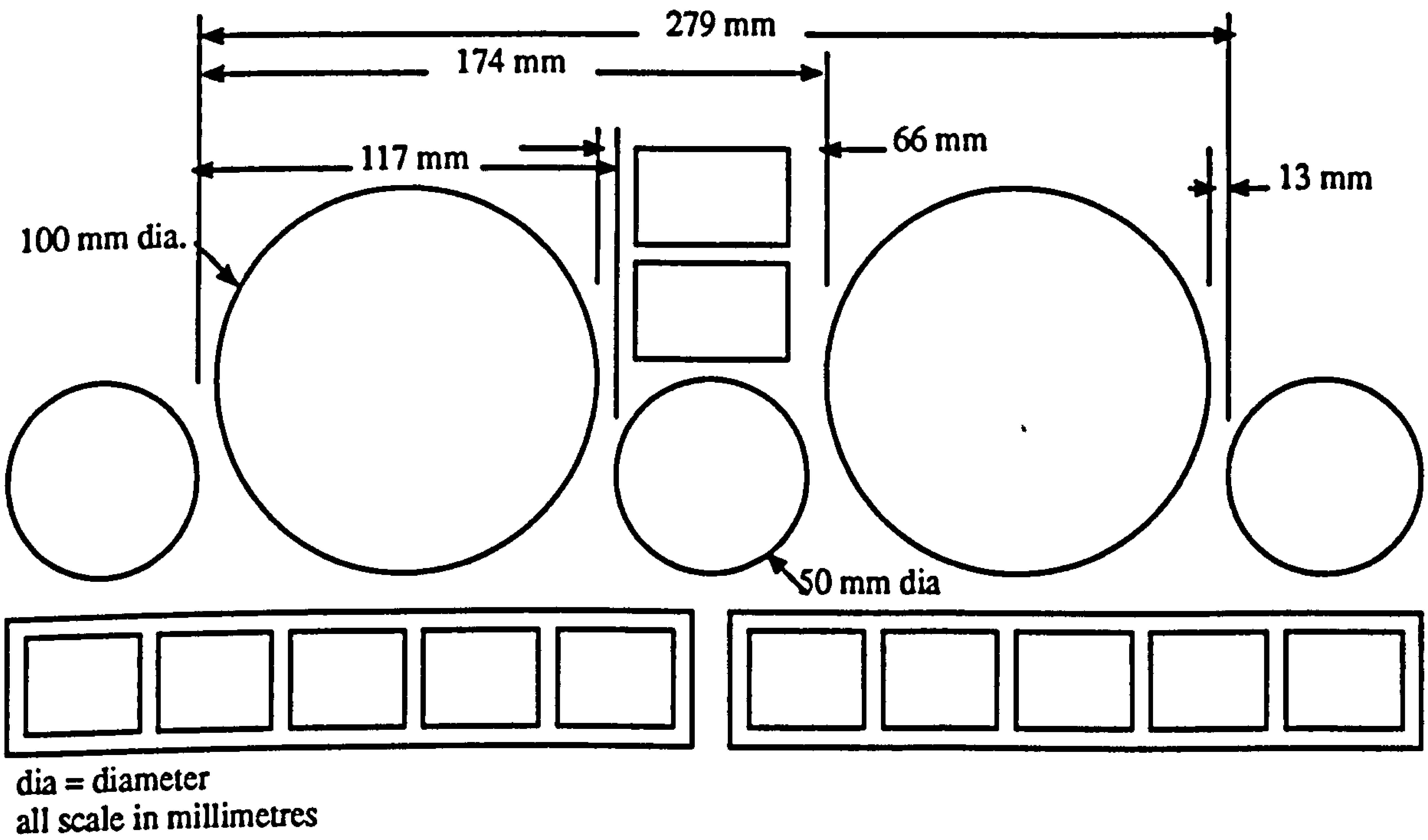


Figure 8.1 Primary displays -To find the measured distances between speedometer, tachometer, etc.

```
&rem Command file to find the distance of two objects from p to t = d and enter
&rem contents into an output file which will be read by KES
&character $f $t $s $d $e $r $l
&real $adistf $adistt $adists $adistd $adiste $adistr $adistl
$f = fuel
$t = tachometer
$s = speedometer
$d = door-open
$e = temperature
$r = right-indicator
$l = left-indicator
&print At 1
&execute %measure object
&execute %measure to_object $t $s
&rvalue measured_distance $adistt
&execute %measure object
&execute %measure to_object $d $t
&rvalue measured_distance $adistd
&execute %measure object
&execute %measure to_object $e $s
&rvalue measured_distance $adiste
&execute %measure object
&execute %measure to_object $f $e
&rvalue measured_distance $adistf
&execute %measure object
&execute %measure to_object $s $f
&rvalue measured_distance $adists
&execute %measure object
&execute %measure to_object $t $e
&rvalue measured_distance $adistt
&execute %measure object
&execute %measure to_object $e $d
&rvalue measured_distance $adiste
&execute %measure object
&execute %measure to_object $s $d
&rvalue measured_distance $adists
&execute %measure object
&execute %measure to_object $f $t
&rvalue measured_distance $adistf
&execute %measure object
&execute %measure to_object $d $f
&rvalue measured_distance $adistd
&execute %measure object
&execute %measure to_object $r $s
&rvalue measured_distance $adistr
&execute %measure object
&execute %measure to_object $l $t
&rvalue measured_distance $adistl
&open #1 spdofile1
&rem Write to file – first the location for test results
&print #1 vstatus = evaluated.
&print #1 tachometer to speedometer = $adistt .
&print #1 door to tachometer = $adistd .
&print #1 engine to speedometer = $adiste .
&print #1 fuel to engine = $adistf .
&print #1 speedometer to fuel = $adists .
```

```
&print #1 tachometer to engine = $adistt .
&print #1 engine to door-open = $adiste .
&print #1 speedometer to door = $adists .
&print #1 fuel to tachometer = $adistf .
&print #1 door-open to fuel = $adistd .
&print #1 right-indicator to speedo = $adistr .
&print #1 left-indicator to tachometer = $adistl .
&rem Now print terminator
&print #1 %
&close #1
&print Processing terminated (md)
&return
```

The result of executing this command file in SAMMIE will be the generation of a formatted data file that contains the geometric information required by various rules in the knowledge base. Typical contents of the file could be:–

```
vstatus = evaluated.
tachometer to speedometer =      65.4 .
door to tachometer =             13.9 .
temperature to speedometer =     13.7 .
fuel to temperature =           279.3 .
speedometer to fuel =           174.8 .
tachometer to temperature =     174.5 .
temperature to door-open =      117.4 .
speedometer to door =            12.9 .
fuel to tachometer =             12.4 .
door-open to fuel =             117.6 .
right-indicator to speedo =      59.4 .
left-indicator to tachometer =   59.1 .
```

'vstatus' indicates whether or not visibility of the displays has been evaluated by the expert system. All distances are measured in millimetres and represent the shortest lateral distance between displays. "door–open", "temperature" and "fuel" refer to the door–open indicator, temperature gauge and fuel gauge respectively.

8.3.3.2 Pedals

An example of geometric information required from SAMMIE concerning the measurement distances of pedals (figure 8.2) is:–

```
&rem Command file to find the distance of two objects from a to c = d and enter
&rem contents into an output file which will be read by KES
&character $a $b $c $d $r $p $h $i $n $o $f
&real $adista $adistb $adistc $adistd $adistr $adistp
&real $adisth $adisti $adistn $adisto $adistf
$a = ACCLTOR
$b = BRKE
$c = CLUTCH
```

```
$d = DRIVER-SEAT
$r = REAR-SEAT
$p = dashfascia
$h = OPERATOR
$i = interior-mirror
$n = nearside-mirror
$o = offside-mirror
$f = floor
&print At 1
&execute %measure object
&execute %measure to_object $a $c
&rvalue measured_distance $adista
&execute %measure object
&execute %measure to_object $b $a
&rvalue measured_distance $adistb
&execute %measure object
&execute %measure to_object $c $b
&rvalue measured_distance $adistc
&execute %measure object
&execute %measure to_object $r $a
&rvalue measured_distance $adistr
&execute %measure object
&execute %measure to_object $i $h
&rvalue measured_distance $adisti
&execute %measure object
&execute %measure to_object $n $h
&rvalue measured_distance $adistn
&execute %measure object
&execute %measure to_object $o $h
&rvalue measured_distance $adisto
&execute %measure object
&execute %measure to_object $p $h
&rvalue measured_distance $adistp
&execute %measure object
&execute %measure to_object $d $a
&rvalue measured_distance $adistd
&execute %measure object
&execute %measure to_object $r $d
&rvalue measured_distance $adistr
&execute %measure object
&execute %measure to_object $f $d
&rvalue measured_distance $adistf

&open #1 seat3file

&rem Write to file – first the location for test results
&print #1 dstatus = non_evaluated.
&print #1 ACCLTOR to CLUTCH = $adista .
&print #1 BRKE to ACCLTOR = $adistb .
&print #1 CLUTCH to BRKE = $adistc .
&print #1 REAR-SEAT to ACCLTOR = $adistr .
&print #1 interior-mirror to OPERATOR = $adisti .
&print #1 nearside-mirror to OPERATOR = $adistn .
&print #1 offside-mirror to OPERATOR = $adisto .
&print #1 dashfascia to OPERATOR = $adistp .
&print #1 DRIVER-SEAT to ACCLTOR = $adistd .
&print #1 REAR-SEAT to DRIVER-SEAT = $adistr .
&print #1 floor to DRIVER-SEAT = $adistf .
&rem Now print terminator
&print #1 %
&rem Close the file
```

&close #1
&print Processing terminated (md)
&return

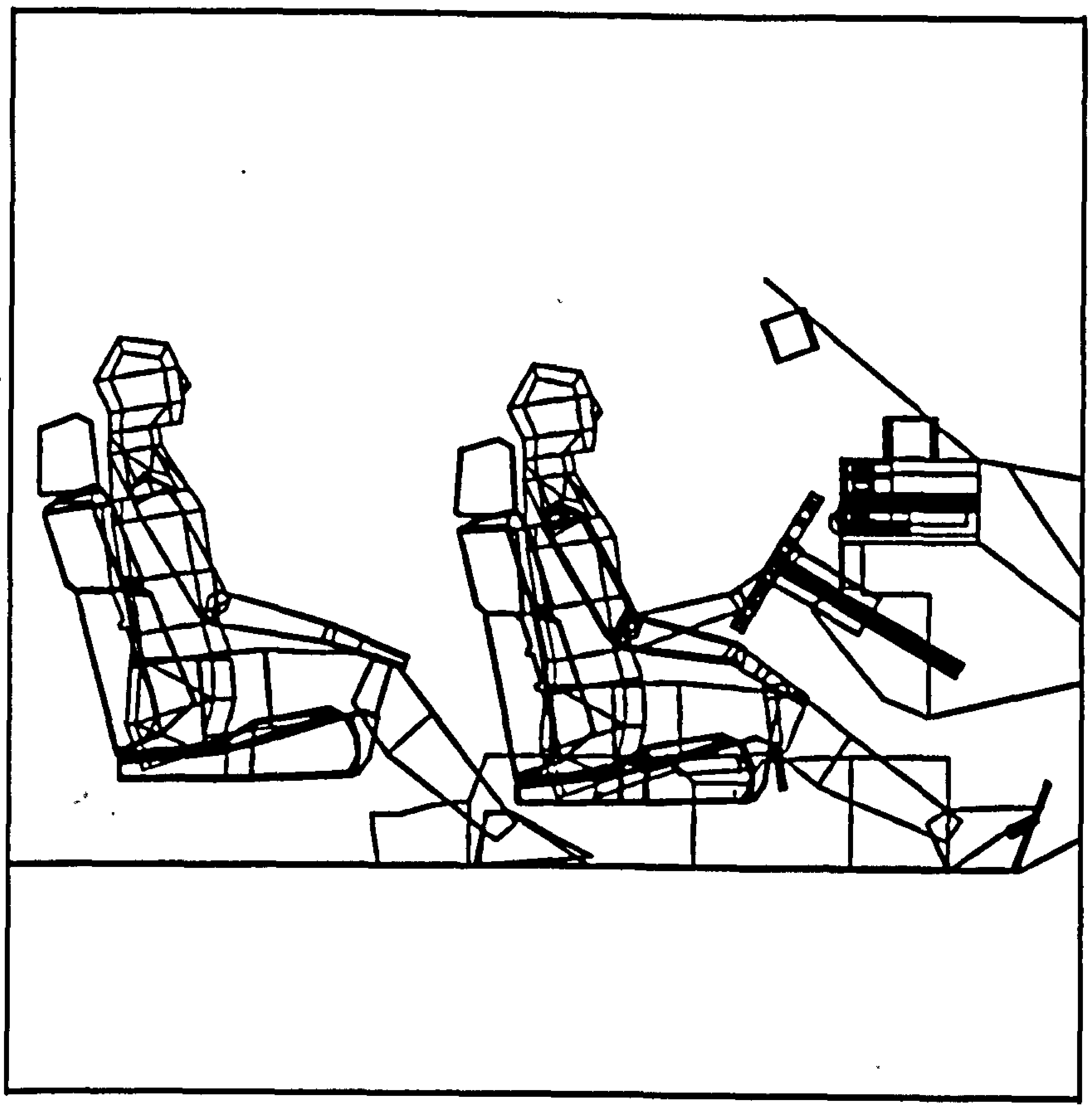


Figure 8.2 To find the measured distances between seats and pedal controls.

The result of executing this command file in SAMMIE will be the generation of a formatted data file that contains the geometric information required by various rules in the knowledge base. Typical contents of the file could be:–

dstatus = evaluated.	
ACCLTOR to CLUTCH =	136.5 .
BRKE to ACCLTOR =	35.0 .
CLUTCH to BRKE =	31.6 .
REAR-SEAT to ACCLTOR =	1918.6 .
interior-mirror to OPERATOR =	774.1 .
nearside-mirror to OPERATOR =	896.0 .
offside-mirror to OPERATOR =	1367.2 .
dashfascia to OPERATOR =	746.6.
DRIVER-SEAT to ACCLTOR =	1038.5 .
REAR-SEAT to DRIVER-SEAT =	746.9 .
floor to DRIVER-SEAT =	295.4 .

"dstatus indicates whether or not the subsequent distance value has been evaluated by the expert system. Distances are in millimetres and represent the shortest lateral distance between pedals. Distances between the OPERATOR and mirror refer to the line of sight distance from the driver's eyepoint to the centre of the mirror.

8.4 Implementation and Recommendations

The geometric information obtained from the SAMMIE model as described in previous sections is used to provide data for the attributes in the knowledge base. The design is then evaluated against the knowledge base and various recommendations made.

This section describes how these recommendations may be implemented by changes to the SAMMIE system under the control of SAMMIE command files. Examples are given here that demonstrate the success in this integration of EDKBES with SAMMIE system.

8.4.1 Speedometer Location for Visibility

Evaluation of the visibility of the speedometer through the steering wheel (figure 8.3) uses geometric information obtained from SAMMIE in the manner described in sections 8.3.1 to 8.3.3. The recommendations from the rules described in these earlier sections can be used to implement changes in the model using the command file below, or by direct manipulation of SAMMIE, and result in model changes depicted in figure 8.3.

Integration EDKBES with SAMMIE for Vehicle Interior Design – Speedometer Location for Visibility

```
$owner = workplace
&character $d $r
$s = speedometer
&execute %workplace global
&execute %workplace z negative
&execute %workplace shift speedo 5
&execute %workplace global
&execute %workplace z positive
&execute %workplace shift speedo
&execute %workplace local
&execute %workplace x positive
&execute %workplace shift speedo 4.7
&execute %workplace local
&execute %workplace x negative
&execute %workplace shift speedo
&rem
&return
```

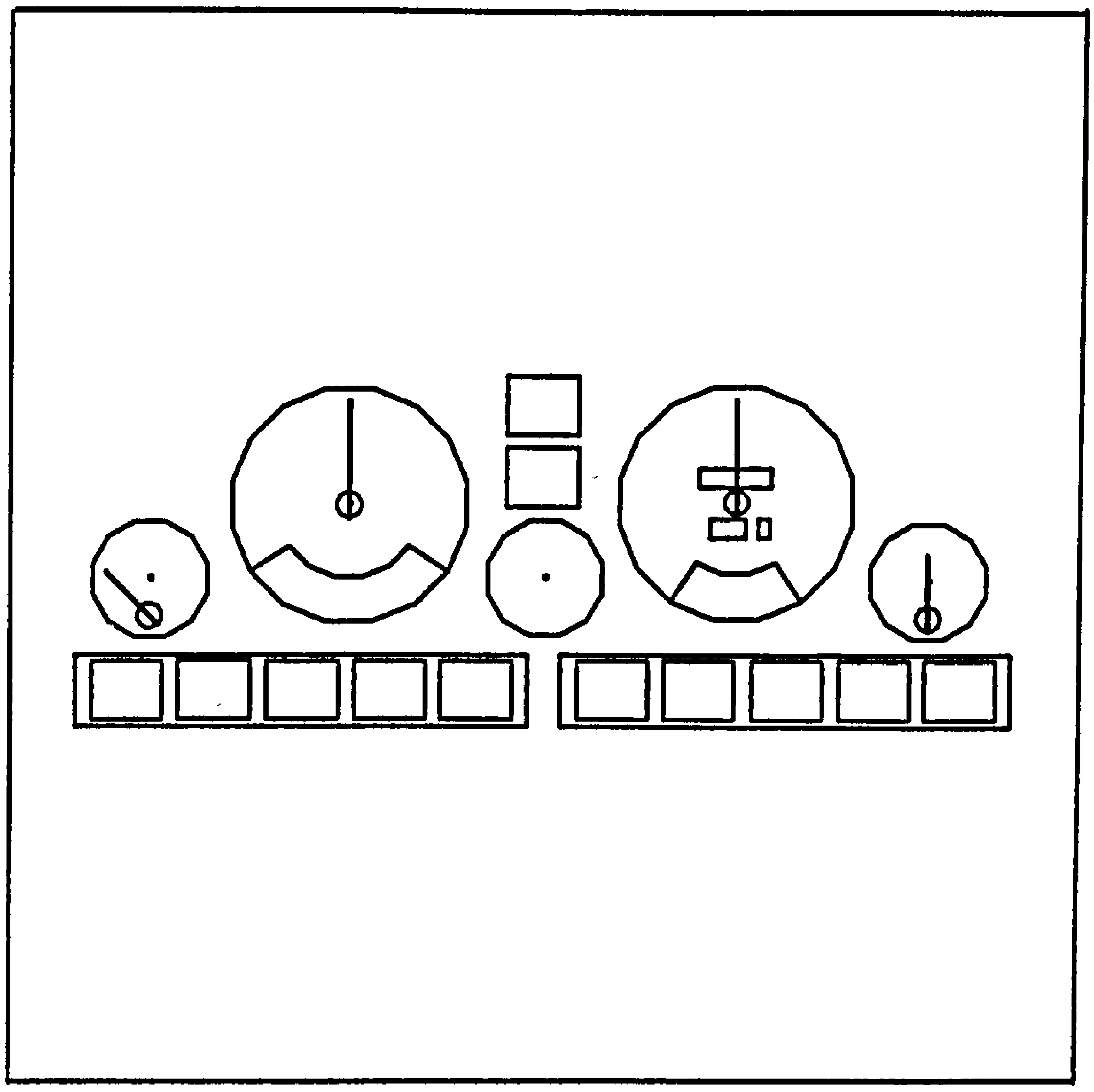


Figure 8.3 The result of driver's visibility for speedometer adjustment.

8.4.2 Driver's View

The second example demonstrates the adjustment of the viewing angle of the display panel, for driver's visibility using either eye or head movement. (figures 8.4 and 8.5).

Integration EDKBES with SAMMIE for Vehicle Interior Design – Driver's View

```
$owner = workplace
&execute %mansview
&execute %mansview mean head
&execute %mansview set_increment
&execute %mansview to right

&execute %mansview
&execute %mansview mean head
&execute %mansview set_increment 4
&execute %mansview to left

&execute %mansview
&execute %mansview mean head
&execute %mansview set_increment
&execute %mansview upward

&execute %mansview
&execute %mansview mean head
&execute %mansview set_increment 4
&execute %mansview downwards

&rem
&return
```

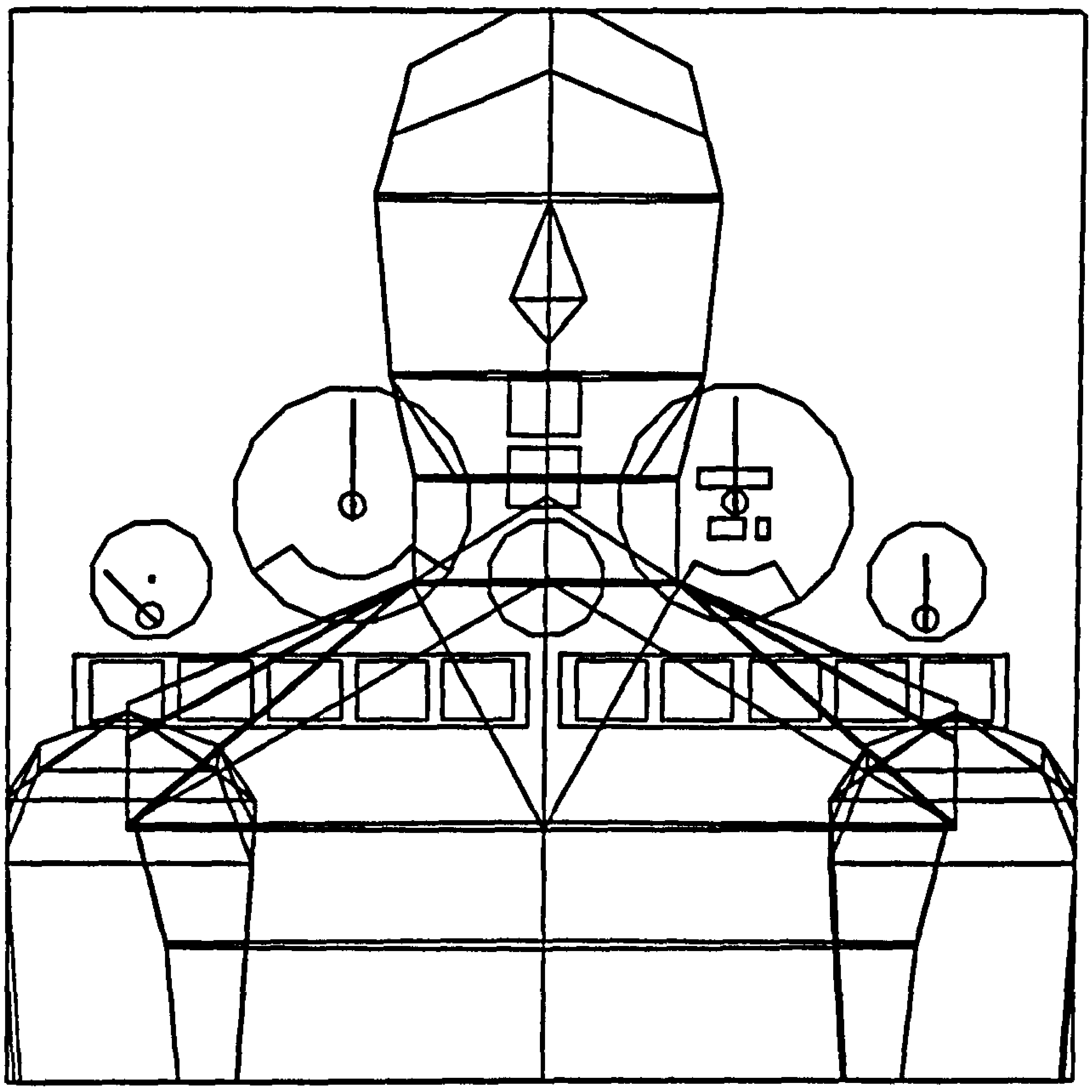


Figure 8.4 Driver's visibility through eye movement.

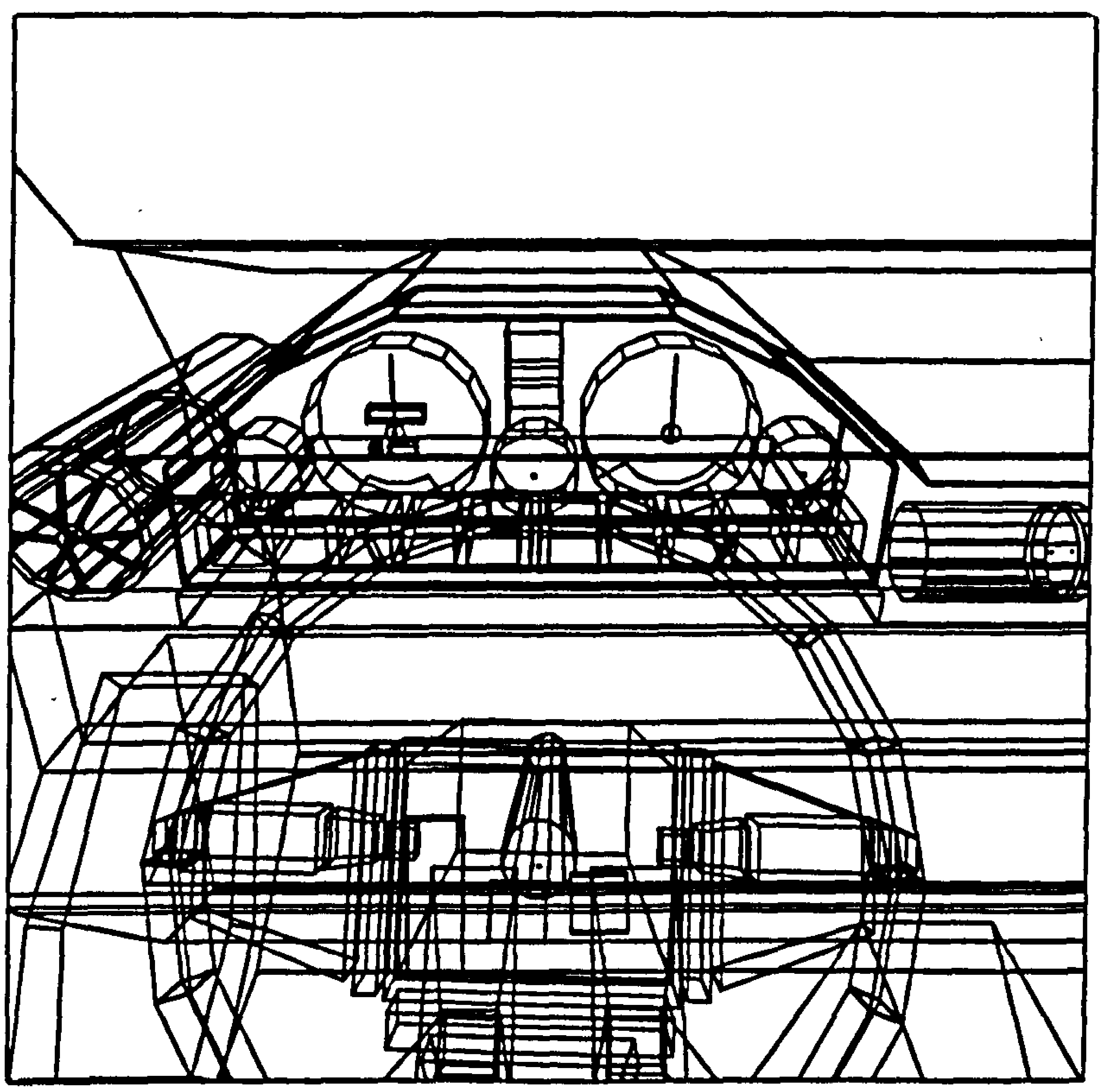


Figure 8.5 Driver's visibility through correcting eye movement

8.4.3 Driver's Seat Adjustment

The third example demonstrates the reachability of pedals using leg movement and driver's seat adjustment, and the situation is illustrated in figures 8.6 and 8.7.

Integration EDKBES with SAMMIE for Vehicle Interior Design – Driver's Seat Adjustment

```
$owner = workplace
&character $d $r
$d = DRIVER-SEAT

&execute %workplace global
&execute %workplace z positive
&execute %workplace shift DRIVER-SEAT -21

&execute %workplace global
&execute %workplace x positive
&execute %workplace rotate DRIVER-SEAT -4

&rem
&return
```

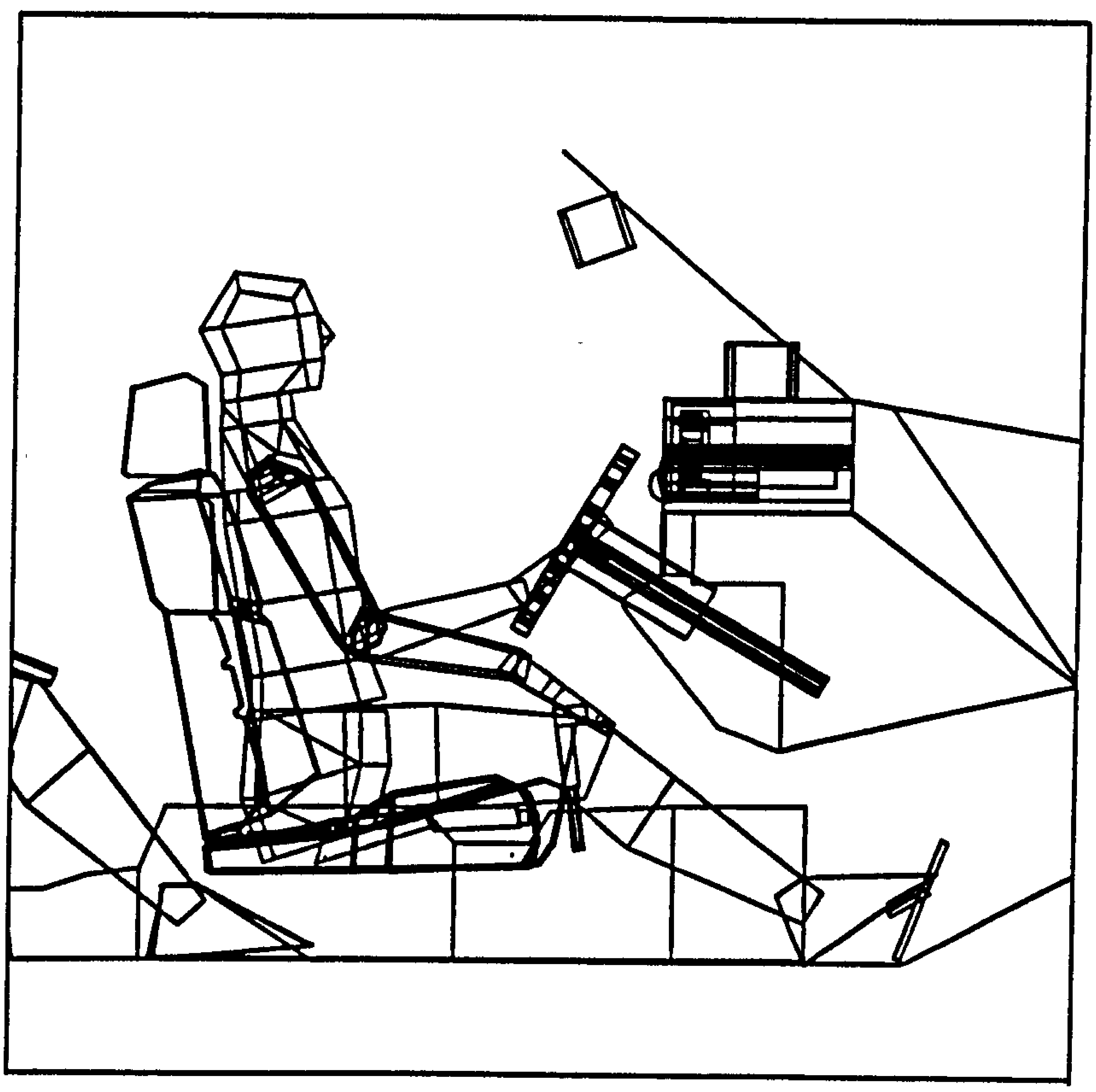


Figure 8.6 Reachability through the driver's seat adjustment

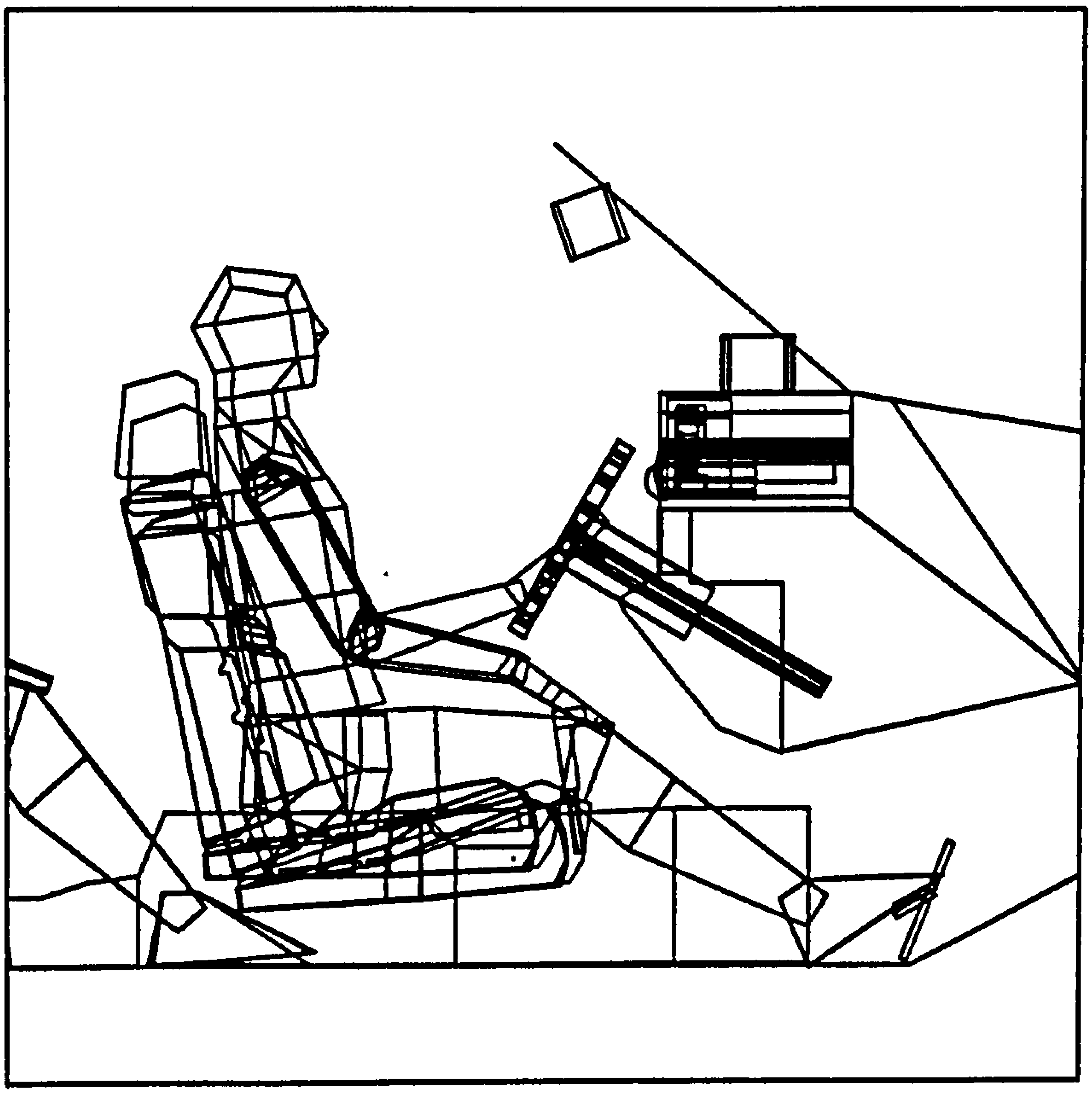


Figure 8.7 The result of reachability through the driver's seat adjustment

8.4.4 Driver's Seat and Rear Passenger Adjustment

The fourth example demonstrates the reachability of pedals using leg movement and driver's seat adjustment, and rear passenger seat adjustment. The situation is illustrated in figures 8.8 and 8.9.

```
$owner = workplace
&character $d $r
$d = DRIVER-SEAT
$r = REAR-SEAT
&execute %workplace global
&execute %workplace z positive
&execute %workplace shift DRIVER-SEAT -21
&execute %workplace global
&execute %workplace x positive
&execute %workplace rotate DRIVER-SEAT -4
&execute %workplace global
&execute %workplace y positive
&execute %workplace shift REAR-SEAT -21
&execute %workplace global
&execute %workplace x positive
&execute %workplace rotate REAR-SEAT -4
&rem
&return
```

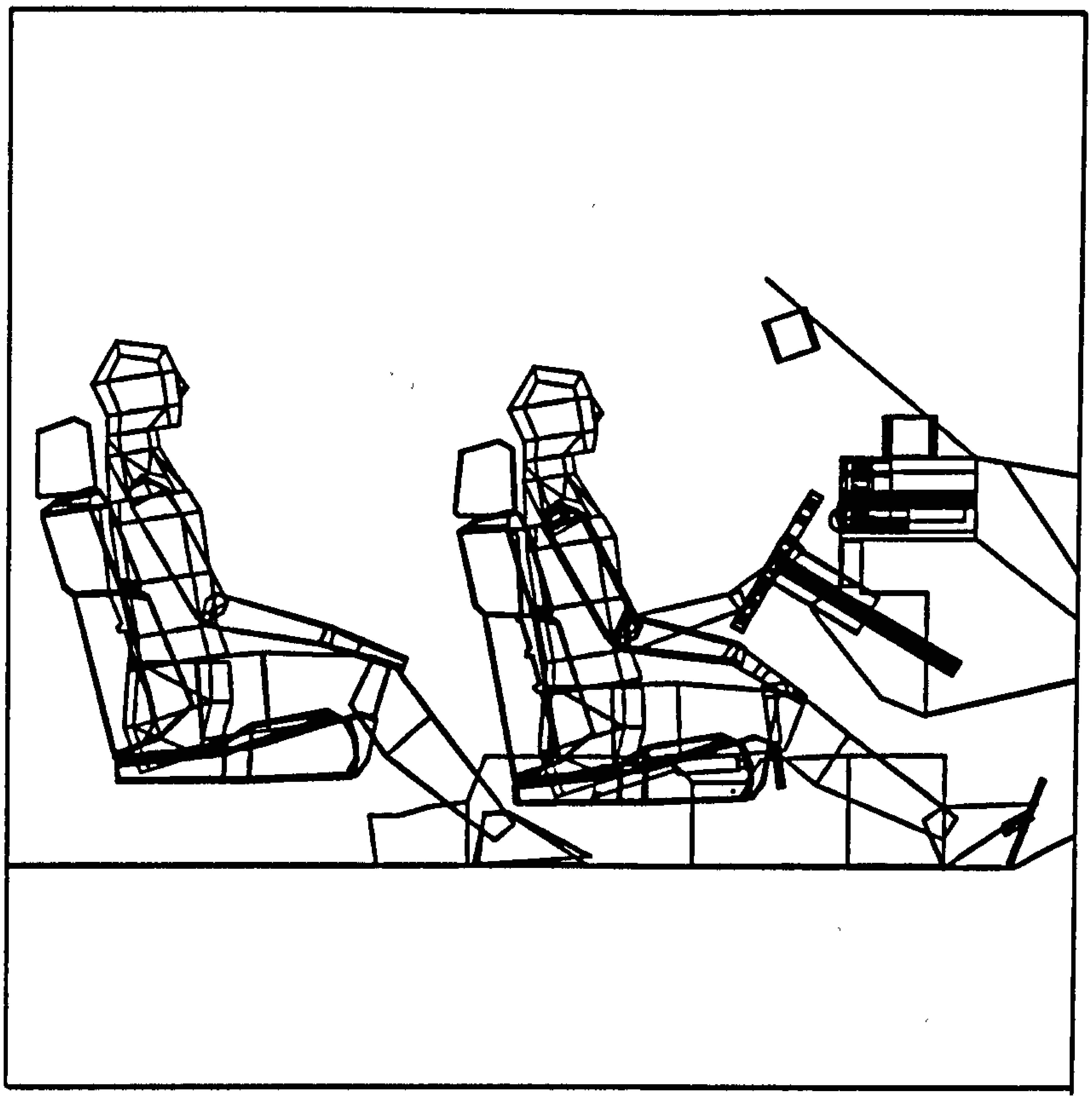


Figure 8.8 The integration of driver's seat and passenger seat.

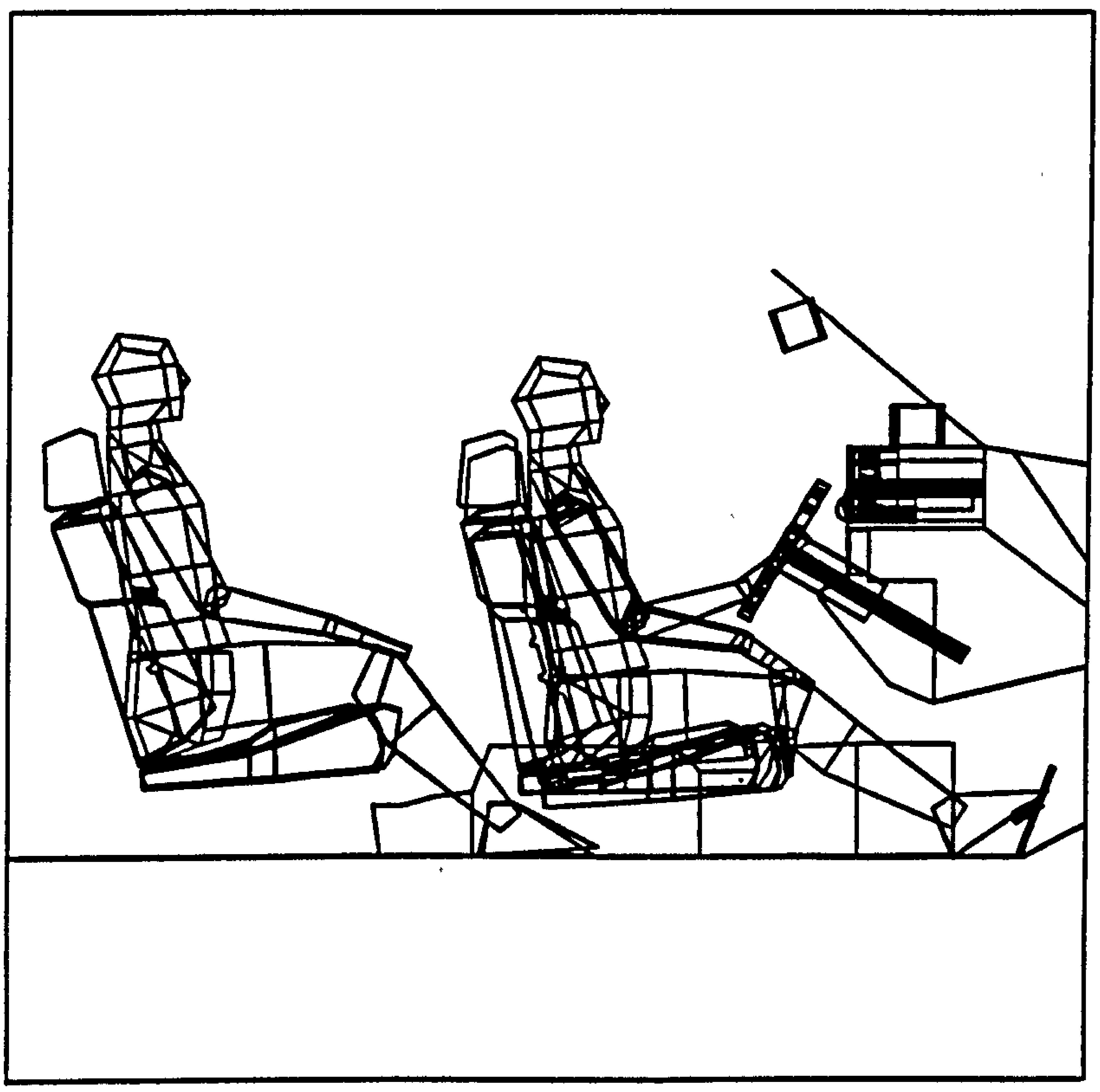


Figure 8.9 Result of reachability through seat adjustment for driver's seat and rear passenger seat

8.4 Conclusions

The testing and validation carried out had the two objectives (a) to determine if the expert system functioned in the expected manner when supplied with appropriate information and (b) to determine the level of success achieved in integrating the expert system with SAMMIE.

The first objective was approached by extensive testing of each component of the expert system with a variety of parameters. Examples of the outcome of this have been given in this chapter with a more extensive set of examples provided in appendix 3. This was effectively an evaluation in 'standalone' mode where no reference was made to the SAMMIE system and all input information was provided directly by the user. This is in itself a potentially useful feature which demonstrates the modular nature of the overall system, and is made possible by the structure and operational method of the expert system. i.e. KES and the knowledge base it was provided with are structured with a hierarchical set of knowledge sources such that any missing information will ultimately be sought from the user. The expert system functioned as expected (as shown by the examples) and in doing so demonstrated that it was an effective interactive tool that allowed the user to approach the problem in a way that suited his needs rather than in an order pre-determined by the system. This could be considered as a valuable attribute of the system, but does create a situation where the designer might be unaware of the overall status of the evaluation. i.e. at any particular point in the evaluation it is not easy for the designer to determine which aspects have been evaluated and which are outstanding. This situation is further complicated by the interdependent nature of the different aspects of the evaluations. The provision of structured command sub-sets and extensive provision of information on the progress of the evaluation goes some way to improving this situation (see appendix 3) but this remains an area where substantial improvements could be made. These improvements would need to be based upon a greater understanding and knowledge of how designers approach these problems in practice and could involve extensive observation of designers at work.

The second objective of evaluating the integration of the expert system with SAMMIE was achieved by replacing user provision of geometric data by the use of KES communication files and SAMMIE command files. This proved to be a relatively straightforward process once the information content and file formats had been established. However, this method is not the preferred method for a final implementation of the system and should be considered only as a useful prototyping technique. Although it does maintain a separation between the two systems which is useful in certain circumstances as described above, it also limits the application in a number of ways. The most obvious aspect of this to the user is that the systems, whilst in communication with each other are not truly integrated. Thus the outputs of both systems are presented in different windows of the same computer screen and very different methods of interaction and information presentation are used. A further consequence of this separation is that user interaction is required to activate the communication between the systems. Again, this could be considered as an advantage that it allows the designer to remain in control of the overall process of design and evaluation by the expert system, but it does not represent a genuine monitoring situation. It may be considered preferable to have the expert system operating unseen in the background and reporting back to the designer at the appropriate time. This a complex issue where conflicts in design objectives would have to be handled by the computer system rather than the designer as at present. A further consequence of the communication method employed is that the full power of SAMMIE and KES cannot be used. For example, only those commands that an interactive user of SAMMIE could issue may be embedded in SAMMIE command files. A closer integration of the two systems would allow the expert system to make more direct and detailed inquiries of the SAMMIE data structure.

A simplified car model was used as the basis of the testing and this was constructed in SAMMIE so as to be suitable for the purpose of validating the system. In a more practical situation there may be problems associated with the model not having such a convenient structure. Attempts were made to investigate this aspect by conducting a further case study (Dan, 1996) based on a pre–existing SAMMIE model of the Rover 800. This had not been constructed for use with the expert system, but had all the essential components of instruments, seats, pedals, etc. The case study demonstrated some interesting aspects of this particular car in relation to its compliance with standards, but from the point view of integration it demonstrated that the model structure has to be in a particular form for successful communication. This is because the KES communication files and the SAMMIE command files rely on naming conventions which would not naturally be found unless the model had been constructed for the purpose. Although it was a simple matter to change names in the SAMMIE model so that the system would function, this remains as a difficulty in communication (as indeed it does in many other areas of 'integration') A partial solution to this problem would be the imposition of a model structure on designers using SAMMIE, but this may be unacceptable where design information has to be used across a wide range of activities and not just ergonomics evaluation.

The testing and validation work described can be considered only as a 'technical' evaluation as it makes no attempt to determine the effectiveness of the situation in real and practical situations. There are many issues that can only be addressed by an extensive survey of the system in use by real designers in real situations, including:–

a) the acceptability of the overall approach and the operational details of the implementation.

b) the extent to which the system is answering questions which designers need solutions to and for which other methods do not exist.

c) the possibilities for integrating these ergonomics aspects with other design, engineering and manufacturing aspect of car design.

These issues are considered to be outside the scope of this research work which has concentrated on establishing the feasibility of the approach. Some comments are however made in chapter nine when considering future work.

CHAPTER 9

CONCLUSIONS, CONTRIBUTION OF KNOWLEDGE AND FUTURE WORK

9.1 Introduction

This chapter summarises the results of this research work, assesses the contribution made to knowledge and also discusses the further work that needs to be carried out so that limitations of the expert system for SAMMIE as currently developed can be overcome, and for the extension of the system to a broader domain of knowledge.

A number of problems were encountered during the development of the Ergonomics Design Knowledge Based Expert System (EDKBES), the design prototype of the vehicle interior in SAMMIE and the building of communication and command files.

The objectives of the research work were to construct the EDKBES, build the prototype computer model of the vehicle interior, and to establish integration of the two using KES communication files and SAMMIE command files. This integration of the expert system within the SAMMIE system and its evaluation for "geometric reasoning" was successfully achieved. The resulting system functioned as intended as shown by the validation carried out.

The chosen domain of application was that of the ergonomics design of vehicle interiors because there is a considerable amount of legislation that requires automobile designers to take account of ergonomics.

The KES expert systems shell was selected on the basis of availability but proved to be a quick and simple methodology for the manipulation and representation of knowledge and creation of the EDKBES. The use of frames for knowledge structures was found to be useful and comprehensive, and the addition of semantic representations, concerned with the meaning and relationships between objects, provided substantial power for "geometric reasoning"; an essential component of the overall approach.

The SAMMIE computer-aided design system provided a sophisticated, easy to use and highly interactive environment for the work. Many facilities were available, such as the ability to construct the model of the vehicle interior and to examine a wide variety of views as seen from the driver's eye point and to evaluate the model for human characteristics such as vision, reach and fit.

The system was implemented in a modular form and this proved useful in gaining easy access for communication and also worked well as a design critic to evaluate the design against a set of specified constraints contained in the EDKBES. The ability to diagnose faults and make recommendations for improvements to the design in a user-friendly dialogue was implemented.

The integration between the EDKBES and SAMMIE systems for vehicle interior design was accomplished through the use of external program communication files and SAMMIE command files. This method represents a two-way connection in which the SAMMIE system extracts and transfers geometric information from the design model. The EDKBES expert system subsequently identifies and utilises the appropriate data for evaluation, and returns information to the SAMMIE system for model modification.

The two systems have different knowledge structures and thus a naming convention was established for external program communication. Thus, the SAMMIE system used the command processing system to present the information in a format that the Expert System could understand and manipulate.

The integration between the Ergonomics Design Knowledge Base Expert System (EDKBES) and SAMMIE functioned successfully and established that the objectives had been achieved in making the standards and legislation, rules and regulations, and design working practice available to vehicle interior design.

The author's intention is that the research work will be continued in UTM, Malaysia and applied for research and development in the Malaysian automotive industry, where the

National Automotive Industry (PROTON) is engaged in the development of new car design.

9.2 The Contribution to Knowledge

The application of an Expert System in the SAMMIE environment would be very useful in supplying the designer or engineer with necessary technical information, acting as a design critic by checking the design against the specifications and constraints, and as a feed–back mechanism. This is seen as being useful at all stages of the design process, but particularly at the early conceptual stages. In this way the designer is relieved of some technical aspects and is freed to concentrate on the creative aspects of the design process.

The specific advantages of the method are:–

a) As a design assistance tool for the designer or engineer to monitor and modify his design repeatedly and observe the results of different designs and specifications.

b) As a decision making assistance tool to provide a high level design environment helpful to both the expert and novice user. An Ergonomics Design Knowledge Base Expert System (EDKBES) is useful in judgmental and problem solving areas, where constraints may not be uniquely resolved by theory or algorithm. EDKBES can help the designer and engineer to solve the problem part of the design process.

The EDKBES is an example of expert systems applied to ergonomics, a field which is appropriate because of the difficulties that ergonomists have in providing their knowledge within areas traditionally dominated by engineering considerations.

The SAMMIE system has effectively been enhanced by the provision of intelligence in the form of expert system support. The general approach has been shown to make a valuable contribution to the development of such systems.

The SAMMIE Computer Aided Ergonomics Design System has specifically been extended for evaluation of "geometric reasoning" to facilitate vehicle interior design in areas such as visibility through the steering wheel for display adjustment (including speedometer, tachometer, etc.), driver's views adjustment and reach to pedals through seat adjustment, and pedal spacing.

An extensive knowledge base of technical information for standards and legislation, rules and regulations and ergonomics design for automobiles has been collected and made available for practical implementation through the expert system.

9.3 Future Research and Development Work

Some of the suggestions that can be made for further research and development or improvement work for the knowledge base EDKBES and the integration of the EDKBES within the SAMMIE system are:–

A considerable amount of knowledge has been collected and rules have been developed and written into a knowledge base for the creation of an Ergonomics Design Knowledge Based Expert System (EDKBES). This was sufficient to establish that standards and legislation, ergonomics guidelines and design working practice can be related to the geometric aspects of design in this way. However, the result of the test and validation has indicated some areas where additions and modifications to the production rules are desirable.

For example there is considerable scope for broadening the knowledge domain within the field of ergonomics. This would include a range of aspects such as mirrors, hand levers and seat design that could be handled in a manner very similar to the approach described in this thesis i.e. these are areas where geometric considerations dominate and where the SAMMIE system can be used as an evaluation tool. Beyond this there are several other ergonomics issues which could be included provided that a suitable evaluation tool were available. For example, thermal comfort as related to seat design is an important aspect of both overall driving performance and the acceptability of cars.

There is also a need to relate ergonomics design issues to engineering design and manufacturing. e.g. Designing for ergonomics alone can generate engineering problems that either cannot be overcome or are very costly. There is a need to investigate how an expert system of the type described can be integrated with similar systems from these other application domains. This idea of distinct expert systems that have their own knowledge base but are able to collaborate with other expert systems is a very active research area.

A potential problem with collaborating expert systems is the opportunity that arises for conflict between their recommendations. This could also been seen in EDKBES itself as its rule base is expanded to cover a wider domain. Further work would be required to identify these conflicts and resolve them either by intervention of the designer or computer control.

The present system evaluates a particular design aspect at the request of the user. A future objective would be to use the expert system in a monitoring role such that it reports violations of design criteria as they occur. In this situation it would be necessary however to provide a 'focus' for the monitoring so that reports are only made on aspects which the designer is currently interested in.

There is scope for improving the detail of the current knowledge and its implementation. Certainty factors play an important role in decision-making and are currently defined in a rather subjective way. A study of the relative importance of the various aspects of design could lead to more precise and objective measures that would hopefully improve the overall performance of the system.

The integration of EDKBES within the SAMMIE system has been achieved by external programs using communication files and command files, to create a two-way connection for geometric reasoning.

It would be desirable to create a closer integration by embedding KES in SAMMIE and the use of languages such as C or C ++. This would allow considerably improved access to the geometric data structure within SAMMIE and thus extend the range of geometric reasoning that could be applied.

The current prototype system would require considerable development for it to become a practical tool for designers. In particular the system should be quick and easy to use which implies a higher level of integration as described above and improved user interfaces.

A more comprehensive knowledge and rule base would be required to ensure that a significant part of the designers overall task can be handled.

REFERENCES

Ali, A.Tajuddin, *Malaysia Required Standards in Next 15 Years Toward 2020 – Sirim's Appointment of the Malaysian Iron and Steel Federation (MISIF) as the Iron and Steel Industries Standards Writing Organisations,* Business Times, Kuala Lumpur, Malaysia, 1995.

Alty, J.L., and Coombs M.J., *Expert System Concepts and Examples,* NCC Publications, 1984.

Alty, J.L., *Use of Expert System,* Computer–Aided Engineering Journal, pp. 2–9, Taylor & Francis, London, UK, 1985.

Austen, B., *Computers and 3D Product Design Education,* The Design Council, London, UK, 1985.

Azarkadeh, S., *The Integration of an Expert System to a Geometric Computer–aided Design System,* M.Sc. Thesis, Loughborough University of Technology, 1988.

Badiru, A.B., *Expert Systems Application in Engineering and Manufacturing,* Prentice–Hall, Englewood Cliffs, New Jersey, USA, 1992.

Badler, N.I., Kamran H. Manoocheri and Graham Walters, *Articulated Figure Positioning by Multiple Constraints,* IEEE Computer Graphics and Applications, pp. 28–38, 1987.

Barr, A., and Feigenbaum, E.A., *The Handbook of Artificial Intelligence,* Volume 1, Heuristech, 1981, William Kaufmann, Los Altos, USA, 1981.

Barr, A., and Feigenbaum, E.A., *The Handbook of Artificial Intelligence,* Volume 2, William Kaufmann, Los Altos, USA, 1982.

Beynon–Davies, P., *Expert Database Systems, a Gentle Introduction,* McGraw–Hill, London, UK, 1991.

Black W.J., *Intelligent Knowledge Based System, an Introduction,* Van Nostrand Reinhold, 1986.

Blakeley, F.M., *CYBERMAN*, Chrysler Corp., Detroit, Michigan, U.S.A., 1980.

Bonnet, A., *Artificial Intelligence Promise and Performance*, Prentice Hall International, 1985.

Bonney, M.C., Taylor, N.K., and Case, K., *Using Computer–aided Design and Expert System for Human Workplace Design*, in Woodwark, J., (Eds), *Geometric Reasoning*, pp. 269–282, Clarendon Press, Oxford, 1989.

Buchanan, B. G. and Duda, R. O., *Principles of Rule-Based Expert System in Advances of Computers*, Vol. 22, Academic Press, London, UK, 1983.

Business Times, *Proton Zooms to Record 1993 Sales*, Business Times, Kuala Lumpur, Malaysia, 1994.

Case, K., Porter, J. M., and Bonney, M.C., *Design of Mirror Systems for Commercial Vehicles*, Applied Ergonomics, Vol. 11, No. 4, pp. 199–206, 1980.

Case, K., and Porter, J. M., *SAMMIE*, The Ergonomist, No.124, 1980.

Case, K., Bonney, M.C. and Porter, J.M., *Computer Graphics Standards For Man Modelling*, Computer Aided Design Journal, Vol. 21, No.5, pp. 257–268, 1989.

Case, K., Bonney, M.C., Porter, J.M. and Freer, M.T., *Application of the SAMMIE CAD System in Workplace Design*, in Haslegrave, C.M., Wilson, J.R., Corlett, E.N., and Manenica, I. (Eds) *Work Design in Practice*, pp 119–127, Taylor and Francis, London, UK, 1990.

Case, K., Porter, J.M. and Bonney, M.C., *Anthropometry and Computer Models of Man*, in Wilson, J.R. and Corlett, E.N. (Eds) *Evaluation of Human Work – A Practical Ergonomics Methodology*, pp. 472–499, Taylor & Francis, London, UK, 1990.

Case, K., Porter, J.M. and Bonney, M.C., *SAMMIE: A Man and Workplace Modelling System*, in Karwowski, W., Genaidy, A.M, and Asfour, S.S. (Eds) *Computer-Aided Ergonomics a Researcher's Guide*, pp. 31–56, Taylor & Francis, London, 1990.

COSTED, *Computer Aided Design in Mechanical Engineering*, Kuala Lumpur, Malaysia July 7–12, 1987, Workshop Organised by COSTED (Committee on Science and Technology in Developing Countries), and University of Technology Malaysia, Malaysia, 1987.

Crapo, H. and Rotage, J.F., *A High–level Language Environment for the Definition and Manipulation of Geometric Forms,* in Woodwark, J., (Eds), *Geometric Reasoning*, pp. 31–41, Clarendon Press, Oxford, 1989.

Dan, M.P., *A Case Srudy – The Rover Car*, Internal Report, Loughborough University of Technology, 1996.

Davenport, J.H., *Robot Motion Planning*, in Woodwark, J., (Eds), *Geometric Reasoning*, pp. 43–51, Clarendon Press, Oxford, 1989.

Dempster, W.T., *Space Requirements of the Seated Operator*, WADC (Wright Air Development Center) Technical Report, pp. 55–159, Ohio, USA, 1955.

Dooley, M, *Anthropometric Modelling Programs – A Survey,* IEEE Computer Graphics and Applications, Vol. 2, pp. 17–25, 1982.

Dreyfuss, H., *The Measure of Man–Human Factors in Design*, Whitney Library of Design, New York, USA, 1966.

Dym, C.L, *Engineering Design – A Synthesis of Views,* Cambridge University Press, London, UK, 1994.

Elias, H.J. and Lux, C., *The Design of Ergonomically Optimized Workstations and Products Using Franky and CAD*, REF A Nachtrichten, Vol. 3, pp 5–12, 1986.

EON, *Statistics on the Exports of Proton Saga*, EON (National Automotive Distributor), Utusan Malaysia, 19th June 1991, Kuala Lumpur, Malaysia, 1991.

Fetter, W.A, *Computer Graphics Human Figure Application of Biostereometrics,* Computer–aided Design Journal, Vol.12, No. 4, pp. 175–179, 1980.

Fetter, W.A., *A Progression of Human Figures Simulated by Computer Graphics*, IEEE Computer Graphics and Applications Journal, pp.9–13, 1982.

Ford DSE, *A Guide to Occupant Safety – For's Dynamic Safety Engineering (DSE)*, Ford Company, United Kingdom, 1994.

Forsyth, R., *Expert System – Principles and Case Studies*, Chapman and Hall Computing, London, 1984.

Fortin, C., Gilbert, R., Beuter, A., Laurent, F., Schiettekatte J., Carrier, R., and Dechamplain, B., *SAFEWORK: A Microcomputer–Aided Workstation Design and Analysis. New Advances and Future Developments.*, in Karwowski, W., Genaidy, A.M, and Asfour, S.S. (Eds) *Computer Aided Ergonomics a Researcher's Guide*, pp 157–180, Taylor & Francis, London, 1990.

Galer, M., *The Presentation of Information in Cars*, Ph.D Thesis, Loughborough University of Technology, 1985.

Galer, I.A.R., *Applied Ergonomics Handbook*, Butterworths, London, UK, 1987.

Gero, John S., *Expert Systems in Computer–Aided Design*, North–Holland, Amsterdam, Netherlands, 1987.

Gero, John S., *Design Prototype: A Knowledge Representation Schema for Design*, AI Magazine, Vol. 11, No. 4, pp. 26–36, 1990.

Goodall, A., *The Guide of Expert Systems*, Learned Information, Cambridge, 1983.

Harmon, Paul and Kings, D., *Expert System*, John Wiley, New York, USA, 1985.

Harmon, Paul, Maus, R. and Morrissey, W., *Expert System Tools and Applications*, John Wiley, New York, USA, 1988.

Hammond, D., and Roe, R.W., *SAE Controls Reach Study*, SAE paper 720199, at the SAE Automotive Engineering Congress and Exposition, Detroit, January 1972.

Hammond, D., Manres, D., and Razgunas, L., *Control Reach – The Hand Reach of Drivers*, SAE paper 750357, at the SAE Automotive Engineering Congress and Exposition, Detroit, February 1975.

Haslegrave, C.M., *Visual Aspect in Vehicle Design*, (Eds) Peacock, B., and Karwowski, W., *Automotive Ergonomics*, pp. 79–98, Taylor and Francis, London, UK, 1993.

Hawkes, B. and Abinett, R., *The Engineering Design Process*, Pitman, London, 1984.

Hayes–Roth, F., Waterman, D.A. and Lenat, D.B., *Building Expert System*, Addison–Wesley, Reading MA, 1983.

Heng, M.S.H., *Why Evolutionary Development of Expert Systems Appear to Work*, Future Generation Computer System Journal, Vol. 3, No. 3, pp 152–168, 1987.

Hickey, D., Pierrynowski, M. and Rothwell, P., *Man–Modeling CAD Programs for Workspace Evaluation*, Downsview, Toronto, 1985.

Hoekstra, P.N., *Computer Aided Anthropometric Assessments: Seeing What you are Doing*, International Conference On Engineering Design (ICED) 1993, Vol. 1, 2 and 3, The Hague, August 17–19, Heurista, Zurich, Switzerland, 1993,

Ibrahim, A., *Second Car by End of 1994*, New Straits Times 3 February 1993, Kuala Lumpur, Malaysia, 1993.

Jared, G.E.M., *Recognizing and Using Geometric Features*, in Woodwark, J., (Eds), *Geometric Reasoning*, pp. 169–183, Clarendon Press, Oxford, 1989.

Jiiptner, H., *ISO Standards in the Field of Ergonomics: A Report of ISO/TC 159 Activities*, Applied Ergonomics, Vol. 15.3, pp. 211–213, Taylor & Francis, London, UK, 1984.

Jiiptner, H., *Ergonomics in Automobile Design–Technology and Human Factors*, International Conference on Auto Technology, November 12–14, 1990, pp 12–14, Chulalongkorn Univerity, Bangkok, Thailand, 1990.

Jones, M., *Fit for Humans*, CADCAM International Journal, Issued November, pp. 44–48, 1988.

Kennedy, S., *Task Modelling in Computer Aided Workplace Design*, Ph.D Thesis, Loughborough University of Technology, Loughborough, 1995.

KES, *KES – Knowledge Engineering System, Knowledge Base Author's Manual*, Knowledge Engineering System – KES PS, and HT, Software Architecture and Engineering, USA, 1990.

Kidd, A.L., *Human Factors in Expert Systems*, Ergonomics Society's Conference Human Engineering, Ergonomics Society, UK, 1983.

Kimberlie, H.P., *Ergonomics Sourcebooks, A Guide to Human Factors Information, Ergosyst Associate*, Incorporation Publication, London, 1987.

Kloke, W.B., WERNER: *A Personal Computer Implementation of an Extensible Anthropometric Workplace Design Tool*, in Karwowski, W., Genaidy, A.M, and Asfour, S.S. (Eds) *Computer–Aided Ergonomics a Researcher's Guide*, pp 57–67, Taylor & Francis, London, UK, 1990.

Kuusisto, A., *Computer–Aided Ergonomics Design: Current Status*, in Haslegrave, C.M., Wilson, J.R., Corlett, E.N., and Manenica, I. (Eds) ***Work Design in Practice***, pp 98–102, Taylor & Francis, London, UK, 1990.

Kuusisto, A. and Mattila, M., *Anthropometric and Biomechanical Man Model in Computer–Aided Ergonomic Design Structure and Experiences of Some Programs*, in Karwowski, W., Genaidy, A.M, and Asfour, S.S. (Eds) *Computer Aided Ergonomics a Researcher's guide*, pp 104–114, Taylor and Francis, London, UK, 1990.

Launis, M., and Lehtela, J., *Ergonomics Design of Workplaces with a Two–Dimensional Micro–CAD System*, in Haslegrave, C.M., Wilson, J.R., Corlett, E.N., and Manenica, I. (Eds) *Work Design in Practice*, pp 110–118, Taylor & Francis, London, UK, 1990.

Launis, M., and Lehtela, J., 1990, *Man Models in the Ergonomic Design of Workplaces with the Microcomputer*, in Karwowski, W., Genaidy, A.M, and Asfour, S.S. (Eds)

Computer–Aided Ergonomics a Researcher's Guide, pp. 68–79, Taylor & Francis, London, UK, 1990.

Levine, R.I., Drang, D.E., and Edelson, B., *A Comprehensive Guide to Artificial Intelligent and Expert System,* A Byte Book, McGraw–Hill, New York, USA, 1986.

Majchrzak, A., Chang, T.C., Barfield, W., Eberts, R. and Salvendy, G., *Human Aspects of Computer–Aided Design,* Taylor and Francis, London, UK, 1987.

McCormick, E. H., and Saunders, M.S., *Human Factors in Engineering and Design,* McGraw–Hill, London, UK, 1993.

McDaniel, J.W., *Models for Ergonomic Analysis and Design COMBIMAN and CREW CHIEF,* in Karwowski, W., Genaidy, A.M, and Asfour, S.S. (Eds) *Computer Aided Ergonomics a Researcher's Guide,* pp 138–156, Taylor & Francis, London, UK, 1990.

Mercedes–Benz, SRS, Mercedes–Benz Airbags for Retro–fitting to 190 and Mid–series Models, Mercedes–Benz Centre, Milton Keynes, UK, 1994.

Minsky, M., *A Framework for Representing Knowledge,* The Psychology of Computer Vision, P.H. Winston, McGraw–Hill, New York, USA, 1972.

Mohamed, M., *The Second Outline Perspective Plan – 1991–2000, Setting the Stage for Vision 2020 – The Special Malaysia Government Report on Second Outline Perspective Plan (OPP2) of Malaysia 1991–2000, Included the Sixth Malaysia Plan 1991–1995,* Message in Parliament Cabinet, Prime Minister of Malaysia, New Straits Time, 18th June 1991.

Oborne, D.J., *Ergonomics at Work, Human Factors in Design and Development,* John Wiley and Sons, New York, 1995.

Okey, R.E., Suffell, C. and Blount, G.N., *Initial Work on a System–Independent Computer Model of a 3D Anthropomorphic Dummy,* Computer Aided Design Journal, Vol. 21, No. 6, pp 393–403, 1989.

Olson, J.R., and Rueter, H.H., *Extracting Expertise from Experts: Methods for Knowledge Acquisition*, Expert System Journal, Vol.4, No.3, pp. 152–168, 1987.

O'Shea, T., et al., *Intelligent Knowledge-Based Systems, an Introduction*, Harper & Row, London, UK, 1987.

Pang, G.K.H., and MacFarlane, A.G.J., *An Expert System Approach to Computer-Aided Design of Multivariable Systems*, Lecture Notes In Control & Information Sciences Series, Vol. 89, Springer–Verlag, Berlin, Germany, 1987.

Patla, A.E., and Eickmeier, W.E., *COMMONS: A Computer-Based Movement Data Management and Simulation System*, Human Movement Science, Vol.8, pp. 161–176, Taylor and Francis, London, UK, 1989.

Peterson, M., Lundberg, D., and Anders, E.K., *Knowledge Engineering – System Design in Diffuse Domains*, Studentlitteratur, Telelogic, Sweden, 1990.

Pheasant, S. T., *A Technique for Estimating Anthropometric Data from the Parameters of the Distribution of Stature in Ergonomics*, Ergonomics Journal, Vol 25, No. 11, pp. 981–992, Taylor and Francis, London, UK, 1982.

Pheasant, S.T., 1984, *Anthropometric – an Introduction for Schools and Colleges, PP 7310 BSI. Education*, British Standards Institution, London, UK, 1984.

Pheasant, S.T., *Body Anthropometry, Ergonomics and Design*, Taylor and Francis, London, UK, 1986.

Porter, J.M., Case, K., Freer, M.T., and Bonney M.C., *Computer-Aided Ergonomics Design of Automobiles*, in Peacock, B. and Karwowski, W. (Eds) *Automotive Ergonomics*, pp. 43–77, Taylor & Francis, London, UK, 1993.

Post, F.H and Smeets, J.W., *ADAPS: Computer Aided Anthropometrical Design*. Tijdschrift Voor Ergonomie, Vol.6, (4), pp. 11–18 (in Dutch), 1981.

Rao, A.R., and Jain, R., *Knowledge Representation and Control in Computer Vision Systems*, IEEE Expert Journal, 1988.

Requicha, A.A.G., *Representations for Rigid Solids: Theory, Methods and Systems, Computer Surveys*, Computer Surveys Journal, Vol. 12 (4), pp. 437–463, 1980.

Requicha, A.A.G. and Voelcker, H.B., *Solid Modelling: a Historical Summary and Contemporary Assessment*, IEEE Computer Graphics and Applications Journal, pp. 9–24, 1982.

Rich, E., *Artificial Intelligence*, McGraw–Hill, New York, USA, 1993.

Roe, R.W., *Reach to Other Types of Controls Design Devices*, SAE Subcommittee, SAE, April 1972.

Roe, R.W., *Occupant Packaging*, (Eds) Peacock, B., and Karwowski, W., *Automotive Ergonomics*, pp. 11–41, Taylor and Francis, London, UK, 1993.

Roebuck, J.A., Kroemer, K.H.E. and Thomson, W.G. *Engineering Anthropometry Methods*, John Wiley & Sons, London, UK, 1975.

Rothwell, P.L. and Hickey, D.T., *Three–Dimensional Computer Models of Man*, Proceedings of the Human Factors Society 30th Annual Meeting, pp. 216–220, Human Factors Society, Dayton, Ohio, USA, 1986.

SAMMIE CAD, *The SAMMIE System Training Manual, Version 4.11*, System for Aiding Man–Machine Interaction Evaluation – SAMMIE CAD Limited, Loughborough, UK, 1990.

Schoen, S., and Sykes, W., *Putting Artificial Intelligence to Work – Evaluating and Implementing Business Applications*, John Wiley, New York, USA, 1987.

Simons, G.L., 1984, *Introducing Artificial Intelligence*, The National Computer Centre Limited, London, UK, 1984.

Simons, G.L., *Expert Systems and Micros*, The National Computer Centre Limited, London, UK, 1985.

Stobart, R.K., Williams, D.J., *The Context of Geometric Reasoning in Manufacturing Cells*, in Woodwark, J., (Eds), *Geometric Reasoning*, pp. 237–265, Clarendon Press, Oxford, 1989.

Tan, B.C., *Anthropometric Measurements of Malaysian Drivers Population,* B.Sc. Mechanical Engineering Thesis, University Technology of Malaysia, Malaysia, 1986.

Thalmann, N.M. and Thalmann, D., *Computer Models of Animating Synthetic Actors,* IEEE Computer Graphics and Applications, Vol. 5, No. 11, pp 32–44, 1991.

Tharumagnanam, T, *The Making of the National Car – not Just a Dream,* Tassmag, Kuala Lumpur, Malaysia, 1994.

Tomiyama, T., and Yoshikawa, H., *Requirements and Principles for Intelligent CAD Systems,* IFIP Working Conference: Knowledge Engineering in Computer Aided Design Journal. Budapest, September 17–19, 1984.

Vellinga, R., *The Human Factor: Ergonomics with AUTOCAD and CADPEOPLE,* ecCADENCE, Issued November, pp 59–62, 1992.

Verriest, J.P., Trasbot, J. and Rebiffe, R., *MAN3D: a Functional and Geometrical Model of the Human Operator for Computer Aided Ergonomic Design,* in Karwowski, W. and Yates, J.W., (Eds) *Advances in Industrial Ergonomics and Safety 3*, pp. 901–908, Taylor and Francis, London, 1991.

Walters, G., *Mannequin Humancad:* User Guide, Biomechanic Corporation of America, New York, USA, 1989.

Walters, G., *Mannequin: A Human Dimension to PC CAD*, Computer Aided Draughting and Design, Vol.11, 1991.

Westerink, J., Tragter, H., Van Der Star, A and Rookmaaker, D.P., *TADAPS: a Three-Dimensional CAD Man Model,* in Karwowski, W., Genaidy, A.M, and Asfour, S.S. (Eds) *Computer-Aided Ergonomics a Researcher's Guide,* pp. 90–103, Taylor & Francis, London, UK, 1990.

Woodson, C., *Human Engineering Guide for Equipment Designers*, University of California Press, 1964.

Yong, S.K.T., *Industrial Technology Development – A National Plan of Action*, Report on the Coordination and Transfer of Industrial Technology, Minister of Science, Technology and the Environment, Government Printer, Kuala Lumpur, Malaysia, 1990.

Zeid, Ibrahim, *CAD/CAM Theory and Practice*, MacGraw–Hill, New York, USA, 1991.

Zhao, X. and Badler, N.I., *Interactive Body Awareness*, Computer–Aided Design, Vol. 26, No. 12, pp. 861–867, Butterworth–Heinemann Ltd., London, 1994.

REFERENCES

STANDARDS AND CODES OF VEHICLE INTERIOR OR AUTOMOTIVES SERIES

ADR – Australian Design Rules: Standards Rules and Regulations.

ADR 18/00, 1988, *Design Rules and Regulations for Instrumentation*, ADR, Australia.

BSI – British Standards Institution: British Standard Automobile Series.

BSI EN 979, CEN/TC 122/WG 1, EN, *Draft for Ergonomics, Draft European Standard – Basic List of Definitions of Human Body Dimensions for Technical Design*, BSI, UK., 1993.

BSI AU 143c, *Specification for Symbols for Controls, Indicators and Tell–Tales for Road Vehicles*. Also refer to ISO 2575, 1982 *ISO: Road Vehicles – Symbols for Controls Indicators and Tell–Tales*, BSI, UK., 1984.

BSI BS AU 176, *Method for Establishment of Eyellipses for Driver's Eye Location*. Also refer to ISO 4513, 1978, *ISO: Road Vehicles – Visibility – Method for Establishment of Eyellipses for Driver's Eye Location*, BSI, UK., 1980.

BSI AU 179, *Dimensional Codes for Passenger Cars*. Also refer to ISO 4131, 1979 *ISO: Road Vehicles – Dimensional Codes for Passenger Cars*, BSI, UK., 1981.

BSI AU 183, *Specification for Symbols for Passive Seat Belt Systems*. Also refer to SAE J383, 1986 *SAE: Design Recommendations – Motor Vehicle Seat Belt Anchorages*, BSI, UK., 1983.

BSI AU 199, *Specification for Location of Hand Controls, Indicators and Tell–tales in Passenger Cars*. Also refer to ISO 4040, 1983 *ISO: Road Vehicles – Passenger Cars – Location of Hand Control, Indicators and Tell–Tales*, BSI, UK., 1984.

CEN – European Committee for Standardization: Standards and Informations.

CEN/TC 122, *Basic List of Definitions of Human Body Dimensions for Technical Design*, 1992.

EEC – European Economic Community: European Economic Community Directives.

EEC Intereurope, *Specifications for Motor Vehicle Identification of Controls/Warning Lights*, EEC, Council Directive, 1979.

EEC Directive 127, (as amended by Directive 1979/795/EEC, 1982/205/EEC, 1988/321/EEC), *The Approximation of the Laws of the Member States Relating to the Rear-view Mirrors of Motor Vehicles*, EEC, Council Directive, 1971.

EEC Directive 317, *Windscreen Wiper and Washer Systems*, EEC, Council Directive, 1978.

EEC Directive 318, *Defrosting and Demisting Systems*, EEC, Council Directive, 1978.

EEC Directive 649, (as amended by Directive 81/643, 88/366), *Field of vision of motor vehicle drivers*, EEC, Council Directive, 1977.

EDS: *Ergonomics of Display Systems for Austin Rover Ltd. The standards ergonomics of display system for Austin Rover*, Ergonomics of Display Systems (EDS), UK.

FMVSS – Federal Motor Vehicle Safety Standard in USA.

FMVSS 103, *Defrost/Demist System*, FMVSS, USA., 1991.

FMVSS 104, *Wash/Wipe System*, FMVSS, USA., 1991.

FMVSS 111, *Wash/Wipe system – Rear View Mirrors*, FMVSS, USA., 1991.

Ford DES, *A Guide to Occupant Safety – Ford's Dynamic Safety Engineering (DSE)*, Ford Company, UK., 1994.

ISO – International Standards Organisation: Standards and Information for Automobile Series.

ISO 2575, *Road Vehicles – Symbols for Controls, Indicators and Tell-Tales*, ISO, USA, 1982.

ISO 3833, *Road Vehicles – Types – Terms and Definitions*, ISO, USA, 1977.

ISO 3958, *Road Vehicles – Passenger Cars – Driver Hand Control Reach*, ISO, pp. 01–15, USA, 1977.

ISO 4040, *Road Vehicles – Location of Hand Controls, Indicators and Tell-tales in Passenger Cars*, ISO, USA, 1983.

ISO 4130, *Road Vehicles – Three-dimensional Reference System and Fiducial Marks – Definitions*, ISO, USA, 1977.

ISO 4131, *Road Vehicles – Dimensional Codes for Passenger Cars*, ISO, USA, 1979.

ISO 4513, *Road Vehicles – Visibility Method for Establishment of Eyellipses for Driver's Eye Location*, ISO, USA, 1978.

ISO 6385, *Road Vehicles – Ergonomics Principles in the Work Systems*, ISO, pp. 06–15, USA., 1981.

ISO 6549, *Road Vehicles – Procedure for Heel-Point Determination*, ISO, USA., 1980.

SAE – Society of Automotive Engineers: The Engineering Resource for Advancing Mobility Land, Sea, Air and Space.

SAE Handbook, *SAE Handbook, On Highway Vehicles and Off Road Highway Machinery*, Vol. 4, SAE, Warrendale, USA, 1990.

SAE P–123, *An Update on Automotive Electronic Displays and Information Systems*, SAE, Warrendale, USA, 1988.

SAE J287, *Driver Hand Control Reach*, SAE, Warrendale, USA, February 1980 and 1990.

SAE J383, *Motor Vehicle Seat Belt Anchorages – Design*, SAE, Warrendale, USA, 1986.

SAE SP–576, *Ergonomics Aspects of Electronic Instrumentation: A Guide for Designers*, SAE, Warrendale, USA, 1984.

SAE SP–654, *Electronic Displays and Information Systems SP–654*, SAE, Warrendale, USA, 1986.

SAE SP–734, *Automotive Electronic Displays and Information Systems*, SAE, Warrendale, USA, 1983 and 1988.

SAE J826, *Devices for Use in Defining and Measuring Vehicle Seating Accommodation*, SAE, Warrendale, USA, 1990.

SAE J834a, *The Approximation of the Laws of the Member States Relating to the Rear–view Mirrors of Motor Vehicles* and subsequently adopted in law by the European Economics Community (EEC 127, 1971), SAE, Warrendale, USA, 1967.

SAE J879b, *Motor Vehicle Seating System*, SAE, Warrendale, USA, 1968.

SAE J941e, SAE J941, *Motor Vehicle Driver's Eye Range*, SAE, Warrendale, USA, 1983 and 1985.

SAE J985, *Vision Factor Considerations in Rear View Mirror Design*, SAE, Warrendale, USA, 1985.

SAE J1050a, *Describing and Measuring The Driver's Field of View*, SAE, Warrendale, USA, 1977.

SAE J1052, *Motor Vehicle Driver and Passenger Head Position*, SAE, Warrendale, USA, 1990.

SAE J1100, *Motor Vehicle Dimension (Both Interior and Exterior)*, SAE, Warrendale, USA, 1990.

SAE J1138, *Design Criteria Driver Hand Controls Location for Passenger Cars*, SAE, Warrendale, USA, 1977.

SAE J1139, *Supplemental Information Driver Hand Control Location for Passenger Cars*, SAE, Warrendale, USA, 1977.

SAE J1516, *Accommodation Tool Reference Point*, SAE, Warrendale, USA, 1990.

SAE J1517, SAE: *Driver Select Seat Position*, SAE, Warrendale, USA, 1990.

APPENDIX 1

1 SEATS

The following list establishes the ranges of significant dimensions extracted from BS AU 179, (1981), ISO 4131, (1979), SAE J1100, (1990), Roebuck, (1975) and Roe, (1989) (Figure 1.1):–

DRIVER'S SEAT DIMENSIONS:

Items/Dimension in mm/deg	BSI/ISO	SAE	Ref.No
Vertical distance from floor SgRP Heel pt to h–point	127–130	177	H30
Horizontal dist. from Heel point to Heel–point SgRP	130–508	917	L53
Vertical seat adjustment	38–50	57	H59
Design Heel–point front travel		189	L17
Normal driving and riding seat track travel	102–165	189	L23
Distance from zero X plane to H–point		727	L31
Head room		938	H61
Maximum leg room (accelerator)		1094	L34
Shoulder room		1348	W3
Hip room		1308	W5
Back angle	5–40 deg	25 deg	L40
Hip angle		98 deg	L42
Knee angle		134 deg	L44
Foot angle		87 deg	L46
Upper body opening to ground		1207	H50

The following list establishes the ranges of significant dimensions for passenger seat extracted from SAE J1100, (1990), and Roe, (1989):–

PASSENGER SEAT DIMENSIONS:

Items/Dimension in mm(inches)	SAE	Ref.No
SgRP Point couple distance	616	L50
Effective Head room	867	H63
Minimum effective leg room	688	L51
Vertical dist. from H–point rear to heel point rear	265	H31
Knee clearance	–112	L48

Shoulder room	1299	W4
Hip room	1146	W6
Back angle	28 deg	L41
Hip angle	82 deg	L43
Knee angle	65 deg	L45
Foot angle	106 deg	L47

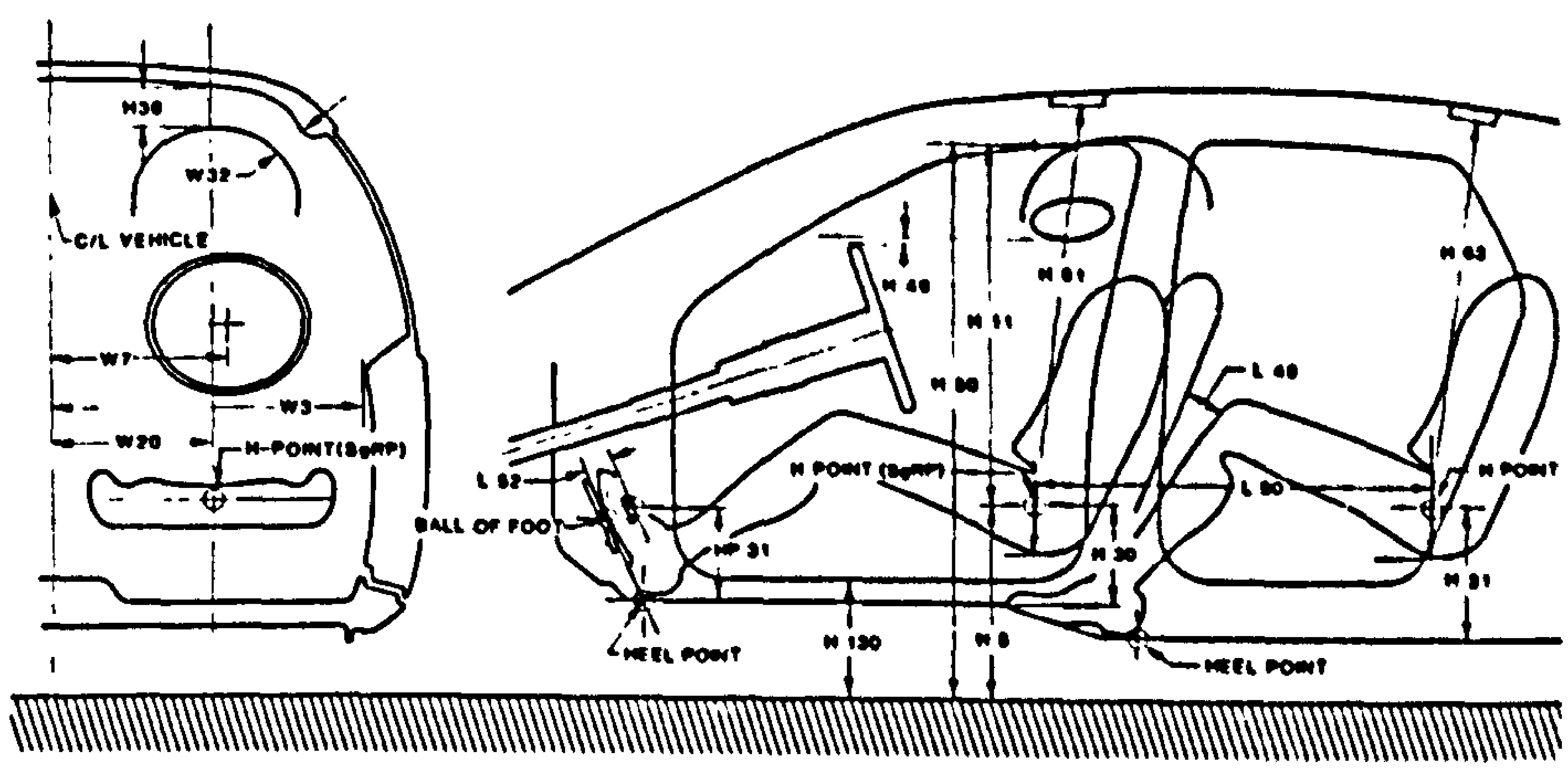


Figure 1.1 Dimensions in SAE J1100 that define interior vehicle design (Source: SAE J1100, 1990).

2 Primary Controls: Steering Wheel

The following list establishes from the standards and legislation, relates to steering wheel dimensions as defined in ISO 3958, (1977) and SAE J1050, (1990) :–

Items Dimension in mm/degrees/Newtons	BSI/ISO	SAE	Ref. No
Steering wheel diameter	330–410	375	W9
Steering wheel weight resistance to motion	56–79 N	56–90 N	W9
Steering wheel angle	10 – 70 deg.	20 deg.	H18
Accele. heel pt. horiz. to steering wheel centre	500–660	506	L11
Accele. heel–pt. vert. to steering wheel centre	530–838	548	H17

3 Steering Wheel: Visibility

Driver Visibility – Visual aspect in interior vehicle design, limits to visual field, eye and head movement are defined in (SAE J985, 1967):–

Dist.visual eye and head movement	Horizontal left/right	Vertical up/down
Binocular field	120 deg.	0 deg.
Right Monocular field	150 deg.	0 deg.
Eye rotation min.	30/30 deg.	15/15 deg.
Eye rotation max.	45/65 deg.	45/65 deg.
Head turn min	45/60 deg.	45/60 deg.
Head turn max	45/60 deg.	45/60 deg.
Head tilt	30/30 deg.	30/30 deg.

APPENDIX 2

AN ERGONOMICS DESIGN KNOWLEDGE BASED EXPERT SYSTEM (EDKBES)

```
\AN ERGONOMICS DESIGN KNOWLEDGE BASED EXPERT SYSTEM (EDKBES)
\INTEGRATION WITHIN SAMMIE AS AN AIDED IN DESIGN PROCESS.

constants:
inputdata:      "spfile, spefile, tafile, tacfile,",
                "fufile, fucfile, tpfile, tpcfile,",
                "dofile, doefile, acfile, acrfile,",
                "bkfile, bkrfile, clfile, clrfile,".
outputdata:     "spfile, spcfile1, tafile1, tacfile1,",
                "fufile1, fucfile1, tpfile1, tpcfile1,",
                "dofile1, doefile1, acfile1, acrfile1,",
                "bkfile1, bkrfile1, clfile1, clrfile1,".

\The following constraints are used for messages in the action section.

welcome:
"                                                                          ",
"WELCOME TO THE INTERIOR OF A CAR DESIGN MONITOR EDKBES.                  ",
"                                                                          ",
"**************************************************************************",
"                                                                          ",
"This Ergonomics Design Knowledge Base Expert System EDKBES knowledge base ",
"attempts to assist, the engineer and designer while designing for driver visibility ",
"through steering wheel, viewing with eye movement and no head movement, driver's ",
" seat adjustment for pedal locations and reach to accelerator, brake and clutch. ",
"                                                                          ",
"The EDKBES provides the necessary assistance in an area where there is a wealth of ",
"legislation. The resources available will be inferred in determining the choice of ",
"various entities for standards and legislation, rules and regulations, information on ",
"ergonomics standards and specifications, the design working practice to be related ",
"to the geometric reasoning aspects of the design process. The design is monitored ",
"against these standards and information in the window environment.        ",
"                                                                          ",
"PROGRAMMER:       MD.DAN BIN MD. PALIL.                                   ",
"FILE NAME:        dan.kb                                                  ",
"ORIGINAL DATE :   01 May 1994.                                            ",
"PURPOSE:          Ergonomics Design - Incorporting EDKBES in SAMMIE       ",
"                  system for Interior Vehicle Design                      ",
"                  running in SUN SPARC workstation on window.             ",
"                                                                          ",
"**************************************************************************",
"                                                                          ",
"    The Wealth of Legislation on Standard and Legislation(s).            ",
"                                                                          ",
"    The Rule and Regulation the Interior of a Car(s).                     ",
"                                                                          ",
"    The Information on Ergonomics Design of Standards & Specification.   ",
"                                                                          ",
"    The Design Working Practice on Aspect of Design Process.             ",
"                                                                          ",
```

```
banner:
"******************************************************************************".
determination:
      "remember to write something here:                              ".
justification:
      "The justification for this knowledge base is:                  ".
rule:
      "to see an explanation of a rule, type:                         ",
      "display attach explanation",
      "remember to check this out", "                                 ".
restart:
      "Type 'c' to continue, or 's' to stop.                          ".
KBA:
"Knowledge Engineering System:Manual and Reference Guide.             ",
"KES Knowledge Base Author's Manual 1986.                             ".
ADR:
"Australian Design Rules: Standards Rules and Regulations.            ",
"Australian Design Rule 18/00 for Instrumentation 1988.               ".
BSI:
"British Standard Institution: Standards & Informations for           ",
"          Automobile Series.                                         ",
"BS AU 143c_84: Symbols for controls, indicators and tell_tales       ",
"          for road vehicles.                                         ",
"BS AU 176_80: Establishment of eyellipses for driver's eye           ",
"          location.                                                  ",
"BS AU 179_81: Dimensional codes for passenger cars.                  ",
"BS AU 183_83: Passive seat belt systems.                             ",
"BS AU 199_84: Location of hand controls,indicators and tell_tales.   ".
CEN:
"European Committee for Standardzation: Standards & Informations.     ",
"CEN/TC 122_92:Basic list of definitions of human body dimensions     ",
"          for technical design.                                      ".
EDS:
"EDS: Ergonomics of Display Systems for Austin Rover Ltd.             ",
"The standards ergonomics of display system for Austin Rover.         ".
EEC:
"INTEREUROPE: Specifications for Motor Vehicle Identification         ",
"          of Controls/Warning Lights 1979.                           ",
"EEC: European Economy Community Directives.                          ",
"Directive 71/127/EEC (amended 79/795/EEC, 82/205/EEC, 88/321/EEC):   ",
"Rear-view mirrors of motor vehicles.                                 ",
"Directive 77/649/EEC (amended 81/643/EEC, 88/366/EEC):               ",
"Field of vision of motor vehicle drivers.                            ",
"Directive 78/317/EEC: Windscreen wiper and washer systems.           ",
"Directive 78/317/EEC: Defrosting and demisting systems.              ".
FMVSS:
"FMVSS: Federal Motor Vehicle Safety Standard in USA                  ",
"FMVSS 103: Defrost/demist system.                                    ",
"FMVSS 104: Wash/wipe system.                                         ",
"FMVSS 111: Rear view mirrors.                                        ".
```

```
ISO:

"International Standards Organisation: Standards and Informations       ",
"            for Automobile Series.                                     ",
"ISO 2575_82: Symbols for controls,indicators and tell_tales.           ",
"ISO 3958_77: Driver hand control reach_Passenger cars.                 ",
"ISO 4040_83: Location of hand controls, indicators and tell_tales.     ",
"ISO 4513_78: Visibility_Method for establishment of eyellipses for     ",
"            driver's eye location.                                     ",
"ISO 6385_81: Ergonomics principles in the design of work systems.      ".

SAE:

"System Automotives Engineer: Handbook,Standards & Informations         ",
"            for Automatives.                                           ",
"The SAE Handbook 1985.                                                 ",
"SAE J287_Jun 88: Driver hand control reach.                            ",
"SAE J879b_68: Motor vehicle seating system.                            ",
"SAE J941_Oct 85: Motor vehicle driver's range.                         ",
"SAE J1050_77: Describing and measuring the driver's field of view.     ",
"SAE J1138_77: Design criteria_Driver hand controls location for        ",
"            passenger cars.                                            ",
"SAE J1139_77: Supplemental information_Driver Hand control             ",
"            location for passenger cars.                               ".

%
\*********************************************************************

text:

\The following text is used for references for the knowledge base.
{References:
            " ",
            KBA,    " ",
            ADR,    " ",
            BSI,    " ",
            CEN,    " ",
            EDS,    " ",
            EEC,    " ",
            FMVSS,  "",
            ISO,    " ",
            SAE}

%
\*********************************************************************

attributes:

\The following are input data attributes to the Knowledge Base.

main menu: sgl
     (See a list_of displays item developed in_the knowledge base,
      See a list_of some usefull information for design,
      Design a particular interior displays item,
      Check the design for styling,
      Check the type for instruments,
      Check the colour for instruments,
      Quit)
     {question: "What would you like to do?", " "}.

info menu: sgl
     (International data on anthropometric for vehicle design,
```

```
        Requirements for the seat system,
        Requirements for the dashboard,
        Requirements for the primary displays,
        Requirements for the display information,
        Requirements for the display_layout,
        Requirements for the display instruments
        Requirements for the visibility of_display,
        Requirements for the visibility of_steering wheel,
        Requirements for the SRS Airbag system,
        Requirements for the mirror view,
        Requirements for the leg_reach heel_point of_pedals,
        Had enough information)
       {question: "Which do you required?", " "}.

interior displays menu: sgl
      (Seats,
       Dashboard,
       Primary Controls,
       Mirrors
       New case,
       Leave this menu)
      {question: "Which interior displays would you like to design?","",
              "It would be a good idea to go through each",
              "stage systematically.", " "}.

mirror view menu: sgl
     (Interior Mirror,
      Exterior Mirror,
      New case,
      Leave this menu)
      {question: "Which mirror view would you like to design?","",
           "It would be a good idea to go through each",
            "stage systematically.", " "}.

seats menu: sgl
     (Driver Seat,
      Front Seat,
      Rear Seat,
      New case,
      Leave this menu)
      {question: "Which seats would you like to design?","",
           "It would be a good idea to go through each",
           "stage systematically.", " "}.

dashboard menu: sgl
     (Primary Displays,
      Secondary Controls,
      New case,
      Leave this menu)
      {question: "Which interior front would you like to design?","",
           "It would be a good idea to go through each",
           "stage systematically.", " "}.

primary displays menu: sgl
     (Speedometer,
      Tachometer,
      Fuel,
      Temperature,
```

Door_open,
Right Indicators, SRSD Airbag,
Left Indicators, Seat Belt,
New case,
Leave this menu)

{question: "Which primary displays would you like to design?","",
"It would be a good idea to go through each",
"stage systematically.", " "}.

speedometer menu: sgl
(Check visibility through the steering wheel,
Check eye movement within the ergonomics range,
Check that no head movement required for visibility,
Finished with this menu)

{question: "what would you like to start with?", " ",
"It would be a good idea to go through each",
"stage systematically.", " "}.

tachometer menu: sgl
(Check visibility through the steering wheel,
Check eye movement within the ergonomics range,
Check that no head movement required for visibility,
Finished with this menu)

{question: "what would you like to start with?", " ",
"It would be a good idea to go through each",
"stage systematically.", " "}.

fuel menu: sgl
(Check ergonomics visibility,
Check allowable eye movement,
Check allowable head movement,
end this menu)

{question: "what would you like to check?", " "}.

temperature menu: sgl
(Check ergonomics visibility,
Check allowable eye movement,
Check allowable head movement,
end this menu)

{question: "what would you like to check?", " "}.

door_open menu: sgl
(Check ergonomics visibility,
Check allowable eye movement,
Check allowable head movement,
end this menu)

{question: "what would you like to check?", " "}.

primary controls menu: sgl
(Pedals,
Hand Lever,
Steering Wheel,
New case,
Leave this menu)

{question: "Which primary controls would you like to design?","",
"It would be a good idea to go through each",
"stage systematically.", " "}.

```
pedals menu: sgl
      (Accelerator,
       Brake,
       Clutch,
       New case,
       Leave this menu)
      {question: "Which pedals would you like to design?","",
            "It would be a good idea to go through each",
            "stage systematically.", " "}.
accelerator menu: sgl
      (Check reachable through the seat adjustment,
       Check leg move within the ergonomics range,
       Check seat move required for reachable,
       Finished with this menu)
      {question: "what would you like to check?", " "}.
brake menu: sgl
      (Check reachable through the seat adjustment,
       Check leg move within the ergonomics range,
       Check seat move required for reachable,
       Finished with this menu)
      {question: "what would you like to check?", " "}.
clutch menu: sgl
      (Check ergonomics reachable,
       Check allowable leg move,
       Check allowable seat move,
       Finished with this menu)
      {question: "what would you like to check?", " "}.
      reason: sgl
      (yes,
       no)
      {question: "would you like to see the reason for this?"," "}.
%
\*********************************************************************
\The following are class sections represented in the form of
\the class hierarchies.
classes:
\-------------------------------------------------------------------------
\This class represents displays with objects of
\the class being number of visual indicators.
displays: [default: speedometer, odometer, trip_odometer, oil pressure,
           tachometer, economy_mpg, fuel, temperature, door_open,
           turn_signal, headlamp, brake_failure, service_car, heater,
           SRS_D airbag, fog_light, parking brake, hazard_warning,
           battery, SRS_P airbag, seat belt, panel]
attributes:
           shape: mlt
                 (rectangle, square, circle, triangle).
           type: sgl
                 (electromechanical, curvilinear, discrete,
                  alphanumeric, representational).
```

```
        colour: sgl
            (red, yellow, green, blue, white).
        statement: mlt
            (distance is to far,
             distance is to short,
             move forward,
             move backward,
             Ok,
             not Ok,
             acceptable angle ok,
             not acceptable angle).
        distance: real.
        angle: real.
        area: real.
        relative position: mlt
            (left_of, right_of, above, below).
        styling: mlt
            (good ergonomics,
             good design,
             good styling,
             popularity,
             balance,
             harmony,
             aesthetics).
%
endclass.
\---------------------------------------------------------------------------
\This class represents the attributes of the steering wheel.
    st_wheel: [default: tp, bp, rp, lp, cg]
        attributes:
            cox: real.
            coy: real.
            coz: real.
%
endclass.
\---------------------------------------------------------------------------
\This class is concerned with speedometer attributes.
    speedo: [default: tp, bp, rp, lp, cg]
\These are members of class speedo that define the geometric positions
\of some vertices on boundaries of the speedometer. Their values are
\obtained either from a comunication file or from the end user.
\These are: tp, an upper most point;
\     bp, a lowest point;
\     rp, a point on the far right side;
\     lp, a point on the far left side;
\     cg, an approximate center of gravity.
\     ag, an acceptable angle.
    attributes:
        cox: real.
        coy: real.
        coz: real.
```

\The above attributes describes co_ordinates of a class member.

visibility: sgl
(visible through st_wheel,
not visible through st_wheel).

\The above are inferred attributes checking visibility of the display
top adjustment: real.
bottom adjustment: real.
left adjustment: real.
right adjustment: real.
angle adjustment: real.

%
endclass.

\---

\This class represents pedals with objects of
\the class being number of reachable distance.

pedals: [default: accele, brake, clutch, controls]

attributes:

shape: mlt
(rectangle, square).

type: sgl
(hand controls, automatic, manual,
foot controls).

statement: mlt
(distance is to far,
distance is to short,
move forward,
move backward,
Ok,
not Ok,
acceptable angle ok,
not acceptable angle).

distance: real.
angle: real.
area: real.

relative position: mlt
(left_of, right_of, centre_of, above, below,
angle_of).

styling: mlt
(good ergonomics,
good design,
good styling,
popularity,
balance,
harmony,
aesthetics,
acceptable angle).

%
endclass.

\---

\This class represents the attributes of the seat adjustment.

seat_adjust: [default: tp, bp, rp, lp, cg, ag]

attributes:

cox: real.
coy: real.
coz: real.

%
endclass.

\---

\This class is concerned with accelerator attributes.

accele: [default: tp, bp, rp, lp, cg, ag]

\These are members of class accele that define the geometric positions
\of some vertices on boundaries of the accelerator. Their values are
\obtained either from a comunication file or from the end user.
\These are: tp, an upper most point;
\ bp, a lowest point;
\ rp, a point on the far right side;
\ lp, a point on the far left side;
\ cg, an approximate center of gravity;
\ ag, an acceptable angle.

attributes:

cox: real.
coy: real.
coz: real.

\The above attributes describes co_ordinates of a class member.

reachable: sgl
(reach through the seat adjustment,
not reach through the seat adjustment).

\The above are inferred attributes checking the reachable of the pedals.
top adjustment: real.
bottom adjustment: real.
left adjustment: real.
right adjustment: real.
angle adjustment: real.

%
endclass.

\---

\This class is concerned with a panel visible through
\the steering wheel from any eye point.
\Once co_ordinates of an eye point is given, together with
\distance of a point on the panel, using the steering wheel
\co-ordinates form the above class, this class then calculates
\the visible area on the panel through the steering wheel.

visible:
attributes:

cox: real.
coy: real.
coz: real.

top limit: real
[default: (st_wheel:tp>coz – visible:eye>coz)

```
            * (visible:panel>coy – visible:eye>coy)
            / (st_wheel:tp>coy – visible:eye>coy)
            + visible:eye>coz ].

            bottom limit: real
            [default: (st_wheel:bp>coz – visible:eye>coz)
            * (visible:panel>coy – visible:eye>coy)
            / (st_wheel:bp>coy – visible:eye>coy)
            + visible:eye>coz ].

            left limit: real
            [default: (st_wheel:lp>cox – visible:eye>cox)
            * (visible:panel>coy – visible:eye>coy)
            / (st_wheel:lp>coy – visible:eye>coy)
            + visible:eye>cox ].

            right limit: real
            [default: (st_wheel:rp>cox – visible:eye>cox)
            * (visible:panel>coy – visible:eye>coy)
            / (st_wheel:rp>coy – visible:eye>coy)
            + visible:eye>cox ].

\The following attributes are concerned with adjustments of displays.

            move speedo down: real
            [default: speedo:tp>coz
            – visible:panel>top limit ].

            move speedo up: real
            [default: speedo:bp>coz
            – visible:panel>bottom limit ].

            move speedo to right: real
            [default: speedo:lp>cox
            – visible:panel>left limit ].

            move speedo to left: real
            [default: speedo:rp>cox
            – visible:panel>right limit ].

%
endclass.
\-----------------------------------------------------------------------------

\The following class describes angles of view with eye movement,
\and head movement and compares them with the allowable limits.

      view_with: [default: eye move, head move]
attributes:

            horizontal left: int.
            horizontal right: int.
            vertical upwards: int.
            vertical downwards: int.
            horizontal left adjustment: int.
            horizontal right adjustment: int.
            vertical upwards adjustment: int.
            vertical downwards adjustment: int.

      groupI eye move horizontal left outside limit: int
      [default: view_with:eye move>horizontal left – 20].

      groupI eye move horizontal right outside limit: int
      [default: – view_with:eye move>horizontal right – 20].
```

groupI eye move vertical upwards outside limit: int
[default: view_with:eye move>vertical upwards – 0].

groupI eye move vertical downwards outside limit: int
[default: – view_with:eye move>vertical downwards – 35].

groupII eye move horizontal left outside limit: int
[default: view_with:eye move>horizontal left – 40].

groupII eye move horizontal right outside limit: int
[default: – view_with:eye move>horizontal right – 25].

groupII eye move vertical upwards outside limit: int
[default: view_with:eye move>vertical upwards – 0].

groupII eye move vertical downwards outside limit: int
[default: – view_with:eye move>vertical downwards – 35].

groupI head move horizontal left outside limit: int
[default: view_with:head move>horizontal left – 0].

groupI head move horizontal right outside limit: int
[default: – view_with:head move>horizontal right – 0].

groupI head move vertical upwards outside limit: int
[default: view_with:head move>vertical upwards – 0].

groupI head move vertical downwards outside limit: int
[default: – view_with:head move>vertical downwards – 0].

groupII head move horizontal left outside limit: int
[default: view_with:head move>horizontal left – 10].

groupII head move horizontal right outside limit: int
[default: – view_with:head move>horizontal right – 10].

groupII head move vertical upwards outside limit: int
[default: view_with:head move>vertical upwards – 0].

groupII head move vertical downwards outside limit: int
[default: – view_with:head move>vertical downwards – 10].

gI eye limit:sgl
(within gI eye limit, outside gI eye limit).

gII eye limit:sgl
(within gII eye limit, outside gII eye limit).

gI head limit:sgl
(within gI head limit, outside gI head limit).

gII head limit:sgl
(within gII head limit, outside gII head limit).

%
endclass.

\~~~~~~

\This class is concerned with a pedals reachable through
\the seat adjustment from SgRP or H-point.
\Once co_ordinates of an H-point is given, together with
\distance of a point on the pedal, using the seat adjustment
\co-ordinates form the above class, this class then calculates
\the reachable area on the pedal through the seat adjustment.

reach:
attributes:

```
        cox: real.
        coy: real.
        coz: real.
                top limit: real
                [default: (seat_adjust:tp>coz – reach:leg>coz)
                * (reach:pedals>coy – reach:leg>coy)
                / (seat_adjust:tp>coy – reach:leg>coy)
                + reach:leg>coz ].
                bottom limit: real
                [default: (seat_adjust:bp>coz – reach:leg>coz)
                * (reach:pedals>coy – reach:leg>coy)
                / (seat_adjust:bp>coy – reach:leg>coy)
                + reach:leg>coz ].
                left limit: real
                [default: (seat_adjust:lp>cox – reach:leg>cox)
                * (reach:pedals>coy – reach:leg>coy)
                / (seat_adjust:lp>coy – reach:leg>coy)
                + reach:leg>cox ].
                right limit: real
                [default: (seat_adjust:rp>cox – reach:leg>cox)
                * (reach:pedals>coy – reach:leg>coy)
                / (seat_adjust:rp>coy – reach:leg>coy)
                + reach:leg>cox ].
                angle limit: real
                [default: (seat_adjust:rp>coz – reach:leg>coz)
                * (reach:pedals>coy – reach:leg>coy)
                / (seat_adjust:rp>coy – reach:leg>coy)
                + reach:leg>coz ].
\The following attributes are concerned with adjustments of pedals.
                move accele down: real
                [default: accele:tp>coz
                – reach:pedals>top limit ].
                move accele up: real
                [default: accele:bp>coz
                – reach:pedals>bottom limit ].
                move accele to right: real
                [default: accele:lp>cox
                – reach:pedals>left limit ].
                move accele to left: real
                [default: accele:rp>cox
                – reach:pedals>right limit ].
                move accele to angle: real
                [default: accele:rp>coz
                – reach:pedals>angle limit ].
%
endclass.
\-----------------------------------------------------------------------------
\The following class describes distance and angles of reach with
\leg move, and seat move and compares them with the allowable limits.
        dist_with: [default: leg move, seat move]
attributes:
```

horizontal left: int.
horizontal right: int.
horizontal angles: int.
vertical upwards: int.
vertical downwards: int.
vertical angles: int.

horizontal left adjustment: int.
horizontal right adjustment: int.
horizontal angles adjustment: int.
vertical upwards adjustment: int.
vertical downwards adjustment: int.
vertical angles adjustment: int.

groupI leg move horizontal left outside limit: int
[default: dist_with:leg move>horizontal left – 20].

groupI leg move horizontal right outside limit: int
[default: – dist_with:leg move>horizontal right – 20].

groupI leg move vertical upwards outside limit: int
[default: dist_with:leg move>vertical upwards – 0].

groupI leg move vertical downwards outside limit: int
[default: – dist_with:leg move>vertical downwards – 35].

groupII leg move horizontal left outside limit: int
[default: dist_with:leg move>horizontal left – 40].

groupII leg move horizontal right outside limit: int
[default: – dist_with:leg move>horizontal right – 25].

groupII leg move vertical upwards outside limit: int
[default: dist_with:leg move>vertical upwards – 0].

groupII leg move vertical downwards outside limit: int
[default: – dist_with:leg move>vertical downwards – 35].

groupI seat move horizontal left outside limit: int
[default: dist_with:seat move>horizontal left – 0].

groupI seat move horizontal right outside limit: int
[default: – dist_with:seat move>horizontal right – 0].

groupI seat move vertical upwards outside limit: int
[default: dist_with:seat move>vertical upwards – 0].

groupI seat move vertical downwards outside limit: int
[default: – dist_with:seat move>vertical downwards – 0].

groupII seat move horizontal left outside limit: int
[default: dist_with:seat move>horizontal left – 10].

groupII seat move horizontal right outside limit: int
[default: – dist_with:seat move>horizontal right – 10].

groupII seat move vertical upwards outside limit: int
[default: dist_with:seat move>vertical upwards – 0].

groupII seat move vertical downwards outside limit: int
[default: – dist_with:seat move>vertical downwards – 10].

gI leg limit: sgl
(within gI leg limit, outside gI leg limit).

gII leg limit: sgl
(within gII leg limit, outside gII leg limit).

```
        gI seat limit: sgl
        (within gI seat limit, outside gI seat limit).

        gII seat limit: sgl
        (within gII seat limit, outside gII seat limit).

%
endclass.
\------------------------------------------------------------------------

\This class is for attributes of tachometer's members co_ordinates,
\similar to that of speedometer's.

        tacho:
attributes:

        cox: real.
        coy: real.
        coz: real.

        visibility: sgl

            (visible through st_wheel,
             not visible through st_wheel).

             top adjustment: real.
             bottom adjustment: real.
             left adjustment: real.
             right adjustment: real.

             upper boundary: real
             [default: tacho:tp>coz
                   - visible:panel>top limit + 10].

             lower boundary: real
             [default: tacho:bp>coz
                   - visible:panel>bottom limit + 10].

             left boundary: real
             [default: tacho:lp>cox
                   - visible:panel>left limit + 20].

             right boundary: real
             [default: tacho:rp>cox
                   - visible:panel>right limit + 20].

%
endclass.
\------------------------------------------------------------------------

\This class is for attributes of fuel's members co_ordinates,
\similar to that of speedometer's.

        fuel:
attributes:

        cox: real.
        coy: real.
        coz: real.

        visibility: sgl
            (within ergonomics limits,
             outside ergonomics limits).

             top adjustment: real.
             bottom adjustment: real.
```

left adjustment: real.
right adjustment: real.

upper boundary: real
[default: fuel:tp>coz
- visible:panel>top limit + 10].

lower boundary: real
[default: fuel:bp>coz
- visible:panel>bottom limit + 10].

left boundary: real
[default: fuel:lp>cox
- visible:panel>left limit + 20].

right boundary: real
[default: fuel:rp>cox
- visible:panel>right limit + 20].

%
endclass.

\---

\This class is for attributes of temperature's members co_ordinates,
\similar to that of speedometer's.

temperature:
attributes:

cox: real.
coy: real.
coz: real.

visibility: sgl
(within ergonomics limits,
outside ergonomics limits).

top adjustment: real.
bottom adjustment: real.
left adjustment: real.
right adjustment: real.

upper boundary: real
[default: temperature:tp>coz
- visible:panel>top limit + 10].

lower boundary: real
[default: temperature:bp>coz
- visible:panel>bottom limit + 10].

left boundary: real
[default: temperature:lp>cox
- visible:panel>left limit + 20].

right boundary: real
[default: temperature:rp>cox
- visible:panel>right limit + 20].

%
endclass.

\---

\This class is for attributes of door_open's members co_ordinates,
\similar to that of speedometer's.

```
        door_open:
  attributes:
        cox: real.
        coy: real.
        coz: real.
        visibility: sgl
             (within ergonomics limits,
              outside ergonomics limits).
              top adjustment: real.
              bottom adjustment: real.
              left adjustment: real.
              right adjustment: real.
              upper boundary: real
              [default: door_open:tp>coz
                   - visible:panel>top limit + 10].
              lower boundary: real
              [default: door_open:bp>coz
                   - visible:panel>bottom limit + 10].
              left boundary: real
              [default: door_open:lp>cox
                   - visible:panel>left limit + 20].
              right boundary: real
              [default: door_open:rp>cox
                   - visible:panel>right limit + 20].
%
endclass.
\------------------------------------------------------------------------
\This class is for attributes of brake's members co_ordinates,
\similar to that of accelerator's.
      brake:
attributes:
      cox: real.
      coy: real.
      coz: real.
      reachable: sgl
           (reach through the seat adjustment,
            not reach through the seat adjustment).
            top adjustment: real.
            bottom adjustment: real.
            left adjustment: real.
            right adjustment: real.
            angle adjustment: real.
            upper boundary: real
            [default: brake:tp>coz
                 - reach:pedals>top limit + 10].
            lower boundary: real
            [default: brake:bp>coz
                 - reach:pedals>bottom limit + 10].
            left boundary: real
            [default: brake:lp>cox
                 - reach:pedals>left limit + 20].
```

```
            right boundary: real
            [default: brake:rp>cox
                  - reach:pedals>right limit + 20].

            angle boundary: real
            [default: brake:rp>cox
                  - reach:pedals>angle limit + 20].

%
endclass.

\-----------------------------------------------------------------------

\This class is for attributes of clutch's members co_ordinates,
\similar to that of accelerator's.

      clutch:
attributes:

      cox: real.
      coy: real.
      coz: real.

      reachable: sgl
          (reach through the seat adjustment,
           not reach through the seat adjustment).

           top adjustment: real.
           bottom adjustment: real.
           left adjustment: real.
           right adjustment: real.
           angle adjustment: real.

           upper boundary: real
           [default: clutch:tp>coz
                 - reach:pedals>top limit + 10].

           lower boundary: real
           [default: clutch:bp>coz
                 - reach:pedals>bottom limit + 10].

           left boundary: real
           [default: clutch:lp>cox
                 - reach:pedals>left limit + 20].

           right boundary: real
           [default: clutch:rp>cox
                 - reach:pedals>right limit + 20].
                 angle boundary: real
           [default: clutch:rp>cox
                 - reach:pedals>angle limit + 20].

%
endclass.

\-----------------------------------------------------------------------
%
\**********************************************************************
\The following are production rule knowledge sources that
\dynamically manipulate the static knowledge representations
\in classes and attributes sections.

rules:

\-----------------------------------------------------------------------
```

```
\The following rules are used for design and styling purposes.
rule1:
    if
        displays:speedo>shape = circle | square and
        displays:tacho>shape = circle | square
    then
        displays:panel>styling
            = good design <1.0> | balance <1.0>.
    endif.
rule2:
    if
        displays:speedo>relative position = right_of and
        displays:tacho>relative position = left_of
    then
        displays:panel>styling
            = good ergonomics <0.5> | good styling <0.5>.
    endif.
rule3:
    d:displays
    if
        d>area lt 400 or
        d>area gt 200
    then
        d>styling = good ergonomics <0.1>.
    endif.
rule4:
    d:displays
    if
        d>area le 400 and
        d>area ge 200
    then
        d>styling = good design <1.0>.
    endif.
rule5:
    if
        pedals:accele>shape = rectangle and
        pedals:brake>shape = square and
        pedals:clutch>shape = square
    then
        pedals:controls>styling
            = good design <1.0> | balance <1.0>.
    endif.
rule6:
    if
        pedals:accele>relative position = right_of and
        pedals:brake>relative position = centre_of and
        pedals:clutch>relative position = left_of
    then
        pedals:controls>styling
            = good ergonomics <0.5> | good styling <0.5>.
    endif.
```

```
rule7:
    p:pedals
    if
        p>distance ge 20 and
        p>angle le 87
    then
        p>styling = good design <0.5> | acceptable angle <0.5>.
    endif.
rule8:
    p:pedals
    if
        p>distance gt 50 and
        p>angle lt 87
    then
        p>styling = good design <0.5> | acceptable angle <0.5>.
    endif.
rule9:
    if
        pedals:clutch>relative position = right_of and
        pedals:clutch>relative position = left_of and
        pedals:clutch>relative position = angle_of
    then
        pedals:controls>styling
        = good design <0.5> | acceptable angle <0.5>.
    endif.
rule10:
    p:pedals
    if
        p>distance gt 800 or
        p>distance lt 400
    then
        p>styling = good ergonomics <0.1>.
    endif.
rule11:
    p:pedals
    if
        p>distance le 800 and
        p>distance ge 400
    then
        p>styling = good design <0.7>.
    endif.
rule12:
    if
        pedals:>statement distance 800 = distance is to short
    then
        not OK.
    endif.
rule13:
    if
        pedals:>statement distance 400 = distance is to far
    then
        Ok.
    endif.
```

\\--

\The following rules determine the visibility of each vertex.

speedo visibility:
sp:speedo, vis:visible

```
if
    speedo:tp>coz lt visible:panel>top limit and
    speedo:bp>coz gt visible:panel>bottom limit and
    speedo:lp>cox lt visible:panel>left limit and
    speedo:rp>cox gt visible:panel>right limit
then
    speedo:area>visibility = visible through st_wheel.
endif
```

{references: ADR, SAE}
{explanation: "1) AUSTRALIAN DESIGN RULE 18/00 (1988) : In the case groupI ",
"'Visual Indicators', (speedometer is considered to be in groupI) the ",
"portion of the steering wheel rim including its supporting arms that is ",
"below a plane tangential to the top of the 95th Percentile Eye Ellipses' ",
"and the underside of the highest rim cross section cut normal to the rim, ",
"should not constitute an obstruction. ",
"2) SAE SP-576-84, SP-654-86 and SP-734-88, SAE J941-90, SAE J1052-90, ",
" BS AU 199-84, ISO 4040-83 Automotive Electronic Displays and ",
" Information Systems. Design principles for steering wheel visibility and locations. ",
" The display panel area shall be visible through the steering wheel without eye ",
" or head movement. Visibility through the steering wheel can be divided into ",
" three zones: ",
" i) Zone one is located on the right side of the display panel and ",
" typically contains the tachometer, economy-meter, fuel indicator, ",
" safety belt indicator, choke, battery condition indicator, hazard-warning, and ",
" parking brake. The display area required to indicate that a quarter or less of the ",
" maximum stored fuel is available shall be visible without head movement. ",
" Illuminated areas for tell-tales in zone one below at least 30x45x25mm ",
" rectangular of the illuminated area of the following tell-tales/indicator shall be ",
" visible without head movement; Parking Brake, Hazard-warning, ",
" Battery charging, SRS-P Airbag and Seat belt-warning. ",
" ii) Zone two is located on the centre of the displays panel and ",
" typically contains door-open indicators, turn-signals, and headlamp ",
" indicator – Upper/Lower beam. ",
" iii)Zone three is located on the left side of the display panel and ",
" typically contains; speedometer, odometer, trip-odometer, oil ",
" pressure gauge, temperature gauge, brake-failure indicator, service ",
" warning, heater and fog-light indicator. Temperature indicator the ",
" remaining parts of the display area shall be also visible with eye ",
" or head movement. "}.

speedo invisibility:
sp:speedo, vis:visible

```
if
    speedo:tp>coz gt visible:panel>top limit or
    speedo:bp>coz lt visible:panel>bottom limit or
    speedo:lp>cox gt visible:panel>left limit or
    speedo:rp>cox lt visible:panel>right limit
then
    speedo:area>visibility = not visible through st_wheel.
endif
```

```
{references: ADR}
{explanation: "1) AUSTRALIAN DESIGN RULE 18/00 (1988) : In the case groupI      ",
"'Visual Indicators', (speedometer is considered to be in groupI ) the portion of the steering  ",
"wheel rim including its supporting arms that is below a plane tangential to the top of  ",
"the 95th Percentile Eye Ellipses' and the underside of the highest rim cross section cut  ",
"normal to the rim, should not constitute an obstruction.  ",
"2) SAE SP-576-84, SP-654-86 and SP-734-88, SAE J941-90, SAE J1052-90,  ",
"  BS AU 199-84, ISO 4040-83 Automotive Electronic Displays and Information Systems.  ",
"  Design principles for steering wheel visibility and locations. The display panel area shall  ",
"  be visible through the steering wheel without eye or head movement. Visibility through  ",
"  the steering wheel can be divided into three zones, speedo in zone three:  ",
"  iii) Zone three is located on the left side of the display panel and typically contains;  ",
"     speedometer, odometer, trip-odometer, oil pressure gauge, temperature gauge,  ",
"     brake-failure indicator, service warning, heater and fog-light indicator. Temperature  ",
"     indicator the remaining parts of the display area  shall be also visible with eye  ",
"     or head movement.  ",
"The speedometer should be positioned fully seen by all drivers, without  ",
"eye movements or head movements.  ",
"i) Analogue speedometer - The standards circular or semi-circular shape is easier to read.  ",
"  Standard scale markings use a conventional progression system of 10, 20, 30, put detail  ",
"  scale markings at 0, 5, 10, 15, 20, and where appropriate use minor markers for individual  ",
"  numbers. Needle pointers of whatever form must line up with scale markings and position  ",
"  on the scale to be easily read and not distract  the driver. The full scale should be available  ",
"  to the driver with the value being indicated by a pointer.  "}.

speedo correcting tp:
sp:speedo, vis:visible

    if
        speedo:area>visibility
            = not visible through st_wheel and
        speedo:tp>coz gt visible:panel>top limit
    then
        sp>top adjustment = vis>move speedo down.
    endif.

speedo correcting bp:
sp:speedo, vis:visible

    if
        speedo:area>visibility
            = not visible through st_wheel and
        speedo:bp>coz lt visible:panel>bottom limit
    then
        sp>bottom adjustment = vis>move speedo up.
    endif.

speedo correcting lp:
sp:speedo, vis:visible

    if
        speedo:area>visibility
            = not visible through st_wheel and
        speedo:lp>cox gt visible:panel>left limit
    then
        sp>left adjustment = vis>move speedo to right.
    endif.

speedo correcting rp:
sp:speedo, vis:visible
```

```
    if
        speedo:area>visibility
            = not visible through st_wheel and
        speedo:rp>cox lt visible:panel>right limit
    then
        sp>right adjustment = vis>move speedo to left.
    endif.
```

\---

tacho visibility:
tc:tacho, vis:visible

```
    if
        tacho:tp>coz lt visible:panel>top limit and
        tacho:bp>coz gt visible:panel>bottom limit and
        tacho:lp>cox lt visible:panel>left limit and
        tacho:rp>cox gt visible:panel>right limit
    then
        tacho:area>visibility = visible through st_wheel.
    endif
```

{references: ADR, SAE}
{explanation: "1) AUSTRALIAN DESIGN RULE 18/00 (1988) : In the case groupI ",
"'Visual Indicators', (tachometer is considered to be in groupI) the portion of the ",
"steering wheel rim including its supporting arms that is below a plane tangential to the top ",
"of the 95th Percentile Eye Ellipses'and the underside of the highest rim cross section cut ",
"normal to the rim, should not constitute an obstruction. ",
"2) SAE SP-576-84, SP-654-86 and SP-734-88, SAE J941-90, SAE J1052-90, ",
" BS AU 199-84, ISO 4040-83 Automotive Electronic Displays and Information Systems. ",
" Design principles for steering wheel visibility and locations. The display panel area shall ",
" be visible through the steering wheel without eye or head movement. Visibility through ",
" the steering wheel can be divided into three zones, tachometer in zone one: ",
" i) Zone one is located on the right side of the display panel and typically contains the ",
" tachometer, economy-meter, fuel indicator, safety belt indicator, choke, battery condition ",
" indicator, hazard-warning, and parking brake. ",
"2) Tachometer - If a tachometer is provided then the display should be permanently in ",
" operation when the engine is running. The tachometer should be positioned to be fully ",
" seen by all drivers, without eye movements or head movements. ",
" i) Analogue Tachometer - The standard shape of circular or semi-circular is ",
" recommended for ease of reading. Standard scale markings use a conventional ",
" progression system of 1, 2, 3,and where appropriate minor markers are used for ",
" individual numbers. Needle pointers of whatever form must line up with scale markings ",
" and position on the scale to be easily read and not distract the driver. The full scale ",
" should be visible to the driver with the value indicated by a pointer. "}..

tacho invisibility:
tc:tacho, vis:visible

```
    if
        tacho:tp>coz gt visible:panel>top limit or
        tacho:bp>coz lt visible:panel>bottom limit or
        tacho:lp>cox gt visible:panel>left limit or
        tacho:rp>cox lt visible:panel>right limit
    then
        tacho:area>visibility = not visible through st_wheel.
    endif
```

{references: ADR, SAE}
{explanation: "1) AUSTRALIAN DESIGN RULE 18/00 (1988) : In the case groupI

"'Visual Indicators', (tachometer is considered to be in groupI) the portion of the steering ",
"wheel rim including its supporting arms that isbelow a plane tangential to the top of ",
"the 95th Percentile Eye Ellipses'and the underside of the highest rim cross section cut ",
"normal to the rim, should not constitute an obstruction. ",
"2) SAE SP-576-84, SP-654-86 and SP-734-88, SAE J941-90, SAE J1052-90, ",
" BS AU 199-84, ISO 4040-83 Automotive Electronic Displays and ",
" Information Systems. Design principles for steering wheel visibility and locations. ",
" The display panel area shall be visible through the steering wheel without eye or head ",
" movement. Visibility through the steering wheel can be divided into three zones, ",
" tachometer in zone one: ",
" i) Zone one is located on the right side of the display panel and typically contains ",
" the tachometer, economy-meter, fuel indicator, safety belt indicator, choke, battery ",
" condition indicator, hazard-warning, and parking brake. ",
"2) Tachometer – If a tachometer is provided then the display should be permanently in ",
" operation when the engine is running. The tachometer should be positioned to be fully ",
" seen by all drivers, without eye movements or head movements. ",
" i) Analogue Tachometer – The standard shape of circular or semi-circular is recommended ",
" for ease of reading. Standard scale markings use a conventional progression system ",
" of 1, 2, 3,and where appropriate minor markers are used for individual numbers. Needle ",
" pointers of whatever form must line up with scale markings and position on the scale ",
" to be easily read and not distract the driver. The full scale should be visible to the driver ",
" with the value indicated by a pointer. "}.

---

fuel visibility:
fu:fuel
if
fuel:tp>coz lt fuel:tp>upper boundary and
fuel:bp>coz ge fuel:bp>lower boundary and
fuel:lp>cox le fuel:lp>left boundary and
fuel:rp>cox ge fuel:rp>right boundary
then
fuel:area>visibility = within ergonomics limits.
endif
{references: ADR, SAE}
{explanation: "1) AUSTRALIAN DESIGN RULE 18/00 (1988) : In the case groupII ",
"Visual Indicators', (fuel is considered to be in groupII) the steering wheel rim including ",
"its supporting arms and attachnements there should not constitute an obstruction. ",
"2) SAE SP-576-84, SP-654-86 and SP-734-88, SAE J941-90, SAE J1052-90, ",
" BS AU 199-84, ISO 4040-83 Automotive Electronic Displays and Information Systems. ",
" Design principles for steering wheel visibility and locations. The display panel area shall ",
" be visible through the steering wheel without eye or head movement. Visibility through ",
" the steering wheel can be divided into three zones, fuel in zone one: ",
" i) Zone one is located on the right side of the display panel and typically contains the ",
" tachometer, economy-meter, fuel indicator, safety belt indicator, choke, battery condition ",
" indicator, hazard-warning, and parking brake. The display area required to indicate that ",
" a quarter or less of the maximum stored fuel is available shall be visibility within ",
" ergonomics limits. ",
"3) Fuel Indicator – Fuel indicator should use an analogue display. With an analogue or ",
" electromechanical display are shown perfect fuel level Qualitative markings are required ",
" for fuel indicator. Further scale markings such as (E) for empty, 1/2 for half full and ",
" (F) for full. Low fuel level warning should be coloured red, orange or yellow. "}.

fuel invisibility:
fu:fuel
if
fuel:tp>coz gt fuel:tp>upper boundary or

fuel:bp>coz lt fuel:bp>lower boundary or
fuel:lp>cox gt fuel:lp>left boundary or
fuel:rp>cox lt fuel:rp>right boundary
then
fuel:area>visibility = outside ergonomics limits.
endif

{references: ADR, SAE}
{explanation: "1) AUSTRALIAN DESIGN RULE 18/00 (1988) : In the case groupII ",
"Visual Indicators', (fuel is considered to be in groupII) the steering wheel rim including its ",
"supporting arms and attachnements there should not constitute an obstruction. ",
"2) SAE SP-576-84, SP-654-86 and SP-734-88, SAE J941-90, SAE J1052-90, ",
" BS AU 199-84, ISO 4040-83 Automotive Electronic Displays and Information Systems. ",
" Design principles for " steering wheel visibility and locations. The display panel area shall ",
" be visible through the steering wheel without eye or head movement. Visibility through the ",
" steering wheel can be divided into three zones, fuel in zone one: ",
" i) Zone one is located on the right side of the display panel and typically contains the ",
" tachometer, economy-meter, fuel indicator, safety belt indicator, choke, battery ",
" condition indicator, hazard-warning, and parking brake. The display area required to ",
" indicate that a quarter or less of the maximum stored fuel is available shall be visibility ",
" within ergonomics limits. ",
"3) Fuel Indicator- Fuel indicator should use an analogue display. With an analogue or ",
" electromechanical display are shown perfect fuel level. Qualitative markings are required ",
" for fuel indicator. Further scale markings such as (E) for empty, 1/2 for half full and ",
" (F) for full. Low fuel level warning should be coloured red, orange or yellow. "}.

---

temperature visibility:
te:temperature

if
temperature:tp>coz lt temperature:tp>upper boundary and
temperature:bp>coz ge temperature:bp>lower boundary and
temperature:lp>cox le temperature:lp>left boundary and
temperature:rp>cox ge temperature:rp>right boundary
then
temperature:area>visibility = within ergonomics limits.
endif

{references: ADR, SAE}
{explanation: "1) AUSTRALIAN DESIGN RULE 18/00 (1988) : In the case groupII ",
"Visual Indicators', (temperature is considered to be in groupII) the steering wheel rim ",
"including its supporting arms and attachnements there should not constitute an obstruction. ",
"2) SAE SP-576-84, SP-654-86 and SP-734-88, SAE J941-90, SAE J1052-90, ",
" BS AU 199-84, ISO 4040-83 Automotive Electronic Displays and Information Systems. ",
" Design principles for steering wheel visibility and locations. The display panel area shall ",
" be visible through the steering wheel without eye or head movement. Visibility through the ",
" steering wheel can be divided into three zones, temp in zone three: ",
" iii)Zone three is located on the left side of the display panel and typically contains; ",
" speedometer, odometer, trip-odometer, oil pressure gauge, temperature gauge, ",
" brake-failure indicator, service warning, heater and fog-light indicator. Temperature ",
" indicator the remaining parts of the display area shall be also visibility within ",
" ergonomics limits. ",
"3) Temperature Indicator - Temperature indicator should use an analogue display. With an ",
" analogue or electromechanical display are shown perfect temperature indicator. ",
" Qualitative markings are required for temperature indicator. Further scale markings such ",
" as (H) for high, (M) for medium and (L) for low. High temperature indicator warning ",
" should be coloured red, orange or yellow. "}.

temperature invisibility:
te:temperature

```
    if
        temperature:tp>coz gt temperature:tp>upper boundary or
        temperature:bp>coz lt temperature:bp>lower boundary or
        temperature:lp>cox gt temperature:lp>left boundary or
        temperature:rp>cox lt temperature:rp>right boundary
    then
        temperature:area>visibility = outside ergonomics limits.
    endif
```

{references: ADR,SAE}
{explanation: "1) AUSTRALIAN DESIGN RULE 18/00 (1988) : In the case groupII ",
"Visual Indicators', (temperature is considered to be in groupII) the steering wheel rim ",
"including its supporting arms and attachnements there should not constitute an obstruction. ",
"2) SAE SP-576-84, SP-654-86 and SP-734-88, SAE J941-90, SAE J1052-90, ",
" BS AU 199-84, ISO 4040-83 Automotive Electronic Displays and Information Systems. ",
" Design principles for steering wheel visibility and locations. The display panel area shall ",
" be visible through the steering wheel without eye or head movement. Visibility through the ",
" steering wheel can be divided into three zones, temp in zone three: ",
" iii)Zone three is located on the left side of the display panel and typically contains; ",
" speedometer, odometer, trip-odometer, oil pressure gauge, temperature gauge, ",
" brake-failure indicator, service warning, heater and fog-light indicator. Temperature ",
" indicator the remaining parts of the display area shall be also visibility within ",
" ergonomics limits. ",
"3) Temperature Indicator – Temperature indicator should use an analogue display. With an ",
" analogue or electromechanical display are shown perfect temperature indicator. ",
" Qualitative markings are required for temperature indicator. Further scale markings such ",
" as (H) for high, (M) for medium and (L) for low. High temperature indicator warning ",
" should be coloured red, orange or yellow. "}.

---

door_open visibility:
dr:door_open

```
    if
        door_open:tp>coz lt door_open:tp>upper boundary and
        door_open:bp>coz ge door_open:bp>lower boundary and
        door_open:lp>cox le door_open:lp>left boundary and
        door_open:rp>cox ge door_open:rp>right boundary
    then
        door_open:area>visibility = within ergonomics limits.
    endif
```

{references: ADR, SAE}
{explanation: "1) AUSTRALIAN DESIGN RULE 18/00 (1988) : In the case groupII ",
"Visual Indicators', (temperature is considered to be in groupII) the steering wheel rim ",
"including its supporting arms and attachnements there should not constitute an obstruction. ",
"2) SAE SP-576-84, SP-654-86 and SP-734-88, SAE J941-90, SAE J1052-90, ",
" BS AU 199-84, ISO 4040-83 Automotive Electronic Displays and Information Systems. ",
" Design principles for steering wheel visibility and locations. The display panel area shall ",
" be visible through the steering wheel without eye or head movement. Visibility through ",
" the steering wheel can be divided into three zones, door-open in zone two: ",
" i) Zone two is located on the centre of the displays panel and typically contains ",
" door-open indicators, turn-signals, and headlamp indicator – Upper/Lower beam. "}.

door_open invisibility:
dr:door_open

if
door_open:tp>coz gt door_open:tp>upper boundary or
door_open:bp>coz lt door_open:bp>lower boundary or
door_open:lp>cox gt door_open:lp>left boundary or
door_open:rp>cox lt door_open:rp>right boundary
then
door_open:area>visibility = outside ergonomics limits.
endif

{references: ADR, SAE}
{references: ADR, SAE}
{explanation: "1) AUSTRALIAN DESIGN RULE 18/00 (1988) : In the case groupII ",
"Visual Indicators', (temperature is considered to be in groupII) the steering wheel rim ",
"including its supporting arms and attachnements there should not constitute an obstruction. ",
"2) SAE SP-576-84, SP-654-86 and SP-734-88, SAE J941-90, SAE J1052-90, ",
" BS AU 199-84, ISO 4040-83 Automotive Electronic Displays and Information Systems. ",
" Design principles for steering wheel visibility and locations. The display panel area shall ",
" be visible through the steering wheel without eye or head movement. Visibility through ",
" the steering wheel can be divided into three zones, door-open in zone two: ",
" i) Zone two is located on the centre of the displays panel and typically contains ",
" door-open indicators, turn-signals, and headlamp indicator – Upper/Lower beam. "}.

groupI eye move within limit:
v_w: view_with
if
view_with:eye move > horizontal left le 20 and
view_with:eye move > horizontal right ge -20 and
view_with:eye move > vertical upwards le 0 and
view_with:eye move > vertical downwards ge -35
then
v_w>gI eye limit = within gI eye limit.
endif

{references: ADR}
{explanation: "AUSTRALIAN DESIGN RULE 18/00 (1988), All 'Visual Indicators' ",
"specified as GroupI in Clause 18.2 shall be totally located between 2 vertical planes inclined ",
"at 20 deg. left and 20 deg. right of the longitudinal axis of the vehicle and passing through ",
"the for most points of the left and right 95th Percentile Eye Ellipses respectively. Such ",
"indicators shall be totally located above a plane inclined downwards at 35 deg. from the ",
"horizontal and including a horizontal transverse line through the for most points of each of ",
"the '95th Percentile Eye Ellipses' and below a plane tangential to the bottom of the 95th ",
"Percentile Eye Ellipses which includes a line at ground level transverse to the longitudinal ",
"axis of the vehicle 11m forward of the rear most eye ellips point. "}.

groupI eye move outside limit:
v_w: view_with
if
view_with:eye move > horizontal left gt 20 or
view_with:eye move > horizontal right lt -20 or
view_with:eye move > vertical upwards gt 0 or
view_with:eye move > vertical downwards lt -35
then
v_w>gI eye limit = outside gI eye limit.
endif

{references: ADR}
{explanation: "AUSTRALIAN DESIGN RULE 18/00 (1988), All 'Visual Indicators' ",

```
"specified as GroupI in Clause 18.2 shall be totally located between 2 vertical planes inclined    ",
"at 20 deg. left and 20 deg. right of the longitudinal axis of the vehicle and passing through      ",
"the for most points of the left and right 95th Percentile Eye Ellipses respectively. Such          ",
"indicators shall be totally located above a plane inclined downwards at 35 deg. from the           ",
"horizontal and including a horizontal transverse line through the for most points of each of      ",
"the '95th Percentile Eye Ellipses' and below a plane tangential to the bottom of the 95th          ",
"Percentile Eye Ellipses which includes a line at ground level transverse to the longitudinal       ",
"axis of the vehicle 11m forward of the rear most eye ellips point.                                 "].
```

groupI eye move horiz_left adjustment:
```
v_w: view_with
    if
        view_with:eye move>gI eye limit
            = outside gI eye limit and
        view_with:eye move>horizontal left gt 20
    then
        v_w>horizontal left adjustment =
        v_w>groupI eye move horizontal left outside limit.
    endif.
```

groupI eye move horiz_right adjustment:
```
v_w: view_with
    if
        view_with:eye move>gI eye limit
            = outside gI eye limit and
        view_with:eye move>horizontal right lt -20
    then
        v_w>horizontal right adjustment =
        v_w>groupI eye move horizontal right outside limit.
    endif.
```

groupI eye move vert_up adjustment:
```
v_w: view_with
    if
        view_with:eye move>gI eye limit
            = outside gI eye limit and
        view_with:eye move>vertical upwards gt 0
    then
        v_w>vertical upwards adjustment =
        v_w>groupI eye move vertical upwards outside limit.
    endif.
```

groupI eye move vert_down adjustment:
```
v_w: view_with
    if
        view_with:eye move>gI eye limit
            = outside gI eye limit and
        view_with:eye move>vertical downwards lt -35
    then
        v_w>vertical downwards adjustment =
        v_w>groupI eye move vertical downwards outside limit.
    endif.
```

\---

groupII eye move within limit:
```
v_w: view_with
```

```
    if
        view_with:eye move > horizontal left le 40 and
        view_with:eye move > horizontal right ge -25 and
        view_with:eye move > vertical upwards = 0 and
        view_with:eye move > vertical downwards ge -35
    then
        v_w>gII eye limit = within gII eye limit.
    endif
```

{references: ADR}
{explanation: "AUSTRALIAN DESIGN RULE 18/00 (1988), All 'Visual Indicators' ",
"specified as GroupII in Clause 18.2 shall be totally located between 2 vertical planes inclined ",
"at 40 deg. left and 25 deg. right of the longitudinal axis of the vehicle and passing through ",
"the for most points of the left and right 95th Percentile Eye Ellipses respectively. Such ",
"indicators shall be totally located above a plane inclined downwards at 35 deg. from the ",
"horizontal and including a horizontal transverse line through the for most points of each of ",
"the '95th Percentile Eye Ellipses' and below a plane tangential to the bottom of the 95th ",
"Percentile Eye Ellipses which includes a line at ground level transverse to the longitudinal ",
"axis of the vehicle 11m forward of the rear most eye ellips point. "}.

groupII eye move outside limit:
v_w: view_with

```
    if
        view_with:eye move > horizontal left gt 40 or
        view_with:eye move > horizontal right lt -25 or
        view_with:eye move > vertical upwards gt 0 or
        view_with:eye move > vertical downwards lt -35
    then
        v_w>gII eye limit = outside gII eye limit.
    endif
```

{references: ADR}
{explanation: "AUSTRALIAN DESIGN RULE 18/00 (1988), All 'Visual Indicators' ",
"specified as GroupII in Clause 18.2 shall be totally located between 2 vertical planes inclined ",
"at 40 deg. left and 25 deg. right of the longitudinal axis of the vehicle and passing through ",
"the for most points of the left and right 95th Percentile Eye Ellipses respectively. Such ",
"indicators shall be totally located above a plane inclined downwards at 35 deg. from the ",
"horizontal and including a horizontal transverse line through the for most points of each of ",
"the '95th Percentile Eye Ellipses' and below a plane tangential to the bottom of the 95th ",
"Percentile Eye Ellipses which includes a line at ground level transverse to the longitudinal ",
"axis of the vehicle 11m forward of the rear most eye ellips point. "}.

groupII eye move horiz_left adjustment:
v_w: view_with

```
    if
        view_with:eye move>gII eye limit
            = outside gII eye limit and
        view_with:eye move>horizontal left gt 40
    then
        v_w>horizontal left adjustment =
        v_w>groupII eye move horizontal left outside limit.
    endif.
```

groupII eye move horiz_right adjustment:
v_w: view_with

```
    if
        view_with:eye move>gII eye limit
            = outside gII eye limit and
```

```
        view_with:eye move>horizontal right lt -25
    then
        v_w>horizontal right adjustment =
        v_w>groupII eye move horizontal right outside limit.
    endif.
groupII eye move vert_up adjustment:
v_w: view_with
    if
        view_with:eye move>gII eye limit
            = outside gII eye limit and
        view_with:eye move>vertical upwards gt 0
    then
        v_w>vertical upwards adjustment =
        v_w>groupII eye move vertical upwards outside limit.
    endif.
groupII eye move vert_down adjustment:
v_w: view_with
    if
        view_with:eye move>gII eye limit
            = outside gII eye limit and
        view_with:eye move>vertical downwards lt -35
    then
        v_w>vertical downwards adjustment =
        v_w>groupII eye move vertical downwards outside limit.
    endif.
\
groupI head move within limit:
v_w: view_with
    if
        view_with:head move > horizontal left = 0 and
        view_with:head move > horizontal right = 0 and
        view_with:head move > vertical upwards = 0 and
        view_with:head move > vertical downwards = 0
    then
        v_w>gI head limit = within gI head limit.
    endif
    {references: ISO}
    {explanation: "The display area of speedometer shall    ",
    "    be visible  without head movement                  ",
    "    GroupI 'Visual Indicators' shall be                ",
    "    visible without head movement.                     "}.
groupI head move outside limit:
v_w: view_with
    if
        view_with:head move > horizontal left gt 0 or
        view_with:head move > horizontal right gt 0 or
        view_with:head move > vertical upwards gt 0 or
        view_with:head move > vertical downwards gt 0
    then
        v_w>gI head limit = outside gI head limit.
    endif
    {references: ISO}
    {explanation: "The display area of speedometer shall    ",
```

```
        "    be visible  without head movement          ",
        "    GroupI 'Visual Indicators' shall be        ",
        "    visible without head movement.             ").
```

groupI head move horiz_left adjustment:

```
v_w: view_with
    if
        view_with:head move>gI head limit
            = outside gI head limit and
        view_with:head move>horizontal left gt 0
    then
        v_w>horizontal left adjustment =
        v_w>groupI head move horizontal left outside limit.
    endif.
```

groupI head move horiz_right adjustment:

```
v_w: view_with
    if
        view_with:head move>gI head limit
            = outside gI head limit and
        view_with:head move>horizontal right gt 0
    then
        v_w>horizontal right adjustment =
        v_w>groupI head move horizontal right outside limit.
    endif.
```

groupI head move vert_up adjustment:

```
v_w: view_with
    if
        view_with:head move>gI head limit
            = outside gI head limit and
        view_with:head move>vertical upwards gt 0
    then
        v_w>vertical upwards adjustment =
        v_w>groupI head move vertical upwards outside limit.
    endif.
```

groupI head move vert_down adjustment:

```
v_w: view_with
    if
        view_with:head move>gI head limit
            = outside gI head limit and
        view_with:head move>vertical downwards gt 0
    then
        v_w>vertical downwards adjustment =
        v_w>groupI head move vertical downwards outside limit.
    endif.
```

---

groupII head move within limit:

```
v_w: view_with
    if
        view_with:head move > horizontal left le 15 and
        view_with:head move > horizontal right ge -15 and
        view_with:head move > vertical upwards = 0 and
        view_with:head move > vertical downwards ge -15
    then
```

```
        v_w>gII head limit = within gII head limit.
    endif
    {references: ISO}
    {explanation: "The identification and those parts of      ",
        "   the display area required to indicate              ",
        "   a critical condition shall be visible              ",
        "   without head movement.                             ",
        "   The remaining parts of the display                 ",
        "   shall also be 'visible'; for these,                ",
        "   head movement is permited.                         "}.

groupII head move outside limit:
v_w: view_with
    if
        view_with:head move > horizontal left gt 15 or
        view_with:head move > horizontal right lt -15 or
        view_with:head move > vertical upwards gt 0 or
        view_with:head move > vertical downwards lt -15
    then
        v_w>gII head limit = outside gII head limit.
    endif
    {references: ISO}
    {explanation: "The identification and those parts of      ",
        "   the display area required to indicate              ",
        "   a critical condition shall be visible              ",
        "   without head movement.                             ",
        "   The remaining parts of the display                 ",
        "   shall also be 'visible'; for these,                ",
        "   head movement is permited.                         "}.

groupII head move horiz_left adjustment:
v_w: view_with
    if
        view_with:head move>gII head limit
            = outside gII head limit and
        view_with:head move>horizontal left gt 15
    then
        v_w>horizontal left adjustment =
        v_w>groupII head move horizontal left outside limit.
    endif.

groupII head move horiz_right adjustment:
v_w: view_with
    if
        view_with:head move>gII head limit
            = outside gII head limit and
        view_with:head move>horizontal right lt -15
    then
        v_w>horizontal right adjustment =
        v_w>groupII head move horizontal right outside limit.
    endif.

groupII head move vert_up adjustment:
v_w: view_with
    if
        view_with:head move>gII head limit
```

```
            = outside gII head limit and
        view_with:head move>vertical upwards gt 0
    then
        v_w>vertical upwards adjustment =
        v_w>groupII head move vertical upwards outside limit.
    endif.
```

groupII head move vert_down adjustment:

```
v_w: view_with
    if
        view_with:head move>gII head limit
            = outside gII head limit and
        view_with:head move>vertical downwards lt -15
    then
        v_w>vertical downwards adjustment =
        v_w>groupII head move vertical downwards outside limit.
    endif.
```

\---

accele reachable:

```
acc:accele, rch:reach
    if
        accele:tp>coz lt reach:pedals>top limit and
        accele:bp>coz gt reach:pedals>bottom limit and
        accele:lp>cox lt reach:pedals>left limit and
        accele:rp>cox gt reach:pedals>right limit
    then
        accele:area>reachable = reach through the seat adjustment.
    endif
```

{references: ISO}
{explanation: "LEG REACH OF PEDALS ISO 3958-77 : In the ",
"case groupI, 'Leg_reach', (accelerator is considered to be in ",
"groupI) the following location for reach H-point of_pedal ",
" – ACCELERATOR PEDAL – ",
"The seating refrence point is terms of the R-point. ",
"The pivot centre of the torso line and thigh centreline ",
"of the two or three-dimensional H-point machine template with ",
"95% leg length used to describe vehicle seating geometry. ",
"The accelerator heel-point or H-point device with floor ",
"covering; the foot angle of the device is restricted to not ",
"less than 87 deg. Horizontal dimensional from the R-point ",
"to the driver H-point (Hx). The vertical dimensions from ",
"R-point to driver H-point (Hz). "}.

accele inreachable:

```
acc:accele, rch:reach
    if
        accele:tp>coz gt reach:pedals>top limit or
        accele:bp>coz lt reach:pedals>bottom limit or
        accele:lp>cox gt reach:pedals>left limit or
        accele:rp>cox lt reach:pedals>right limit
    then
        accele:area>reachable = not reach through the seat adjustment.
    endif
```

{references: ISO}
{explanation: "LEG REACH OF PEDALS ISO 3958-77 : In the ",

```
"case groupI, 'Leg_reach', (accelerator is considered to be in       ",
"groupI ) the following location for leg_reach H-point of_pedal       ",
"    -     ACCELERATOR PEDAL     -                                     ",
"The seating refrence point is terms of the R-point.                   ",
"The pivot centre of the torso line and thigh centreline               ",
"of the two or three-dimensional H-point machine template with         ",
"95% leg length used to describe vehicle seating geometry.            ",
"The accelerator heel-point or H-point device with floor               ",
"covering; the foot angle of the device is restricted to not           ",
"less than 87 deg. Horizontal dimensional from the R-point             ",
"to the driver H-point (Hx). The vertical dimensions from              ",
"R-point to driver H-point (Hz).                                       "}.
```

accele correcting tp:

```
acc:accele, rch:reach
    if
        accele:area>reachable =
        not reach through the seat adjustment and
        accele:tp>coz gt reach:pedals>top limit
    then
        acc>top adjustment = rch>move accele down.
    endif.
```

accele correcting bp:

```
acc:accele, rch:reach
    if
        accele:area>reachable =
        not reach through the seat adjustment and
        accele:bp>coz lt reach:pedals>bottom limit
    then
        acc>bottom adjustment = rch>move accele up.
    endif.
```

accele correcting lp:

```
acc:accele, rch:reach
    if
        accele:area>reachable =
        not reach through the seat adjustment and
        accele:lp>cox gt reach:pedals>left limit
    then
        acc>left adjustment = rch>move accele to right.
    endif.
```

accele correcting rp:

```
acc:accele, rch:reach
    if
        accele:area>reachable =
        not reach through the seat adjustment and
        accele:rp>cox lt reach:pedals>right limit
    then
        acc>right adjustment = rch>move accele to left.
    endif.
```

\---

brake reachable:

```
bk:brake, rch:reach
    if
        brake:tp>coz lt reach:pedals>top limit and
```

```
        brake:bp>coz gt reach:pedals>bottom limit and
        brake:lp>cox lt reach:pedals>left limit and
        brake:rp>cox gt reach:pedals>right limit
    then
        brake:area>reachable = reach through the seat adjustment.
    endif
{references: ISO}
{explanation: "ROAD VEHICLES PASSENGER CARS DRIVER HAND                ",
"CONTROL REACH, ISO 3958-77. Leg_reach H_point of the pedals:             ",
"In the case groupI, 'Leg_reach', (clutch is considered to be             ",
"in groupI). The following location for leg_reach H-point of clutch pedals. ",
"        -     BRAKE PEDAL     -                                          ",
"The seating refrence point is terms of the R-point. The pivot centre of the ",
"torso line and thigh centreline of the two or three-dimensional H-point  ",
"machine template with 95% leg length used to describe vehicle seating    ",
"geometry.The accelerator heel-point or H-point device with floor covering; ",
"the foot angle of the device is restricted to not less than 87 deg. Horizontal ",
"dimensional from the R-point to the driver H-point (Hx). The vertical    ",
"dimensions from R-point to driver H-point (Hz).                          ",
"The brake heel-point or H-point device height 3-10 inches                ",
"max., 2 inches max. travel with 15 lbs.max. force brake.                 ",
"The distance between other pedal 2 inches min. gap.                      "}.

brake inreachable:
bk:brake, rch:reach
    if
        brake:tp>coz gt reach:pedals>top limit or
        brake:bp>coz lt reach:pedals>bottom limit or
        brake:lp>cox gt reach:pedals>left limit or
        brake:rp>cox lt reach:pedals>right limit
    then
        brake:area>reachable = not reach through the seat adjustment.
    endif
{references: ISO}
{explanation: "ROAD VEHICLES PASSENGER CARS DRIVER HAND                ",
"CONTROL REACH, ISO 3958-77. Leg_reach H_point of the pedals:             ",
"In the case groupI, 'Leg_reach', (clutch is considered to be             ",
"in groupI). The following location for leg_reach H-point of clutch pedals. ",
"        -     BRAKE PEDAL     -                                          ",
"The seating refrence point is terms of the R-point. The pivot centre of the ",
"torso line and thigh centreline of the two or three-dimensional H-point  ",
"machine template with 95% leg length used to describe vehicle seating    ",
"geometry.The accelerator heel-point or H-point device with floor covering; ",
"the foot angle of the device is restricted to not less than 87 deg. Horizontal ",
"dimensional from the R-point to the driver H-point (Hx). The vertical    ",
"dimensions from R-point to driver H-point (Hz).                          ",
"The brake heel-point or H-point device height 3-10 inches                ",
"max., 2 inches max. travel with 15 lbs.max. force brake.                 ",
"The distance between other pedal 2 inches min. gap.                      "}.
\---------------------------------------------------------------------------

clutch reachable:
clu:clutch
    if
        clutch:tp>coz lt clutch:tp>upper boundary and
        clutch:bp>coz ge clutch:bp>lower boundary and
```

clutch:lp>cox le clutch:lp>left boundary and
clutch:rp>cox ge clutch:rp>right boundary
then
clutch:area>reachable = reach through the seat adjustment.
endif

{references: ISO}
{explanation: "ROAD VEHICLES PASSENGER CARS DRIVER HAND ",
"CONTROL REACH, ISO 3958-77. Leg_reach H_point of the pedals: ",
"In the case groupI, 'Leg_reach', (clutch is considered to be ",
"in groupI). The following location for leg_reach H-point of clutch pedals. ",
" - CLUTCH PEDAL - ",
"The seating refrence point is terms of the R-point. The pivot centre of the ",
"torso line and thigh centreline of the two or three-dimensional H-point ",
"machine template with 95% leg length used to describe vehicle seating ",
"geometry. The accelerator heel-point or H-point device with floor covering; ",
"the foot angle of the device is restricted to not less than 87 deg. Horizontal ",
"dimensional from the R-point to the driver H-point (Hx). The vertical ",
"dimensions from R-point to driver H-point (Hz). The clucth heel-point or ",
"H-point device height 3-10 inches max., 4 inches max. travel with ",
"80-90 lbs. max. force brake. The distance between other pedal ",
"2 inches min. gap. "}.

clutch inreachable:
clu:clutch

if
clutch:tp>coz gt clutch:tp>upper boundary or
clutch:bp>coz lt clutch:bp>lower boundary or
clutch:lp>cox gt clutch:lp>left boundary or
clutch:rp>cox lt clutch:rp>right boundary
then
clutch:area>reachable = not reach through the seat adjustment.
endif

{references: ISO}
{explanation: "ROAD VEHICLES PASSENGER CARS DRIVER HAND ",
"CONTROL REACH, ISO 3958-77. Leg_reach H_point of the pedals: ",
"In the case groupI, 'Leg_reach', (clutch is considered to be ",
"in groupI). The following location for leg_reach H-point of clutch pedals. ",
" - CLUTCH PEDAL - ",
"The seating refrence point is terms of the R-point. The pivot centre of the ",
"torso line and thigh centreline of the two or three-dimensional H-point ",
"machine template with 95% leg length used to describe vehicle seating ",
"geometry. The accelerator heel-point or H-point device with floor covering; ",
"the foot angle of the device is restricted to not less than 87 deg. Horizontal ",
"dimensional from the R-point to the driver H-point (Hx). The vertical ",
"dimensions from R-point to driver H-point (Hz). The clucth heel-point or ",
"H-point device height 3-10 inches max., 4 inches max. travel with ",
"80-90 lbs. max. force brake. The distance between other pedal ",
"2 inches min. gap. "}.

\~~~~~~

groupI leg_move within limit:
d_w: dist_with

if
dist_with:leg move > horizontal left le 20 and
dist_with:leg move > horizontal right ge -20 and
dist_with:leg move > vertical upwards le 0 and

```
        dist_with:leg move > vertical downwards ge -35
    then
        d_w>gI leg limit = within gI leg limit.
    endif

{references: ISO}
{explanation: "ROAD VEHICLES PASSENGER CARS DRIVER HAND                  ",
"CONTROL REACH, ISO 3958-77. Leg_reach H_point of the pedals.             ",
"The following location for leg_reach H-point of_pedal:                   ",
"   -    ACCELERATOR PEDAL    -                                           ",
"The seating refrence point is terms of the Reach -point. The pivot centre of ",
"the torso line and thigh centreline of the two or three-dimensional H-point ",
"machine template with 95% leg_reach used to describe the seating         ",
"geometry. The accelerator heel-point or H-point device with floor        ",
"covering; the foot angle of the device is restricted to not less than 87 deg. ",
"Horizontal dimensional from the R-point to the driver H-point (Hx). The  ",
"vertical dimensions from Reach -point to driver H-point (Hz).            ",
"   -    BRAKE PEDAL    -                                                 ",
"The brake heel-point or H-point device height 3-10 inches max., 2 inches ",
"max. travel with 15 lbs.max. force brake. The distance between other     ",
"pedal 2 inches min. gap.                                                 ",
"   -    CLUCTH PEDAL    -                                                ",
"The clucth heel-point or H-point device height 3-10 inches max.,         ",
"4 inches max. travel with 80-90 lbs.max. force brake.                    ",
"The distance between other pedal 2 inches min. gap.                      "}.

groupI leg move outside limit:
d_w: dist_with

    if
        dist_with:leg move > horizontal left gt 20 or
        dist_with:leg move > horizontal right lt -20 or
        dist_with:leg move > vertical upwards gt 0 or
        dist_with:leg move > vertical downwards lt -35
    then
        d_w>gI leg limit = outside gI leg limit.
    endif

{references: ISO}
{explanation: "ROAD VEHICLES PASSENGER CARS DRIVER HAND                  ",
"CONTROL REACH, ISO 3958-77. Leg_reach H_point of the pedals.             ",
"The following location for leg_reach H-point of_pedal:                   ",
"   -    ACCELERATOR PEDAL    -                                           ",
"The seating refrence point is terms of the Reach -point. The pivot centre of ",
"the torso line and thigh centreline of the two or three-dimensional H-point ",
"machine template with 95% leg_reach used to describe the seating         ",
"geometry. The accelerator heel-point or H-point device with floor        ",
"covering; the foot angle of the device is restricted to not less than 87 deg. ",
"Horizontal dimensional from the R-point to the driver H-point (Hx). The  ",
"vertical dimensions from Reach -point to driver H-point (Hz).            ",
"   -    BRAKE PEDAL    -                                                 ",
"The brake heel-point or H-point device height 3-10 inches max., 2 inches ",
"max. travel with 15 lbs.max. force brake. The distance between other     ",
"pedal 2 inches min. gap.                                                 ",
"   -    CLUCTH PEDAL    -                                                ",
"The clucth heel-point or H-point device height 3-10 inches max.,         ",
"4 inches max. travel with 80-90 lbs.max. force brake.                    ",
"The distance between other pedal 2 inches min. gap.                      "}.
```

```
groupI leg move horiz_left adjustment:
d_w: dist_with
     if
          dist_with:leg move>gI leg limit
               = outside gI leg limit and
          dist_with:leg move>horizontal left gt 20
     then
          d_w>horizontal left adjustment =
          d_w>groupI leg move horizontal left outside limit.
     endif.
groupI leg move horiz_right adjustment:
d_w: dist_with
     if
          dist_with:leg move>gI leg limit
               = outside gI leg limit and
          dist_with:leg move>horizontal right lt -20
     then
          d_w>horizontal right adjustment =
          d_w>groupI leg move horizontal right outside limit.
     endif.
groupI leg move vert_up adjustment:
d_w: dist_with
     if
          dist_with:leg move>gI leg limit
               = outside gI leg limit and
          dist_with:leg move>vertical upwards gt 0
     then
          d_w>vertical upwards adjustment =
          d_w>groupI leg move vertical upwards outside limit.
     endif.
groupI leg move vert_down adjustment:
d_w: dist_with
     if
          dist_with:leg move>gI leg limit
               = outside gI leg limit and
          dist_with:leg move>vertical downwards lt -35
     then
          d_w>vertical downwards adjustment =
          d_w>groupI leg move vertical downwards outside limit.
     endif.
\----------------------------------------------------------------------
groupII leg move within limit:
d_w: dist_with
     if
          dist_with:leg move > horizontal left le 40 and
          dist_with:leg move > horizontal right ge -25 and
          dist_with:leg move > vertical upwards = 0 and
          dist_with:leg move > vertical downwards ge -35
     then
          d_w>gII leg limit = within gII leg limit.
     endif
{references: ISO}
{explanation: "ROAD VEHICLES PASSENGER CARS DRIVER HAND      ",
```

```
"CONTROL REACH, ISO 3958-77. Leg_reach H_point of the pedals.          ",
"The following location for leg_reach H-point of_pedal:                ",
"    -     ACCELERATOR PEDAL     -                                     ",
"The seating refrence point is terms of the R-point. The pivot centre of the  ",
"torso line and thigh centreline of the two or three-dimensional H-point     ",
"machine template with 95% leg length used to describe vehicle seating       ",
"geometry. The accelerator heel-point or H-point device with floor covering;  ",
"the foot angle of the device is restricted to not  less than 87 deg. Horizontal  ",
"dimensional from the R-point to the driver H-point (Hx). The vertical       ",
"dimensions from R-point to driver H-point (Hz).                      ",
"    -     BRAKE PEDAL     -                                           ",
"The brake heel-point or H-point device height 3-10 inches max.,       ",
"2 inches max. travel with 15 lbs.max. force brake.                    ",
"The distance between other pedal 2 inches min. gap.                   ",
"    -     CLUCTH PEDAL     -                                          ",
"The clucth heel-point or H-point device height 3-10 inches max.,      ",
"4 inches max. travel with 80-90 lbs.max. force brake.                 ",
The distance between other pedal 2 inches min. gap.                    "}.

groupII leg move outside limit:
d_w: dist_with
    if
        dist_with:leg move > horizontal left gt 40 or
        dist_with:leg move > horizontal right lt -25 or
        dist_with:leg move > vertical upwards gt 0 or
        dist_with:leg move > vertical downwards lt -35
    then
        d_w>gII leg limit = outside gII leg limit.
    endif
{references: ISO}
{explanation: "ROAD VEHICLES PASSENGER CARS DRIVER HAND              ",
"CONTROL REACH, ISO 3958-77. Leg_reach H_point of the pedals.          ",
"The following location for leg_reach H-point of_pedal:                ",
"    -     ACCELERATOR PEDAL     -                                     ",
"The seating refrence point is terms of the R-point. The pivot centre of the  ",
"torso line and thigh centreline of the two or three-dimensional H-point     ",
"machine template with 95% leg length used to describe vehicle seating       ",
"geometry. The accelerator heel-point or H-point device with floor covering;  ",
"the foot angle of the device is restricted to not  less than 87 deg. Horizontal  ",
"dimensional from the R-point to the driver H-point (Hx). The vertical       ",
"dimensions from R-point to driver H-point (Hz).                      ",
"    -     BRAKE PEDAL     -                                           ",
"The brake heel-point or H-point device height 3-10 inches max.,       ",
"2 inches max. travel with 15 lbs.max. force brake.                    ",
"The distance between other pedal 2 inches min. gap.                   ",
"    -     CLUCTH PEDAL     -                                          ",
"The clucth heel-point or H-point device height 3-10 inches max.,      ",
"4 inches max. travel with 80-90 lbs.max. force brake.                 ",
The distance between other pedal 2 inches min. gap.                    "}.

groupII leg move horiz_left adjustment:
d_w: dist_with
    if
        dist_with:leg move>gII leg limit
              = outside gII leg limit and
        dist_with:leg move>horizontal left gt 40
    then
```

```
        d_w>horizontal left adjustment =
        d_w>groupII leg move horizontal left outside limit.
    endif.
groupII leg move horiz_right adjustment:
d_w: dist_with
    if
        dist_with:leg move>gII leg limit
            = outside gII leg limit and
        dist_with:leg move>horizontal right lt -25
    then
        d_w>horizontal right adjustment =
        d_w>groupII leg move horizontal right outside limit.
    endif.
groupII leg move vert_up adjustment:
d_w: dist_with
    if
        dist_with:leg move>gII leg limit
            = outside gII leg limit and
        dist_with:leg move>vertical upwards gt 0
    then
        d_w>vertical upwards adjustment =
        d_w>groupII leg move vertical upwards outside limit.
    endif.
groupII leg move vert_down adjustment:
d_w: dist_with
    if
        dist_with:leg move>gII leg limit
            = outside gII leg limit and
        dist_with:leg move>vertical downwards lt -35
    then
        d_w>vertical downwards adjustment =
        d_w>groupII leg move vertical downwards outside limit.
    endif.
```

```
groupI seat move within limit:
d_w: dist_with
    if
        dist_with:seat move > horizontal left = 0 and
        dist_with:seat move > horizontal right = 0 and
        dist_with:seat move > vertical upwards = 0 and
        dist_with:seat move > vertical downwards = 0
    then
        d_w>gI seat limit = within gI seat limit.
    endif
{references: ISO}
{explanation: "The identification and those parts of    ",
    "   the pedals area required to indicate          ",
    "   a critical condition shall be leg_reach       ",
    "   without seat movement.                        ",
    "   The remaining parts of the pedals             ",
    "   shall also be 'reach'; for these,             ",
    "   seat movement is permited.                    "}.
```

```
groupI seat move outside limit:
d_w: dist_with

    if
        dist_with:seat move > horizontal left gt 0 or
        dist_with:seat move > horizontal right gt 0 or
        dist_with:seat move > vertical upwards gt 0 or
        dist_with:seat move > vertical downwards gt 0
    then
        d_w>gI seat limit = outside gI seat limit.
    endif

{references: ISO}
{explanation: "The identification and those parts of     ",
    "     the pedals area required to indicate        ",
    "     a critical condition shall be leg_reach     ",
    "     without seat movement.                      ",
    "     The remaining parts of the pedals           ",
    "     shall also be 'reach'; for these,           ",
    "     seat movement is permited.                  "}.

groupI seat move horiz_left adjustment:
d_w: dist_with

    if
        dist_with:seat move>gI seat limit
            = outside gI seat limit and
        dist_with:seat move>horizontal left gt 0
    then
        d_w>horizontal left adjustment =
        d_w>groupI seat move horizontal left outside limit.
    endif.

groupI seat move horiz_right adjustment:
d_w: dist_with

    if
        dist_with:seat move>gI seat limit
            = outside gI seat limit and
        dist_with:seat move>horizontal right gt 0
    then
        d_w>horizontal right adjustment =
        d_w>groupI seat move horizontal right outside limit.
    endif.

groupI seat move vert_up adjustment:
d_w: dist_with

    if
        dist_with:seat move>gI seat limit
            = outside gI seat limit and
        dist_with:seat move>vertical upwards gt 0
    then
        d_w>vertical upwards adjustment =
        d_w>groupI seat move vertical upwards outside limit.
    endif.

groupI seat move vert_down adjustment:
d_w: dist_with

    if
        dist_with:seat move>gI seat limit
```

```
            = outside gI seat limit and
        dist_with:seat move>vertical downwards gt 0
    then
        d_w>vertical downwards adjustment =
        d_w>groupI seat move vertical downwards outside limit.
    endif.
```

\---

groupII seat move within limit:

```
d_w: dist_with
    if
        dist_with:seat move > horizontal left le 15 and
        dist_with:seat move > horizontal right ge -15 and
        dist_with:seat move > vertical upwards = 0 and
        dist_with:seat move > vertical downwards ge -15
    then
        d_w>gII seat limit = within gII seat limit.
    endif
{references: ISO}
{explanation: "The identification and those parts of    ",
    "    the pedals area required to indicate          ",
    "    a critical condition shall be leg_reach       ",
    "    without seat movement.                        ",
    "    The remaining parts of the pedals             ",
    "    shall also be 'reach'; for these,             ",
    "    seat movement is permited.                    "}.
```

groupII seat move outside limit:

```
d_w: dist_with
    if
        dist_with:seat move > horizontal left gt 15 or
        dist_with:seat move > horizontal right lt -15 or
        dist_with:seat move > vertical upwards gt 0 or
        dist_with:seat move > vertical downwards lt -15
    then
        d_w>gII seat limit = outside gII seat limit.
    endif
{references: ISO}
{explanation: "The identification and those parts of    ",
    "    the pedals area required to indicate          ",
    "    a critical condition shall be leg_reach       ",
    "    without seat movement.                        ",
    "    The remaining parts of the pedals             ",
    "    shall also be 'reach'; for these,             ",
    "    seat movement is permited.                    "}.
```

groupII seat move horiz_left adjustment:

```
d_w: dist_with
    if
        dist_with:seat move>gII seat limit
            = outside gII seat limit and
        dist_with:seat move>horizontal left gt 15
    then
        d_w>horizontal left adjustment =
        d_w>groupII seat move horizontal left outside limit.
    endif.
```

```
groupII seat move horiz_right adjustment:
d_w: dist_with
    if
        dist_with:seat move>gII seat limit
            = outside gII seat limit and
        dist_with:seat move>horizontal right lt -15
    then
        d_w>horizontal right adjustment =
        d_w>groupII seat move horizontal right outside limit.
    endif.
groupII seat move vert_up adjustment:
d_w: dist_with
    if
        dist_with:seat move>gII seat limit
            = outside gII seat limit and
        dist_with:seat move>vertical upwards gt 0
    then
        d_w>vertical upwards adjustment =
        d_w>groupII seat move vertical upwards outside limit.
    endif.
groupII seat move vert_down adjustment:
d_w: dist_with
    if
        dist_with:seat move>gII seat limit
            = outside gII seat limit and
        dist_with:seat move>vertical downwards lt -15
    then
        d_w>vertical downwards adjustment =
        d_w>groupII seat move vertical downwards outside limit.
    endif.
\-----------------------------------------------------------------------
%
\**********************************************************************
\The following area control actions.
actions:
\-----------------------------------------------------------------------
    message " ",banner,
        welcome.
    message " ", banner.
    display attach References of kb.
    message " ", banner.
\-----------------------------------------------------------------------
    while main menu # Quit do
    if
        main menu = See a list_of displays item developed in_the knowledge base
    then
    erase.
    eraseclass.
message " ", banner,
    "At the moment knowledge base for only a few of              ",
```

```
       "these items have been completed.", "                                       ",
message"Speedometer:location for visibility through the st_wheel                   ",
       "eye movement within the ergonomics range                                   ",
       "and no head movement required for visibility", "                           ",
message "Tachometer:location for visibility through the st_wheel                   ",
       "eye movement within the ergonomics range                                   ",
       "and no head movement required for visibility","                            ",
message "Fuel_level: ergonomics visibility                                         ",
       "allowable eye movement                                                     ",
       "and allowable head movement          ","                                   ",
message "Engine_temperature: ergonomics visibility                                 ",
      "allowable eye movement                                                      ",
       "and allowable head movement          ","                                   ",
message "Door_open: ergonomics visibility                                          ",
      "allowable eye movement                                                      ",
       "and allowable head movement          ","                                   ",
message "Accelerator: location for leg_reach of driver                             ",
      "leg_reach within the ergonomics range                                       ",
       "and distance between other pedals    ","                                   ",
message "Brake: location for leg_reach of driver                                   ",
      "leg_reach within the ergonomics range                                       ",
       "and distance between other pedals    ","                                   ",
message "Clutch: location for leg_reach of driver                                  ",
      "leg_reach within the ergonomics range                                       ",
       "and distance between other pedals    ","                                   ",
message banner, "                                                                  ".
       message restart.
       break.
endif.
\""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
       if
              main menu = See a list_of some usefull information for design
       then
              erase.
              eraseclass.
\.............................................................................
       while info menu # Had enough information do
       if
              info menu = International data on anthropometric for vehicle design
       then
              erase.
              eraseclass.
       message " ", banner, " ",
"INTERNATIONAL DATA ON ANTHROPOMETRY - JURGENS 1990.  ","",
"Anthropometric classification of the world population into two          ",
"categories: smaller type and larger type (Jurgens 1990)",  "             ",
"Body measurement                          Smaller              Larger     ",
"                                          type                 type       ",
"                                          P5        P95/P5     P95        ",
"Stature                                   1390      1650       1910       ",
"Sitting height                             740       870       1000       ",
"Eye height, sitting                        620       750        880       ",
"Forward reach (fingertios)                 670       810        950       ",
"Shoulder breadth (bideltoid)               320       410        500       ",
"Shoulder breadth (biacromial)              285       360        430       ",
```

```
"Hip breadth (standing)                  260     335     410    ",
"Knee height                             405     505     600    ",
"Lower leg length (popliteal height)     320     410     505    ",
"Elbow-grip length                       270     340     410    ",
"Buttock-knee length                     450     360     670    ",
"Buttock-heel length                     830     1010    1190   ",
"Hip breadth (sitting)                   260     350     440    ",
"Hand length                             140     170     200    ",
"Hand breadth                             65      90     110    ",
"Foot length                             200     250     300    ",
"Head circumference                      475     540     600    ",
"Head length                             160     185     205    ",
"Head breadth                            120     145     170    ".
      message banner, " ".
      message restart.
      break.
endif.
\-------------------------------------------------------------------------

while info menu # Had enough information do
     if
          info menu = Requirements for the seat system
     then
          erase.
          eraseclass.
     message " ",
     banner, " ",

"Seat System are defined in ISO 3958-77, SAE J1100, J826, J1517. Seat can be divided into   ",
"two categories, performances or touring. ","                                               ",
"Performances seats - stiffer with more contour (bucket type) additional adjustable features:- ",
"such as lateral cushion and back bolsters.                                                 ",
"Touring seats tend to be more comfortable, softer and place greater emphasis on           ",
"comfort and safety. In designing the seats there is a need to suit styling to the vehicle  ",
"purpose and functions. The geometric features of the seat design can be divided           ",
"into accomodation and comfort requirements. The vehicle seat design, a task analysis       ",
"reveals three different occupants in the vehicle: driver seat, front-seat passenger, and  ",
"rear-seat passenger.","                                                                    ",
"     - GEOMETRIC FEATURES OF SEAT DESIGN -                                                 ",
"Seat design can be divided into accomodation and comfort:-                                 ",
"Accomodations - refers to seat size and adjustments for horizontal distance from controls, ",
"height and back angle.                                                                    ",
"Comfort - refers to stiffness, contour, climate, memory and vehicle features that promote ",
"occupant comfort.", "                                                                      ",
"     - DRIVER SEAT (SAE J1517) -                                                           ",
"Designing a driver's seat there are four design criteria:                                  ",
"1)  Driver seat position with unobstructed vision and within reach of all vehicle controls; ",
"2)  The seat must accommodate the driver's size, shape, weight;                            ",
"3)  The seat should be comfortable for extended periods;                                   ",
"4)  The seat should provide a safety zone for the driver in an accident or crash.          ",
"Seat comfort can be divided into two parts, attitude comfort is a measurement of human     ",
"posture of ,seat comfort due to occupant position in the vehicle. The foot, knee, hip and back ",
"angles provide an indication of relative position and or vehicle work space. Small hip and ",
"knee angles reflect restricted work space. The back angle provides an indication of seat back ",
"inclination, and foot angles an indication of toeboard or pedals angle relative to seat position. ",
"Seat comfort is the degree of support that a seat provides the occcupant.                  ",
"     - DRIVER SEAT DIMENSIONS -                                                            ",
```

```
"Driver's seat compartment dimensional ergonomics range                                   ",
"Items/Dimension in mm (inches)          BSI/ISO        SAE            Ref.no.            ",
"Vert. dist. SgRP to heel-pt.            127-130        177(7.0)       H30                ",
"Horiz. dist.SgRP to heel-pt.            130-508        917(36.1)      L53                ",
"Vert. seat adjt.rise                    0-38           38(1.5)        L59                ",
"Normal driver & seat adjt.              102-165        189(7.4)       L23                ",
"Design H-pt. front travel                              189(7.4)       L17                ",
"SgRP front X coordinate                                1494(58.8)     L31                ",
"Head room                                              938(36.9)      H61                ",
"Max. leg room (accel)                                  1094(43.1)     L34                ",
"Shoulder room                                          1348(53.1)     W3                 ",
"Hip room                                               1308(51.1)     W5                 ",
"Back angle-front                        5-40 deg.      25 deg.        L40                ",
"Hip angle                                              98 deg.        L42                ",
"Knee angle                                             134 deg.       L44                ",
"Foot angle                                             87 deg.        L46                ",
"Upper body opening to ground                           1207(47.5)     H50                ",
"                                                                                         ",
"    - PASSENGER SEAT DIMENSIONS -                                                        ",
"SgRP-point couple distance                             616(24.3)      L50                ",
"Head room                                              867(34.1)      H63                ",
"Min. leg room                                          688(27.1)      L51                ",
"SgRP (second to heel)                                  265(10.4)      H31                ",
"Knee clearance                                         -112(-4.4)     L48                ",
"Shoulder room                                          1299(51.1)     W4                 ",
"Hip room                                               1146(45.1)     W6                 ",
"Back angle                                             28,30 deg.     L41                ",
"Hip angle                                              82,45 deg.     L43                ",
"Knee angle                                             65,51 deg.     L45                ",
"Foot angle                                             106,48 deg.    L47                ",
"Depressed floor thickness                              25             H67                ".
message banner, " ".
    message restart.
    break.
endif.
\----------------------------------------------------------------------------------------

    while info menu # Had enough information do
    if
        info menu = Requirements for the dashboard
    then
        erase.
        eraseclass.
    message " ",
    banner, " ",

"Standards and legislation for dashboard are defined in ISO 4040, 1983, BS AU 199, 1984,  ",
"SAE SP-576, 1984. Dashboard are consists; primary displays, and secondary displays       ",
"The dashboard runs the entire width of the interior front of the vehicle. The standard   ",
"orientation is at an angle of 15 degrees away from the driver so that the the displays   ",
" may read with a minimum amount of distortion.                                           ",
"Primary displays is an instruments giving an essential information for driver's and      ",
"there are consists of speedometer (Odometer, Trip-odometer, and Oil pressure),           ",
"Tachometer (Economy-meter), fuel indicator, Temperature indicator, Door-open indicator,  ",
"Turn-signal indicator abd Headlamps (Main beam/flasher indicator).                       ",
"1)  Standards and legislation for the selection, design and layout of primary displays are ",
" intended to meet the objective of compatibility with the characteristics of human perception ",
```

```
"as follows:                                                                        ",
"  a) The nature and number of signals and displays design shall be compatible  with the   ",
"characteristics of the information.                                                ",
"  b) Clear identification of information where primary displays are numerous. The layout  ",
"     should be simple, spacious and arranged so as to promote clear and rapid orientation  ",
"  c) Primary displays should be designed for clear visibility and good visual perception.  ",
"     Account shall be taken, for instance, of the  intensity, shape, size and constrast.   ",
"  d) Rate and direction of change of display of information should be compatible with     ",
"     rate and direction of change of the primary source of that information.              ",
"  e) Function of the primary displays shall be identifiable to avoid confusion.           ",
"  f) Displays and control reach/movements, equipment response, and                        ",
"     display information should be mutually compatible.                                   ",
"  g) Where controls are numerous there is a need to ensure safe, unambiguous and quick     ",
"     operation. The displays and signals should be grouped them according to their functions. ",
"2)  ISO and BSI define verious zones for location of primary displays [ISO 4040, 1983,    ",
"    BS AU 199, 1984].                                                                     ",
"  i) Zone one is located on the right side of the display panel and should contain the    ",
"     tachometer, economy-mpg, fuel indicator, and five indicators at below rectangular in ",
"     shape dimensions 30x45x25 mm are parking-brake indicator, battery condition indicator, ",
"     hazard-warning indicator, supplementary restraint system (SRS-D) and seat-belt indicator.",
"  ii) Zone two  is located in the centre of the display panel and should contain the door-open, ",
"     turn-signal, and headlamp indicator - upper/lower beam. The remaining parts of       ",
"     the display area shall also be visible; with head movement is required.              ",
"  iii) Zone three is located on the left side of the display panel and  should contain the ",
"     speedometer, odometer, trip-odometer, oil  pressure meter, temperature gauge,  and   ",
"     five indicators at below rectangular in shape dimensions 30x45x25 mm                 ",
"     are brake-failure indicators,  service warning indicator, heater                     ",
"     controls indicator, supplementary restraint system (SRS-P) and fog-light indicator.  ".

    message banner, " ".
    message restart.
    break.
endif.
```

```
    while info menu # Had enough information do
    if
        info menu = Requirements for the primary displays
    then
        erase.
        eraseclass.
    message " ",
    banner, " ",
"Standards and legislation for primary displays are defined in SAE SP-576          ",
"Ergonomics Aspects of Electronic Instrumentation: A Guide for designers and        ",
"engineers [SAE SP-576, 1984, Galer, 1985].                                         ",
"Primary displays are intended to show information to the driver to confirm correct ",
"function during the  operation of controls.                                        ",
"1)  Analogue Displays - Analogue displays typically use a needle pointer on a scale to show ",
"    the value represented. Often it is used to convey qualitative information,  and can be ",
"    enhanced by a red portion of the scale to signify danger. Types of analogue displays   ",
"    include circular dials, linear scales and curvilinear combinations, and typical        ",
"    applications would be the tachometer, and fuel gauge. Analogue display are generally   ",
"    better than small digital equivalents for quick check reading, and for rate of change  ",
"    and direction information.                                                             ",
"2)  Discrete Displays - Discrete displays are also analogue displays but the markings of the ",
"    scale are discrete rather continuous. An example is an 8-segment or discrete sections fuel ",
```

```
"     gauge, providing quantity information but without the detail or accuracy of scalar displays.",
"3)   Digital Displays – With digital displays the information is presented directly as a number.  ",
"     A good example is odometer. Digital displays are better than analogue displays where        ",
"     precise readings and perfect indications are required.                                      ",
"4)   Alphanumeric Displays – Alphanumeric displays present information as                        ",
"     textual messages in full or abbreviated form e.g. FASTEN SEAT BELT.                         ",
"5)   Representational Displays – Representational displays present information as graphic       ",
"     diagram or working models, such as the plan drawing of the car used as a door-open          ",
"     indicators. The graphic diagrams enable the user to observe the function of items such as   ",
"     doors, bonnet and boot in relation to the whole, and to locate faults quickly, and can for  ",
"     example be used for vehicle diagnostics.                                                    ".

     message banner, " ".
     message restart.
     break.
endif.

\-----------------------------------------------------------------------------------------

     while info menu # Had enough information do

     if
          info menu = Requirements for the display information

     then
          erase.
          eraseclass.

     message " ",
     banner, " ",

"Standards and legislation for display information is specified in SAE SP-576                     ",
"Ergonomics Aspects of Electronic Instrumentation: A Guide for designer and                       ",
"engineers [SAE SP-576, 1984, Galer, 1985].                                                       ",
"Display information are intended to show information to the driver to be aware                  ",
"during the operation.                                                                            ",
"1)   Warning – Warning information is very important to the safe running of                      ",
"     the vehicle. Red is used as a – WARNING – indicator e.g.  and brake                         ",
"     failure or brake engaged signals use red lights.                                            ",
"  i) The driver's attention should be attracted to the warning; the                              ",
"     significance of the warning must be apparent through the red colour.                        ",
"2)   Advisory – Advisory information is very useful to the safe running of                       ",
"     the vehicle, and is also used to convey vehicle state information, e.g.                     ",
"     headlight main beam ON, FASTEN SEAT BELT.                                                   ",
"  i) The driver's attention should be attracted to the information but  it should distract       ",
"     him from the driving task. This covers a wide  range of  information devices                ",
"     from simple tell tales and indicators, to trip odometers/computers.                         ",
"3)   Diagnostic – Diagnostic information concerns the condition of the vehicle for               ",
"     maintenance purposes, e.g. warning light  for battery charging, and SERVICE indicator.      ",
"  i) The driver should be able to choose the appropriate opportunity to                          ",
"     assimilate or take action on such diagnostic                                                ",
"4)   Entertainment – In some vehicles  related information is availablè via                      ",
"     the entertainment facilities, e.g. traffic bulletins transmitted by radio.                  ",
"  i) Make sure that other audible forms of information presentation are                          ",
"     not masked by the entertainment system.                                                     ".

     message banner, " ".
     message restart.
     break.
endif.

\-----------------------------------------------------------------------------------------
```

```
while info menu # Had enough information do
if
     info menu = Requirements for the display_layout
then
     erase.
     eraseclass.
message " ", banner, " ",
"Standards and legislation for display panel layout are defined in SAE SP-576-84,     ",
"SP-654-86 and SP-734-88 Automotive Electronic Displays and Information Systems.     ",
"Design principles and standards for the layout of a display panel include.     ",
"1)  Visibility - The driver should be able to see all displays on the     ",
"    panel from normal driving position allowing for some head movement.     ",
"2)  Identification - It should be easy for the driver to find and identify displays,     ",
"3)  Grouping - The display instruments should be arranged in functional or sequential groups.",
"4)  Associations - Displays should be arranged to be compatible with the controls     ",
"    to which it is related.     ",
"    A)  Layout for good visibility     ",
"1)  The plane in which the display lies should be perpendicular to the line of the sight.     ",
"2)  The driver's view of displays should be unobstructed by the steering wheel,     ",
"    parts of his own body, and windscreen reflection.     ",
"3)  The distance between displays should be minimised to reduce eye and head movement.     ",
"    However, it is also useful to spatially separate displays to avoid confusion when reading     ",
"    them quickly.     ",
"    B)  Layout for good design and identification     ",
"1)  All displays should be clearly labelled.     ",
"2)  Location and separation - Good layout on the panel is one of the best aids to     ",
"    identification. Primary instruments, e.g. speedometer, tachometer and warning indicators     ",
"    should be located in primary display space.     ",
"3)  Functional grouping -  Displays should be grouped in terms of functional use.     ",
"    This reduces the area over which the driver has to search for a particulars display.     ",
"4)  Standardised location - If possible standardise the location of  displays or     ",
"    functional groups of displays.     ".
     message banner, " ".
     message restart.
     break.
endif.
\-----------------------------------------------------------------------------------------
     while info menu # Had enough information do
     if
               info menu = Requirements for the display instruments
     then
          erase.
          eraseclass.
     message " ",
      banner, " ",
"Standards and legislation for display instruments are defined in SAE SP-576-84,     ",
"SP-654-86 and SP-734-88 Automotive Electronic Displays and Information Systems.     ",
"Standards and design working practice for the display  include.     ",
"1)  Speedometer     ",
"  a) The speedometer should be positioned fully seen by all drivers, without     ",
"     eye movements or head movements.     ",
"  b) Digital displays are recognised as being faster than analogue displays for information     ",
"     acquisition.     ",
```

```
" c) A combination analogue-digital/electromechanical speedometer display provides good ",
"    ergonomics and functionality. ",
" d) Analogue displays are not as good as digital displays for rate of change information. ",
" e) Digital displays are read more quickly and accurately than analogue displays. ",
"2) Analogue speedometer - ",
" a) The standards circular or semi-circular shape is easier to read. ",
" b) Standard scale markings use a conventional progression system of 10, 20, 30, ",
"    put detail scale markings at 0, 5, 10, 15, 20, and where appropriate use minor ",
"    markers for individual numbers. ",
" c) Needle pointers of whatever form must line up with scale markings ",
"   and position on the scale to be easily read and not distract the driver. ",
" d) The full scale should be available to the driver with the value being indicated by a pointer.",
"3) Digital speedometer - ",
" a) Digital display values must remain visible long enough to be read accurately, ",
"    (500 - 1000 meters per seconds). ",
" b) Characters should be upright rather than slanted and their height should be 15 - 20 mm. ",
"1.1)Odometer, Trip-odometer, and Oil pressure meter ",
" a) This analogue-digital/electromechanical display is best placed within the speedometer ",
"    above the trip odometer to avoid clutter and confusion. ",
" b) When illuminated the odometer should be less bright than the speedometer. ",
" c) Trip Odometer analogue-digital display is best placed within the speedometer below the ",
"    odometer where it can be easily read by the driver. ",
" d) Odometer and Trip-odometer a square shape is recommended and not so large as to ",
"    interfere with speedometer readings (10 mm minimum character size). ",
" e) The trip odometer should be less bright than the speedometer and the control knob ",
"    should be easy to reach. ",
" f) Oil pressure meter indicator should use an analogue needle pointer with scale markings ",
"    and be positioned on the scale to be easily read and so as not to distract the driver. ",
"2) The Tachometer ",
" a) If a tachometer is provided then the display should be permanently in operation when ",
"    the engine is running. ",
" b) The tachometer should be positioned to be fully seen by all drivers, without eye or ",
"    head movements. ",
" c) Confusion between the tachometer and the speedometer can be avoided by ",
"    differentiating by styling, colour, relative brightness, etc. ",
" d) Maximum limits to engine speed should be indicated on the scale in red, yellow or orange.",
"2.1) Analogue Tachometer ",
" a) The standard shape of circular or semi-circular is recommended for ease of reading. ",
" b) Standard scale markings use a conventional progression system of 1, 2, 3,and where ",
"    appropriate minor markers are used for individual numbers. ",
" c) Needle pointers of whatever form must line up with scale markings and position on the ",
"    scale to be easily read and not distract the driver. The full scale should be visible to the ",
"    driver with the value indicated by a pointer. ",
"2.2) Digital speedometer ",
" a) Digital display values must remain visible long enough to be read accurately, ",
"    (500 - 1000 meter per seconds). ",
" b) The characters should be upright rather than slanted and their height should ",
"    be 15 - 20 mm. ",
"3) Fuel Indicator ",
" a) Fuel indicator should use an analogue or digital display. With an analogue or ",
"    electromechanical display are shown perfect fuel level. ",
" b) Qualitative markings are required for fuel indicator. Further scale markings such as ",
"    (E) for empty, 1/2 for half full and (F) for full. ",
" c) Low fuel level warning should be coloured red, orange or yellow. ".

    message banner, " ".
    message restart.
```

```
        break.
endif.

\---------------------------------------------------------------------------

        while info menu # Had enough information do
        if
            info menu = Requirements for the visibility of_display
        then
            erase.
            eraseclass.
        message " ",
        banner, " ",

"Standards and legislative requirements for location of visibility of displays - eyellipses for     ",
"driver eye location are defined in BS AU 176, 1980, ISO 4513, 1978, SAE J985, 1967.               ",
"The display area of the following display shall be visibility without head movement:-             ",
"1)  ZONE ONE: TACHOMETER - Economy_mpg, FUEL INDICATOR, SAFETY BELT,                               ",
"    CHOKE, BATTERY, HAZARD-WARNING and PARKING BRAKE.                                              ",
"    Zone one is located on the right side of the display panel. The display area required to      ",
"    indicate that a quarter or less of the maximum stored fuel is available shall be visible       ",
"    without head movement. The remaining parts of the display area shall also be visible;          ",
"    with eye movement and head movement.                                                           ",
"    - FUEL INDICATOR -                                                                             ",
"    Parts of the display area required to indicate a critical condition shall be visible           ",
"    without head movement for the following indicators:                                            ",
"    - ILLUMATED AREA TELL-TALES -                                                                  ",
"    Zone one below for tell-tales/indicator dimensions 18x18mm rectangular of the                  ",
"    illuminated area shall be visible without head movement; Parking Brake,                        ",
"    Hazard-warning, Battery charging, SRS-P Airbag and Seat belt-warning.                          ",
"                                                                                                   ",
"2)  ZONE TWO:      DOOR_OPEN, TURN_SIGNAL INDICATOR,                                               ",
"                   HEADLAMP Upper/Lower Beam                                                       ",
"    Zone two  is located in the centre of the display panel, the remaining parts of the display    ",
"     area shall also be visible; with head movement is required.                                   ",
"3)  ZONE THREE:  SPEEDOMETER - Odometer, Trip_Odometer, OIL PRESSURE                               ",
"    GAUGE, TEMPERATURE GAUGE, BRAKE-FAILURE INDICATORS,                                            ",
"    SERVICE WARNING, HEATER, SRS_D AIRBAG, and FOG-LIGHT INDICATOR.                                ",
"    Zone three is located on the left side of the display panel, the remaining parts of the       ",
"    display area shall also be visible with eye  movements and head movement;                      ",
"    - ILLUMATED AREA TELL-TALES -                                                                  ",
"    Zone three below for tell-tales/indicator dimensions 18x18 mm rectagular of the                ",
"    illumated area of the following tell-tales/indicator shall be visible without head             ",
"    movements; brake_failure, service_car, heater, SRS_D airbag, fog_light.                        ",
"Driver vision - visual aspect in vehicle design SAE J985-67 table shown  limits of                 ",
"visual field, eye movements and head movements.                                                    ",
"Distance visual eye and  head movement.                    Horizontal      Vertical                ",
"                                                           left/right      up/down                 ",
"Binocular field                                            120 deg.        0 deg.                  ",
"Right Monocular field                                      150 deg.        0 deg.                  ",
"Eye rotation min.                                          30/30 deg.      15/15 deg.              ",
"Eye rotation max.                                          45/65 deg.      45/65 deg.              ",
"Head turn min                                              45/60 deg.      45/60 deg.              ",
"Head turn max                                              45/60 deg.      45/60 deg.              ",
"Head tilt                                                  30/30 deg.      30/30 deg.              ".

        message banner, " ".
        message restart.
```

```
     break.
endif.
\-------------------------------------------------------------------------------

     while info menu # Had enough information do
     if
          info menu = Requirements for the visibility of_steering wheel
     then
          erase.
          eraseclass.
     message " ",
     banner, " ",
"Standards and Legislation for Steering Wheel Visibility and Location are defined in SAE     ",
"J941, 1990, SAE J1052, 1990, BS AU 199, 1984, ISO4040, 1983 and SAE J985, 1967.             ",
"Established standards and legislation, are described below:                                 ",
"1)  Eye-ellipses are perimeters of envelopes formed in side and plan view by an infinite    ",
"    number of planes dividing the eye position so that percentages of eye is one side of the ",
"    plane and 100 percentages are on the other side. The eye-ellipses and head contour      ",
"    located. A locus of points (in one degree increments) that is used to locate the        ",
"    (x-x and z-z) datum lines of the eye-ellipses and the driver head position contour in   ",
"    horizontally adjustable seat with seat back angles from 5 degrees to 40 degrees.        ",
"2)  The head contour locator line; with fixed seat, a locus of points is used to locate the ",
"    passenger head position contours in fixed seats with back angles from 5 degrees        ",
"    to 45 degrees.                                                                          ",
"3)  The standard viewing distance or visibility distance from the driver's eye to the primary ",
"    displays during driving will vary to some small extent due to eye or head movements.   ",
"    The standard viewing distance in cars is about 750 mm. For drivers wearing bifocal      ",
"    lenses, the reading lens is usually focused at 300mm with the distance lens at infinity. ",
"4)  The angle between the line of sight and a perpendicular to the primary display screen is ",
"    called the viewing angle or acceptable viewing angle. The acceptable viewing angle for  ",
"    legibility of displays is affected by ambient illumination, screen curvature, use of lenses, ",
"    contrast, resolution and character size. A relatively wide range of acceptable viewing ",
"    angles generally accepted in ergonomics practice: 15 degrees - comfortable viewing      ",
"    angle. 30 degrees - maximum  acceptable angle.                                          ",
"The following list established from standards and legislation, relates to the steering     ",
"wheel dimensions:                                                                           ",
"Item/Dimension in mm (inches)          BS/ISO         SAE            Ref.No.                ",
"Steering wheel dia.                    330-410        375(14.8)      W9                     ",
"Steering wheel weight                  25 - 35 lbs    25 - 40 lbs    W9                     ",
"Steering wheel angle                   10 - 70 deg.   20 deg.        H18                    ",
"Whl.cntr.to accel.H-pt.Horiz.          152-660        506(19.9)      L11                    ",
"Whl.cntr.to accel.H-pt.Vert.           530-838        548(21.6)      H17                    ",
"Steering wheel rim size 300mm to 460mm (women 300mm) dia.,                                  ",
"Resistance 5 to 30 pounds.                                                                  ",
"The steering wheel position angles at 10 to 70 deg. and displacement limited to about a 120 deg.",
"turn, no remove hands during turning.                                                       ".

     message banner, " ".
     message restart.
     break.
endif.
\-------------------------------------------------------------------------------

while info menu # Had enough information do
     if
               info menu = Requirements for the SRS Airbag system
     then
```

```
        erase.
        eraseclass.
    message " ",
    banner, " ",

"Standards and Legislation of Supplementary Restraint System (SRS) Airbag Systems as      ",
"references to Federal Motor Vehicle Safety Standard (FMVSS) and Ford DSE [FMVSS 208,     ",
"1990, Ford DSE, 1994].                                                                   ",
"Design principles for SRS airbag systems, safety features, specifications and design practice ",
"are describe below [Ford DSE, 1994]:                                                     ",
"1)  The SRS airbag systems, or so called Supplementary Restraint System is accommodated  ",
"    in the padded boss at centre of the steering wheel and driver's seat is provided with ",
"    anti-submarine ramps attached together with seat belt grabbers and pretensioners     ",
"    (Restraint System) to provide a new level of fast response safety that gives additional ",
"    protection to the driver's head, face and chest.                                     ",
"2)  The SRS airbag system for front passenger seat is accommodated in a padded boss at   ",
"    centre of the dashboard drawer and the seat is similarly provided with anti-submarine ",
"    ramps, seat belt grabbers and pretensioners.                                         ",
"    - GEOMETRIC FEATURES OF AIRBAG SYSTEM -                                              ",
"1)  All steering wheels should have safety protection with SRS airbag systems for the driver ",
"    and seats should have anti-submarine ramps with seat belt fastener, seat belt grabber ",
"    and pretensioners,                                                                   ",
"2)  The front passenger seat should have SRS airbag system on the dashboard drawer and   ",
"    the seat should have an anti-submarine ramp with seat belt fastener, seat belt grabber ",
"    and pretensioners,                                                                   ",
"3)  Rear passenger seats should have anti-submarine ramps with seat fastener, seat belt  ",
"    grabber and pretensioners,                                                           ",
"4)  The internal structure including all four doors should incorporate a cross-car beams and ",
"    a safety cell or cage.                                                               ",
"5)  The standard materials for airbags is neoprene lined textile with an internal coating of ",
"    silicone to protect the inner surface from the gas generator used to inflate the airbag. ",
"    Silicone materials are used as they are thin, and light than the neoprene and offer a much ",
"    improved packing density [Ford DSE, 1994].                                           ",
"6)  SRS-D AIRBAG should be stamped on the padded boss of the steering wheel and          ",
"    an SRS-D indicator light should be provided on the right of the instrument cluster.  ",
"7)  For the front passenger seat the lettering SRS-P AIRBAG is stamped on the dashboard  ",
"    drawer of the front passenger seat and an SRS-P indicator light should be provided on ",
"    the left of the instrument cluster.                                                  ",
"Design principles for SRS airbag systems, safety features, specifications and design     ",
"practice are describe below [Ford DSE, 1994]:                                            ",
"1)  The SRS airbag systems, or so called Supplementary Restraint System is accommodated  ",
"    in the padded boss at centre of the steering wheel and driver's seat is provided with ",
"    anti-submarine ramps attached together with seat belt grabbers and pretensioners     ",
"    (Restraint System) to provide a new level of fast response safety that gives additional ",
"    protection to the driver's head, face and chest.                                     ",
"2)  The SRS airbag system for front passenger seat is accommodated in a padded boss at   ",
"    centre of the dashboard drawer and the seat is similarly provided with anti-submarine ",
"    ramps, seat belt grabbers and pretensioners.                                         ",
"Design principles for SRS airbag systems described below:                                ",
"- FUNCTIONS OF AIRBAG SYSTEM -                                                           ",
"1)  Firstly, the operational readiness of the Restraint System (airbag) is indicated by the ",
"    SRS indicator light in the instrument cluster. If the key in the steering lock is turned to "
"    position 1 or 2, the indicator light stays on for approximately 4 seconds. Should an ",
"    impact at the steering wheel the airbag will fully inflate the moment a signal is received ",
"    from the sensor, make contact to protect the driver, then deflate as it absorbs the impact, ",
"    all in about 50 milliseconds. If the indicator light fails to come on when starting the car ",
"    or comes on while driving there is a fault in the system. The SRS airbag, however, is ",
```

```
"     not activated by this fault. Th airbag is so designed as to be activated only in severe     ",
"     head on collisions.The driver should have fastened his belt as otherwise the airbag        ",
"     cannot provide the envisaged protection.                                                    ",
"2)   Secondly during the activation of the SRS airbag a small volume of air will be released.   ",
"     Then the seat belt inertia reels are locked and in addition other sensor will have          ",
"     activated clamps which grab and hold the belts to minimise any paying out due to the        ",
"     effect of spooling. In a more severe impact the pretensioners work in conjunction          ",
"     with the grabbers to further enhance seat belt efficiency by pulling the belt  buckles      ",
"     downwards to reduce any slack in the diagonal and lap belts.                               ",
"3)   Front and rear seats have anti-submarine ramps so as to reduce sliding forward under        ",
"     the seat belt during impact. In the most severe impact the steering wheel airbag will      ",
"     inflate and deflate as it absorbs the impact. The air is neither injurious to health nor   ",
"     does it indicate a fire in the vehicle.                                                     ".

     message banner, " ".
     message restart.
     break.

endif.
\-----------------------------------------------------------------------------------------------

     while info menu # Had enough information do
     if
         info menu = Requirements for the mirrors
     then
           erase.
           eraseclass.
     message " ",
     banner, " ",

"Mirror - ISO 6549, SAE J941, J985-67, J826-87, J1050-77                                           ",
"Field of vision of motor vehicle drivers - Directive 77/649/EEC                                  ",
"Rear view mirror of motor vehicles - Directive 71/127/EEC                                        ",
"Windscreen wiper and washer systems - Directive 78/317/EEC                                       ",
"Rear view mirror - FMVSS 111 and Wash/wipe system - FMVSS 104","                                 ",
"_    DRIVER MIRROR VIEW        -                                                                  ",
"Driver mirror view standards, legislations and specifications:-                                  ",
"1)   Define the feild of mirror view provide for driver, establishing any                        ",
"     obstructions/obscurations caused by the body structure and fixtures such as                 ",
"     windsreen wipers or by the internal centre rear mirrors;                                    ",
"2)   Design suitable interior and exterior mirror, and to determine where they are               ",
"     best mounted to maximize the field of view while avoiding unnecessary obscuration           ",
"     of the direct view. The regulations for interior CR mirror (ICRM) should not further        ",
"     than 60 m or 200 ft to the rear of the driver's from rear of vehicle (EEC, FMVSS).          ",
"     The mirror must have sufficient width to give the driver 20 m or 20 deg. wide view          ",
"     horizontal angle view beyond these points (EEC, FMVSS).                                     ",
"     The regulation for exterior SR mirror  (ESRM) on the driver's side, required field          ",
"     extend rearward from vertical plane (ground level and horizon) 10 m or 35 ft.               ",
"     behind the driver's eyes (EEC, FMVSS) and is bounded by a plane through the                 ",
"     widest point on the side of the vehicle and a second plane 2.5 m or 8 ft. away from         ",
"     this (EEC, FMVSS). Driver visibility - visual aspect in interior vehicle design limits      ",
"     of visual field, eye movement and head movement are defined in SAE J985 1967.               ",
"Distance visual eye and  head movement.            Horizontal       Vertical                     ",
"                                                   left/right       up/down                      ",
"Binocular field                                    120 deg.         0 deg.                       ",
"Right Monocular field                              150 deg.         0 deg.                       ",
"Eye rotation min.                                  30/30 deg.       15/15 deg.                   ",
"Eye rotation max.                                  45/65 deg.       45/65 deg.                   ",
"Head turn min                                      45/60 deg.       45/60 deg.                   ",
```

```
"Head turn max                                   45/60 deg.      45/60 deg.        ",
"Head tilt                                       30/30 deg.      30/30 deg.        ".
      message banner, " ".
      message restart.
      break.
endif.
\------------------------------------------------------------------------------------
      while info menu # Had enough information do
      if
          info menu = Requirements for the leg_reach heel_point of_pedals
      then
           erase.
           eraseclass.
      message " ",
      banner, " ",
"Road Vehicles Passenger Cars Driver Hand Controls Reach, ISO 3958-77.               ",
"Leg_reach Heel_point of pedals reach distance.                                      ",
The following location for leg_reach Heel-point of_pedal: ", "                        ",
"    - ACCELERATOR PEDAL -                                                           ",
"The seating refrence point is terms of the R-point. The pivot centre of the torso line and thigh ",
"centreline of the two or three-dimensional Heel-point machine template with 95% leg ",
"length used to describe vehicle seating geometry. The accelerator heel-point or Heel-point ",
"device with floor covering; the foot angle of the device is restricted to not less than 87 deg. ",
"Horizontal dimensional from the R-point to the driver H-point (Hx).                 ",
"The vertical dimensions from R-point to driver H-point (Hz).",                    "",
"    - BRAKE PEDAL -                                                                 ",
"The brake heel-point or H-point device height 3-10 inches max., 2 inches max. travel with ",
"15 lbs.max. force brake. The distance between other pedal 2 inches min. gap.        ",
"    - CLUCTH PEDAL -                                                                ",
"The clucth heel-point or Heel-point device height 3-10 inches max., 4 inches max. travel ",
"with 80-90 lbs.max. force brake. The distance between other pedal 2 inches min. gap. ",
"The following list establishes the ranges of the operator workspace dimensions:    ",
"Item/Dimension in mm (inches)                        BS/ISO        Ref.No           ",
"Max. leg room (accel)                                1094          L34              ",
"Hip angle                                            98 deg.       L42              ",
"Knee angle                                           134 deg.      L44              ",
"Foot angle                                           87 deg.       L46              ",
"Back angle (B)                                       9/33 deg.     L40              ",
"Vertical R_point to H-point(Hz)                      130/520       H30              ",
"Horizontal R-point travel                            130           L53              ".
      message banner, " ".
      message restart.
      break.
endif.
\....................................................................................
      endwhile.
\------------------------------------------------------------------------------------
      if
           main menu = Design a particular interior displays item
      then
           erase.
           eraseclass.
\------------------------------------------------------------------------------------
```

```
while interior displays menu # Leave this menu do
if
      interior displays menu = New case
then
      erase.
      eraseclass.
endif.
\-----------------------------------------------------------------------
if
      interior displays menu = Dashboard
then
      erase.
      eraseclass.
\-----------------------------------------------------------------------
while dashboard menu # Leave this menu do
if
      dashboard menu = New case
then
      erase.
      eraseclass.
endif.
\-----------------------------------------------------------------------
if
      dashboard menu = Primary Displays
then
      erase.
      eraseclass.
\-----------------------------------------------------------------------
while primary displays menu # Leave this menu do
if
      primary displays menu = New case
then
      erase.
      eraseclass.
endif.
\-----------------------------------------------------------------------
if
      primary displays menu = Speedometer
then
      erase.
      eraseclass.
while speedometer menu # Finished with this menu do
\"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
if
      speedometer menu = Check visibility through the steering wheel
then
      erase.
      eraseclass.
read "spfile", speedo, speedo(cox, coy, coz).
read "spfile", st_wheel, st_wheel(cox, coy, coz).
read "spfile", visible, visible(cox, coy, coz).
```

```
        obtain speedo:area>visibility.
    if
        status (speedo:area>visibility) = known
    then message
        " ",
        banner,
        combine (" for values of vertices given : "),
        combine (" top vertex tp = ", speedo:tp>coz),
        combine (" bottom vertex bp = ", speedo:bp>coz),
        combine (" right vertex rp = ", speedo:lp>cox),
        combine (" left vertex lp = ", speedo:rp>cox),
        combine (" the speedometer is ",
                speedo:area>visibility),
        " ",
        banner,
        " ".
    endif.
\..........................................................................................................

    if reason = yes then
        message banner, " ".
        justify speedo:tp>visibility.
        message banner, " ".
    endif.

\..........................................................................................................
    if
        speedo:area>visibility = not visible through st_wheel
    then
        message " ",
        banner,
        combine(" the adjustment for speedometer ",
            " position is to : ").

    if speedo:tp>coz gt visible:panel>top limit
    then message " ",
        combine(" move down at least by a min ",
        visible:panel>move speedo down ," mm" ).
    endif.

    if speedo:bp>coz lt visible:panel>bottom limit
    then message " ",
        combine(" move up at least by a min ",
        visible:panel>move speedo up ," mm" ).
    endif.

    if speedo:lp>cox gt visible:panel>left limit
    then message " ",
        combine(" move right at least by a min ",
        visible:panel>move speedo to right ," mm" ).
    endif.

    if speedo:rp>cox lt visible:panel>right limit
    then message " ",
        combine(" move left at least by a min ",
        visible:panel>move speedo to left ," mm" ).
    endif.
    message banner,
    " ".
```

```
     write "spfile1", speedo, speedo(cox, coy, coz).
     write "spfile1", st_wheel, st_wheel(cox, coy, coz).
     write "spfile1", visible, visible(cox, coy, coz).
endif.
\.........................................................................................
          message restart.
          break.
endif.
\.........................................................................................
     if
          speedometer menu = Check eye movement
                              within the ergonomics range
     then
          erase.
          eraseclass.
     read "spefile", view_with, view_with(horizontal left,
          horizontal right, vertical upwards, vertical downwards).
          obtain view_with:eye move>gl eye limit.
     if
          status (view_with:eye move>gl eye limit) = known
     then message
          " ",
          banner,
          combine(" for angles of eye movement given:")," ",
          combine(" horizontal left =          ",
               view_with:eye move>horizontal left )," ",
          combine(" horizontal right =         ",
               view_with:eye move>horizontal right )," ",
          combine(" vertical upwards =         ",
               view_with:eye move>vertical upwards )," ",
          combine(" vertical downwards =       ",
               view_with:eye move>vertical downwards)," ",
          combine(" these angles of eye movement are: ",
               view_with:eye move>gl eye limit    )," ",
     banner,
     " ".
endif.
\.........................................................................................
     if reason = yes then
          message banner," ".
          justify view_with:eye move>gl eye limit.
          message banner," ".
endif.
\.........................................................................................
     if
          view_with:eye move>gl eye limit = outside gl eye limit
     then
          message
          " ",
          banner,
          combine(" You can adjust the position of the ",
               "speedometer so that it "),
```

```
        combine(" lies within the allowable ergonomics eye ",
              "movement by : ")," ".
    if view_with:eye move>horizontal left ge 20
    then message
        combine(" move to right to reduce angle of eye ",
        "movement by : ",view_with:eye move>groupI eye
         move horizontal left outside limit," deg" )," ".
    endif.
    if view_with:eye move>horizontal right le -20
    then message
        combine(" move to right to reduce angle of eye ",
        "movement by : ",view_with:eye move>groupI eye
         move horizontal right outside limit," deg" )," ".
    endif.
    if view_with:eye move>vertical upwards gt 0
    then message
        combine(" move down to reduce angle of eye ",
        "movement by : ",view_with:eye move>groupI eye
         move vertical upwards outside limit," deg" )," ".
    endif.
    if view_with:eye move>vertical downwards lt -35
    then message
        combine(" move up to reduce angle of eye ",
        "movement by : ",view_with:eye move>groupI eye
         move vertical downwards outside limit," deg" )," ".
    endif.
    message banner,
    " ".
    write "spefile1", view_with, view_with(horizontal left,
        horizontal right, vertical upwards, vertical downwards).
endif.
\.......................................................................................................
    message restart.
    break.
endif.
\.......................................................................................................
    if
        speedometer menu = Check that no head movement
                                    required for visibility
    then
        erase.
        eraseclass.
    read "spefile", view_with, view_with (horizontal left,
        horizontal right, vertical upwards, vertical downwards).
        obtain view_with:head move>gI head limit.
    if
        status (view_with:head move>gI head limit) = known
    then message
        " ",
        banner,
        combine(" for angles of head movement given:")," ",
```

```
        combine(" horizontal left =                 ",
              view_with:head move>horizontal left )," ",
        combine(" horizontal right =                ",
              view_with:head move>horizontal right )," ",
        combine(" vertical upwards =                ",
              view_with:head move>vertical upwards )," ",
        combine(" vertical downwards =              ",
              view_with:head move>vertical downwards)," ",
        combine(" these angles of head movement are: ",
              view_with:head move>gI head limit     )," ",
        banner,
        " ".
endif.
\..............................................................................................................
     if reason = yes then
        message banner," ".
        justify view_with:head move>gI head limit.
        message banner," ".
endif.
\..............................................................................................................
     if
        view_with:head move>gI head limit = outside gI head limit
     then
        message
        " ",
        banner,
        combine(" You can adjust the position of the ",
              "speedometer so that it "),
        combine(" lies within the allowable ergonomics head ",
              "movement by : ")," ".
     if view_with:head move>horizontal left gt 0
     then message
        combine(" move to left to reduce angle of head ",
        "movement by : ",view_with:head move>groupI head
         move horizontal left outside limit," deg" )," ".
     endif.
     if view_with:head move>horizontal right gt 0
     then message
        combine(" move to right to reduce angle of head ",
        "movement by : ",view_with:head move>groupI head
         move horizontal right outside limit," deg" )," ".
     endif.
     if view_with:head move>vertical upwards gt 0
     then message
        combine(" move down to reduce angle of head ",
        "movement by : ",view_with:head move>groupI head move
         vertical upwards outside limit," deg" )," ".
     endif.
     if view_with:head move>vertical downwards gt 0
     then message
        combine(" move up to reduce angle of head ",
        "movement by : ",view_with:head move>groupI head
         move vertical downwards outside limit," deg" )," ".
     endif.
```

```
        message banner,
        " ".

    write "spefile1", view_with, view_with (horizontal left,
        horizontal right, vertical upwards, vertical downwards).

    endif.
    message restart.
    break.
endif.
\""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
    erase speedometer menu.
    endwhile.
    erase.
    eraseclass.
endif.
\*************************************************************************
    if
        primary displays menu = Tachometer
    then
        erase.
        eraseclass.

    while tachometer menu # Finished with this menu do

\""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
    if
        tachometer menu = Check visibility through the steering wheel
    then
        erase.
        eraseclass.

    read "tafile", tacho, tacho(cox, coy, coz).
    read "tafile", st_wheel, st_wheel(cox, coy, coz).
    read "tafile", visible, visible(cox, coy, coz).

        obtain tacho:area>visibility.
    if
        status (tacho:area>visibility) = known
    then message
        " ",
        banner,
        combine (" for values of vertices given : "),
        combine (" top vertex tp = ", tacho:tp>coz),
        combine (" bottom vertex bp = ", tacho:bp>coz),
        combine (" right vertex rp = ", tacho:lp>cox),
        combine (" left vertex lp = ", tacho:rp>cox),
        combine (" the tachometer is ",
                tacho:area>visibility),
    " ",
    banner,
    " ".
endif.
\.............................................................................................
    if reason = yes then
        message banner, " ".
        justify tacho:tp>visibility.
```

```
        message banner, " ".
endif.
\..............................................................................................................
    if
        tacho:area>visibility = not visible through st_wheel
    then
        message " ",
        banner,
        combine(" the adjustment for tachometer ",
            " position is to : ").
    if tacho:tp>coz gt tacho:tp>upper boundary
    then message " ",
        combine(" lower the top point by : ",
            tacho:tp>upper boundary ," mm" ).
    endif.
    if tacho:bp>coz lt tacho:bp>lower boundary
    then message " ",
        combine(" move up the bottom point by : ",
            tacho:bp>lower boundary ," mm" ).
    endif.
    if tacho:lp>cox gt tacho:lp>left boundary
    then message " ",
        combine(" move the left point to the right by : ",
            tacho:lp>left boundary ," mm" ).
    endif.
    if tacho:rp>coz gt tacho:rp>right boundary
    then message " ",
        combine(" move the right point to the left by : ",
            tacho:rp>right boundary ," mm" ).

    endif.
    message banner,
    " ".
    write "tafile1", tacho, tacho(cox, coy, coz).
    write "tafile1", st_wheel, st_wheel(cox, coy, coz).
    write "tafile1", visible, visible(cox, coy, coz).
endif.
\..............................................................................................................
    message restart.
    break.
endif.
\..............................................................................................................
    if
        tachometer menu = Check eye movement within the ergonomics range
    then
        erase.
        eraseclass.
    read "taefile", view_with, view_with(horizontal left,
        horizontal right, vertical upwards, vertical downwards).
        obtain view_with:eye move>gl eye limit.
    if
```

```
        status (view_with:eye move>gI eye limit) = known
    then message
        " ",
        banner,
        combine(" for angles of eye movement given:")," ",
        combine(" horizontal left =              ",
            view_with:eye move>horizontal left )," ",
        combine(" horizontal right =             ",
            view_with:eye move>horizontal right )," ",
        combine(" vertical upwards =             ",
            view_with:eye move>vertical upwards )," ",
        combine(" vertical downwards =           ",
            view_with:eye move>vertical downwards)," ",
        combine(" these angles of eye movement are: ",
            view_with:eye move>gI eye limit    )," ",
    banner,
    " ".
endif.
\.........................................................................................................................

    if reason = yes then
        message banner," ".
        justify view_with:eye move>gI eye limit.
        message banner," ".
endif.
\.........................................................................................................................

    if
        view_with:eye move>gI eye limit = outside gI eye limit
    then
        message
        " ",
        banner,
        combine(" You can adjust the position of the ",
            "tachometer so that it "),
        combine(" lies within the allowable ergonomics eye ",
            "movement by : ")," ".

    if view_with:eye move>horizontal left ge 20
    then message
        combine(" move to right to reduce angle of eye ",
        "movement by : ",view_with:eye move>groupI eye
         move horizontal left outside limit," deg" )," ".
    endif.

    if view_with:eye move>horizontal right le -20
    then message
        combine(" move to right to reduce angle of eye ",
        "movement by : ",view_with:eye move>groupI eye
         move horizontal right outside limit," deg" )," ".
    endif.

    if view_with:eye move>vertical upwards gt 0
    then message
        combine(" move down to reduce angle of eye ",
        "movement by : ",view_with:eye move>groupI eye
         move vertical upwards outside limit," deg" )," ".
    endif.
```

```
      if view_with:eye move>vertical downwards lt -35
      then message
            combine(" move up to reduce angle of eye ",
            "movement by : ",view_with:eye move>groupI eye
             move vertical downwards outside limit," deg" )," ".
      endif.
      message banner,
      " ".
      write "taefile1", view_with, view_with(horizontal left,
            horizontal right, vertical upwards, vertical downwards).
endif.
\.....................................................................................................
      message restart.
      break.
endif.
\.....................................................................................................
      if
            tachometer menu = Check that no head movement required for visibility
      then
            erase.
            eraseclass.
      read "taefile", view_with, view_with (horizontal left,
            horizontal right, vertical upwards, vertical downwards).
            obtain view_with:head move>gI head limit.
      if
            status (view_with:head move>gI head limit) = known
      then message
            " ",
            banner,
            combine(" for angles of head movement given:")," ",
            combine(" horizontal left =               ",
                  view_with:head move>horizontal left )," ",
            combine(" horizontal right =              ",
                  view_with:head move>horizontal right )," ",
            combine(" vertical upwards =              ",
                  view_with:head move>vertical upwards )," ",
            combine(" vertical downwards =            ",
                  view_with:head move>vertical downwards)," ",
            combine(" these angles of head movement are: ",
                  view_with:head move>gI head limit     )," ",
            banner,
            " ".
endif.
\.....................................................................................................
      if reason = yes then
            message banner," ".
            justify view_with:head move>gI head limit.
            message banner," ".
endif.
\.....................................................................................................
      if
            view_with:head move>gI head limit = outside gI head limit
```

```
then
      message
      " ",
      banner,
      combine(" You can adjust the position of the ",
            "tachometer so that it "),
      combine(" lies within the allowable ergonomics head ",
            "movement by : ")," ".

if view_with:head move>horizontal left gt 0
then message
      combine(" move to left to reduce angle of head ",
      "movement by : ",view_with:head move>groupI head
       move horizontal left outside limit," deg" )," ".
endif.

if view_with:head move>horizontal right gt 0
then message
      combine(" move to right to reduce angle of head ",
      "movement by : ",view_with:head move>groupI head
       move horizontal right outside limit," deg" )," ".
endif.

if view_with:head move>vertical upwards gt 0
then message
      combine(" move down to reduce angle of head ",
      "movement by : ",view_with:head move>groupI head move
       vertical upwards outside limit," deg" )," ".
endif.

if view_with:head move>vertical downwards gt 0
then message
      combine(" move up to reduce angle of head ",
      "movement by : ",view_with:head move>groupI head
       move vertical downwards outside limit," deg" )," ".
endif.

      message banner,
      " ".

write "taefile1", view_with, view_with (horizontal left,
      horizontal right, vertical upwards, vertical downwards).

endif.
message restart.
break.
endif.
\""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""

erase tachometer menu.
endwhile.
erase.
eraseclass.
endif.
\***************************************************************************

if
      primary displays menu = Fuel
then
      erase.
      eraseclass.
```

```
      while fuel menu # end this menu do
\.................................................................................................
      if
            fuel menu = Check ergonomics visibility
      then
            erase.
            eraseclass.
      read "fufile", fuel, fuel(cox, coy, coz).
      read "fufile", st_wheel, st_wheel(cox, coy, coz).
      read "fufile", visible, visible(cox, coy, coz).
            obtain fuel:area>visibility.
      if
            status (fuel:area>visibility) = known
      then message
            " ",
            banner,
            combine (" for values of vertices given : "),
            combine (" top vertex tp = ", fuel:tp>coz),
            combine (" bottom vertex bp = ", fuel:bp>coz),
            combine (" right vertex rp = ", fuel:lp>cox),
            combine (" left vertex lp = ", fuel:rp>cox),
            combine (" the fuel is ",
                  fuel:area>visibility),
      " ",
      banner,
      " ".
endif.
\.................................................................................................
      if reason = yes then
            message banner, " ".
            justify fuel:area>visibility.
            message banner, " ".
endif.
\.................................................................................................
      if
            fuel:area>visibility = outside ergonomics limits
      then
            message " ",
            banner,
            combine(" the adjustment for fuel ",
                  " position is to : ").
      if fuel:tp>coz gt fuel:tp>upper boundary
      then message " ",
            combine(" lower the top point by : ",
                  fuel:tp>upper boundary ," mm" ).
      endif.
      if fuel:bp>coz lt fuel:bp>lower boundary
      then message " ",
            combine(" move up the bottom point by : ",
                  fuel:bp>lower boundary ," mm" ).
      endif.
      if fuel:lp>cox gt fuel:lp>left boundary
      then message " ",
```

```
            combine(" move the left point to the right by : ",
                  fuel:lp>left boundary ," mm" ).
      endif.
      if fuel:rp>coz gt fuel:rp>right boundary
      then message " ",
            combine(" move the right point to the left by : ",
                  fuel:rp>right boundary ," mm" ).
      endif.
      message banner,
            " ".

      write "fufile1", fuel, fuel(cox, coy, coz).
      write "fufile1", st_wheel, st_wheel(cox, coy, coz).
      write "fufile1", visible, visible(cox, coy, coz).
endif.
\..........................................................................................................
      message restart.
      break.
endif.
\..........................................................................................................
      if
            fuel menu = Check allowable eye movement
      then
            erase.
            eraseclass.
      read "fuefile", view_with, view_with (horizontal left,
            horizontal right, vertical upwards, vertical downwards).
            obtain view_with:eye move>gII eye limit.
      if
            status (view_with:eye move>gII eye limit) = known
      then message
            " ",
            banner,
            combine(" for angles of eye movement given:")," ",
            combine(" horizontal left =                 ",
                  view_with:eye move>horizontal left )," ",
            combine(" horizontal right =                ",
                  view_with:eye move>horizontal right )," ",
            combine(" vertical upwards =                ",
                  view_with:eye move>vertical upwards )," ",
            combine(" vertical downwards =              ",
                  view_with:eye move>vertical downwards)," ",
            combine(" these angles of eye movement are: ",
                  view_with:eye move>gII eye limit    )," ",
      banner,
      " ".
endif.
\..........................................................................................................
      if reason = yes then
            message banner," ".
            justify view_with:eye move>gII eye limit.
            message banner," ".
endif.
```

```
\.......................................................................................................................
      if
            view_with:eye move>gII eye limit = outside gII eye limit
      then
            message
            " ",
            banner,
            combine(" You can adjust the position of the ",
                  "fuel so that it "),
            combine(" lies within the allowable ergonomics eye ",
                  "movement by : ")," ".

      if view_with:eye move>horizontal left gt 40
      then message
            combine(" move to right to reduce angle of eye ",
            "movement by : ",view_with:eye move>groupII eye
             move horizontal left outside limit," deg" )," ".
      endif.

      if view_with:eye move>horizontal right lt -25
      then message
            combine(" move to right to reduce angle of eye ",
            "movement by : ",view_with:eye move>groupII eye
             move horizontal right outside limit," deg" )," ".
      endif.

      if view_with:eye move>vertical upwards gt 0
      then message
            combine(" move down to reduce angle of eye ",
            "movement by : ",view_with:eye move>groupII eye
             move vertical upwards outside limit," deg" )," ".
      endif.

      if view_with:eye move>vertical downwards lt -35
      then message
            combine(" move up to reduce angle of eye ",
            "movement by : ",view_with:eye move>groupII eye
             move vertical downwards outside limit," deg" )," ".
      endif.

      message banner,
      " ".

      write "fuefile1", view_with, view_with (horizontal left,
            horizontal right, vertical upwards, vertical downwards).
endif.
\.......................................................................................................................
      message restart.
      break.
endif.
\,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
      if
            fuel menu = Check allowable head movement
      then
            erase.
            eraseclass.

      read "fuefile", view_with, view_with (horizontal left,
            horizontal right, vertical upwards, vertical downwards).
```

```
          obtain view_with:head move>gII head limit.
     if
          status (view_with:head move>gII head limit) = known
     then message
          " ",
          banner,
          combine(" for angles of head movement given:")," ",
          combine(" horizontal left =              ",
               view_with:head move>horizontal left  )," ",
          combine(" horizontal right =             ",
               view_with:head move>horizontal right )," ",
          combine(" vertical upwards =             ",
               view_with:head move>vertical upwards  )," ",
          combine(" vertical downwards =           ",
               view_with:head move>vertical downwards)," ",
          combine(" these angles of head movement are: ",
               view_with:head move>gII head limit   )," ",
     banner,
     " ".
endif.
\..........................................................................................................................................

     if reason = yes then
          message banner," ".
          justify view_with:head move>gII head limit.
          message banner," ".
endif.
\.......................................................................................................................................

     if
          view_with:head move>gII head limit = outside gII head limit
     then
          message
          " ",
          banner,
          combine(" You can adjust the position of the ",
               "fuel so that it "),
          combine(" lies within the allowable ergonomics head ",
               "movement by : ")," ".

     if view_with:head move>horizontal left gt 15
     then message
          combine(" move to right to reduce angle of head ",
          "movement by : ",view_with:head move>groupII head
           move horizontal left outside limit," deg" )," ".
     endif.

     if view_with:head move>horizontal right lt -15
     then message
          combine(" move to left to reduce angle of head ",
          "movement by : ",view_with:head move>groupII head
           move horizontal right outside limit," deg" )," ".
     endif.

     if view_with:head move>vertical upwards gt 0
     then message
          combine(" move down to reduce angle of head ",
          "movement by : ",view_with:head move>groupII head move
```

```
            vertical upwards outside limit," deg" )," ".
      endif.
      if view_with:head move>vertical downwards lt -15
      then message
            combine(" move up to reduce angle of head ",
            "movement by : ",view_with:head move>groupII head
             move vertical downwards outside limit," deg" )," ".
      endif.
      message banner, " ".
      write "fuefile1", view_with, view_with (horizontal left,
            horizontal right, vertical upwards, vertical downwards).
      endif.
      message restart.
      break.
endif.
\---------------------------------------------------------------------------
      erase fuel menu.
      endwhile.
      erase.
      eraseclass.
endif.
\***************************************************************************
      if
            primary displays menu = Temperature
      then
            erase.
            eraseclass.
      while temperature menu # end this menu do
\"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
      if
            temperature menu = Check ergonomics visibility
      then
            erase.
            eraseclass.
      read "tpfile", temperature, temperature(cox, coy, coz).
      read "tpfile", st_wheel, st_wheel(cox, coy, coz).
      read "tpfile", visible, visible(cox, coy, coz).
            obtain temperature:area>visibility.
      if
            status (temperature:area>visibility) = known
      then message
            " ",
            banner,
            combine (" for values of vertices given : "),
            combine (" top vertex tp = ", temperature:tp>coz),
            combine (" bottom vertex bp = ", temperature:bp>coz),
            combine (" right vertex rp = ", temperature:lp>cox),
            combine (" left vertex lp = ", temperature:rp>cox),
            combine (" the temperature is ",
                        temperature:area>visibility),
      " ",
      banner, " ".
endif.
```

```
\..........................................................................................................................................
      if reason = yes then
            message banner, " ".
            justify temperature:area>visibility.
            message banner, " ".
endif.
\..........................................................................................................................................
      if
            temperature:area>visibility = outside ergonomics limits
      then
            message " ",
            banner,
            combine(" the adjustment for temperature ",
                  " position is to : ").

      if temperature:tp>coz gt temperature:tp>upper boundary
      then message " ",
            combine(" lower the top point by : ",
                  temperature:tp>upper boundary ," mm" ).
      endif.

      if temperature:bp>coz lt temperature:bp>lower boundary
      then message " ",
            combine(" move up the bottom point by : ",
                  temperature:bp>lower boundary ," mm" ).
      endif.

      if temperature:lp>cox gt temperature:lp>left boundary
      then message " ",
            combine(" move the left point to the right by : ",
                  temperature:lp>left boundary ," mm" ).
      endif.

      if temperature:rp>coz gt temperature:rp>right boundary
      then message " ",
            combine(" move the right point to the left by : ",
                  temperature:rp>right boundary ," mm" ).
endif.

      message banner, " ".
      write "tpfile1", temperature, temperature(cox, coy, coz).
      write "tpfile1", st_wheel, st_wheel(cox, coy, coz).
      write "tpfile1", visible, visible(cox, coy, coz).

endif.
\..........................................................................................................................................
      message restart.
      break.
endif.
\..........................................................................................................................................
      if
            temperature menu = Check allowable eye movement
      then
            erase.
            eraseclass.

      read "tpefile", view_with, view_with (horizontal left,
            horizontal right, vertical upwards, vertical downwards).
```

```
        obtain view_with:eye move>gII eye limit.
    if
        status (view_with:eye move>gII eye limit) = known
    then message
        " ",
        banner,
        combine(" for angles of eye movement given:")," ",
        combine(" horizontal left =              ",
            view_with:eye move>horizontal left )," ",
        combine(" horizontal right =             ",
            view_with:eye move>horizontal right )," ",
        combine(" vertical upwards =             ",
            view_with:eye move>vertical upwards )," ",
        combine(" vertical downwards =           ",
            view_with:eye move>vertical downwards)," ",
        combine(" these angles of eye movement are: ",
            view_with:eye move>gII eye limit    )," ",
    banner,
    " ".
endif.
\..........................................................................................
    if reason = yes then
        message banner," ".
        justify view_with:eye move>gII eye limit.
        message banner," ".
endif.
\..........................................................................................
    if
        view_with:eye move>gII eye limit = outside gII eye limit
    then
        message
        " ",
        banner,
        combine(" You can adjust the position of the ",
            "tmperature so that it "),
        combine(" lies within the allowable ergonomics eye ",
            "movement by : ")," ".

    if view_with:eye move>horizontal left gt 40
    then message
        combine(" move to right to reduce angle of eye ",
        "movement by : ",view_with:eye move>groupII eye
         move horizontal left outside limit," deg" )," ".
    endif.

    if view_with:eye move>horizontal right lt -25
    then message
        combine(" move to right to reduce angle of eye ",
        "movement by : ",view_with:eye move>groupII eye
         move horizontal right outside limit," deg" )," ".
    endif.

    if view_with:eye move>vertical upwards gt 0
    then message
        combine(" move down to reduce angle of eye ",
        "movement by : ",view_with:eye move>groupII eye
```

```
            move vertical upwards outside limit," deg" )," ".
      endif.
      if view_with:eye move>vertical downwards lt -35
      then message
            combine(" move up to reduce angle of eye ",
            "movement by : ",view_with:eye move>groupII eye
             move vertical downwards outside limit," deg" )," ".
      endif.
      message banner, " ".
      write "tpefile1", view_with, view_with (horizontal left,
            horizontal right, vertical upwards, vertical downwards).
endif.
\.........................................................................................................
      message restart.
      break.
endif.
\.........................................................................................................
      if
            temperature menu = Check allowable head movement
      then
            erase.
            eraseclass.
      read "tpefile", view_with, view_with (horizontal left,
            horizontal right, vertical upwards, vertical downwards).
            obtain view_with:head move>gII head limit.
      if
            status (view_with:head move>gII head limit) = known
      then message
            " ", banner,
            combine(" for angles of head movement given:")," ",
            combine(" horizontal left =                  ",
                  view_with:head move>horizontal left  )," ",
            combine(" horizontal right =                 ",
                  view_with:head move>horizontal right  )," ",
            combine(" vertical upwards =                 ",
                  view_with:head move>vertical upwards  )," ",
            combine(" vertical downwards =               ",
                  view_with:head move>vertical downwards)," ",
            combine(" these angles of head movement are: ",
                  view_with:head move>gII head limit    )," ",
            banner,
            " ".
endif.
\.........................................................................................................
      if reason = yes then
            message banner," ".
            justify view_with:head move>gII head limit.
            message banner," ".
endif.
\.........................................................................................................
      if
            view_with:head move>gII head limit = outside gII head limit
```

```
then
      message
      " ",
      banner,
      combine(" You can adjust the position of the ",
            "temperature so that it "),
      combine(" lies within the allowable ergonomics head ",
            "movement by : ")," ".
if view_with:head move>horizontal left gt 15
then message
      combine(" move to right to reduce angle of head ",
      "movement by : ",view_with:head move>groupII head
       move horizontal left outside limit," deg" )," ".
endif.
if view_with:head move>horizontal right lt -15
then message
      combine(" move to left to reduce angle of head ",
      "movement by : ",view_with:head move>groupII head
       move horizontal right outside limit," deg" )," ".
endif.
if view_with:head move>vertical upwards gt 0
then message
      combine(" move down to reduce angle of head ",
      "movement by : ",view_with:head move>groupII head move
       vertical upwards outside limit," deg" )," ".
endif.
if view_with:head move>vertical downwards lt -15
then message
      combine(" move up to reduce angle of head ",
      "movement by : ",view_with:head move>groupII head
       move vertical downwards outside limit," deg" )," ".
endif.
message banner, " ".
write "tpefile1", view_with, view_with (horizontal left,
      horizontal right, vertical upwards, vertical downwards).
endif.
message restart.
break.
endif.
\-----------------------------------------------------------------------------
erase temperature menu.
endwhile.
erase.
eraseclass.
endif.
\**************************************************************************
if
      primary displays menu = Door_Open
then
      erase.
      eraseclass.
while door_open menu # end this menu do
```

```
\..........................................................................................................
    if
        door_open menu = Check ergonomics visibility
    then
        erase.
        eraseclass.
    read "dofile", door_open, door_open(cox, coy, coz).
    read "dofile", st_wheel, st_wheel(cox, coy, coz).
    read "dofile", visible, visible(cox, coy, coz).
        obtain door_open :area>visibility.
    if
        status (door_open :area>visibility) = known
    then message
        " ", banner,
        combine (" for values of vertices given : "),
        combine (" top vertex tp = ", door_open :tp>coz),
        combine (" bottom vertex bp = ", door_open:bp>coz),
        combine (" right vertex rp = ", door_open:lp>cox),
        combine (" left vertex lp = ", door_open:rp>cox),
        combine (" the door_open is ",
            door_open:area>visibility),
    " ",
    banner, " ".
endif.
\..........................................................................................................
    if reason = yes then
        message banner, " ".
        justify door_open:area>visibility.
        message banner, " ".
endif.
\..........................................................................................................
    if
        door_open:area>visibility = outside ergonomics limits
    then
        message " ",
        banner,
        combine(" the adjustment for door_open ",
            " position is to : ").
    if door_open:tp>coz gt door_open :tp>upper boundary
    then message " ",
        combine(" lower the top point by : ",
            door_open :tp>upper boundary ," mm" ).
    endif.
    if door_open:bp>coz lt door_open:bp>lower boundary
    then message " ",
        combine(" move up the bottom point by : ",
            door_open :bp>lower boundary ," mm" ).
    endif.
    if door_open:lp>cox gt door_open:lp>left boundary
    then message " ",
        combine(" move the left point to the right by : ",
            door_open:lp>left boundary ," mm" ).
    endif.
```

```
    if door_open:rp>coz gt door_open:rp>right boundary
    then message " ",
        combine(" move the right point to the left by : ",
            door_open :rp>right boundary ," mm" ).
    endif.
    message banner, " ".
    write "dofile1", door_open, door_open(cox, coy, coz).
    write "dofile1", st_wheel, st_wheel(cox, coy, coz).
    write "dofile1", visible, visible(cox, coy, coz).
endif.
\..................................................................................................................
    message restart.
    break.
endif.
\..................................................................................................................
    if
        door_open  menu = Check allowable eye movement
    then
        erase.
        eraseclass.
    read "doefile", view_with, view_with (horizontal left,
        horizontal right, vertical upwards, vertical downwards).
        obtain view_with:eye move>gII eye limit.
    if
        status (view_with:eye move>gII eye limit) = known
    then message
        " ", banner,
        combine(" for angles of eye movement given:")," ",
        combine(" horizontal left =             ",
            view_with:eye move>horizontal left )," ",
        combine(" horizontal right =            ",
            view_with:eye move>horizontal right )," ",
        combine(" vertical upwards =            ",
            view_with:eye move>vertical upwards )," ",
        combine(" vertical downwards =          ",
            view_with:eye move>vertical downwards)," ",
        combine(" these angles of eye movement are: ",
            view_with:eye move>gII eye limit    )," ",
    banner,
    " ".
endif.
\..................................................................................................................
    if reason = yes then
        message banner," ".
        justify view_with:eye move>gII eye limit.
        message banner," ".
endif.
\..................................................................................................................
    if
        view_with:eye move>gII eye limit = outside gII eye limit
    then
        message
```

```
          " ", banner,
          combine(" You can adjust the position of the ",
                "door_open  so that it "),
          combine(" lies within the allowable ergonomics eye ",
                "movement by : ")," ".
     if view_with:eye move>horizontal left gt 40
     then message
          combine(" move to right to reduce angle of eye ",
          "movement by : ",view_with:eye move>groupII eye
           move horizontal left outside limit," deg" )," ".
     endif.
     if view_with:eye move>horizontal right lt -25
     then message
          combine(" move to right to reduce angle of eye ",
          "movement by : ",view_with:eye move>groupII eye
           move horizontal right outside limit," deg" )," ".
     endif.
     if view_with:eye move>vertical upwards gt 0
     then message
          combine(" move down to reduce angle of eye ",
          "movement by : ",view_with:eye move>groupII eye
           move vertical upwards outside limit," deg" )," ".
     endif.
     if view_with:eye move>vertical downwards lt -35
     then message
          combine(" move up to reduce angle of eye ",
          "movement by : ",view_with:eye move>groupII eye
           move vertical downwards outside limit," deg" )," ".
     endif.
     message banner, " ".
     write "doefile1", view_with, view_with (horizontal left,
          horizontal right, vertical upwards, vertical downwards).
endif.
\..........................................................................................................
     message restart.
     break.
endif.
\..........................................................................................................
     if
          door_open  menu = Check allowable head movement
     then
          erase.
          eraseclass.
     read "doefile", view_with, view_with (horizontal left,
     horizontal right, vertical upwards, vertical downwards).
          obtain view_with:head move>gII head limit.
     if
          status (view_with:head move>gII head limit) = known
     then
          message
          " ",
```

```
        banner,
        combine(" for angles of head movement given:")," ",
        combine(" horizontal left =              ",
             view_with:head move>horizontal left )," ",
        combine(" horizontal right =             ",
             view_with:head move>horizontal right )," ",
        combine(" vertical upwards =             ",
             view_with:head move>vertical upwards )," ",
        combine(" vertical downwards =           ",
             view_with:head move>vertical downwards)," ",
        combine(" these angles of head movement are: ",
             view_with:head move>gII head limit   )," ",
        banner,
        " ".
endif.
\...............................................................................................
    if reason = yes then
        message banner, " ".
        justify view_with:head move>gII head limit.
        message banner," ".
endif.
\...............................................................................................
    if
        view_with:head move>gII head limit = outside gII head limit
    then
        message " ", banner,
        combine(" You can adjust the position of the ",
             "door_open  so that it "),
        combine(" lies within the allowable ergonomics head ",
             "movement by : ")," ".
    if view_with:head move>horizontal left gt 15
    then message
        combine(" move to right to reduce angle of head ",
        "movement by : ",view_with:head move>groupII head
         move horizontal left outside limit," deg" )," ".
    endif.
    if view_with:head move>horizontal right lt -15
    then message
        combine(" move to left to reduce angle of head ",
        "movement by : ",view_with:head move>groupII head
         move horizontal right outside limit," deg" )," ".
    endif.
    if view_with:head move>vertical upwards gt 0
    then message
        combine(" move down to reduce angle of head ",
        "movement by : ",view_with:head move>groupII head move
         vertical upwards outside limit," deg" )," ".
    endif.
    if view_with:head move>vertical downwards lt -15
    then message
        combine(" move up to reduce angle of head ",
        "movement by : ",view_with:head move>groupII head
         move vertical downwards outside limit," deg" )," ".
    endif.
```

```
    message
        banner, " ".

    write "doefile1", view_with, view_with (horizontal left,
       horizontal right, vertical upwards, vertical downwards).
    endif.
    message restart.
    break.
endif.
\""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
    erase door_open menu.
    endwhile.
    erase.
    eraseclass.
endif.
\*************************************************************************
    if
        main menu = Design a particular interior displays item
    then
        erase.
        eraseclass.
\~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    while interior displays menu # Leave this menu do
    if
        interior displays menu = New case
    then
        erase.
        eraseclass.
    endif.
\~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    if
        interior displays menu = Primary Controls
    then
        erase.
        eraseclass.
\~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    while primary controls menu # Leave this menu do
    if
       primary controls menu = New case
    then
        erase.
        eraseclass.
    endif.
\~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    if
        primary controls menu = Pedals
    then
        erase.
        eraseclass.
\~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    while pedals menu # Leave this menu do
```

```
    if
          pedals menu = New case
    then
          erase.
          eraseclass.
    endif.
\--------------------------------------------------------------------------------
    if
          pedals menu = Accelerator
    then
          erase.
          eraseclass.

    while accelerator menu # Finished with this menu do

\""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
    if
          accelerator menu = Check reachable through the seat adjustment
    then
          erase.
          eraseclass.

    read "acfile", accele, accele(cox, coy, coz).
    read "acfile", seat_adjust, seat_adjust(cox, coy, coz).
    read "acfile", reach, reach(cox, coy, coz).

          obtain accele:area>reachable.
    if
          status (accele:area>reachable) = known
    then
          message " ", banner,
          combine (" for values of vertices given : "),
          combine (" top vertex tp = ", accele:tp>coz),
          combine (" bottom vertex bp = ", accele:bp>coz),
          combine (" right vertex rp = ", accele:lp>cox),
          combine (" left vertex lp = ", accele:rp>cox),
          combine (" the accelerator is ", accele:area>reachable),
    banner,
          " ".
endif.
\..................................................................................
    if reason = yes then
          message banner, " ".
          justify accele:tp>reachable.
          message banner, " ".
endif.
\..................................................................................
    if
          accele:area>reachable = not reach through the seat adjustment

    then
          message " ", banner,
          combine(" the adjustment for accelerator position is to : ").

    if accele:tp>coz gt reach:pedal>top limit
    then message " ",
          combine(" move down at least by a min   ",
```

```
            reach:pedal>move accele down ," mm" ).
        endif.

        if accele:bp>coz lt reach:pedal>bottom limit
        then message " ",
            combine(" move up at least by a min     ",
            reach:pedal>move accele up ," mm" ).
        endif.

        if accele:lp>cox gt reach:pedal>left limit
        then message " ",
            combine(" move right at least by a min  ",
            reach:pedal>move accele to right ," mm" ).
        endif.

        if accele:rp>cox lt reach:pedal>right limit
        then message " ",
            combine(" move left at least by a min ",
            reach:pedal>move accele to left ," mm" ).
        endif.

        message
            banner, " ".

        write "acfile1", accele, accele(cox, coy, coz).
        write "acfile1", seat_adjust, seat_adjust(cox, coy, coz).
        write "acfile1", reach, reach(cox, coy, coz).

endif.

\..............................................................................................................................

        message restart.
        break.
endif.

\"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"
        if
            accelerator menu = Check leg move within the ergonomics range
        then
            erase.
            eraseclass.

        read "acrfile", dist_with, dist_with(horizontal left,
            horizontal right, vertical upwards, vertical downwards).

            obtain dist_with:leg move>gl leg limit.
        if
            status (dist_with:leg move>gl leg limit) = known
        then
             message " ", banner,
            combine(" for angles of leg movement given:")," ",
            combine(" horizontal left =  ",
                dist_with:leg move>horizontal left )," ",
            combine(" horizontal right =    ",
                dist_with:leg move>horizontal right )," ",
            combine(" vertical upwards =    ",
                dist_with:leg move>vertical upwards )," ",
            combine(" vertical downwards =  ",
                dist_with:leg move>vertical downwards)," ",
            combine(" these angles of leg movement are: ",
```

```
                dist_with:leg move>gl leg limit )," ",
            banner,
            " ".
endif.
\..................................................................................................................
        if reason = yes then
            message banner," ".
            justify dist_with:leg move>gl leg limit.
            message banner," ".
endif.
\..................................................................................................................
        if
            dist_with:leg move>gl leg limit = outside gl leg limit
        then
            message " ", banner,
            combine(" You can adjust the position of the   ",
                "accelerator so that it "),
            combine(" lies within the allowable  ",
                "ergonomics leg movement by : ")," ".
        if dist_with:leg move>horizontal left ge 20
        then message
            combine(" move to right to reduce angle of leg ",
            "movement by : ",dist_with:leg move>groupl leg
             move horizontal left outside limit," deg" )," ".
        endif.
        if dist_with:leg move>horizontal right le -20
        then message
            combine(" move to right to reduce angle of leg ",
            "movement by : ",dist_with:leg move>groupl leg
             move horizontal right outside limit," deg" )," ".
        endif.
        if dist_with:leg move>vertical upwards gt 0
        then message
            combine(" move down to reduce angle of leg ",
            "movement by : ",dist_with:leg move>groupl leg
             move vertical upwards outside limit," deg" )," ".
        endif.
        if dist_with:leg move>vertical downwards lt -35
        then message
            combine(" move up to reduce angle of leg ",
            "movement by : ",dist_with:leg move>groupl leg
             move vertical downwards outside limit," deg" )," ".
        endif.
        message
            banner, " ".
        write "acrfile1", dist_with, dist_with(horizontal left,
            horizontal right, vertical upwards, vertical downwards).
endif.
\..................................................................................................................
            message restart.
            break.
endif.
```

```
\..............................................................................................................
    if
        accelerator menu = Check seat move required for reachable
    then
        erase.
        eraseclass.
    read "acrfile", dist_with, dist_with (horizontal left,
        horizontal right, vertical upwards, vertical downwards).
        obtain dist_with:seat move>gI seat limit.
    if
        status (dist_with:seat move>gI seat limit) = known
    then
        message " ", banner,
        combine(" for angles of seat adjsutment given:"),"",
        combine(" horizontal left =    ",
            dist_with:seat move>horizontal left),"",
        combine(" horizontal right =   ",
            dist_with:seat move>horizontal right),"",
        combine(" vertical upwards =   ",
            dist_with:seat move>vertical upwards),"",
        combine(" vertical downwards =   ",
            dist_with:seat move>vertical downwards),"",
        combine(" these angles of seat movement are: ",
            dist_with:seat move>gI seat limit)," ",
        banner,
        " ".
endif.
\..............................................................................................................
    if reason = yes then
        message banner," ".
        justify dist_with:seat move>gI seat limit.
        message banner," ".
endif.
\..............................................................................................................
    if
        dist_with:seat move>gI seat limit = outside gI seat limit
    then
        message
        " ", banner,
        combine(" You can adjust the position of the ",
            " accelerator so that it "),
        combine(" lies within the allowable ergonomics seat ",
            "adjustment by : ")," ".
    if dist_with:seat adjus>horizontal left gt 0
    then message
        combine(" move to left to reduce angle of seat ",
        "adjustment by : ",dist_with:seat move>groupI seat
         move horizontal left outside limit," deg" )," ".
    endif.
    if dist_with:seat move>horizontal right gt 0
    then message
        combine(" move to right to reduce angle of seat ",
        "adjustment by : ",dist_with:seat move>groupI seat
```

```
            move horizontal right outside limit," deg" )," ".
      endif.

      if dist_with:seat move>vertical upwards gt 0
      then message
            combine(" move down to reduce angle of seat ",
            "adjustment by : ",dist_with:seat move>groupI seat
             move vertical upwards outside limit," deg" )," ".
      endif.

      if dist_with:seat move>vertical downwards gt 0
      then message
            combine(" move up to reduce angle of seat ",
            "adjustment by : ",dist_with:seat move>groupI seat
             move vertical downwards outside limit," deg" )," ".
      endif.

            message banner, " ".

      write "acrfile1", dist_with, dist_with (horizontal left,
            horizontal right, vertical upwards, vertical downwards).

      endif.
      message restart.
      break.
endif.

\,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
      erase accelerator menu.
      endwhile.
      erase.
      eraseclass.
endif.
\***************************************************************************
      if
            pedals menu = Brake
      then
            erase.
            eraseclass.

      while brake menu # Finished with this menu do

\,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
      if
            brake menu = Check reachable through the seat adjustment
      then
            erase.
            eraseclass.

      read "bkfile", brake, brake(cox, coy, coz).
      read "bkfile", seat_adjust, seat_adjust(cox, coy, coz).
      read "bkfile", reach, reach(cox, coy, coz).

            obtain brake:area>reachable.
      if
            status (brake:area>reachable) = known
      then
            message " ", banner,
            combine (" for values of vertices given : "),
            combine (" top vertex tp = ", brake:tp>coz),
            combine (" bottom vertex bp = ", brake:bp>coz),
```

```
        combine (" right vertex rp = ", brake:lp>cox),
        combine (" left vertex lp = ", brake:rp>cox),
        combine (" the brake is brake:area>reachable),
    banner,
    " ".
endif.
\..........................................................................................................
    if reason = yes then
        message banner, " ".
        justify brake:tp>reachable.
        message banner, " ".
endif.
\..........................................................................................................
    if
        brake:area>reachable = not reach through the seat adjustment
    then
        message
        " ", banner,
        combine(" the adjustment for brake ",
            " position is to : ").
    if  brake:tp>coz gt brake:tp>upper boundary
    then message " ",
        combine(" lower the top point by : ",
            brake:tp>upper boundary ," mm" ).
    endif.
    if  brake:bp>coz lt brake:bp>lower boundary
    then message " ",
        combine(" move up the bottom point by : ",
            brake:bp>lower boundary ," mm" ).
    endif.
    if  brake:lp>cox gt brake:lp>left boundary
    then message " ",
        combine(" move the left point to the right by : ",
            brake:lp>left boundary ," mm" ).
    endif.
    if  brake:rp>coz gt  brake:rp>right boundary
    then message " ",
        combine(" move the right point to the left by : ",
            brake:rp>right boundary ," mm" ).

    endif.
    message banner,
    " ".
    write "bkfile1", brake, brake(cox, coy, coz).
    write "bkfile1", seat_adjust, seat_adjust(cox, coy, coz).
    write "bkfile1", reach, reach(cox, coy, coz).

endif.
\..........................................................................................................
    message restart.
    break.
endif.
```

```
\.........................................................................................................
    if
        brake menu = Check leg move within the ergonomics range
    then
        erase.
        eraseclass.
    read "bkrfile", dist_with, dist_with(horizontal left,
        horizontal right, vertical upwards, vertical downwards).
        obtain dist_with:leg move>gl leg limit.
    if
        status (dist_with:leg move>gl leg limit) = known
    then
         message " ", banner,
        combine(" for angles of leg movement given:")," ",
        combine(" horizontal left =  ",
            dist_with:leg move>horizontal left )," ",
        combine(" horizontal right =  ",
            dist_with:leg move>horizontal right )," ",
        combine(" vertical upwards =  ",
            dist_with:leg move>vertical upwards )," ",
        combine(" vertical downwards = ",
            dist_with:leg move>vertical downwards)," ",
        combine(" these angles of leg movement are: ",
            dist_with:leg move>gl leg limit )," ",
    banner, " ".
endif.
\.........................................................................................................
    if reason = yes then
        message banner," ".
        justify dist_with:leg move>gl leg limit.
        message banner," ".
endif.
\.........................................................................................................
    if
        dist_with:leg move>gl leg limit = outside gl leg limit
    then
        message " ", banner,
        combine(" You can adjust the position of the ",
            "brake so that it "),
        combine(" lies within the allowable ergonomics leg ",
            "movement by : ")," ".
    if dist_with:leg move>horizontal left ge 20
    then message
        combine(" move to right to reduce angle of leg ",
        "movement by : ",dist_with:leg move>groupl leg
         move horizontal left outside limit," deg" )," ".
    endif.
    if dist_with:leg move>horizontal right le -20
    then message
        combine(" move to right to reduce angle of leg ",
        "movement by : ",dist_with:leg move>groupl leg
         move horizontal right outside limit," deg" )," ".
    endif.
```

```
        if dist_with:leg move>vertical upwards gt 0
        then message
              combine(" move down to reduce angle of leg ",
              "movement by : ",dist_with:leg move>groupI leg
               move vertical upwards outside limit," deg" )," ".
        endif.
        if dist_with:leg move>vertical downwards lt -35
        then message
              combine(" move up to reduce angle of leg ",
              "movement by : ",dist_with:leg move>groupI leg
               move vertical downwards outside limit," deg" )," ".
        endif.
        message banner, " ".
        write "bkrfile1", dist_with, dist_with(horizontal left,
              horizontal right, vertical upwards, vertical downwards).
endif.
\...........................................................................................................
        message restart.
        break.
endif.
\"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
        if
              brake menu = Check seat move required for reachable
        then
              erase.
              eraseclass.
        read "bkrfile", dist_with, dist_with (horizontal left,
              horizontal right, vertical upwards, vertical downwards).
              obtain dist_with:seat move>gI seat limit.
        if
              status (dist_with:seat move>gI seat limit) = known
        then
              message " ", banner,
              combine(" for angles of seat adjustment given:"),"",
              combine(" horizontal left = ",
                    dist_with:seat move>horizontal left),"",
              combine(" horizontal right = ",
                    dist_with:seat move>horizontal right),"",
              combine(" vertical upwards = ",
                    dist_with:seat move>vertical upwards),"",
              combine(" vertical downwards = ",
                    dist_with:seat move>vertical downwards),"",
              combine(" these angles of seat movement are: ",
                    dist_with:seat move>gI seat limit)," ",
        banner, " ".
endif.
\...........................................................................................................
        if reason = yes then
              message banner," ".
              justify dist_with:seat move>gI seat limit.
              message banner," ".
endif.
```

```
\.............................................................................................................................
    if
         dist_with:seat move>gI seat limit = outside gI seat limit
    then
         message
         " ", banner,
         combine(" You can adjust the position of the brake so that it "),
         combine(" lies within the allowable ergonomics seat  adjustment by : ")," ".
    if dist_with:seat adjus>horizontal left gt 0
    then message
         combine(" move to left to reduce angle of seat ",
         "adjust by : ",dist_with:seat move>groupI seat
          move horizontal left outside limit," deg" )," ".
    endif.
    if dist_with:seat move>horizontal right gt 0
    then message
         combine(" move to right to reduce angle of seat ",
         "adjustment by : ",dist_with:seat move>groupI seat
          move horizontal right outside limit," deg" )," ".
    endif.
    if dist_with:seat move>vertical upwards gt 0
    then message
         combine(" move down to reduce angle of seat ",
         "adjustment by : ",dist_with:seat move>groupI seat
          move vertical upwards outside limit," deg" )," ".
    endif.
    if dist_with:seat move>vertical downwards gt 0
    then message
         combine(" move up to reduce angle of seat ",
         "adjustment by : ",dist_with:seat move>groupI seat
          move vertical downwards outside limit," deg" )," ".
    endif.
         message banner, " ".
    write "bkrfile1", dist_with, dist_with (horizontal left,
         horizontal right, vertical upwards, vertical downwards).
    endif.
    message restart.
    break.
endif.
\""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
    erase brake menu.
    endwhile.
    erase.
    eraseclass.
endif.
\*********************************************************************
    if
         pedals menu = Clutch
    then
         erase.
         eraseclass.
```

```
      while clutch menu # Finished with this menu do
\.........................................................................................................
      if
            clutch menu = Check ergonomics reachable
      then
            erase.
            eraseclass.
      read "clfile", clutch, clutch(cox, coy, coz).
      read "clfile", seat_adjust, seat_adjust(cox, coy, coz).
      read "clfile", reach, reach(cox, coy, coz).
            obtain clutch:area>reachable.
      if
            status (clutch:area>reachable) = known
      then message
            " ", banner,
            combine (" for values of vertices given : "),
            combine (" top vertex tp = ", clutch:tp>coz),
            combine (" bottom vertex bp = ", clutch:bp>coz),
            combine (" right vertex rp = ", clutch:lp>cox),
            combine (" left vertex lp = ", clutch:rp>cox),
            combine (" the clutch is ",
                        clutch:area>reachable),
      banner,
      " ".
endif.
\.........................................................................................................
      if reason = yes then
            message banner, " ".
            justify clutch:tp>reachable.
            message banner, " ".
endif.
\.........................................................................................................
      if
            clutch:area>reachable = not reach through the seat adjustment
      then
      message " ",
            banner,
            combine(" the adjustment for clutch ",
                  " position is to : ").
      if clutch:tp>coz gt clutch:tp>upper boundary
      then message " ",
            combine(" lower the top point by : ",
                  clutch:tp>upper boundary ," mm" ).
      endif.
      if clutch:bp>coz lt clutch:bp>lower boundary
      then message " ",
            combine(" move up the bottom point by : ",
                  clutch:bp>lower boundary ," mm" ).
      endif.
      if clutch:lp>cox gt clutch:lp>left boundary
      then message " ",
            combine(" move the left point to the right by : ",
```

```
                    clutch:lp>left boundary ," mm" ).
        endif.

        if  clutch:rp>coz gt  clutch:rp>right boundary
        then message " ",
                combine(" move the right point to the left by : ",
                        clutch:rp>right boundary ," mm" ).
        endif.

        message banner,
        " ".

        write "clfile1", clutch, clutch(cox, coy, coz).
        write "clfile1", seat_adjust, seat_adjust(cox, coy, coz).
        write "clfile1", reach, reach(cox, coy, coz).
endif.
\.........................................................................................................................
                message restart.
                break.
endif.
\.........................................................................................................................
        if
                clutch menu = Check allowable leg move
        then
                erase.
                eraseclass.
        read "clrfile", dist_with, dist_with(horizontal left,
                horizontal right, vertical upwards, vertical downwards).
                obtain dist_with:leg move>gII leg limit.
        if
                status (dist_with:leg move>gII leg limit) = known
        then message
                " ",
                banner,
                combine(" for angles of leg movement given:")," ",
                combine(" horizontal left =                ",
                        dist_with:leg move>horizontal left )," ",
                combine(" horizontal right =               ",
                        dist_with:leg move>horizontal right )," ",
                combine(" vertical upwards =               ",
                        dist_with:leg move>vertical upwards )," ",
                combine(" vertical downwards =             ",
                        dist_with:leg move>vertical downwards)," ",
                combine(" these angles of leg movement are: ",
                        dist_with:leg move>gII leg limit     )," ",
        banner,
        " ".
endif.
\.........................................................................................................................
        if reason = yes then
                message banner," ".
                justify dist_with:leg move>gII leg limit.
                message banner," ".
endif.
```

```
\.......................................................................................................................
    if
        dist_with:leg move>gII leg limit = outside gII leg limit
    then message
        " ",
        banner,
        combine(" You can adjust the position of the ",
            "clutch so that it "),
        combine(" lies within the allowable ergonomics leg ",
            "movement by : ")," ".
    if dist_with:leg move>horizontal left ge 40
    then message
        combine(" move to right to reduce angle of leg ",
        "movement by : ",dist_with:leg move>groupII leg
         move horizontal left outside limit," deg" )," ".
    endif.
    if dist_with:leg move>horizontal right le -20
    then message
        combine(" move to right to reduce angle of leg ",
        "movement by : ",dist_with:leg move>groupII leg
         move horizontal right outside limit," deg" )," ".
    endif.
    if dist_with:leg move>vertical upwards gt 0
    then message
        combine(" move down to reduce angle of leg ",
        "movement by : ",dist_with:leg move>groupII leg
         move vertical upwards outside limit," deg" )," ".
    endif.
    if dist_with:leg move>vertical downwards lt -35
    then message
        combine(" move up to reduce angle of leg ",
        "movement by : ",dist_with:leg move>groupII leg
         move vertical downwards outside limit," deg" )," ".
    endif.
    message banner,
    " ".
    write "clrfile1", dist_with, dist_with(horizontal left,
        horizontal right, vertical upwards, vertical downwards).
endif.
\.......................................................................................................................
    message restart.
    break.
endif.
\"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
    if
        clutch menu = Check allowable seat move
    then
        erase.
        eraseclass.
    read "clrfile", dist_with, dist_with (horizontal left,
        horizontal right, vertical upwards, vertical downwards).
```

```
        obtain dist_with:seat move>gII seat limit.
    if
        status (dist_with:seat move>gII seat limit) = known
    then message
        " ",
        banner,
        combine(" for angles of seat adjustment given:"),"",
        combine(" horizontal left =             ",
              dist_with:seat move>horizontal left),"",
        combine(" horizontal right =            ",
              dist_with:seat move>horizontal right),"",
        combine(" vertical upwards =            ",
              dist_with:seat move>vertical upwards),"",
        combine(" vertical downwards =          ",
              dist_with:seat move>vertical downwards),"",
        combine(" these angles of seat movement are: ",
              dist_with:seat move>gII seat limit)," ",
    banner,
    " ".
endif.
\.......................................................................................
    if reason = yes then
        message banner," ".
        justify dist_with:seat move>gII seat limit.
        message banner," ".
endif.
\.......................................................................................
    if
        dist_with:seat move>gII seat limit = outside gII seat limit
    then message
        " ",
        banner,
        combine(" You can adjust the position of the ",
              " clutch so that it "),
        combine(" lies within the allowable ergonomics seat ",
              "adjustment by : ")," ".

    if dist_with:seat move>horizontal left gt 15
    then message
        combine(" move to left to reduce angle of seat ",
        "adjustment by : ",dist_with:seat move>groupII seat
         move horizontal left outside limit," deg" )," ".
    endif.

    if dist_with:seat move>horizontal right lt -15
    then message
        combine(" move to right to reduce angle of seat ",
        "adjustment by : ",dist_with:seat move>groupII seat
         move horizontal right outside limit," deg" )," ".
    endif.

    if dist_with:seat move>vertical upwards gt 0
    then message
        combine(" move down to reduce angle of seat ",
        "adjustment by : ",dist_with:seat move>groupII seat
         move vertical upwards outside limit," deg" )," ".
    endif.
```

```
        if dist_with:seat move>vertical downwards lt -15
        then message
               combine(" move up to reduce angle of seat ",
               "adjustment by : ",dist_with:seat move>groupII seat
                move vertical downwards outside limit," deg" )," ".
        endif.
               message banner,
               " ".

        write "clrfile1", dist_with, dist_with (horizontal left,
               horizontal right, vertical upwards, vertical downwards).

endif.
        message restart.
        break.
endif.
\--------------------------------------------------------------------------
        erase clutch menu.
        endwhile.
        erase.
        eraseclass.
endif.
\**********************************************************************
               erase pedals menu.
               erase.
               eraseclass.
               erase main menu.
               endwhile.
\""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
        while main menu # Quit do

        if
               main menu = Check the design for styling
        then
               erase.
               eraseclass.

        read "spfile", speedo, speedo(cox, coy, coz).

               obtain displays:panel>styling.

        if
               status (displays:panel>styling) = known
        then
        message " ", banner,
               combine ("styling is rated to be ",
                        displays:panel>styling),
               " ", banner.
               endif.

               obtain pedals:controls>styling.
        if
               status (pedals:controls>styling) = known
        then
        message " ", banner,
               combine ("styling is rated to be ",
                        pedals:controls>styling),
               " ", banner.
```

```
        endif.
    endif.
```

```
    if
        main menu = Check the type for instruments
    then
        erase.
        eraseclass.
        obtain displays:panel>type.
    if
        status (displays:panel>type) = known
    then
    message " ", banner,
        combine ("type is rated to be ",
                    displays:panel>type),
        " ", banner.
        endif.
    endif.
```

```
    if
        main menu = Check the colour for instruments
    then
        erase.
        eraseclass.
        obtain displays:panel>colour.
    if
        status (displays:panel>colour) = known
    then
    message " ", banner,
        combine ("colour is rated to be ",
    "Red is danger, damage to equipment immediate or imminent,",
    "hot in climate control system or temperature indicators.",
                    displays:panel>colour),
        " ", banner.
        endif.
        obtain displays:panel>colour.
    if
        status (displays:panel>colour) = known
    then
    message " ", banner,
        combine ("colour is rated to be ",
    "Yellow is caution, vehicle system malfunction, danger in,",
    "vehicle likely,or other condition which may produce hazard",
    "in the longer term.",
                    displays:panel>colour),
        " ", banner.
        endif.
        obtain displays:panel>colour.
    if
        status (displays:panel>colour) = known
    then
    message " ", banner,
        combine ("colour is rated to be ",
    "Green is safe, normal operation of the vehicle system,",
```

```
       "when blue or yellow are not required.",
                        displays:panel>colour),
              " ", banner.
              endif.
              obtain displays:panel>colour.
       if
              status (displays:panel>colour) = known
       then
       message " ", banner,
              combine ("colour is rated to be ",
       "Blue is driving_upper_high beam tell-tales only,and cold,",
       "in climate control systems or temperature indicators.",
                        displays:panel>colour),
              " ", banner.
              endif.
              obtain displays:panel>colour.
       if
              status (displays:panel>colour) = known
       then
       message " ", banner,
              combine ("colour is rated to be ",
       "White is other conditions where none of the above colours,",
       "are appropriate.",
                        displays:panel>colour),
              " ", banner.
       endif.
endif.
\-------------------------------------------------------------------
              erase primary displays menu.
              erase.
              eraseclass.
              erase main menu.
\-------------------------------------------------------------------
              erase primary controls menu.
              erase.
              eraseclass.
              erase main menu.
\-----------------------------------------------------------------------

stop.
endwhile.
endif.
endwhile.
endif.
endwhile.
endif.
endwhile.
endif.
endwhile.
endif.
endwhile.
endif.
endwhile.
endif.
endwhile.
```

```
endwhile.
endwhile.
endwhile.
endwhile.
endwhile.
endwhile.
endwhile.
endwhile.
endwhile.
endif.
endwhile.
%
\*************************************************************************
```

APPENDIX 3

EDKBES Inference Mechanism/Engine Run–Time System

In running the EDKBES in windowed KES PS run–time system requires information concerning the user intentions – what he is going to do in the session. On startup user has to load >kesp dan.kb and it is run to parsed the knowledge base. Saving parsed knowledge base in 'dan.pkb'. Then, the user have to load >kesr dan.pkb. The initial dialogue is given below:–

Knowledge Engineering System (KES), Release 3.0.
Copyright 1990, Software Architecture & Engineering, Inc.
Parsing the knowledge base 'dan.kb'.
Saving parsed knowledge base in 'dan.pkb'.
YES>kesr dan.pkb

Knowledge Engineering System (KES), Release 3.0.
Copyright 1990, Software Architecture & Engineering, Inc.
Loading the knowledge base 'kesr dan.pkb'.

****
WELCOME TO THE INTERIOR OF A CAR DESIGN MONITOR EDKBES.
****

This Ergonomics Design Knowledge Base Expert System EDKBES knowledge base attempts to assist, the engineer and designer while designing for driver visibility through steering wheel, viewing with eye movement and no head movement, driver's seat adjustment for pedal locations and reach to accelerator, brake and clutch.

The EDKBES provides the necessary assistance in an area where there is a wealth of legislation. The resources available will be inferred in determining the choice of various entities for standards and legislation, rules and regulations, information on ergonomics standards and specifications, the design working practice to be related to the geometric reasoning aspects of the design process. The design is monitored against these standards and information in the window environment.

PROGRAMMER:	**MD.DAN BIN MD. PALIL.**
FILE NAME:	**dan.kb**
ORIGINAL DATE:	**01 May 1994.**
PURPOSE:	**Ergonomics Design – Integeration EDKBES within SAMMIE system for Interior Vehicle Design running in SUN SPARC workstation on window.**

****

The Wealth of Legislation on Standard and Legislation(s).
The Rule and Regulation the Interior of a Car(s).
The Information on Ergonomics Design of Standards & Specification.

The Design Working Practice on Aspect of Design Process.

****

Knowledge Engineering System:Manual and Reference Guide.
KES Knowledge Base Author's Manual 1986.
Australian Design Rules: Standards Rules and Regulations.
Australian Design Rule 18/00 for Instrumentation 1988.
British Standard Institution: Standards & Informations for Automobile Series.
BS AU 143c_84: Symbols for controls, indicators and tell_tales
for road vehicles.
BS AU 176_80: Establishment of eyellipses for driver's eye location.
BS AU 179_81: Dimensional codes for passenger cars.
BS AU 183_83: Passive seat belt systems.
BS AU 199_84: Location of hand controls,indicators and tell_tales.
European Committee for Standardzation: Standards & Informations.
CEN/TC 122_92:Basic list of definitions of human body dimensions
for technical design.
EDS: Ergonomics of Display Systems for Austin Rover Ltd.
The standards ergonomics of display system for Austin Rover.
INTEREUROPE: Specifications for Motor Vehicle Identification
of Controls/Warning Lights 1979.
EEC: Economy European Community Directives.
Directive 71/127/EEC (amended 79/795/EEC, 82/205/EEC, 88/321/EEC):
Rear-view mirrors of motor vehicles.
Directive 77/649/EEC (amended 81/643/EEC, 88/366/EEC):
Field of vision of motor vehicle drivers.
Directive 78/317/EEC: Windscreen wiper and washer systems.
Directive 78/317/EEC: Defrosting and demisting systems.
FMVSS: Federal Motor Vehicle Safety Standard in USA.
FMVSS 103: Defrost/demist system.
FMVSS 104: Wash/wipe system.
FMVSS 111: Rear view mirrors.
International Standards Organisation: Standards and Informations
for Automobile Series.
ISO 2575_82: Symbols for controls,indicators and tell_tales.
ISO 3958_77: Driver hand control reach_Passenger cars.
ISO 4040_83: Location of hand controls, indicators and tell_tales.
ISO 4513_78: Visibility_Method for establishment of eyellipses for
driver's eye location.
ISO 6385_81: Ergonomics principles in the design of work systems.
System Automotives Engineer: Handbook,Standards & Informations
for Automatives.
The SAE Handbook 1985.
SAE J287_Jun 88: Driver hand control reach.
SAE J879b_68: Motor vehicle seating system.
SAE J941_Oct 85: Motor vehicle driver's range.
SAE J1050_77: Describing and measuring the driver's field of view.
SAE J1138_77: Design criteria_Driver hand controls location for passenger cars.

SAE J1139_77: Supplemental information_Driver Hand control
location for passenger cars.

****

What would you like to do?

1. See a list of displays item developed in the knowledge base
2. See a list of some useful information for design
3. Design a particular interior displays item
4. Check the design for styling
5. Check the type of instruments
6. Check the colour of instruments
7. Quit

=? 1

****

At the moment knowledge base for only a few of these items have been completed.
Speedometer:location for visibility through the st_wheel eye movement
within the ergonomics range and no head movement required for visibility
Tachometer:location for visibility through the st_wheel eye movement
within the ergonomics range and no head movement required for visibility
Fuel_level: ergonomics visibility allowable eye movement
and allowable head movement
Engine_temperature: ergonomics visibility allowable eye movement
and allowable head movement
Door_open: ergonomics visibility allowable eye movement
and allowable head movement
Accelerator: location for leg_reach of driver leg_reach
within the ergonomics range and distance between other pedals
Brake: location for leg_reach of driver leg_reach
within the ergonomics range and distance between other pedals
Clutch: location for leg_reach of driver leg_reach
within the ergonomics range and distance between other pedals

****

Type 'c' to continue, or 's' to stop.
Ready for command: c

What would you like to do?

1. See a list_of displays item developed in_the knowledge base
2. See a list_of some usefull information for design
3. Design a particular interior displays item
4. Check the design for styling
5. Check the type for instruments
6. Check the colour for instruments
7. Quit

=? 2

Which do you required?

1. International data on anthropometric for vehicle design
2. Requirements for the seat system
3. Requirements for the dashboard

4. Requirements for the primary displays
5. Requirements for the display information
6. Requirements for the display panel layout
7. Requirements for the display instruments
8. Requirements for the visibility of_display
9. Requirements for the visibility of_steering wheel
10.Requirements for the SRS Airbag system
11.Requirements for the mirrors
12.Requirements for the leg_reach Heel_point of_pedals
13. Had enough information
=? 1

****

International Data on Anthropometry – Jurgens 1990.
Anthropometric classification of the world population into two categories:–
smaller type and larger type (Jurgens 1990).

Body measurement	Smaller type P5	P95/P5	Larger type P95
Stature	1390	1650	1910
Sitting height	740	870	1000
Eye height, sitting	620	750	880
Forward reach (fingertios)	670	810	950
Shoulder breadth (bideltoid)	320	410	500
Shoulder breadth (biacromial)	285	360	430
Hip breadth (standing)	260	335	410
Knee height	405	505	600
Lower leg length (popliteal height)	320	410	505
Elbow–grip length	270	340	410
Buttock–knee length	450	360	670
Buttock–heel length	830	1010	1190
Hip breadth (sitting)	260	350	440
Hand length	140	170	200
Hand and breadth	65	90	110
Foot length	200	250	300
Head circumference	475	540	600
Head length	160	185	205
Head breadth	120	145	170

****

Type 'c' to continue, or 's' to stop.
Ready for command: c

Which do you required?
1. International data on anthropometric for vehicle design
2. Requirements for the seat system
3. Requirements for the dashboard
4. Requirements for the primary displays
5. Requirements for the display information

6. Requirements for the display panel layout
7. Requirements for the display instruments
8. Requirements for the visibility of_display
9. Requirements for the visibility of_steering wheel
10.Requirements for the SRS Airbag system
11.Requirements for the mirrors
12.Requirements for the leg_reach heel_point of_pedals
13. Had enough information
=? 2

****

Seat System are defined in ISO 3958, 1977, SAE J1100, 1990, SAE J826, 1990, SAE J1517, 1990. Seat can be divided into two categories, performances or touring:-
Performances seats - stiffer with more contour (bucket type) additional adjustable features: such as lateral cushion and back bolsters.
Touring seats tend to be more comfortable, softer and place greater emphasis on comfort and safety. In designing the seats there is a need to suit styling to the vehicle purpose and functions. The geometric features of the seat design can be divided into accomodation and comfort requirements. The vehicle seat design, a task analysis reveals three different occupants in the vehicle: driver seat, front-seat passenger, and rear-seat passenger.

GEOMETRIC FEATURES OF SEAT DESIGN

Seat design can be divided into accomodation and comfort:-
Accomodations - refers to seat size and adjustments for horizontal distance from controls, height and back angle.
Comfort - refers to stiffness, contour, climate, memory and vehicle features that promote occupant comfort. DRIVER SEAT (SAE J1517,1990)
Designing a driver's seat there are four design criteria:

1 Driver seat position with unobstructed vision and within reach of all vehicle controls;
2 The seat must accommodate the driver's size, shape, weight;
3 The seat should be comfortable for extended periods;
4 The seat should provide a safety zone for the driver in an accident or crash.

Seat comfort can be divided into two parts, attitude comfort is a measurement of human posture of seat comfort due to occupant position in the vehicle. The foot, knee, hip and back angles provide an indication of relative position and or vehicle work space. Small hip and knee angles reflect restricted work space. The back angle provides an indication of seat back inclination, and foot angles an indication of toeboard or pedals angle relative to seat position. Seat comfort is the degree of support that a seat provides the occcupant.

- DRIVER SEAT DIMENSIONS -

Driver's seat compartment dimensional ergonomics range:-

Items/Dimension in mm (inches)	BSI/ISO	SAE	Ref.no.
Vertical distance SgRP to heel-pt.	127-130	177(7.0)	H30
Horizontal distance SgRP to heel-pt.	130-508	917(36.1)	L53
Vertical seat adjustable rise	0 - 38	38(1.5)	L59
Normal driver & seat adjustable	102-165	189(7.4)	L23
Design Heel-pt. front travel		189(7.4)	L17
SgRP front X coordinate		1494(58.8)	L31
Head room		938(36.9)	H61

Max. leg room (accel)		1094(43.1)	L34
Shoulder room		1348(53.1)	W3
Hip room		1308(51.1)	W5
Back angle–front	5 – 40 deg.	25 deg.	L40
Hip angle		98 deg.	L42
Knee angle		134 deg.	L44
Foot angle		87 deg.	L46
Upper body opening to ground		1207(47.5)	H50
– PASSENGER SEAT DIMENSIONS –			
SgRP–point couple distance		616(24.3)	L50
Head room		867(34.1)	H63
Min. leg room		688(27.1)	L51
SgRP (second to heel)		265(10.4)	H31
Knee clearance		–112(–4.4)	L48
Shoulder room		1299(51.1)	W4
Hip room		1146(45.1)	W6
Back angle		28,30 deg.	L41
Hip angle		82,45 deg.	L43
Knee angle		65,51 deg.	L45
Foot angle		106,48 deg.	L47
Depressed floor thickness		25	H67

****

Type 'c' to continue, or 's' to stop.
Ready for command: c

Which do you required?
1. International data on anthropometric for vehicle design
2. Requirements for the seat system
3. Requirements for the dashboard
4. Requirements for the primary displays
5. Requirements for the display information
6. Requirements for the display panel layout
7. Requirements for the display instruments
8. Requirements for the visibility of_display
9. Requirements for the visibility of_steering wheel
10.Requirements for the SRS Airbag system
11.Requirements for the mirrors
12.Requirements for the leg_reach heel_point of_pedals
13. Had enough information
=? 3

****

Standards and Legislation for dashboard are defined in [ISO 4040, 1983, BS AU 199, 1984, SAE SP–576, 1984]. Dashboard are consists; primary displays and secondary displays.

The dashboard runs the entire width of the interior front of the vehicle. The standard orientation is at an angle of 15 deg. away from the driver so that the the displays may read with a min. amount of distortion.

Primary displays is an instruments giving an essential information for driver's and there are consists of speedometer (Odometer, Trip-odometer, and Oil pressure), Tachometer (Economy-meter), fuel indicator, Temperature indicator, Door-open indicator, Turn-signal indicator abd Headlamps (Main beam/flasher indicator).

1 Standards and legislation for the selection, design and layout of primary displays are intended to meet the objective of compatibility with the characteristics of human perception as follows:

a) The nature and number of signals and displays design shall be compatible with the characteristics of the information.

b) Clear identification of information where primary displays are numerous. The layout should be simple, spacious and arranged so as to promote clear and rapid orientation,

c) Primary displays should be designed for clear visibility and good visual perception. Account shall be taken, for instance, of the intensity, shape, size and constrast.

d) Rate and direction of change of display of information should be compatibl with rate and direction of change of the primary source of that information.

e) Function of the primary displays shall be identifiable to avoid confusion.

f) Displays and control reach/movements, equipment response, and display information should be mutually compatible.

g) Where controls are numerous there is a need to ensure safe, unambiguous and quick operation. The displays and signals should be grouped according to their functions.

2) ISO and BSI define various zones for the location of primary displays [ISO 4040, 1983, BS AU 199, 1984].

i) Zone one is located on the right side of the display panel and should contain the tachometer, economy-mpg, fuel indicator, and five indicators at below rectangular in shape dimensions 30x45x25 mm are parking-brake indicator, battery condition indicator, hazard-warning indicator, supplementary restraint system (SRS-D) and seat-belt indicator,

ii) Zone two is located in the centre of the display panel and should contain the door-open, turn-signal, and headlamp indicator – upper/lower beam. The remaining parts of the display area shall also be visible; with head movement is required.

iii) Zone three is located on the left side of the display panel and should contain the speedometer, odometer, trip-odometer, oil pressure meter, temperature gauge, and five indicators at below rectangular in shape dimensions 30x45x25 mm are brake-failure indicators, service warning indicator, heater controls indicator, supplementary restraint system (SRS-P) and fog-light indicator.

****

Type 'c' to continue, or 's' to stop.
Ready for command: c

Which do you required?
1. International data on anthropometric for vehicle design
2. Requirements for the seat system
3. Requirements for the dashboard
4. Requirements for the primary displays
5. Requirements for the display information
6. Requirements for the display panel layout
7. Requirements for the display instruments
8. Requirements for the visibility of_display
9. Requirements for the visibility of_steering wheel
10.Requirements for the SRS Airbag system
11.Requirements for the mirrors

12.Requirements for the leg_reach heel_point of_pedals
13. Had enough information
=? 4
****

Standards and legislation for primary display are defined in SAE SP-576 Ergonomics Aspects of Electronic Instrumentation: A Guide for designer and engineers [SAE SP-576, 1984, Galer, 1985]. Displays are intended to show information to the driver to confirm correct function during the operation of controls.

1) Analogue Displays – Analogue displays typically use a needle pointer on a scale to show the value represented. Often it is used to convey qualitative information, and can be enhanced by a red portion of the scale to signify danger. Types of analogue displays include circular dials, linear scales and curvilinear combinations, and typical applications would be the tachometer, and fuel gauge. Analogue displays are generally better than small digital equivalents for quick check reading, and for rate of change and direction information.

2) Discrete Displays – Discrete displays are also analogue displays but the markings of the scale are discrete rather continuous. An example is an 8-segment or discrete sections fuel gauge, providing quantity information but without the detail or accuracy of scalar displays.

3) Digital Displays – With digital displays the information is presented directly as a number. A good example is odometer. Digital displays are better than analogue displays where precise readings and perfect indications are required.

4) Alphanumeric Displays – Alphanumeric displays present information as textual messages in full or abbreviated form e.g. FASTEN SEAT BELT.

5) Representational Displays – Representational displays present information as graphic diagram or working models, such as the plan drawing of the car used as a door-open indicators. The graphic diagrams enable the user to observe the function of items such as doors, bonnet and boot in relation to the whole, and to locate faults quickly, and can for example be used for vehicle diagnostics.
****

Type 'c' to continue, or 's' to stop.
Ready for command: c

Which do you required?
1. International data on anthropometric for vehicle design
2. Requirements for the seat system
3. Requirements for the dashboard
4. Requirements for the primary displays
5. Requirements for the display information
6. Requirements for the display panel layout
7. Requirements for the display instruments
8. Requirements for the visibility of_display
9. Requirements for the visibility of_steering wheel
10.Requirements for the SRS Airbag system
11.Requirements for the mirrors
12.Requirements for the leg_reach heel_point of_pedals
13. Had enough information
=? 5
****

Standards and legislation for display information is specified in SAE SP-576 Ergonomics Aspects of

Electronic Instrumentation: A Guide for designer and engineers [SAE SP-576, 1984, Galer, 1985].
Display information are intended to show information to the driver to be aware during the operation.

1) Warning – Warning information is very important to the safe running of the vehicle. Red is used as a – WARNING – indicator e.g. and brake failure or brake engaged signals use red lights.

i) The driver's attention should be attracted to the warning; the significance of the warning must be apparent through the red colour.

2) Advisory – Advisory information is very useful to the safe running of the vehicle, and is also used to convey vehicle state information, e.g. headlight main beam ON, FASTEN SEAT BELT.

i) The driver's attention should be attracted to the information but it should distract him from the driving task. This covers a wide range of information devices from simple tell tales and indicators, to trip odometers/computers.

3) Diagnostic – Diagnostic information concerns the condition of the vehicle for maintenance purposes, e.g. warning light for battery charging, and – SERVICE – indicator.

i) The driver should be able to choose the appropriate opportunity to assimilate or take action on such diagnostic

4) Entertainment – In some vehicles related information is available via the entertainment facilities, e.g. traffic bulletins transmitted by radio.

i) It is necessary to make sure that other audible forms of information presentation are not masked by the entertainment system.

****

Type 'c' to continue, or 's' to stop.
Ready for command: c

Which do you required?
1. International data on anthropometric for vehicle design
2. Requirements for the seat system
3. Requirements for the dashboard
4. Requirements for the primary displays
5. Requirements for the display information
6. Requirements for the display panel layout
7. Requirements for the display instruments
8. Requirements for the visibility of_display
9. Requirements for the visibility of_steering wheel
10.Requirements for the SRS Airbag system
11.Requirements for the mirrors
12.Requirements for the leg_reach heel_point of_pedals
13. Had enough information
=? 6

****

Standards and legislation for Display Layout. SAE SP-576-84, SP-654-86 and SP-734-88 Automotive Electronic Displays and Information Systems.
Design principles and standards for the layout of a display panels include.

1) Visibility – The driver should be able to see all displays on the panel from normal driving position allowing for some head movement.

2) Identification – It should be easy for the driver to find and identify displays,

3) Grouping – The display instruments should be arranged in functional or sequential groups.

4) Associations – Displays should be arranged to be compatible with the controls to which it is related.

A) Layout for good visibility

1) The plane in which the display lies should be perpendicular to the line of the sight.

2) The driver's view of displays should be unobstructed by the steering wheel, parts of his own body, and windscreen reflection.

3) The distance between displays should be minimised to reduce eye and head movement. However, it is also useful to spatially separate displays to avoid confusion when reading them quickly.

B) Layout for good design and identification

1) All displays should be clearly labelled.

2) Location and separation – Good layout on the panel is one of the best aids to identification. Primary instruments, e.g. speedometer, tachometer and warning indicators should be located in primary display space.

3) Functional grouping – Displays should be grouped in terms of functional use. This reduces the area over which the driver has to search for a particulars display.

4) Standardised location – If possible standardise the location of displays or functional groups of displays.

****

Type 'c' to continue, or 's' to stop.
Ready for command: c

Which do you required?
1. International data on anthropometric for vehicle design
2. Requirements for the seat system
3. Requirements for the dashboard
4. Requirements for the primary displays
5. Requirements for the display information
6. Requirements for the display panel layout
7. Requirements for the display instruments
8. Requirements for the visibility of_display
9. Requirements for the visibility of_steering wheel
10.Requirements for the SRS Airbag system
11.Requirements for the mirrors
12.Requirements for the leg_reach heel_point of_pedals
13. Had enough information
=? 7

****

Standards and legislation for Display Instruments SAE SP-576-84, SP-654-86 and SP-734-88 Automotive Electronic Displays and Information Systems.

Standards and design working practice for the display include.

1) Speedometer

a) The speedometer should be positioned fully seen by all drivers, without eye movements or head movements.

b) Digital displays are recognised as being faster than analogue displays for information acquisition.

c) A combination analogue-digital/electromechanical speedometer display provides good ergonomics and functionality.

d) Analogue displays are not as good as digital displays for rate of change information.

e) Digital displays are read more quickly and accurately than analogue displays.

2) Analogue speedometer

a) The standards circular or semi–circular shape is easier to read.

b) Standard scale markings use a conventional progression system of 10, 20, 30, put detail scale markings at 0, 5, 10, 15, 20, and where appropriate use minor markers for individual numbers.

c) Needle pointers of whatever form must line up with scale markings and position on the scale to be easily read and not distract the driver.

d) The full scale should be available to the driver with the value being indicated by a pointer.

3) Digital speedometer

a) Digital display values must remain visible long enough to be read accurately, (500 – 1000 meters per seconds).

b) Characters should be upright rather than slanted and their height should be 15 – 20 mm.

1.1) Odometer, Trip–odometer, and Oil pressure meter

a) This analogue–digital/electromechanical display is best placed within the speedometer above the trip odometer to avoid clutter and confusion.

b) When illuminated the odometer should be less bright than the speedometer.

c) Trip–odometer analogue–digital display is best placed within the speedometer below the odometer where it can be easily read by the driver.

d) Odometer and Trip–odometer a square shape is recommended and not so large as to interfere with speedometer readings (10 mm minimum character size).

e) The trip odometer should be less bright than the speedometer and the control knob should be easy to reach.

f) Oil pressure meter indicator should use an analogue needle pointer with scale markings and be positioned on the scale to be easily read and so as not to distract the driver.

2) Tachometer

If a tachometer is provided then the display should be permanently in operation when the engine is running.

a) The tachometer should be positioned to be fully seen by all drivers, without eye or head movements.

b) Confusion between the tachometer and the speedometer can be avoided by differentiating by styling, colour, relative brightness, etc.

c) Maximum limits to engine speed should be indicated on the scale in red, yellow or orange.

2.1) Analogue Tachometer

a) The standard shape of circular or semi–circular is recommended for ease of reading.

b) Standard scale markings use a conventional progression system of 1, 2, 3,and where appropriate minor markers are used for individual numbers.

c) Needle pointers of whatever form must line up with scale markings and position on the scale to be easily read and not distract the driver. The full scale should be visible to the driver with the value indicated by a pointer.

2.2) Digital speedometer

a) Digital display values must remain visible long enough to be read accurately, (500 – 1000 meter per seconds).

b) The characters should be upright rather than slanted and their height should be 15–20 mm. 3) Fuel Indicator

a)Fuel indicator should use an analogue or digital display. With an analogue or electromechanical display are shown perfect fuel level.

b) Qualitative markings are required for fuel indicator. Further scale markings such as (E) for empty, 1/2 for half full and (F) for full.

c)Low fuel level warning should be coloured red, orange or yellow.

****

Type 'c' to continue, or 's' to stop.
Ready for command: c

Which do you required?
1. International data on anthropometric for vehicle design
2. Requirements for the seat system
3. Requirements for the dashboard
4. Requirements for the primary displays
5. Requirements for the display information
6. Requirements for the display panel layout
7. Requirements for the display instruments
8. Requirements for the visibility of_display
9. Requirements for the visibility of_steering wheel
10.Requirements for the SRS Airbag system
11.Requirements for the mirrors
12.Requirements for the leg_reach heel_point of_pedals
13. Had enough information
=? 8

****

Standards and Legislative Requirements for Visibility of Display – Eyellipses for driver eye location are defined in BS AU 176, 1980, ISO 4513, 1978, SAE J985, 1967.
The display area of the following display shall be visibility without head movement:

1) ZONE ONE: TACHOMETER – Economy_mpg, FUEL INDICATOR, SAFETY BELT, CHOKE, BATTERY, HAZARD-WARNING and PARKING BRAKE.

Zone one is located on the right side of the display panel. The display area required to indicate that a quarter or less of the maximum stored fuel is available shall be visible without head movement. The remaining parts of the display area shall also be visible; with eye movement and head movement.

– FUEL INDICATOR –

Parts of the display area required to indicate a critical condition shall be visible without head movement for the following indicators:

– Illumated area tell-tales –

Zone one below for tell-tales/indicator dimensions 18x18mm rectangular of the illuminated area shall be visible without head movement; Parking Brake, Hazard-warning, Battery charging, SRS-P Airbag and Seat belt-warning.

2) ZONE TWO: DOOR_OPEN, TURN_SIGNAL INDICATOR, HEADLAMP Upper/Lower Beam

Zone two is located in the centre of the display panel, The remaining parts of the display area shall also be visible; with head movement is required.

3) ZONE THREE: SPEEDOMETER – Odometer, Trip_Odometer, OIL PRESSURE GAUGE, TEMPERATURE GAUGE, BRAKE-FAILURE INDICATORS, SERVICE WARNING, HEATER, SRS_D AIRBAG, and FOG-LIGHT INDICATOR.

Zone three is located on the left side of the display panel, The remaining parts of the display area shall also be visible with eye movements and head movement;

- Illumated area tell-tales -

Zone three below for tell-tales/indicator dimensions 18x18mm rectagular of the illumated area of the following tell-tales/indicator shall be visible without head movements; brake_failure, service_car, heater, SRS_D airbag, fog_light.

Driver vision - visual aspect in vehicle design SAE J985-67 table shown limits of visual field, eye movements and head movements:-

Distance visual eye and head movement.	Horizontal left/right	Vertical up/down
Binocular field	120 deg.	0 deg.
Right Monocular field	150 deg.	0 deg.
Eye rotation min.	30/30 deg.	15/15 deg.
Eye rotation max.	45/65 deg.	45/65 deg.
Head turn min	45/60 deg.	45/60 deg.
Head turn max	45/60 deg.	45/60 deg.
Head tilt	30/30 deg.	30/30 deg.

****

Type 'c' to continue, or 's' to stop.
Ready for command: c

Which do you required?
1. International data on anthropometric for vehicle design
2. Requirements for the seat system
3. Requirements for the dashboard
4. Requirements for the primary displays
5. Requirements for the display information
6. Requirements for the display panel layout
7. Requirements for the display instruments
8. Requirements for the visibility of_display
9. Requirements for the visibility of_steering wheel
10.Requirements for the SRS Airbag system
11.Requirements for the mirrors
12.Requirements for the leg_reach heel_point of_pedals
13. Had enough information
=? 9

****

Standards and Legislation for Steering Wheel Visibility and Location are defined in SAE J941, 1990, SAE J1052, 1990, BS AU 199, 1984, ISO 4040, 1983 and SAE J985, 1967. Established standards and legislation, are described below:

1) Eye-ellipses are perimeters of envelopes formed in side and plan view by an infinite number of planes dividing the eye position so that percentages of eye is one side of the plane and 100 percentages are on the other side. The eye-ellipses and head contour located. A locus of points (in one degree increments) that is used to locate the (x-x and z-z) datum lines of the eye-ellipses and the driver head position contour in horizontally adjustable seat with seat back angles from 5 degrees to 40 degrees.

2) The head contour locator line; with fixed seat, a locus of points is used to locate the passenger head position contours in fixed seats with back angles from 5 degrees to 45 degrees.

3) The standard viewing distance or visibility distance from the driver's eye to the primary displays during driving will vary to some small extent due to eye or head movements. The standard viewing

distance in cars is about 750 mm. For drivers wearing bifocal lenses, the reading lens is usually focused at 300mm with the distance lens at infinity.

4) The angle between the line of sight and a perpendicular to the primary display screen is called the viewing angle or acceptable viewing angle. The acceptable viewing angle for legibility of displays is affected by ambient illumination, screen curvature, use of lenses, contrast, resolution and character size. A relatively wide range of acceptable viewing angles generally accepted in ergonomics practice: 15 degrees – comfortable viewing angle. 30 degrees – maximum acceptable angle.

The following list established from standards and legislation, relates to the steering wheel dimensions: Item/Dimension in mm (inches) BS/ISO SAE Ref.No.

Item/Dimension in mm (inches)	BS/ISO	SAE	Ref.No.
Steering wheel dia.	330–410	375(14.8)	W9
Steering wheel weight	25 – 35 lbs	25 – 40 lbs W9	
Steering wheel angle	10 – 70 deg.	20 deg.	H18
Whl.cntr.to accel.H-pt.Horiz.	152–660	506(19.9)	L11
Whl.cntr.to accel.H-pt.Vert.	530–838	548(21.6)	H17
Steering wheel rim size	300mm and 460mm.(women 300mm.) dia.,		

Resistance 5 to 30 pounds. The steering wheel position angles at 10 to 70 deg. and displacement limited to about a 120 deg. turn, no remove hands during turning.

****

Type 'c' to continue, or 's' to stop.

Ready for command: c

Which do you required?

1. International data on anthropometric for vehicle design
2. Requirements for the seat system
3. Requirements for the dashboard
4. Requirements for the primary displays
5. Requirements for the display information
6. Requirements for the display panel layout
7. Requirements for the display instruments
8. Requirements for the visibility of_display
9. Requirements for the visibility of_steering wheel
10.Requirements for the SRS Airbag system
11.Requirements for the mirrors
12.Requirements for the leg_reach heel_point of_pedals
13. Had enough information

=? 10

****

Standards and Legislation of Supplementary Restraint System (SRS) Airbag Systems as references to Federal Motor Vehicle Safety Standard (FMVSS) and Ford DSE [FMVSS 208, 1990, Ford DSE, 1994]. Design principles for SRS airbag systems, safety features, specifications and design practice are describe below [Ford DSE, 1994]:

1) The SRS airbag systems, or so called Supplementary Restraint System is accommodated in the padded boss at centre of the steering wheel and driver's seat is provided with anti-submarine ramps attached together with seat belt grabbers and pretensioners (Restraint System) to provide a new level of fast response safety that gives additional

protection to the driver's head, face and chest.

2) The SRS airbag system for front passenger seat is accommodated in a padded boss at centre of the

dashboard drawer and the seat is similarly provided with anti-submarine ramps, seat belt grabbers and pretensioners.

GEOMETRIC FEATURES OF AIRBAG SYSTEM

1) All steering wheels should have safety protection with SRS airbag systems for the driver and seats should have anti-submarine ramps with seat belt fastener, seat belt grabber and pretensioners,

2) The front passenger seat should have SRS airbag system on the dashboard drawer and the seat should have an anti-submarine ramp with seat belt fastener, seat belt grabber and pretensioners,

3) Rear passenger seats should have anti-submarine ramps with seat fastener, seat belt grabber and pretensioners,

4) The internal structure including all four doors should incorporate a cross-car beams and a safety cell or cage.

5) The standard materials for airbags is neoprene lined textile with an internal coating of silicone to protect the inner surface from the gas generator used to inflate the airbag. Silicone materials are used as they are thin, and light than the neoprene and offer a much improved packing density [Ford DSE, 1994].

6) SRS-D AIRBAG should be stamped on the padded boss of the steering wheel and an SRS-D indicator light should be provided on the right of the instrument cluster.

7) For the front passenger seat the lettering SRS-P AIRBAG is stamped on the dashboard drawer of the front passenger seat and an SRS-P indicator light should be provided on the left of the instrument cluster. Design principles for SRS airbag systems, safety features,
specifications and design practice are describe below [Ford DSE, 1994]:

1) The SRS airbag systems, or so called Supplementary Restraint System is accommodated in the padded boss at centre of the steering wheel and driver's seat is provided with anti-submarine ramps attached together with seat belt grabbers and pretensioners (Restraint System) to provide a new level of fast response safety that gives additional protection to the driver's head, face and chest.

2) The SRS airbag system for front passenger seat is accommodated in a padded boss at centre of the dashboard drawer and the seat is similarly provided with anti-submarine ramps, seat belt grabbers and pretensioners. Design principles for SRS airbag systems described below:-

FUNCTIONS OF AIRBAG SYSTEM

1) Firstly, the operational readiness of the Restraint System (airbag) is indicated by the SRS indicator light in the instrument cluster. If the key in the steering lock is turned to position 1 or 2, the indicator light stays on for approximately 4 seconds. Should an impact at the steering wheel the airbag will fully inflate the moment a signal is received from the sensor, make contact to protect the driver, then deflate as it absorbs the impact, all in about 50 milliseconds. If the indicator light fails to come on when starting the car or comes on while driving there is a fault in the system. The SRS airbag, however, is not activated by this fault. The airbag is so designed as to be activated only in severe head on collisions. The driver should have fastened his belt as otherwise the airbag cannot provide the envisaged protection.

2) Secondly during the activation of the SRS airbag a small volume of air will be released. Then the seat belt inertia reels are locked and in addition other sensor will have activated clamps which grab and hold the belts to minimise any paying out due to the effect of spooling.

In a more severe impact the pretensioners work in conjunction with the grabbers to further enhance seat belt efficiency by pulling the belt buckles downwards to reduce any slack in the diagonal and lap belts.

3) Front and rear seats have anti-submarine ramps so as to reduce sliding forward under the seat belt during impact. In the most severe impact the steering wheel Airbag will inflate and deflate as it absorbs the impact. The air is neither injurious to health nor does it indicate a fire in the vehicle.

****

Type 'c' to continue, or 's' to stop.
Ready for command: c

Which do you required?
1. International data on anthropometric for vehicle design
2. Requirements for the seat system
3. Requirements for the dashboard
4. Requirements for the primary displays
5. Requirements for the display information
6. Requirements for the display panel layout
7. Requirements for the display instruments
8. Requirements for the visibility of_display
9. Requirements for the visibility of_steering wheel
10.Requirements for the SRS Airbag system
11.Requirements for the mirrors
12.Requirements for the leg_reach heel_point of_pedals
13. Had enough information
=? 11

****

Mirrors – ISO 6549, SAE J941, J985–1967, J826–1987, J1050–1977.
Field of vision of motor vehicle drivers – Directive 77/649/EEC
Rear view mirror of motor vehicles – Directive 71/127/EEC
Windscreen wiper and washer systems – Directive 78/317/EEC
Rear view mirror – FMVSS 111 and Wash/wipe system – FMVSS 104
– DRIVER MIRROR VIEW –
Driver mirror view standards, legislations and specifications:

1 Define the feild of mirror view provide for driver, establishing any obstructions and obscurations caused by the body structure and fixtures such as windsreen wipers or by the internal centre rear mirrors;

2 Design suitable interior and exterior mirror, and to determine where they are best mounted to maximize the field of view while avoiding unnecessary obscuration of the direct view. The regulations for interior CR mirror (ICRM) should not further than 60 m or 200 ft to the rear of the driver's from rear of vehicle (EEC, FMVSS). The mirror must have sufficient width to give the driver 20 m or 20 deg. wide view horizontal angle view beyond these points (EEC, FMVSS).

The regulation for exterior SR mirror (ESRM) on the driver's side, required field extend rearward from vertical plane (ground level and horizon) 10 m or 35 ft. behind the driver's eyes (EEC, FMVSS) and is bounded by a plane through the widest point on the side of the vehicle and a second plane 2.5m or 8ft. away from this (EEC, FMVSS). Driver visibility – visual aspect in interior vehicle design limits of visual field, eye movement and head movement are defined in SAE J985 1967:-

Distance visual eye & head movemt.	Horizontal left/right	Vertical up/down
Binocular field	120 deg.	0 deg.
Right Monocular field	150 deg.	0 deg.
Eye rotation min.	30/30 deg.	15/15 deg.
Eye rotation max.	45/65 deg.	45/65 deg.
Head turn min	45/60 deg.	45/60 deg.
Head turn max	45/60 deg.	45/60 deg.

Head tilt 30/30 deg. 30/30 deg.

****

Type 'c' to continue, or 's' to stop.
Ready for command: c

Which do you required?
1. International data on anthropometric for vehicle design
2. Requirements for the seat system
3. Requirements for the dashboard
4. Requirements for the primary displays
5. Requirements for the display information
6. Requirements for the display panel layout
7. Requirements for the display instruments
8. Requirements for the visibility of_display
9. Requirements for the visibility of_steering wheel
10.Requirements for the SRS Airbag system
11.Requirements for the mirrors
12.Requirements for the leg_reach heel_point of_pedals
13. Had enough information
=? 12

****

Road Vehicles Passenger Cars Driver Hand Controls Reach, ISO 3958, 1977, Leg_reach Heel_point of pedals reach distance. The following location for leg_reach Heel-point of_pedal:

– ACCELERATOR PEDAL –

The seating refrence point is terms of the R-point. The pivot centre of the torso line and thigh centreline of the two or three-dimensional H-point machine template with 95% leg length used to describe vehicle seating geometry. The accelerator heel-point or H-point device with floor covering; the foot angle of the device is restricted to not less than 87 deg. Horizontal dimensional from the R-point to the driver H-point (Hx). The vertical dimensions from R-point to driver H-point (Hz).

– BRAKE PEDAL –

The brake heel-point or H-point device height 3–10 inches max., 2 inches max. travel with 15 lbs.max. force brake. The distance between other pedal 2 inches min. gap.

– CLUCTH PEDAL –

The clucth heel-point or H-point device height 3–10 inches max., 4 inches max. travel with 80–90 lbs.max. force brake. The distance between other pedal 2 inches min. gap.

The following list establishes the ranges of the operator workspace dimensions:

Item/Dimension in mm (inches)	BS/ISO	Ref.No
Max. leg room (accel)	1094	L34
Hip angle	98 deg.	L42
Knee angle	134 deg.	L44
Foot angle	87 deg.	L46
Back angle (B)	9/33 deg.	L40
Vertical R_point to H-point(Hz)	130/520	H30
Horizontal R-point travel	130	L53

****

Type 'c' to continue, or 's' to stop.
Ready for command: c

Which do you required?
1. International data on anthropometric for vehicle design
2. Requirements for the seat system
3. Requirements for the dashboard
4. Requirements for the primary displays
5. Requirements for the display information
6. Requirements for the display panel layout
7. Requirements for the display instruments
8. Requirements for the visibility of_display
9. Requirements for the visibility of_steering wheel
10.Requirements for the SRS Airbag system
11.Requirements for the mirrors
12.Requirements for the leg_reach Heel_point of_pedals
13. Had enough information
=? 13

What would you like to do?
1. See a list of displays item developed in the knowledge base
2. See a list of some useful information for design
3. Design a particular interior displays item
4. Check the design for styling
5. Check the type of instruments
6. Check the colour of instruments
7. Quit
=? 3

Which interior displays would you like to design?
It would be a good idea to go through each stage systematically.
1. Interior Front
2. Mirror View
3. Seats
4. New case
5. Leave this menu
=? 1

Which interior front would you like to design?
It would be a good idea to go through each stage systematically.
1. Dashboard
2. Primary Controls
3. New case
4. Leave this menu
=? 1

Which dashboard would you like to design?
It would be a good idea to go through each stage systematically.
1. Primary Displays
2. Secondary Displays
3. New case
4. Leave this menu
=? 1

Which primary displays would you like to design?
It would be a good idea to go through each stage systematically.
1. Speedometer
2. Tachometer
3. Fuel
4. Temperature
5. Door_Open
6. Right Indicators
7. SRSD Airbag
8. Left Indicators
9. Seat Belt
10. New case
11. Leave this menu
=? 1

what would you like to start with?
It would be a good idea to go through each stage systematically.
1. Check visibility through the steering wheel
2. Check eye movement within the ergonomics range
3. Check that no head movement required for visibility
4. Finished with this menu
=? 1

The members of the class 'speedo' have already been determined.
The members of the class 'st_wheel' have already been determined.
The members of the class 'visible' have already been determined.
****
for values of vertices given :
top vertex tp = 65
bottom vertex bp = 156
right vertex rp = 144
left vertex lp = 100
the speedometer is not visible through st_wheel <1.00>
****
would you like to see the reason for this?
1. yes
2. no
=? 2
****
the adjustment for speedometer position is to :
move down at least by a min 5 mm
move up at least by a min -4 mm
move right at least by a min 4.7 mm
move left at least by a min -5 mm
****
Type 'c' to continue, or 's' to stop
Ready for command: c

what would you like to start with?
It would be a good idea to go through each stage systematically.
1. Check visibility through the steering wheel
2. Check eye movement within the ergonomics range
3. Check that no head movement required for visibility
4. Finished with this menu
=? 2

The members of the class 'view_with' have already been determined.
****
for angles of eye movement given:

horizontal left =	22
horizontal right =	20
vertical upwards =	15
vertical downwards =	18

these angles of eye movement are: outside gI eye limit <1.00>
****
would you like to see the reason for this?
1. yes
2. no
=? 1
****
The value of view_with:eye move>gI eye limit = outside gI eye limit <1.00>.
This is due to the following knowledge sources:
  rule: groupI eye move outside limit

Would you like to see the supporting knowledge sources and demons? (y/n) y

Name: groupI eye move outside limit
Kind of entity: Production Rule

```
groupI eye move outside limit:
v_w: view_with
    if
        view_with:eye move > horizontal left gt 20 or
        view_with:eye move > horizontal right lt -20 or
        view_with:eye move > vertical upwards gt 10 or
        view_with:eye move > vertical downwards lt -35
    then
        v_w>gI eye limit = outside gI eye limit
    endif
```

{references: ADR}
{explanation: "AUSTRALIAN DESIGN RULE 18/00 (1988)   ",
"All 'Visual Indicators' specified as GroupI in Clause 18.2 shall",
"be totally located between 2 vertical planes inclined at 20 deg.",
"left and 20 deg. right of the longitudinal axis of the vehicle ",
"and passing through the for most points of the left and right   ",
"95th Percentile Eye Ellipses respectively. Such indicators shall",
"be totally located above a plane inclined downwards at 35 deg. ",

"from the horizontal and including a horizontal transverse line ",
"through the for most points of each of the '95th Percentile Eye ",
"Ellipses' and below a plane tangential to the bottom of the 95th",
"Percentile Eye Ellipses which includes a line at ground level ",
"transverse to the longitudinal axis of the vehicle 11m forward ",
"of the rear most eye ellips point. "].

****

You can adjust the position of the speedometer so that it
lies within the allowable ergonomics eye movement by :
move to right to reduce angle of eye movement by : 2 deg
move down to reduce angle of eye movement by : 5 deg

****

Type 'c' to continue, or 's' to stop.
Ready for command: c

what would you like to start with?
It would be a good idea to go through each stage systematically.
1. Check visibility through the steering wheel
2. Check eye movement within the ergonomics range
3. Check that no head movement required for visibility
4. Finished with this menu
=? 3

The members of the class 'view_with' have already been determined.

****

for angles of head movement given:
horizontal left = 14
horizontal right = 11
vertical upwards = 14
vertical downwards = 11
these angles of head movement are: outside gI head limit <1.00>

****

would you like to see the reason for this?
1. yes
2. no
=? 1

****

The value of view_with:head move>gI head limit = outside gI head limit <1.00>.
This is due to the following knowledge sources:
rule: groupI head move outside limit

Would you like to see the supporting knowledge sources and demons? (y/n) y

Name: groupI head move outside limit
Kind of entity: Production Rule

```
 groupI head move outside limit:
v_w: view_with
     if
          view_with:head move > horizontal left gt 10 or
          view_with:head move > horizontal right gt 10 or
          view_with:head move > vertical upwards gt 10 or
          view_with:head move > vertical downwards gt 10
     then
          v_w>gI head limit = outside gI head limit.
     endif
{references: ISO}
     {explanation: "The display area of speedometer shall ",
          "     be visible  without head movement   ",
          "     GroupI 'Visual Indicators' shall be  ",
          "     visible without head movement.      "}.
*****************************************************************
You can adjust the position of the speedometer so that it
lies within the allowable ergonomics head movement by :
move to left to reduce angle of head movement by :       4 deg
move to right to reduce angle of head movement by :     -21 deg
move down to reduce angle of head movement by :         4 deg
move up to reduce angle of head movement by :           -21 deg
*****************************************************************
Type 'c' to continue, or 's' to stop.
Ready for command: c

what would you like to start with?
It would be a good idea to go through each  stage systematically.
1. Check visibility through the steering wheel
2. Check eye movement within the ergonomics range
3. Check that no head movement required for visibility
4. Finished with this menu
=? 4

Which primary displays would you like to design?
It would be a good idea to go through each stage systematically.
1. Speedometer
2. Tachometer
3. Fuel
4. Temperature
5. Door_Open
6. Right Indicators
7. SRSD Airbag
8. Left Indicators
9. Seat Belt
10. New case
```

11. Leave this menu
=? 2

what would you like to start with?
It would be a good idea to go through each stage systematically.
1. Check visibility through the steering wheel
2. Check eye movement within the ergonomics range
3. Check that no head movement required for visibility
4. Finished with this menu
=? 1

The members of the class 'tacho' have already been determined.
The members of the class 'st_wheel' have already been determined.
The members of the class 'visible' have already been determined.
****
for values of vertices given :
top vertex tp = 110
bottom vertex bp = 120
right vertex rp = 110
left vertex lp = 110
the tachometer is not visible through st_wheel <1.00>
****
would you like to see the reason for this?
1. yes
2. no
=? 1

****
The value of 'visibility' has not been determined.
****
the adjustment for tachometer position is to :
move the left point to the right by : 15 mm
move the right point to the left by : 10 mm
****
Type 'c' to continue, or 's' to stop.
Ready for command: c

what would you like to start with?
It would be a good idea to go through each stage systematically.
1. Check visibility through the steering wheel
2. Check eye movement within the ergonomics range
3. Check that no head movement required for visibility
4. Finished with this menu
=? 3

The members of the class 'view_with' have already been determined.
****
for angles of head movement given:

horizontal left = 15
horizontal right = 20
vertical upwards = 25
vertical downwards = 40
these angles of head movement are: outside gI head limit <1.00>
****
would you like to see the reason for this?
1. yes
2. no
=? 1

****
The value of view_with:head move>gI head limit = outside gI head limit <1.00>.
This is due to the following knowledge sources:
rule: groupI head move outside limit

Would you like to see the supporting knowledge sources and demons? (y/n) y

Name: groupI head move outside limit
Kind of entity: Production Rule

```
groupI head move outside limit:
v_w: view_with
    if
        view_with:head move > horizontal left gt 10 or
        view_with:head move > horizontal right gt 10 or
        view_with:head move > vertical upwards gt 10 or
        view_with:head move > vertical downwards gt 10
    then
        v_w>gI head limit = outside gI head limit.
    endif

{references: ISO}
    {explanation: "The display area of speedometer shall ",
        "    be visible  without head movement    ",
        "    GroupI 'Visual Indicators' shall be  ",
        "    visible without head movement.  "}.
```

****
You can adjust the position of the tachometer so that it
lies within the allowable ergonomics head movement by :-
move to left to reduce angle of head movement by : 5 deg
move to right to reduce angle of head movement by : -30 deg
move down to reduce angle of head movement by : 15 deg
move up to reduce angle of head movement by : -50 deg
****
Type 'c' to continue, or 's' to stop.
Ready for command: c

what would you like to start with?
It would be a good idea to go through each stage systematically.
1. Check visibility through the steering wheel
2. Check eye movement within the ergonomics range
3. Check that no head movement required for visibility
4. Finished with this menu
=? 4

Which primary displays would you like to design?
It would be a good idea to go through each stage systematically.
1. Speedometer
2. Tachometer
3. Fuel
4. Temperature
5. Door_Open
6. Right Indicators
7. SRSD Airbag
8. Left Indicators
9. Seat Belt
10. New case
11. Leave this menu
=? 3

what would you like to check?
1. Check ergonomics visibility
2. Check allowable eye movement
3. Check allowable head movement
4. end this menu
=? 1

The members of the class 'fuel' have already been determined.
The members of the class 'st_wheel' have already been determined.
The members of the class 'visible' have already been determined.
****
for values of vertices given :
top vertex tp = 120
bottom vertex bp = 120
right vertex rp = 115
left vertex lp = 125
the fuel is outside ergonomics limits <1.00>
****
would you like to see the reason for this?
1. yes
2. no
=? 1
****
The value of fuel:area>visibility = outside ergonomics limits <1.00>.
This is due to the following knowledge sources:

rule: fuel invisibility

Would you like to see the supporting knowledge sources and demons? (y/n) y

```
Name: fuel invisibility
Kind of entity: Production Rule
fuel invisibility: fu:fuel
    if
        fuel:tp>coz gt fuel:tp>upper boundary or
        fuel:bp>coz lt fuel:bp>lower boundary or
        fuel:lp>cox gt fuel:lp>left boundary or
        fuel:rp>cox lt fuel:rp>right boundary
    then
        fuel:area>visibility = outside ergonomics limits.
    endif
```

{references: ADR, SAE}
{explanation: "1) AUSTRALIAN DESIGN RULE 18/00 (1988) : In the case ",
"groupII Visual Indicators', (fuel is considered to be in groupII) the ",
"steering wheel rim including its supporting arms and attachnements there",
"should not constitute an obstruction. ",
"2) SAE SP-576-84, SP-654-86 and SP-734-88, SAE J941-90, SAE J1052-90, ",
" BS AU 199-84, ISO 4040-83 Automotive Electronic Displays and ",
" Information Systems. Design principles for steering wheel visibility ",
"and locations. The display panel area shall be visible through the ",
"steering wheel without eye or head movement. Visibility through the ",
"steering wheel can be divided into three zones, fuel in zone one: ",
" i) Zone one is located on the right side of the display panel and ",
" typically contains the tachometer, economy-meter, fuel indicator, ",
" safety belt indicator, choke, battery condition indicator, ",
" hazard-warning, and parking brake. The display area required to ",
" indicate that a quarter or less of the maximum stored fuel is ",
" available shall be visibility within ergonomics limits. ",
"3) Fuel Indicator- Fuel indicator should use an analogue display. With an",
" analogue or electromechanical display are shown perfect fuel level ",
" Qualitative markings are required for fuel indicator. Further scale ",
" markings such as (E) for empty, 1/2 for half full and (F) for full. ",
" Low fuel level warning should be coloured red, orange or yellow. "}.

****

the adjustment for fuel position is to :
lower the top point by : 15 mm
move the left point to the right by : 25 mm
move the right point to the left by : 35 mm

****

Type 'c' to continue, or 's' to stop.
Ready for command: c

```
what would you like to check?
1. Check ergonomics visibility
2. Check allowable eye movement
3.Check allowable head movement
4. end this menu
=? 2

The members of the class 'view_with' have already been determined.
*******************************************************************
for angles of eye movement given:
horizontal left =          55
horizontal right =         10
vertical upwards =         60
vertical downwards =       20
these angles of eye movement are: outside gII eye limit <1.00>
*******************************************************************
would you like to see the reason for this?
1. yes
2. no
=? 1
*******************************************************************
The value of view_with:eye move>gII eye limit = outside gII eye limit <1.00>.
This is due to the following knowledge sources:
  rule: groupII eye move outside limit

Would you like to see the supporting knowledge sources and demons? (y/n) y

Name: groupII eye move outside limit
Kind of entity: Production Rule
groupII eye move outside limit:
v_w: view_with
      if
          view_with:eye move > horizontal left gt 40 or
          view_with:eye move > horizontal right lt -25 or
          view_with:eye move > vertical upwards gt 10 or
          view_with:eye move > vertical downwards lt -35
     then
          v_w>gII eye limit = outside gII eye limit.
     endif

{references: ADR}
{explanation: "AUSTRALIAN DESIGN RULE 18/00 (1988)            ",
"All 'Visual Indicators' specified as GroupII in Clause 18.2 shall",
"be totally located between 2 vertical planes inclined at 40 deg. ",
"left and 25 deg. right of the longitudinal axis of the vehicle  ",
"and passing through the for most points of the left and right  ",
"95th Percentile Eye Ellipses respectively. Such indicators shall ",
```

"be totally located above a plane inclined downwards at 35 deg. ",
"from the horizontal and including a horizontal transverse line ",
"through the for most points of each of the '95th Percentile Eye ",
"Ellipses' and below a plane tangential to the bottom of the 95th ",
"Percentile Eye Ellipses which includes a line at ground level ",
"transverse to the longitudinal axis of the vehicle 11m forward ",
"of the rear most eye ellips point. "}.
****
You can adjust the position of the fuel so that it
lies within the allowable ergonomics eye movement by :
move to right to reduce angle of eye movement by : 15 deg
move down to reduce angle of eye movement by : 50 deg
****
Type 'c' to continue, or 's' to stop.
Ready for command: c

what would you like to check?
1. Check ergonomics visibility
2. Check allowable eye movement
3. Check allowable head movement
4. end this menu
=? 3

The members of the class 'view_with' have already been determined.
****
for angles of head movement given:
horizontal left = 55
horizontal right = 50
vertical upwards = 20
vertical downwards = 25
these angles of head movement are: outside gII head limit <1.00>
****
would you like to see the reason for this
1. yes
2. no
=? 1
****
The value of view_with:head move>gII head limit = outside gII head limit <1.00>.
This is due to the following knowledge sources:
  rule: groupII head move outside limit

Would you like to see the supporting knowledge sources and demons? (y/n) y

Name: groupII head move outside limit
Kind of entity: Production Rule
groupII head move outside limit:
v_w: view_with

```
 if
      view_with:head move > horizontal left gt 15 or
      view_with:head move > horizontal right lt -15 or
      view_with:head move > vertical upwards gt 15 or
      view_with:head move > vertical downwards lt -15
 then
      v_w>gII head limit = outside gII head limit.
 endif

{references: ISO}
   {explanation:  "The identification and those parts of ",
        "     the display area required to indicate ",
        "     a critical condition shall be visible ",
        "     without head movement.          ",
        "     The remaining parts of the display   ",
        "     shall also be 'visible'; for these,  ",
        "     head movement is permited.          "}.
********************************************************************
You can adjust the position of the fuel so that it
lies within the allowable ergonomics head movement by :
move to right to reduce angle of head movement by :     40 deg
move down to reduce angle of head movement by :         5 deg
********************************************************************
Type 'c' to continue, or 's' to stop.
Ready for command: c

what would you like to check?
1. Check ergonomics visibility
2. Check allowable eye movement
3. Check allowable head movement
4. end this menu
=? 4

Which primary displays would you like to design?
It would be a good idea to go through each stage systematically.
1. Speedometer
2. Tachometer
3. Fuel
4. Temperature
5. Door_Open
6. Right Indicators
7. SRSD Airbag
8. Left Indicators
9. Seat Belt
10. New case
11. Leave this menu
=? 4
```

what would you like to check?
1. Check ergonomics visibility
2. Check allowable eye movement
3. Check allowable head movement
4. end this menu
=? 1

The members of the class 'temperature' have already been determined.
The members of the class 'st_wheel' have already been determined.
The members of the class 'visible' have not been determined.
****
for values of vertices given :
top vertex tp = 120
bottom vertex bp = 120
right vertex rp = 115
left vertex lp = 125
the temperature is outside ergonomics limits <1.00>
****
would you like to see the reason for this?
1. yes
2. no
=? 1
****
The value of temperature:area>visibility = outside ergonomics limits <1.00>.
This is due to the following knowledge sources:
rule: temperature invisibility

Would you like to see the supporting knowledge sources and demons? (y/n) y

Name: temperature invisibility
Kind of entity: Production Rule
temperature invisibility:
te:temperature

```
    if
        temperature:tp>coz gt temperature:tp>upper boundary or
        temperature:bp>coz lt temperature:bp>lower boundary or
        temperature:lp>cox gt temperature:lp>left boundary or
        temperature:rp>cox lt temperature:rp>right boundary
    then
        temperature:area>visibility = outside ergonomics limits.
    endif
```

{references: ADR,SAE}
{explanation: "1) AUSTRALIAN DESIGN RULE 18/00 (1988) : In the case ",
"groupII Visual Indicators', (temperature is considered to be in groupII)",
"the steering wheel rim including its supporting arms and attachnements ",
"there should not constitute an obstruction. ",

"2) SAE SP-576-84, SP-654-86 and SP-734-88, SAE J941-90, SAE J1052-90, ",
" BS AU 199-84, ISO 4040-83 Automotive Electronic Displays and ",
" Information Systems. Design principles for steering wheel visibility ",
"and locations. The display panel area shall be visible through the ",
"steering wheel without eye or head movement. Visibility through the ",
"steering wheel can be divided into three zones, temp in zone three: ",
" iii)Zone three is located on the left side of the display panel and ",
" typically contains; speedometer, odometer, trip-odometer, oil ",
" pressure gauge, temperature gauge, brake-failure indicator, service",
" warning, heater and fog-light indicator. Temperature indicator the",
" remaining parts of the display area shall be also visibility within",
" ergonomics limits. ",
"3) Temperature Indicator – Temperature indicator should use an analogue ",
" display. With an analogue or electromechanical display are shown ",
" perfect temperature indicator. Qualitative markings are required for ",
" temperature indicator. Further scale markings such as (H) for high, ",
" (M) for medium and (L) for low. High temperature indicator warning ",
" should be coloured red, orange or yellow. "].

****

the adjustment for temperature position is to :
lower the top point by : -30 mm
move the left point to the right by : 75 mm
move the right point to the left by : 17.5 mm

****

Type 'c' to continue, or 's' to stop.
Ready for command: c

what would you like to check?
1. Check ergonomics visibility
2. Check allowable eye movement
3. Check allowable head movement
4. end this menu
=? 2
The members of the class 'view_with' have already been determined.

****

for angles of eye movement given:
horizontal left = 55
horizontal right = 10
vertical upwards = 60
vertical downwards = 20
these angles of eye movement are: outside gII eye limit <1.00>

****

would you like to see the reason for this?
1. yes
2. no
=? 1

****

The value of view_with:eye move>gII eye limit = outside gII eye limit <1.00>.
This is due to the following knowledge sources:
  rule: groupII eye move outside limit

Would you like to see the supporting knowledge sources and demons? (y/n) y

```
Name: groupII eye move outside limit
Kind of entity: Production Rule
groupII eye move outside limit:
v_w: view_with
     if
          view_with:eye move > horizontal left gt 40 or
          view_with:eye move > horizontal right lt -25 or
          view_with:eye move > vertical upwards gt 10 or
          view_with:eye move > vertical downwards lt -35
     then
          v_w>gII eye limit = outside gII eye limit.
     endif

{references: ADR}
{explanation: "AUSTRALIAN DESIGN RULE 18/00 (1988)            ",
"All 'Visual Indicators' specified as GroupII in Clause 18.2 shall",
"be totally located between 2 vertical planes inclined at 40 deg. ",
"left and 25 deg. right of the longitudinal axis of the vehicle   ",
"and passing through the for most points of the left and right   ",
"95th Percentile Eye Ellipses respectively. Such indicators shall ",
"be totally located above a plane inclined downwards at 35 deg.  ",
"from the horizontal and including a horizontal transverse line  ",
"through the for most points of each of the '95th Percentile Eye  ",
"Ellipses' and below a plane tangential to the bottom of the 95th ",
"Percentile Eye Ellipses which includes a line at ground level   ",
"transverse to the longitudinal axis of the vehicle 11m forward  ",
"of the rear most eye ellips point.                  "}.
```

****

You can adjust the position of the tmperature so that it
lies within the allowable ergonomics eye movement by :
move to right to reduce angle of eye movement by :   15 deg
move down to reduce angle of eye movement by :   50 deg

****

Type 'c' to continue, or 's' to stop.
Ready for command: c

what would you like to check?
1. Check ergonomics visibility
2. Check allowable eye movement
3. Check allowable head movement
4. end this menu

```
=? 3

The members of the class 'view_with' have already been determined.

*******************************************************************
for angles of head movement given:
horizontal left =          30
horizontal right =         70
vertical upwards =         40
vertical downwards =       60
these angles of head movement are: outside gII head limit <1.00>
*******************************************************************

would you like to see the reason for this
1. yes
2. no
=? 1

*******************************************************************
The value of view_with:head move>gII head limit = outside gII head limit <1.00>.
This is due to the following knowledge sources:
  rule: groupII head move outside limit

Would you like to see the supporting knowledge sources and demons? (y/n) y

Name: groupII head move outside limit
Kind of entity: Production Rule
groupII head move outside limit:
v_w: view_with
      if
          view_with:head move > horizontal left gt 15 or
          view_with:head move > horizontal right lt -15 or
          view_with:head move > vertical upwards gt 15 or
          view_with:head move > vertical downwards lt -15
    then
          v_w>gII head limit = outside gII head limit.
    endif

{references: ISO}
    {explanation: "The identification and those parts of ",
         "    the display area required to indicate ",
         "    a critical condition shall be visible ",
         "    without head movement.              ",
         "    The remaining parts of the display   ",
         "    shall also be 'visible'; for these,  ",
         "    head movement is permited.           "}.
*******************************************************************
```

You can adjust the position of the temperature so that it
lies within the allowable ergonomics head movement by :
move to right to reduce angle of head movement by : 15 deg
move down to reduce angle of head movement by : 25 deg
****

Type 'c' to continue, or 's' to stop.
Ready for command: c

what would you like to check?
1. Check ergonomics visibility
2. Check allowable eye movement
3. Check allowable head movement
4. end this menu
=? 4

Which primary displays would you like to design?
It would be a good idea to go through each stage systematically.
1. Speedometer
2. Tachometer
3. Fuel
4. Temperature
5. Door_Open
6. Right Indicators
7. SRSD Airbag
8. Left Indicators
9. Seat Belt
10. New case
11. Leave this menu
=? 5

what would you like to check?
1. Check ergonomics visibility
2. Check allowable eye movement
3. Check allowable head movement
4. end this menu
=? 1

The members of the class 'door_open' have already been determined.
The members of the class 'st_wheel' have already been determined.
The members of the class 'visible' have already been determined.
****
for values of vertices given :
top vertex tp = 150
bottom vertex bp = 160
right vertex rp = 140
left vertex lp = 130

the door_open is outside ergonomics limits <1.00>
****

would you like to see the reason for this?
1. yes
2. no
=? 1
****

The value of door_open:area>visibility = outside ergonomics limits <1.00>.
This is due to the following knowledge sources:
  rule: door_open invisibility

Would you like to see the supporting knowledge sources and demons? (y/n) y

```
Name: door_open invisibility
Kind of entity: Production Rule
door_open invisibility:
dr:door_open
     if
          door_open:tp>coz gt door_open:tp>upper boundary or
          door_open:bp>coz lt door_open:bp>lower boundary or
          door_open:lp>cox gt door_open:lp>left boundary or
          door_open:rp>cox lt door_open:rp>right boundary
     then
          door_open:area>visibility = outside ergonomics limits.
     endif

{references: ADR, SAE}
{explanation: "1) AUSTRALIAN DESIGN RULE 18/00 (1988) : In the case      ",
"groupII Visual Indicators', (temperature is considered to be in groupII)",
"the steering wheel rim including its supporting arms and attachnements  ",
"there should not constitute an obstruction.                    ",
"2) SAE SP-576-84, SP-654-86 and SP-734-88, SAE J941-90, SAE J1052-90,
"  BS AU 199-84, ISO 4040-83 Automotive Electronic Displays and        ",
"  Information Systems. Design principles for steering wheel visibility  ",
"and locations. The display panel area shall be visible through the      ",
"steering wheel without eye or head movement. Visibility through the      ",
"steering wheel can be divided into three zones, door-open in zone two:  ",
"  i) Zone two is located on the centre of the displays panel and        ",
"     typically contains door-open indicators, turn-signals, and headlamp",
"     indicator - Upper/Lower beam.                              "}.
```

****

the adjustment for door_open position is to :
lower the top point by : 48 mm
move the left point to the right by : -5 mm
move the right point to the left by : 30 mm
****

Type 'c' to continue, or 's' to stop.

```
Ready for command: c

what would you like to check?
1. Check ergonomics visibility
2. Check allowable eye movement
3. Check allowable head movement
4. end this menu
=? 2

The members of the class 'view_with' have already been determined.
*******************************************************************
for angles of eye movement given:
horizontal left =          65
horizontal right =         15
vertical upwards =         45
vertical downwards =       20
these angles of eye movement are: outside gII eye limit <1.00>
*******************************************************************
 would you like to see the reason for this?
1. yes
2. no
=? 1
*******************************************************************
The value of view_with:eye move>gII eye limit = outside gII eye limit <1.00>.
This is due to the following knowledge sources:
  rule: groupII eye move outside limit

Would you like to see the supporting knowledge sources and demons? (y/n) y

Name: groupII eye move outside limit
Kind of entity: Production Rule
groupII eye move outside limit:
v_w: view_with
     if
         view_with:eye move > horizontal left gt 40 or
         view_with:eye move > horizontal right lt -25 or
         view_with:eye move > vertical upwards gt 10 or
         view_with:eye move > vertical downwards lt -35
    then
         v_w>gII eye limit = outside gII eye limit.
    endif

{references: ADR}
{explanation: "AUSTRALIAN DESIGN RULE 18/00 (1988)              ",
"All 'Visual Indicators' specified as GroupII in Clause 18.2 shall",
"be totally located between 2 vertical planes inclined at 40 deg. ",
"left and 25 deg. right of the longitudinal axis of the vehicle   ",
```

"and passing through the for most points of the left and right ",
"95th Percentile Eye Ellipses respectively. Such indicators shall ",
"be totally located above a plane inclined downwards at 35 deg. ",
"from the horizontal and including a horizontal transverse line ",
"through the for most points of each of the '95th Percentile Eye ",
"Ellipses' and below a plane tangential to the bottom of the 95th ",
"Percentile Eye Ellipses which includes a line at ground level ",
"transverse to the longitudinal axis of the vehicle 11m forward ",
"of the rear most eye ellips point. "}.
****
You can adjust the position of the door_open so that it
lies within the allowable ergonomics eye movement by :
move to right to reduce angle of eye movement by : 25 deg
move down to reduce angle of eye movement by : 35 deg
****
Type 'c' to continue, or 's' to stop.
Ready for command: c

what would you like to check?
1. Check ergonomics visibility
2. Check allowable eye movement
3. Check allowable head movement
4. end this menu
=? 3

The members of the class 'view_with' have already been determined.
****
for angles of head movement given:
horizontal left = 45
horizontal right = 25
vertical upwards = 30
vertical downwards = 25
these angles of head movement are: outside gII head limit <1.00>
****
would you like to see the reason for this?
1. yes
2. no
=? 1

****
The value of view_with:head move>gII head limit = outside gII head limit <1.00>.
This is due to the following knowledge sources:
rule: groupII head move outside limit

Would you like to see the supporting knowledge sources and demons? (y/n) y

Name: groupII head move outside limit

```
Kind of entity: Production Rule
groupII head move outside limit:
v_w: view_with
    if
        view_with:head move > horizontal left gt 15 or
        view_with:head move > horizontal right lt -15 or
        view_with:head move > vertical upwards gt 15 or
        view_with:head move > vertical downwards lt -15
   then
        v_w>gII head limit = outside gII head limit.
   endif      {references: ISO}
   {explanation: "The identification and those parts of ",
        "    the display area required to indicate ",
        "    a critical condition shall be visible ",
        "    without head movement.          ",
        "    The remaining parts of the display   ",
        "    shall also be 'visible'; for these,  ",
        "    head movement is permited.         "}.
*********************************************************************
You can adjust the position of the door_open  so that it
lies within the allowable ergonomics head movement by :
move to right to reduce angle of head movement by :    30 deg
move down to reduce angle of head movement by :        15 deg
*********************************************************************
 Type 'c' to continue, or 's' to stop.
Ready for command: c

what would you like to check?
1. Check ergonomics visibility
2. Check allowable eye movement
3. Check allowable head movement
4. end this menu
=? 4

What would you like to do?
1. See a list_of displays item developed in_the knowledge base
2. See a list_of some usefull information for design
3. Design a particular interior displays item
4. Check the design for styling
5. Check the type for instruments
6. Check the colour for instruments
7. Quit
=? 3

Which interior displays would you like to design?
It would be a good idea to go through each stage systematically.
1. Dashboard
```

2. Primary Controls
3. Seats
4. Mirror
5. New case
6. Leave this menu
=? 2

Which primary controls would you like to design?
It would be a good idea to go through each stage systematically.
1. Pedals
2. Hand Lever
3. Steering Wheel
4. New case
5. Leave this menu
=? 1

Which pedals would you like to design?
It would be a good idea to go through each stage systematically.
1. Accelerator
2. Brake
3. Clutch
4. New case
5. Leave this menu
=? 1

what would you like to check?
1. Check reachable through the seat adjustment
2. Check leg move within the ergonomics range
3. Check seat move required for reachable
4. Finished with this menu
=? 1

The members of the class 'accele' have already been determined.
The members of the class 'seat_adjust' have already been determined.
The members of the class 'reach' have already been determined.
****
for values of vertices given :
top vertex tp = 65
bottom vertex bp = 150
right vertex rp = 140
left vertex lp = 100
the accelerator is not reach through the seat adjustment <1.00>
****
would you like to see the reason for this?
1. yes
2. no
=? 1

****
The value of 'reachable' has not been determined.
****
the adjustment for accelerator position is to :
****

Type 'c' to continue, or 's' to stop.
Ready for command: c

what would you like to check?
1. Check reachable through the seat adjustment
2. Check leg move within the ergonomics range
3. Check seat move required for reachable
4. Finished with this menu
=? 2

The members of the class 'dist_with' have already been determined.
****
for angles of leg movement given:
horizontal left = 21
horizontal right = 25
vertical upwards = 9
vertical downwards = 7
these angles of leg movement are: outside gI leg limit <1.00>
****
would you like to see the reason for this?
1. yes
2. no
=? 1
****
The value of dist_with:leg move>gI leg limit = outside gI leg limit <1.00>.
This is due to the following knowledge sources:
rule: groupI leg move outside limit

Would you like to see the supporting knowledge sources and demons? (y/n) y

Name: groupI leg move outside limit
Kind of entity: Production Rule
groupI leg move outside limit:
d_w: dist_with
    if
        dist_with:leg move > horizontal left gt 20 or
        dist_with:leg move > horizontal right lt -20 or
        dist_with:leg move > vertical upwards gt 10 or
        dist_with:leg move > vertical downwards lt -35
    then

```
        d_w>gl leg limit = outside gl leg limit.
    endif

{references: ISO}
{explanation: "ROAD VEHICLES PASSENGER CARS DRIVER HAND       ",
"CONTROL REACH, ISO 3958-77. Leg_reach H_point of the pedals. ",
"The following location for leg_reach H-point of_pedal:     ",
"    -     ACCELERATOR PEDAL    -                ",
"The seating refrence point is terms of the R-point.        ",
"The pivot centre of the torso line and thigh centreline    ",
"of the two or three-dimensional H-point machine template with ",
"95% leg length used to describe vehicle seating geometry.   ",
"The accelerator heel-point or H-point device with floor     ",
"covering; the foot angle of the device is restricted to not  ",
"less than 87 deg. Horizontal dimensional from the R-point   ",
"to the driver H-point (Hx). The vertical dimensions from    ",
"R-point to driver H-point (Hz).                    ",
"    -     BRAKE PEDAL    -                 ",
"The brake heel-point or H-point device height 3-10 inches   ",
"max., 2 inches max. travel with 15 lbs.max. force brake.    ",
"The distance between other pedal 2 inches min. gap.         ",
"    -     CLUCTH PEDAL      -                 ",
"The clucth heel-point or H-point device height 3-10 inches  ",
"max., 4 inches max. travel with 80-90 lbs.max. force brake. ",
"The distance between other pedal 2 inches min. gap.       "}.
******************************************************************

******************************************************************
You can adjust the position of the accelerator so that it
lies within the allowable ergonomics leg movement by :
move to right to reduce angle of leg movement by :      1 deg
******************************************************************
 Type 'c' to continue, or 's' to stop.
Ready for command: c

what would you like to check?
1. Check reachable through the seat adjustment
2. Check leg move within the ergonomics range
3. Check seat move required for reachable
4. Finished with this menu
=? 3

The members of the class 'dist_with' have already been determined.
******************************************************************
for angles of seat adjsutment given:
 horizontal left =          14
horizontal right =          11
```

```
vertical upwards =          13
vertical downwards =        16
these angles of seat movement are: outside gI seat limit <1.00>
*********************************************************************
would you like to see the reason for this?
1. yes
2. no
=? 1
*********************************************************************
The value of dist_with:seat move>gI seat limit = outside gI seat limit <1.00>.
This is due to the following knowledge sources:
  rule: groupI seat move outside limit

Would you like to see the supporting knowledge sources and demons? (y/n) y

Name: groupI seat move outside limit
Kind of entity: Production Rule
groupI seat move outside limit:
d_w: dist_with
     if
          dist_with:seat move > horizontal left gt 10 or
          dist_with:seat move > horizontal right gt 10 or
          dist_with:seat move > vertical upwards gt 10 or
          dist_with:seat move > vertical downwards gt 10
     then
          d_w>gI seat limit = outside gI seat limit.
     endif

{references: ISO}
{explanation: "The identification and those parts of    ",
     "    the pedals area required to indicate     ",
     "    a critical condition shall be leg_reach  ",
     "    without seat movement.                   ",
     "    The remaining parts of the pedals        ",
     "    shall also be 'reach'; for these,        ",
     "    seat movement is permited.               "}.
*********************************************************************
You can adjust the position of the  accelerator so that it
lies within the allowable ergonomics seat adjustment by :
move to right to reduce angle of seat adjustment by :     -21 deg
move down to reduce angle of seat adjustment by :          3 deg
move up to reduce angle of seat adjustment by :           -26 deg
*********************************************************************
Type 'c' to continue, or 's' to stop.
Ready for command: c

what would you like to check?
```

1. Check reachable through the seat adjustment
2. Check leg move within the ergonomics range
3. Check seat move required for reachable
4. Finished with this menu
=? 4

Which pedals would you like to design?
It would be a good idea to go through each stage systematically.
1. Accelerator
2. Brake
3. Clutch
4. New case
5. Leave this menu
=? 2

what would you like to check?
1. Check reachable through the seat adjustment
2. Check leg move within the ergonomics range
3. Check seat move required for reachable
4. Finished with this menu
=? 1

The members of the class 'brake' have already been determined.
The members of the class 'seat_adjust' have already been determined.
The members of the class 'reach' have already been determined.
****
for values of vertices given :
top vertex tp = 112
bottom vertex bp = 115
right vertex rp = 120
left vertex lp = 114
the brake is not reach through the seat adjustment <1.00>
****

would you like to see the reason for this?
1. yes
2. no
=? 1

****
The value of 'reachable' has not been determined.
****
the adjustment for brake position is to :
lower the top point by : –5.5 mm
move the left point to the right by : 39.2 mm
****
Type 'c' to continue, or 's' to stop.

```
Ready for command: c

what would you like to check?
1. Check reachable through the seat adjustment
2. Check leg move within the ergonomics range
3. Check seat move required for reachable
4. Finished with this menu
=? 2

The members of the class 'dist_with' have already been determined.
*******************************************************************
for angles of leg movement given:
horizontal left =          25
horizontal right =         10
vertical upwards =         35
vertical downwards =       20
these angles of leg movement are: outside gI leg limit <1.00>
*******************************************************************

would you like to see the reason for this?
1. yes
2. no
=? 1
*******************************************************************
The value of dist_with:leg move>gI leg limit = outside gI leg limit <1.00>.
This is due to the following knowledge sources:
  rule: groupI leg move outside limit

Would you like to see the supporting knowledge sources and demons? (y/n) y

Name: groupI leg move outside limit
Kind of entity: Production Rule
 groupI leg move outside limit:
d_w: dist_with
     if
         dist_with:leg move > horizontal left gt 20 or
         dist_with:leg move > horizontal right lt -20 or
         dist_with:leg move > vertical upwards gt 10 or
         dist_with:leg move > vertical downwards lt -35
    then
         d_w>gI leg limit = outside gI leg limit.
    endif

{references: ISO}
{explanation: "ROAD VEHICLES PASSENGER CARS DRIVER HAND      ",
"CONTROL REACH, ISO 3958-77. Leg_reach H_point of the pedals. ",
"The following location for leg_reach H-point of_pedal:     ",
"    -     ACCELERATOR PEDAL     -            ",
```

"The seating refrence point is terms of the R-point. ",
"The pivot centre of the torso line and thigh centreline ",
"of the two or three-dimensional H-point machine template with ",
"95% leg length used to describe vehicle seating geometry. ",
"The accelerator heel-point or H-point device with floor ",
"covering; the foot angle of the device is restricted to not ",
"less than 87 deg. Horizontal dimensional from the R-point ",
"to the driver H-point (Hx). The vertical dimensions from ",
"R-point to driver H-point (Hz). ",
" - BRAKE PEDAL - ",
"The brake heel-point or H-point device height 3-10 inches ",
"max., 2 inches max. travel with 15 lbs.max. force brake. ",
"The distance between other pedal 2 inches min. gap. ",
" - CLUCTH PEDAL - ",
"The clucth heel-point or H-point device height 3-10 inches ",
"max., 4 inches max. travel with 80-90 lbs.max. force brake. ",
"The distance between other pedal 2 inches min. gap. "}.
****
You can adjust the position of the brake so that it
lies within the allowable ergonomics leg movement by :
move to right to reduce angle of leg movement by : 5 deg
move down to reduce angle of leg movement by : 25 deg
****

Type 'c' to continue, or 's' to stop.
Ready for command: c

what would you like to check?
1. Check reachable through the seat adjustment
2. Check leg move within the ergonomics range
3. Check seat move required for reachable
4. Finished with this menu
=? 3

The members of the class 'dist_with' have already been determined.
****
for angles of seat adjustment given:
horizontal left = 15
horizontal right = 40
vertical upwards = 15
vertical downwards = 45
these angles of seat movement are: outside gI seat limit <1.00>
****

would you like to see the reason for this?
1. yes
2. no
=? 1

****

The value of dist_with:seat move>gI seat limit = outside gI seat limit <1.00>.
This is due to the following knowledge sources:
  rule: groupI seat move outside limit

Would you like to see the supporting knowledge sources and demons? (y/n) y

Name: groupI seat move outside limit
Kind of entity: Production Rule
groupI seat move outside limit:
d_w: dist_with
      if
            dist_with:seat move > horizontal left gt 10 or
            dist_with:seat move > horizontal right gt 10 or
            dist_with:seat move > vertical upwards gt 10 or
            dist_with:seat move > vertical downwards gt 10
      then
            d_w>gI seat limit = outside gI seat limit.
      endif

{references: ISO}
{explanation: "The identification and those parts of ",
      " the pedals area required to indicate ",
      " a critical condition shall be leg_reach ",
      " without seat movement. ",
      " The remaining parts of the pedals ",
      " shall also be 'reach'; for these, ",
      " seat movement is permited. "}.

****

You can adjust the position of the brake so that it
lies within the allowable ergonomics seat adjustment by :
move to right to reduce angle of seat adjustment by : -50 deg
move down to reduce angle of seat adjustment by : 5 deg
move up to reduce angle of seat adjustment by : -55 deg

****

Type 'c' to continue, or 's' to stop.
Ready for command: c

what would you like to check?
1. Check reachable through the seat adjustment
2. Check leg move within the ergonomics range
3. Check seat move required for reachable
4. Finished with this menu
=? 4

Which pedals would you like to design?
It would be a good idea to go through each stage systematically.

1. Accelerator
2. Brake
3. Clutch
4. New case
5. Leave this menu
=? 3

what would you like to check?
1. Check ergonomics reachable
2. Check allowable leg move
3. Check allowable seat move
4. Finished with this menu
=? 1

The members of the class 'clutch' have already been determined.
The members of the class 'seat_adjust' have already been determined.
The members of the class 'reach' have already been determined.
****
for values of vertices given :
top vertex tp = 115
bottom vertex bp = 110
right vertex rp = 120
left vertex lp = 110
the clutch is not reach through the seat adjustment <1.00>
****
would you like to see the reason for this?
1. yes
2. no
=? 1
****
The value of 'reachable' has not been determined.
****
the adjustment for clutch position is to :
lower the top point by : -95 mm
move the left point to the right by : 30 mm
****
Type 'c' to continue, or 's' to stop.
Ready for command: c

what would you like to check?
1. Check ergonomics reachable
2. Check allowable leg move
3. Check allowable seat move
4. Finished with this menu
=? 2

The members of the class 'dist_with' have already been determined.

```
************************************************************************
for angles of leg movement given:
horizontal left =          95
horizontal right =         75
vertical upwards =         85
vertical downwards =       54
these angles of leg movement are: outside gII leg limit <1.00>
************************************************************************

would you like to see the reason for this?
1. yes
2. no
=? 1

************************************************************************
The value of dist_with:leg move>gII leg limit = outside gII leg limit <1.00>.
This is due to the following knowledge sources:
  rule: groupII leg move outside limit

Would you like to see the supporting knowledge sources and demons? (y/n) y

Name: groupII leg move outside limit
Kind of entity: Production Rule
groupII leg move outside limit:
d_w: dist_with
    if
        dist_with:leg move > horizontal left gt 40 or
        dist_with:leg move > horizontal right lt -25 or
        dist_with:leg move > vertical upwards gt 10 or
        dist_with:leg move > vertical downwards lt -35
    then
        d_w>gII leg limit = outside gII leg limit.
    endif

{references: ISO}
{explanation: "ROAD VEHICLES PASSENGER CARS DRIVER HAND      ",
"CONTROL REACH, ISO 3958-77. Leg_reach H_point of the pedals. ",
"The following location for leg_reach H-point of_pedal:     ",
"   -    ACCELERATOR PEDAL    -                  ",
"The seating refrence point is terms of the R-point.      ",
"The pivot centre of the torso line and thigh centreline     ",
"of the two or three-dimensional H-point machine template with",
"95% leg length used to describe vehicle seating geometry.   ",
"The accelerator heel-point or H-point device with floor     ",
"covering; the foot angle of the device is restricted to not ",
"less than 87 deg. Horizontal dimensional from the R-point   ",
"to the driver H-point (Hx). The vertical dimensions from    ",
"R-point to driver H-point (Hz).                  ",
"   -    BRAKE PEDAL    -                  ",
```

```
"The brake heel-point or H-point device height 3-10 inches   ",
"max., 2 inches max. travel with 15 lbs.max. force brake.   ",
"The distance between other pedal 2 inches min. gap.        ",
"    -    CLUCTH PEDAL     -                     ",
"The clucth heel-point or H-point device height 3-10 inches  ",
"max., 4 inches max. travel with 80-90 lbs.max. force brake. ",
"The distance between other pedal 2 inches min. gap.        "}.
*********************************************************************
You can adjust the position of the clutch so that it
lies within the allowable ergonomics leg movement by :
move to right to reduce angle of leg movement by :      55 deg
move down to reduce angle of leg movement by :          75 deg
*********************************************************************

 Type 'c' to continue, or 's' to stop.
Ready for command: c
what would you like to check?
1. Check ergonomics reachable
2. Check allowable leg move
3. Check allowable seat move
4. Finished with this menu
=? 3
The members of the class 'dist_with' have already been determined.
*********************************************************************
for angles of seat adjustment given:
 horizontal left =          50
 horizontal right =         75
vertical upwards =          84
vertical downwards =        43
these angles of seat movement are: outside gII seat limit <1.00>
*********************************************************************
would you like to see the reason for this?
1. yes
2. no
=? 1

*********************************************************************
The value of dist_with:seat move>gII seat limit = outside gII seat limit <1.00>.
This is due to the following knowledge sources:
  rule: groupII seat move outside limit
Would you like to see the supporting knowledge sources and demons? (y/n) y
Name: groupII seat move outside limit
Kind of entity: Production Rule
groupII seat move outside limit:
d_w: dist_with
      if
          dist_with:seat move > horizontal left gt 15 or
```

dist_with:seat move > horizontal right lt -15 or
dist_with:seat move > vertical upwards gt 15 or
dist_with:seat move > vertical downwards lt -15
then
d_w>gII seat limit = outside gII seat limit.
endif
{references: ISO}
{explanation: "The identification and those parts of ",
" the pedals area required to indicate ",
" a critical condition shall be leg_reach ",
" without seat movement. ",
" The remaining parts of the pedals ",
" shall also be 'reach'; for these, ",
" seat movement is permited. "}.

****

You can adjust the position of the clutch so that it
lies within the allowable ergonomics seat adjustment by :
move to left to reduce angle of seat adjustment by : 35 deg
move down to reduce angle of seat adjustment by : 69 deg

****

Type 'c' to continue, or 's' to stop.
Ready for command: c
what would you like to check?
1. Check ergonomics reachable
2. Check allowable leg move
3. Check allowable seat move
4. Finished with this menu
=? 4
Which pedals would you like to design?
It would be a good idea to go through each stage systematically.
1. Accelerator
2. Brake
3. Clutch
4. New case
5. Leave this menu
=? 4
Which pedals would you like to design?
It would be a good idea to go through each stage systematically.
1. Accelerator
2. Brake
3. Clutch
4. New case
5. Leave this menu
=? 5
What would you like to do?
1. See a list_of displays item developed in_the knowledge base

2. See a list_of some usefull information for design
3. Design a particular interior displays item
4. Check the design for styling
5. Check the type for instruments
6. Check the colour for instruments
7. Quit
=? 4

For 'panel' of class 'displays':
area
(Enter a number)
=? 1
****
styling is rated to be good ergonomics <0.10>
****

For 'accele' of class 'pedals':shape
1. rectangle
2. square
(Multiple answers allowed)
=? 1

For 'accele' of class 'pedals':relative position
1. left_of
2. right_of
3. centre_of
4. above
5. below
6. angle_of
(Multiple answers allowed)
=? 1
****
styling is rated to be good ergonomics <0.10>
****

What would you like to do?
1. See a list_of displays item developed in_the knowledge base
2. See a list_of some usefull information for design
3. Design a particular interior displays item
4. Check the design for styling
5. Check the type for instruments
6. Check the colour for instruments
7. Quit
=? 5

For 'panel' of class 'displays': type
1. electromechanical
2. curvilinear
3. discrete
4. alphanumeric

5. representational
=? 1

****

type is rated to be electromechanical <1.00>

****

What would you like to do?
1. See a list_of displays item developed in_the knowledge base
2. See a list_of some usefull information for design
3. Design a particular interior displays item
4. Check the design for styling
5. Check the type for instruments
6. Check the colour for instruments
7. Quit
=? 6

For 'panel' of class 'displays':
colour
1. red
2. yellow
3. green
4. blue
5. white
=? 1

****

colour is rated to be Red is danger, damage to equipment immediate or imminent,hot in climate control system or temperature indicators.red <1.00>

****

****

colour is rated to be Yellow is caution, vehicle system malfunction, danger in,vehicle likely,or other condition which may produce hazardin the longer term.red <1.00>

****

****

colour is rated to be Green is safe, normal operation of the vehicle system,when blue or yellow are not required.red <1.00>

****

****

colour is rated to be Blue is driving_upper_high beam tell-tales only,and cold,in climate control systems or temperature indicators.red <1.00>

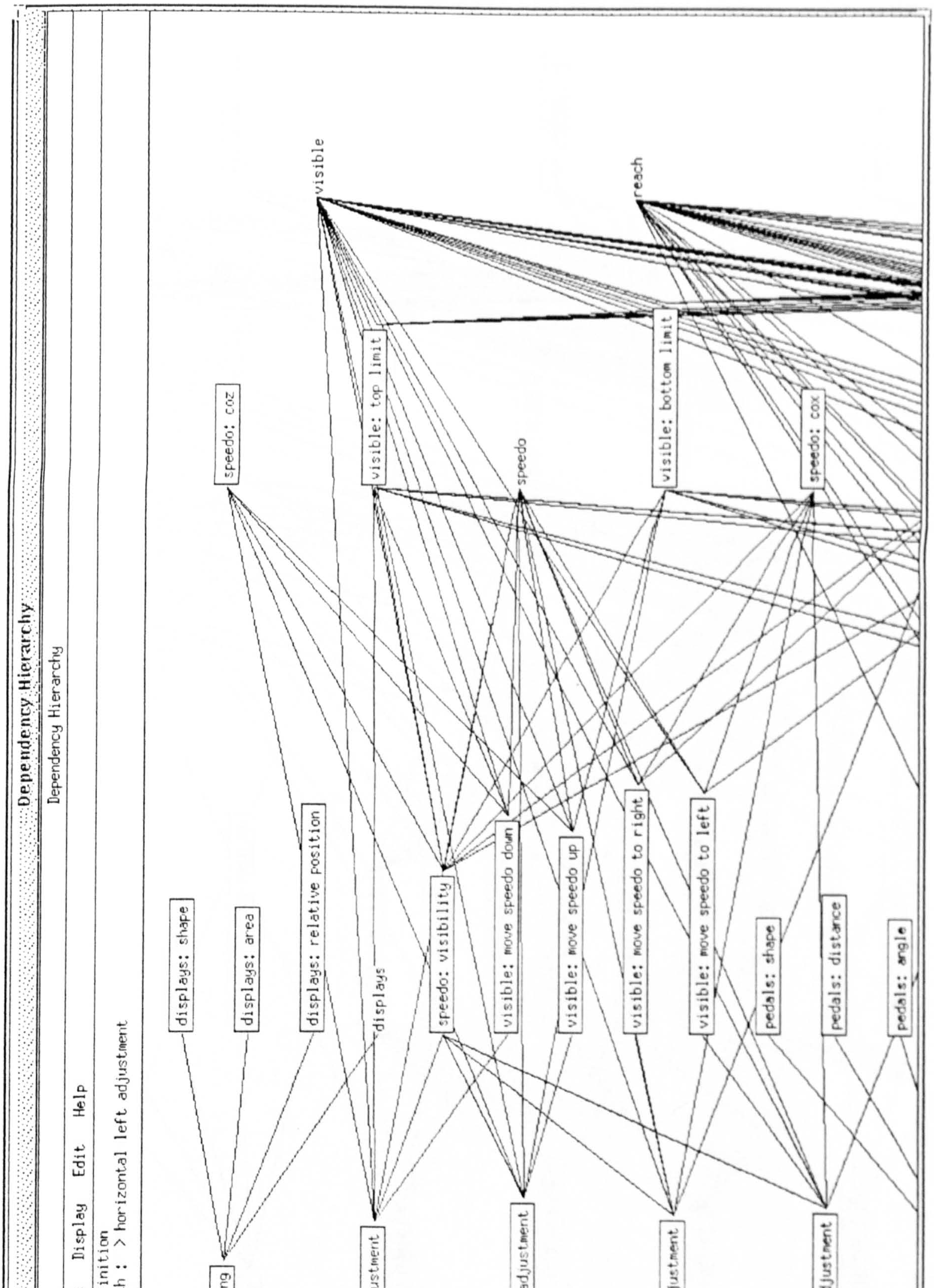
****

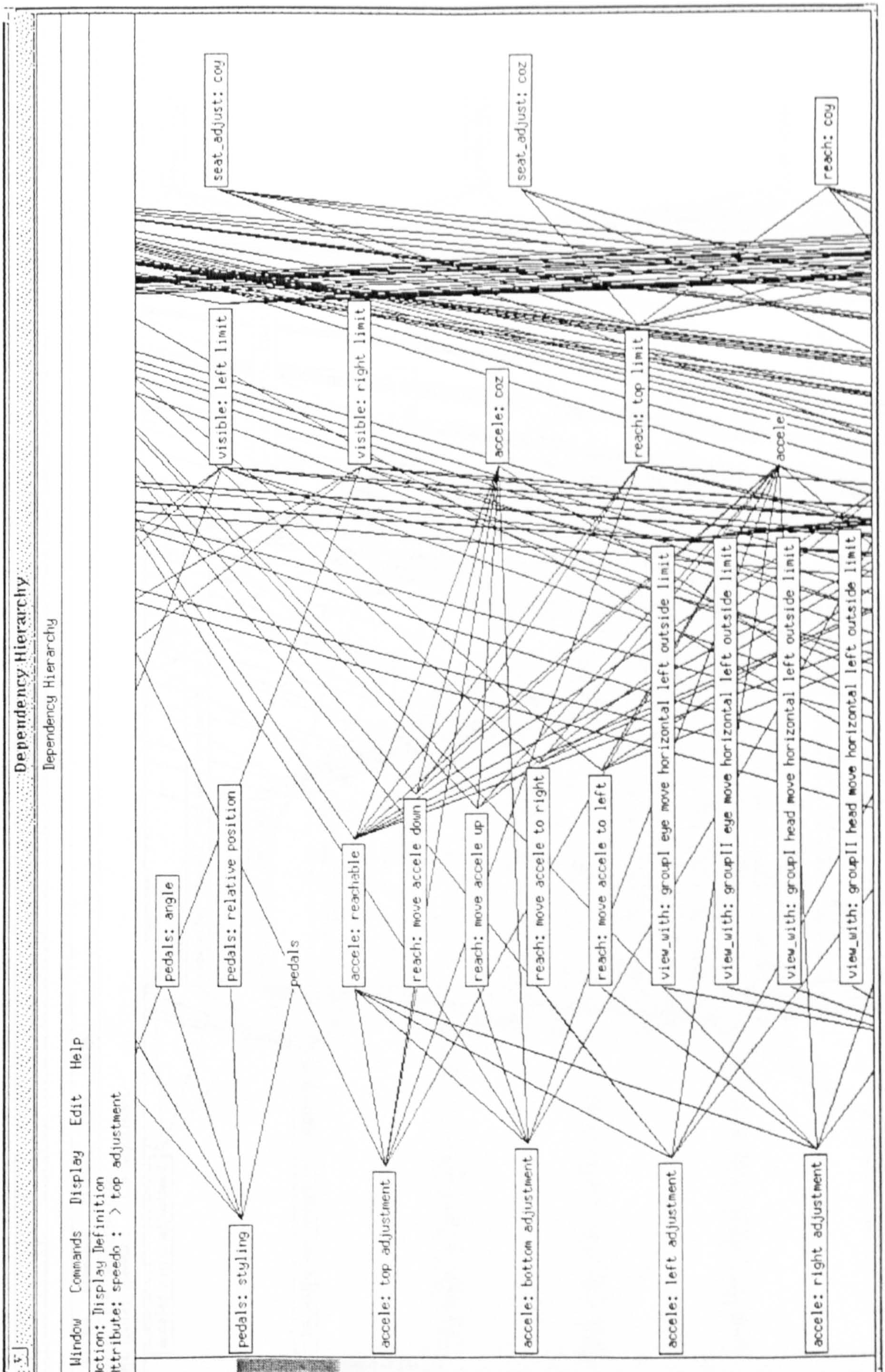
****

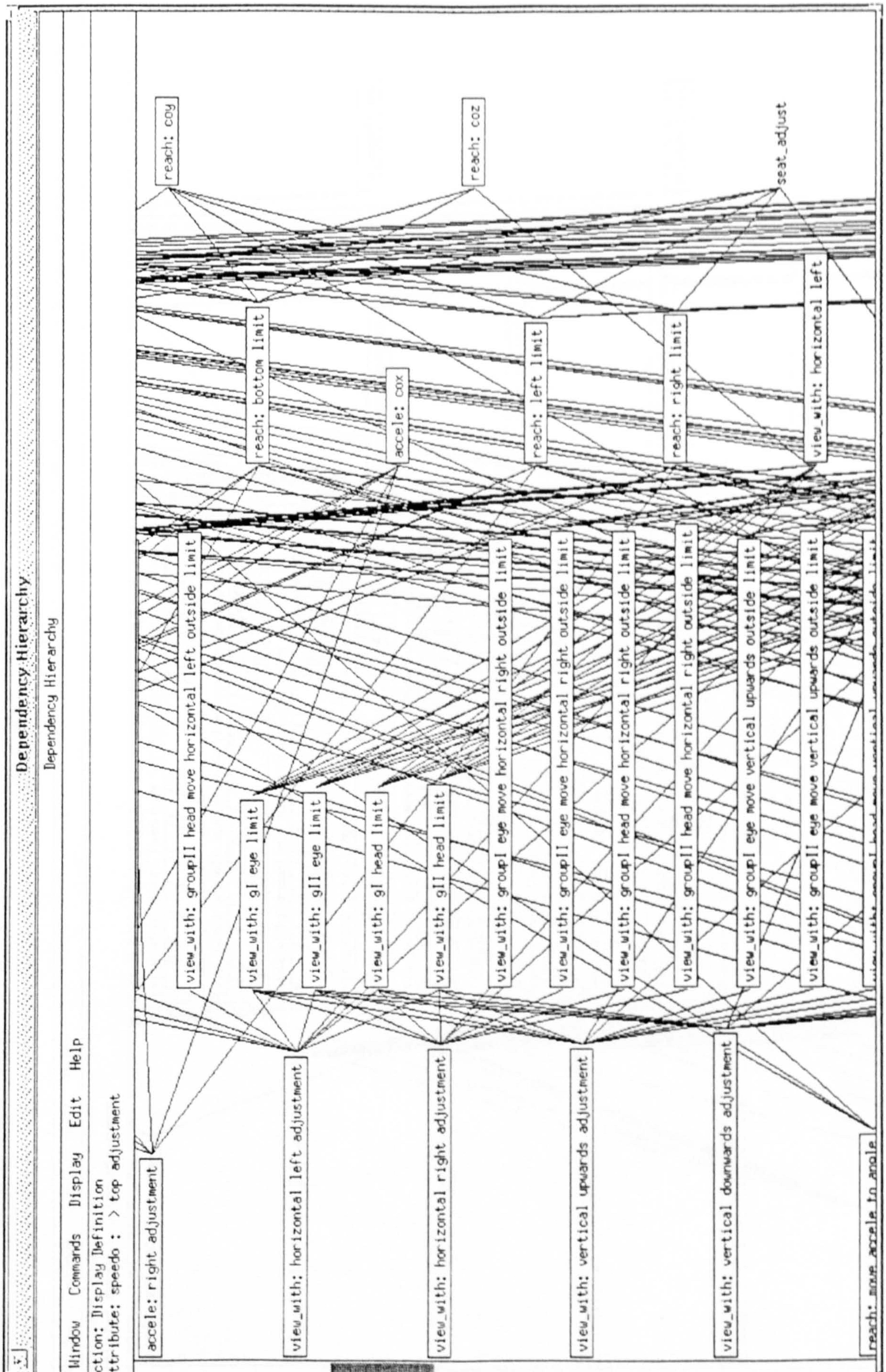
colour is rated to be White is other conditions where none of the above colours,are appropriate.red <1.00>

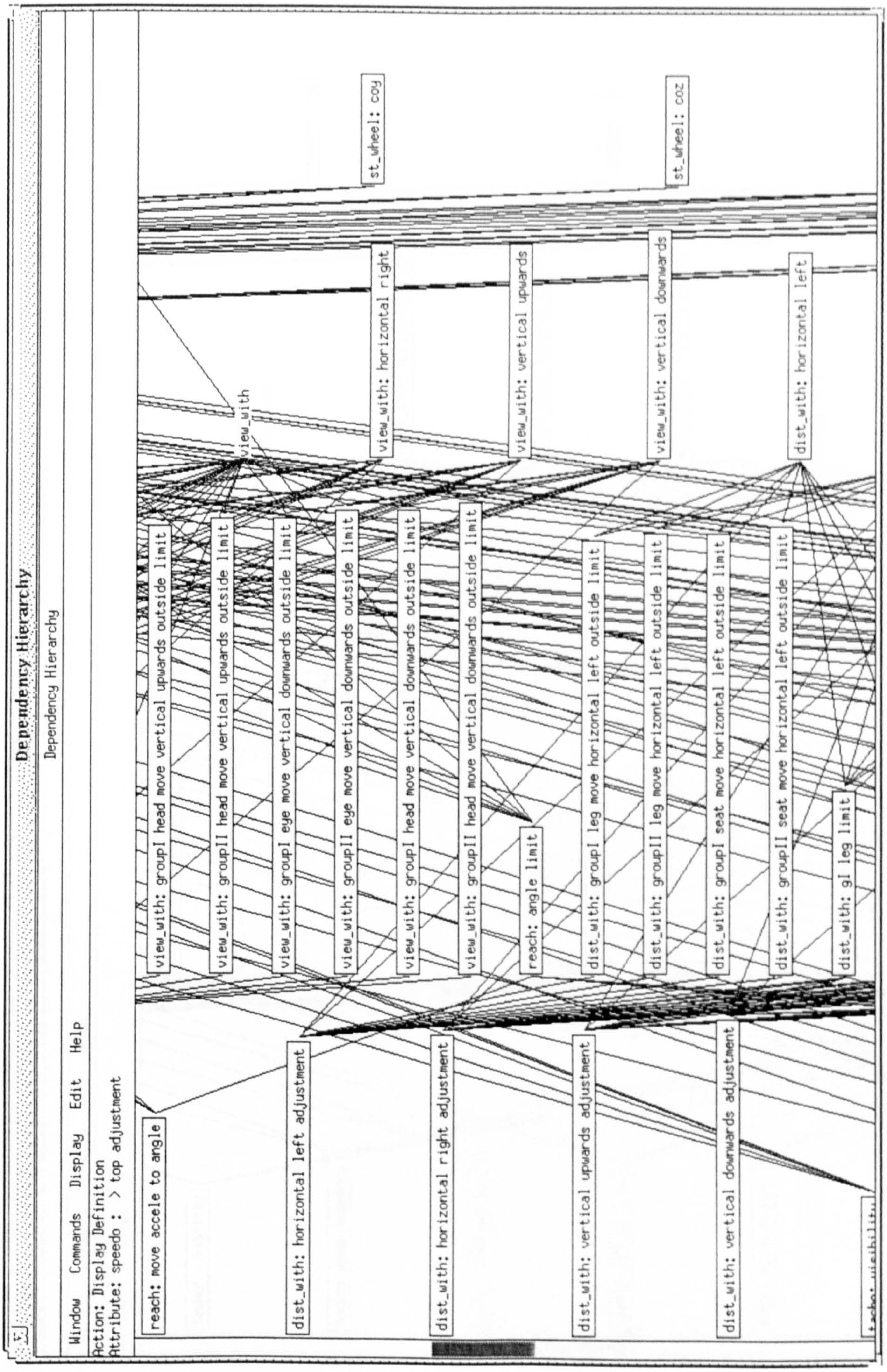
****

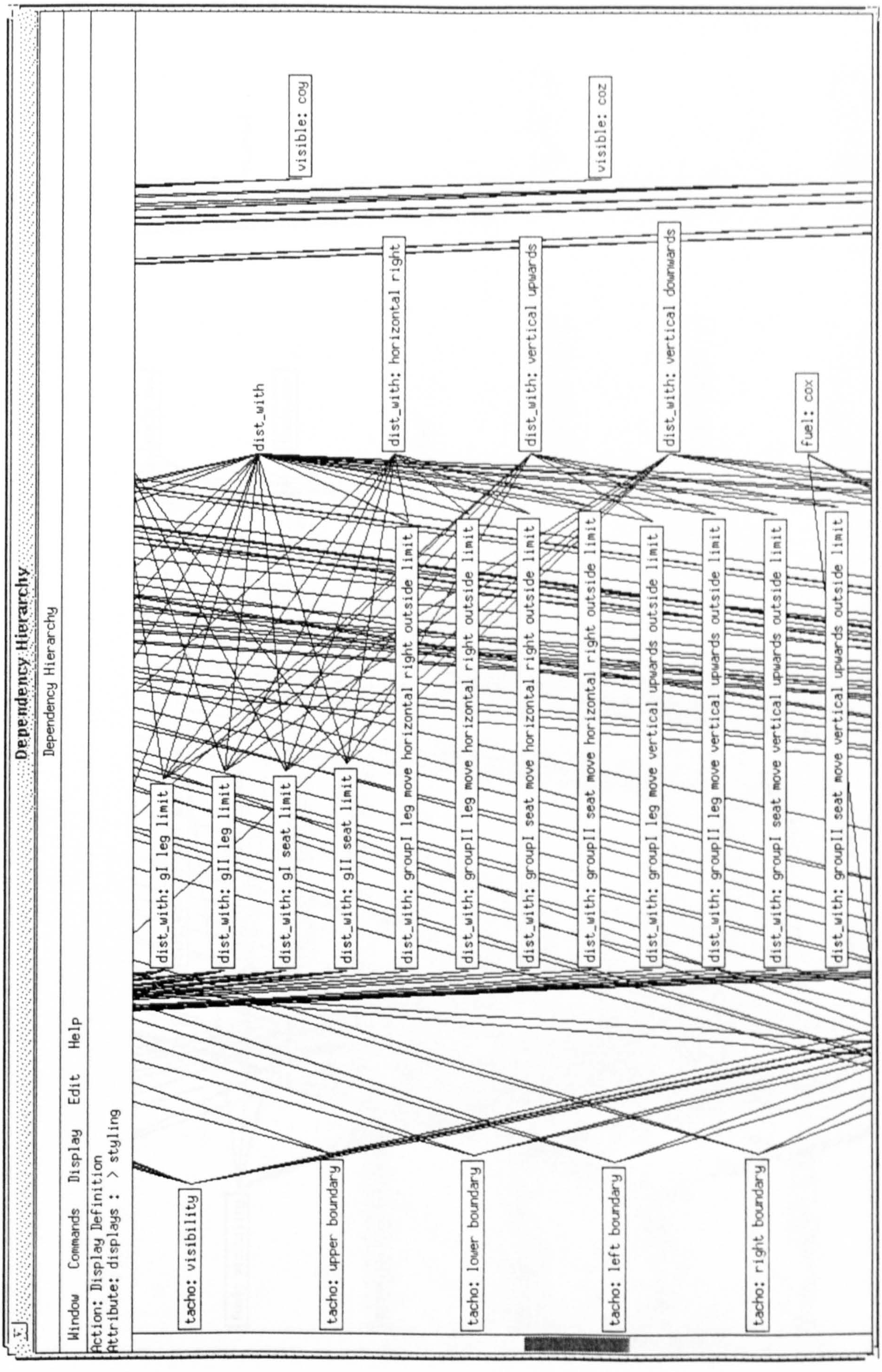
APPENDIX 4

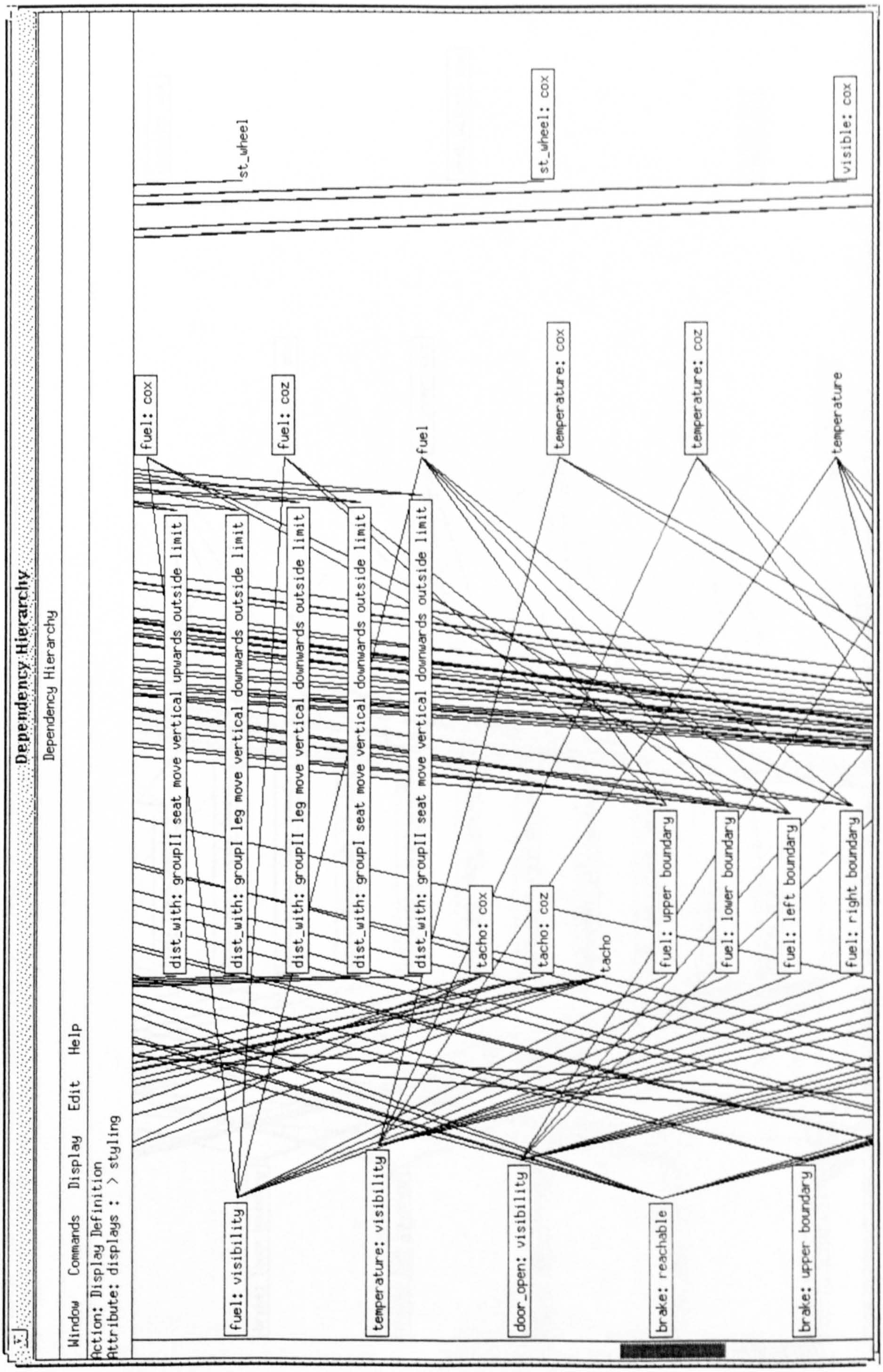

Dependency Hierarchy
Dependency Hierarchy
Window Commands Display Edit Help
Action: Display Definition
Attribute: view_with : > horizontal left adjustment
visible
reach
speedo: coz
visible: top limit
speedo
visible: bottom limit
speedo: cox
displays: shape
displays: area
displays: relative position
displays
speedo: visibility
visible: move speedo down
visible: move speedo up
visible: move speedo to right
visible: move speedo to left
pedals: shape
pedals: distance
pedals: angle
displays: styling
speedo: top adjustment
speedo: bottom adjustment
speedo: left adjustment
speedo: right adjustment

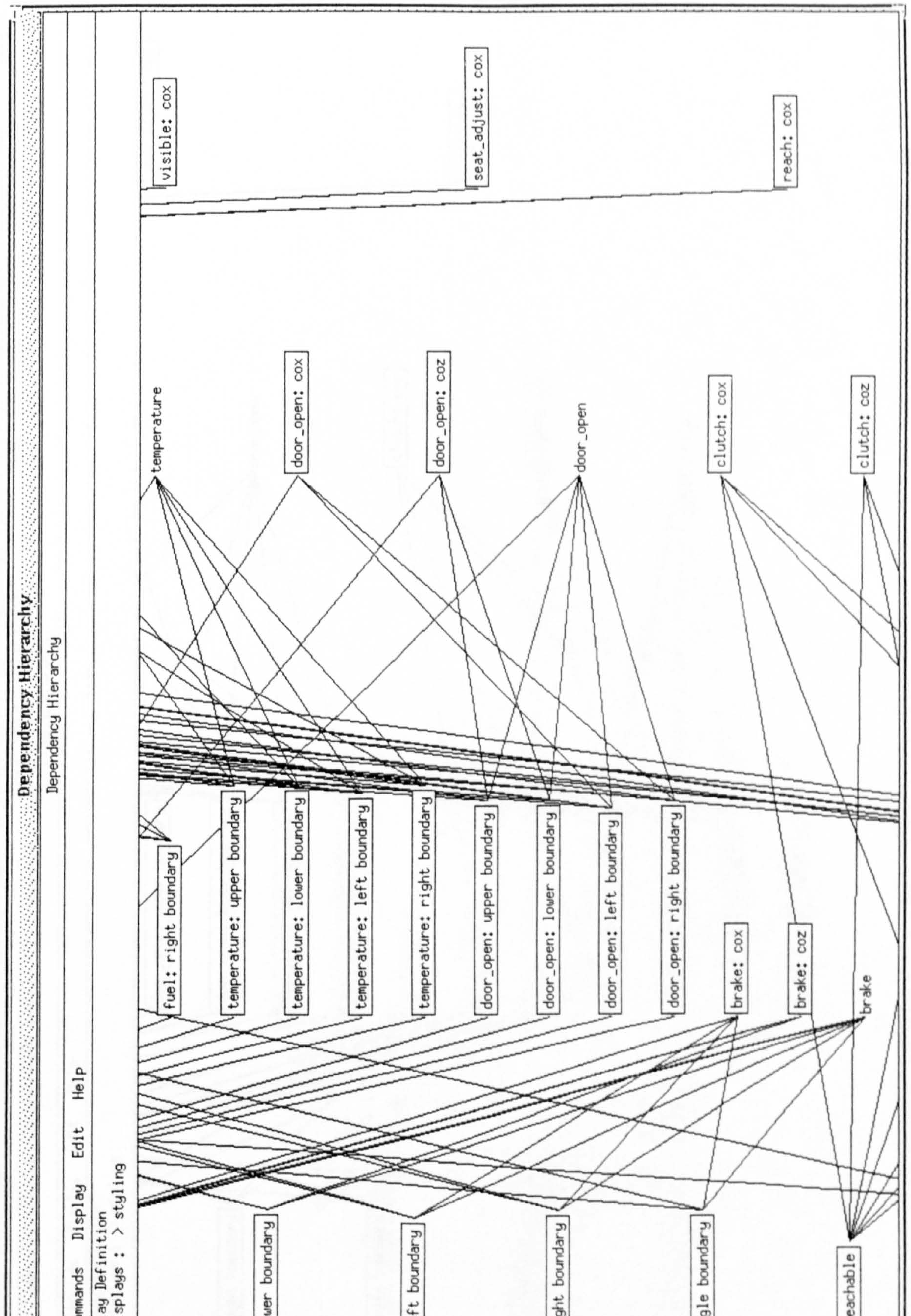

Dependency Hierarchy
Dependency Hierarchy
Window
Commands
Display
Edit
Help
Action: Display Definition
Attribute: speedo : > top adjustment
seat_adjust: coy
seat_adjust: coz
reach: coy
visible: left limit
visible: right limit
accele: coz
reach: top limit
accele
view_with: groupI eye move horizontal left outside limit
view_with: groupII eye move horizontal left outside limit
view_with: groupI head move horizontal left outside limit
view_with: groupII head move horizontal left outside limit
pedals: angle
pedals: relative position
pedals
accele: reachable
reach: move accele down
reach: move accele up
reach: move accele to right
reach: move accele to left
pedals: styling
accele: top adjustment
accele: bottom adjustment
accele: left adjustment
accele: right adjustment

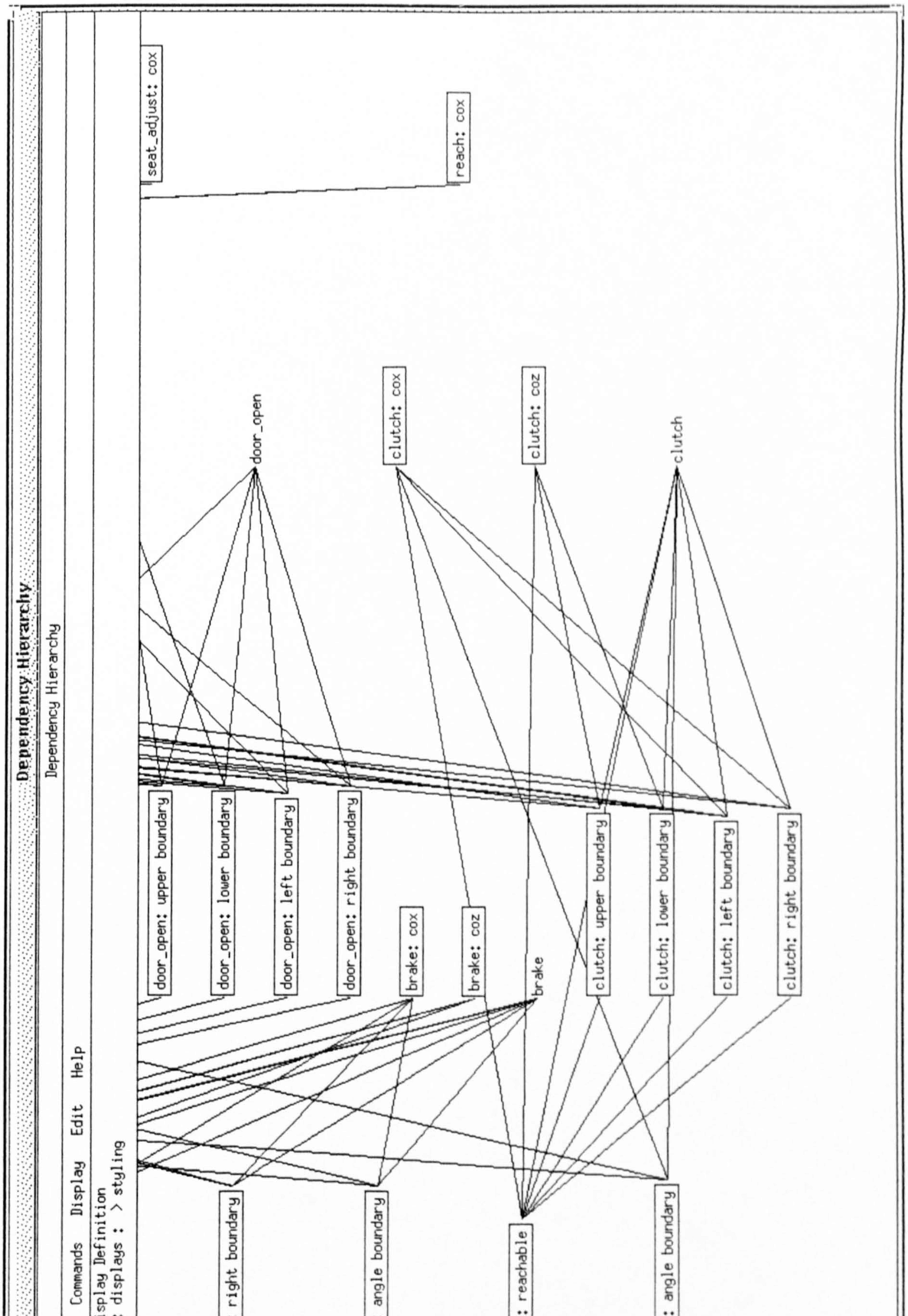

Dependency Hierarchy
Dependency Hierarchy
Window Commands Display Edit Help
Action: Display Definition
Attribute: speedo : > top adjustment
accele: right adjustment
view_with: horizontal left adjustment
view_with: horizontal right adjustment
view_with: vertical upwards adjustment
view_with: vertical downwards adjustment
reach: move accele to angle
view_with: groupII head move horizontal left outside limit
view_with: gI eye limit
view_with: gII eye limit
view_with: gI head limit
view_with: gII head limit
view_with: groupI eye move horizontal right outside limit
view_with: groupII eye move horizontal right outside limit
view_with: groupI head move horizontal right outside limit
view_with: groupII head move horizontal right outside limit
view_with: groupI eye move vertical upwards outside limit
view_with: groupII eye move vertical upwards outside limit
reach: bottom limit
accele: cox
reach: left limit
reach: right limit
view_with: horizontal left
reach: coy
reach: coz
seat_adjust


Dependency Hierarchy
Dependency Hierarchy
Window Commands Display Edit Help
Action: Display Definition
Attribute: speedo : > top adjustment
reach: move accele to angle
dist_with: horizontal left adjustment
dist_with: horizontal right adjustment
dist_with: vertical upwards adjustment
dist_with: vertical downwards adjustment
view_with: groupI head move vertical upwards outside limit
view_with: groupII head move vertical upwards outside limit
view_with: groupI eye move vertical downwards outside limit
view_with: groupII eye move vertical downwards outside limit
view_with: groupI head move vertical downwards outside limit
view_with: groupII head move vertical downwards outside limit
reach: angle limit
dist_with: groupI leg move horizontal left outside limit
dist_with: groupII leg move horizontal left outside limit
dist_with: groupI seat move horizontal left outside limit
dist_with: groupII seat move horizontal left outside limit
dist_with: gI leg limit
view_with
view_with: horizontal right
view_with: vertical upwards
view_with: vertical downwards
dist_with: horizontal left
st_wheel: coy
st_wheel: coz


Dependency Hierarchy
Dependency Hierarchy
Window Commands Display Edit Help
Action: Display Definition
Attribute: displays : > styling
visible: coy
visible: coz
dist_with: horizontal right
dist_with: vertical upwards
dist_with: vertical downwards
dist_with
fuel: cox
dist_with: gI leg limit
dist_with: gII leg limit
dist_with: gI seat limit
dist_with: gII seat limit
dist_with: groupI leg move horizontal right outside limit
dist_with: groupII leg move horizontal right outside limit
dist_with: groupI seat move horizontal right outside limit
dist_with: groupII seat move horizontal right outside limit
dist_with: groupI leg move vertical upwards outside limit
dist_with: groupII leg move vertical upwards outside limit
dist_with: groupI seat move vertical upwards outside limit
dist_with: groupII seat move vertical upwards outside limit
tacho: visibility
tacho: upper boundary
tacho: lower boundary
tacho: left boundary
tacho: right boundary


Dependency Hierarchy
Dependency Hierarchy
Window Commands Display Edit Help
Action: Display Definition
Attribute: displays : > styling
st_wheel
st_wheel: cox
visible: cox
fuel: cox
fuel: coz
fuel
temperature: cox
temperature: coz
temperature
dist_with: groupII seat move vertical upwards outside limit
dist_with: groupI leg move vertical downwards outside limit
dist_with: groupII leg move vertical downwards outside limit
dist_with: groupI seat move vertical downwards outside limit
dist_with: groupII seat move vertical downwards outside limit
tacho: cox
tacho: coz
tacho
fuel: upper boundary
fuel: lower boundary
fuel: left boundary
fuel: right boundary
fuel: visibility
temperature: visibility
door_open: visibility
brake: reachable
brake: upper boundary


Dependency Hierarchy
Dependency Hierarchy
Window Commands Display Edit Help
Action: Display Definition
Attribute: displays : > styling
visible: cox
seat_adjust: cox
reach: cox
temperature
door_open: cox
door_open: coz
door_open
clutch: cox
clutch: coz
fuel: right boundary
temperature: upper boundary
temperature: lower boundary
temperature: left boundary
temperature: right boundary
door_open: upper boundary
door_open: lower boundary
door_open: left boundary
door_open: right boundary
brake: cox
brake: coz
brake
brake: lower boundary
brake: left boundary
brake: right boundary
brake: angle boundary
clutch: reachable


Dependency Hierarchy
Dependency Hierarchy
Window Commands Display Edit Help
Action: Display Definition
Attribute: displays : > styling
seat_adjust: cox
reach: cox
door_open
clutch: cox
clutch: coz
clutch
door_open: upper boundary
door_open: lower boundary
door_open: left boundary
door_open: right boundary
brake: cox
brake: coz
brake
clutch: upper boundary
clutch: lower boundary
clutch: left boundary
clutch: right boundary
brake: right boundary
brake: angle boundary
clutch: reachable
clutch: angle boundary