The Colour Concept Generator:
A Computer Tool to Propose Colour Concepts for Products.

William Robert Eves BEd(Hons), MSc.

A thesis submitted in partial fulfilment of the requirements of

Bournemouth University

for the degree of

Doctor of Philosophy.

July 1997

Abstract

This thesis documents research undertaken into the design and evaluation of a computer tool (Colour Concept Generator) to produce colour schemes for products from verbal descriptors depicting a required aesthetic image or style. The system was designed to translate between descriptive words and colour combinations and aims to provide a form of ideas stimulus for a product designer at the initial stages of the design process.

The computer system uses elements of artificial intelligence (AI) to 'learn' colour and descriptor semiotic relations from a product designer based upon a proposed objective criteria or to reflect a designers personal style. Colour concepts for products can then be generated from descriptors based upon these semiotic relations.

The philosophy of the research is based upon the idea of computing colour aesthetics at the front end of the design process and the design of an AI software mechanism to facilitate this. The problem was analysed with respect to the available literature on colour and a set of detail requirements for the system were presented. The system was then designed and code based upon the requirements and evaluated in terms of the overall philosophy, system methodology and application of computer media.

The research is a contribution to the field of computer aided design regarding colour aesthetics and demonstrates the possibility of using an artificial intelligent machine to inspire and stimulate creative human thought. The AI software mechanism of the Colour Concept Generator is presented as an application of AI to aesthetic design.

Contents

1 Introduction

1.1 Inspiration 1
1.2 Previous Research 1
1.3 Scenario 2
1.4 The Idea and Aim of the Research 3
1.5 Initial Problem Statement 4
1.6 Thesis Outline 5

2 Literature Review

2.1 Introduction 6
2.2 Physical Definitions of Colour 7
2.3 Visual Effects of Colour Perception 9
2.4 Psychological Interpretations of Colour 11
2.5 Symbolic Use of Colour 12
2.6 Semiotics 15
2.6.1 Descriptive Attributes of Colour 15
2.6.2 Universal and Relative Colour Semiotics 16
2.6.3 Colour Lexicons 17
2.6.4 Theoretical Models of Colour Semiotics 18
2.7 Colour in Commercial Design 21
2.8 Artificial Intelligence (AI) 24
2.8.1 Background to AI 25
2.8.2 Examples of AI Systems 26
2.8.3 Computer Models of Colour Semiotics 29
2.9 Summary 31

3 Problem Review

3.1 Introduction..35
3.2 Problem Analysis.. 36
3.2.1 Product Colour Aesthetics..37
3.2.2 Integration of the System into the Product Design Process.............. 38
3.2.3 Translating Between an Aesthetic Description and a Colour Concept.. 39
3.2.4 Criteria for Forming Colour Semiotic Relations..............................40
3.2.5 The Form of a Colour Concept.......................................42
3.2.6 Form of an Aesthetic Image or Style Description.............................42
3.2.7 Aesthetic Descriptions from Product Designers.............................. 44
3.2.8 Colour Semiotics in a Cultural Context..45
3.2.9 Machine Learning of Colour Semiotic Relations............................. 46
3.2.10 Generating Colour Concepts.. 48
3.3 Detail Specification of the System................................ 50
3.3.1 Boundary of the System.. 50
3.3.2 Ethos for the Role of the System.................................. 50
3.3.3 Colour Concept Display.. 51
3.3.4 Interface Menu and Functions...................................... 52
3.3.5 Colour Semiotics Learning Input...................................53
3.3.6 Colour Semiotics Learning Process................................. 53
3.3.7 Colour Concept Generating Input...................................54
3.3.8 Colour Concept Generating Process................................. 55
3.3.9 Colour Concept Generating Methodology............................ 55
3.3.10 User Process... 56
3.4 Summary.. 57

4 Computer System Design

4.1 Introduction..61
4.2 System Architecture Design..63
4.2.1 Neural Systems... 63
4.2.2 Perceptrons.. 64
4.2.3 Artificial Neural Networks (ANN).................................64
4.2.4 Learning in an ANN... 65
4.2.5 Neural Semantics... 66
4.2.6 Intelligent Knowledge Based Systems (IKBS)...................... 66
4.2.7 Production System Architecture................................... 67
4.2.8 Colour Concept Generator Architecture............................ 68

4.3 Data Structure Design....69
4.3.1 Database in a Production System....69
4.3.2 Rulebase in a Production System....70
4.3.3 Colour Concept Generator Data Structure....70
4.4 Inference Mechanism Design....71
4.4.1 Inference Engine in a Production System....71
4.4.2 Forward and Backward Chaining....72
4.4.3 Colour Concept Generator Inference Engines....72
4.5 Matching Rules in a Rulebase.... 74
4.5.1 Methods of Matching Rules....74
4.5.2 Rule Matching in the Colour Concept Generator.... 74
4.6 Conflict Resolution Strategy....76
4.6.1 Methods of Conflict Resolution....76
4.6.2 Conflict Resolution in the Colour Concept Generator.... 76
4.7 Programming Language and Development Environment....77
4.7.1 Prolog....77
4.7.2 Lisp....79
4.7.3 AutoCad and AutoLisp.... 79
4.8 System Design Specification.... 81
4.8.1 Methodology for Analysis and Design.... 81
4.8.2 Computer Media.... 82
4.8.3 Database Design....82
4.8.4 Current Colour Set....82
4.8.5 Current Descriptor Set.... 82
4.8.6 Rulebase Structure.... 83
4.8.7 User Interface.... 83
4.8.8 Menu Commands.... 85
4.8.9 Learn Inference Engine....86
4.8.10 Generate Inference Engine....86
4.8.11 Describe Inference Engine.... 87
4.8.12 File Handling Process.... 87
4.9 Example of System Processes.... 87
4.9.1 Rulebase Learning of Colours and Descriptors.... 88
4.9.2 Generating the Current Colours List....89
4.9.3 Producing the Associated Descriptors List....90
4.10 Sample Test Run.... 91
4.10.1 Colours and Proportions....91
4.10.2 Descriptors and Proportions.... 92
4.10.3 Creating the Colour Concept Pie-Chart.... 92
4.10.4 Removing Colours from the Colour Concept Pie-Chart.... 93
4.10.5 Re-proportioning Colour in the Colour Concept....93
4.10.6 Creating the Description....94
4.10.7 Removing Descriptors from the Description.... 95

4.10.8 Learning a Colour Semiotic Relation........ 95
4.10.9 Generating a Colour Concept........ 96
4.10.10 Describing a Colour Concept........ 97
4.11 Summary........ 97

5 Evaluation

5.1 Introduction........ 100
5.2 Experimental Method Using Product Designers........ 101
5.2.1 Participating Product Designers........ 101
5.2.2 Format and Organisation........ 101
5.2.3 Questionnaire Design........ 102
5.3 Results from the Questionnaires........ 103
5.3.1 Learning and Operating the System........ 104
5.3.2 Creating Colour Concepts........ 105
5.3.3 Creating Descriptors........ 112
5.3.4 Teaching Colour Concepts........ 115
5.3.5 Generating Colour concepts........ 119
5.3.6 Describing Colour Concepts........ 120
5.3.7 Operating Methodology........ 121
5.3.8 The Design Process........ 123
5.4 Evaluation of the System Philosophy Based Upon Observations and Discussions........ 124
5.4.1 Integration of the System into the Design Process........ 124
5.4.2 Objective Criteria for System Learning of Colour Semiotics........ 125
5.4.3 Translating Between Aesthetic Descriptors and Colour Concepts........ 126
5.5 Evaluation of the System Methodology Based Upon Observations and Discussions........ 128
5.5.1 Learning Colour Semiotic Relations........ 128
5.5.2 Generating Colour Concepts from Aesthetic Descriptors........ 129
5.5.3 Describing Colour Concepts........ 131
5.6 Evaluation of the Computer Media Based Upon Observations and Discussions........ 133
5.6.1 Production System........ 133
5.6.2 AutoLisp........ 135
5.7 Summary........ 136

6 Conclusions and Recommendations for Further Research

6.1 Introduction..139
6.2 Conclusions.. 139
6.3 Further Research.. 142

References.. 143

Appendices.. 147

Appendix A Aesthetic Descriptions...148
Appendix B System Code.. 151
Appendix C User Interface..172
Appendix D Operating Instructions...175
Appendix E Evaluation Questionnaire...................................... 185
Appendix F Colour Concepts...193
Appendix G Data Flow Diagrams.. 208

Acknowledgements

This research and thesis would not have been possible without the kind help of the following people:

Professor Peter Hogarth for agreeing to fund the research project within the Department of Product Design and Manufacture at Bournemouth University.

Dr Kamran Tabeshfar for directing the research as first supervisor, believing in the idea in the first place and always having an open door.

Dr Martin Lefley for being a second supervisor on all aspects of the research.

Nik Hills for being a second supervisor on aspects related to product design.

Ray Maycock for being an advisor on colour in context.

Andy Wilson for helping me with AutoCad and AutoLisp.

Lynne Rutter for finding information from the most remote recesses.

Gareth Angel for photographing the colour concepts.

Sarah Palmer for commenting upon the system.

Peter Edwards for commenting upon the thesis.

Tim Hartley for teaching me musical scales.

Zoe Brocklesby for helping me type the thesis and being just Zoe.

Glossary of Terms

Artificial Intelligence (AI), machine simulation of human thought processes.

Chromatic, of or in colour.

Concepts, ideas for possible solutions to aspects of a design.

Expert System, a machine that simulates an expert in a specific domain.

Inference, the reasoning of a decision from knowledge.

Lexicon, a structured definition of terms (eg. a dictionary).

Neural Network, machine simulation of human brain neurones.

Semantics, the meaning or interpretation of any form of communication.

Semiotics, the implication or connotation of signs and symbols.

Symbolic, something that stands for or represents something else.

1 Introduction

1.1 Inspiration

The original inspiration for this research came during a BEd(Hons) degree in Design and Technology when research into human creativity highlighted a psychological state of mind termed 'Synesthesia'. Synesthesia is a mental process involving the overlapping and interconnecting of human sensory perception mechanisms, eg. seeing sound, tasting colour. It is experienced by many creative people including the painter David Hockney and was reviewed more recently in a Horizon documentary aptly titled 'Orange Sherbet Kisses' (1).

Further inspiration was gained from the work of Mike Leonard (2), a lecturer in Architecture at Regent Street Polytechnic at the same time as the members of the pop group 'Pink Floyd' were students. Leonard created light machines to project coloured light onto the stage as the Pink Floyd played their music. The light displays were intended to complement the music to be experienced simultaneously as integrated media. The combination of sound and vision became indicative of Pink Floyd performances up to and including the present day where the process involves more sophisticated computer technology.

1.2 Previous Research

The inspiration gained from synesthesia and Leonard's light machines led to a final project during a MSc degree in Computer Studies entitled 'The Development of a System to Produce Computer Graphical Art Representative of Musical Input' (3). The aim of the project was to explore the objective association of musical and graphical structures and develop a system capable of generating and manipulating visual coloured images in response to musical input.

The project involved the application of real-time parallel processing using transputer technology and was programmed in the Occam language. The relations between colour and sound were based purely upon physical associations, ie. frequency and amplitude. A selection of graphical structures were created based upon the mapping of colour onto musical frequency and dimension onto amplitude. Two versions of the system were produced, the first culminated in a final 'picture' and the second was in real-time response.

One of the conclusions of the MSc thesis was that the relationship between colour and music needed to include the more expressive side of experiencing music and colour. It was proposed that an AI interface between the two media could interpret music and translate it into coloured images more indicative of the musical impression. It is from the basis of this point that the idea for this research project developed.

1.3 Scenario

Product Design is a broad area that includes diverse fields encompassing the aesthetic and functional. A product has an aesthetic style characteristic of the desired image it is intended to project. The aesthetic attributes can be defined as those relating to colour, shape, texture and form, as well as detailed features that give the product a certain finesse and combine as an aesthetic concept or theme.

The colour of a product is limited by the available technology such as the material and manufacturing process as well as available pigments and dyes. The advancement of colouring process technology promotes the feasibility to produce products in a greater variety of colours to appeal to a broad range of consumers.

Commercial product aesthetics and the competition for maximum sales could be enhanced by appropriate colour concept design, as the consumer might choose a product for its colour at the expense of purchasing a less sophisticated product with respect to other design criteria. This could be cost effective in terms of product development, as it might be significantly cheaper to change the colour scheme of a product than to redesign it. A 'new' product could be launched in a different colour scheme and perceived as new even though the only 'new' attributes are related to colour.

Colour can provide a corporate image for a product expressing a distinctive style and association with a new or established identity or status of a commercial enterprise. Colour can be used to group a product range or indicate by distinction the hierarchical status of a line of similar products. Colour gives the designer an additional creative medium and dimension to control the impact of a desired aesthetic character for a product.

1.4 The Idea and Aim of the Research

The main aim of this research was to design a computer tool for a product designer to generate colour concepts for products. The tool was to translate an aesthetic image or style description (originating from a product design brief) into a combination of colours representative of the aesthetic description.

The design of products is a holistic process that includes the human, aesthetic and technological aspects resulting in products that possess colour, shape and form. The holistic design ideal is supported by a design process or methodology for which many computer systems exist to assist the product designer. These range from 2D drafting (eg. 'AutoCad'), to 3D solid modelling (eg. 'Alias'), to finite element analysis (eg. 'Pro Engineer'), to fluid analysis (eg. 'Ideas'), to expert systems that borrow from the field of AI and deal with the more specialised knowledge and experienced based problems such as product material and process selection.

Most of the available computer systems deal more with the technical aspects of product design and are mostly used at the development stage of the design process. The development of AI promotes the feasibility to provide front-end computer tools capable of assisting the product designer with the more conceptual aspects of design encountered in the initial stages of the design process. An example is morphological analysis of a 3D product using multi-variable parametrics to manipulate product shape and spatial relations (4).

The idea for the Colour Concept Generator was to link a desired aesthetic image or style to the creation of an appropriate colour concept for a product as a form of ideas stimulus for the product designer. It was envisaged that the computer tool would be placed at the ideas and concepts stage of a design methodology as this is where other conceptual design is performed. The research has been an experiment into whether an AI computer tool can be produced to deal with a more humanistic aspect of product design (colour aesthetics) at the front end of the design process.

1.5 Initial Problem Statement

The problem was initially defined as the translation between semiotic media, ie. between the description of an intended aesthetic image or style for a product and a colour scheme. A software mechanism was to be produced that could learn these colour semiotic connections from which colour concepts could be generated. The issue of how to do this objectively was considered and it was decided to propose an objective method that integrated with the software mechanism although a subjective personal interpretation was also allowed.

The idea of computing colour aesthetics was seen as a controversial notion. It implied the capability of a computer machine to deal with phenomena commonly related to factors deemed characteristic of human creativity and expression. Colour is an essential ingredient in the composition of visual art providing an added dimension and medium for expression and communication of human thought and emotion. It was a controversial idea to involve computer media in an intrinsically human process. The idea of computing colour aesthetics was seen as providing an extension and reflection of human creativity to be used to enhance rather than stifle creativity.

.

The process of product design is such that although colour can be used in a creative way the product is designed to constraints and specifications. The pure intrinsic expression found in art is less prominent in product aesthetics through a need to be more objective. The colour aesthetics reflect a kind of product character or personality which does require expressive use of colour but to serve a specific purpose of reflecting a pre-defined image or style. The Colour Concept Generator was seen as providing an objective translation of an aesthetic character description into a representative colour scheme.

1.6 Thesis Outline

The rest of this thesis is divided into five chapters. Chapter two is a review of the available literature deemed most relevant to the research. The information is diverse in origin and transcends many academic fields. The chapter is divided into the following parts: Colour as a physical attribute of light, visual effects of colour perception, psychological interpretations of colour, symbolic use of colour, descriptive attributes of colour, universal and relative colour semiotics, colour lexicons, theoretical models of colour semiotics, colour in commercial design, a background to artificial intelligence (AI), examples of AI systems and computer models of colour semiotics. The chapter is then summarised in terms of the research.

Chapter three is a review of the problem as a whole and the idea of computing colour aesthetics. The problem is analysed in terms of: Product aesthetics, integration of the computer system into the product design process, translating between aesthetic descriptors and colour concepts, a criteria for forming colour semiotic relations, the form of colour concepts, the form of an aesthetic image or style description, aesthetic descriptions from product designers, cultural context, machine learning and generating colour concepts. From the analysis a detail specification of the Colour Concept Generator is presented defining the boundary, ethos and methodology of the system. The chapter is then summarised in terms of the problem.

Chapter four is the design of the computer system beginning with an outline of the selected computer media regarding the system architecture, programming language and development environment. The design of the Colour Concept Generator system architecture and processes are then presented using data flow diagrams from which the system was coded. The chapter is then summarised in terms of the computer system. Chapter five is an evaluation of the research beginning with an outline of the experimental method in line with the computer system methodology. The research is evaluated in terms of the philosophical idea of computing colour aesthetics, the system methodology for learning colour semiotics and generating colour concepts and the computer media used. Chapter six contains conclusions of the research based upon the evaluation and proposals are put forward for future development of the project as recommendations for further research in the field.

2 Literature Review

2.1 Introduction

This chapter reviews the existing literature deemed most relevant to the research. The material is broad in nature and includes the physical, perceptual, psychological, symbolic, and semiotic aspects of colour as well as the use of colour in design. A background to artificial intelligence (AI) and examples of AI systems are provided as well as existing computer systems for colour semantics and semiotics.

The chapter begins with a look at the physical definitions of colour, as in order to understand colour semiotics it was considered necessary to understand the medium of colour. An aim was to see if the physical structure of colour could be mapped onto a semiotic structure of colour meaning. There then follows the human perception of colour that was considered in order to establish visual effects of combining colours. An aim was to see how placing different colours next to each other altered the image that was actually seen.

The psychological interpretations of colour considers the relationship between colours and mental states and was looked at to see if colour had any influence at a subliminal level. An aim was to see if colour imagery influenced or was the result of patterns of mental behaviour and to see if there were any semiotic connections. This is followed by an examination of the symbolic use of colour as a possible means of objectively forming colour semiotic relations. The use of colour symbolism gave a useful insight for translating between aesthetic descriptions and colour concepts.

Semiotics was considered as the philosophical basis for the whole research project. An aim was to find theoretical mechanisms or some form of logistics that could be used to translate aesthetic descriptions into colour concepts. This is followed by the use of colour in commercial design that was looked at to see how decisions were made in a commercial situation. Methods for selecting colour schemes were found to range from designer led decisions to research based field tests.

The background to artificial intelligence and the examples of AI systems were reviewed to gain insight into the origin and development of AI. It was envisaged that the computer system would require a certain degree of intelligence as it was to deal with conceptual design, so it would require some kind of reasoning strategy. Existing computer models were found that dealt with colour naming and colour descriptions but did not go into any detail or cover the breadth of colour semiotics for aesthetic descriptions of products.

2.2 Physical Definitions of Colour

The physical definitions of colour were initially looked at to establish what colour is as an entity. This was considered to understand the characteristics and attributes of colour to see if what colour *is* could be in any way related to what colour has been termed to *mean*. An aim was to see if the physical structure of colour could be mapped onto a semiotic structure of colour meaning and to see if physical attributes (eg. frequency) were characteristic of semiotic meanings (eg. speed, as in 'fast red').

To Aristotle and the ancient Greeks (5) colour was attributed to the interaction of light and dark, as daylight bordered night and the sky portrayed the appearance of successively varying colour. Colour was associated with the boundary edges of light and dark and lay between the extreme poles of black and white.

Leonardo Da Vinci (6) also held Aristotle's view of colour. At the two extreme poles of a linear colour model Leonardo placed light and dark whereby the light was darkened to form yellow and the dark was lightened to form blue.

Johann Wolfgang Von Goethe (7) also reasoned colour in terms of Aristotle's theory of boundary conditions between light and dark. He concluded in line with Leonardo, light darkened to yellow then yellow-red (orange) and dark lightened to blue then blue-red (violet). Goethe classified three modes of colour: Physiological, Physical and Chemical.

Aristotle's theory held until Isaac Newton (8) demonstrated the presence of the colour spectrum (red, orange, yellow, green, blue, indigo, violet) in white light. Newton was not the first to do this, the Dalmation Jesuit, Marco Antonio De Dominis (5) described the same model but explained the spectrum in terms of Aristotle's theory. Newton postulated that light behaved as waves that differed in wavelength for each colour.

James Clerk Maxwell (9) discovered the electromagnetic spectrum placing the visible light spectrum between infra red and ultra violet from red (long wavelength, low frequency) decreasing in wavelength and increasing in frequency to violet (short wavelength, high frequency). The waves (not coloured themselves) appeared as colour when reflected from a surface.

The development of Quantum theory explained that different colours possessed different light energy. Phillip Leonard (5) discovered the photo-electric effect, Max Planck (5) proposed that atoms omit and absorb quanta and Albert Einstein (5) proved that light is composed of quanta (photons). Photons of different energy for different light wavelengths were found to increase in energy through the colour spectrum from red (low energy) to violet (high energy).

Physically colours are waves of light that differ in frequency and energy through the spectrum from red (low frequency and energy) to violet (high frequency and energy). The semiotics of colour do not necessarily correspond to these physical attributes as culturally red is typically associated with fast and high energy, eg. sports cars in 'fast red'. The highest energy source, the sun, is perceived as yellow which is only a medium energy colour, so determining what colour *is* does not necessarily equate to what colour has become termed to *mean*.

2.3 Visual Effects of Colour Perception

Colour perception was considered to understand how humans see colour and the visual effects of colour perception. This was looked at to realise the result of what happens when combinations of different colours are seen together. An aim was to see how combining colours (eg. a white dot on red) altered what was actually seen (eg. white dot looks green). In the design of products that include visual interfaces (eg. a calculator) an understanding of the perceptual effects of adjacent colours would be critical to allow for visual clarity.

Goethe (7) defined colour perception in terms of opposing states, as a plus and minus polarity. In a circular colour model he positioned yellow (light) opposite blue (darkness). He categorised opponent colour pairs, diametrically opposed to each other in a circular colour model: Blue and yellow, red and green, orange and violet. Goethe referred to colour perception phenomena termed simultaneous and successive contrast.

Simultaneous contrast is the perception of opposite coloured 'halos' impeding upon adjacent colours. The opponent colour of a colour impedes upon any other colours placed adjacent to it and alters the perception. Successive contrast is the perception of opposite coloured 'after-images' after a period of viewing a coloured image. An image can be seen in the opponent colour of the viewed image for some time after.

The visual effects of opponent colours were observed by Eugene Chevreul (10), a colour chemist at the Gobelins tapestry works near Paris. Coloured fabrics were being returned to Gobelins due to claims that they were soiled. Chevreul found that the soiled effects were due to simultaneous and successive contrast. Both phenomena were used by the Pointillist painters (inspired by Chevreul's findings) to create the perception of colours that were not physically present.

M. Bagley and M. S. Maxfield (11) conducted research into the perception of colour, to establish complementary colour matches with reversible after-images that were of use to designers. The reversibility of after-images generated a range of hues that were close but not totally identical as there was no single colour match for a single after-image. The broadest range spanned four hue families and the remaining nine colours spanned three hue families. The majority of responses belonged to a single hue family and the after-images all appeared light in value and tint. Reversibility was not exact as some were virtually reversible yet some were not consistent.

Johannes Itten (12) organised the visual effects of colour contrast into seven categories: (i) Contrast of hue was that pertaining to different colours in a colour combination. (ii) Light and dark contrast referred to a difference in the brightness of colours in a colour combination. (iii) Cold and warm contrast compared colours according to a visual feeling. (iv) Complementary contrast related to the combination of opponent colour pairs. (v) Simultaneous contrast was associated with the visual effects of a colour upon adjacent colours. (vi) Saturation contrast referred to the difference in intensity of colours in a combination and (vii) contrast of extension was to do with the proportions of colours in a colour combination.

The contemporary physiological view of the human colour perception process is a combination of the tri-chromatic (8) and opponent (13) theories. Light is detected by a chemical reaction in an array of receptors in the retina called cones. There are three types of cone to detect red (long wavelength), green (medium wavelength) and blue (short wavelength) light. The electrical responses of a red and green cone form an opponent colour pair. The combined resultant electrical response of a red and green cone (yellow) form an opponent colour pair with that of a blue cone. The electrical responses of red, green and blue cones form a black and white opponent colour pair. The opponent processing is performed in a neural circuit of cells prior to transmission to the brain via the optic nerve (14).

Colour is a visual perception that exists in the mind and is a result of processing light energy via the eye and brain. Variations of contrast in colour combinations influence the perceived colour image. The colours (hues), lightness of colours, complementary colours, the saturation or intensity of the colours and proportions influence the visual image. Simultaneous contrast influences the perception of adjacent colours and successive contrast affects the impression of colour images viewed over a period of time. These effects contributed to the design of the form of a colour concept in the computer system and the visual interface as well as the way in which colour semiotic relations were learnt.

2.4 Psychological Interpretations of Colour

The psychological interpretations of colour were investigated to see if colour could relate to different states of mind. Colour was found to be used as a psychological diagnostic tool as well as a medium for personality tests. The presence of colour imagery in dreams was looked at to see whether colour had subliminal associations manifest in the subconscious mind. An aim was to see if colour imagery influenced or was the result of patterns of mental behaviour and to see if there were any semiotic connections.

Carl Gustav Jung (5) believed that the subconscious held clues regarding a psychological state of mind, subliminally exhibited in dreams and paintings in which colour imagery played a characteristic role. A similar approach was taken by Sigmund Freud (5) who used the colours seen in his patient's dreams to analyse their psychological state of mind. Jung found pseudo-scientific medieval Alchemist texts that paralleled his belief and subsequent theory, he believed he had stumbled upon the historic counterpart of his psychology of the unconscious.

The Alchemist philosophy was the purification and transformation of the self, a process termed Magnum Opus (Great Work) and was symbolically allied to the progression through the Alchemist spectrum, green, black, white, red to gold. Jung found that the Alchemist use of colour symbolism directly correlated with images in his patients' dreams and spontaneous paintings. He believed that the correlation was related to a common memory inherited from distant ancestors, locked in the subconscious mind. Jung proposed that progression through the Alchemist spectrum was directly related to psychological development, he had, he believed found the key.

H. Rorshach (5) claimed that the description of colour images expressed by patients gave an indication of their psychological state of mind. He believed that an interpretation of colour mirrored aspects of the subconscious, where lay the essence of a psychological make-up. Rorshach devised a diagnostic test consisting of a series of ten coloured ink blots, the patient projected an image onto each blot, upon which was based a diagnosis of the psychological condition. Rorshach claimed that a patient's response to colour reflected their typical method of dealing with affect and that failure to respond could be just as significant as a detailed description.

Max Luscher (15) developed a personality test based upon the selective ordering of colours. Luscher believed that personality could be analysed from the order in which colours were chosen. The colour chosen first was most liked and indicated a means to achieve objectives, the second indicated the objective, the third and fourth were the actual state of affairs of the present situation and the existing circumstances. The fifth and sixth indicated an indifference or reservation, the seventh and eighth referred to an unsympathetic, rejected inhibited need. Luscher believed that colour portrayed a profound psychological significance, exemplifying a subliminal nature.

The psychological interpretations of colour as a diagnostic tool indicates the prevalence of colour imagery in the subconscious mind. Subliminal messages could be decoded from the colours seen in dreams and personality characteristics established through a colour selection process. This was extended to the point of understanding stages of mental change and psychological development. The information showed that colour can have influences at a subliminal level to illustrate that colour semiotics could be a conscious or subconscious activity.

2.5 Symbolic Use of Colour

Colour symbolism was examined as a possible objective criteria for forming colour semiotic relations. Colour symbolism was contemplated as a less subjective method as interpretation is based upon reference to existing entities. Semiotic associations could be reasoned and justified by the object that the colour represents, eg. red is 'danger' as a symbol of blood. The use of symbolism highlighted the use of any media as being 'something that stands for something else'. This provided a useful insight for translating between aesthetic descriptions and colour concepts.

Chinese colour symbolism (16) was based upon the association of colour with the five elements, expanded to include compass point locations and the seasons. The five basic colours were white (gold, west and autumn), blue (wood, east and spring), black (water, north and winter), red (fire, south and summer) and yellow (earth, centre of the earth and all seasons). The red of fire represented the positive essence Yang and the yellow of gold was the negative essence Ying.

The Hindu Upanishads (6) allocated red to fire, white to water and black to earth and the Jewish historian Josephus (6) allocated white to earth, red to fire, purple to water and yellow to air. To Aristotle and the ancient Greeks (6) white was of air and water, yellow of fire and the sun. White was also of the earth and became black for elements in transmutation as a symbol of smoke. Empedocles (6) described colour as the soul of life and the root of all existence, yellow symbolised earth, black was of air, red of fire and white of water.

Leonardo Da Vinci (6) arranged the relationships between colour and the elements in order from light to darkness. White was light, yellow was earth, green was water, blue was air, red was fire and black was total darkness. Itten (12) associated colours with seasons, spring was yellow, yellow-green, light pink, light blue and lilac and autumn was dull brown and violet.

Victor Turner (5) conducted research into the use of colour symbolism by the Ndembu tribe in Central Africa. The symbolic meaning of white, red and black highlighted a raw imagery and indicated primitive tribal associations to the functional and physical aspects of the human body.

The meanings changed according to the context in which they were applied. White related to the giving of new life, symbolic of the mother's milk and semen and was a prominent colour in male initiation rites. As water, white represented 'cleansing and purification' allied to the white-washing of pubescent menstrual females. Red was symbolic of 'power' in the blood-shed as the hunter killed for food and subsequent survival. Red was the living of life and 'action' as well as the agonies and pain endured in doing so. Black was an image of 'war' and the inevitable coming of death. In mourning the tribes-women would remain dirty (black) to symbolise the departing of a life.

Rummel (6) developed a cosmetic principle based upon colour symbolism and interpreted meanings. Red was believed to express cognitive signals manifest in symbolic implications of the tongue or genitals. He interpreted red as 'attraction, attention, wonder, sexual exhibition, temptation and exploration'. Dark blue or green were deemed emblems of 'possessing, belonging or caring', symbolic of the home and unity. He represented yellow, orange and white as 'extroverted, forbidden, prohibited and revered' and were the colours of priests, prophets and hermaphrodites. Black and violet were for the 'privileged and distinctive' yet also stood for 'illness, slavery, treason and widowhood'.

Newton (17) paralleled the 'magic seven' colours of the spectrum to the seven notes on the musical scale, C (red), D (orange), E (yellow), F (green), G (blue), A (indigo), B (violet). The colour spectrum was described by Itten (12) as one harmonic interval representing one octave. August Aeppli (17) claimed that colour and sound were related by a factor of pi on the basis that sound = time = diameter and colour = space = circumference. A 'B' flat (224 Hz) was symbolised as violet (754 nm wavelength = 224 x pi), however not all wavelengths corresponded to the assumed colours.

Kandinsky (17) related colour to musical instruments and styles. Red (fanfares and drums) was persistent, intrusive and powerful and orange was a church bell. Yellow (a loud trumpet) was powerful intense and intolerable and green a quiet, meditative violin. Blue was a flute, progressively darkening to a cello and base organ. Purple (a violin) was youth and joy and violet represented the woodwind instruments. Goethe (17) rejected the idea that colour and sound were symbolically related and described the difference as comparing two rulers side by side that differed in length and mark.

The use of colour symbolism for developing colour semiotic relations illustrated a possible method of being objective about a subjective perception. Colour interpretation by reference to other objects (eg. red for blood is 'danger') could provide a justification for reasoning colour semiotic relations. Symbolism showed how object representation can be used as a form of communication of a concept. A potential problem was that not all cultures were in agreement with interpretation of colour symbols. Reference to other objects might require that those objects had been seen by whoever was being shown the colour symbol. In this sense having knowledge of a 'code' of colour symbolism might give preconceptions or misunderstandings. The 'code' might be acquired or learnt culturally at either a conscious or subconscious level.

2.6 Semiotics

Semiotics was considered as the philosophical basis for the whole research project. An aim was to find theoretical mechanisms or some form of logistics that could be used to translate aesthetic descriptions into colour concepts. The characteristic attributes of a colour image that could be described were analysed to see what types of words could be used in an aesthetic description. Cultural and contextual issues were deliberated in the debate of whether colour semiotics is a universal or relative concept. The theory of the structure of a lexicon was used to see how colour semiotic knowledge might be organised and presented. Theoretical models of colour semiotics from colour theorists were looked at to see examples of structures for relating colour semiotic knowledge.

2.6.1 Descriptive Attributes of Colour

Possibly the earliest debate regarding the semiotic nature of colour was a conversation held in Rome in the 2nd century AD between Fronto (a poet) and Favorinus (a philosopher) (18). The debate revolved around the comparison between the colour vocabulary of Latin and that of Greek. Favorinus remarked that the eye was able to isolate more colours than words could name (unknowingly introducing a contemporary scientific distinction between identification and discrimination) and that Latin made sole reference to coloured objects. The debate developed into an erudite argument around poetic invention, metaphor, simile, rhetoric and precise colour categorisation with analogies drawn from the respective cultures.

Cicero (19) classified one of the five parts of rhetoric as invention that contained a subset of artificial proof consisting of three types, Ethos (evocation), Pathos (emotion) and Logos (logic) allied to verbal language communication of an expression or concept. Regarding visual language communication of an illusion or image, the painter Grunewald (19) outlined colour synthesis in three specific categories, Realistic Signification, Psychological Expressive Power and Symbolic Verity fused as a single entity.

The parallel between verbal and visual communication was exemplified in Itten's (12) observation of three exhibitions of colour, Impression (visual), Expression (emotion) and Construction (symbolism). Impression was to do with what was seen and was influenced by visual characteristics relating to appearance. Expression pertained to the sensory and mood response to colour and Construction was the symbolic imagery of colour inherent in the associative relation of colour to external entities.

Itten developed a system for colour semiotics using red, yellow and blue as a base. The semiotic meaning of mixtures of these colours was subsequently derived from the combined meaning of the base colours, eg. green (yellow + blue) expressed contentment (knowledge + faith). A kind of semiotic lexicon tree structure was formed with the base colours at the root, branching out to combined colours and meanings as more colours were mixed.

2.6.2 Universal and Relative Colour Semiotics

Brent Berlin and Paul Kay (20) allocated eleven basic colour names after a study of 98 different languages by asking participants to indicate the foci of a particular colour from a range of colour chips. The colour terms were subsequently ordered, (black / white), red, (yellow-green or green-yellow), blue, brown, (purple / pink / orange / grey). Berlin and Kay postulated that as a language developed, the basic colour vocabulary evolved accordingly and claimed that colour semantics was a universal concept common to all cultures.

Umberto Eco (18) highlighted the semiotic problem of identification (categorisation) and discrimination regarding colour naming with reference to coloured objects. He formed a semiotic relationship between linguistic expression (reference) and cultural content (meaning). Eco described a signification system deemed essential for communication. The system formed a relationship between expression and content whereby each portrayed a substance and a form. The substance of linguistic expression he termed 'etic' (sound) and the form was called 'emic' (category). The content of the signification system became dependent upon the emic categorisation.

Eco stated that the content meaning depended upon the number of variables (words) which in turn influenced the ability to categorise concepts as analysis was performed by reference to other expressions. The problem of colour semiotics lay in describing something without the available words. Eco described perception as an intermediate process between sensory discrimination and semiotic categorisation as a form of selective adaptation. He stated that the perception of colours was to do with the emic analysis of expressions and not the correlation between expression and content. Eco claimed that colour semiotics depended upon cultural content systems and was a relative concept.

Rolf Kuschel and Torben Monberg (21) researched colour semantics on Bellona Island to find that the Bellonese had no term for colour. They differentiated surfaces with terms like 'vary-wayed' and 'vary-kind' but these were also used to describe other things such as skin and smell. The only colour terms used were white (light), black (dark) and red, yet some included white (light) on the grounds that this was merely a removal of the others. Colour blind tests were used, however not one case of colour blindness could be found.

2.6.3 Colour Lexicons

G. A. Miller and P. N. Johnson-Laird (22) outlined the criteria required in the formulation of a lexicon whereby each lexicon item must include, phonological shapes, meaning, properties, role of predicate arguments, apt use conditions, and conceptual and morphological relations to other lexicon items. They divided colour terms into nouns and adjectives, abstract nouns became metaphoric adjectives. Nouns denoted the position of a perceptual property relative to the concept, adjectives formed predicates to specify or weight the perceptual property in denoting the noun.

Miller and Johnson-Laird proposed that a lexicon definition had formal and functional properties of three types, Definitional (fixed), Characteristic (fixed / variable) and Peripheral (free variable). The relational description followed a three stage algorithm:

(1) Does noun have property that adjective could specify ?
IF peripheral property THEN co-join two schemata.
IF characteristic THEN (2).

(2) Interpret adjective within characteristic range for property of noun.
IF fail THEN (3).

(3) Replace characteristic weighting with adjective weighting for property.

The process was one of relating words to objects whereby colour labels were learnt through a process of repeated pattern matching with lexicon items analysed into semantic components.

The Inter Society Colour Council for the National Bureau of Standards (ISCC-NBS) (22) developed a colour lexicon consisting of a 28 term nomenclature which included combinations of red, orange, yellow, green, blue, violet, purple, pink, brown and olive with 'ish' suffixes. The terms were based upon a 3D colour model with hue on the circumference, lightness on the vertical axis and saturation on the horizontal axis. The contrast terms were divided into lightness (very dark, dark, medium light, very light) and saturation (greyish, moderate, strong, vivid). Combinations of lightness and saturation gave rise to brilliant (light + strong), pale (light + greyish) and deep (dark + strong).

2.6.4 Theoretical Models of Colour Semiotics

Goethe (23) polarised yellow (plus, active) and blue (minus, passive). Yellow he termed 'action, light, brightness, force, warmth, proximity, repulsion and acidic'. Blue he termed as 'negation, shadow, darkness, weakness, coldness, distance, attraction and alkali'. Green (yellow + blue) was a 'harmonious relation or union'. Goethe depicted 'plus' colours as yellow, red-yellow (orange) and yellow-red and 'minus' colours as blue, red-blue and blue-red.

The plus and minus colour categories formed two extremes of a colour semiotics chart depicting 'the sensual and moral effects of colour'. The chart began with red (strong) at the top as the highest of all colour manifestations and ideal satisfaction. The plus colours, initially 'serious, dignity and warm' ranged from yellow-red (vigorous, convulsive) to red-yellow (splendid, agreeable, gently stimulating, cheerful, alert) to yellow (light) where they became 'passive, turbulent and soft longing'. The minus colours, initially 'grace, charm and dark' ranged from blue-red (turbulent, unbearable) to red-blue (lively, shadowy, wide open) to blue (cold) where they became 'active, agile, lively and aspiring'. Green (weak) was at the bottom as 'simplicity and real satisfaction'.

Goethe designed a colour tetrahedron as 'a symbol of soul power' which consisted of red (imagination), yellow (reason), green (sensuality) and blue (understanding). This led to a diagram of 'the symbolic significance of the colours in the circle' which included orange (noble), yellow (good), green (useful), blue (common), violet (superfluous) and purple (beautiful).

Itten (12) developed a colour semiotic system based upon a tree structure. Red, yellow and blue formed the base from which all other colours and meanings grew through colour mixing and amalgamation of meanings. Itten reasoned that yellow was the brightest and most intelligent colour so it represented 'knowledge and truth'. Diluting yellow resulted in 'envy, betrayal, falseness, doubt, distrust and unreason'. Red symbolised 'power and the active', associated with 'war, revolution and blood'. Red ranged from red-orange (dense, opaque, warm, glowing, passion, vitalised earth, physical and spiritual love) to red-blue (purple, of cardinals and spiritual power) to pink (sweet, angelic). In contrast to red, blue was 'passive, cold, contracted, introverted, retiring, shadowy, superstition, fear, grief and meek' yet pointed to the realm of the transcendental, represented the immortal and was ultimately 'faith'.

The mixing of the base colours and meanings led to other colours. Orange was 'maximum radiant activity, warm, active, energy, festive, ostentation, proud self respect', the lightening of which gave beige (warm and intimate). Violet was the antipode of yellow 'consciousness' so was 'unconscious, mysterious, impressive (oppressive), menacing, encouraging, death and piety'. Violet encompassed red-violet (divine love, spiritual domain) and blue-violet (solitude and dedication). Green (yellow + blue) was the fusion of 'knowledge and faith' and symbolised 'contentment, tranquillity and hope'.

Itten's justification for colour symbolism was based upon two methods. Firstly that complementary colours possessed opposed meanings, ie. yellow /

violet (bright knowledge / dark emotional piety), blue / orange (submissive faith / proud self respect), red / green (material force / sympathy). Secondly that mixed colours had combined meanings of their component colours, ie. red (power) + yellow (knowledge) = orange (proud self respect), red (love) + blue (faith) = violet (piety), yellow (knowledge) + blue (faith) = green (compassion).

David Capon (24) devised a colour circle incorporating colour interpretation. Red (function, need, utility, economics), orange (content, substance, materials, process, construction), yellow (history, meaning, style, archetypes), green (community, empathy, human scale, vernacular, picturesque), blue (form, pattern, symmetry, unity), violet (radicalism, futurism, need, emotion). The colours were grouped into 'classical / rationalistic' (red, yellow, blue) and 'expressive / romantic' (orange, green, violet) families of concepts. Red depicted 'technological appropriateness and efficiency', yellow 'typological historicism' and blue 'mathematical purity'. Orange depicted 'interplay and substance', green 'community and communication' and violet 'politics and decision making'.

The literature on semiotics explained the essence of this research project. The process of categorising colour images, or absolutely anything for that matter was shown to be that of relating objects through pattern matching. In this sense an aesthetic descriptor (object) or description (pattern) could be linked to a colour (object) or a colour concept (pattern). The visual look (impression) of a colour (eg. luminous), the emotion (expression) it incurs or coveys (eg. tranquil) and the symbolic (construction) association (eg. white for snow is cleanliness) could all be types of descriptors that could be linked to colours.

The semiotics problem becomes more complex in accounting for the contextual issues in different cultures. A model had to be adopted that would account for various cultural interpretations without limiting the system to just one culture. The computer organisation of the colour semiotic information in the form of a colour lexicon had to be structured so that it could be manipulated in the Colour Concept Generator. The theoretical models of colour semiotics illustrated possible structures for semiotic relations whereby a translation between colours and descriptors could be performed.

2.7 Colour in Commercial Design

The use of colour in commercial design was looked at to see how decisions were made in a commercial situation. Comments from product designers and colour consultants on the use of colour gave an insight into the potential power of colour in design and the need for research in the field. Methods for selecting colour schemes were found to range from designer led decisions to research based field tests. For the Colour Concept Generator to be a tool for a product designer both methods had to be accounted for as the system had to allow for both.

Dale Russell (25) outlined the breadth of colour in the design of commercial products. Describing colour as 'the Cinderella of the design process', Russell illustrated a dismissive attitude towards colour, 'the chairman's wife chose the colour'. With reference to a selection of case studies an insight was provided into the advantage of colour in commercial product design. Regarding product development, it was claimed that it was significantly cheaper to change the colour of a product than to re-design it. Russell hit on an important point with respect to the integration of colour into the design process, that if colour specification was included at the conception of the product, it would allow greater calculated freedom. Colour aesthetic descriptions could be included in a product design specification.

The case studies included Addis Housewares, Pentagram, Jaguar Cars, Rimmel Cosmetics, Newell and Sorrell Stationary, TCS Furniture, Giant Packaging and Scantel Ltd. The Pentagram design for the Wilkinson Sword Protractor Razor exemplified the consideration of a product colour prior to the design of a product. Ken Grange (Pentagram) claimed that from the initial stages they had envisaged the razor as being red. A somewhat controversial move for a razor. A view shared by Wilkinson Sword's chief executive, commenting that red would be used "over my dead body !". However red proved successful against their leading competitor Gillette with the 'Blue II'. Wilkinson had a corporate colour image that subsequently possessed commercial appeal.

Scantel (25) developed the 'Scantest', a colour market research strategy named 'The Scantest Technique for Forecasting the Future Performance of Designs and Colours', designed as a single methodology. The first part of the test involved interviewing a cross section of the target public market. Reactions were grouped and recorded and input into a computer program that ranked the colours in order. The aim was to reduce subjectivity and optimise the colour range.

William Horton (26) pointed out the unlimited nature of colour meanings and the necessity to accept multiple colour associations and interpretations. He indicated that red was used for many different purposes, stoplights, sports cars and the world wide emblem of Socialism. Horton categorised natural colour codes, red (blood = danger, fire = heat), yellow (sun = warmth), blue (water = cool), green (young leaves = life + youth), brown (dead vegetation = age + death), paler colours (atmospheric haze = distance). An example was the age of military computer files represented with icons that progressively yellowed and turned brown.

Regarding occupational fields, Horton (27) illustrated profession related colour conventions. In finance (red = loss, black = gain), politics (red = radicalism, blue = conservatism), medicine (red = artery, blue = vein) and in industry (red = hot, blue = cold). The industrial standards (ANSI) form associations with red (danger, stop), orange (dangerous parts of equipment), yellow (caution), blue (non-safety messages), green (safety), black on yellow (radiation), and black on white (traffic markings).

Horton (26) sighted the comparative meanings of colour to different cultures. In Europe and the West, red (danger), yellow (caution, cowardice), green (safe, sour) and blue (masculine, calm, sweet, authority). In Japan, red (anger, danger), yellow (grace, nobility, childish gaiety), green (future, youth, energy) and blue (villainy). In China, red (joy, festive occasions) and yellow (honour, royalty). In Arabic, yellow (happiness, prosperity), green (fertility, strength) and blue (virtue, faith, truth). Horton stated that colour emphasised or de-emphasised, grouped spatially separate items, distinguished or discriminated items from each other and expressed quantity or quality whereby categories of items were ranked (sometimes ordered via the spectrum).

Eric P. Danger (28) divided the colour spectrum into 'warm / impulsive' (red, orange, yellow) and 'cool / retarding' (green, blue, violet). He allocated colour associations to red (warmth, passion, excitement, fire, winter, aggressive, stop), blue (cool, water, sea, summer, men, law), green (cool, country, rest, spring, go), yellow (sunshine, heat, cheerful, caution, bright), grey (dignity, quite), white (cleanliness) and pink (gentle, women). Danger went on to define 23 steps towards selecting appropriate colours for consumer products progressing from the consumer and market to trends and conclusions.

As a follow up, Danger wrote a monograph on colour consulting entitled 'The Colour Handbook' (29), by far the most extensively publicly available literature on the commercial application of colour which included a colour catalogue listing symbolic, consumer and product colour associations. The colours were divided into 'warm, cool, hard, soft, bright, muted, light, dark, pure, modified, neutral, strong and deep' categories. Each hue was then presented under headings of character, attributes, functions, applications, combinations, uses and legend.

The Nippon Colour and Design Research Institute developed the Colour Image Scale (30). The system was based upon objective classifications of personal preferences, object images, corporate images, product images and other abstract or subjective qualities rendered in common languages to be used in the development of strategies and design. The Colour Image Scale was developed to classify images commonly associated with single colours consisting of 130 colours divided according to hue and tone, with 120 chromatic colours (10 hues with 12 tones each) and 10 achromatic colours.

The colours were distributed along three axes running from 'warm to cool', 'soft to hard' and 'clear to opaque'. An image locus for each tone could be drawn by connecting the hues in order commencing with red. The colours and images were distributed equally throughout the scale enabling relationships to be made between colours and images and images and colours.

Research into the relationship between words and colours led to the development of a verbal image scale which used the same three axes to classify colour images objectively, without the influence of personal preferences or cultural differences. 180 carefully chosen adjectives were distributed throughout the three axes. For example, adjectives found on the warm-soft plane were broadly summarised as familiar and casual, those on the cool-soft plane sensitive and clear and those on the cool-hard plane reliable and formal.

Using terms from the fashion business like 'romantic' and 'elegant', image patterns were formed. Adjectives associated with similar images were then grouped into image patterns to facilitate communication of colour image information. In bringing three-colour combinations together, a Colour Combination Image Scale was developed which enabled overall images for colour combinations to be systematically determined. The descriptive adjectives of images included 'romantic, pretty, natural, clear, casual, elegant, casual (cool), gorgeous, chic, classic, dynamic, stylish and modern'.

The use of colour in commercial design showed where colour aesthetics might be considered in the design process and how colour semiotic relations were decided through interaction between designers and their clients. There was a need for the requirements of colour aesthetics to be stated earlier in the design process at the initial conception of an idea for a product. Field test research showed the possibility of consulting consumers in the decisions for selecting colour schemes and how the research information might be presented. The problem of designing a product for a world market where different cultural interpretations might contradict for a selected colour scheme indicated the need for colour semiotic learning to be organised so that cultural labels could be stated.

2.8 Artificial Intelligence (AI)

The background to artificial intelligence and the examples of AI systems were reviewed to gain insight into the origin and development of AI. It was envisaged that the computer system would require a certain degree of intelligence as it was to deal with conceptual design so it would require some kind of reasoning strategy. Existing computer models were found that dealt with colour naming and colour descriptions but did not go into any detail or cover the breadth of colour semiotics for aesthetic descriptions of products.

2.8.1 Background to AI

The idea of machine intelligence is not as new as some might claim as it dates back to the ancient Greeks. In the early 19th century Babbage created a mechanical machine for number crunching figures. In 1937 Alan Turing (31) produced the Turing Machine that could solve any problem that could be accurately specified in a finite amount of time and in an abstract manner using a binary code.

At Bletchley during the war Turing produced a machine which cracked the German cipher machine 'Enigma'. In 1969 (after his death) a Turing paper was published entitled 'Intelligent Machinery' pointing out that human faculties could be reproduced by a machine, drawing an analogy between computer circuitry and human nerves in a sense that they can both transmit information and store it. This in theory made the machine capable of problem solving.

Turing suggested that an attempt should be made to build a 'brain' capable of applying itself to playing chess, learning and translating languages, encoding and decoding and doing mathematics. He made a wide ranging comparison between the computer and the brain of the new-born child. The latter was unorganised until routines, general knowledge and heuristics (a technical term meaning rules of thumb) were fed into it. He saw no real difference between this and what his thinking machine would need to learn.

Turing proposed a test (31) to probe the idea of machine intelligence. The test became known as the Turing Test for AI and requires two people and one machine, one person is the interrogator and the other person shares a room with the computer. The interrogator communicates via a keyboard and VDU with either the other person or with the computer system. The problem is that the interrogator does not know which is which or whether he is addressing the person (A) or the computer (B). The role of the interrogator is to find out which is the person, A or B. It is the computer's role to fool the interrogator, if it succeeds then it can properly be called intelligent as it is showing that it can think.

The first conference on artificial intelligence was held at Dartmouth, New Hampshire (1956) (32, 33) and included four pioneers of AI, John McCarthy (who invented LISP), Marvin Minsky, Nathaniel Rochester (an IBM researcher) and Claude Shannon (a mathematician interested in information theory). These four had previously put up a research proposal for a 'study of artificial intelligence, to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it.' The conference was seen as a turning point in the history of AI.

One major debate concerned the title of the new 'science'. McCarthy argued strongly for the title of 'artificial intelligence' because he wanted to distinguish it from simple automata theory, concerned with the mechanical principles for constructing electromechanical systems. He wanted to construct machines which could play games, solve problems, learn and talk, ie. artificial intelligent machines (32, 33).

The background to AI provided an understanding of the definition of artificial intelligence and how this might relate to the design of the Colour Concept Generator. It was clear that the computer system was to perform a humanistic activity so required the sophistication to be a useful tool as opposed to just a novel idea. The system had to behave in an intelligent way in order to interpret an aesthetic description and translate it into a suitable colour scheme. The creative aspects of designing a colour scheme had to be promoted so the system had to enhance human creativity in an intelligent way.

2.8.2 Examples of AI Systems

A famous seminal chess program was that of Bernstein, a mathematician, in the late 1950s (34). Bernstein managed to build into his program two aspects which became crucial in AI, knowledge and heuristics. The knowledge was in the form of information from chess books, the heuristics were derived from experience and intuition.

The number of possible states in a game of chess is 10^{120}, which means that the world would finish before a game finishes. If every possible move was considered (even calculating a billion billion moves per second) it is clear why some means of cutting down the number of possibilities to be considered had to be devised. Bernstein built in a means of economising on the search strategies by using heuristics.

In 1957 Newell, Shaw, and Simon constructed a famous program known as the General Problem Solver (GPS) (31). The idea was that a really intelligent machine had to be able to turn its hand to a whole range of problems. Their idea was that when human beings are faced with a problem they use strategies. GPS was an attempt to identify and use those strategies.

An example of one famous outcome of this study was 'means-ends analysis'. Consider the present state of affairs (call it A), the start state. Then look at the desired state of affairs (call it B) a goal state. Then ask what step could be taken to reduce the difference between the two. Take that step then do the process all over again until eventually A and B are identical. It sounded simple but proved to be a very powerful means of solving multi-stage problems.

GPS used a directed search to build a network of states until the goal could be reached. In other less constrained areas it became clear that far more world knowledge and specialised knowledge was needed than could be handled. It was the first program designed as a detailed simulation of human behaviour as opposed to a program which could achieve the same ends in its own machine-like way. It provided AI with a set of tools which have been modified and incorporated in many later programs.

Colby's 'Simulation of a Neurotic Process' program in 1962 (34) was the first major research into human simulation with a medical aim. Colby wanted to design a program that was a working model of the kind of free association that takes place in psychotherapy. The program was meant to suffer from some classic neurotic symptoms or fantasies. If he could get the machine to output typical patient responses to therapist's questions, he could then work backwards to the sort of networks of association that accounted for them. More importantly perhaps, as a practising psychoanalyst, he could predict and test which input from the therapist was beneficial and which was counter-productive. This sort of 'what-if' procedure in model building is typical of a great deal of both AI and expert systems.

Weizenbaum's ELIZA (1966) (31) was interested in simulating conversational ability as the title of his paper shows, 'ELIZA, a computer program for the study of natural language communication between man and machine'. It could carry on an apparently ordinary conversation and this made it something of a novel idea. ELIZA was not really very intelligent at all, it did one of two things, rather like a therapist. It picked up a statement made by the human and played around with it, if it got stuck then it had a built in number of conversational ploys which were meant to generate a response and provide it with new statements to play around with. ELIZA could not say anything interesting on its own part, it could only act the therapist and was wholly dependent upon the human. It could do pattern matching but always needed the pattern provided first. In one sense ELIZA could pass the Turing test (for a while), but finally would not be very intelligent at all.

MYCIN (1976) (34) is an expert system developed at Stanford University between the schools of medicine and computer science including the Buchanan and Shortliffe research teams. MYCIN provides consultative advice on diagnosis and treatment of infectious diseases of the blood such as meningitis. A hospital patient recovering from an operation can develop an infection as the body is in a weakened state. The operating physician might not be an expert on the type of infection that develops. Also the type of drug therapy prescribed to stabilise the patient might not be compatible with the new infection. A further complication is that decisions have to be made on the basis of incomplete information. It can take up to 48 hours to identify an infection and the drug therapy needed for the post-operative condition might not wait that long.

MYCIN takes the initiative in the consultation as it interviews the physician, collects information and provides a diagnosis with recommendations. In arriving at its conclusions it makes extensive use of ‘confidence factors’ as it knows it is dealing with categories that are not exact. Confidence factors are processed automatically as the system runs, enabling it to state how confident it is about its conclusions. If a conclusion arrived at is unsatisfactory, the expert can inspect the process by which the conclusion was arrived at and tune it so that it does not occur again. MYCIN can state exactly how it arrived at a conclusion, effectively assisting with its own debugging.

MYCIN is not a static system as it can acquire and apply new knowledge via a related program called TEIRESIAS (34). It drives the session with questions aimed at backward chaining to the point where an error occurred. Once the problem is isolated, TEIRESIAS establishes what changes must be made to avoid the problem from occurring again. It is a knowledge-based program with rules for reasoning about a MYCIN knowledge base. The program systematically searches back along the trace of the reasoning to find the source of the problem. Combined with TEIRESIAS, MYCIN can learn so it is flexible, adaptable and dynamic.

From the examples of AI systems it was realised that the Colour Concept Generator had to acquire knowledge as it learnt new colour semiotic relations because the relations were not a static collection of data. The system had to operate as an AI tool as the generation of colour concepts requires the interpretation of a request that is translated into colours with reference to colour semiotic knowledge. Although the knowledge might be highly developed, without some intelligent means of manipulating it the computer system would just be a gallery of images.

2.8.3 Computer Models of Colour Semiotics

Shoji Tominaga (35) developed a colour naming method for computer colour vision based upon a 3D colour space of hue, value and chroma. The method consisted of four levels. Level one was a basic classification dividing the colour solid into red, yellow, green, blue, white, black, grey, pink, orange, brown, purple, olive, yellow-green, blue-green, violet and red-purple. Level two was a gross classification that expanded the basics into more accurate descriptions such as beige, lilac, lavender and sky. Level three was a medium classification and included modifiers applied to the tone of value and chroma of levels one and two. The modifiers were 'pale, light, bright, vivid, deep, dark, dull, light-greyish and greyish'. Level four was the minute classification and increased the tone modifiers to include 'soft, strong and dark-greyish' as well as 'ish' suffixes.

Shiojenn Tseng, Jyh-Ping Hsu and Tsao-Hung Wei (36) researched the quantification of qualitative relations of colour through fuzzy and statistical operations. Using descriptive adjectives such as 'warm, cool, soft, hard, light and heavy' they formed associations with red, orange, yellow, green, blue and purple. They argued that human impression was based upon subjectivity so probability theory was used to define an objective uncertainty due to a lack of appropriately defined reasoning rules in conventional analysis. They defined a systematic approach for quantifying qualitative statements for colours based upon statistical analysis.

The aim was to determine significant colours suitable for consumer products by following four steps. Step one arranged the colours in decreasing order according to the frequency of selection. Step two formed a ratio of the frequency of selection of two consecutive colours staring from the first colour. Step three assigned the ratio to the former colour of each colour pair. Step four considered significant the colours with a smaller than or equal to ratio of 1.5. Regarding the colour of electrical appliances blue was calculated as 'cool, hard and heavy' for an air conditioner, red and yellow were 'warm' for an electromagnetic cooker and blue was 'hard' for a personal computer.

The fuzzy operations were defined in terms of the air conditioner (AC) as follows:
F[x] is a function of a fuzzy set of colours pertaining to a certain characteristic of a colour.
$F(x \cup y)$ is a function of x union y and $F(x \cap y)$ is the function of x intersection y, where x and y are fuzzy colour characteristic sets.
$F(x \cup y) = \max(F[x], F[y])$, to determine the maximum colour wavelength in the range.
$F(x \cap y) = \min(F[x], F[y])$, to determine the minimum colour wavelength in the range.
Given the following fuzzy sets: B = blue, C = cool, Hd = hard and Hy = heavy.
The range of colours for the air conditioner could be determined as follows:
$AC = B \cup C \cup Hd \cup Hy$ gives the maximum colour wavelength in the range and $AC = B \cap C \cap Hd \cap Hy$ gives the minimum colour wavelength in the range. A suitable colour for the air conditioner would have a colour wavelength within this colour range.

Existing computer models for colour semiotic relations were found to see at what stage of development and in what situations current systems were aiming towards. The most complicated model performed qualitative statistical analysis of quantitative interpretations using fuzzy logic to determine the most viable single colour for a single product. The descriptions used were reduced to single words that were only concerned with visual impression and not aesthetic images or styles.

Although the model allowed for combined descriptors, it did not cater for colour combinations where the perceived impression might alter due to contrast effects. The model did demonstrate a possible method of how to find an optimum colour using statistical analysis. The Colour Concept Generator had to be able to deal with many aesthetic descriptions to cater for the many different images and styles that a product could portray. It was envisaged that a software mechanism to do this would require some means of weighting aesthetic descriptions to give an emphasis to aspects of the learnt colour semiotic relations.

2.9 Summary

Physically colours are waves of light that differ in frequency and energy through the spectrum from red (low frequency and energy) to violet (high frequency and energy). The semiotics of colour do not necessarily correspond to these physical attributes as culturally red is typically associated with fast and high energy, eg. sports cars in 'fast red'. The highest energy source, the sun, is perceived as yellow which is only a medium energy colour, so determining what colour *is* does not necessarily equate to what colour has become termed to *mean*.

Colour is a visual perception that exists in the mind and is a result of processing light energy via the eye and brain. Variations of contrast in colour combinations influence the perceived colour image. The colours (hues), lightness of colours, complementary colours, the saturation or intensity of the colours and proportions influence the visual image. Simultaneous contrast influences the perception of adjacent colours and successive contrast affects the impression of colour images viewed over a period of time. These effects contributed to the design of the form of a colour concept in the computer system and the visual interface as well as the way in which colour semiotic relations were learnt.

The psychological interpretations of colour as a diagnostic tool indicated the prevalence of colour imagery in the subconscious mind. Subliminal messages could be decoded from the colours seen in dreams and personality characteristics established through a colour selection process. This was extended to the point of understanding stages of mental change and psychological development. The information showed that colour can have influences at a subliminal level to illustrate that colour semiotics could be a conscious or subconscious activity.

The use of colour symbolism for developing colour semiotic relations illustrated a possible method of being objective about a subjective perception. Colour interpretation by reference to other objects (eg. red for blood is 'danger') could provide a justification for reasoning colour semiotic relations. Symbolism showed how object representation can be used as a form of communication of a concept. A potential problem was that not all cultures were in agreement with interpretation of colour symbols. Reference to other objects might require that those objects had been seen by whoever was being shown the colour symbol. In this sense having knowledge of a 'code' of colour symbolism might give preconceptions or misunderstandings. The 'code' might be acquired or learnt culturally at either a conscious or subconscious level.

The literature on semiotics explained the essence of this research project. The process of categorising colour images, or absolutely anything for that matter was shown to be that of relating objects through pattern matching. In this sense an aesthetic descriptor (object) or description (pattern) could be linked to a colour (object) or a colour concept (pattern). Connecting two different media in this way, in order to eventually translate one into the other might pose problems simply by the fact that the two media are different and might not translate directly.

The types of characteristics of a colour image that can be described were defined as impression, expression and symbolism. The visual look of a colour (eg. pastel), the emotion it incurs or coveys (eg. romantic) and the symbolic association (eg. red for blood is 'danger') could all be commented upon. The semiotics problem becomes more complex when interpretations by different cultures differ so the contextual issues have to be considered. How this information is structured in a colour lexicon had to be examined regarding the organisation of the colour semiotic information in the Colour Concept Generator.

The theoretical models of colour semiotics illustrated possible structures for semiotic relations proposed by the colour theorists. The examples showed possible methods of how to structure the relations between two different media from which a translation between the two could be performed. The

Colour Concept Generator required some form of colour semiotic model as the information had to be arranged in such a way that it could be contained and manipulated in a computer system.

The use of colour in commercial design showed where colour might be considered in the design process and how colour semiotic relations were established by doing consumer field research. There was a need for descriptions and specifications of a colour scheme to be stipulated earlier in the design process at the initial conception of an idea for a product. This influenced where the Colour Concept Generator could be suitably placed in the design process to indicate possible inputs and outputs for the computer system.

The forming of colour semiotic relations by a commercial organisation indicated the desire by the company to structure the information in a form that it could be used in commercial design. Field test research showed the possibility of consulting consumers in the decisions for selecting colour schemes and how the research information might be presented. For the Colour Concept Generator this influenced the consideration of how inputs could be arranged for learning colour semiotic relations. The problem of designing a product for a world market where different cultural interpretations might contradict for a selected colour scheme indicated the need for colour semiotic learning to be organised so that cultural labels could be stated.

The background to AI and the example AI systems provided an insight into the definition of artificial intelligence and how this might relate to the design of the Colour Concept Generator. It was clear that the computer system was to perform a humanistic activity so required the sophistication to be a useful tool as opposed to just a novel idea. It was realised that the Colour Concept Generator had to develop in knowledge as it learnt new colour semiotic relations because the relations were not a fixed set of facts. Also that although the knowledge might be incredibly sophisticated without some intelligent means of interpreting it the computer system would just be a library of data. The system required a degree of artificial intelligence to be able to learn colour semiotic relations as well as reason a proposal for a product colour scheme.

Existing computer models for colour semiotic relations were found to see at what level of sophistication and in what areas current systems were directed towards. The most sophisticated model performed qualitative statistical analysis of quantitative interpretations using fuzzy logic to select the most suitable single colour for a product. The descriptions used were limited to single words that were mainly to do with visual impression and not aesthetic images or styles.

Although the method accounted for combined descriptors, it did not account for combinations of colours where the perceived impression might change due to contrast effects. The method did illustrate a possible way of how to find an optimum colour using statistical analysis. The Colour Concept Generator had to be able to manipulate many aesthetic descriptions to allow for the many different images and styles that a product could portray. It was realised that a software mechanism to do this required some means of weighting aesthetic descriptions to give a balanced interpretation of the learnt colour semiotic relations.

This chapter aimed to review the relevant literature and put the research into context by illustrating influences on the research. The influences contributed towards indicating and solving problems that might be needed to be overcome in order to achieve the design of the Colour Concept Generator. The following chapter is a review of the problem in light of the information given in the literature review. The problem is analysed more specifically, in more depth and detail and the philosophical nature of the research is made more clear. The problem analysis leads to a detail specification of the system upon which the design of the computer system is based.

3 Problem Review

3.1 Introduction

The previous chapter illustrated the influential fields that contributed to the research. It was seen that the semiotics of colour do not necessarily correspond to the physical attributes and that colour is a visual perception that exists in the mind and is a result of processing light energy via the eye and brain. Psychological interpretations indicated the prevalence of colour imagery in the subconscious mind and colour symbolism illustrated a possible method of being objective about a subjective perception. Literature on semiotics explained the essence of the problem of this research project and the translation between semiotic media.

Characteristics of a colour image that could be described were defined as impression, expression and symbolism and theoretical models of colour semiotics illustrated possible structures for semiotic relations. The use of colour in commercial design showed where colour might be considered in the design process and how colour semiotic relations were established by doing consumer field research. The background to AI and the example AI systems provided an insight into the definition of artificial intelligence and how this might relate to the design of the Colour Concept Generator. The most sophisticated computer model that could be found used descriptors that were limited to single words mainly to do with visual impression and not aesthetic images or styles.

This chapter is a review of the whole problem of this research. It begins with an analysis of the problem to understand the essence of the research and examine the detail problems that had to be investigated. Aesthetic aspects of products were considered to establish how colour related to a product as a whole. The product design process was examined to determine a suitable position where the Colour Concept Generator could be used. Translating between an aesthetic description and a colour concept was analysed along with the criteria for forming colour semiotic relations to see how the system could learn colour semiotic relations from which colour concepts could then be produced.

The form of a colour concept and an aesthetic description was looked at to decide how the information could be represented. A collection of aesthetic descriptions from product designers was acquired to examine the types of words used to describe the image and style of a product. The problem of colour semiotics in a cultural context was considered to determine how this could be accounted for in the computer system. Machine learning of colour semiotics was analysed to see how a computer could acquire and learn colour semiotic knowledge. Generating colour concepts was looked at to see how the learnt knowledge could be manipulated in a computer system interactively with a product designer to design a colour scheme.

From the problem analysis a detailed specification of the system was produced outlining the needs and requirements and the intended design criteria for the system. The specification provides details of the following: Boundary of the system, ethos for the role of the system, colour concept display, interface menu and functions, colour semiotics learning input, colour semiotics learning process, colour concept generating input, colour concept generating process and colour concept generating methodology.

3.2 Problem Analysis

The problem mainly involved the translation between semiotic media, ie. between the description of an intended aesthetic image or style for a product and a colour scheme. A computer system had to be produced that could learn these colour semiotic associations from which colour concepts could be generated. The issue of how to do this objectively had to be considered and an objective method that could be used with the computer system had to be designed, although a subjective personal style could also be adopted. This part is an analysis of the problem and the detailed aspects that had to be considered in the design of the Colour Concept Generator. Reference is made to literature reviewed in the previous chapter as possible solutions and to illustrate points.

3.2.1 Product Colour Aesthetics

Products are designed based upon technical, aesthetic and ergonomic criteria relating to visual image and functional requirements and are indicative of a compromise of ideals relevant to a specific problem (37). Aesthetic design decisions for a product are based upon integrated influential criteria. The cultural context of the product dictates issues relating to visual image and aesthetic style (38). Visual, ergonomics and the semantics of product operation enhance human interaction and perception of product function (25). These factors contribute towards and are influenced by aesthetic aspects pertaining to colour, texture, shape and form in a visual and tactile composition (25).

Products are 3D objects that exude an image or style representative of an aesthetic character. The colour scheme adopted for a product portrays the exhibitions of colour defined by Itten (12) relating to impressive, expressive and symbolic qualities. These vary in significance according to the type of product which could be anything from a cosmetics case to a power tool or a new invention. Products often include more than one component so a colour concept for a product could include more than one colour. Colour concepts for lipsticks could be generated by the same processing mechanism as colour concepts for lawn mowers both of which could have more than one colour that vary in proportions (25).

The problem regarding product colour aesthetics was how to cater for the impressive, expressive and symbolic aspects of colour semiotics in a way that is applicable to a variety of different products. A software mechanism to do this either had to be transferable within the colour semiotic, aesthetic and product areas or be flexible enough to accommodate all colour semiotics for all aesthetic styles and images for all products.

3.2.2 Integration of the System into the Product Design Process

The product design process or methodology originates in the form of a design brief (an outline of the general requirements of a product) and generally takes a linear route. The design brief is analysed and research undertaken in order to produce a problem statement and establish a product design specification (PDS) for the required product (37). Research relating to the product aesthetic image and style might include descriptions such as the fashion adjectives used in the Nippon Colour Image Scale (30), eg. 'romantic, chic'. The PDS is based upon the analysis and research and is a definitive statement of the aesthetic, ergonomic and functional requirements of the product (37).

Concepts are then generated in relation to aesthetic, ergonomic and functional requirements either on an individual macro basis or for the whole product as an integrated theme. Concepts can be presented as 2D sketches or 3D models in a variety of media and are ideas for solutions to problems stipulated in the problem statement and PDS. The process is very fluid as no firm decision is made at this point. The concepts are developed in an integrated manner into the proposed solution for the product from which a prototype model is produced. The development process involves fine-tuning detailed features relating to the technical, aesthetic and ergonomic aspects of the product to allow for mass production (37).

A software mechanism to generate colour concepts for a product could be included anywhere in the design process. Pentagram (25) decided the colour (red) for the Wilkinson Sword Protractor Razor prior to the commencement of the design. Dale Russell (25) claimed that there would be a greater degree of calculated freedom given an indication of the required colour during the initial stages of the design process. The Scantel (25) technique considered the potential future performance of a colour scheme separate from the product.

The problem regarding the integration of the system into the product design process was to choose a suitable point where the system could be used and design it for that point. The initial idea and aim for the system was to provide a turn around of an aesthetic description of an image or style into abstract coloured images as proposed ideas for visual colour schemes for a product. The Colour Concept Generator could be used anywhere after the design brief as it has a specific task of converting an aesthetic description into a colour concept.

3.2.3 Translating Between an Aesthetic Description and a Colour Concept

A major problem in the design of the system was in the translation between the semiotic nature of visual coloured images and the semantic meaning of an aesthetic description. This was described by Miller and Johnson-Laird (22) as a fundamental process of relating objects whereby colour labels were learnt through a process of repeated pattern matching and items analysed into semantic components. The problem in this research required the analysis of colour images using descriptions that could subsequently be used to synthesise colour concepts through a generating process. So if white is indicative of 'purity' then 'purity' would generate white. However the translation process was not so simple to model due to the complexity of semantics and semiotics.

The problem was introduced in the Aulus Gellius (18) account where Favorinus remarked that the eye was able to isolate more colours than words could name. This was defined by Eco (18) as a problem of identification and discrimination. Eco's semiotic relation between linguistic expression (reference) and cultural content (meaning) holds the same semiotic problem of translating between aesthetic descriptions and colour concepts. A software mechanism to do this could be an artificial version of Eco's signification system to interface the semiotic relation between the descriptions and colours. Colour concepts form the expression (reference) and an aesthetic description the content (meaning).

The colour semiotic translation process is not limited to one description for one colour, like turning a coloured card over to reveal a single meaning. William Horton (26) pointed out the unlimited nature of colour meanings and the need to accept multiple and diverse interpretations. There are no singular rules, a description could evoke a variety of colours and a colour scheme a variety of descriptions. Green is symbolic of 'lush, fresh vegetation' yet it is also the colour of 'decayed mould'. So 'lush or fresh or decayed or mould' would generate green and green would imply 'lush and fresh and decayed and mould'.

The translation process needed to account for variations in shades of colours pertaining to the same hue. The ISCC-NBS (22) colour lexicon was based upon a 3D colour model of hue, saturation and lightness. Specifications of colours were based upon a saturation and lightness description of a hue (dark orange, pale lilac). However it is not enough to say 'dark' orange or 'pale' lilac as there are more colours than categorised terms. The translation process had to deal with colours as colours and not their categorised terms so required a visual colour interface to view the colours.

The main problem in translating between an aesthetic description and a colour concept could be simplified to the connection of expressive media (eg. words, music, wavy hand movements, pulling faces) to colours. A required aesthetic theme could be described in a variety of media relating to any form of expression. The communication of the exact aesthetic description is limited to the sophistication and power of expression of the media available. Translation could be performed by establishing connecting links between expressive media and colours, so the translation could go from the expression to give the colours or from the colours to give the expression via the established connecting links.

3.2.4 Criteria for Forming Colour Semiotic Relations

The main problem in designing a criteria for forming colour semiotic relations was in the dilemma between subjectivity and objectivity. The Oxford English Dictionary (OED) defines objective as 'external to the mind, actually existing, real'. Physiologically colour is the result of visual processing, so internal to the mind and does not physically exist. However the OED also defines objective as 'exhibiting facts uncoloured (!) by feelings or opinions'. In this sense (omitting any debate of whether anything is real or reality exists solely in the mind) a criteria could be developed from which colour images could be analysed and colour concepts synthesised in an objective manner. A major problem of this research was how this could be performed most effectively by a product designer.

A solution might be to base the objectivity upon statistical analysis such as that in the fuzzy operations in the research of Shiojenn Tseng, Jyh-Ping Hsu and Tsao-Hung Wei (36). A broad and extensive accumulation of descriptions of colour images could be acquired from a variety of individuals from which the most frequent of responses might be determined as the most objective. However this could be basing objectivity upon a consensus of subjectivity and is probably a more suitable method for colour preference schemes such as Scantest (25).

The statistical accumulation process could be incredibly time consuming if a comprehensive collection of colour semiotic relations were to be found. Variations of hues by saturation and intensity as well as the variation in a colour character when placed with other colours had to be accounted for. White might be indicative of 'purity', but when placed with black and orange change the impression. Also the impression might alter according to the proportion of colours used, white placed with a small amount of pale orange has a different impression to pale orange placed with a small amount of white

The criteria had to acknowledge realistic time constraints, a method had to be developed whereby objectivity was more immediate and automatic and could be performed by the product designer. Either that or a designer's subjective personal style was allowed. A possible solution could be to describe a colour image by a denoting noun of origin (eg. surgeon) and any associated descriptions of a contextual nature (eg. medical, clinical, sterile). This could allow for creative colour expression based upon objective criteria as the colour semiotic association is justified by symbolic reference to an existing entity.

If the system was to reflect a designer's personal style and be purely subjective then more freedom for expression is allowed. The nature of colour semiotics is such that more abstract terms could be used to describe a colour image such as 'sherbet kisses', 'effervescing cyanide', 'rippling phantoms', 'acrobatic seashells', 'marshmallows on a desolate shore', 'wispy beard candyfloss'. The list is endless if metaphoric and poetic connotations could be used. The main problem in designing a criteria for forming colour semiotic relations was to come up with an objective methodology that still allowed for more subjective expression.

3.2.5 The Form of a Colour Concept

A colour concept could consist of single colours as well as combinations of different colours that include the consideration of variation of hue, saturation, lightness and proportion. The colours are conceptual in nature as they are abstract visual images and could be generated prior to the design of the product. A selection of colour concepts could be proposed or a single concept could be varied by slight alteration of the aesthetic description from which it was generated.

Itten (12) defined the proportional inclusion of each colour in a multiple colour combination as contrast of extension. This had to be considered as the visual impression and expression of a colour concept alters as the proportion of colour alters, blue with a little bit of pink expresses a different character to pink with a little bit of blue. The proportional inclusion of colours could be governed by weighting the aesthetic description with numerical quantities, it could be inherently present within the computer system or ideally both. The number of colours required could be specified prior to the generation of a colour concept or decided upon later so as not to restrict the potential for expression.

The form of the interface representing the colour concept needed to be abstract and neutral so as not to distort the impression by emphasis of shape or location of colour. Also any colours next to the colour concept would affect the impression of the image by simultaneous contrast. A simple geometric shape had to be adopted such as a square, triangle or circle and plain graphical representations such as bar graphs or pie charts.

3.2.6 The Form of an Aesthetic Image or Style Description

The aesthetic description forms an equivalent expression of a colour concept. It exhibits a semantic meaning in its own right from which an image is portrayed of an aesthetic concept. The aesthetic description is the communication of an aesthetic character embedded within the composition of a colour image. The form of the description is critical to the accuracy of communication and expressive power for the product designer. The aesthetic description connects the product and its colour scheme to the required aesthetic image and style. The problem lay in selecting a method of aesthetic description that would serve a purpose of describing the proposed colours in the context of designing products.

The description of an aesthetic image or style could take any form as semiotics is all about the meaning of signs and symbols whereby symbols are something that stand for something else. An aesthetic description could be an expression or gesture of a face or a wavy hand movement, it could be a piece of music or a collection of sounds. The description might make reference to other media such as books or films and characters. A style of dance or performance could be enacted in such a way as to convey a certain image as could the style in which something was said. The sound of weather or atmospheric conditions might be descriptive of an aesthetic image as might the sounds from any place, a cave, quarry, monastery, ice rink, speedway, underwater and so on. The main problem was how this could be achieved within the context of designing products.

The most specific means of describing a colour are the scientific numerical criteria relating to colour wavelength and intensity. However these do not adequately describe a product image or style so quantification is not necessarily prescriptive of qualification. The same applies to magnitude of light energy and wave frequency. Violet is a high frequency (fast), high energy colour and red is a low frequency (slow), low energy colour yet there is no parallel between these characteristics and the contextual meaning. The term 'fast red' as the colour of sports cars contradicts the scientific characteristics and no high energy products could be found in violet.

The limited expression of descriptions relating to the visual impression of colour was exemplified in the research of Shiojenn Tseng, Jyh-Ping Hsu and Tsao-Hung Wei (36) into the qualitative relations of colours. The descriptions were limited to 'warm, cool, soft, hard, light and heavy'. Although relevant to an aesthetic specification the descriptions are too general. A 'warm, light, soft' cosmetics case might imply a multitude if hues in varied proportions. An aesthetic description might require a specific and definite meaning but the potential for expression must be diverse. The Colour Concept Generator required the potential to develop an extensive vocabulary for a more direct means of communication of a desired image. This allows for the converging towards a more refined colour concept from a diverse means of expression.

The Nippon Image Scale (30) fashion adjectives provided an example of the type of descriptions transferable into the field of product design. The adjectives posses the potential for a more specific and expressive description of a product image or style. However the type of description need not be limited to adjectives alone. If the colour combinations were derived from carefully selected natural, commercial or contemporary coloured images then the colours could be described by the denoting noun of the image source (as mentioned previously in the surgeon example). Prior knowledge of colour image sources might be needed to make any sense or significance of them, however if the designer assigned these descriptions then the semantic relevance would be self-contained. On this basis the descriptions could hold metaphoric and poetic implications and include verbs, nouns and adjectives that allowed for personal style and interpretation.

3.2.7 Aesthetic Descriptions from Product Designers

Twenty final year Product Design students (all with at least one year industrial experience) were asked to provide an aesthetic description of the required image and style for the products they were designing. The products were in the initial stages of the design process during the formulation of the PDS. The designers were told to state the product they were designing and list no more than twenty single words that described the desired aesthetic concept. The aim was to find out what type of words might be used in an aesthetic description for a product. The aesthetic descriptions are listed in appendix A.

The types of words used were nouns, adjectives and verbs. The words described the aesthetics in terms of the product function (eg. communication) as well as the image and style (eg. aquatic). Some of the words were references to other objects that were symbolic of the required aesthetics of the product (eg. money, suit). Attitudes were expressed (eg. friendly, sympathetic) and finishes (eg. hard, soft). There were words relating to the visual impression (eg. clean, clear) and emotional feelings (eg. calm, safe).

The environment that the product would be used in (eg. camping, outdoors) and the product field (eg. musical, sporty) were expressed. There were descriptors that related to the form of the product (eg. streamlined, compact) and the texture (eg. smooth, tactile). Descriptors relating to the integrity of the product (eg. reliable, durable) and that of the consumer (eg. professional, sophisticated) were stated as well as the cost (eg. expensive).

The aesthetic descriptions showed initially that product designers understand and respond to the request for an aesthetic description. They expressed the required image and style using a variety of different types of words and the types of words used were from many different perspectives. When reading the aesthetic description it was possible to realise the type of image and style that the product designers wanted for their products even though they were expressed using single words only.

3.2.8 Colour Semiotics in a Cultural Context

The cultural context in which a product would be designed posed a problem in describing colour semiotic relations. The problem is focused in the debate of whether colour semiotics is universal or relative. Berlin and Kay (20) proposed that colour naming is universal and that interpretation is an inherent process common to all cultures. This might be the case with colour categorisation however cultural colour semiotics can be variable as Horton (26) illustrated in the diverse interpretations of colour by different cultures. Blue was authority in the west and villainy in Japan, red was anger in Japan and joy in China. The difference in the allocation of colour to the elements by the Hindus, Chinese, Greeks and Jews (6) indicated a symbolic association of colour that was not universal. Water was white to the Greeks and Hindus, purple to the Jews and green to Leonardo.

The problem extended in considering the comparative conceptual interpretation of descriptions in different cultures and if identical or different descriptions imply identical or different meanings. The translation of terms, the specific relation to abstract conceptual meaning and the exact point of reference illustrated the problem of colour semiotics being based upon interpretations of interpretations of interpretations and so on.

Eco (18) illustrated the point in detail in the description of the signification system. This indicated colour semiotics as being relative in a way that considered the influence of cultural ideology upon human reasoning. The system indicated that content meaning is an abstract concept contained within the human mind and that communication with the outside world relies upon an established convention of descriptions which in turn is a cultural property. This related back to the Aulus Gellius (18) account as without the available expressions or vocabulary the description cannot serve justice no matter how sophisticated the content.

Colour semiotics appeared to be relative and dependent upon cultural context. The methodology adopted by the Colour Concept Generator had to account for different cultural contexts. As a starting point it appeared suitable to base colour semiotic relations within the context of a western society to draw upon aspects from the culture relating to colour aesthetic image and style.

3.2.9 Machine Learning of Colour Semiotic Relations

Colour semiotics relates to the significant meaning and interpretation of abstract colour images. This involves making sense of single colours and combinations of different colours varying in hue, saturation, lightness and proportion. Aesthetic descriptions provide a communication media for interpretations of colour images. The association of colour concepts to concepts of image and style regarding the design of products exemplifies the semiotic connection of abstract themes. The conceptual semantics of image and style characteristics, the semantic interpretations of aesthetic descriptions and the conceptual semiotics of colour combinations relate in a linear process. The problem of machine learning of colour semiotic relations was how to translate between themes of an abstract conceptual nature through some reasoning mechanism.

The Colour Concept Generator required a colour semiotics base of colours and descriptions. The knowledge could be prescribed and fixed in a static way or it could be an ongoing dynamic process capable of being updated as the system was used. A methodology for learning colour semiotic relations was required from which colour concepts could later be generated.

Itten (12) defined the characteristics prevalent in all colour concepts as relating to impression, expression and symbolism, these were also paralleled by Cicero (19) and Grunewald (19) in different terms. The impression of a colour could be influenced by factors other than colour such as texture and reflection. The colours in a computer system could be generated on the screen so impression qualities were limited to what the interface could deal with. However descriptions of impression could relate to colour groups such as pastel or fluorescent.

The expression or mood qualities could be dependent upon the type and geometry of a product and the amount of time it was viewed for. Expression would be more applicable to the design of say interiors where the designer might want to create an ambience or atmosphere. In this case the user would be surrounded by the colour, on the inside, whereas with a product the user is usually on the outside of a 3D object that exudes an image and style. However descriptions of expression could be paralleled to descriptions of aesthetic image and style. A person could be in a 'romantic' mood and a lipstick container could have a 'romantic' style.

Colour symbolism corresponds to cultural context issues as it could be based upon interpretations of cultural images. Machine learning of colour semiotics through symbolism provides an interpretation of something, as something that stands for something else. Learning within the cultural context allows the system to be a reflection of the culture. Problems might occur if an interpretation of a cultural image was influenced by other aspects of the image. If this was based upon factors other than colour then a clear interpretation might not be portrayed. However as previously mentioned in the 'surgeon' example, denoting nouns and expressive expansion of aesthetic descriptions could suffice. The integrity of the computer system would be based upon the quality of image selection and description.

The way in which the semiotic relations could be structured and reasoned was a critical factor in maintaining the integrity of the system. Itten (12) proposed a kind of semantic tree structure that formed a hierarchical topology whereby mixed colours inherited characteristics from the component colours. However if descriptions were to correspond to the same structure then base descriptions would have to be decided from which all other descriptions would be formed from combined meaning. The impression of a single colour could change when placed with other colours, as could the combined meanings. Considering the number of possible colours and combinations Itten's method would be too rigid. The structure had to be more fluid to allow for multiple meanings of single and combined colours, varying proportions and the number of available colours and combinations.

Machine learning of colour semiotics need not be a prescriptive process to be rigidly fixed as this assumes that all colours and combinations have been accounted for. The computer system needed to learn in a developing and dynamic way. This implies that all colour semiotic knowledge is in a fluid state and able to be altered. Any colour concept generated by the computer system is based upon what is known by the system at that particular time. Without ongoing development new interpretations could not be adopted and the system might become defunct in terms of contemporary significance.

3.2.10 Generating Colour Concepts

The computer generation of a colour concept consists of the translation of an aesthetic description into a colour scheme via a reasoning mechanism that interacts with the currently known colour semiotics. The system needed to create a colour image based upon an ability to interpret the meaning of an aesthetic description depicting the image and style of a product. From this interpretation an intelligently reasoned colour combination could be produced depicting image and style characteristics from colour images that were previously learnt. The generation process is a mapping of abstract concepts related under a common theme, ie. the aesthetic description and the colours on the screen illustrate a picture of an aesthetic composition.

The product designer might enter single descriptions one at a time or an entire description, the intermediate stages could control the variety of colour concepts generated. A problem that might occur is if the system were to generate too many colours for the product or just a single colour. The reason might be a incompatibility between the learnt colour semiotic relations and the aesthetic description. The designer needs an insight into the system knowledge in order to see the current state of the system.

The problem was overcome by providing a more interactive process, as well as generating colour concepts, the system needed to provide a common description of a colour concept. This could give an insight into other possible semiotic descriptions to achieve the same thing. A menu could display currently known semiotic relations from which the designer could select the most relevant to the required aesthetic description.

In describing a colour concept, the system indicates different semantic meanings for the same description. To say 'cosmetic' could imply association with face paint yet could also imply 'synthetic' or 'artificial'. If the designer entered 'cosmetic' to reveal a colour combination from which a set of related common descriptions was then produced, the system effectively prompts for a more specific meaning. The description could then be expanded from the set of common descriptions to provide a more refined and definite request. The process could then be repeated until a colour concept was accepted.

The interactive process could start with colours as opposed to an aesthetic description. A description could produce a combination of colours, some of which might be removed or colours added. The colours could then be described from which the required aesthetic description could be altered. The description could then generate a colour concept of which the colours might be altered, then describe, alter description, generate, alter colours, describe, and so on. This method gives greater control to the product designer through a more interactive approach and places the computer system more in the realms of a design tool as opposed to a dictating media.

It was envisaged that the generation of colour concepts would be consistent assuming that no new colour semiotic relations had been learnt, ie. the same aesthetic description would generate the same colours and vice-versa. However the colour proportions might vary depending upon any weighting of the aesthetic description. This allows for a point of emphasis in the description to be a more specific definition. The description could then be evaluated and used in determining the proportion of the colours in the colour concept. This gives the system more sensitivity and integrity.

3.3 Detail Specification of the System

3.3.1 Boundary of the System

The computer system is a design tool for product designers that provides a translation of colour semiotics from an aesthetic description to a colour concept. The conceptual and abstract nature of the colours it produces indicates that the system can be used at any point after the design brief. However, it is typically placed at the concepts generation stage of the design process as this is the point at which other conceptual design is undertaken. In this case the system is a form of ideas stimulus for a product designer and is used after product research and the PDS have been carried-out and precedes the development of the product.

The input is taken from the design brief, analysis and research, or an aesthetic PDS describing the required aesthetic image and style of the product. This can be reduced to a set of descriptors quantified according to the desired weighting of emphasis. The output produces a selection of colour concepts varying in number of colours, actual colours and proportional area of colour. The colour concepts are expressive of the input descriptors and weightings and are passed on to the development stage of the design process. The process between the input and output includes the learning, generation and description of colour concepts

3.3.2 Ethos for the Role of the System

Generating colour concepts as a design methodology is a concurrent process running alongside the generation of product concepts relating to aesthetic form. The idea is to promote concept generation in an integrated and combined manner as opposed to separately considering aspects of conceptual aesthetic design. The computer system is part of an existing computer aided design system to integrate with other conceptual design carried-out on the same system so providing a consolidated method.

The process of learning colour semiotics and generating colour concepts aims to be as interactive as possible. The computer system is to play more of an advisory and suggestive role rather than being prescriptive and dictatorial, stimulating creativity rather than stifling it. The colour concepts are proposals for colour schemes. The aim is to provide an ideas stimulus to inspire the product designer to come up with ideas for colour schemes for products. It is envisaged that the computer system is a design tool as opposed to a design rule.

The product designer can be involved in the development of machine learning of colour semiotics. This is an ongoing process to allow for a more fluid and dynamic approach to respond to new variations in meaning as new colour semiotic relations are learnt. The Colour Concept Generator is an ongoing, developing expert building upon a colour semiotics base through interaction with the product designer.

3.3.3 Colour Concept Display

The colour concept display takes the form of a circular pie chart to minimise any distortion of impression of the image by shape or proximity. A pie chart is considered to be the most neutral representation with all colours taking a more balanced standpoint. The colours are viewed in totality within one location to make the concept more universal.

A pie chart indicates proportional weighting of each colour by respective allocation of angle within the 360 degrees of a circle. Colours are also displayed by computer colour number and proportional weighting on another part of the screen for more accurate detail and reference. Examples of colour concept pie charts along with the associated descriptors (from the evaluation exercise) are contained in appendix F.

The colour concept pie chart is surrounded by a colour palette of available system colours (shown in appendix C) from which colours can be selected to add to the concept. The colour palette consists of all the available computer colours with the fully saturated hues shown in the palette. The remaining colours (varying in saturation and alternating between pastel and bold) can be accessed via the user interface. Any colours selected are initially viewed in the centre of the concept prior to being added to the pie chart in order to evaluate a potential colour in a more prominent place next to the other colours.

3.3.4 Interface Menu and Functions

The interface menu consists of functions for creating, manipulating and displaying colour concepts and descriptors. The colour functions include adding colours and weightings to the concept, removing colours from the concept, proportioning colours in the concept and displaying current concept colours. The descriptor functions include adding descriptors to the description, removing descriptors from the description, displaying the descriptor menu of currently known colour semiotics and displaying the current descriptors. The interface is shown in appendix C.

The activation of processes to learn colour semiotics, generate colour concepts and describe colour concepts is performed via the interface menu. This enables the possibility of new concepts to be learnt as colours are generated on the screen or described. Part of the learning process allows for inspired learning as the system is running. Activation of the learn process involves adding the current colour and descriptor relations to the system colour semiotic knowledge. The generate process produces a colour concept on the screen from the current descriptors and the describe process produces a list of common descriptors associated with the current colour concept on the screen.

The interface is mouse driven except for the input of descriptors and weightings which are input via the keyboard. Colours are selected from the colour palette by clicking the mouse on a colour and appear for viewing in the centre. Variations in saturation and pastel / bold colours can be found by clicking the mouse in the centre or to the outside of the viewed colour. When selected (by clicking on the colour pie chart) the colour appears in the colour pie chart and is given a numerical proportion if another colour is already there. The proportion given is that which the new colour has in the colour pie chart with all other colours re-proportioned according to their previous proportions. When a colour concept is complete the colour palette is removed by clicking outside the palette circle. All other functions in the interface menu can be activated by clicking the mouse on the menu items.

3.3.5 Colour Semiotics Learning Input

The colour learning input consists of a set of colours and weightings and a set of descriptors and weightings. The colours can be an abstract representation of an existing colour image of either a natural, cultural, commercial or personal nature. The image can be described by any denoting nouns and an extended range of expressive descriptors relating to product image and style attributes or by descriptors relating to metaphoric simulation. The descriptors can be weighted according to their appropriateness and significance in representing the whole picture. The sets are:

colours (colour, weighting, colour, weighting ....)
descriptors (descriptor, weighting, descriptor, weighting ....)

The proportional weighting of each colour or descriptor in the set is calculated by dividing the weighting by the sum of the other weightings in the set:

proportional weighting = weighting / sum of weightings in set

3.3.6 Colour Semiotics Learning Process

The colour learning process involves development and modification of the colour semiotics knowledge in the system. This is a dynamic ongoing process based upon assigning descriptors and weightings to individual colours. The colour descriptor weightings are a working average of all that has been learnt by the system. The individual colour set is as follows:

[(descriptor, average), (descriptor, average) ....   colour]

The working average consists of a sum of all learnt weightings and a count of the number of times the descriptor has been assigned to the colour in the learning process. This does not imply one to one allocation as many descriptors can be assigned to many colours. The whole individual colour set is as follows:

[(des, sum, count), (des, sum, count) ....   colour]

The working average is calculated by dividing the sum by the count.

working average = sum / count

When a colour and descriptor relation is learnt, the descriptor count is incremented by one and the descriptor sum is assigned a new weighting based upon the descriptor and colour proportions within the semiotic relation. The new weighting is the old weighting added to the multiple of the descriptor and colour proportions.

new sum = old sum + (descriptor proportion X colour proportion)

This allows a dynamic learning process that can be continually updated as new colour semiotic relations are formed.

The whole process can be described as matching colour and descriptor combinations to assign relational weightings through matrix multiplication. The colour and descriptor sets form a 2D matrix of attributes and proportional weightings. The new colour descriptor weightings are the result of averaging the multiple of the colour and descriptor matrices.

3.3.7 Colour Concept Generating Input

The colour concept generating input takes an input of a set of descriptors and weightings. The descriptor set can be based upon the aesthetic specification of the required product and describes the image and style attributes. The descriptor weightings can be indicative of emphasis or bias of descriptor weighting. The descriptor set is as follows:

descriptors (descriptor, weighting, descriptor, weighting ....)

The proportional weighting of each descriptor in the set is calculated by dividing the weighting by the sum of the other weightings in the set.

proportional weighting = weighting / sum of weightings in set

3.3.8 Colour Concept Generating Process

The colour concept generating process involves producing a colour combination from a set of descriptors and weightings. The colours are determined by matching the descriptors with the learnt colour semiotic knowledge. The colour proportions in the colour concept are based upon the descriptor weightings and the colour descriptor working weightings assigned through colour semiotic learning. The colour weighting is determined by summing the multiple of the descriptor proportions and colour descriptor working weightings.

colour weighting = sum of (descriptor proportion X working average)

When all colour weightings have been allocated to the matched colours, the colour proportion for the colour concept pie chart are calculated by dividing the colour weighting by the sum of colour weightings in the colour concept set.

colour proportion = colour weighting / sum of weightings in set

3.3.9 Colour Concept Generating Methodology

The methodology for generating colour concepts is an iterative, interactive process. The method initially involves generating the colour concept from the aesthetic descriptors which can be subsequently modified with reference to the currently known colour semiotics. This requires a facility in the system to provide a description of a colour concept, ie. a set of common descriptors associated with the set of colours. The methodology is as follows:

1 Produce descriptors from aesthetic specification.
2 Quantify descriptors for emphasis weighting.
REPEAT
3 Generate colour concept.
4 Modify colour concept.
5 Describe colour concept.
6 Modify descriptors and weightings with reference to description.
UNTIL COLOUR CONCEPT ACCEPTED

The descriptors for the description are selected by matching the colours in the concept with the currently known colour semiotics to extract all of the common associated descriptors. The descriptors assist in establishing accurate semantic meaning of the aesthetic descriptors as by expanding the description the meaning is more specific.

3.3.10 User Process

The user initially develops colour semiotics knowledge in the system by allocating descriptors to colour pie charts. The colours could have been derived from existing natural, commercial or cultural images or could be compositions created by the user. The descriptors could be derived from symbolic association with the image or be a personal interpretation by the user. Any specific cultural or contextual relevance can be labelled in order to place the colour semiotic association in a particular realm or domain. The descriptors are weighted to allow for any particular emphasis within a description.

The semiotic associations can be single or combined colours and descriptors. These are learnt by the system and the new semiotic association is integrated with all previous knowledge. Any descriptors in a new colour semiotic association that are already in the system are connected with the new descriptors via associated colours and descriptors and colours are connected via common descriptors.

Having developed the colour semiotics knowledge in the system the user can then generate or describe colour concepts. To generate a colour concept the user can begin with single or combined descriptors selected from a descriptor menu. In the case of combined descriptors, each can be allocated a weighting to give point of emphasis. A colour concept can be generated of which colours can be added or removed. The user can then obtain a general description of all the colours to prompt for other descriptors that when added to the generating descriptors produce a more refined image. The generate / describe process is repeated until a colour concept is accepted. New colour semiotic associations can be learnt during the generate / describe process to allow for spontaneous learning as inspiration for new ideas might occur while using the system.

3.4 Summary

The problem relating to colour aesthetics of a product was how to provide for the impressive, expressive and symbolic interpretations of colours in a way that caters for many different products. A software mechanism to do this either had to be transferable within the colour semiotic, aesthetic and product areas or flexible enough to accommodate all colour semiotics for all aesthetic styles and images for all products. This problem was overcome by linking aesthetic descriptors to colours in a many to many relationship so any aesthetic descriptor could be a label for any colour. The descriptors could be weighted and the colours proportioned to allow for emphasis and differentiation. When combined together the colours were determined and proportioned based upon statistical refinement.

The integration of the system into the product design process was considered to determine an appropriate stage where the Colour Concept Generator could be used. The conceptual and abstract nature of the colours it produces indicates that the system could be used at any point after the design brief. However, it would typically be placed at the concepts generation stage of the design process as this is the point at which other conceptual design is undertaken. The system provides a form of ideas stimulus for a product designer and would be used after product research and the PDS had been carried-out and precede the development of the product.

The main problem of the research lay in translating between an aesthetic description and a colour concept, this was reduced to the relation of expressive media to colours. In this sense the two media are connected as objects where the connecting link forms the semiotic relationship. The idea shows how the expression of an aesthetic description can be communicated in any media, the colour concept expresses the aesthetic concept in the media of colour. Linking colour media to any other (eg. words) simply forms the relation and implies that the two mean the same thing, ie. the aesthetic concept.

The main problem in designing a method for forming colour semiotic relations was in the question of subjectivity and objectivity and how this might be performed by a product designer. A method was proposed to base the colour semiotic relations upon symbolism whereby a colour concept could be described by the denoting noun of origin and any other expressive descriptors. This is still subjective as it relies upon the discretion of the designer however the subjectivity is reduced by reference to other external objects. The computer system still allows for a personal style and more freedom of expression (eg. using metaphors) as all semiotic relations are based upon linking words to colours.

The form of the interface representing the colour concept needed to be abstract and neutral so as not to distort the impression by emphasis of shape or location of colour. A pie chart was decided upon as a circle was considered to be the most neutral shape and colour proportions could be indicated by the size of the angle in each colour wedge.

The form of an aesthetic description of a product was considered to determine a way of describing the proposed colours in the context of designing products. The types of words used by product designers in the aesthetic descriptions given for products were nouns, adjectives and verbs. The words described the aesthetics in terms of the product function, image and style, references to other objects that were symbolic of the required aesthetics, attitudes, finishes, visual impression, emotional feelings, environment, product field, the form of the product, texture, integrity of the product and the consumer and the cost. From this it was decided to use verbal aesthetic descriptions as these would be present in a design brief and an aesthetic specification from a PDS and it is the most common form of communication.

The cultural context in which a product would be designed posed a problem in describing colour semiotic relations. The problem is focused in the debate of whether colour semiotics is universal or relative. The problem extends in considering the comparative conceptual interpretation of descriptions in different cultures and if identical or different descriptions imply identical or different meanings. The methodology adopted by the Colour Concept Generator had to account for different cultural contexts. As a starting point it appeared suitable to base colour semiotic relations within the context of a western society to draw upon aspects from the culture relating to colour aesthetic image and style. However as the system learns semiotic relations by linking words and colours, the problem can be overcome by adding a word representative of the culture (eg. China) to the description.

A methodology for learning colour semiotic relations was required from which colour concepts could later be generated. The problem of machine learning of colour semiotic relations was how to translate between themes of an abstract conceptual nature through some reasoning mechanism. Machine learning of colour semiotics need not be a prescriptive process to be rigidly fixed as this assumes that all colours and combinations had been accounted for. The computer system needed to learn in a developing and dynamic way. This is possible as the computer system builds upon a base of colour semiotic relations as new knowledge is learnt.

The colour concept generation process is a mapping of abstract concepts related under a common theme, ie. the aesthetic description and the colours on the screen illustrate a picture of an aesthetic composition. This had to be designed in such a way that the product designer made the decisions with the help of the computer system. The problem could be overcome by providing a more interactive process, as well as generating colour concepts, the system could provide a common description of a colour concept. This gives an insight into other possible semiotic descriptions to achieve the same thing. The process could be iterative and involve starting from a aesthetic descriptors (or colours), generating colours, describing colours, product designer modifying descriptors, generating colours and so on.

The colour semiotic information (colours, descriptors) are represented and manipulated (learnt, generated) in the following way:

REPRESENTATION
colour set: (colour, weighting, colour, weighting ....)
descriptor set: (descriptor, weighting, descriptor, weighting ....)
proportional weighting = weighting / sum of weightings in set

LEARNING
semiotic relation: [(descriptor, average), (descriptor, average) .... colour]
expanded to: [(des, sum, count), (des, sum, count) .... colour]
working average = sum / count
new sum = old sum + (descriptor proportion X colour proportion)

GENERATING
colour weighting = sum of (descriptor proportion X working average)
colour proportion = colour weighting / sum of weightings in set

This chapter aimed to analyse the research problem as a whole and propose a detail specification and design criteria for the Colour Concept Generator. The problems were approached by proposing methods of operation and by reducing the generic base of the research solution to the above equations. The following chapter is the design of the Colour Concept Generator computer system based upon the detail specification outlined in this chapter. An architecture for the system is proposed and the system is analysed and designed using data flow diagrams and coded for implementation.

4 Computer System Design

4.1 Introduction

The previous chapter analysed the problem of the whole research and proposed a detail specification for the design of the Colour Concept Generator. This was reduced down to a collection of mathematical representations in the form of sets and formulae for manipulating the sets as the basis for the design of the computer system. The colour semiotic information (colours, descriptors) was to be represented as sets:

colour set(colour, weighting, colour, weighting ....)
descriptor set(descriptor, weighting, descriptor, weighting ....)

The proportional weighting of each element (colour, descriptor) in the set was to be calculated from:

proportional weighting = weighting / sum of weightings in set

The sets were to be manipulated initially to learn a colour semiotic relation to be expressed in the format:

semiotic relation[(descriptor, average), (descriptor, average) .... colour]

The information contained in this set could be expanded to give the necessary variables to calculate the working average weighting of each associated descriptor:

semiotic relation[(des, sum, count), (des, sum, count) .... colour]
working average = sum / count

In learning a new relation between a descriptor and a colour or relearning a previously taught relation, the new sum was to be calculated from:

new sum = old sum + (descriptor proportion X colour proportion)

A colour concept was to be generated from the descriptor set whereby the colour weighting was to be determined from:

colour weighting = sum of (descriptor proportion X working average)

The colour proportion for the colour concept pie chart was to be calculated from:

colour proportion = colour weighting / sum of weightings in set

The ethos for the system was to be an ideas stimulus for a product designer to create colour schemes that was not dictatorial but worked interactively with the product designer. This was to be the case with the acquisition of colour semiotic knowledge and in the generating of colour concepts. The system boundary was to be aimed at the concept generation stage of the design process and was to take aesthetic descriptors (descriptor set) from a PDS and translate them into a colour concept (colour set) using the outlined formulae.

This chapter is the design of the Colour Concept Generator computer system based upon the detail specification outlined in the previous chapter. An architecture for the system is proposed and the system is analysed and designed using data flow diagrams and coded for implementation. The chapter begins with a look at the computer media. This includes the description of potential AI architectures, neural networks, expert systems and Intelligent Knowledge Based Systems (IKBS).

The architecture of a production system is then described as this is the general format for the Colour Concept Generator. Reference is made to the design of the Colour Concept Generator as each specific part of the AI architectures and a production system is explained. The data structure design in the Colour Concept generator is presented along with the inference mechanism design with reference to a production system. Methods of matching rules and conflict resolution strategies are described and the Colour Concept Generators method of performing these is explained. The programming language from which to code the system is then considered as a fourth generation language (4GL) suitable for the development of an AI system. An existing CAD system is then outlined as the development environment in which the system was coded and tested.

The design for the Colour Concept Generator computer system is then presented in detail in the form of a design specification. The methodology used in the analysis and design is stated and the computer media specified. A description of the design of the system is then provided with reference to the base data flow diagram from which further systems analysis was performed to produce the system code. An example of the system processes is then provided to illustrate the operation of the learning, generating and describing of colour semiotic relations. This is followed by a sample test run using test data to illustrate the response from the system.

4.2 System Architecture Design

This part outlines the architecture of the Colour Concept Generator computer system as an AI computer tool. A description of potential AI architectures and a production system architecture as an Intelligent Knowledge Based System (IKBS) is provided with reference to the design of the Colour Concept Generator system architecture. The computer system is an AI production system that learns colour semiotic relations (created in a database) as facts in a rulebase that holds all of the learnt colour semiotic knowledge. The system generates colours (or descriptions) from descriptors (or colours) held in sets in a database by reference to the rulebase knowledge.

4.2.1 Neural Systems

Neural systems (39) take the form of a connectionist model relating to the nature of the brain's structure. The nodes or units are analogous to the brain's neurones in a sense that they are weighted as a product of the inputs to exceed a threshold. A biological neurone is seen to conduct a similar running process and will only fire when a threshold is reached. The units can be joined to form networks by connections that excite or inhibit other units. The connections are similar to the synapses in the brain and can be excited or inhibited to vary the degree of connection to control the influence of the connected neurone. Activation of units is conveyed in parallel across the units. Neural systems are adaptable as they have the ability to learn and re-learn.

Neural networks model the physical structure of an organic brain based upon a simple model of the brain as a massive connected set of simple decision nodes. Neural networks are 'bottom up' or data driven links and are automatically found to interpret the behaviour of that data. Since all processing is developed from numeric data, explanations of behaviour can only take the form numerical representations such as graphs.

4.2.2 Perceptrons

Layered networks of neurones called 'perceptrons' (40, 41) can govern a strength of unit connection or 'weight'. Careful adjustment of these weights could be used to train a neural network's response to an input pattern. The input pattern can be mapped onto an output pattern based upon a weighted matrix. The system was based upon statistical principles as opposed to Boolean logic as the weights could vary until a desired output was generated from an input. An artificial neurone takes a series of different weighted inputs and adds them together. The result is then compared to a threshold to govern the weighting of the output.

4.2.3 Artificial Neural Networks (ANN)

A commonly used neural network architecture is a 'feedforward' (42) network having units arranged in layers with uni-directional connections. A feedforward network could be a 'basic pattern associator' with an input and output layer or a 'multi-layer' that includes a hidden layer of units as well. In a basic pattern associator all the input units are connected uni-directionally to the output units. There are only two layers (input and output), the input layer units pass their activation on to the output layer units whose activations become the networks solution to the presented problem. The basic task of a pattern associator is to recognise an input and provide a specific output.

A multi-layer feedforward network is used for more complex tasks and requires hidden layers. These operate in the same way as a basic pattern associator and are placed between the input and output layers. A layer of 'context' units can be added that give an additional input to the hidden layer to feed back results from the output to combine with the original inputs from the input layer.

An 'interactive' neural network consists of bi-directional connections so process occurs slightly differently. One or a number of units could be externally activated and their outputs passed onto those units that they were connected to. The nature of the bi-directional connections means that the passing of activations can occur over several cycles until a stable state is reached, a process known as 'parallel relaxation'. When the network stabilises it has reached its lowest possible energy state with the presented input. Those units with the highest activations are seen as the solution to the problem.

4.2.4 Learning in an ANN

Within a connectionist network learning is achieved by adjustment of the connection weights (42). Since short term, volatile knowledge or memories are stored by activation weightings of the units, any enduring characteristics which want to be changed are applied to the connections. The problem is in getting the network to change the connections by the correct way. This means that if the system is to adopt such changes without the aid of a programmer then all weight changes should occur on the basis of local information. The learning in an ANN is a method by which the correct matrix of weights can be found to map the presented input onto the respective output.

Learning methods can be supervised or unsupervised. Supervised learning requires the system being specifically told the desired output and it making weight adjustments by comparing the desired with the actual output. Unsupervised learning involves the system classifying input without any specifically defined outputs. This involves the network trying to group each training pattern into a number of groups. This method is particularly useful when not enough is known about the data to classify it.

4.2.5 Neural Semantics

The success of a neural network design can often depend upon semantic interpretation (42). This refers to how a problem is presented and how the output is interpreted as a solution to the presented problem. Two approaches can be applied, 'localist' whereby one concept is assigned to one unit and 'distributed' whereby one concept is represented across a number of units. The localist method allows for simple interpretation as a unit can be labelled singularly, however this assumes that a particular concept and its semantics are conveyed to the network when they might not be.

The process of interpreting a neural network is an external one, so which concepts a network requires to produce a solution are important in determining meaning from its outputs. In distributed network a particular concept, whether presenting the input or interpreting the output, is presented across a number of units. A method of doing this is to perform an analysis of a concept and use as many units as a concept has features.

Neural networks learn by strengthening links between nodes based upon an extensive training set which must cover all aspects of data behaviour. This training is continued until the net no longer makes any mistakes. The net can generalise to any case that is similar to and within the domain of the training set. Neural networks cannot explain or justify their actions. Mathematical tools can be used to investigate neural network behaviour however multi-dimensional data is difficult to comprehend.

4.2.6 Intelligent Knowledge Based Systems (IKBS)

Expert systems (33) model the processes of the brain of a human expert. The knowledge of the expert must be obtained and broken down into component parts. Expert systems break a problem down into smaller, more manageable parts. Typically these parts consist of production rules written in a natural language form. Many such rules are decision rules and so often take the form of IF .... THEN .... rules. Using natural language means that explanations can be constructed from rules used as part of some reasoning in a natural language like form. Expert systems require designers to elicit, manipulate and represent knowledge from sources of human expertise. Typically they then manually link natural language like IF .... THEN .... rules. Expert systems can explain their reasoning by displaying the rules used to reach a conclusion. They can also explain why they chose a question by displaying the rules used to select it.

The Colour Concept Generator had to have expertise in the application of colour semiotics to products so needed to be a form of expert system and required an element of artificial intelligence. As colour semiotics requires extensive knowledge of colour and related aesthetic descriptors (colour semiotics), the Colour Concept Generator needed to be a form of Intelligent Knowledge Based System (IKBS) (33).

The conceptual architecture of an IKBS consists of three main parts, a knowledge base, an inference engine and a user interface. The knowledge base contains everything that is currently known about the specific domain of expertise and is literally a set of facts or rules of knowledge. In the Colour Concept Generator this includes colour and descriptor semiotic relations.

The inference engine is a mechanism to control they way in which the knowledge is used and reasons logically using rules or facts from the knowledge base in order to solve a specific problem or learn new knowledge. The user interface is the mediator between the user and the system and provides an interactive media for interpreting and manipulating the current state of the system.

4.2.7 Production System Architecture

A production system (33) is a form of IKBS within the field of expert systems and follows a similar conceptual architecture format to an IKBS. The knowledge base consists of a set of conditional statements, productions or rules, termed the rulebase. The database is a collection of given or derived facts providing a working memory and the inference engine controls the execution of rules from the rulebase.

In some production systems the inference engine operates in cycles consisting of three phases, (i) match rules, (ii) select rules and (iii) execute rules. In the match phase the inference engine finds rules in the rulebase corresponding to the current contents of the database according to the rule condition, action or both. The matched rules (termed the conflict set) are potential candidates for execution and the most appropriate rule candidates are subsequently selected for execution through a control method termed the conflict resolution strategy. Rules are executed to alter the current contents of the database and the whole repeating cycle is termed the recognise / act cycle.

4.2.8 Colour Concept Generator Architecture

The Colour Concept Generator system architecture had to be one that could learn new colour semiotic relations and be adaptive (like a neural network) but also be able to deal with natural language (like an expert system or IKBS) as this was to be the aesthetic description medium. The adaptable learning process is a similar architecture to a multi-layer perceptron arranged as a feedforward neural network. For generating colour concepts the inputs are the descriptor weightings and the outputs are the colour proportions.

The hidden layer is a set of descriptor weightings allocated to colours stored as rules in a production system rulebase. The descriptor weightings go through a process of matrix multiplication that includes the descriptor weightings in the colour rules from which the colour proportion outputs are determined. The mapping between the inputs and the outputs is achieved by pattern matching the descriptors with those contained in the rules in the rulebase.

The learning process is supervised by the user who forms colour semiotic associations by allocating colours and proportions to descriptors and weights. However the internal learning in the system is unsupervised as many other colour semiotic relations could have been learnt. The newly learnt colour semiotic knowledge is integrated with all previously learnt knowledge. In this sense colour concepts generated relating to the new colour semiotic relation might not be an exact duplicate to that which was learnt as it depends upon other previously learnt knowledge that has included any of the new colours or descriptors.

The Colour Concept Generator system architecture allows for both localised and distributed semantics. The colours are stored in a production system rulebase with each colour having its own rule, ie. the same number of colour units as there are colours. The semantics can be localised in a sense that one descriptor can be allocated to one colour. Generalised semantics apply as a colour semiotic relation can be distributed over the entire network. Many descriptors can be allocated to many colours so the colour semiotic relation is integrated into the whole rulebase. The application of neural systems in the colour concept generator allows it to be an adaptable system capable of learning in a dynamic way.

The Colour Concept Generator system architecture is structured in a similar way to a production system to allow for the use of natural language. The system required a graphical user interface incorporating a visual image of the colour concept (colour pie chart), colour palette and user menu. All colour and descriptor processing and manipulation is activated via the user interface. An inference engine was required for an examination of colour and descriptor relations in a knowledge base to select colours and proportions for the colour concept based upon given descriptors and weightings.

Other inference engines were needed for updating the knowledge base as new colours and descriptor relations are learnt and for producing a list of associated descriptors from colours in a colour concept. A database is used to represent the current semiotic relation in progress and a rulebase stores the learnt colour semiotic relations. The application of a production system architecture is explained in more detail in the following parts relating to the data structure and inference mechanism.

4.3 Data Structure Design

4.3.1 Database in a Production System

The database in a production system (33) represents the current state of the world and acts as a short term memory mechanism responding to any changes. It contains facts of assertions about the world that might have been put in by the user or generated on account of the inference engine executing rules resulting in a changed current state. The database is a global store that holds knowledge (elements) accessible to the entire system.

The database content determines and controls which part of the production system program is operating, as only rules matching the database content can be executed by the program. The database is the only means of communication between rules as there is no mechanism for passing data from one rule to another as this would require a program function executed by the database content. Database elements can be created, modified or removed by the user or production system program.

4.3.2 Rulebase in a Production System

The rulebase in a production system (33) contains a set of rules representing knowledge facts about the specific expert field. The rules are composed of two parts, the premises and the conclusion. The premises form the condition (situation) part of the rule and the conclusion forms the action (consequent) part. The premises can include combinations of Boolean clauses (predicates) that might specify a restriction on the weighting of some particular property or attribute of some object or element to be represented in the database.

The conclusion forms a list of some modifications to be made to certain elements in the database when the rule is executed upon the satisfaction of the rule premises by another database element. The actions create, modify or delete database elements and can perform external communication with the outside world. Rule interaction is limited but rules can communicate via data in the database or alternatively with a complex message passing process.

4.3.3 Colour Concept Generator Data Structure

In the Colour Concept Generator the database contains the current colours and weightings, the current descriptors and weightings and the current rulebase. The data can be created, modified or removed by the user and production system program through the inference engine executing rules. The current colours and weightings can be altered or generated from the current user-given descriptors and weightings taken by the inference engine to execute the associated rules to produce new colours and weightings.

The rulebase in the Colour Concept Generator holds a set of rules relating descriptors and colours. The rule premises are composed of sets of descriptors and weightings applicable and weighted according to the conclusion which is the associated colour (the relationship having been learnt by the system beforehand). The descriptors and weightings form the condition upon which a particular colour was selected and govern the proportion of which that colour is used in the generated colour concept.

In terms of conditional logic the rule reads:

IF	descriptor OR other descriptor OR other descriptor OR .... present in the current descriptors in the database
THEN	use this colour in the proportion calculated from the database descriptor weightings, the rulebase descriptor weightings and the total proportional weightings of all other colour actions executed as a result of all other descriptors in the database and other rule colours.

The rules are equal in number to the number of colours available in the computer so limiting the number of rules to the number of colours. Each rule contains the colour and the allocated descriptors and average weightings.

4.4 Inference Mechanism Design

4.4.1 Inference Engine in a Production System

The inference engine performs the select / execute loop or recognise / act cycle within the production system. In each loop a selection mechanism is applied repeatedly in order to choose a rule applicable to the current state of the database, after which the rule is sorted for executing. In some cases the execution of the rule results in a modified database and the select phase begins again. Rules in a production system can be applied in either direction corresponding to the type of reasoning strategy adopted by the inference engine (43).

4.4.2 Forward and Backward Chaining

The inference engine (43) examines the database to establish which rules have their premises (condition) satisfied, selects one of them and executes it by performing the corresponding action. The rules that are executed are chosen from information in the database. The method is termed forward chained or data driven, the control strategy moves from conditions to actions or from data to goals. The generalised rule, descriptor generates colour, reads that upon presence of descriptor the outcome is colour.

The same rule could be open to another interpretation whereby upon an outcome of colour the descriptor is presented. In this alternative execution strategy each cycle selects rules containing current goals or actions by the rule conclusion. If the goal is satisfied by the database then using that rule would be relevant in satisfying the goals, if not then re-select. The method might use the same or different information more applicable to goals. The selection of rules to be executed is determined more by what is expected to be found rather than what is currently in the database. The method is termed backward chaining or goal driven strategy.

Both methods are applicable depending upon the knowledge domain and both forward and backward chaining can be used with respect to search spaces. The control system decides which direction is most applicable according to the given type of reasoning and moves from data to goal for forward reasoning and from goal to data for backward reasoning. Forward chaining is more applicable to synthesis and production whereas backward chaining is more applicable to analysis and diagnosis.

4.4.3 Colour Concept Generator Inference Engines

The Colour Concept Generator mainly consists of two inference engines, one for learning colour semiotics and one for generating colour concepts. The learning inference engine takes the current descriptors and weightings and the current colours and weightings and updates the rulebase. The descriptors and weightings in the rulebase can be created or modified by processes activated by the learning inference engine. The rule descriptor weightings are determined by the current colour weightings, the current descriptor weightings and the old weighting of the rule descriptor. This provides a dynamic learning strategy and an ongoing learning process forming rules of colour and descriptor relations that develop as the system learns more about colour semiotics.

The colour concept generating inference engine generates a set of current colours and weightings (colour concept colours and proportions) by selecting and executing rules (in the rulebase) based upon the set of current descriptors and weightings from the database. The control strategy is forward chaining in a sense that rule colours (actions) are selected by satisfaction of rule descriptors (conditions) in the rule.

The executed rules produce the current colours with weightings (proportions) calculated by current descriptor weightings, rule descriptor weightings and proportional weightings of all selected colours. The forward chaining of the generate inference engine is enhanced by providing a third inference engine (describe) to provide a description of the current colour concept. This mechanism allows for more user interaction with the production system as description information could be used to expand the current descriptors to make the description more specific.

Upon expanding the description the user could then re-generate the colour concept. The process could then be repeated, ie. produce current descriptors and weightings, generate colour concept, modify colour concept, describe colour concept, modify current descriptors, generate colour concept and so on. The describe inference engine provides a list of all common descriptors associated with the modified colour concept which the user could use in making the current descriptors and weightings more specific. The control strategy method for the describe inference engine is backward chaining as the common rule descriptors (conditions) are selected by satisfaction of a rule colour (action) in the rule. The selected rules are executed to produce a common description of the colours in the colour concept.

4.5 Matching Rules in a Rulebase

Rules can be selected by matching database elements with either the rule premises, rule conclusion or both. The matching process in a production system can be based upon a single identity (literal match), a pattern (pattern match) or a method that supports both sides of a rule (unification) (32).

4.5.1 Methods of Matching Rules

A literal match would give a colour from a descriptor or vice-versa, ie. if the rule reads IF descriptor THEN colour, matching by either descriptor or colour would result in colour or descriptor.

A pattern match would give a result if the candidate descriptors were inserted into the rule premises as variables and satisfy the rule premise conditions. If the rule reads IF colour A > colour B THEN colour A lighter than colour B, the rule would be executed if colour A > colour B. If this was the case and the variables were swapped then the rule would not be selected as the rule premise was not met.

A unification match would satisfy both rule premises and rule conclusion. If say the colour numbers were to be ordered numerically then the previous example could be used on all colours in a repeating process. The ordered colours could be checked against the same rule to check that the higher was greater than the lower so satisfying both sides of the rule.

4.5.2 Rule Matching in the Colour Concept Generator

The Colour Concept Generator uses a combination of both literal and pattern matching within the generating inference engine and unification and literal matching within the learning inference engine. Generating inference requires the selection of colours from the descriptors, the rule holds a colour and a set of associated descriptors to be matched with the current descriptors. A literal match is applicable as if all the current descriptors are present within the colour rule descriptor sets (rule premises) then the colour is a potential candidate. By literal matching logic, if all of the current descriptors were not present in the colour rule descriptor set then the colour is not applicable as all database elements were not present in the rule premises.

A pattern match might find that all current descriptors are present within the rule premises yet not all colour rule descriptors are in the current descriptors. By pattern matching logic the rule is not selected yet it is a potential candidate on account of it satisfying all elements in the database. The matching solution for the Colour Concept Generator was a compromise of both literal and pattern matching. A colour is selected if all current descriptors are present within the colour rule descriptor set irrespective of whether or not all the rule premises are present within the current descriptors.

The describe inference engine is based upon literal matching. Rules are matched by the current colours as a rule colour present within the current colours returns all of the common descriptors from the matched rule premises. This does not modify the current descriptors but is a separate list of descriptors without weightings that consists of all of the common descriptors from the matched rule descriptor sets.

The learn inference engine matches rules by literal matching and by unification. It modifies the rulebase descriptor weightings from the current descriptors and weightings, current colours and weightings and existing colour rule descriptors set weightings. Learning inference rule selection selects rules for modifying by unification, ie. rules that hold descriptors present in the current descriptors and a colour present in the current colours. It also selects rules for modifying by literal matching and backward chaining to create a new rule colour descriptor set and weighting for the rule colour. Forward chaining does not apply in rule selection for modification as a backward chained match by colour selects all the necessary rules.

4.6 Conflict Resolution Strategy

The performance of an inference engine and a production system as a whole depends upon the conflict resolution strategy for both sensitivity and stability. Sensitivity relates to the response of the system to dynamically changing states. Stability is the systems continuity of behaviour and consistency of response to similar situations. The conflict resolution strategy decides which rules are to be executed from a set of potential candidates (the conflict set) and can be achieved by refraction, data ordering, specificity ordering or rule ordering (33).

4.6.1 Methods of Conflict Resolution

Refraction prevents rules from being executed more than once upon the same data. It is intended to prevent a form of infinite looping that might occur if an executed rule does not change the current state of the database. Data ordering orders data by recency of activation, giving preference to rules that have matched database elements most recently. Specificity ordering favours rules that are special cases of other selected rules or are more specific by some other criteria. Rule ordering provides a static ordering of the rulebase predetermined by the particular knowledge field or area of expertise in the production system.

4.6.2 Conflict Resolution in the Colour Concept Generator

The Colour Concept Generator resolves the conflict set through proportional weighting of the current descriptors, current colours and colour rule descriptor set weightings. The refraction method is used to prevent infinite looping of rule execution and is interactive with the user through the process previously explained of create current descriptors, generate colours, modify colours, describe colours, modify current descriptors, generate colours and so on.

The system follows an expansive and accumulative approach initially, with the intention of diverging to all colour semiotic relations inherent within the system for the current descriptors. The colour concept becomes more refined through interaction with the user, as the current descriptors expand and become more specific the process converges in a more selective manner to produce a more refined colour concept. Conflict resolution within the computer system is mainly an interactive process controlled by the user.

4.7 Programming Language and Development Environment

The programming language had to be one that was suitable for developing an AI system. Although an AI system could be coded in any language it was decided to use a forth generation language (4GL) as these are specific to the field of AI. Lisp and Prolog were considered as they are recognised 4GLs for AI. The languages are outlined here to explain the basic structure and operation. The reasons for selection of the language and prototype development environment is stated based upon the compatibility with an existing CAD system that is used in the field of product design, colour rendering capabilities, previous knowledge, experience and familiarity by the researcher and availability within the Department of Product Design and Manufacture at Bournemouth University.

4.7.1 Prolog

Prolog (PROgramming LOGic) (44) involves pattern matching, automatic backward chaining and using tree-like data structures. This type of programming is suitable for problems that are object oriented with specified inter-relationships. Prolog has a built in backward chaining reasoning mechanism and it is possible to develop and add a forward chained reasoning facility to it. Programming in Prolog consists of declaring some facts about objects and their relationships, defining some rules about objects and their relationships and asking questions about objects and their relationships. A program consists of clauses of three types, (i) facts, (ii) rules and (iii) questions.

Facts declare things that are always unconditionally true. They consist of several objects and a relationship, eg. complements (red, green). This fact can be interpreted as red complements green. In fact the order is arbitrary but once a convention is made to which order facts should be interpreted, it should be consistent for the other rules. A relation can be specified by facts, simply stating the number of objects that satisfy the relation, or by rules about the relation.

Rules declare things that are true depending upon a given condition. A rule is a general statement about objects and their relationship. Head and body Prolog clauses consist of a head and a body. The body is a list of goals separated by commas understood as conjunctions, facts are clauses that have an empty body, questions only have a body and rules have a head and a body. By means of questions the user can determine what things are true. Once there are some facts in the database queries can be made about them.

Prolog programs can be understood in two ways, (i) declarative and (ii) procedural. The difference between declarative and procedural interpretation is that the procedural reading does not only define the logical relationship between the head and the body of a goal but also the order in which the goals are processed.

With declarative interpretation, facts, rules and goals are all structures. A fact is a structure used as an assertion and a goal is a structure used as a question. Each fact and each rule is called a clause, a rule is a clause which has a head and a body, a fact is a clause which has a head and no body. In a database a collection of clauses whose heads have the same functor and the same arity is called a procedure. A procedure is a set of clauses about the same relation. The set of clauses of a procedure defines a relationship. The relationship defined by a procedure is called a predicate (45).

The declarative semantics of Prolog defines whether a goal is true and if it is then for what instantiation of variables it is true. The procedural semantics of Prolog is a procedure for satisfying a list of goals. The procedure outputs the truth or falsity of the goal list and the corresponding instantiation of variables. In pure Prolog the declarative concept of programs does not depend on the order of clauses and goals in clauses. However, the order of clauses is directly related to the procedural meaning of the programs. The order can affect the efficiency of the program (avoiding open loops).

The only data type provided by Prolog is called a term. All the objects in the problem and all the relationships between objects are represented using terms, eg. date(08, Oct, 1973). This is an example of a structured term, or a structure for short.

4.7.2 Lisp

The Lisp programming language (46) is a recognised vehicle for the development of AI. It is based upon functional programming and symbol processing, specifically designed for symbolic manipulation. The main technique of Lisp relies upon recursive function calls in conjunction with conditional statements. Lisp involves the manipulation of symbolic expressions (s-expressions) which can be atoms, lists or empty lists (nil). Atoms can be numeric (integers or reals) or symbolic identifiers (variables).

The list is the format of both program and data, hence the name Lisp (LISt Processing). In programs the s-expressions are seen as functions with the first expression applied to the list arguments, eg. (add 1 1) will return 2. In nested expressions the deepest most indented arguments are evaluated first with the result being passed on to encapsulating functions, eg. (minus 4 (add 1 1)) will return 2. All s-expressions are evaluated by the interpreter which returns the result to encapsulating functions or the screen.

The Lisp environment includes all standard arithmetic and logic functions as well as the fundamental aspects of a programming language, sequence, selection, repetition and procedure. Sequence is dictated by ordered function calls within lists, ie. (do this (with that (and the other))). Selection is achieved through the COND statement which replaces nested Ifs and CASE OFs. Repetition employs standard WHILE DO loops and recursion. Procedure is provided by activation of a function.

Recursion (47) involves looping internally by recalling a function from within the same function. This can include alteration to the list as an input parameter to the function, ie.

```
(defun add list)
        (+      (car list) (add (cdr list)))
)
```

The above function might be required to add together a list of numbers. Defun defines a function named add and takes an input parameter of a list. The function requires an addition of the first list element (car list) to the addition of the rest of the list (cdr list). The add function is recalled recursively until the end of the list (nil) is encountered and all the list elements are added. The total is then evaluated and returned.

Lisp was considered to be the most suitable programming language for the computer system as a production system. The database could be expressed as lists, ie. a current colours and weightings list and a current descriptors and weightings list. The rulebase could be a list of lists (rules) with each rule being a list of descriptors (descriptor set) containing the descriptor name, weighting information (sum and count) and associated colour for the rule. The inference engine could be a list manipulation mechanism that selected rules by matching colour and descriptor lists (database) with the rule list (rulebase). The execution of rules results in the modification of database lists. The inference engine activates the appropriate Lisp functions in the order required for colour semiotics learning, colour concept generation and colour concept description.

4.7.3 AutoCad and AutoLisp

AutoCad (48) is a general purpose computer aided design package widely used within the field of product design. It provides drawing and illustration facilities to cater for all visual presentation of products from BS308 engineering drawing to rendering. The system has a readily accessible colour palette consisting of 256 colours.

AutoCad is written in Lisp and provides and interactive programming environment called AutoLisp (49) which follows a similar format to Lisp. Macro programmes and entire systems can be written in AutoLisp to be called up from the AutoCad menu or respond in real time to drawing activities. The AutoCad functions can be activated from the AutoLisp environment so the AutoCad system can effectively be controlled by an external program.

The availability of AutoCad within product design, the colour rendering and drawing capabilities and the AutoLisp programming environment made

AutoCad a suitable prototype design and development environment for the Colour Concept Generator. The AutoCad and AutoLisp systems were accessible in the Department of Product Design and Manufacture at Bournemouth University. A production system was implemented in AutoLisp and used the AutoCad graphical functions to present the colour concept and colour palette in the user interface. All processes excluding the input of descriptors (typed on the keyboard) were controlled from the AutoCad menu and computer mouse.

4.8 System Design Specification

This part is a detailed account of the design of the Colour Concept Generator computer system. The methodology and computer media are stated as is the operation of the computer system as an AI production system. Reference is made to the base data flow diagram (DFD 0 Colour Concept Generator) presented in appendix G to explain the structure of the database and rulebase and the operation of the learn, generate and describe inference engines. The DFDs are listed in appendix G and the system code is listed in appendix B to provide a more detailed representation of the design of the computer system.

4.8.1 Methodology for Analysis and Design

The system was analysed and designed using DFDs (Data Flow Diagrams) in a similar way to SSADM (Structured System Analysis and Design Methodology) (50) in accordance with the system specification for the selected computer media. The analysis was performed using the TDSW (Top Down Step Wise) method that consisted of a hierarchical breakdown of the system into discrete processes and functions. The processes were indexed according to their position in the hierarchy and relation to other processes. An algorithm was produced for each process with reference to sequence, selection, control and interaction with other processes and base functions. The base functions were built bottom up in line with each algorithm.

4.8.2 Computer Media

The system was developed in the AutoCad environment with the program coded in AutoLisp to interface with the AutoCad graphics commands and colour palette. The architecture for the system is in the form of an AI rule based production system and includes a user interface, database, rulebase and inference engines. The system architecture is shown in DFD 0 (Colour Concept Generator). The analysis DFDs are listed in appendix G and the system code is listed in appendix B. The following is a description of the system architecture referring to DFD 0.

4.8.3 Database Design

The database is a short term working memory that stores the current state of the colour concept generating process at a particular time. The current colours, current descriptors and up to date rulebase are available to all relevant processes via the database. The colours and weightings and the descriptors and weightings are presented in list format. The rulebase is a list of rules containing 254 rules (one for each colour) each of which holds the learnt, related descriptors together with the information required to calculate the working average descriptor weighting (sum and count).

4.8.4 Current Colour Set

The current colours and weightings are stored in the database in the form of a list in the following format:

(colour, weighting, colour, weighting, colour, weighting ....)

4.8.5 Current Descriptor Set

The descriptors and weightings are stored in the database in the form of a list in the following format:

(descriptor, weighting, descriptor, weighting, descriptor, weighting ....)

4.8.6 Rulebase Structure

The rulebase contains descriptor and colour relations in the form of a many to many relationship. The rules are stored in the database in the form of a list of lists (rules). Each rule contains a list of associated descriptors (rule premises) and a colour number (rule conclusion). The descriptors are a list of descriptor sets containing the descriptor, sum and count for that particular rule colour. The rulebase is composed in the following way:

descriptor set: (descriptor, sum, count)

rule premises: ((descriptor set), (descriptor set), (descriptor set) ....)

rule: ((rule premises), colour)

The expanded rulebase is in the following format:

```
(
        (       (       (descriptor, sum, count) .... ) colour)
        (       (       (descriptor, sum, count) .... ) colour)
        (       (       (descriptor, sum, count) .... ) colour)
                                                                .... )
```

4.8.7 User Interface

The interface provides the user with an editing facility to manipulate the current colour concept and current descriptors including adding, deleting, proportioning, weighting and displaying colours and descriptors. A display operates via a mouse driven menu to activate the colour palette and colour concept which is presented in the form of a circular pie chart and is plotted from a colour palette list and a current colours list.

The background colour is a neutral black so as not to distort the impression of the colours through simultaneous contrast. A neutral grey and a white were tested as background colours but black proved to be visually the most suitable. The aim was to provide the user with an interactive visual picture of the colour concept generation process. Colours, descriptors and their weightings can be input by the user via the user interface and can be displayed on screen having been stored in the database. The design of the user interface is shown in appendix C.

The interface is mouse driven apart from the input of descriptors and weighting weightings which are typed in via the keyboard. Colours are chosen from the colour palette by clicking the mouse on a colour and are initially seen in the centre to evaluate the colour as a potential candidate to be added to the colour pie chart. Variations in saturation and pastel / bold colours are changed by clicking the mouse in the centre or to the outside of the viewed colour. When a colour is accepted (by clicking on the colour pie chart) the colour transfers to the colour pie chart and is allocated a numerical weighting if another colour is already present. The weighting allocated is the proportion which the new colour has in the colour pie chart with all other colours re-proportioned according to their previous proportions. When a colour concept is finished the colour palette is removed by clicking outside the palette circle. All other functions in the interface menu can be activated by clicking the mouse on the menu items.

4.8.8 Menu Commands

CCGENRUN	-Initialisation of system, resets current colour concept, descriptors and weightings.
DESADD	-Adds a descriptor to the current descriptors.
DESDEL	-Deletes a descriptor from the current descriptors.
DALLDES	-Deletes all the current descriptors.
DESVIEW	-Displays all the current descriptors.
DESCRIPS	-Displays a list of all descriptors known by the system.
COLADD	-Adds a colour to the current colour concept.
COLDEL	-Deletes a colour from the current colour concept.
PROPCOL	-Re-proportions a colour in the current colour concept.
DALLCOL	-Deletes all colours in the current colour concept.
COLVIEW	-Displays the current colour concept.
TEACH	-Forms a semiotic relation in the system between the current colour concept and descriptors.
GENERATE	-Generates a colour concept from the current descriptors.
(NEXT)	-Toggles between the main menu and 'DESCRIBE'.
DESCRIBE	-Generates a list of descriptors from the current colour concept.

4.8.9 Learn Inference Engine

The learn process is an inference engine that forms colour and descriptor semiotic relations to modify the rulebase. The process takes the list of colours and weightings, the list of descriptors and weightings and the rulebase from the database and generates the new modified rulebase. The new rulebase is placed back in the database and a copy is automatically saved to disc.

This is achieved by taking the current colours list and the current descriptors list from the database and extracting the colour rules from the rulebase by matching descriptors and colours. Each colour rule is modified by calculating the new sum weighting and incrementing the count weighting by one for each matched descriptor in the rule. If a descriptor is not already present in the colour rule then that descriptor is inserted with a count of one. The sum for each matched descriptor is calculated by multiplication of the descriptor proportion and colour proportion in the current lists.

4.8.10 Generate Inference Engine

The generate process is an inference engine that generates the colour concept from the descriptors. The process takes the list of descriptors and weightings from the database and generates the new list of colours and weightings from which the colour concept is plotted on the screen.

This is achieved by taking the current descriptors list from the database and extracting rules from the rulebase by matching the descriptors with the descriptor sets in the rule. A rule will not be selected unless it contains all of the descriptors in the current descriptors list. The working average for each matched rule descriptor is determined by dividing the sum by the count. The colour proportions for the colour concept pie chart are calculated by multiplying the matched rule descriptor proportion by the current descriptor proportion and translating to the angle in radians.

4.8.11 Describe Inference Engine

The describe process is an inference engine that generates a list of common descriptors associated with the colours in the colour concept without modifying the current descriptors. The process takes the list of colours and weightings and the rulebase from the database and generates a list of common descriptors associated with the current colours. This is achieved by taking the current colour list from the database and extracting rules from the rulebase by matching the current colours with the rule colour. If the rule colour is in the current colours list then the rule is selected. All of the descriptors in the selected rule are removed and added to the associated descriptors list.

4.8.12 File Handling Process

The file handling process saves the rulebase to the disc from the database and loads the rulebase from the disc to the database. This occurs upon initialisation of the system and upon modification of the rulebase executed through the learn process.

4.9 Example of System Processes

The previous section detailed the specification for the design of the computer system. The following is an example of the system processes and includes rulebase learning of colours and descriptors, generating colours from descriptors and producing an associated descriptors list from colours. The example details the mathematical procedures applied and is presented in a step-by-step process.

4.9.1 Rulebase Learning of Colours and Descriptors

The rulebase consists of a list of rules, one for each colour. Each colour rule contains descriptor sets consisting of a descriptor, sum and count. When the learn process is activated a new descriptor set is created if the descriptor is not already contained in the rule. The sum becomes the result of the matrix multiplication of the current colours and descriptors weightings and the count is given a weighting of one.

Assuming that the current colours and descriptors lists to be learnt are as follows:

current colours = (blue, 0.2, orange, 0.5, white, 0.3)

current descriptors = (des1, 0.4, des2, 0.6)

The colour rules are found in the rulebase by pattern matching the current colours with the rulebase colours. The sum weightings in the descriptor sets are calculated by matrix multiplication of the weightings as follows:

			des1	des2
blue	0.2			
orange	0.5	X	0.4	0.6
white	0.3			

The resulting matrix gives the sum weightings for each colour rule as follows:

	des1 sum	des2 sum
blue	0.08	0.12
orange	0.2	0.3
white	0.12	0.18

The subsequent colour rules read as follows:

((des1	0.08	1)	(des2	0.12	1)	blue)
((des1	0.2	1)	(des2	0.3	1)	orange)
((des1	0.12	1)	(des2	0.18	1)	white)

Another learn acquisition could occur that included descriptors that were already present in the colour rules for the colours they were to be associated with. If the descriptor is already present in the rule, the new sum weighting becomes the old sum weighting added to the result of the matrix multiplication of the current colours and descriptors weightings and the count is incremented by one. This enables a working average to be calculated by dividing the sum by the count.

4.9.2 Generating the Current Colours List

The current colours list is generated from the current descriptors list containing descriptors and weightings. If, for example other colour semiotic associations had been learnt that included a descriptor 'des3' and a colour 'pink', the rulebase might read as follows:

((des1 0.08 1)	(des2 0.12 1)	(des3 0.4 1)	blue)
((des1 0.2 1)	(des2 0.3 1)		orange)
((des1 0.12 1)	(des2 0.18 1)	(des3 0.6 1)	white)
((des1 0.18 1)	(des3 0.22 1)		pink)

Assuming that the current descriptors list from which to generate the colour concept is as follows:

current descriptors = (des1, 0.3, des3, 0.7)

To generate the colour concept the colours are found by pattern matching colour rules that only include all of the current descriptors. In this particular case the colour rules blue, white and pink would be selected as follows:

((des1 0.08 1)	(des2 0.12 1)	(des3 0.4 1)	blue)
((des1 0.12 1)	(des2 0.18 1)	(des3 0.6 1)	white)
((des1 0.18 1)	(des3 0.22 1)		pink)

A particular colour weighting for the colour concept to be generated is determined by summing the multiple of the current descriptor proportions and the rule descriptor weightings:

colour weighting = sum of
(current descriptor proportion X rule descriptor weighting)

The weightings for each colour are determined as follows:

blue =		des1	(0.3 X 0.08)		
	+	des3	(0.7 X 0.4)	=	0.304
white =		des1	(0.3 X 0.12)		
	+	des3	(0.7 X 0.6)	=	0.456
pink =		des1	(0.3 X 0.18)		
		des3	(0.7 X 0.22)	=	0.208

The proportioned current colours for the generated colour concept are as follows:

current colours = (blue, 0.314, white, 0.471, pink, 0.215)

4.9.3 Producing the Associated Descriptors List

The associated descriptors list provides a description of the current colours. The associated descriptors are found by pattern matching the colours in the current colours list with the colour rules and extracting all of the descriptors common to each colour rule. Assuming the rulebase is the same as the previous example as follows:

((des1 0.08 1)	(des2 0.12 1)	(des3 0.4 1)	blue)
((des1 0.2 1)	(des2 0.3 1)		orange)
((des1 0.12 1)	(des2 0.18 1)	(des3 0.6 1)	white)
((des1 0.18 1)	(des3 0.22 1)		pink)

Assuming that the current colours list is as follows:

current colours = (orange, 0.2, white, 0.8)

Using the matched colour rules for orange and white:

((des1 0.2 1)	(des2 0.3 1)		orange)
((des1 0.12 1)	(des2 0.18 1)	(des3 0.6 1)	white)

The descriptors common to both colours are des1 and des2, so the associated descriptors list will be as follows:

associated descriptors = (des1, des2)

4.10 Sample Test Run

A sample test run was performed to test the operation of the user interface and menu function commands as well as the operation of the system code. A colour image of a surgeon was used to create the colour concept to which descriptors were allocated based upon a medical theme. The colour pie-chart was created and descriptors and weightings entered and the colour concept was learnt by the system. Descriptors were entered from which colours were generated and descriptions were produced from colours.

4.10.1 Colours and Proportions

A surgeons clothing and equipment was used for an image of a surgical and medical theme. Through averaging out the exact colours the following colours and percentage proportions for the colour concept were proposed:

COLOUR	NUMBER	PERCENTAGE (approx.)
Pale Grey	254	10
Pale Blue	130	10
Pale Green	120	10
White	7	40
Grey	252	10
Blue	140	10
Green	110	10

4.10.2 Descriptors and Proportions

The following descriptors and percentage proportions were allocated to the colour concept:

DESCRIPTOR	PERCENTAGE (approx.)
Surgical	25
Medical	15
Hygienic	15
Caring	15
Clinical	15
Sterile	15

4.10.3 Creating the Colour Concept Pie-Chart

Six of the colours are in equal proportion so these were entered into the concept first. 'COLADD' was activated and the colour green (110) was selected, this was accepted into the colour concept by clicking in between the central colour circle and the outer colour palette. Next the colour blue (140) was selected and accepted into the colour concept. The prompt requested a weighting, this colour had a weighting of 50% as it is required at the same proportion as the other. A weighting of 50 was typed in and return pressed.

The next colour entered was pale green (120), this was selected and accepted and the weighting 33.33333 (one third as it is the same proportion as the other two) was entered. The same was carried out for pale blue (130) using a weighting of 25 (a quarter). Grey (252) and pale grey (254) were entered using weightings of 20 (a fifth) and 16.66666 (a sixth) respectively. The central colour circle was required to move through the greys to 252 and 254. White (7) was entered at a weighting of 40 and the colour concept was then accepted by clicking outside the pie-chart. The colour concept was displayed along with the following colours prompt:

current colours:

7	40.0
254	10.0
252	10.0
130	10.0
120	10.0
140	10.0
110	10.0

4.10.4 Removing Colours from the Colour Concept Pie-Chart

Having created the colour concept the following colours were removed from the colour pie-chart: grey (252), blue (140) and green (110). The 'COLDEL' menu option was activated and the grey section of the pie chart was clicked on. The process was repeated for blue and green and the colour concept was displayed with the following colours prompt:

current colours:

7	57.1429
254	14.2857
130	14.2857
120	14.2857

4.10.5 Re-proportioning Colours in the Colour Concept

Re-proportioning the colour pie-chart was tested by making the pale green and blue bigger and the white and smaller. The 'PROPCOL' menu option was activated and clicks towards the outer edge of the pale green and blue wedges (to make bigger) and towards the thinner end of the white wedge (to make smaller) altered the proportions. 'PROPCOL' needed only to be activated once to repeatedly alter the colour wedges. Each time a click was performed the colour pie-chart was displayed showing the colours in the new proportions as was also indicated at the colours prompt. The colour wedges were re-proportioned until the pale green, pale blue and white colours were approximately equal in proportion (approx. 30% each) then accepted into the colour concept by clicking outside the colour pie-chart. The colour prompt read:

current colours:

7	30.3777
254	10.8884
130	29.367
120	19.3669

4.10.6 Creating the Description

The descriptors were entered by selecting the 'DESADD' menu option. The current descriptors were displayed as blank as was the descriptor menu and the 'enter descriptor' prompt appeared. Five of the descriptors had an equal proportion (15%) so these were entered first, "Medical" was typed in and return pressed. The descriptor was entered and had a weighting of 100% (as indicated at the descriptor prompt) as no other descriptors had been entered.

The other descriptors of equal proportions (hygienic, caring, clinical and sterile) were entered by repeating the process of selecting 'DESADD' and typing in a descriptor and a weighting. Each weighting was according to the proportion at which it was entered, ie. a half (50), a third (33.33333), a quarter (25) and a fifth (20). Once these were entered the descriptor prompt listed all descriptors with a weighting of 20 (a fifth) as follows:

current descriptors:

medical	20.0
hygienic	20.0
caring	20.0
clinical	20.0
sterile	20.0

4.10.7 Removing Descriptors from the Description

The descriptor "caring" was removed by selecting 'DESDEL' from the menu, typing in "caring" and pressing return. The descriptor was removed from the current descriptors and all descriptors had an equal weighting of 25 (a quarter) as follows:

current descriptors:

medical	25.0
hygienic	25.0
clinical	25.0
sterile	25.0

Finally the descriptor "surgical" was entered with a weighting of 25, the descriptor prompt read (depending on the order of entry) the following:

current descriptors:

surgical	25.0
medical	18.75
hygienic	18.75
clinical	18.75
sterile	18.75

4.10.8 Learning a Colour Semiotic Relation

The colour semiotic relation was learnt by the system by activating the 'TEACH' menu option. The colour and descriptor data was accepted into the rulebase and the colour rules were as follows:

DES	SUM	COUNT	COL. NO.	COLOUR
surgical	0.0759442	1		
medical	0.0569581	1		
hygienic	0.0569581	1		
clinical	0.0569581	1		
sterile	0.0569581	1	7	white
surgical	0.027221	1		
medical	0.0204158	1		
hygienic	0.0204158	1		
clinical	0.0204158	1		
sterile	0.0204158	1	254	pale grey
surgical	0.0734174	1		
medical	0.0550631	1		
hygienic	0.0550631	1		
clinical	0.0550631	1		
sterile	0.0550631	1	130	pale blue
surgical	0.0734174	1		
medical	0.055063	1		
hygienic	0.055063	1		
clinical	0.055063	1		
sterile	0.055063	1	120	pale green

4.10.9 Generating a Colour Concept

The current descriptors were blanked by activation of the function 'DALLDES'. The descriptor "medical" was entered and the 'GENERATE' function activated. The colour concept appeared with the following colours and weightings:

current colours:

7	30.3777
120	29.3669
130	29.367
254	10.8884

4.10.10 Describing a Colour Concept

The current colours were blanked by activation of the function 'DALLCOL'. The colour white was entered into the colour pie-chart and the 'DESCRIBE' function activated. The description was produced with the associated descriptors as follows:

associated descriptors:

sterile
clinical
hygienic
medical
surgical

The sample test run was performed to test the operation of the user interface and system functions. The numerical results from the test data were checked and showed that the system was working according to the requirements of the design specification. The system code was calculating proportions and weightings according to the method stipulated. The system code was functioning in line with the proposed architecture. Colours, descriptors and weightings were accumulating in the database lists and the rulebase allocated descriptors to their respective colours during the learning process. The generate and describe processes matched the appropriate colours and descriptors to produce the associated descriptors and colours.

4.11 Summary

The Colour Concept Generator computer system was analysed and designed using DFDs (Data Flow Diagrams) in accordance with the system specification for the selected computer media. An algorithm was produced for each process and the base functions were built bottom up in line with the algorithms. The system was developed in the AutoCad environment with the program coded in AutoLisp to interface with the AutoCad graphics commands and colour palette. The architecture for the system was in the form of an AI rule based production system and included a user interface, database, rulebase and three inference engines (learn, generate and describe).

A database contains the current colours, current descriptors and their respective weightings and up to date rulebase all presented in list format. The rulebase is a list of rules containing a rule for each colour and contains descriptor and colour relations in the form of a many to many relationship. Rules are stored in the database in the form of a list of lists (rules). Each rule contains a list of associated descriptors (rule premises) and a colour number (rule conclusion).

A user interface provides the user with an editing facility to manipulate the current colour concept and current descriptors including adding, deleting, proportioning, weighting and displaying colours and descriptors and operates via a mouse driven menu. The colour concept is presented in the form of a circular pie chart and the background colour is a neutral black so as not to distort the impression of the colours through simultaneous contrast. The design of the user interface is shown in appendix C.

A learn inference engine forms colour and descriptor semiotic relations to modify the rulebase. The process takes the list of colours and weightings, the list of descriptors and weightings and the rulebase from the database and generates the new modified rulebase. The new rulebase is placed back in the database and a copy is automatically saved to disc. A generate inference engine generates the colour concept from the descriptors. The process takes the list of descriptors and weightings from the database and generates the new list of colours and weightings from which the colour concept is plotted on the screen.

A describe inference engine generates a list of common descriptors associated with the colours in the colour concept without modifying the current descriptors. The process takes the list of colours and weightings and the rulebase from the database and generates a list of common descriptors associated with the current colours. A file handling process saves the rulebase to the disc from the database and loads the rulebase from the disc to the database. All of the interface and menu functions performed according to the requirements stipulated in the design specification when a sample test run was undertaken.

This chapter provided the design of the Colour Concept Generator computer system based upon the detail specification outlined in the previous chapter. A production system architecture and the AutoLisp programming language within the AutoCad system were considered suitable for the development of the Colour Concept Generator. The following chapter is an evaluation of the whole research project, undertaken to obtain an objective opinion of the idea of computing colour aesthetics, the system philosophy, methodology and the computer media used in the Colour Concept Generator.

5 Evaluation

5.1 Introduction

The previous chapter provided the analysis, design and implementation of the Colour Concept Generator computer system, this chapter is an evaluation of the whole research project. The design of the evaluation for the system had to account for the nature of the research and the Colour Concept Generator. The research was evaluated through a qualitative analysis by observation, questionnaire and discussion using as many participants as possible. The participants were potential users of the system in the field of product design. This was undertaken to obtain an objective opinion of the idea of computing colour aesthetics, the system philosophy, the system methodology and the computer media used in the Colour Concept Generator. From the personal opinions expressed in the questionnaires and discussion and the observations of the participants, a qualitative analysis was performed to establish conclusions for the research and consider further research in this area for the future.

The evaluation was based upon the proposed system methodology for learning colour semiotic relations and generating colour concepts. This involved participants teaching the system colour semiotic relations either from existing colour images or personally created colour concepts for analysis and allocation of aesthetic descriptors. The teaching could start from descriptors or from colours using single or many descriptors and colours.

The colour concepts were created on the screen by matching colours and proportions by eye and the descriptors could be any type of word associated with the colours. The generation of colour concepts could involve the method stipulated for the system of generate colours, describe colours, modify descriptors, generate colours and so on, or some other way deemed suitable by the participants. The participants then discussed their experience of using the Colour Concept Generator and completed questionnaires from which, combined with the observations, the analysis for the evaluation was undertaken.

5.2 Experimental Method Using Product Designers

The following is a breakdown of the procedure undertaken to evaluate the research of the Colour Concept Generator. It begins with a profile of the participating product designers. The format is then outlined in terms of the design of the operating instructions for the participants and the process they went through in using the system to evaluate it. The design of a questionnaire that each participant completed is outlined in terms of the main areas that were questioned and the type and number of questions asked.

5.2.1 Participating Product Designers

In order to obtain a realistic evaluation of the Colour Concept Generator it was considered essential to involve potential real users of the system. Twenty two final year product design students (all with at least one year industrial experience) volunteered to take part in the evaluation. All of the designers were designing potentially real products as their final year projects. The projects were at the initial stages of the design process at the point where the design brief had been analysed and researched and the PDS had been written. The participants were told that they could use their projects in the evaluation if they desired but did not have to. The researcher participated in the evaluation as an observer and went through the same process as the participants.

5.2.2 Format and Organisation

Each participant initially went through a set of operating instructions and a tutorial (sample test run) to learn all of the commands for the computer system. The operating instructions initially went through the menu commands to explain each function. This was followed by a description of all the processes in the system, creating and modifying colour concepts, creating and modifying descriptors, teaching colour semiotic relations, generating colour concepts and describing colour concepts.

The tutorial detailed all the anticipated processes that might be performed during the evaluation. The tutorial went through the creation of a colour concept into the current state from a colour image of a surgeon. Descriptors were then allocated to the colour concept (eg. clinical, medical) and input into the current state. The participants then taught the system the colour semiotic relation and generated and described colour concepts. All of this involved using every command in the user interface as well as using the colour palette and keying in descriptors. The operating instructions are listed in appendix D.

The operating instructions concluded with a request for the participants to use the system in a realistic way to teach it colour semiotic relations, generate and describe colour concepts and to then complete the questionnaire. Each participant then spent a minimum of three hours using the Colour Concept Generator, the researcher was present to make observations and answer questions though the participant was encouraged to be as independent as possible. At the end of the session the researcher had a discussion with the participant about what their views were about the Colour Concept Generator. The participant then took the questionnaire away to fill in. The observations, discussions and returned questionnaires were qualitatively analysed together to establish results and draw conclusions for the research.

5.2.3 Questionnaire Design

The questionnaire was composed of nine main parts: (i) learning how to use the system, (ii) operating the system, (iii) creating colour concepts, (iv) creating descriptors, (v) teaching colour concepts, (vi) generating colour concepts, (vii) describing colour concepts, (viii) operating methodology and (ix) the design process. The questionnaire consisted of a total of 81 questions with each main part requesting for any additional comments. Some questions that were asked at the beginning of the questionnaire were repeated later on in more detail to probe for a more in-depth response.

Part (i) learning how to use the system and part (ii) operating the system, were a general overview of grasping the idea and operation of the system regarding creating and modifying colour concepts, creating and modifying descriptors, teaching colour semiotic relations, generating colour concepts and describing colour concepts. The remaining parts were more detailed questions of these areas.

Part (iii) creating colour concepts, was concerned with the origin of colour images and the formulation of colour pie charts. Part (iv) creating descriptors, considered the origin and source from which the words were found (eg. dictionary) and the type of vocabulary used. Part (v) teaching colour concepts, addressed the issues of going from descriptors to colours or vice versa, how objective the process was and translating between media.

Part (vi) generating colour concepts and part (vii) describing colour concepts, investigated the generation and description of colour concepts using single and combined descriptors and colours as well as whether the responses from the system were anticipated or a surprise. Part (viii) operating methodology, examined the process of producing a colour concept through the iterative method of generating colours, describing colours, modifying descriptors, generating colours, describing and so on. Part (ix) the design process, was to do with how the system might be used in a design methodology in the context of designing products. The questionnaire is listed in appendix E.

5.3 Results from the Questionnaires

The following is a breakdown of the main points determined from the analysis of the questionnaires. The order of presentation is the same as the main parts of the questionnaire. Pie charts are used to indicate a visual picture of the proportional weightings of responses. The pie chart weightings were determined by a statistical overview of the participant responses from the questionnaires. The comments made are points that were generally felt by the group after the discussions. A sample of some of the colour concepts generated during the evaluation are shown in appendix F.

5.3.1 Learning and Operating the System

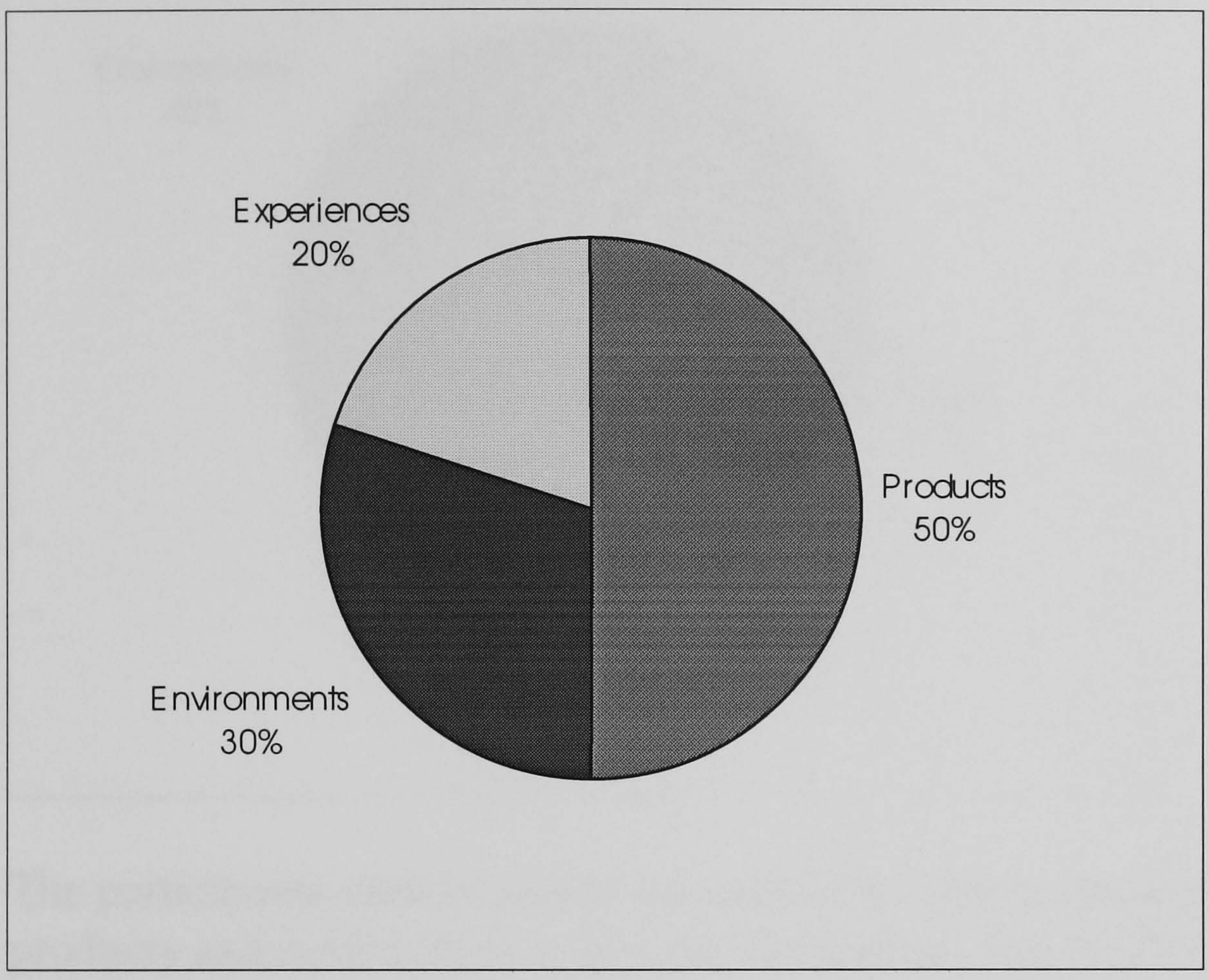
The participants generally found learning how to use the system fairly easy and straight forward, mainly because the instructions took them through all of the processes in the system. They found the colour palette difficult to learn especially finding specific tones or shades of colours. It was generally felt that more colours were required and a full colour palette of all colours or a means of creating a colour by varying the mix of red, green and blue as well as light and dark would be better.

Proportioning colours and descriptors by percentages was found to be confusing especially entering the next input as the actual percentage that it would be in the group. Entering groups of descriptors could prompt / click for finished as opposed to adding one at a time. More prompts were required and a need to differentiate between prompts and screen information. The participants felt the system should be icon driven and descriptors should be selected from a list by clicking the mouse. It was felt that the system was too slow especially at producing the colour concept.

The times taken to learn how to operate the system ranged from the fastest at 30 mins to the slowest at 2 hrs 30 mins. The average time taken was 68 mins. The speed of coming up with ides for colour concepts was varied (some are easier than others) and there was a learning curve (quicker with practice) though some found it quite quick (5 to 10 mins).

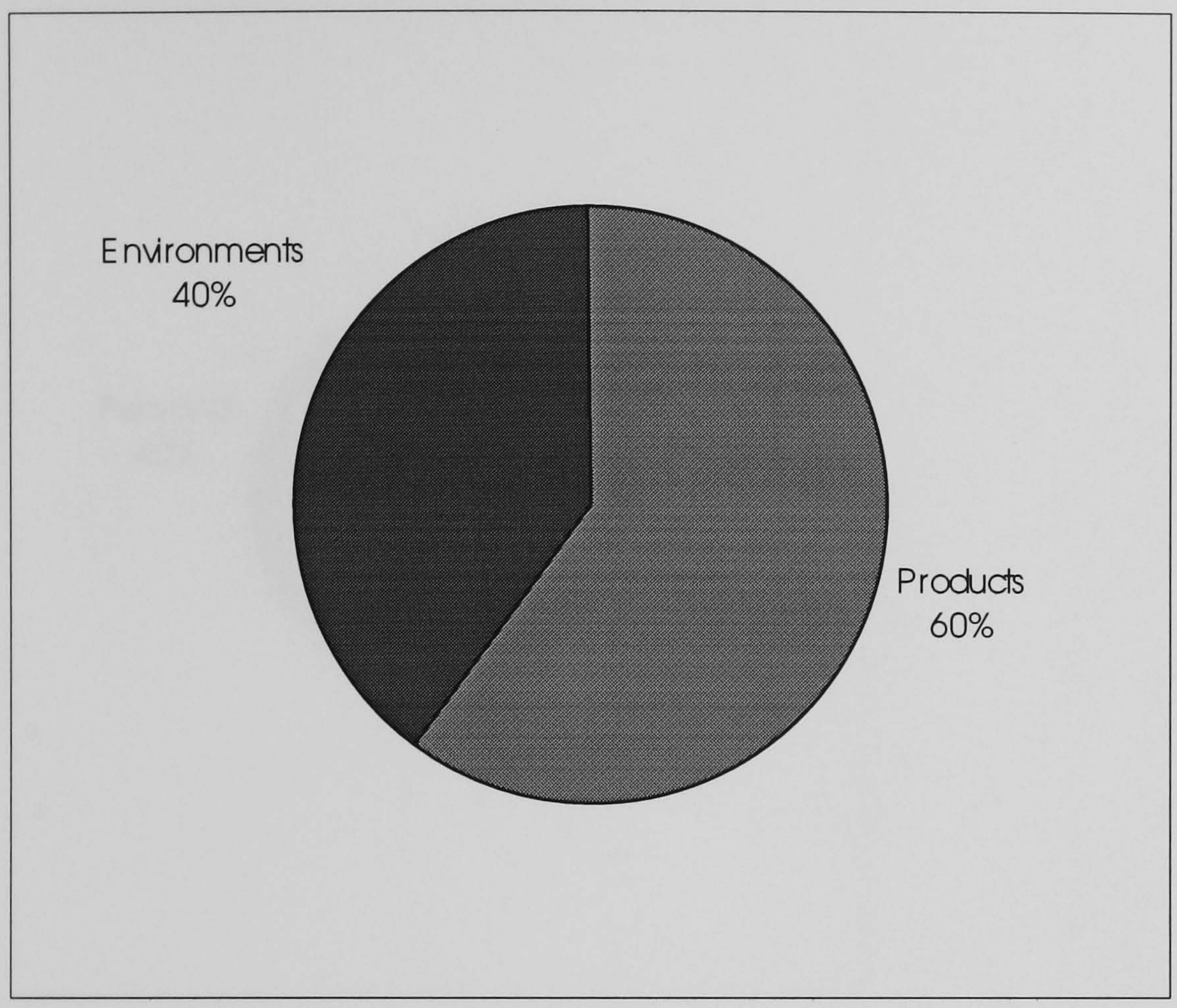
5.3.2 Creating Colour Concepts

5.3.2.1 Imagining Colours in the Mind



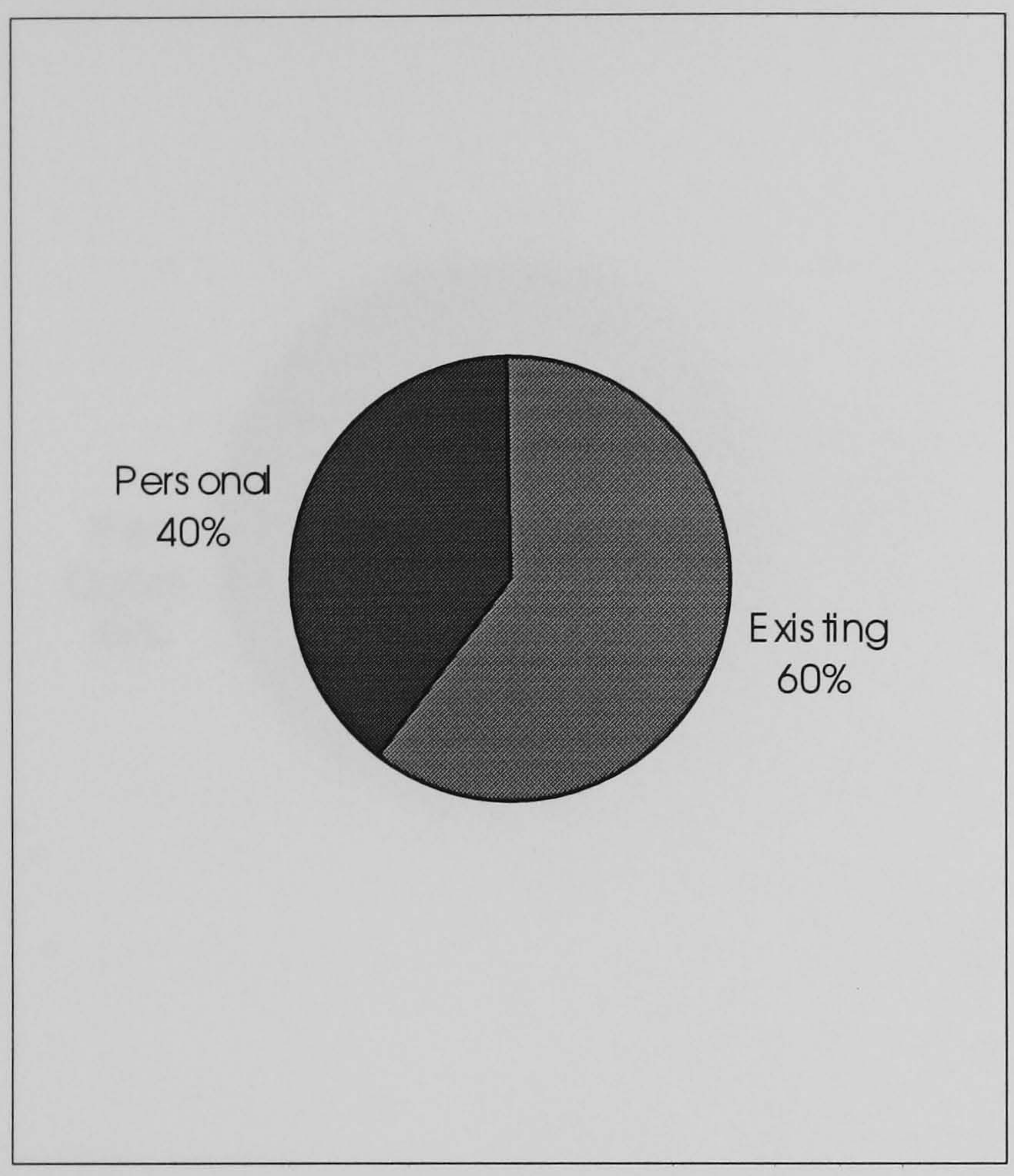
The participants mainly imagined the colours on products, in environments or situations and as personal experiences. The pie chart shows that most of the participants imagined the colours on products.

5.3.2.2 Inspiration for Colour Concepts



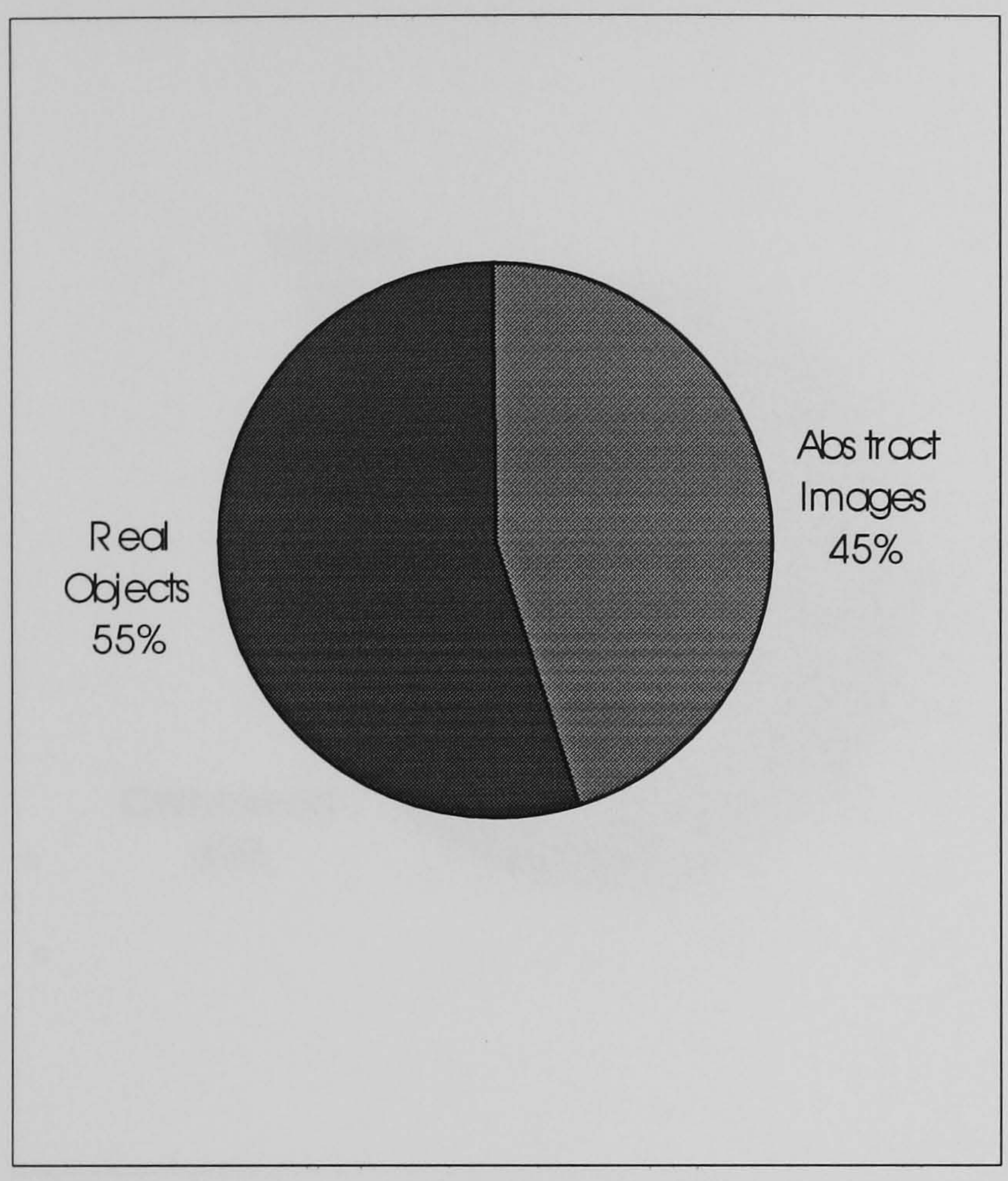
The participants mainly gained inspiration for colour concepts from existing products and environments. The pie chart shows that most inspiration was gained from products.

5.3.2.3 Creating Colour Concepts



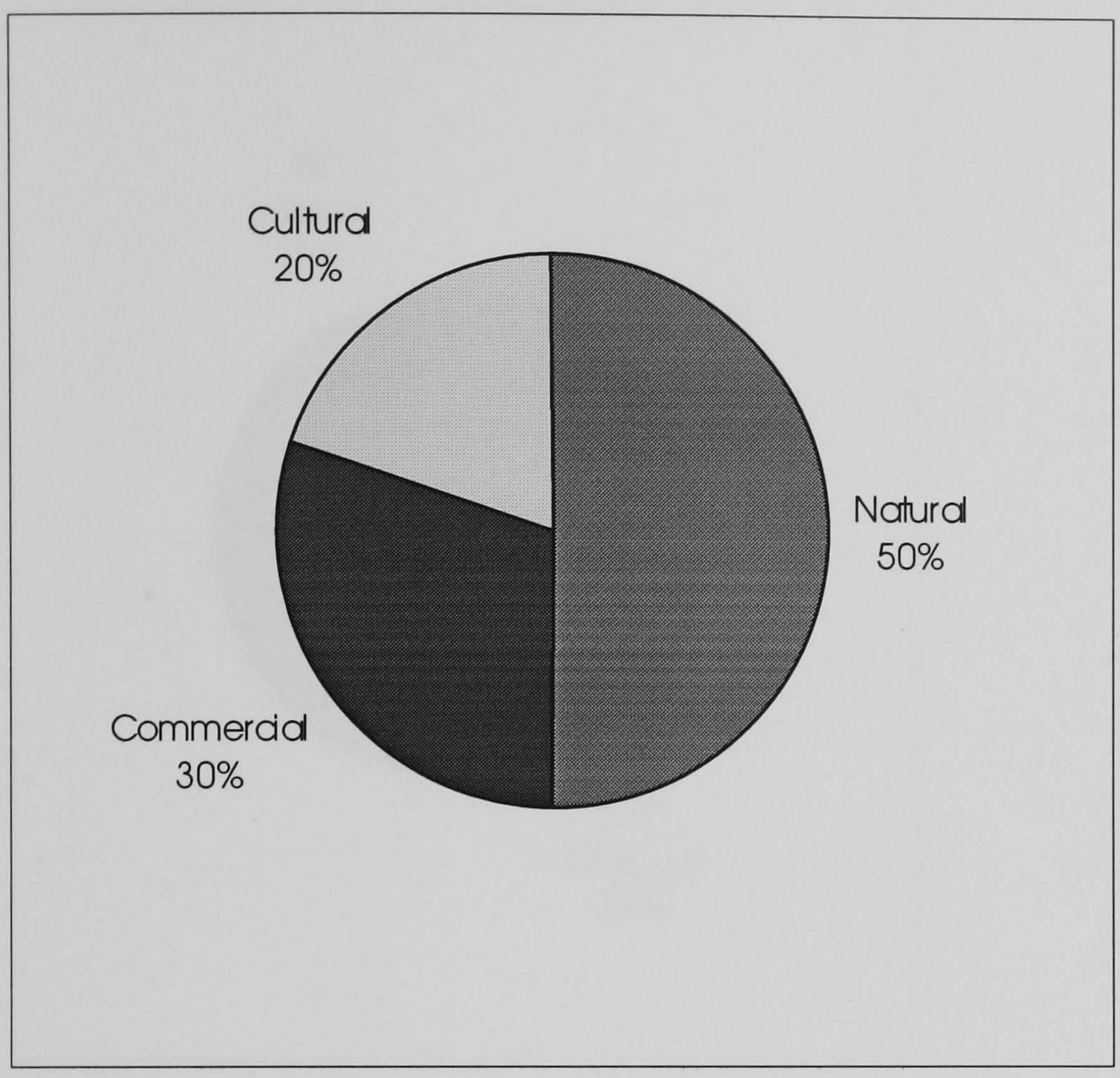
The participants created colour concepts from existing colour images and created personal colour concepts as subjective colour compositions. The pie chart shows that most of the colour concepts were based upon existing colour images.

5.3.2.4 Colour Images for Colour Concepts



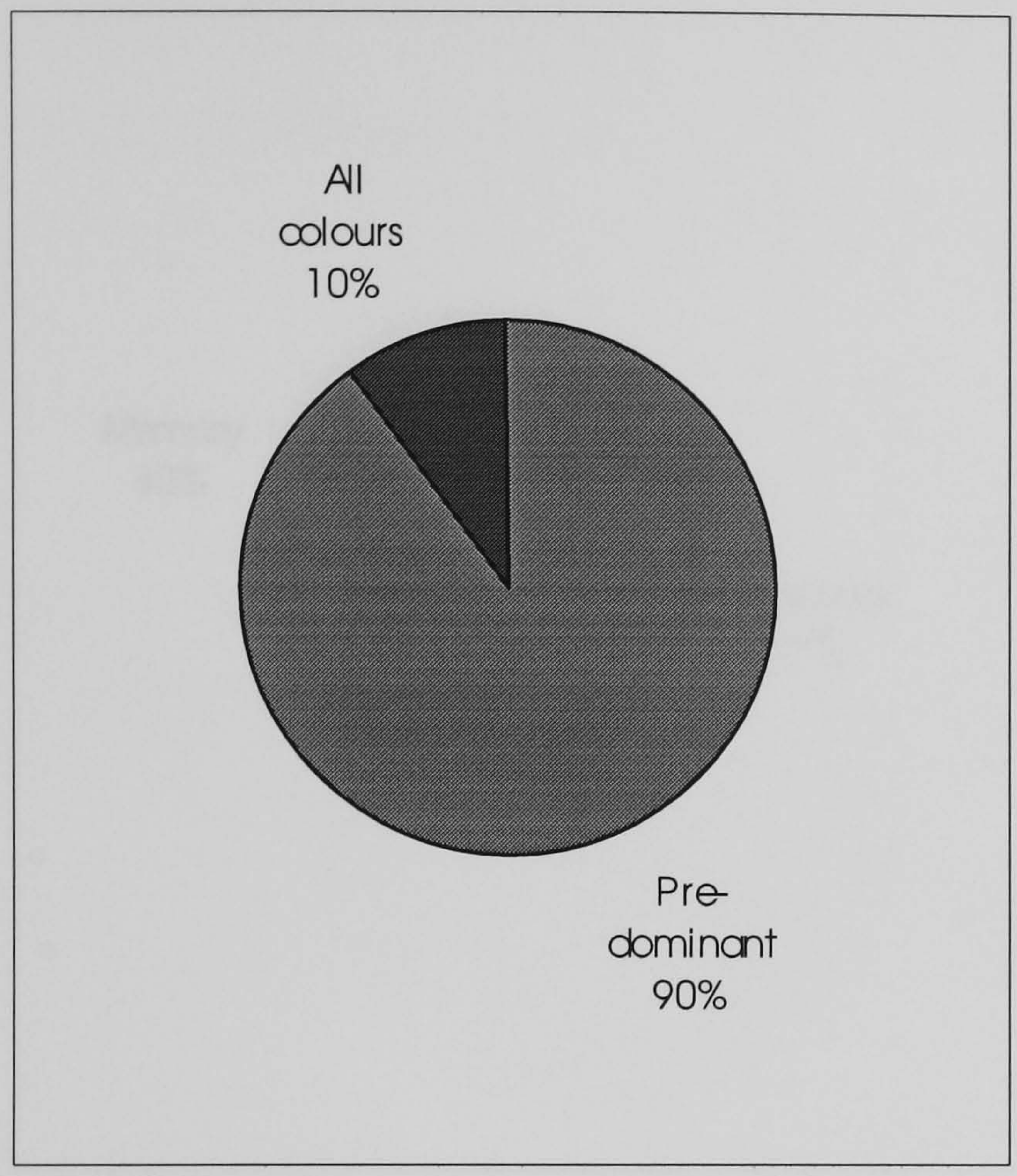
The participants used existing colour images for ideas for colour concepts and were influenced by personal preference in their selection of colour images. Both abstract images and pictures of real objects were used to derive colours though as the pie chart shows, slightly more used pictures of real objects. The participants found the number of colours available was restricting and it was felt that textures for the concepts would be a good idea.

5.3.2.5 Influences Upon Colour Concepts



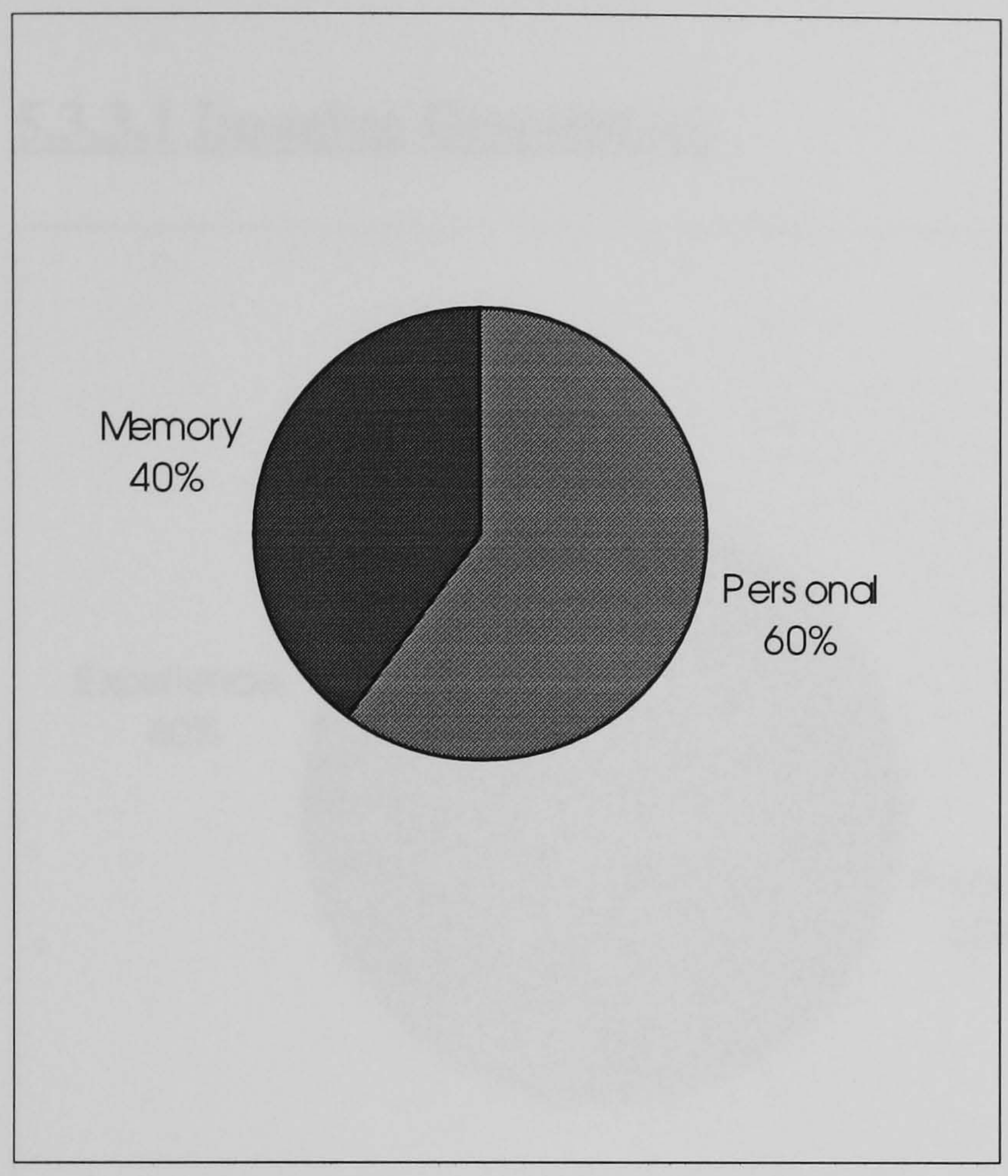
The colour concepts that the participants produced were influenced by natural, cultural (mostly media) and commercial (mostly products) fields. The pie chart shows that most of the colour concepts were influenced by nature.

5.3.2.6 Choosing Colours from a Colour Image for a Colour Concept



The pie chart shows that colour proportions for a colour concept were mostly based upon the more predominant colours in a colour image. Colours were omitted from an image for a colour concept because they were influenced by textures (surface finishes etc.) and some because they were not available (eg. gold and cream). Participants often changed the colour harmony for a colour concept from that of the colour image. Colours were added to a colour concept one at a time and from separate sources. The participants mainly created colour concepts with many colours though some did produce single colours only. Most participants wanted more colours in the system.

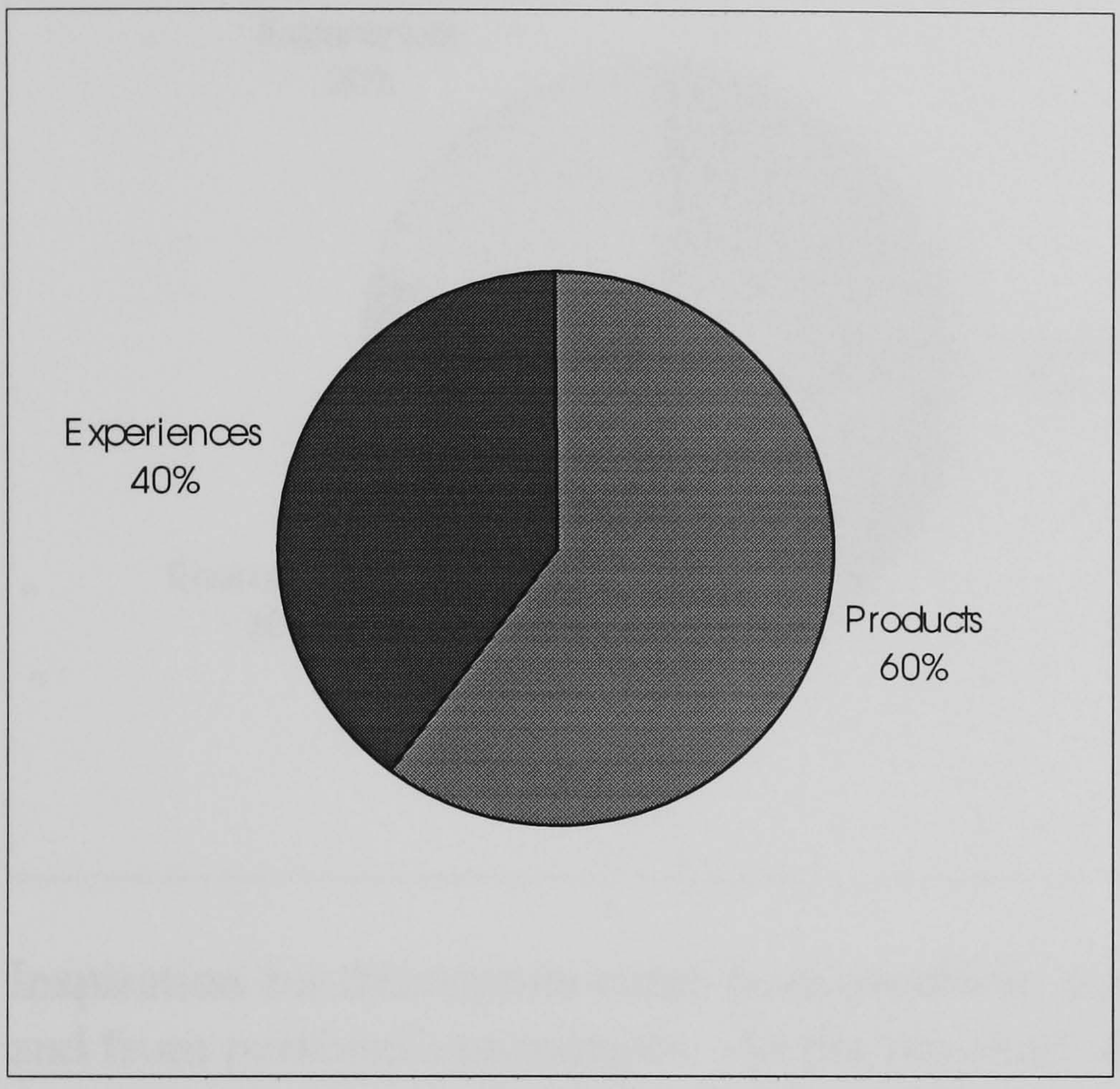
5.3.2.7 Deciding Computer Colours for a Colour Concept



Deciding which computer colours to use for a colour concept was based upon personal taste and memory. As the pie chart shows, personal taste mostly influenced decisions made for the actual computer colours used. Colour proportions were mostly matched by eye on the screen and were based upon the more prominent colours from a colour image. The participants found matching colours on the screen difficult but thought representing colour concepts as pie charts was a good idea.

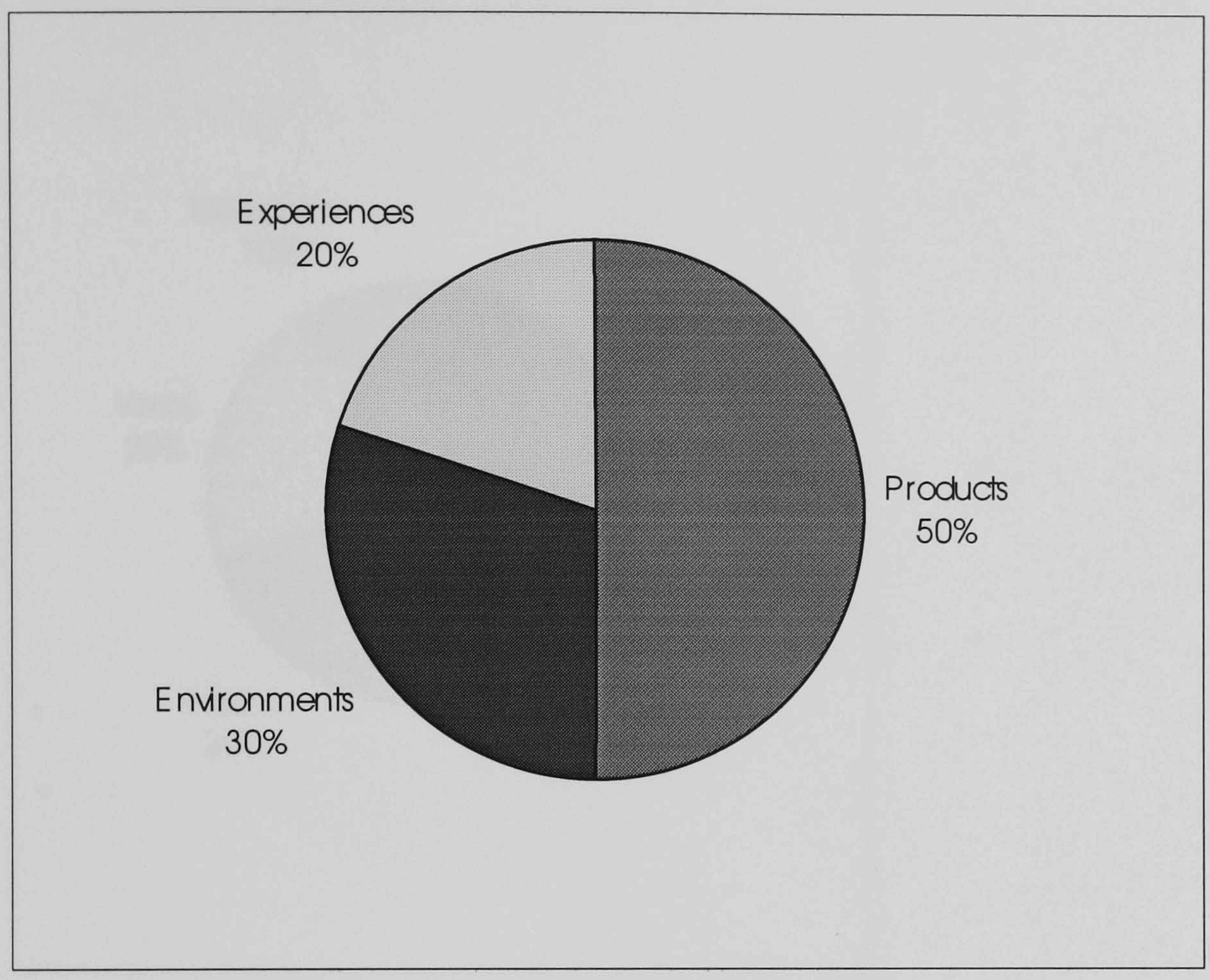
5.3.3 Creating Descriptors

5.3.3.1 Imaging Descriptors



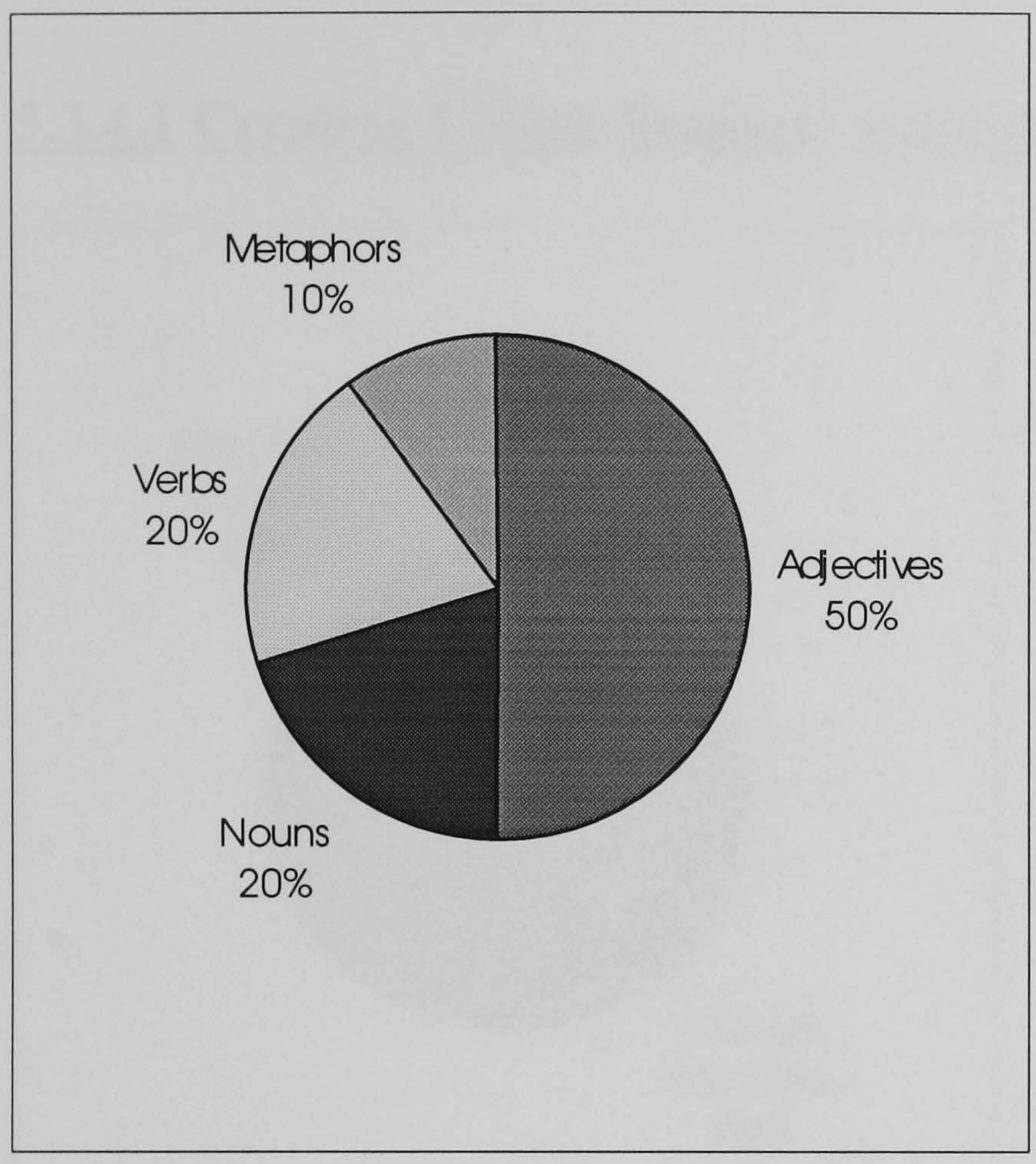
Descriptors were imagined based upon experiences and products. As the pie chart shows, descriptors were mostly imagined on products.

5.3.3.2 Inspiration for Descriptors



Inspiration for descriptors came from products, environments or situations and from personal experiences. As the pie chart shows, inspiration for descriptors came mostly from products. The speed for coming up with descriptors was varied (up to 20 mins) and only a few participants used a dictionary or thesaurus.

5.3.3.3 Descriptor Vocabulary

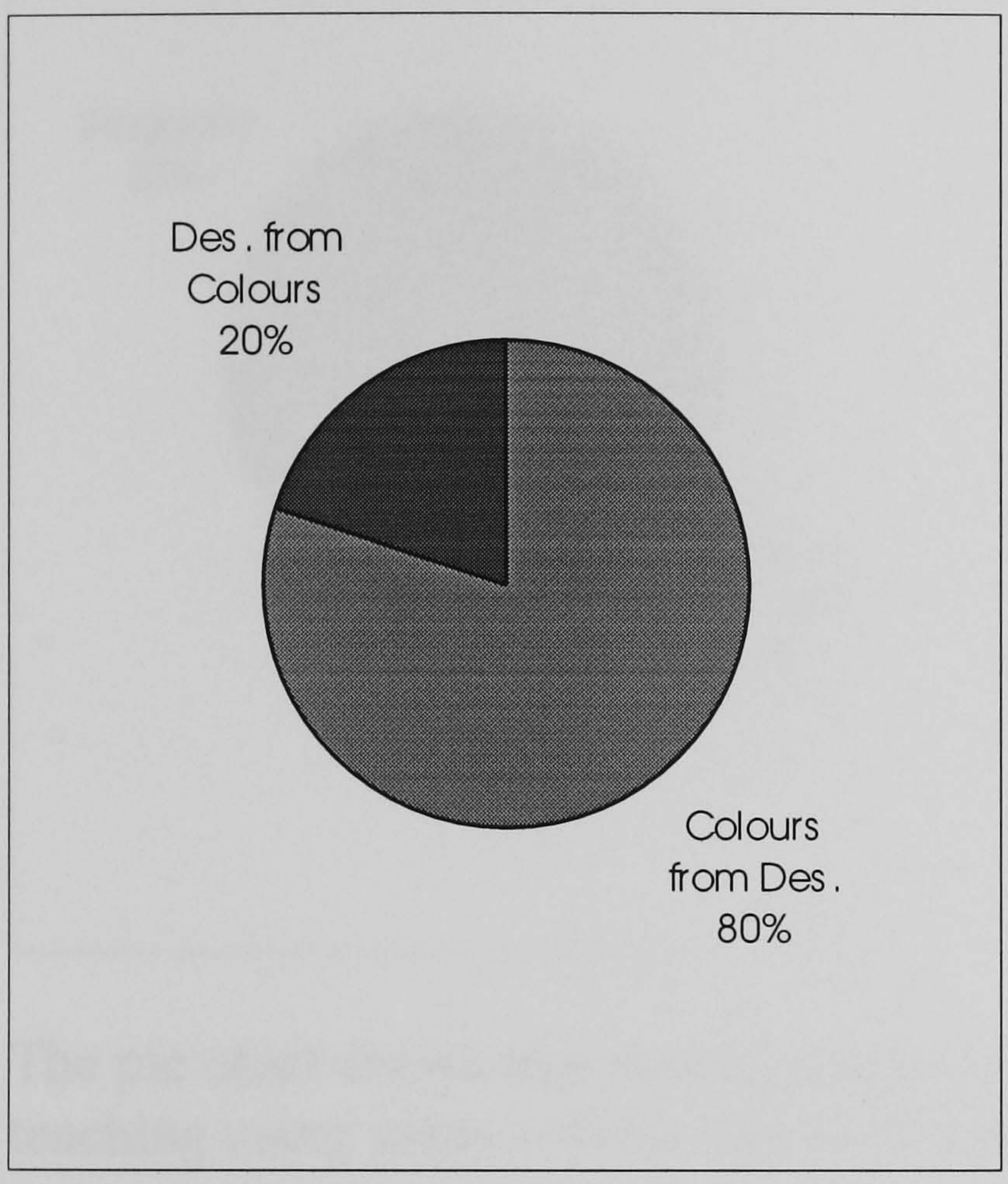


Vocabulary mainly consisted of simple words and the types of words used were adjectives, nouns, verbs and metaphors. As the pie chart shows, the types of words used were mostly adjectives. No participants made up their own words. Most of the participants found it easy to come up with descriptive words and made reference to other colour images when deciding descriptors.

The participants were influenced by personal experience when deciding descriptors and mainly allocated many descriptors to colour concepts, some used single words only. Weighting descriptors was mostly determined randomly as participants found it difficult to allocate weightings to descriptors.

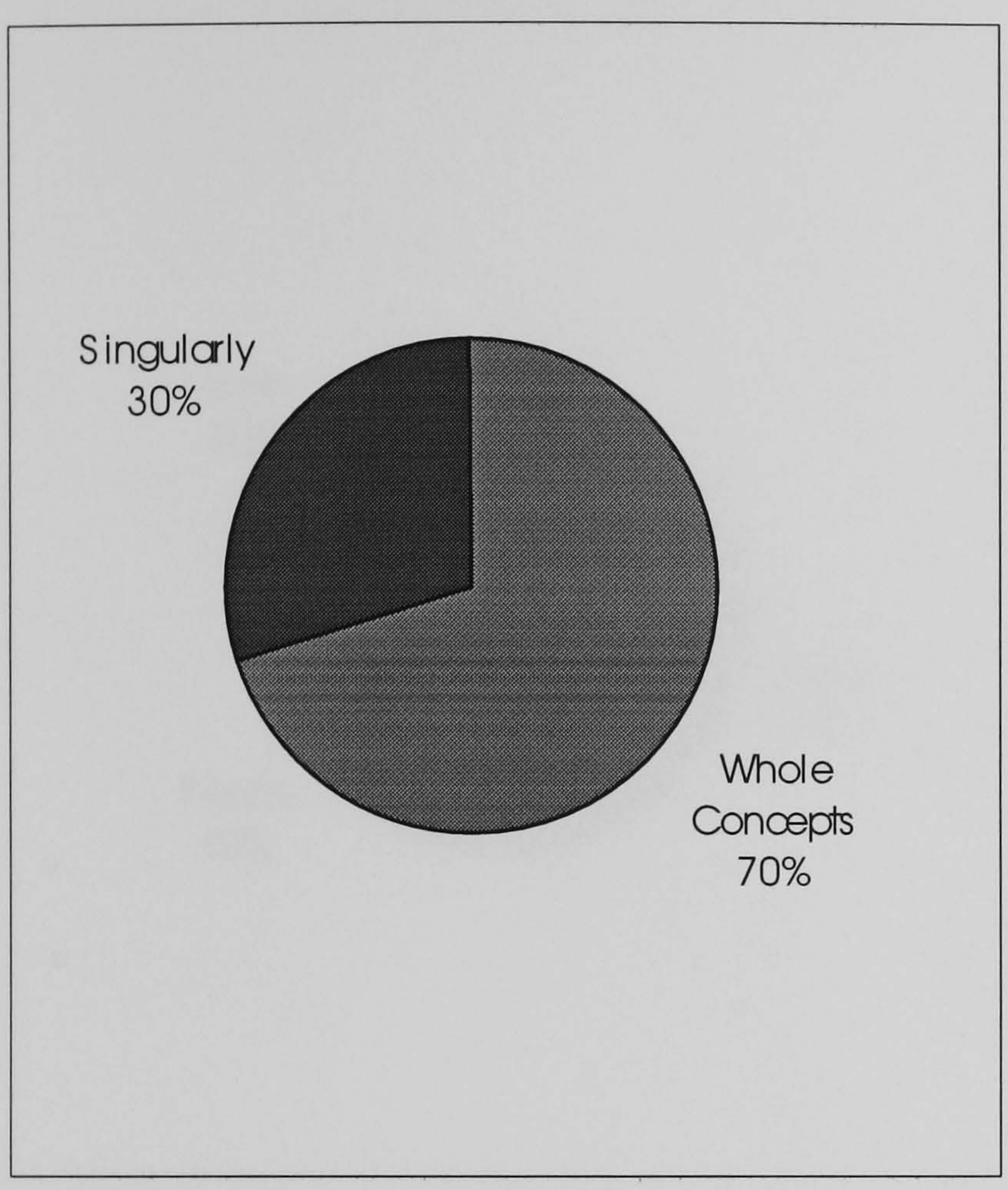
5.3.4 Teaching Colour Concepts

5.3.4.1 Creating Colour Semiotic Relations



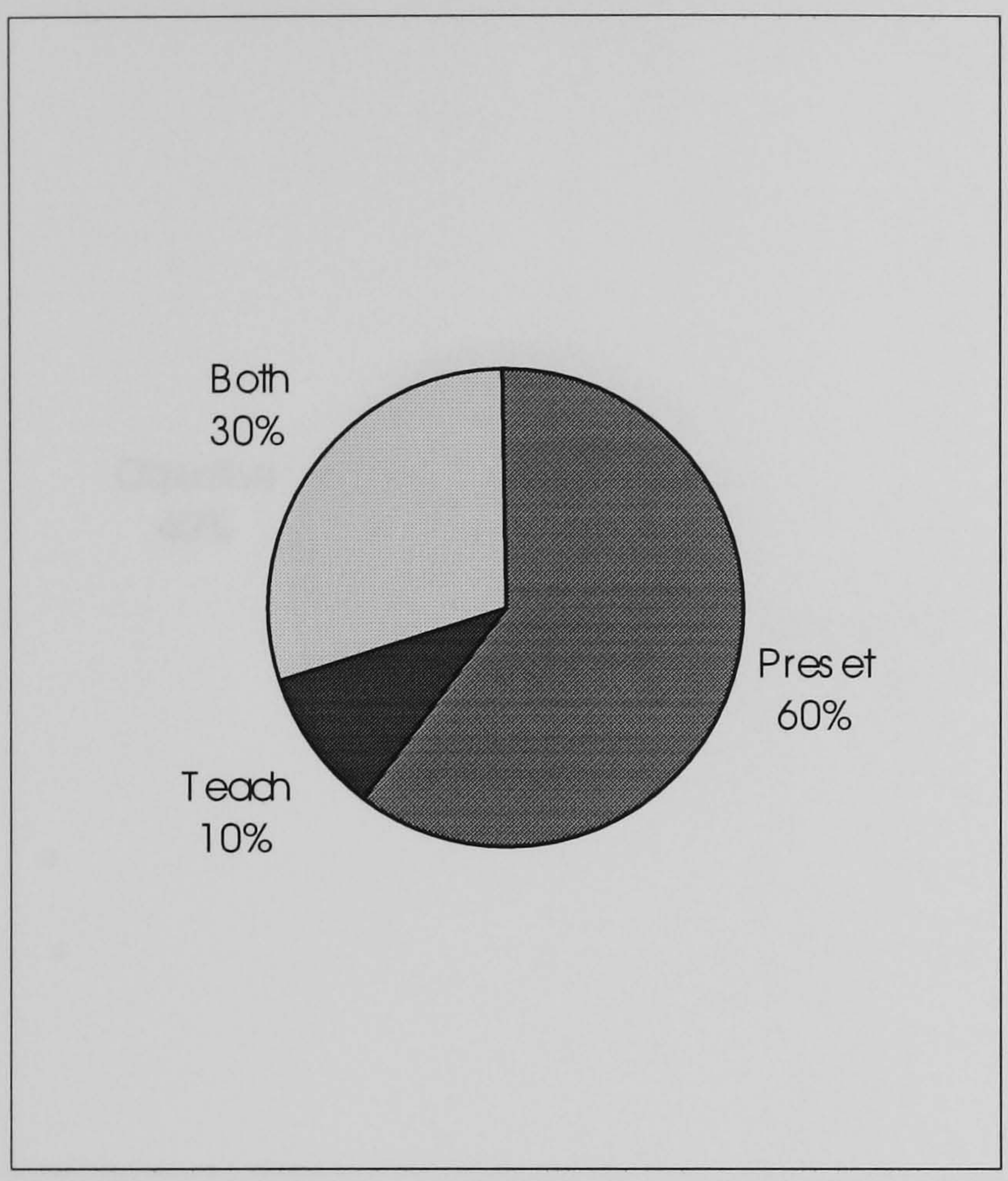
The pie chart shows that most of the participants preferred to create colours from descriptors than descriptors from colours, when creating colour semiotic relations to teach to the system.

5.3.4.2 Teaching Semiotic Concepts



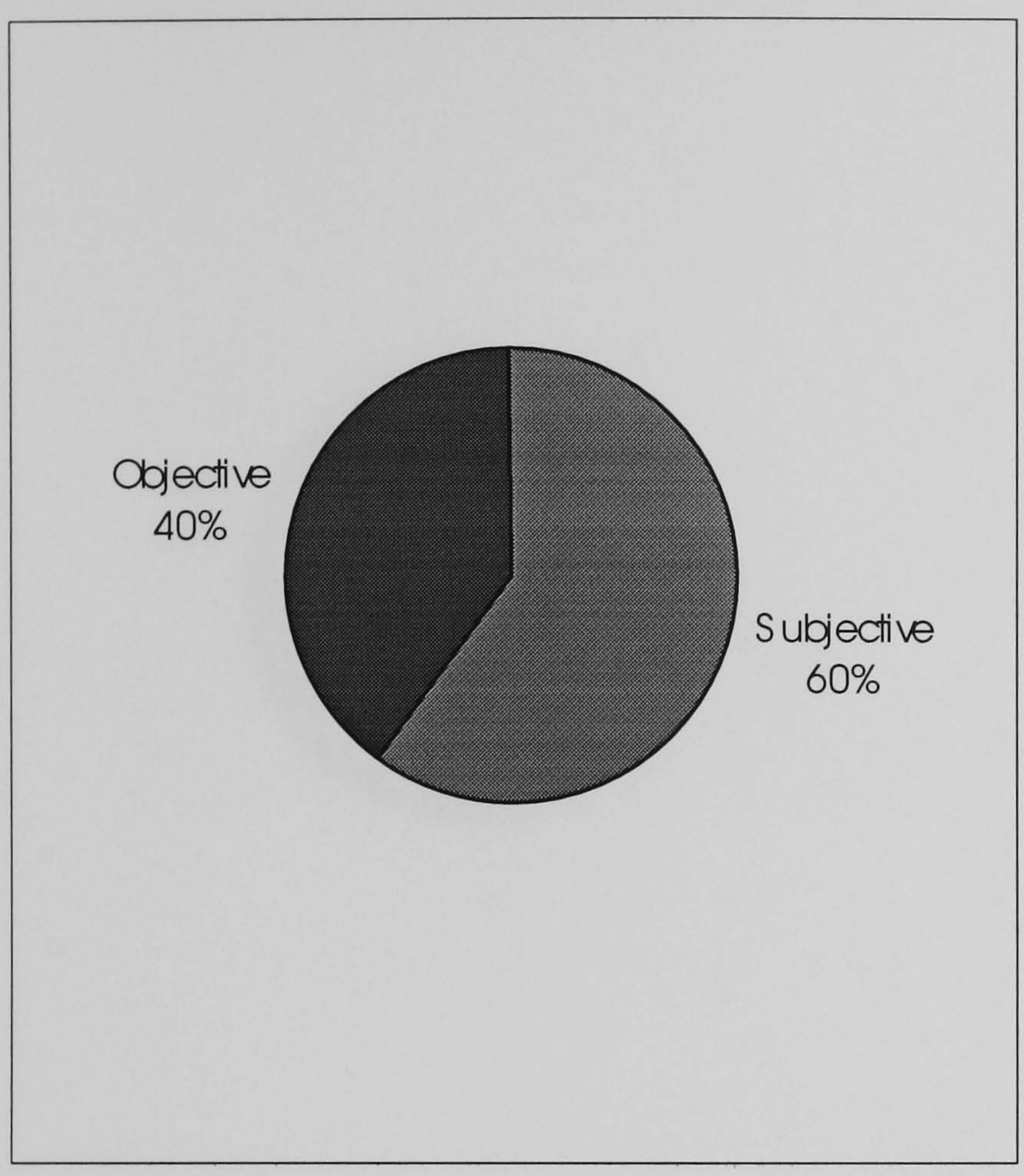
The pie chart shows that most participants produced whole concepts for teaching using many colours and descriptors as opposed to one at a time.

5.3.4.3 Acquisition of System Knowledge



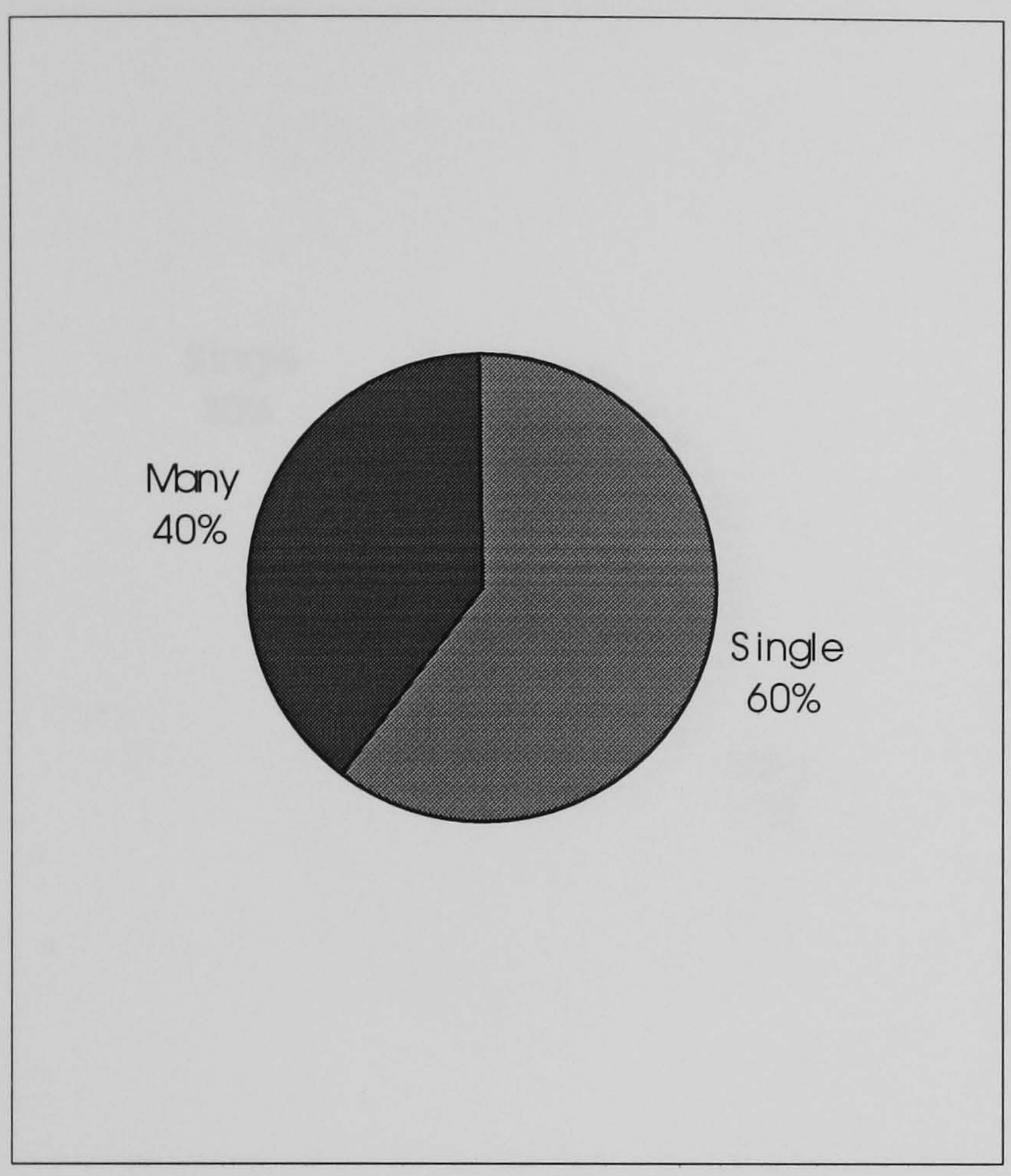
The pie chart shows that most participants preferred pre-set knowledge though some wanted both pre-set and the facility to teach it themselves, only a few wanted to just teach it themselves. The participants thought it was necessary for the system to be subjective and personal but wanted a more widespread database based upon statistical weighting of field tests.

5.3.4.4 Objectivity of the System



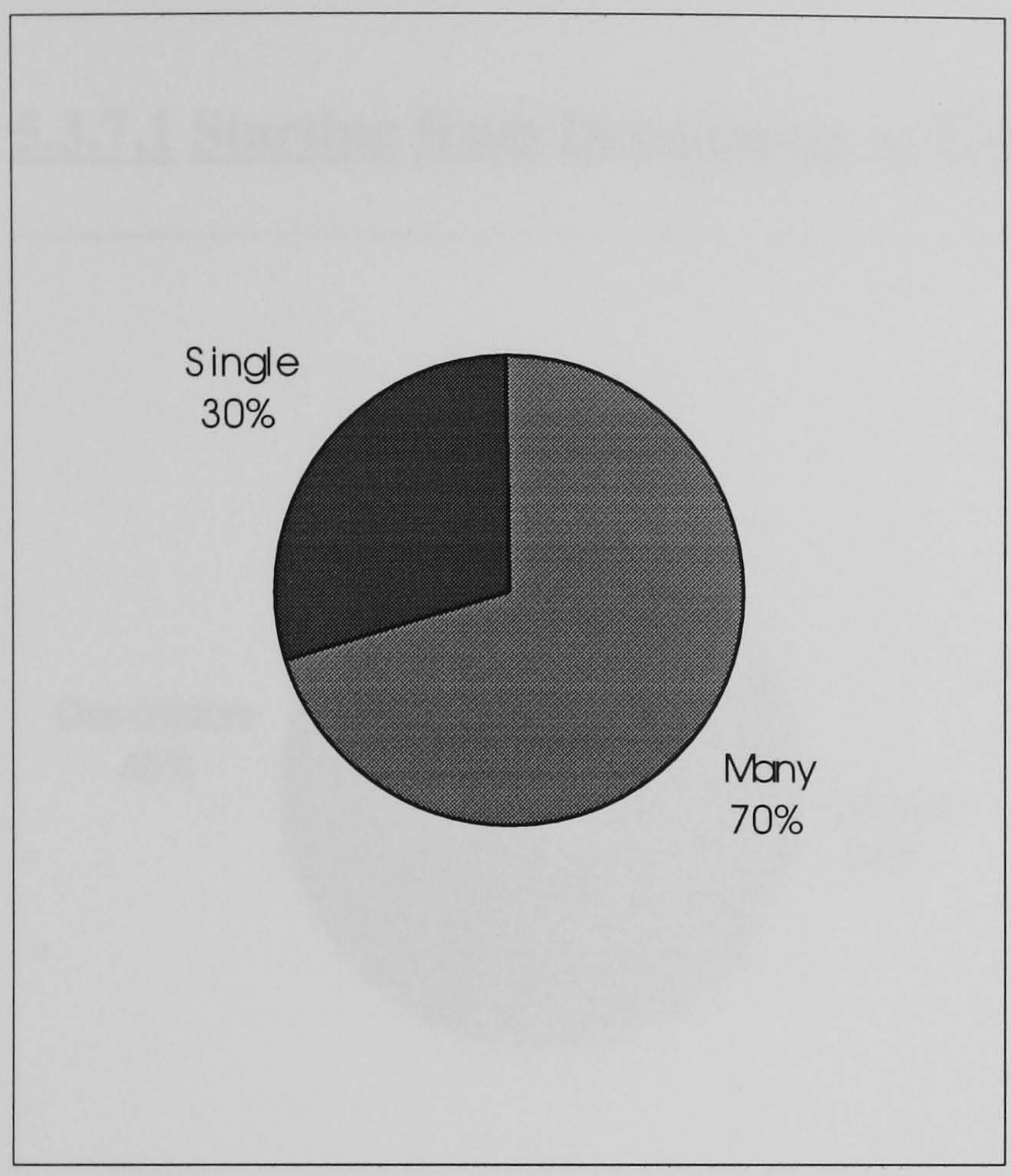
Some participants thought the process was objective, most thought it was subjective but needed a consensus of colour interpretation. Most participants thought it was appropriate to translate between the media of words and colours.

5.3.5 Generating Colour Concepts



The pie chart shows that most participants preferred to generate colours from single descriptors than from many descriptors. The participants found that the visual effects of a generated colour concept changed when the colours were generated in a different order as the balance altered by proximity and proportions of colours. The colours generated were what was expected having taught the system themselves.

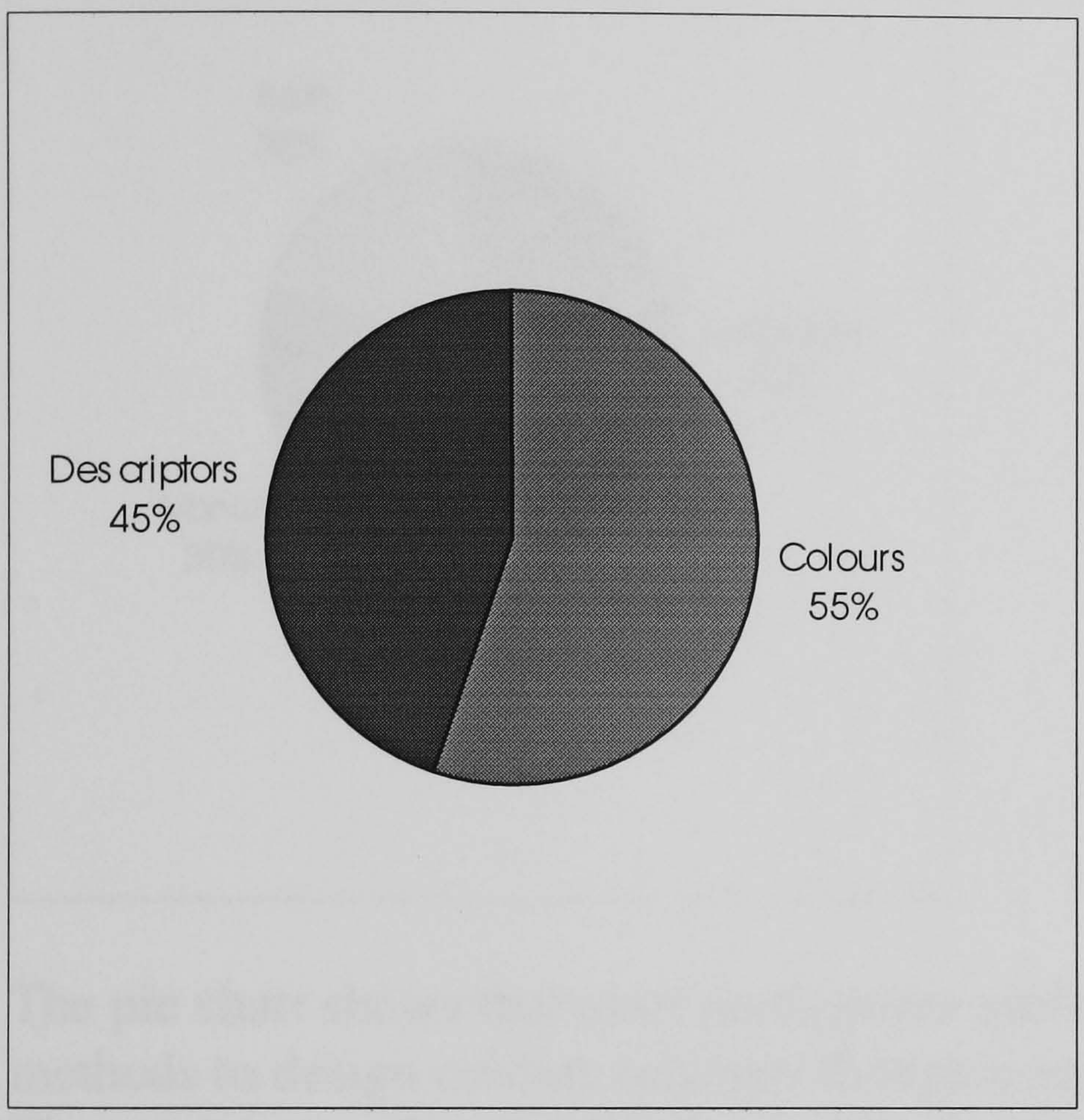
5.3.6 Describing Colour Concepts



The pie chart shows that the participants preferred to describe many colours than single colours. The participants found the describing process helped them to refine the current descriptors, some found this inspired them to re-teach the system. The describing descriptors were what they expected having taught the system themselves.

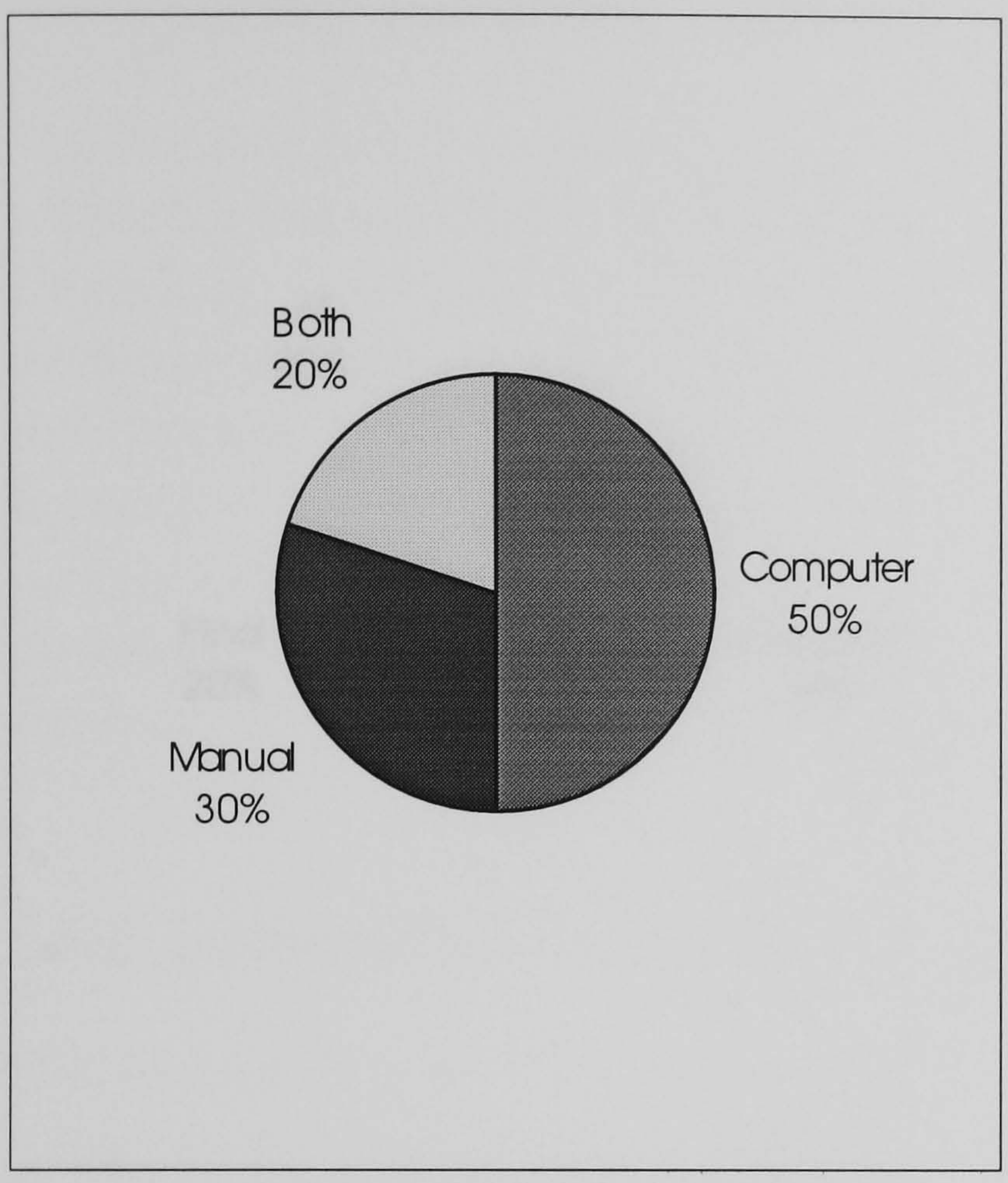
5.3.7 Operating Methodology

5.3.7.1 Starting from Descriptors or Colours to Generate a Concept



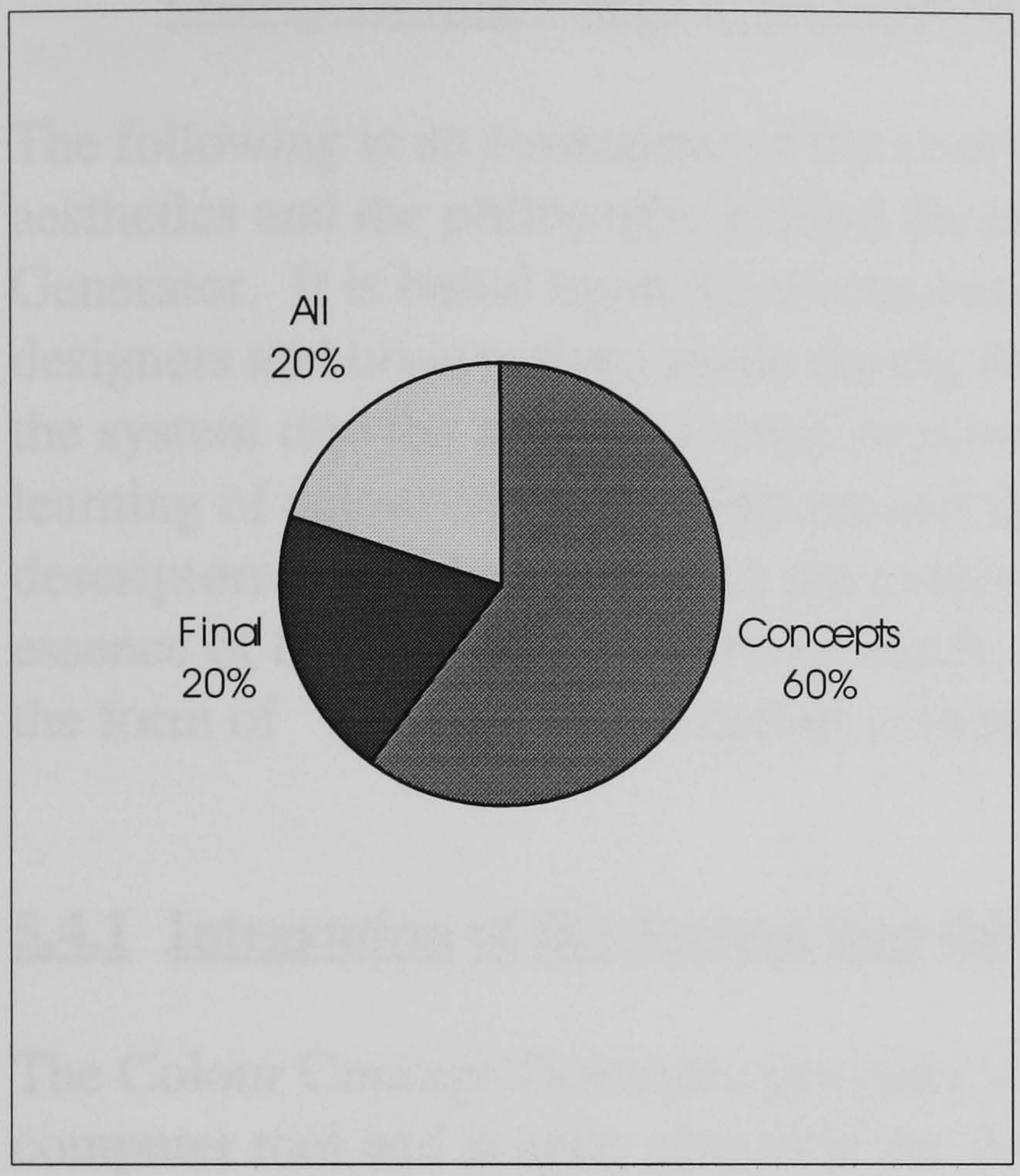
The pie chart shows that slightly more participants preferred to start from colours first to generate a colour concept. They felt the process became automatic after a while and that the iterative process was effective. Most wanted to create a bigger database to establish common descriptor and colour associations.

5.3.7.2 Computer and Manual Methods



The pie chart shows that most participants preferred a computer to manual methods to design colours schemes though some preferred manual and both. The participants thought that the system should include other aesthetic attributes such as form and texture.

5.3.8 The Design Process



The pie chart shows that most of the participants thought the system should be used at the concepts generation stage of the design process. Some participants thought it could be used at all stages and at the final design stage.

The participants thought the system provided an ideas stimulus for ideas for colour schemes (especially their final projects). They thought that the system could be used in a commercial environment (with further development) but more thought it would be suitable as an educational tool. All participants thought that colour was important in product design, mostly as a commercial media to sell the product.

The participants thought that doing the evaluation exercise made them think more deeply about colour and that the issue was more complex than they originally anticipated. The participants thought that a colour scheme could be created before or after the design of a product. Most participants felt that the next stage towards finalising the colour scheme would be to include texture and to integrate the colour concept with product concepts. They thought that the system would boost creativity but as an ideas stimulus as opposed to be creative itself. The participants found the system fun to use and enjoyed the exercise.

5.4 Evaluation of the System Philosophy Based Upon Observations and Discussions

The following is an evaluation of the overall idea of computing colour aesthetics and the philosophy behind the design of the Colour Concept Generator. It is based upon the discussions with the participating product designers and observations made during the evaluation. The integration of the system into the design process, proposed objective criteria for system learning of colour semiotic relations and the translating between aesthetic descriptors and colour concepts are evaluated as these were considered the essence of the philosophy of the research. Reference is made by example in the form of '(eg. ....)' to the colour concepts shown in appendix F.

5.4.1 Integration of the System into the Design Process

The Colour Concept Generator provides a developing, interactive AI computer tool and is aptly placed at the front end of the design process. Colour concepts depict an abstract representation of a colour aesthetic character for a commercial product (eg. any of the colour concepts can be imagined as a potential colour scheme for any product). The process promotes an experimental and explorative approach as opposed to a prescriptive approach as the system is an interactive medium capable of suggesting ideas as opposed to dictatorial specification. So the philosophy of using a computer system as an ideas stimulus for colour concepts at the front end of the design process is feasible providing it is in line with the ethos of generating ideas for concepts in general.

The front-end computing of colour concepts can act as a filtering process to precede research for consumer preference or interpretation of product colour. The idea of generating colour concepts at the initial stages of the design process allows for a refinement of numerous colour concepts into a select few (eg. take a selection of the colour concepts to test on consumers) deemed most suitable through interaction with the computer system.

The idea of computing colour aesthetics alone limits the aesthetic criteria to a single medium. Product aesthetics are a composition of colour, form and texture so generating colour concepts in isolation might be defeating the integrated intentions of product design. The colour concepts could be more broadly defined as aesthetic concepts and include the effects of texture, surface finish and light reflection characteristics to increase the numerous possible combinations of aesthetic media.

5.4.2 Objective Criteria for System Learning of Colour Semiotics

The description of colour images through symbolic interpretation can provide an objective criteria for colour semiotics as descriptor allocation becomes justified by reference to an existing entity (eg. egg shell, coral, wheat, wasp). Colour semiotics can vary according to the cultural context (eg. priest) so learning colour semiotic relations can include contextual specific words in the associated descriptors. The context in which the generated colour concept was to be used could then be included in the generating descriptors.

The selection of colour images can create a perfectionist tendency in the search for suitable images so the level of quality of colour image selection increases with experience. Some images stand out as typical icons or even stereotypes. A filtering of image selection through human intervention and evaluation occurs as selecting colour images promotes quality of human learning as much as system learning.

Natural images were more commonly used and are readily available (eg. ocean, desert) yet difficult to expand the description as contradictions can occur between describing the colours and describing the object (eg. eggshell). Commercial images provide well defined and specific colour concepts (eg. neon) and obvious image and style attributes. Cultural images tend to be concentrated in occupations and cultural roles (eg. surgeon, astronaut, executive, priest).

Personal and subjective colour images allow for freedom of expression and progression into abstract descriptors of a more poetic and metaphoric nature. There can be a liberation from being restricted to real images, although real images do provide a stimulating inspiration for personal images. An integrated creative approach, using real images for inspiration allows for subjective colour expression with an objective reference. Ideas for personal colour images inspired by other images often preceded seeing the image resulting in a tendency to rely upon colour memory.

Creating colour concepts as a composition from a descriptive word using colour images for inspiration can be less restricting than describing colour concepts derived from images. A creative approach was generally more natural to product designers than an analytical one. Creating a colour concept from descriptors can result in each colour concept becoming a project in its own way, thinking can be more concentrated and the search for associated images more directed (eg. blood, wasp).

Colour image matching on the screen by eye proved difficult in accounting for effects of texture and light as the colours in images of identical objects varied in different pictures, resulting in selecting computer colours that were a nearest match (eg. watermelon). The perceived colours in images were sometimes not the actual colours but instead a result of simultaneous contrast and proportionate mixing in the eye or brain, influenced also by viewing distance as closer views of an image resulted in more colours being perceived.

5.4.3 Translating Between Aesthetic Descriptors and Colour Concepts

Translating colour concepts into descriptors and vice-versa proved to be one of relating objects whereby the descriptors become labels of colour aesthetic image and style themes. By basing colour concept generation upon translating descriptors through a semiotic software mechanism, new colour concepts can be created through the combination of previously learnt colour concepts. The integration of the colour semiotic knowledge allows for colour aesthetic associations to develop that had not previously been directly learnt by the system.

The contextual aspect of colour images proved problematic depending upon the depth of understanding and periodic perspective. Images might be interpreted from the present day or in light of the contextual issues at the time of origin (eg. priest). The system could become a kind of encyclopaedia of colour semiotics to provide expressive aesthetic descriptors that are applicable to products. The interpretation of the colours and descriptors is at the discretion of the product designer who can be involved in the learning process as well as translating process.

The colour concept pie-charts visualise into abstract pictorial compositions as aesthetic image themes as opposed to fragmented colours. Colour concepts tend to appear as exaggerated 'characters' of colour images and some seem complete with only the denoting noun as a descriptor, rather like a picture title. The colour concepts look as divorced and abstract from the original image as they are indicative of it, ie. as well as holding together as an individual visual composition the association through the descriptors to the original image is still prevalent.

Temptation to omit certain colours present in a colour image from the colour concept can occur due to a bias imposed by the desired descriptors chosen to express the colours. The symbolic descriptor association might be based upon colours in an image of an object that are most typical or in greater proportion. Allowing for interpretation by the product designer provides for a refined translation based upon human intervention.

Colour images of edible objects (eg. watermelon) cause a tendency to create colour concepts and descriptors more suited to the semiotics of consuming the object. This is characteristic of images of objects that might involve an element of human interaction that affects senses other than vision. A tendency towards allocating descriptors based upon feelings encountered in experiencing the object (eg. iceberg) occurs, which in some cases do seem to overlap with the perception of the image when viewed in abstract isolation.

A dictionary and a thesaurus were useful aids for expanding the descriptors, especially in the allocation of common descriptors across different colour concepts and subsequent colour rules. Allocating descriptors to colour concepts by symbolic association allowed for objectivity and creating personal colour concepts allowed for freedom of expression. The combination of both allowed for a balanced approach.

5.5 Evaluation of the System Methodology Based Upon Observations and Discussions

The following is an evaluation of the methodology used in the Colour Concept Generator. It is based upon the discussions with the participating product designers and observations made during the evaluation. The learning, generating and describing of colour concepts are evaluated as these were considered to be the main processes in the computer system.

5.5.1 Learning Colour Semiotic Relations

The development of the knowledge base operates in accordance with the anticipated process of allocating descriptors and weightings to descriptor sets in colour rules. The learning process proportions rule descriptor weightings through proportional matrix multiplication of current colour sets and current descriptor sets and maintains a working average of descriptor weightings in the rule descriptor sets.

The rule descriptor weightings reduce in proportion as more descriptors are allocated to colour rules, subsequently reducing the effective weighting of each colour and descriptor as new concepts are adopted. This can be difficult to accept as colours and descriptors belonging to concepts that have been allocated many colours and descriptors lose proportional weighting even though they are synonymous with each other.

However the basis of the learning process is to proportionally weight semiotic relations, so if a colour or descriptor has many other connotations then potential candidacy for a particular application is limited by alternative possibilities, ie. breadth of feasibility limits specific feasibility.

The learning process takes the long term view of absolute knowledge of all colour semiotics with the intention of integrity increasing as more semiotic relations are learnt as opposed to the short term view of specific knowledge being learnt in isolation. The semiotic knowledge is learnt as a whole body of integrated knowledge as opposed to a list of facts having no connection with each other.

Where colour concepts were found to have many descriptors, the descriptors were grouped according to semantic relation and presented to the system in groups with each group being proportioned out of 100 percent. Although the proportional significance of descriptors still reduced, the logistics problem of weighting many descriptors is alleviated.

If a descriptor was omitted from a description when a semiotic relation was learnt by the system it was complicated to then allocate that descriptor with an accurate proportion. The colour concept can be recalled by generating from one of the previously allocated descriptors and the associated descriptors recalled from the colour concept.

However there is no indication of the learnt proportional weightings without entering the rulebase file and manually calculating the original proportions, erasing the descriptors and weightings from the rulebase and redoing the learn process with the omitted descriptor included. If the omitted descriptor is presented alone with the colour concept then it automatically has a weighting of 100 percent, ie. equivalent to the denoting noun. This questions the dynamic nature of the learning mechanism as it assumes that a semiotic relation is complete upon every acquisition of the learn process.

5.5.2 Generating Colour Concepts from Aesthetic Descriptors

The generating process operates in accordance with the anticipated process of allocating colours to descriptors by matching current descriptors with rule descriptors. The colour weightings are accurately calculated by summing the multiples of the matched current descriptor weightings and the rule descriptor working averages for each colour.

Generating a colour concept from a single descriptor produces a colour concept that contains all colours that include the descriptor in the colour rule descriptor set. However the proportional areas of the colours do not vary according to the descriptor weighting as a single descriptor is fixed at 100 percent, so variability is denied for a single descriptor.

If a colour concept has many colours and many descriptors during the learn acquisition, all of the colours will be generated from any one of the associated descriptors. However the colour proportions will be the same for each descriptor (irrespective of its learnt weighting) when singularly used to generate the colours. Combining any of the descriptors in varying proportions still results in the same colours and proportions generated. This indicates that the matrix multiplication process does not alter proportions when generating colours from descriptors used with the same colours in the same learning acquisition. So emphasis by weighting does not apply for descriptors and colours contained within a single learn acquisition.

Increasing the number of descriptors generally results in a reduction of the number of colours in the generated concept as a more specific description tends to refine matching of colour rules. However removal of a single descriptor from the current descriptors with a subsequent generation can expand the number of colours in the concept. The removed descriptor might have been 'blocking' the execution of colour rules due to it not being a member of the colour rule descriptors as the system only selects colours with rules containing all of the current descriptors.

The generating process does not allow for combining semiotic relations that have been adopted in isolation, ie. opposite or conflicting descriptors generate no colours. It proves impossible to accumulate colours in a concept by combining descriptors relating to separate colour concepts as the system will only generate colour concepts from groups of descriptors that all fall in at least one colour rule.

The process is logically correct as for an intelligent system to respond to a combined descriptor meaning it must only generate colours that hold that same meaning. If the generation process were accumulative then the long term view of absolute semiotic knowledge would result in a generation of many colours from only a few descriptors. The generating process must be based upon refinement as opposed to expansion regarding specific semiotic definition.

However as a design tool the refinement method could appear restricting as it might be the case that descriptors unrelated in the knowledge base might be specified as the required aesthetic image and style attributes. In this case the system is limited in creating new colour concepts from current semiotic knowledge. There is the option of providing both refined and expansive methods of generation, ie. 'and' and 'or' Boolean inference associations respectively. Although common descriptors will result in the generation of many colours in the case of expansion.

When a colour concept is generated from a single descriptor that was included in the learning of more than one colour semiotic relation, the proportional area of colours in the concept is influenced by the number of descriptors allocated to each semiotic relation, ie. the generating descriptor has a different weighting in each isolated colour concept. For example, if three concepts were adopted by the system separately and each concept contained a common descriptor, the individual concepts do not occupy a third of the colour concept generated from the descriptor.

5.5.3 Describing Colour Concepts

The describing process operates in accordance with the anticipated process of allocating descriptors to colour combinations by matching current descriptors with colour rule descriptors. The descriptors are produced by selecting descriptors that only fall in all of the colour rules for the colours in the concept.

A colour concept description contains the same descriptors that were used to generate the colour concept as the system uses the same logic to describe the colours as it does to generate them, ie. a refinement based upon a Boolean 'and' association, so maintaining a semiotic consistency. Regarding the response of an intelligent system this is logically correct in terms of specific semiotic association.

However the descriptor list lends little insight into possible expansion of the descriptors until a colour is removed from the concept and the new concept is described. From the point of view of a design tool the generating problem of refinement or expansion is also found in the describing process. So provision of an option facility based upon Boolean 'and' and 'or' inference association methods for both describing and generating colour concepts is possible. Although common colours will result in the production of many descriptors in the case of expansion.

If a colour is removed from the concept and a list of descriptors produced, the descriptors might alter by expansion. This is the result of a single colour 'blocking' the matching of other descriptors. Although all other colours in the concept might have many descriptors in common, the system will only select descriptors common to all colours. Therefore removal of a colour with few descriptors in common with all the other colours results in descriptor expansion. The situation is the same as removing descriptors for generating a colour concept.

The alteration of the describing descriptors might be the result of removing a colour from an identical colour concept to one that had been originally learnt. The remaining colours are still regarded by the system as being indicative of the original learnt descriptors as well as any other semiotic learning in which they have been placed as combined colours. This questions the integrity of the system in limiting the descriptor response if all the descriptor related colours are not present in the colour concept.

Describing a single colour produces a list of all the colour rule descriptors associated with the colour, all the possible connotations are presented. If the starting point for generating a colour concept commences with describing a single colour, the refinement reduces the describing descriptors as more colours are added. This is the result of additional colours 'blocking' the matching of descriptors as descriptors are only selected if they are associated with all of the colours in the concept.

The descriptors produced for a colour concept are the same regardless of the proportions of the colours. The describing mechanism matches descriptors to colours by colour and not by proportion so a lot of pink with a little blue is described by the same descriptors as a lot of blue with a little pink. This indicates a failure in the integrity of the system to differentiate in specifically describing colour concepts of differing proportions. Although this was never the original intention it is an aspect characteristic of the describing process.

5.6 Evaluation of the Computer Media Based Upon Observations and Discussions

The following is an evaluation of the computer media used in the design of the Colour Concept Generator. It is based upon the discussions with the participating product designers and observations made during the evaluation. The use of production system and AutoLisp are evaluated as these were the tools used in the design and development of the computer system.

5.6.1 Production System

The application of a production system in the computer system prototype proved to be a feasible method for computer learning of colour semiotic relations and producing colour concepts from verbal descriptors. From process and data design to program coding, the organisation and structure of a production system lends a flexible domain in which to build an AI computer tool.

A production system is conducive by methodology to the process of creating colour concepts as it is ultimately concerned with producing an inferred response to a pre-defined situation which might have been a previously inferred response. The cyclic iteration is amenable to the design of an interactive system whereby process looping can be interrupted and current situations modified by the user.

By fixing the number of rules in the computer system rulebase to the number of colours available compared to expanding as new semiotic relations are adopted, the knowledge is continually revised and updated as a whole rather than accumulating pro rata. In this sense colour and descriptor relations are contained in an integrated body of knowledge as opposed to isolated facts. Also a 'combinatorial explosion' is prevented as infinite looping of rule firing is not possible.

The database provides a short-term memory of the current situation regarding a semiotic relation to be learnt or one that has been generated or described. Accessing the database via the interface provides a 'window' at a suitable point for user reference and for manipulating colours and descriptors. All processing revolves around manipulating database elements through semiotic inference so placing the database at the centre as an intermediate between the user and the system as a whole.

The inference engines are a key strategic mechanism for optimisation of the integrity of the system and are the focal point of artificial intelligence in the computer system. The forward and backward chaining allows for generating and describing colour concepts as rules can be matched by descriptors or colours. However in producing colours and weightings through proportional matrix multiplication by forward chaining from descriptors and weightings it was observed that the reverse was not possible.

To reverse the matrix weightings from calculated colour weighting results is to assume that the original descriptor weightings from which the colour weightings were determined can be calculated. However this was found to be impossible as identical colour weightings can be produced from different descriptor weightings through matrix multiplication even if colours and descriptors are known.

The inability to solve this problem is partly indicative of why the describe inference engine only backward chains a set of associated descriptors as opposed to reproducing the current descriptors and weightings. If the generating process was to be exactly reversed then repetitive generating and describing would eventually result in no change to either the colours or descriptors, each would be the result of the other.

The user interface proved to be more suitable as an acquisition interface for an experimental domain than a user-friendly display. Colours on the screen do not easily match to colours in images limiting colour accuracy. Terminal brightness and contrast dials must be pre-set (high) for colour constancy. Colours appear different on the screen in different lighting conditions, as do the colours in images. The number of colours is limited due to simultaneous contrast when selecting colours from a colour palette. Successive contrast from previous colours alters colour perception after a period of time.

The whole system response is slow, due to the AutoLisp and AutoCad interpreter. It is difficult to select descriptors from a long list, better to provide a semantic interpreter. Manual typing of descriptors is prone to error, better as a mouse-driven descriptor menu. Colour concept generating methodology is difficult to ascertain from limited menus. In general the computer system is a prototype that requires considerable development and optimisation to becomes a suitable domain in which to experiment with the idea of computing colour aesthetics.

5.6.2 AutoLisp

AutoLisp proved to be a feasible language from which to code the program prototype for the system. It was easy to learn and allows for more complex manipulation of data through processing simple lists. The list proved to be a suitable construct for system data, ie. current descriptors, colours and their weightings. Also the organisation of the rulebase into a list of colour rules, each of which formed a list of descriptor sets provided a simple structure for a body of integrated data relations. Regarding process coding AutoLisp allows for modular coding of base functions from which more complex constructs can be formed and literally allows for a language to be written from which the system can operate. In this sense the base functions become concise commands with function titles that combine into sentence-like statements at a higher level. This permits coding from algorithms to be performed ‘bottom-up’, ‘top-down’ or ‘middle-out’ allowing for a flexibility of programming style.

The use of recursion and the iterative calling of a function from within itself is an ideal construct for the cyclic looping of a production system as matching rules to descriptors or colours can be achieved by matching lists or list elements. Recursion provides an appropriate programming construct for forward and backward chaining in a production system. AutoLisp is a slow interpreted language and limitations were imposed in the AutoCad system. Colour concept pie-charts were displayed by plotting radial lines (like radar) and the response was slow. The AutoLisp drawing command (grdraw) is not recognised by the AutoCad system so colour concepts could not be saved as AutoCad drawing files. In general the AutoLisp language was suitable for prototyping the computer system though not recommended for development of the system for commercial use.

5.7 Summary

The research was evaluated through a qualitative analysis by observation, questionnaire and discussion using as many participants as possible. It was considered essential to involve potential real users of the system. Final year product design students (all with at least one year industrial experience) volunteered to take part in the evaluation. The researcher participated in the evaluation as an observer and went through the same process as the participants.

Each participant initially went through a set of operating instructions and a tutorial to learn all of the commands for the computer system. The participants were then asked to use the system in a realistic way to teach it colour semiotic relations, generate and describe colour concepts and to then complete the questionnaire. The observations, discussions and returned questionnaires were qualitatively analysed together to establish results and draw conclusions for the research. This was undertaken to obtain an objective opinion of the idea of computing colour aesthetics, the system philosophy, the system methodology and the computer media used in the Colour Concept Generator.

The philosophy of using a computer system as an ideas stimulus for colour concepts at the front end of the design process was found to be feasible providing it is in line with the ethos of generating ideas for concepts in general. The colour concepts might be more broadly defined as aesthetic concepts and include the effects of texture, surface finish and light reflection characteristics to increase the numerous possible combinations of aesthetic media.

The description of colour images through symbolic interpretation provided an objective criteria for colour semiotics as descriptor allocation is justified by reference to an existing entity. Personal and subjective colour images allowed for freedom of expression and progression into abstract descriptors of a more poetic and metaphoric nature. By basing colour concept generation upon translating descriptors through a semiotic software mechanism, new colour concepts could be created by combining colour aesthetic themes.

The development of the knowledge base is in accordance with the anticipated process of allocating descriptors and weightings to descriptor sets in colour rules. The learning process proportions rule descriptor weightings through proportional matrix multiplication of current colour sets and current descriptor sets and maintains a working average of descriptor weightings in the rule descriptor sets.

The learning process takes the long term view of absolute knowledge of all colour semiotics with the intention of integrity increasing as more semiotic relations are learnt as opposed to the short term view of specific knowledge being learnt in isolation. The semiotic knowledge is learnt as a whole body of integrated knowledge as opposed to a list of facts having no connection with each other.

The generating process operates in accordance with the anticipated process of allocating colours to descriptors by matching current descriptors with rule descriptors. The colour weightings are accurately calculated by summing the multiples of the matched current descriptor weightings and the rule descriptor working averages for each colour.

The generating process is logically correct as for an intelligent system to respond to a combined descriptor meaning it must only generate colours that hold that same meaning. If the generation process were accumulative then the long term view of absolute semiotic knowledge would result in a generation of many colours from only a few descriptors. The generating process must be based upon refinement as opposed to expansion regarding specific semiotic definition. However the ability to use both accumulation and refinement would be more suited in the context of designing products.

The describing process operates in accordance with the anticipated process of allocating descriptors to colour combinations by matching current descriptors with colour rule descriptors. The descriptors are produced by selecting descriptors that only fall in all of the colour rules for the colours in the concept.

The descriptors produced for a colour concept are the same regardless of the proportions of the colours. The describing mechanism matches descriptors to colours by colour and not by proportion so a lot of white with a little orange is described by the same descriptors as a lot of orange with a little white. This indicates a failure in the integrity of the system to differentiate in specifically describing colour concepts of differing proportions.

The application of a production system in the computer system prototype proved to be a feasible method for computer learning of colour semiotic relations and producing colour concepts from verbal descriptors. From process and data design to program coding, the organisation and structure of a production system lends a flexible domain in which to build an AI computer tool.

This chapter provided an evaluation of the whole research project. The research was evaluated through a qualitative analysis by observation, questionnaire and discussion using potential system users. This was undertaken to obtain an objective opinion of the idea of computing colour aesthetics, the system philosophy, the system methodology and the computer media used in the Colour Concept Generator. The following chapter is based upon the results from the evaluation and provides conclusions for the research and recommendations for further research in this area for the future.

6 Conclusions and Recommendations for Further Research

6.1 Introduction

The previous chapter was an evaluation of the whole of the research and the Colour Concept Generator computer system. Results were determined from observations, discussions and questionnaires taken from potential real users of the computer tool participating in an evaluation exercise. This chapter is a summary of the research and is based upon the evaluation. A summary of conclusions of the research are provided together with a summary of recommendations for further development of the research.

6.2 Conclusions

The research, analysis, design and evaluation of the Colour Concept Generator as a computer tool to propose colour concepts for products has been presented in this thesis. Primarily the research has demonstrated that it is possible to use an AI software mechanism to compute colour aesthetics. A computer tool has been developed that can translate a verbal description of a required aesthetic image or style into a colour concept and vice versa.

The research indicates that the philosophical idea of the Colour Concept Generator need not be limited to colour alone. The AI mechanism in the Colour Concept Generator has the potential to develop to incorporate other aesthetic criteria such as texture, shape and form. This opens the scope of the research to the idea of an 'Aesthetic Concept Generator' and the possibility of an AI system to assist with holistic conceptual aesthetic design.

The Colour Concept Generator is a feasible computer tool to provide an ideas stimulus for a product designer to create product colour schemes. An AI system with the potential to inspire creative human thought for colour aesthetics has been produced. The philosophical idea of using a computer system as an ideas stimulus for colour concepts at the front end of the design process is feasible with the ethos of generating ideas for product concepts.

The system learning mechanism allows for symbolic interpretation of colour semiotics as colour and descriptor allocation can be referenced to a descriptor relating to an existing entity. A product designer can create personal colour images allowing for freedom of expression and progression into abstract descriptors of a more poetic and metaphoric nature. The learning inference mechanism is adaptable for both symbolic and personal colour semiotics, providing a balanced and integrated approach. The system has the power to link any kind of descriptors to any colours. Context specific colour semiotic associations such as cultural differences can be accommodated as linking colours to descriptors can include contextual descriptors in a descriptor group.

The research illustrates the possibility of machine learning of colour semiotic knowledge acquired through an interactive learning process. The knowledge can be developed in a production system rulebase through the application of proportional matrix multiplication. Re-weighting descriptors in a rulebase using proportional matrix multiplication provides a dynamic mechanism for machine learning of colour semiotic relations. This is in line with a multi-layer perceptron in the form of a feedforward neural network.

The machine learning process should take the long term view of developing knowledge of colour semiotics, with the intention of integrity increasing as more semiotic relations are learnt. This is opposed to the short term view of specific knowledge being learnt in isolation. The research shows that semiotic knowledge can be learnt as a whole body of integrated knowledge instead of a list of facts having no connection with each other. This has been achieved through the distributed learning mechanism and the ability to link colours and descriptors in a 'many to many' relation.

Generating colour concepts can be an accumulative process whereby all colours matching all descriptors are selected or it can be a refined process whereby only colours matching all descriptors are selected. The option to apply both methods for both generating and describing colour concepts must be made available for the system to more practical in the context of designing products.

The research indicates that generating colour concepts from weighted aesthetic descriptors and weighted rule descriptors can be achieved by proportional matrix multiplication. Proportional matrix multiplication does not allow for reversibility of colour and descriptor weightings for generating and describing colour concepts.

The application of a multi-layer perceptron as a feedforward neural network allows for a dynamic learning mechanism. The application of a rulebased production system allows for the use of symbolic processing whereby colour semiotic knowledge is processed in terms of colours and descriptors. This combination provides an AI computer tool that is both adaptable and uses natural language as a communication medium.

The application of the AutoLisp programming language and AutoCad for the computer system proved to be a feasible method for computer learning of colour semiotic relations and generating colour concepts from descriptors. From process and data design to program coding, the organisation and structure of AutoLisp and AutoCad were feasible computer media for building an AI computer tool to translate between descriptors and colours.

6.3 Further Research

Further research should be directed towards the idea of computing aesthetics in a holistic way that includes all aesthetic attributes relating to colour, texture shape and form. The philosophical aspects included in the design of the Colour Concept Generator should be considered. This involved integration into the design process, translating between semiotic aesthetic media, machine learning of aesthetic semiotic knowledge, generating and describing aesthetic media and selecting appropriate computer processing media and interfaces. The following points are recommendations for developing the research:

- Include semiotic interpretations of all aesthetic attributes relating to colour, texture, shape and form.

- Provide a semantic descriptor inference engine (like a thesaurus) to interpret and expand aesthetic descriptions.

- Analyse the problem of reversibility of weightings for generating and describing aesthetic concepts.

- Provide a means of generating and describing aesthetic concepts that allows for both expansion and refinement in the processing mechanism.

- Develop the AI software mechanism to be more creative and original.

- Experiment with more sophisticated technological media (eg. holograms) for the interface.

References

(1) Horizon (1994), 'Orange Sherbet Kisses', BBC Education.

(2) Omnibus (1994), 'The Pink Floyd Story', BBC Production.

(3) Eves, W. R. (1991), 'The Development of a System to Produce Computer Graphical Art Representative of Musical Input', MSc Computer Studies Thesis, Sheffield City Polytechnic.

(4) Owen, C. L. (1994), 'Design Research, Building the Knowledge Base', RCA Design Research Conference.

(5) Varley, H. (Ed) (1980), 'Colour', Marshall Editions Ltd.

(6) Brusatin, M. (1991), 'A History of Colours', Shambhala Pubs Inc.

(7) Goethe, J. W. (1840), 'Theory of Colours, 1967 Translation', Frank Cass & Co Ltd.

(8) Padgham, C. A. & Saunders, J. E. (1975), 'The Perception of Light and Colour', Bell.

(9) Overheim, R. D. & Wagner, D. C. (1982), 'Light and Colour', Wiley.

(10) Chevreul, M. E. (1967), 'The Principles of Harmony and Contrast of Colours', Pelican.

(11) Bagley, M. & Maxfield, M. S. (1986), 'Afterimage Colour Perception for Designers', from 'Perceptual Motor Skills'.

(12) Itten, J. (1973), 'The Art of Colour', V. N. R.

(13) Goldstein, E. B. (1989), 'Sensation and Perception', Wadsworth.

(14) Brill, D. & Falk, D. & Stork, D. (1986), 'Seeing the Light', Harper & Row.

(15) Cumming, R. & Porter, T. (1990), 'The Colour Eye', BBC Books.

(16) Yau, V. (1994), 'Use of Colour in China', from 'British Journal of Aesthetics, Vol 4 No 2'.

(17) Gerstner, K. (1986), 'The Forms of Colour', M. I. T.

(18) Eco, U. (1985), 'How Culture Conditions the Colours We See', from Blansky, M. (Ed), 'On Signs', Basil Blackwell Pubs.

(19) Trummel, P. (1991), 'Gutten and Itten Revisited: An Imaginary Letter to Johannes from Johannes', from 'IEEE Transactions on Professional Communication, Vol 34 No 3'.

(20) Berlin, B. & Kay, P. (1969), 'Basic Colour Terms', Uni of California Press.

(21) Kutschel, R. & Monberg, T. (1974), 'We Don't Talk Much About Colour Here: A Study of Colour Semantics on Bellona Island', from 'Man, Vol 9 Part 2'.

(22) Miller, G. A. & Johnson-Laird, P. N. (1976), 'Language and Perception', Cambridge Uni Press.

(23) Matthei, R. (1971), 'Goethe's Colour Theory', Studio Vista Ltd.

(24) Lloyd-Jones, P. (1991), 'Taste Today, the Role and Appreciation of Consumerism in Design', Pergaman Press.

(25) Russell, D. (1991), 'Colour and Industrial Design', Design Council.

(26) Horton, W. (1991), 'Overcoming Chromaphobia: A Guide to the Appropriate and Confident Use of Colour', from 'IEEE Transactions on Professional Communication, Vol 34 No 3'.

(27) Horton, W. (1993), 'Guide to the Confident and Appropriate Use of Colour', from 'SPIE, Vol 1912'.

(28) Danger, E. P. (1968), 'Using Colour to Sell', Gower Press.

(29) Danger, E. P. (1987), 'The Colour Handbook', Gower Tech Press.

(30) Linton, H. (1991), Colour Consulting, a Survey of International Colour Design', V. N. R.

(31) Hogg, D. , Hutchison, C. , Sharples, M. , Torrance, S. & Young, D. (1994), 'Computers and Thought', The MIT Press.

(32) Charniak, E. & Mc Dermott, D. (1985), 'Introduction to AI', Addison Wesley.

(33) Patterson, D. W. (1990), ' Introduction to AI and Expert Systems', Prentice-Hall.

(34) Lefley, M. 'Background to AI' (lecture notes).

(35) Tominaga, S. (1985), 'A Colour-Naming Method for Computer Colour Vision', from 'IEEE Proceedings - Cybernetics and Society International Conference'.

(36) Hsu, J. & Tseng, S. & Wei, T. (1990), 'Quantifying Qualitative Relations about Colours Through Fuzzy and Statistical Operations', from 'Colour Research and Application, Vol 15 No 2'.

(37) Pugh, S. (1991), 'Total Design', Addison-Wesley.

(38) Sparke, P. (1987), 'Design in Context', Bloomsbury.

(39) Searle, J. R. (1980), 'Minds, Brains and Programs', The Behavioural and Brain Sciences', Vol 3.

(40) Rossenblatt, F. (1962), 'The Principles of Neurodynamics', Spartan.

(41) Minsky, M. A. & Papert, S. (1969), 'Perceptrons', MIT Press.

(42) McClelland, J. L. & Rummelhart, D. E. (1986), 'Explorations in Parallel Distributed Processing', MIT Press.

(43) Knight, K. & Rich, E. (1991), 'Artificial Intelligence', Mc Graw-Hill Inc.

(44) Rowe, N. C. (1988), 'AI Through Prolog', Prentice-Hall.

(45) Hayes, J. E. (1988), 'Machine Intelligence II', Clarendon Press.

(46) Coxhead, P. (1987), 'Staring Lisp for AI', Blackwell.

(47) Tanimoto, S. L. (1990), 'The Elements of Artificial Intelligence Using Common Lisp', Computer Science Press.

(48) AutoDesk, (1990), 'AutoCad Reference Manual', AutoDesk Ltd.

(49) AutoDesk, (1990), 'AutoLisp Programmers Reference', AutoDesk Ltd.

(50) Cutts, G. (1991), 'Structured Systems Analysis and Design Methodology', Blackwell.

Appendices

Appendix A

Aesthetic Descriptions

1. Child Carrier for Wheelchair Users
Soft, warm, cuddly, comfortable, welcoming, loving, happy, affectionate, playful, vitality, fun, youthful, adventure, caring, gentle, clean safe, protective.

2. Peak Flow Meter for Children Suffering from Asthma
Sympathetic, friendly, soothing, calm, dynamic, gentle, reliable, alive, pure, hygienic, energetic, lively, active, radiant, intense, stimulating, exciting, soft.

3. Man Overboard Alarm System
Durable, positive, dynamic, clear, aggressive, adventure, liberating, Atlantic, protect, faithful, trustworthy, aquatic, energetic, fresh, thrilling, freedom, honest.

4. Portable Over Head Projector
Futuristic, organic, powerful, brutal, technical, professional, business, fast, alien, innovative, expensive, light, aggressive, competitive, office, sexy, dynamic, contract, money, suit.

5. Portable Baby Milk Cooler
Soft, clean, safe, parents, reliable, friendly, expensive, childlike, quality, cuddly, dairy, fresh, robust, important, smooth, busy, young, lively.

6. Doorbell Answering Machine
Security, safety, quality, reliability, police, communication, technology, protection, talking, alarm, robust, elderly, strength, sound, message, speech, dynamic, durable, electronics.

7. Baby Carrier
Robust, safe, lightweight, comfortable, practical, simple, versatile, adventurous, cosy, freedom, fun, outgoing, friendly, innovative, mobile, reliable, novel, compact.

8. Portable Shower
Convenient, portable, private, compact, powerful, robust, camping, water, sport, beach, travel, recycling, practical, leisure, outdoors, reliable, hygienic, environment, efficient, spacious.

9. Dash Mounted Control Panel
Animated, comfort, compact, conservative, control, convenient, cool, essential, instant, reliable, relieving, responsive, road, safe, simple, travel, vehicle, vigorous.

10. Educational Toy to Increase Awareness of Over-exposure to the Sun
Sunny, friendly, bright, summer, fun, happy, safe, cool, warm, educational, cute, healthy, solid, clean, innovative, children, beach, robust, light, play.

11. Independently Powered Travel Iron
Adventurous, carefree, streamlined, clean, bright, simple, compact, tough, energetic, vibrant, action, fast, international, comfortable, boats, family, aqua, tactile, travel.

12. Toy Spacecraft
Action, adventure, fighting, technology, information, weapons, space, planets, metallic, overload, organic, alien, war, laser, futuristic, mechanical, galaxy, flight, boost, explore.

13. Musical Effects Instrument
Durable, musical, creative, inspiring, intricate, strong, powerful, youthful, vivid, artistic, virile, comfortable, stable, clean, scientific, state-of-the-art, modern, acoustics, radiant, poetic.

14. Car Balancing Scales
Powerful, precise, accurate, professional, reliable, solid, lightweight, modern, compact, functional, fast, exciting, new, fresh, bright, expensive, sophisticated, tough, stylish, smooth.

15. Portable Battery Powered Industrial Vacuum Cleaner
Lightweight, clean, industrial, professional, fast, rugged, environmental, powerful, portable, comfortable, modern, simple, engineered, hygienic, durable, mobile, innovative, new, efficient, safe.

16. Live Bait Storage System
Tough, lively, neat, clean, angling, boat, sea, peaceful, floating, rigid, stability, secure, ripples, reflection, solid, water, safe, fish.

17. Golf Practice System
Professional, sophisticated, environmental, portable, outdoors, metamorphosis, quality, fun, sporty, durable, circular, symmetrical, grass, reliable, simple, stable, unique, original, business, synthetic.

18. Electric Guitar
Modern, fast, cool, fun, sophisticated, technical, sweat, sexy, power, steel, flash, refined, racy, accessible, mechanical, warrior, magical, sleek, unconventional.

19. Bathroom Hairdryer
Simplistic, elegant, coastal, pastel, clean, smooth, sandstone, natural, Art Deco, hygienic, modern, functional, business, leisure, relaxing, subtle, European, polished, soothing, romantic.

20. Cycle Trailer
Fun, mountain, fast, speed, explorer, wildlife, trees, mission, agile, global, sherpa, trekking, strong, young, outdoors, technology, lightweight, reliable.

Appendix B

System Code

```
(defun setup ()
; initialisation of system, activated by CCGRUN on the menu
	(command "redraw")
; initialisation of colour & descriptor lists
	(setq cc ())
	(setq clist ())
	(setq cd ())
	(setq dlist ())
	(setq dd ())
	(setq newccl ())
; positions for centre of pie chart
	(setq piex 200.0)
	(setq piey 200.0)
	(setq piecen (list piex piey))
; sets radii for pie chart & colour palette
	(setq smarad 40.0)
	(setq bigrad 100.0)
	(setq palrad 120.0)
	(setq aoned (/ smarad 3.0))
	(setq toned smarad)
; initialisation of rulebase
	(setq ccgrb ())
	(setq temprb ())
	(setq artist 0)
; sets AutoCad screen limits
	(command "limits" (list 0 0) (list 320 320))
	(command "zoom" "all")
; loads rulebase & creates colour palette
	(loadrb)
	(setq ccgrb inrbas)
	(pallet)
	(command "ucsicon" (setq turn "off"))
)
```

```
(defun ades ()
; adds a descriptor & weighting to the descriptor set if not already present
; proportions the descriptor weighting relative to all descriptor weightings in the set
	(vcd)
	(desmen)
	(setq dess (getdes))
	(cond	((member dess cd) (indes))
		((not (member dess cd))
			(setq cd (percent (addto dess (propval (getval cd)) cd)))
			(vcd)
)))

(defun ddes ()
; removes a descriptor & weighting from the descriptor set
; proportionally re-weights the remaining descriptors in the set
	(cond	((null cd) (nocd))
		(t	(vcd)
			(setq dez (getdes))
			(cond	((not (member dez cd)) (notin))
				((member dez cd)
					(setq cd (percent (order (remaav dez cd))))
					(vcd)
)))))

(defun deldes ()
; deletes all descriptors & weightings in the descriptor set
	(setq cd ())
)

(defun vcd ()
; prepares to view the descriptors & weightings
	(space)
	(prompt "\ncurrent descriptors:  ")
	(space)
	(vcd2 cd)
)
(defun vcd2 (kd)
; displays the descriptors & weightings on the screen
	(cond	((null kd) (space) nil)
		(t	(space)
			(princ (car kd))
			(princ "  ")
			(princ (car (cdr kd)))
			(vcd2 (cdr (cdr kd)))
)))
```

```
(defun desmen ()
; prepares to view the descriptors from the rulebase as a descriptor menu
        (setq mlist ())
        (domenu ccgrb)
        (space)
        (prompt "\ndescriptor menu:  ")
        (space)
        (menout mlist)
)

(defun domenu (rulbas)
; reads the descriptors from the rulebase
        (while  (not (equal rulbas nil))
                (mendon (prems (car rulbas)))
                (setq rulbas (cdr rulbas))
))

(defun mendon (premz)
; reads a descriptor from a descriptor set in the descriptor premises in a rule
; providing the descriptor has not been read from another rule
        (cond   ((null premz) nil)
                ((not (member (des (desset premz)) mlist))
                        (setq mlist (cons (des (desset premz)) mlist))
                        (mendon (cdr premz)))
                (t      (mendon (cdr premz))
)))

(defun menout (ml)
; displays the descriptor menu on the screen
        (cond   ((null ml) (space) nil)
                (t    (space)
                        (princ (car ml))
                        (menout (cdr ml))
)))

(defun acol ()
; prepares for a colour to be added to the colour set
        (setq colno 254)
        (setq chosen nil)
        (select)
)
```

```
(defun select ()
; displays the colours & weightings & colour palette
        (vcc) (vcp) (saycol)
; user points cursor on a colour & clicks mouse
; reads cursor position and determines selection
        (while  (null chosen) (getcol)
                (cond   ((<= dist aoned)
; colour number is to be increased by one
                                (addone))
                        ((and   (> dist aoned) (<= dist toned))
; colour number is to be reduced by one
                                (takone))
                        ((and   (> dist smarad) (<= dist bigrad)
                                (not (member colno (aolist cc))))
; colour is to be added to the pie
                                        (addpie))
                        ((and   (> dist bigrad) (<= dist palrad))
; colour number to be selected
                                        (palsel))
; selection done
                        ((> dist palrad) (seldon)
))))

(defun dcol ()
; prepares for a colour & weighting to be removed from the colour set
; re-proportions the remaining colour weightings having removed the colour & weighting
        (cond   ((null cc) (nocol))
                (t      (vcc)
                        (getcol)
                        (cond   ((> dist bigrad) nil)
                                ((<= dist bigrad)
                                        (setq pietot 0.0) (setq copycc cc)
                                        (setq cc
                                        (percent (order (remaav (findatt ang cc) cc))))
                                        (vcc)
)))))

(defun proport ()
; prepares to re-proportion a colour weighting in the colour set
        (cond   ((null cc) (nocol))
                (t      (proportion)
)))
```

```
(defun proportion ()
; obtains cursor position after a mouse click indicating a colour to be re-proportioned
; determines whether colour proportion is to increase or decrease
        (getcol)
        (cond   ((<= dist bigrad)
; colour proportion is to be increased or decreased
                        (preprop) (vcc) (proportion))
                ((> dist bigrad)
; re-proportioning completed & view new colours and proportions
                        (vcc)
)))

(defun preprop ()
; prepares to find a colour to be re-proportioned in the colour set
; creates the new colours set with new proportional weightings
        (setq pietot 0.0)
        (setq copycc cc)
        (setq cc ())
        (doprop (findatt ang copycc) copycc)
        (setq cc (percent (order cc)))
)

(defun doprop (attcol attcc)
; finds a colour to be re-proportioned in the colours set
        (cond   ((null attcc) nil)
                ((not (equal attcol (car attcc)))
                        (nochange attcc)
                        (doprop attcol (cdr (cdr attcc))))
; checks threshold for increasing or decreasing a colour proportion
; colour increased if proportion less than 95%
; colour decreased if proportion more than 5%
                ((equal attcol (car attcc))
                        (cond   ((and (< (val attcc) 95.0) (> dist smarad))
                                        (increase attcc))
                                ((and (> (val attcc) 5.0) (<= dist smarad))
                                        (decrease attcc))
                                (t      (nochange attcc)))
                        (doprop attcol (cdr (cdr attcc)))
)))

(defun delcol ()
; deletes all colours & weightings in the colours set
        (setq cc ())
        (command "redraw")
)
```

```
(defun vcc ()
; prepares to view the colours & weightings in the colour set
        (command "redraw")
        (cond   ((null cc) (nocol))
                (t      (setq radius bigrad)
                        (setq plsum (sumval cc))
                        (plot cc)
                        (space)
                        (prompt "\ncurrent colours:  ")
                        (space)
                        (vcd2 cc)
)))

(defun learn ()
; prepares to create or modify descriptor weightings in the rule premises in the rulebase
        (cond   ((or (null cc) (null cd)) (noinfo))
                (t      (setq temprb ccgrb)
                        (setq ccgrb ())
                        (learnt temprb)
                        (savrb)
)))

(defun learnt (rbase)
; reads a rule from the rulebase
        (while  (not (equal rbase nil))
                (check (car rbase))
                (setq rbase (cdr rbase))
))

(defun gccon ()
; prepares to generate a colour set from a descriptor set
        (cond   ((null cd) (nocd))
                (t      (setq cc ())
                        (setq newccl ())
                        (rulseq ccgrb)
                        (setq cc (percent newccl))
                        (setq newccl ())
                        (vcc)
)))
```

```
(defun gcdes ()
; prepares to produce an associated descriptors set from a colour set
         (cond  ((null cc) (nocol))
                (t     (setq dd ())
                       (seqrul ccgrb)
                       (vdd)
)))

(defun vcp ()
; prepares to view the colour palette
         (setq artist 1)
         (setq radius palrad)
         (setq plsum 27.0)
         (plot pclist)
         (setq artist 0)
)

(defun palsel ()
; prepares to select a colour from the colour palette
         (palget)
         (saycol)
)

(defun palget ()
; determines the colour numbers for the colour palette
         (setq palseq 0)
         (while  (<= palseq (hueno))
                 (setq palseq (+ palseq 10)))
         (setq colno (- palseq 10))
)

(defun hueno ()
; determines the angular position for a colour number
          (fix (* 260 (/ ang (* 2.0 pi))))
)

(defun pallet ()
; creates the colour number list for the colour palette
         (setq pclist ())
         (setq palcol 250)
         (while  (>= palcol 0)
                        (setq pclist (append (list palcol 1) pclist))
                        (setq palcol (- palcol 10))
))
```

```
(defun vrb ()
; prepares to view the rulebase
        (space)
        (prompt "\nrulebase:  ")
        (space)
        (vrb2 ccgrb)
)

(defun vrb2 (kbase)
; reads the rulebase and displays the contents on the screen
        (cond   ((null kbase) (space) nil)
                (t      (space)
                        (princ (car kbase))
                        (vrb2 (cdr kbase))
)))

(defun plot (cl)
; prepares to plot the colour pie chart or colour palette on the screen
        (cond   ((null cl) nil)
                (t      (setq rads 0.0)
                        (setq prop (/ (* 2.0 pi) plsum))
                        (radfil cl)
)))

(defun radfil (colors)
; prepares to plot an angular proportion in the colour pie chart or colour palette
        (cond   ((null colors) nil)
                (t      (setq colbit (* prop (car (cdr colors))))
                        (piebit (car colors) colbit)
                        (radfil (cdr (cdr colors)))
)))

(defun sumval (alist)
; determines the sum of the weightings in a colour or descriptor set
        (cond   ((null alist) 0.0)
                (t      (+      (val alist)
                                (sumval (cdr (cdr alist)))
))))
```

```
(defun piebit (col cprop)
; prepares to plot lines in an angular proportion of a colour pie chart or colour palette
        (setq rcount 0.0)
        (while  (<= rcount cprop)
                (doline rads col)
                (setq rads (+ rads 0.01))
                (setq rcount (+ rcount 0.01))
))

(defun doline (angle colour)
; plots lines in a colour pie chart or colour palette in the required colour
        (cond   ((equal artist 1)
                        (grdraw (polar piecen angle bigrad)
                                (polar piecen angle radius) colour)
                        (grdraw (polar piecen (+ angle 0.005) bigrad)
                                (polar piecen (+ angle 0.005) radius) colour))
                (t      (grdraw piecen (polar piecen angle radius) colour)
                        (grdraw piecen (polar piecen (+ angle 0.005) radius) colour)
)))

(defun addone ()
; increases a selected colour number if it is less than 254
        (setq colno (+ colno 1))
        (cond   ((>= colno 255)
                        (setq colno 0)))
        (saycol)
)
(defun takone ()
; decreases a selected colour number if it is greater than 0
        (setq colno (- colno 1))
        (cond   ((<= colno -1)
                        (setq colno 254)))
        (saycol)
)

(defun saycol ()
; displays a colour number on the screen
        (space)
        (prompt "\ncolour number:  ")
        (princ colno)
        (space)
        (setq radius smarad)
        (setq plsum 1)
        (plot (list colno 1))
)
```

```
(defun addpie ()
; creates the new colours set when adding a new colour & weighting to the set
        (setq cc (percent (addto colno (propval (getval cc)) cc)))
        (vcc)
        (vcp)
)
(defun seldon ()
; indicates that a selection process has ended and displays the colours & weightings
        (setq chosen t)
        (vcc)
)

(defun findatt (attang seasea)
; finds a colour or descriptor in a colour or descriptor set
        (cond   ((null seasea) nil)
                ((>= attang pietot)
                        (setq pietot (+ pietot (valprop (val seasea))))
                        (setq seqvar (car seasea))
                        (findatt attang (cdr (cdr seasea))))
        (eval seqvar)
)

(defun remaav (attrib avlis)
; removes a colour or descriptor and their weighting from a colour or descriptor set
        (cond   ((null avlis) nil)
                ((not (equal attrib (car avlis)))
                        (append (remaav attrib (cdr (cdr avlis)))
                                (list (car avlis) (val avlis))))
                ((equal attrib (car avlis))
                        (remaav attrib (cdr (cdr avlis)))
)))

(defun aolist (liszt)
; reads a colour or descriptor from a colour or descriptor set
        (cond   ((null liszt) nil)
                (t (cons (car liszt) (aolist (cdr (cdr liszt))))
)))

(defun fcolv (cconc collis)
; finds a colour or descriptor weighting in a colour or descriptor set
        (cond   ((null collis) nil)
                ((equal cconc (car collis))
                        (val collis))
                (t (fcolv cconc (cdr (cdr collis)))
)))
```

```
(defun rdlist (prms)
; builds a rule descriptor set
        (cond   ((null prms) nil)
                (t (cons (des (desset prms)) (rdlist (cdr prms)))
)))

(defun getdes ()
; prompts to enter a descriptor
        (space)
        (getstring "\nenter descriptor:  ")
)

(defun getcol ()
; prompts to select a colour by a mouse click & reads the position of the cursor
        (space)
        (setq sel (getpoint "\nselect colour:  "))
        (setq dist (distance piecen sel))
        (setq ang (angle piecen sel))
)

(defun getval (cdorcc)
; prompts to enter a weighting for a colour or a descriptor
; checks that the weighting entered is within the threshold of 1 to 99
        (cond   ((null cdorcc) 99.0)
                (t      (space)
                        (setq wal (getreal "\nenter weighting 1 to 99:  "))
                        (cond   ((and (>= wal 1.0) (<= wal 99.0)) wal)
                                (t      (getval cdorcc)
)))))

(defun order (lost)
; reverses the order of a colour or descriptor set
        (cond   ((null lost) nil)
                (t      (append (order (cdr (cdr lost)))
                                (list (car lost) (val lost))
))))

(defun percent (lisd)
; prepares to percentage colour or descriptor weightings in a colour or descriptor set
        (setq listwo lisd)
        (persent lisd)
)
```

```
(defun persent (litht)
; reads a colour or descriptor set
; determines the percentage proportions of colour or descriptor weightings in the set
        (cond   ((null litht) nil)
                (t      (append (list   (car litht)
                                                (*      (/ 100.0 (sumval listwo))
                                                        (val litht)))
                                        (persent (cdr (cdr litht)))
))))

(defun space ()
; produces a character space
        (prompt "\n")
)
(defun val (lst)
; reads a weighting from a colour or descriptor set
        (car (cdr lst))
)
(defun sum (dset)
; reads a sum from a rule descriptor set
        (car (cdr dset))
)
(defun count (dset)
; reads a count from a rule descriptor set
        (car (cdr (cdr dset)))
)
(defun prems (rool)
; reads the descriptor premises from a colour rule
        (car rool)
)
(defun conc (rool)
; reads a colour conclusion from a colour rule
        (car (cdr rool))
)
(defun des (deset)
; reads a descriptor from a rule descriptor set
        (car deset)
)
(defun desset (plems)
; reads a descriptor set from the rule descriptor premises
        (car plems)
)
```

```
(defun cpcen (weighting)
; determines a colour proportion in a colour set
        (/ weighting (sumval cc))
)
(defun dpcen (weighting)
; determines a descriptor proportion in a descriptor set
        (/ weighting (sumval cd))
)
(defun insum (dval cval dessum)
; adds a learnt descriptor weighting to a rule descriptor set sum
        (+ (* (dpcen dval) (cpcen cval)) dessum)
)
(defun icount (dcount)
; increases a rule descriptor set count
        (1+ dcount)
)
(defun makeds (descri desv colv dsum desc)
; builds a descriptor set in a colour rule
        (list descri (insum desv colv dsum) (icount desc))
)
(defun desval (som cownt)
; determines a rule descriptor working average
        (/ som cownt)
)
(defun valprop (vs)
; determines the proportional angle for a colour in a pie chart
        (* (/ vs (sumval copycc)) (* 2.0 pi))
)
(defun propval (vaal)
; determines a new proportion for a colour added to a pie chart
        (/ (* 100.0 vaal) (- 100.0 vaal))
)
(defun addto (ite vel lest)
; adds a colour or descriptor and their weighting to a colour or descriptor list
        (append (list ite vel) lest)
)
(defun nochange (atlizt)
; builds the colour set
        (setq cc (addto (car atlizt) (val atlizt) cc))
)
(defun increase (atlist)
; increases the proportions of a colour in the colour set
        (setq cc (addto (car atlist) (+ (val atlist) 5.0) cc))
)
```

```
(defun decrease (attlist)
; decreases the proportions of a colour in the colour set
        (setq cc (addto (car attlist) (- (val attlist) 5.0) cc))
)
(defun makepm (wule)
; builds the descriptor premises in a colour rule
        (append (noprem (prems wule)) (doprem cd wule))
)

(defun nocol ()
; prompts to indicate no colours in the colour set
        (space)
        (prompt "\nno colours  ")
        (space)
)
(defun nocd ()
; prompts to indicate no descriptors in the descriptor set
        (space)
        (prompt "\nno descriptors  ")
        (space)
)
(defun notin ()
; prompts to indicate not found a descriptor in a descriptor set
        (space)
        (prompt "\nnot found  ")
        (space)
)
(defun indes ()
; prompts for incorrect descriptor type
        (space)
        (prompt "\nbad descriptor  ")
        (space)
)
(defun noinfo ()
; prompts for no colours or descriptors when describing or generating
        (space)
        (prompt "\nnot enough information  ")
        (space)
)
```

```
(defun check (rule)
; checks if the colour in a rule is in the colours set
; if colour present then reads the rule
        (cond   ((not (member (conc rule) (aolist cc)))
                        (setq ccgrb (cons rule ccgrb)))
                ((member (conc rule) (aolist cc))
                        (setq ccgrb (cons (dorule rule cc) ccgrb))
)))

(defun dorule (rul ccl)
; checks a colour in the colour set matches a rule colour
; if the colours match then read the rule & the rule colour
        (cond   ((null ccl) nil)
                ((equal (car ccl) (conc rul))
                        (list (makepm rul) (conc rul)))
                (t (dorule rul (cdr (cdr ccl)))
)))

(defun noprem (plums)
; checks if a rule descriptor is in the descriptor set
; if the descriptor is not present then read the descriptor set
        (cond   ((null plums) nil)
                ((member (des (desset plums)) cd)
                        (noprem (cdr plums)))
                ((not (member (des (desset plums)) cd))
                        (cons   (desset plums)
                                (noprem (cdr plums))))
                (t (noprem (cdr plums))
)))

(defun doprem (cdl ruul)
; checks if a descriptor in the descriptor set is in a rule descriptor set
; if the descriptor is not present then build a new descriptor set in the rule
; if the descriptor is present then re-weights the descriptor set in the rule
        (cond   ((null cdl) nil)
                ((member (car cdl) (rdlist (prems ruul)))
                        (cons   (makeds (car cdl) (val cdl) (fcolv (conc ruul) cc)
                                        (sum (madset (car cdl) (prems ruul)))
                                        (count (madset (car cdl) (prems ruul))))
                                (doprem (cdr (cdr cdl)) ruul)))
                ((not (member (car cdl) (rdlist (prems ruul))))
                        (cons   (makeds (car cdl) (val cdl) (fcolv (conc ruul) cc) 0 0)
                                (doprem (cdr (cdr cdl)) ruul)))
                (t (doprem (cdr (cdr cdl)) ruul)
)))
```

```
(defun madset (desc pms)
; matches descriptors with rule descriptors
        (cond   ((null pms) nil)
                ((equal desc (car (desset pms)))
                        (desset pms))
                (t (madset desc (cdr pms))
)))

(defun rulseq (arbase)
; checks if a rule colour is in the colour set
; if the rule colour is not present then read the next rule
        (while  (not (equal arbase nil))
                (cond   ((not (member (conc (car arbase)) (aolist newccl)))
                                (dmatch cd (car arbase))))
                (setq arbase (cdr arbase))
))

(defun dmatch (seedy reign)
; builds a new colour set from matched rule colours
        (cond   ((not (null (cmatch seedy reign)))
                        (setq newccl (addto (conc reign) (valcol reign seedy) newccl))
)))

(defun cmatch (ceedee ruuuul)
; checks if a descriptor in a descriptor set is in a rule descriptor set
; if the descriptor is present then read the next descriptor from the descriptor set
        (cond   ((null ceedee) t)
                ((member (car ceedee) (rdlist (prems ruuuul)))
                        (cmatch (cdr (cdr ceedee)) ruuuul))
                ((not (member (car ceedee) (rdlist (prems ruuuul)))) nil)
))

(defun valcol (roole curdes)
; determines the sum of a colours weightings for a pie chart
        (cond   ((null curdes) 0.0)
                ((member (car curdes) (rdlist (prems roole)))
                        (+      (cvalu roole (prems roole) (car curdes) (val curdes))
                                (valcol roole (cdr (cdr curdes)))))
                (t      (valcol roole (cdr (cdr curdes)))
)))
```

```
(defun cvalu (rewl rprems dee vee)
; determines a colour weighting from descriptor set weightings & rule descriptor
weighting
        (cond   ((null rprems) nil)
                ((equal dee (des (desset rprems)))
                        (*      (dpcen vee)
                                (desval (sum (desset rprems)) (count (desset rprems)))))
                (t      (cvalu rewl (cdr rprems) dee vee)
)))

(defun seqrul (arbass)
; checks if a rule colour is in the colours set
; if the colour is present then it builds a descriptor set
        (while  (not (equal arbass nil))
                (cond   ((member (conc (car arbass)) (aolist cc))
                                (setq dd (makedd (rdlist (prems (car arbass))) dd))))
                (setq arbass (cdr arbass))
))

(defun makedd (desses ddl)
; makes the associated descriptors set
        (cond   ((null desses) nil)
                ((null ddl) desses)
                (t      (filter desses ddl)
)))

(defun filter (rddes ddlis)
; filters matched descriptors for associated descriptors set
        (cond   ((null ddlis) nil)
                ((member (car ddlis) rddes)
                        (cons (car ddlis) (filter rddes (cdr ddlis))))
                ((not (member (car ddlis) rddes))
                        (filter rddes (cdr ddlis))
)))

(defun vdd ()
; prompts to display the associated descriptors set
        (space)
        (prompt "\nassociated descriptors:  ")
        (space)
        (vdd2 dd)
)
```

```
(defun vdd2 (ddd)
; displays the associated descriptors set on the screen
        (cond  ((null ddd) (space) nil)
               (t     (space)
                      (princ (car ddd))
                      (vdd2 (cdr ddd))
)))

 (defun dobase ()
; creates a blank rulebase on the disc
        (setq ccgrb ())
        (setq colerNo 0)
        (while  (<= colerNo 254)
                      (setq ccgrb (cons (list () colerNo) ccgrb))
                      (setq colerNo (+ colerNo 1)))
        (savrb)
)

(defun loadrb ()
; prepares to load the rulebase from the disc
        (setq inrbas ())
        (setq okbase (open "a:ccgrules.bas" "r"))
        (loadrb2)
        (close okbase)
)

(defun loadrb2 ()
; loads the rulebase from the disc
; checks for file prompt EZ to indicate end of file
; checks for file prompt RZ to indicate a rule
        (setq filatt ())
        (while  (not (equal filatt "EZ"))
                      (setq filatt (read-line okbase))
                      (cond  ((equal filatt "RZ")
                                    (setq dblis ())
                                    (premsbit)
))))
```

```
(defun premsbit ()
; reads the descriptor premises from the rulebase
; checks for file prompt DZ to indicate a descriptor set
; checks for file prompt CZ to indicate a colour
        (setq nexatt (read-line okbase))
        (cond  ((equal nexatt "DZ")
                        (desbit)
                        (premsbit))
               ((equal nexatt "CZ")
                        (rulbit)
)))

(defun desbit ()
; reads a descriptor sum & count from a rule descriptor set
        (setq dblis   (cons  (list  (read-line okbase)
                                    (read (read-line okbase))
                                    (read (read-line okbase))) dblis)
))

(defun rulbit ()
; builds the system rulebase from the disc rulebase
        (setq inrbas   (cons  (list dblis (read (read-line okbase)))
                               inrbas
)))

(defun savrb ()
; prepares to save the rulebase to disc
; inserts end of file indicator EZ
        (setq ikbase (open "a:ccgrules.bas" "w"))
        (savrb2 ccgrb)
        (fspace) (princ "EZ" ikbase)
        (close ikbase)
)

(defun savrb2 (ikbs)
; saves the rulebase to disc
; inserts rule indicator RZ and colour indicator CZ
        (while  (not (null ikbs))
                (fspace) (princ "RZ" ikbase)
                (cond  ((equal (prems (car ikbs)) nil)
                                (fspace) (princ "CZ" ikbase)
                                (fspace) (princ (conc (car ikbs)) ikbase))
                       (t     (rulsav (prems (car ikbs)) (conc (car ikbs)))))
                (setq ikbs (cdr ikbs))
))
```

```
(defun rulsav (rprems hugh)
; saves the  rules in the rulebase to disc
; inserts colour indicator CZ & descriptor set indicator DZ
        (cond   ((null rprems)
                        (fspace) (princ "CZ" ikbase)
                        (fspace) (princ hugh ikbase)nil)
                (t      (fspace) (princ "DZ" ikbase)
                        (fspace) (princ (des (desset rprems)) ikbase)
                        (fspace) (princ (sum (desset rprems)) ikbase)
                        (fspace) (princ (count (desset rprems)) ikbase)
                        (rulsav (cdr rprems) hugh)
)))

(defun fspace ()
; inserts a new line in the rulebase
        (princ "\n" ikbase)
)
```

Appendix C

User Interface

Plate 1

Colour Palette

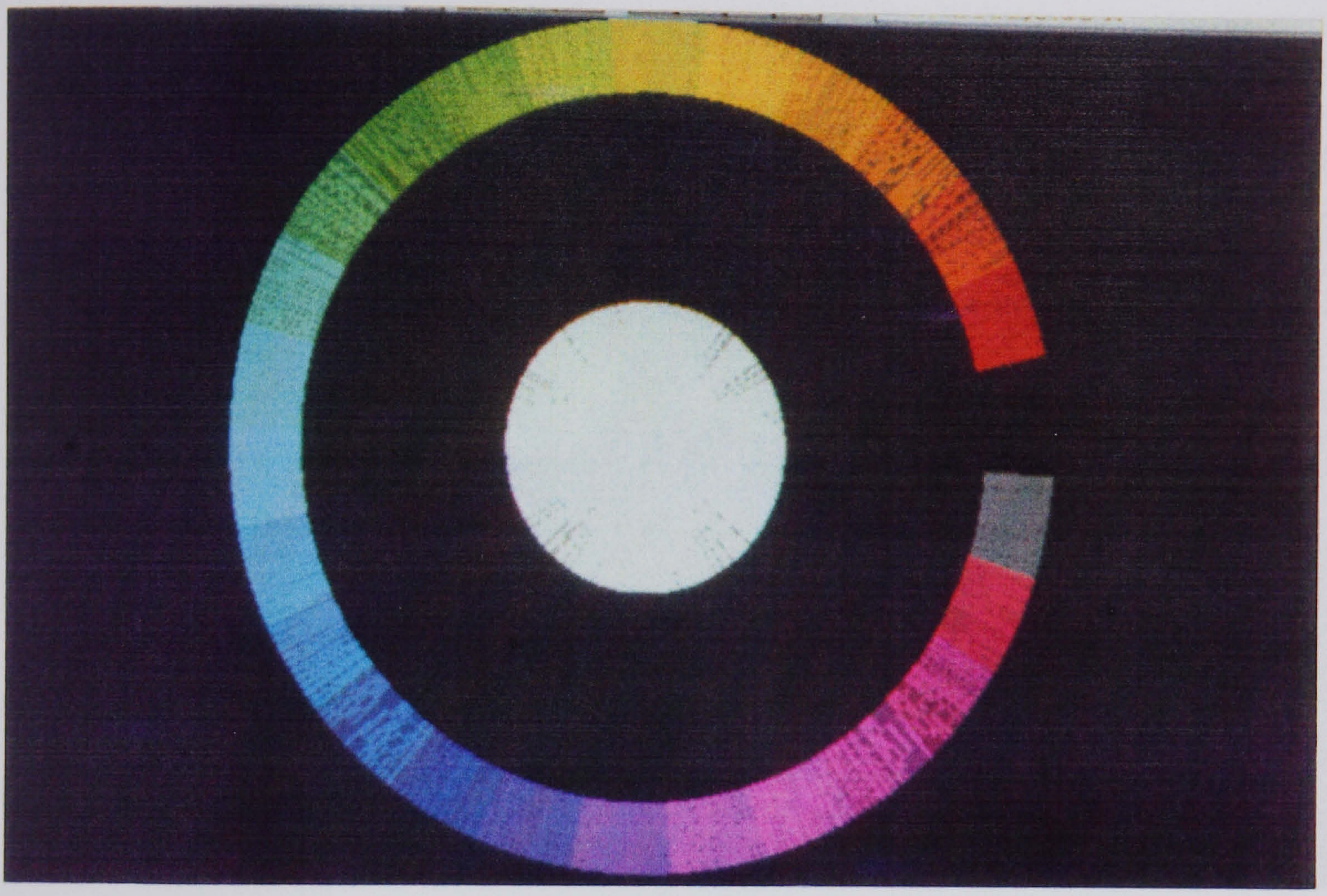
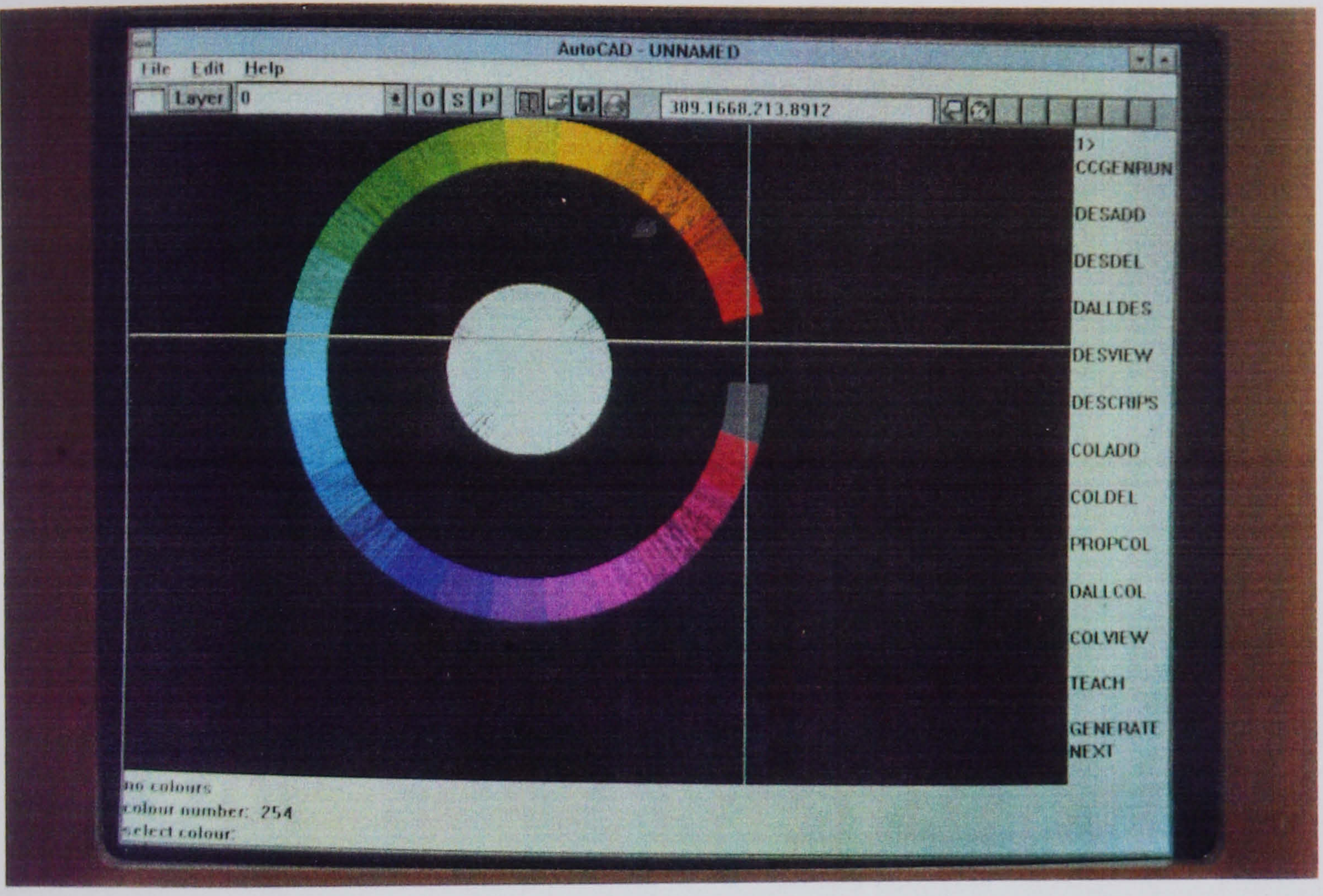


Plate 2

User Interface



Appendix D

Operating Instructions

Operating Methodology

The Colour Concept Generator is an artificial intelligent system designed to assist the product designer in the selection of colour schemes for commercial products. The system 'learns' colour semiotic relations from the user, this is done by firstly creating a colour concept on the screen in the form of a pie-chart paying particular attention to colours and proportions. The colour concept is then described using words such as fresh, clean, clear, calm, organic, oceanic, coral, priest, medical etc.... The relationship is subsequently learnt by the system.

Colour concepts can be made up by the user or derived from existing natural, cultural or commercial colour images and the colour concepts can be formed from words or the words can be formed from the colour concepts (ie. either way round). The words used to describe the colour concept can be adjectives, nouns, verbs or even poetic metaphor, whatever is deemed appropriate. The teaching and learning process is ongoing and can take place at any time. Once that the system has accumulated enough information it can be used to generate and describe colour schemes based upon what it has learnt.

To generate a colour concept the system is given a list of words (current descriptors with weightings) from its menu, describing the required aesthetic image and style of a commercial product. A colour concept is then generated which can be modified by adding, deleting or re-proportioning colours. The modified colour concept is then described and a list of words is produced from which the current descriptors can be altered by adding, deleting or re-weighting descriptors. The colour concept is then re-generated and the whole process repeats until a colour concept is accepted.

Initialising the System

- Activate the AutoCad environment and insert the system disc into drive 'a'.
- Type "menu" at the command prompt, select drive 'a' and open the file 'congen.mnu'.
- The system menu now appears on the right of the screen, initialise the system by clicking on the top command 'CCGENRUN'.
- Information appears at the command prompt, toggle between the prompt and the AutoCad display by pressing F2.

A new rulebase can be created by typing "(dobase)" at the command prompt. NOTE, this will destroy any existing rulebases and create a new blank rulebase in its place. Only use this command at initial installation of the system.

Menu Commands

CCGENRUN -Initialisation of system, resets current colour concept, descriptors and weightings.

DESADD -Adds a descriptor to the current descriptors.

DESDEL -Deletes a descriptor from the current descriptors.

DALLDES -Deletes all the current descriptors.

DESVIEW -Displays all the current descriptors.

DESCRIPS -Displays a list of all descriptors known by the system.

COLADD -Adds a colour to the current colour concept.

COLDEL -Deletes a colour from the current colour concept.

PROPCOL -Re-proportions a colour in the current colour concept.

DALLCOL -Deletes all colours in the current colour concept.

COLVIEW -Displays the current colour concept.

TEACH -Forms a semiotic relation in the system between the current colour concept and descriptors.

GENERATE -Generates a colour concept from the current descriptors.

(NEXT) -Toggles between the main menu and 'DESCRIBE'.

DESCRIBE -Generates a list of descriptors from the current colour concept.

Creating a Colour Concept

To add a colour to a colour concept (or put one in for the first time) click on the menu command 'COLADD', a circular colour palette appears with a grey circle in the centre. All of the available colours (254 in all) are not visible though it is through this interface that all colours can be accessed. The colours shown are all those with a colour number of multiple 10 (ie. 10 (red, at the middle right-hand side) to 20, 30 etc.... moving anti-clockwise, up to 250) clicking on one of the palette colours causes that colour to appear in the central circle and the colour number is displayed at the colours prompt. This is a sample colour that has not yet been fully accepted but has been selected as a possible colour to add to the colour concept.

Accessing the colours in between those shown (eg. 11, 12 ....up to.... 18, 19) is done via the central circle. Clicking in the centre advances the colour number by one, clicking towards the outer of the central colour circle reduces the colour number by one, each time the next or previous colour appears at the central colour circle. Notice that the colours alternate between pastel and bold, even colour numbers are bold and odd colour numbers are pastel, this is the AutoCad colour numbering system. Once that a colour is found that is desired it can be added to the colour concept by clicking between the central colour circle and the outer colour palette.

If the colour added is the first one in the colour concept then the colour fills the larger circle (note, if this was black (colour number 0) there would be no difference in appearance), the system prompts 'current colours:' and list the colour numbers and percentage proportions (in the case of the first colour 100%). Toggle between the command prompt and the system using the F2 function key, always check this prompt especially when black has been selected. If other colours are required they can be selected ad infinitum, there is no need to re-activate 'COLADD', simply select another colour from the palette and move forward and back through the different shades of that colour via the central colour circle, clicking between the central colour circle and the outer palette to select that colour.

When a colour is selected and there is already colour in the concept the system prompts to 'enter weighting', this is the required percentage proportion of the colour to be added. All other colours and proportions remain relative to each other while the one to be added goes in at the proportion entered. Enter a number between 1 and 99 for the colour proportion and press return, the colour appears in the colour pie-chart in the proportion stated. Repeat this process until all desired colours are entered. To fully accept all colours in the concept click outside the colour palette, the colour pie-chart appears without the palette. This is not an ultimate acceptance as more colours can be added by re-selecting 'COLADD' or the colour concept can be modified. To recall the colour concept select 'COLVIEW' from the menu.

Modifying a Colour Concept

To remove a colour from a colour concept click on the menu option 'COLDEL' the system prompts to 'select colour' (to delete), click on the colour in the pie-chart for deleting and the pie-chart is redrawn without that colour. All other colours are relatively percentaged according to previous proportions. To delete all of the colours, ie. the entire colour concept, select 'DALLCOL' from the menu, all colours disappear.

To re-proportion a colour in the pie-chart click on the menu item 'PROPCOL', the prompt requests to 'select colour' (for proportioning). To increase the percentage proportion of a colour click towards the wider, outer edge of the colour wedge in the pie-chart, the colour percentage proportion increases by a few percent. To decrease the percentage proportion of a colour click towards the thinner, inner point of the colour wedge in the pie-chart, the colour percentage proportion reduces by a few percent. All other colours in the pie-chart are re-percentaged according to their previous proportions. Once a colour proportion has been adequately modified the colour concept is accepted by clicking on the outside of the pie-chart and the colour concept is displayed.

Creating Descriptors

To enter a descriptor into the current scenario select 'DESADD' from the menu, a list of current descriptors (if any already entered) appears at the prompt, a descriptor menu (descriptors known to the system) is listed and the prompt 'enter descriptor' appears (use F2 to toggle between the prompt and the system). Type in a descriptor (check the spelling as the system does not do this) and press return, the prompt 'current descriptors:' appears along with a list of current descriptors and percentage proportions (100% for the first descriptor).

To enter other descriptors select 'DESADD' again, a list of current descriptors and weightings appears along with the menu and 'enter descriptor' prompt. Type in a descriptor and press return, the system prompts to 'enter weighting 1 to 99', this is the same process as colours whereby certain descriptors can be given greater emphasis by their percentage proportion. Enter a weighting for the descriptor to be added, as with colours all other percentages are taken as relative based upon previous proportions. A list of the current descriptors and weightings appears at the prompt. To view the current descriptors and weightings select 'DESVIEW' from the menu, to view all descriptors known to the system select 'DESCRIPS' from the menu.

Modifying Descriptors

To remove a descriptor from the current descriptors select 'DESDEL' from the menu, the current descriptors and weightings appear at the prompt and the system prompts to 'enter descriptor' (for deleting). Type in the descriptor (check spelling) and press return, the descriptor has been removed and the system displays the amended descriptors and weightings. As with the colours, all descriptors are percentaged relatively according to previous proportions. To delete all current descriptors activate the menu item 'DALLDES'. To re-proportion a descriptor simply remove the descriptor and re-enter (using 'DESADD') with the required weighting.

Teaching Colour Semiotic Relations

To teach the system a semiotic association between a colour concept pie-chart and a list of descriptive words first check that the colours and descriptors are correct. To do this view the current colours by clicking on 'COLVIEW' and view the current descriptors by clicking on 'DESVIEW' if any modifications are required then do these and view the colours and descriptors again to check. Once happy with both current colours and current descriptors select 'TEACH' (don't do this yet - see the sample test run !!) from the menu, the system prompts "(learn) nil" and the semiotic relationship has been learnt by the system. If the colour concept and descriptors are to be used further then they can be worked upon immediately, if not it is advisable to delete the current colours and descriptors by activating 'DALLCOL' and 'DALLDES', alternatively the system can be cleared and initialised by re-booting with 'CCGENRUN'.

To check that the system has learnt the colour semiotic relation select 'DESCRIPS', this displays all the descriptors known to the system, any totally new descriptors now appear in this list. Activating 'DESADD', entering one of the previously learnt descriptors then activating 'GENERATE' brings back the colour concept (plus any other colours associated with that descriptor acquired through a previous execution of 'TEACH'). If 'DESCRIBE' (via 'NEXT') is subsequently activated, then a list of descriptors is produced that includes the previously learnt descriptors (plus any other descriptors associated with all those colours). Note that once a colour association has been learnt it cannot be unlearnt.

Generating a Colour Concept

A colour concept is generated from one or more current descriptors (and weightings) that have been input via the 'DESADD' process. The descriptors must be known to the system and therefore appear in the descriptor menu produced upon execution of 'DESADD' or viewed in isolation via 'DESCRIPS'. Create a set of descriptors (with weightings) and activate the 'GENERATE' option on the menu. A colour concept appears on the screen and a list of the current colours and weightings appears at the command prompt. The colour concept and descriptors can be modified using the previously explained commands.

Describing a Colour Concept

A description can be produced of a colour concept that has been created using the 'COLADD' function or one that has been generated by the system via the 'GENERATE' process. Taking a colour concept displayed on the screen, first activate 'COLVIEW' to ensure that any modifications have been accounted for, the current colour concept pie-chart appears on the screen. Click on 'DESCRIBE' (via 'NEXT') and a set of associated descriptors are produced at the command prompt. Note there are no weightings as these are not the current descriptors (therefore cannot be used as a generate list) but are an indication of all descriptors that have been associated with all the colours on the screen through previous acquisitions of 'TEACH'.

The associated descriptors produced in the 'DESCRIBE' process can be added to the current descriptors via 'DESADD', by typing in the descriptors and weightings. Harping back to the beginning of the instructions and the operating methodology the process for generating colour concepts becomes one of:

1. Teach the system as many colour semiotic relations as possible.
2. Input descriptors (selected from descriptor menu) for required colours.
3. Repeat until colour concept accepted.
    4. Generate colour concept.
    5. Modify colour concept.
    6. Describe colour concept.
    7. Modify descriptors.

Sample Test Run

It has been proposed that a typical example of apt colour use is that on a surgeons clothing and equipment as it is felt that these colours are indicative of the image of a surgical and medical theme. Through averaging out the exact colours it has been decided on a compromise of the following colours and percentage proportions for the colour concept:

COLOUR	NUMBER	PERCENTAGE (approx.)
Pale Grey	254	10
Pale Blue	130	10
Pale Green	120	10
White	7	40
Grey	252	10
Blue	140	10
Green	110	10

The following descriptors and percentage proportions are to be allocated to the colour concept:

DESCRIPTOR	**PERCENTAGE (approx.)**
Surgical	25
Medical	15
Hygienic	15
Caring	15
Clinical	15
Sterile	15

The first thing to do is to create the colour concept pie-chart, notice that six of the colours are in equal proportion so these must be entered into the concept first. To do this activate 'COLADD' and select the colour green (110), accept this into the colour concept by clicking in between the central colour circle and the outer colour palette. Next select the colour blue (140) accept into the colour concept, the prompt requests a weighting, this colour goes in at 50% as it is required as the same proportion as the other. Type 50 and press return.

The next colour to enter is pale green (120), select and accept this colour and enter the weighting 33.33333 (one third as it is the same proportion as the other two). Do the same for pale blue (130) using a weighting of 25 (a quarter). Now select and accept grey (252) and pale grey (254) using weightings of 20 (a fifth) and 16.66666 (a sixth) respectively. Notice that the central colour circle is required to move through the greys to 252 and 254. Now enter white (7) at a weighting of 40, the colour concept can now be accepted by clicking outside the pie-chart, the colour concept is displayed along with the following colours prompt:

current colours:

7	40.0
254	10.0
252	10.0
130	10.0
120	10.0
140	10.0
110	10.0

Having created the colour concept it has now been deemed necessary to remove the following colours from the pie-chart: darker grey (252), darker blue (140) and darker green (110). To do this click on the 'COLDEL' menu option and click on grey, repeat the process for blue and green and the colour concept is displayed with the following colours prompt:

current colours:

7	57.1429
254	14.2857
130	14.2857
120	14.2857

It is now necessary to re-proportion the colour pie-chart by making the pale green and blue bigger and the white and smaller. To do this activate the 'PROPCOL' menu option and click towards the outer edge of the pale green and blue wedges (to make bigger) and towards the thinner end of the white wedge (to make smaller). Notice that 'PROPCOL' need only be activated once to repeatedly alter the colour wedges. Each time a click is done the colour pie-chart is displayed showing the colours in the new proportions as is also indicated at the colours prompt. Re-proportion the colour wedges until the pale green, pale blue and white colours are approximately equal in proportion (approx. 30% each) then accept the colour concept by clicking outside the colour pie-chart.

The descriptors can now be entered, to do this select the 'DESADD' menu option, the current descriptors are displayed as blank, the descriptor menu lists all the descriptors known by the system and the 'enter descriptor' prompt appears. Notice that five of the descriptors have an equal proportion (15%) so these can be entered first, type in "medical" (omit the speech marks on all descriptor entries, remember to check the spelling) and press return. The descriptor has been entered and has a weighting of 100% (as indicated at the descriptor prompt) as no other descriptors have been entered.

To enter the other descriptors of equal proportions (hygienic, caring, clinical and sterile) repeat the process of selecting 'DESADD', type in a descriptor and weight each one according to the proportion at which it is entered, ie. a half (50), a third (33.33333), a quarter (25) and a fifth (20). Once these are entered the descriptor prompt lists all descriptors with a weighting of 20 (a fifth). It has however been decided to remove the descriptor "caring", to do this select 'DESDEL' from the menu, type in "caring" and press return. The descriptor is now removed and all descriptors now have an equal weighting of 25 (a quarter). Finally enter the descriptor "surgical" with a weighting of 25, the descriptor prompt now reads (depending on the order of entry) the following:

current descriptors:

surgical	25.0
medical	18.75
hygienic	18.75
clinical	18.75
sterile	18.75

The system is now ready to learn the colour semiotic relation between the current colours in the pie-chart and the current descriptors. First of all check that the colours and descriptors are all there and are accurate by selecting the menu items 'COLVIEW' and 'DESVIEW' respectively. When happy with the colours and descriptors activate 'TEACH' and the system learns the relation and returns the prompt '(learn) nil'.

To check that the system has learnt the relation delete all the current colours by selecting 'DALLCOL' from the menu then select 'GENERATE' to produce the colour concept. This instructs the system to generate a colour concept from the current descriptors and weightings (ie. the ones that are still current from the previous teaching acquisition). The colours are then displayed on the screen (possibly in a different order) in the same proportions as before.

To check the descriptors activate 'DESCRIBE' (via 'NEXT') and the list of associated descriptors (without weightings) appears at the descriptor prompt. Notice that this list is not the current descriptors as these are still in place from before, it is important to note at this point that the current descriptors and associated descriptors are separate, ie. the current descriptors (if any) do not change when 'DESCRIBE' is activated. The associated descriptor list is a reference list from which the current descriptors can be modified.

The system has learnt the colour semiotic relation as the colours are generated from the descriptors and vice-versa. Now try generating colours from single descriptors and combinations of different descriptors using different weightings and describing single colours and combinations of different colours in different proportions. Observe variations in colour proportions. Once happy with the general manipulation and operation of the system start teaching it colour semiotic relations and using the information to generate colour schemes for products. Consider natural, commercial and cultural coloured images for inspiration, try creating colour concepts from descriptors as well as producing descriptors for colour concepts.

Try creating colour concepts that have a few common colours and descriptors, for example the medical theme could be expanded to include the red and white of a first aid box or even the fleshy tones of a band aid plaster. Having formed these overlapping colour concept themes try generating and describing colour concepts, observe how the colours, descriptors, weightings and proportions vary as the very same entities vary. Try to be as free as possible in using the system as it is designed to be used as a creative design tool for the personal use of a product designer. As you use the system and become more familiar with its operation please refer to and fill in the attached questionnaire.

Thankyou for your help in evaluating this Ph.D. project.

Bob Eves BEd(Hons), MSc.

Appendix E

Evaluation Questionnaire

Please answer all of the following questions making additional comments and points that you think are important. You might need to refer back to the system, if you have any queries please ask.

Learning how to use the system:

How did you find learning to create colour concepts ?

How did you find learning to modify colour concepts ?

How did you find learning to create descriptor lists ?

How did you find learning to modify descriptor lists ?

How did you find learning to teach colour semiotic relations ?

How did you find learning to generate colour concepts ?

How did you find learning to describe colour concepts ?

How long did it take you to learn how to operate the system ?

Additional comments:

Operating the system:

How did you find creating colour concepts ?

How did you find modifying colour concepts ?

How did you find creating descriptor lists ?

How did you find modifying descriptor lists ?

How did you find teaching colour semiotic relations ?

How did you find generating colour concepts ?

How did you find describing colour concepts ?

Additional comments:

Creating colour concepts:

How did you imagine the colours in your mind ?

How did you gain inspiration for colour concept ideas ?

Were you influenced by existing products ?

What was your speed for coming up with colour concept ideas and how did it vary ?

Did you use existing colour images for ideas ?

How did personal preference influence the design of the colour concepts ?

Were the colours derived from abstract images, real objects or what ?

How restricting were the number of available colours in the system ?

Were there any colours in an image that were omitted from a colour concept ?

Were there any cultural or commercial influences in your design of colour concepts ?

How were the colour concepts influenced by colour in nature ?

Did you attempt to create a colour harmony different to that of an image ?

How did you decide on the colours ?

How did you decide on the proportions ?

Were all the colours in an existing image ?

Did you use all of the colours from an image or just the more prominent ones ?

Did you add colours one at a time from separate sources ?

Did you base proportions on existing images or on what looked best on the screen ?

Did anything influence the order of the colours input ?

Did you mainly use single colours or many colours ?

How did you find matching colours with those on the screen ?

What do you think about representing colour concepts as pie-charts ?

Additional comments:

Creating descriptors:

How did you imagine the descriptors in your mind ?

How did you gain inspiration for ideas for descriptors ?

What influenced the speed of coming up with ideas for descriptors ?

Did you use a dictionary or a thesaurus ?

How extensive or sophisticated was the vocabulary you used ?

What types of words did you use (eg. adjectives, nouns, verbs, metaphor) ?

Did you make up any words of your own ?

How easy / difficult is it to come up with words ?

Did you refer to other colour images when deciding descriptors ?

How did personal experience influence your choice of descriptors ?

Did you mostly allocate single descriptors or many descriptors to colour concepts ?

How did you find selecting weightings for descriptors ?

Additional comments:

Teaching colour concepts:

Which is easier, creating colours from descriptors or descriptors from colours ?

Did you enter whole concepts for teaching or do one colour / descriptor at a time ?

How objective do you think this process is ?

Do you prefer to teach the system yourself or would you like pre-set knowledge ?

How do you feel about the system knowledge being subjective and personal ?

Can you think of a more objective way of doing this ?

How do you feel about translating between the media of words and colours ?

Additional comments:

Generating colour concepts:

How did you find generating colour concepts from single descriptors ?

How did you find generating colour concepts from more than one descriptor ?

How does the visual effect change if colours are generated in a different order ?

Were the colours generated what you expected having taught the system yourself ?

Additional comments:

Describing colour concepts:

How did you find describing colour concepts for a single colour ?

How did you find describing colour concepts for more than one colour ?

Did describing colour concepts help you to refine the current descriptors ?

Were the associated descriptors what you expected having taught the system yourself ?

Additional comments:

Operating methodology:

Did you start from colours first or descriptors first in generating a colour concept ?

Which was the most effective ?

Did the process become automatic after a while ?

How effective is the iterative process 'to-ing and fro-ing' between generate and describe ?

What happened in colour concepts where colours / descriptors were common ?

Did you find the system fun to use ?

Do you prefer to use a computer system or manual methods to design colour schemes ?

How do you feel about the system including other aesthetic attributes such as form and texture ?

Additional comments:

The design process:

Where do you think that the system should be used in the design process ?

Did the system provide a stimulus for ideas for colour schemes ?

Can you see this system being used in a commercial capacity ?

How effective do you think this system will be as an educational tool ?

Why do you think colour is important in product design ?

How has doing this exercise changed your view of colour in design ?

How do you feel about designing a colour scheme before the product is designed ?

What would you propose as the next stage towards finalising the colour scheme ?

Do you think the system boosts or stifles creativity ?

Additional comments:

Thankyou for your help in evaluating this Ph.D. project.

Bob Eves BEd(Hons), MSc.

Appendix F

Colour Concepts

Plate 3

Desert

barren
sparse
sandy
beach

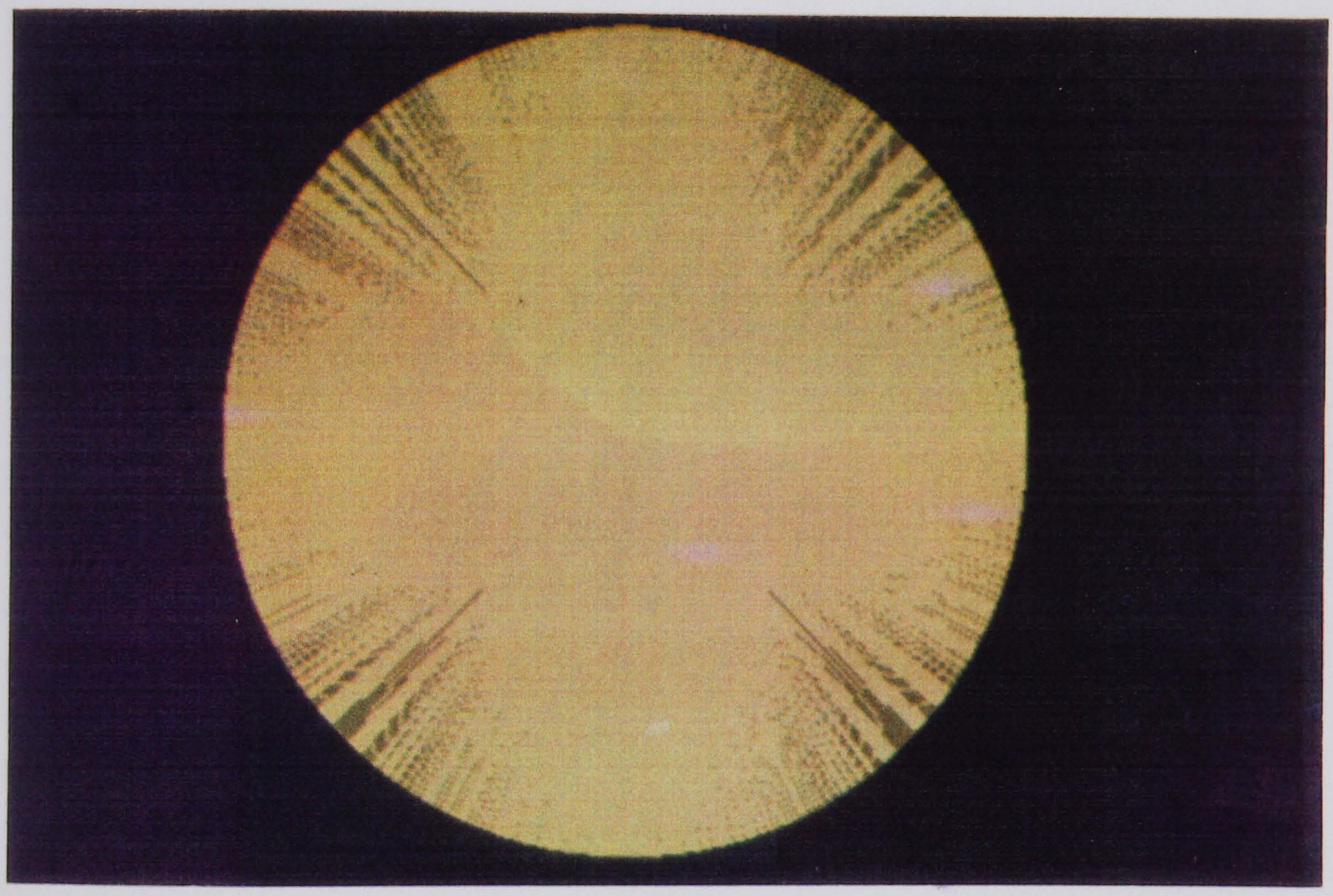


Plate 4

Priest

spiritual
divine
piety
virtuous

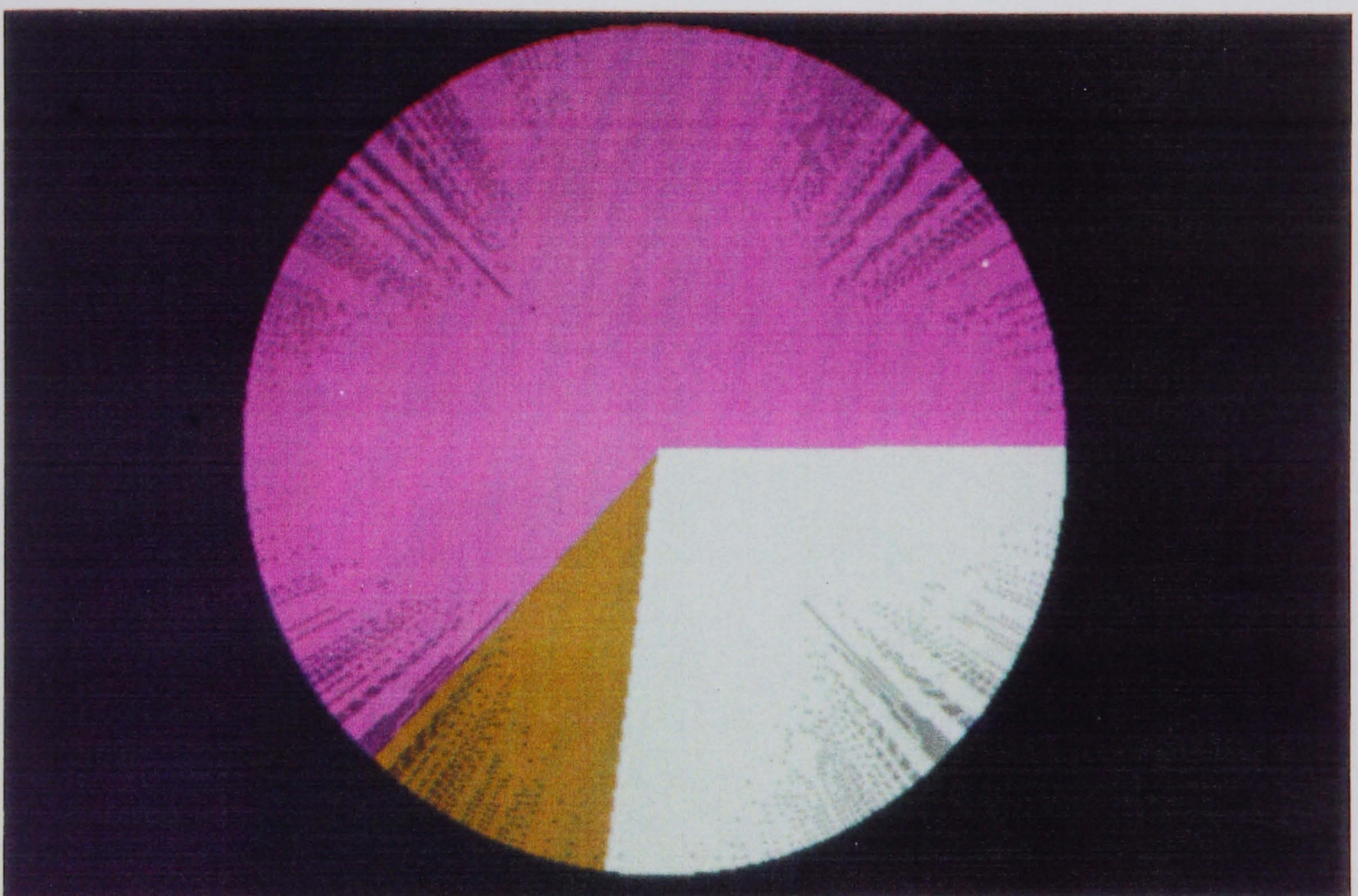


Plate 5

Wasp

poison
toxic
hazard
warning

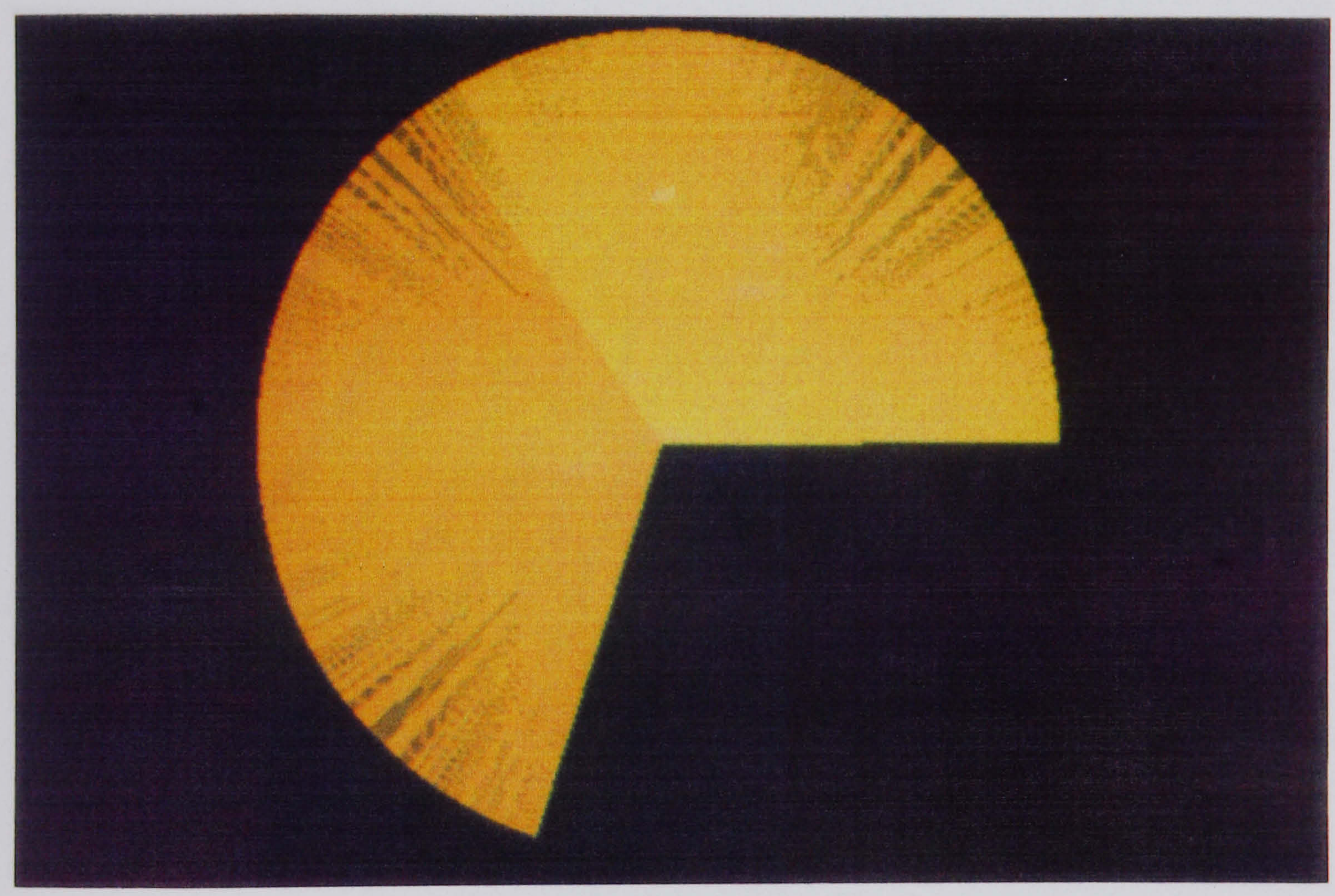


Plate 6

Neon

illuminate
electric
lightning
bright

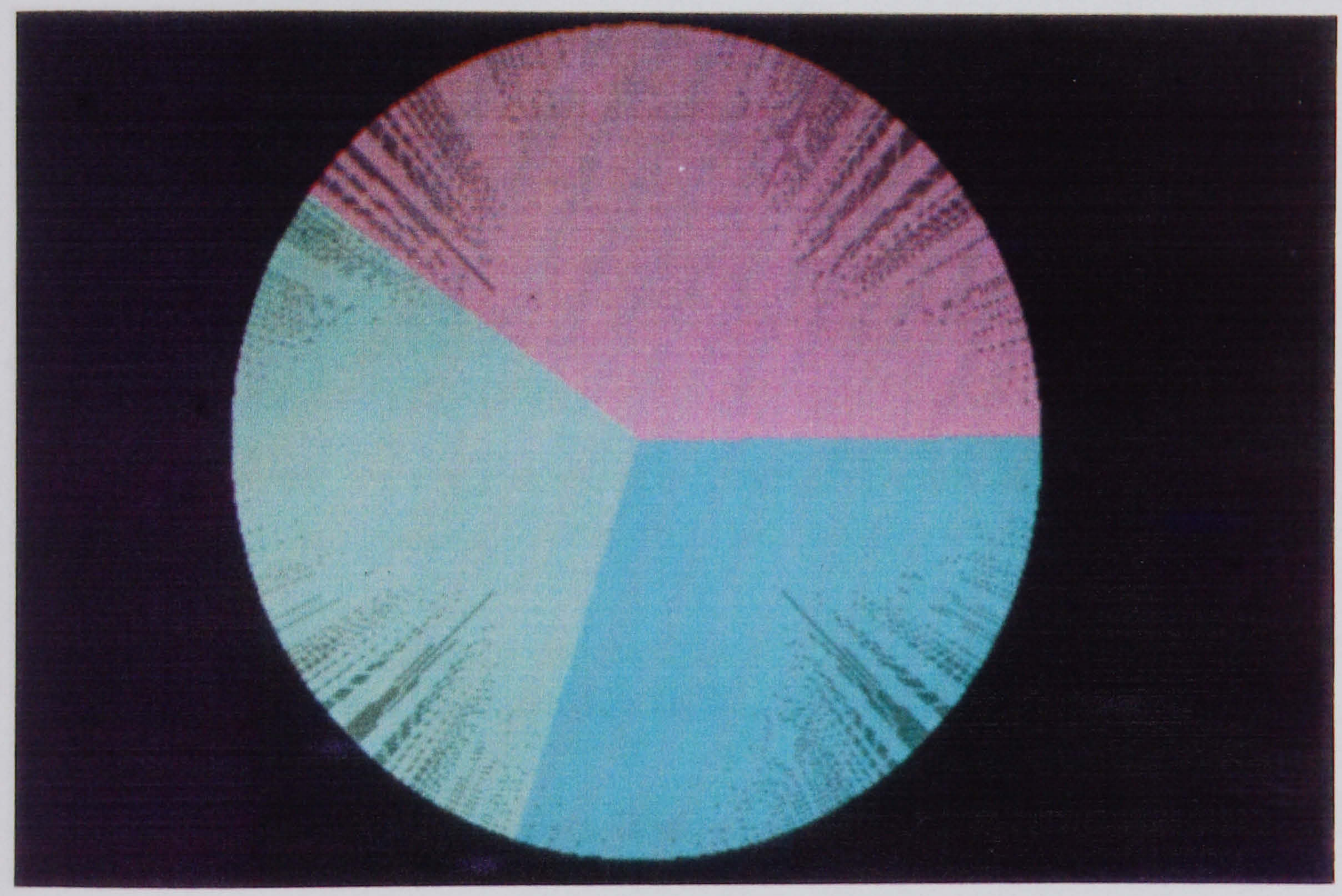


Plate 7

Surgeon

medical
clinical
sterile
hygienic

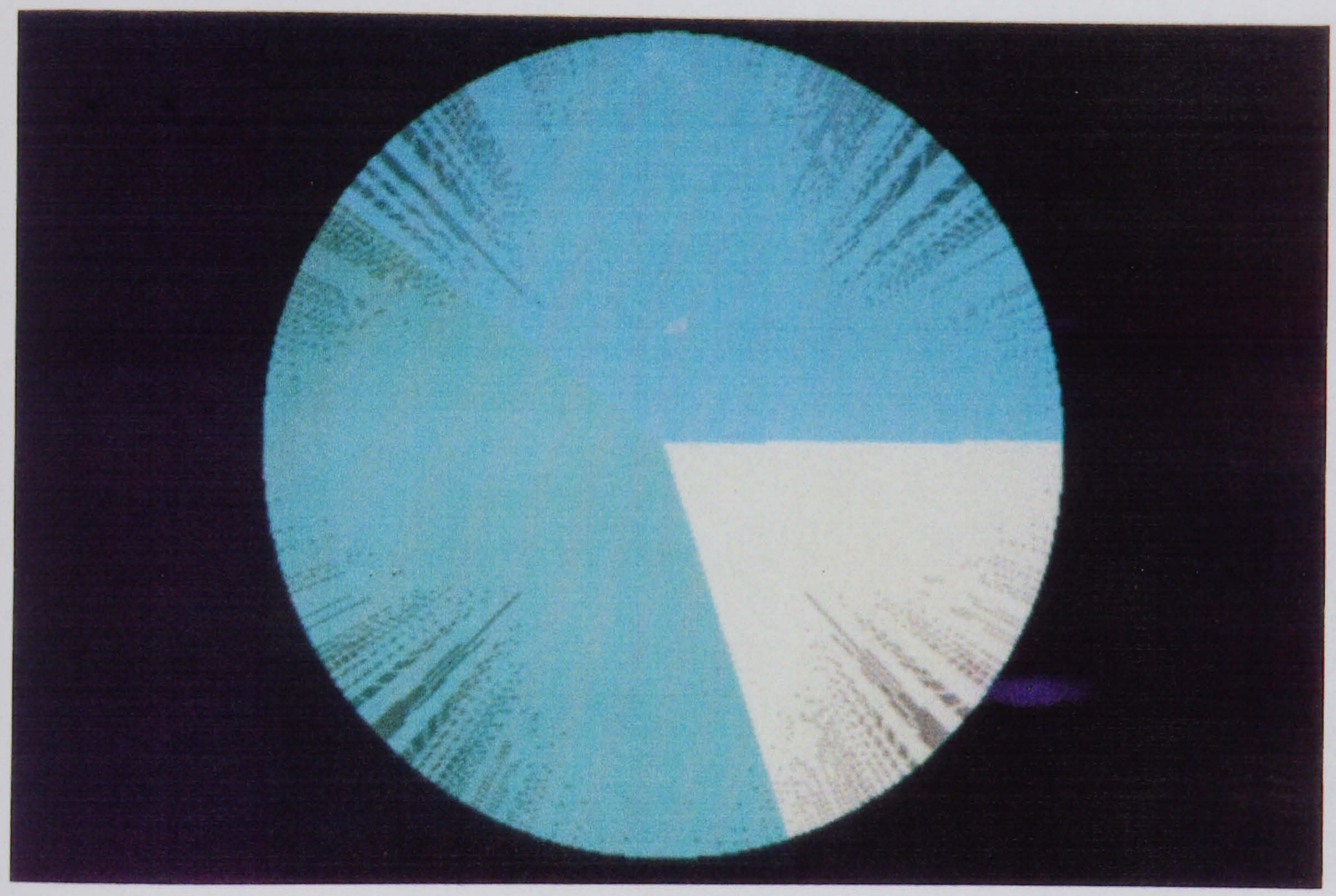


Plate 8

Eggshell

fragile
brittle
smooth
delicate

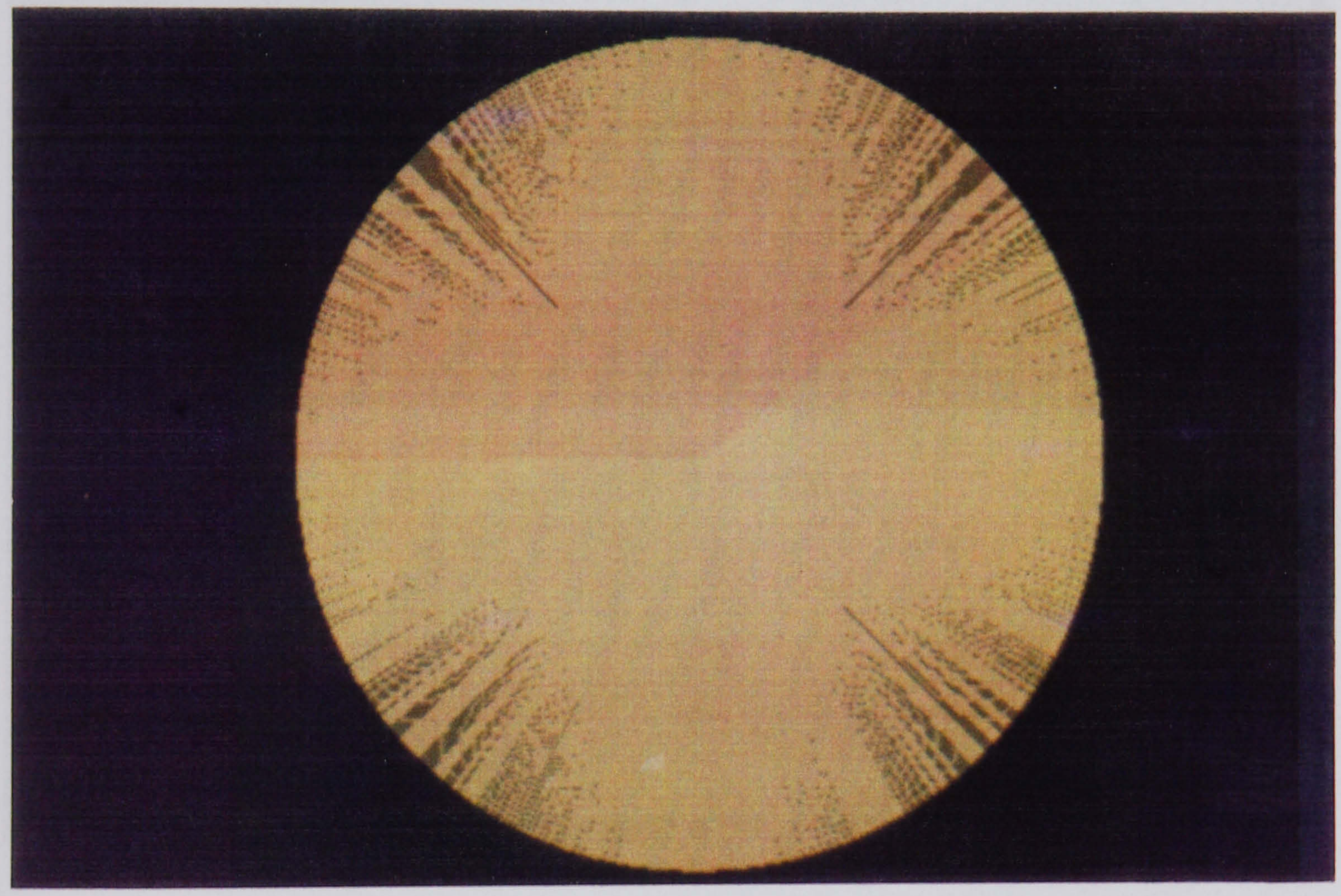


Plate 9

Astronaut

space-age
cosmos
moon
astral

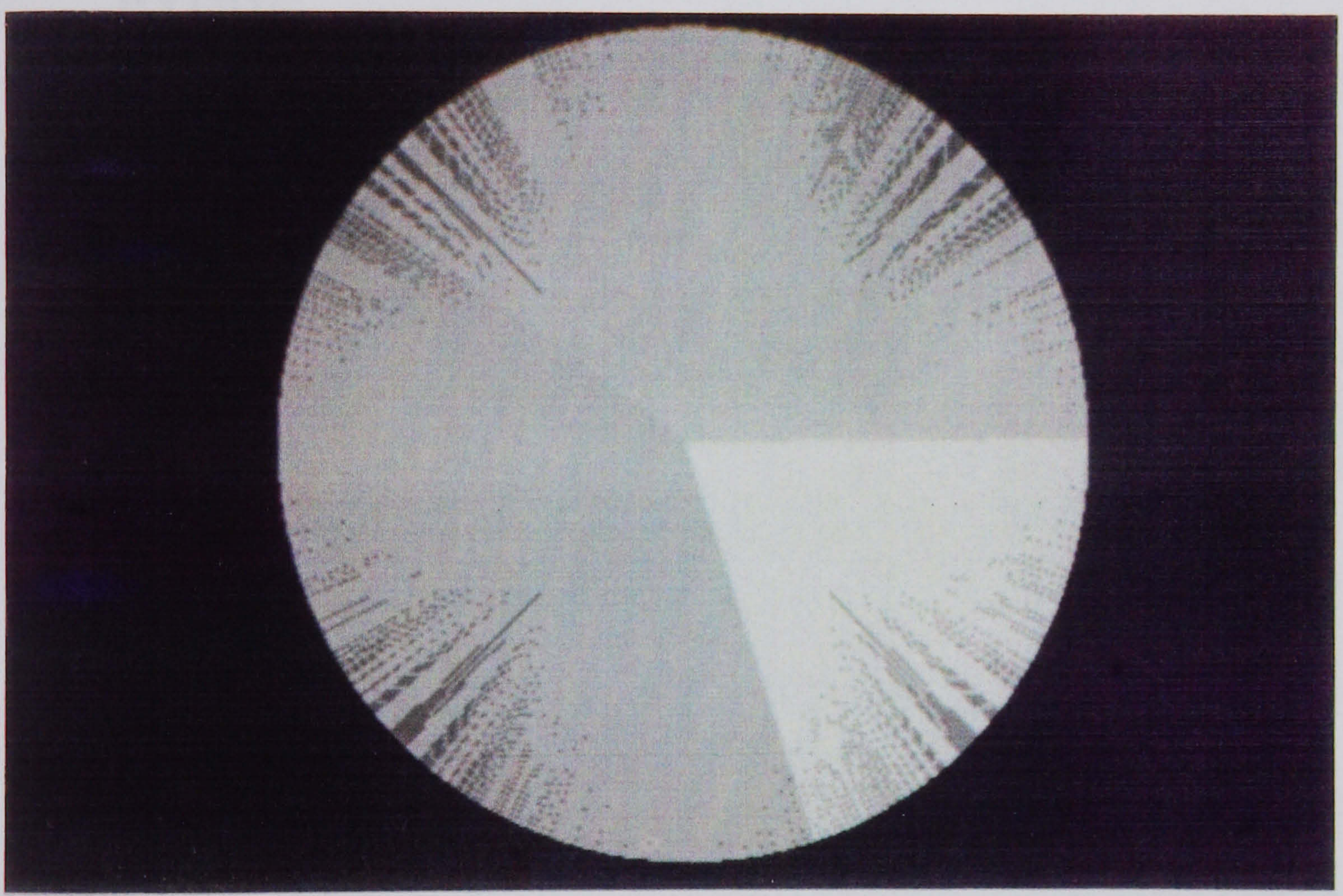


Plate 10

Coral

aquatic
crustacean
reef
exotic

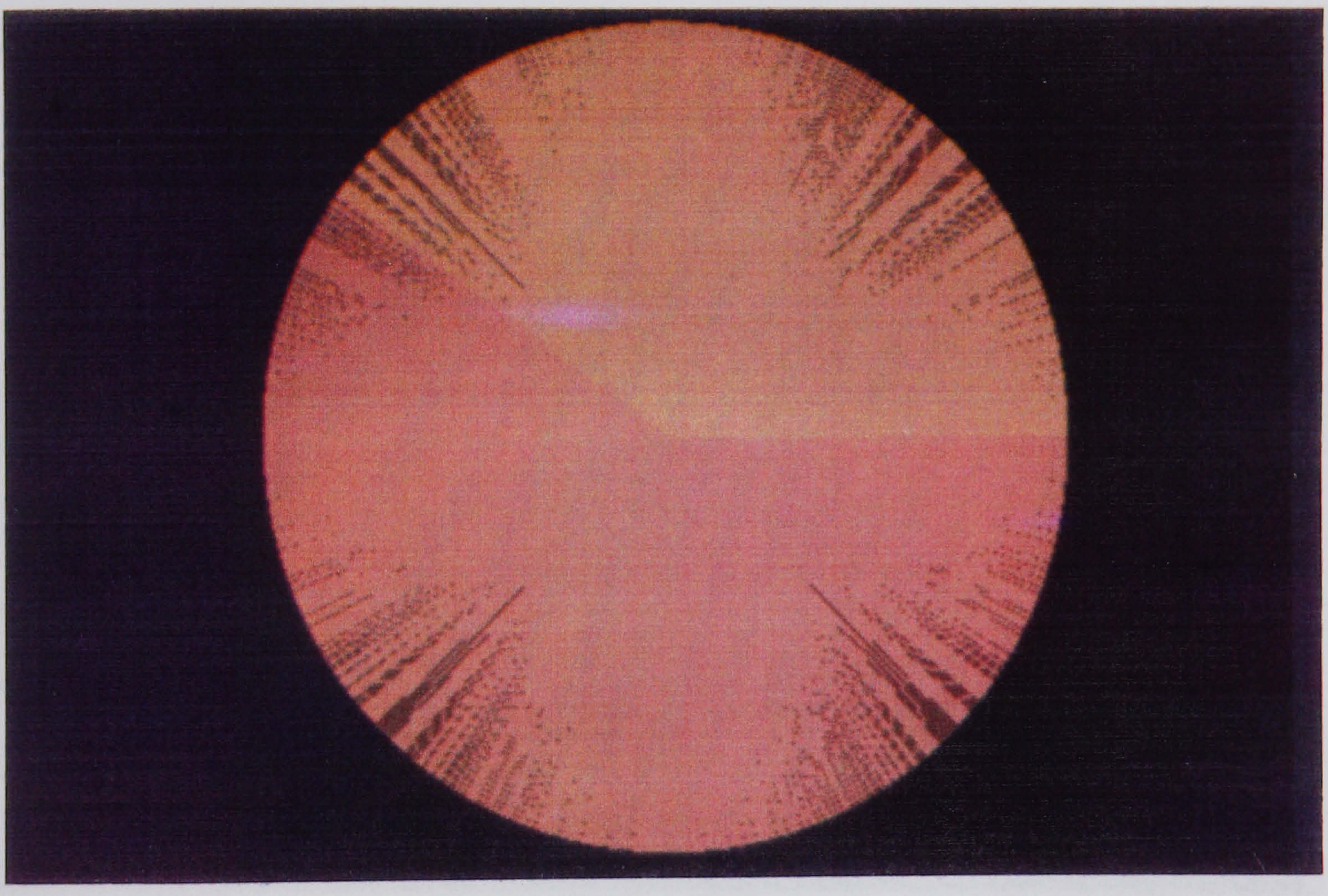


Plate 11

Blood

danger
life
action
power

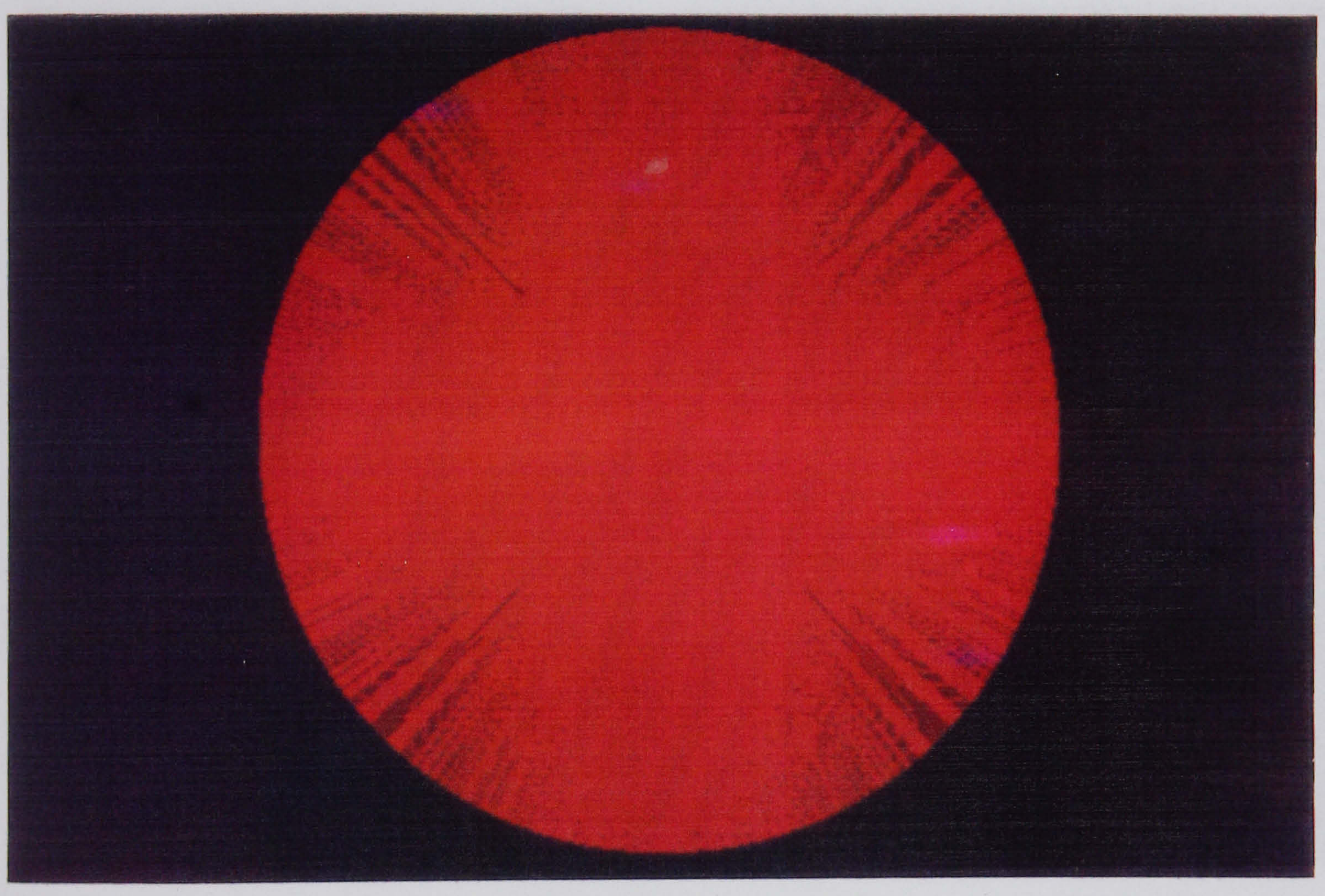


Plate 12

Ocean

aquatic
wave
lagoon
sea

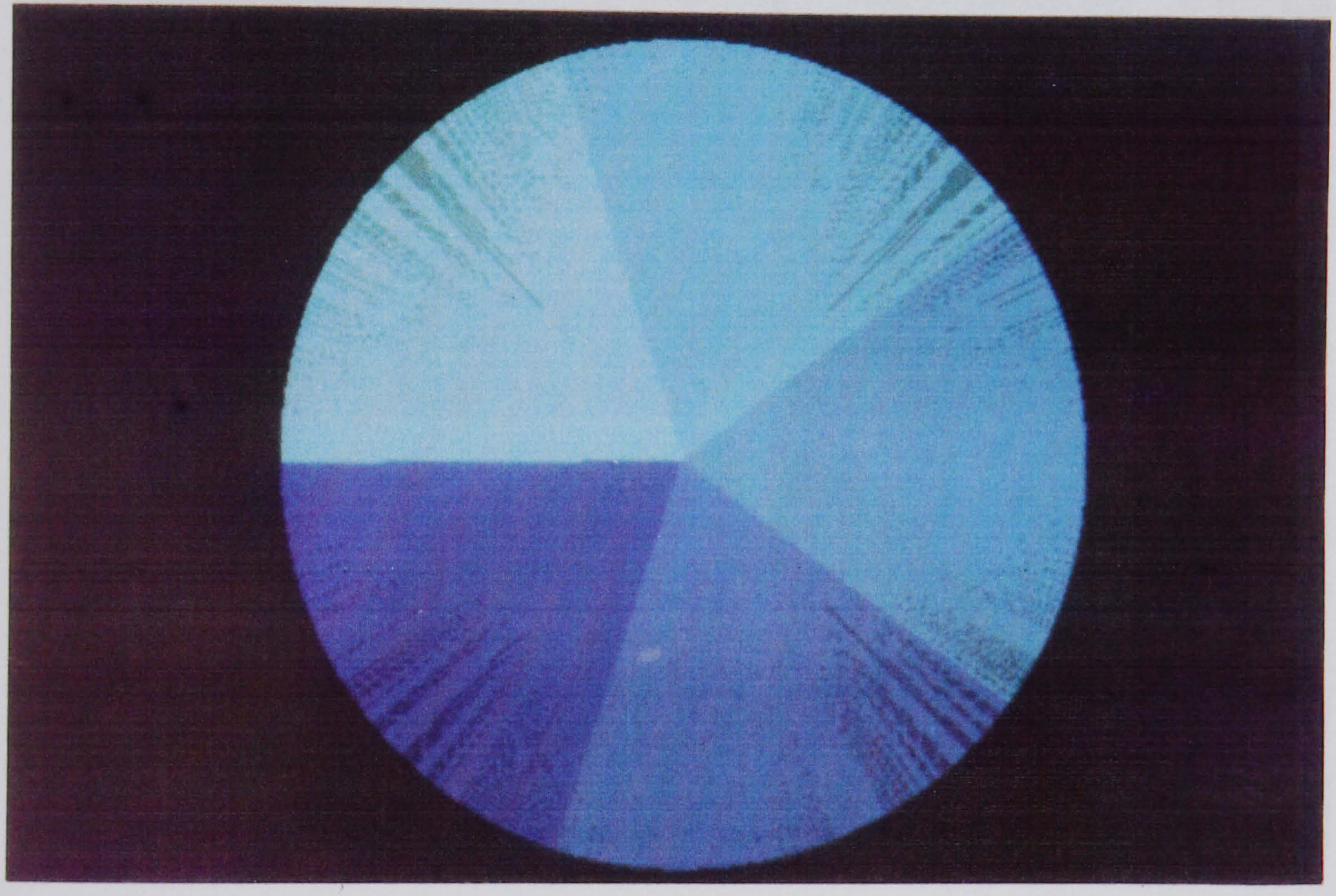


Plate 13

Water Melon

lush
succulent
juicy
ripe

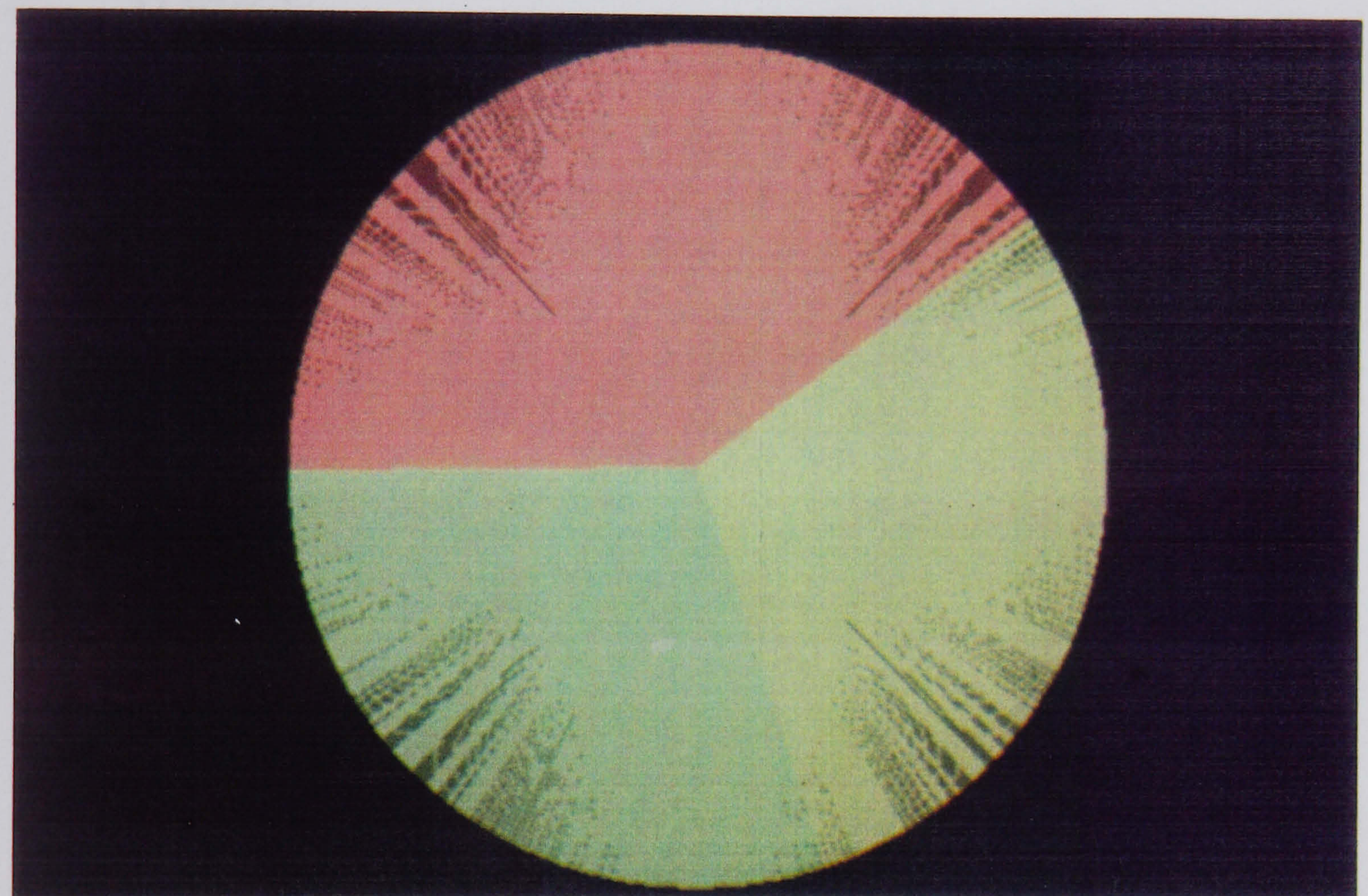


Plate 14

Iceberg

cool
fresh
clear
freeze

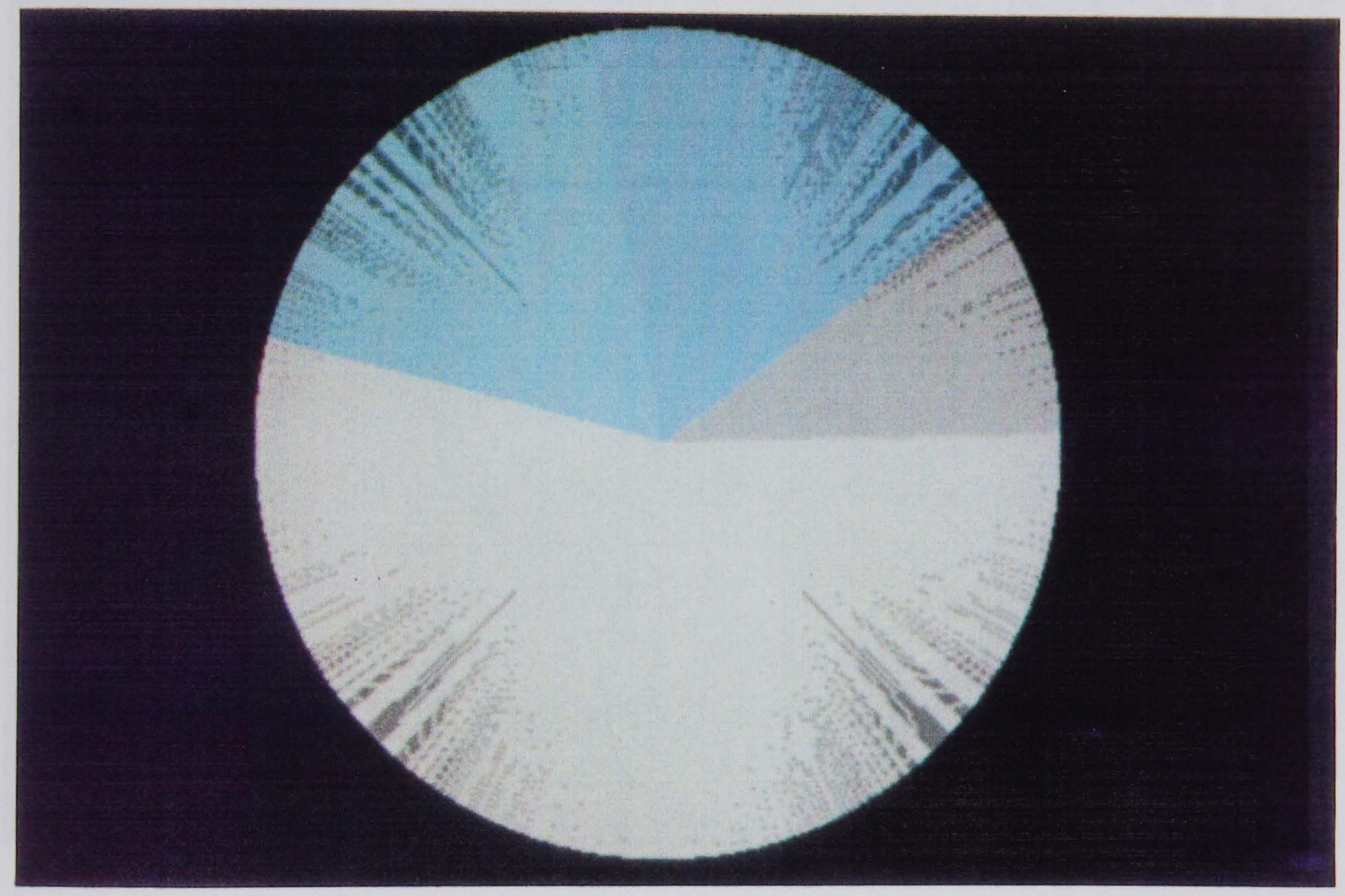


Plate 15

Executive

business
commerce
city
finance

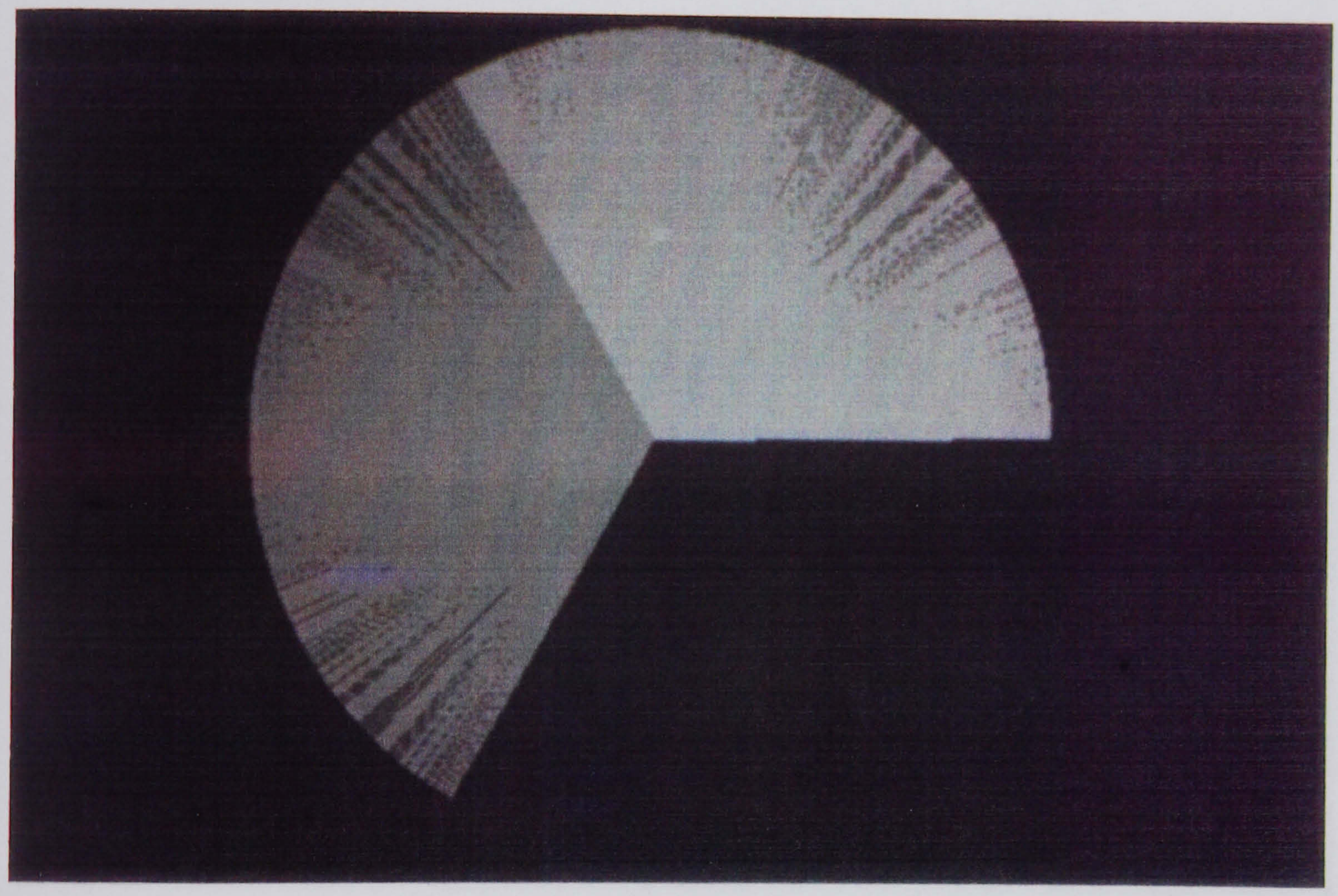
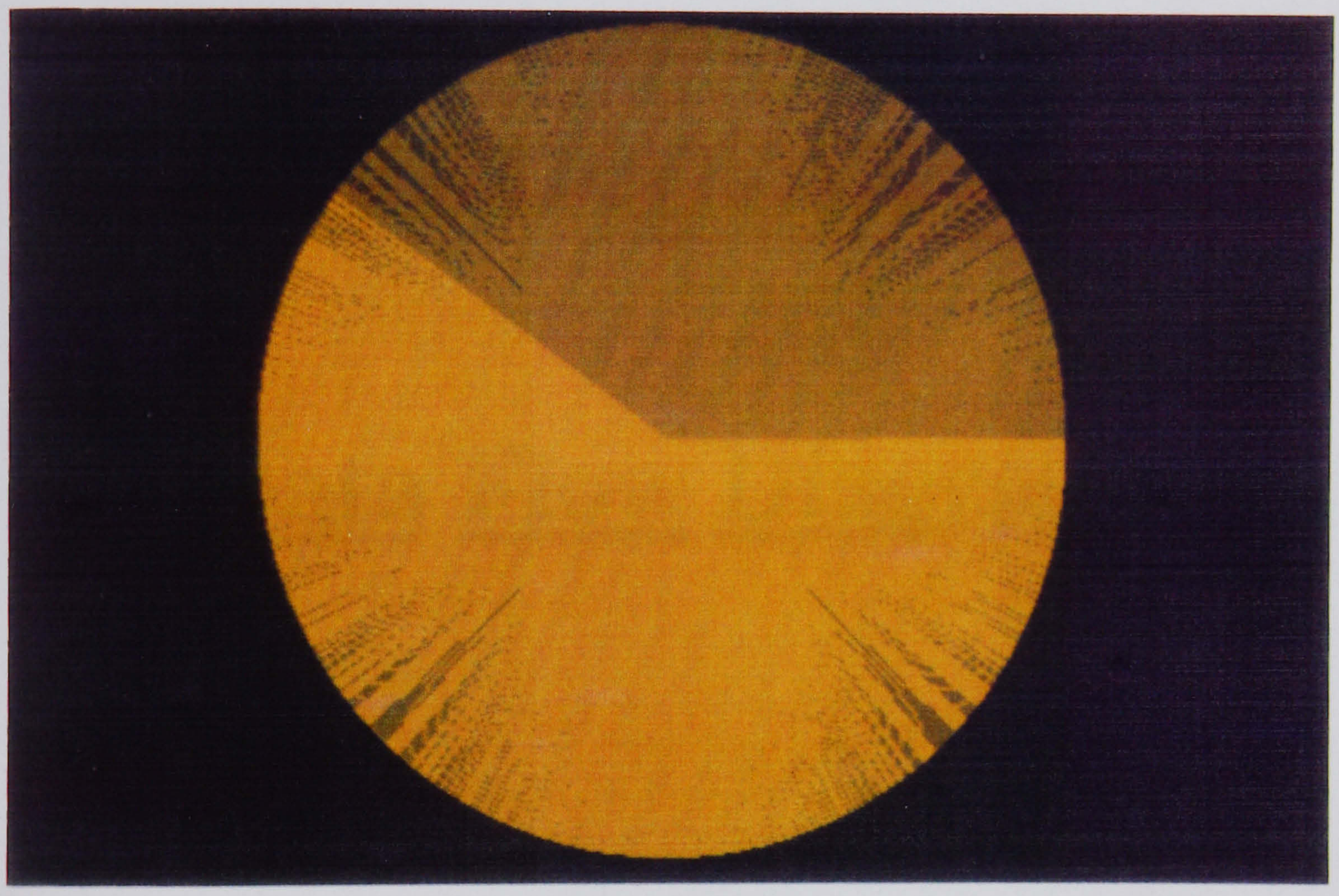


Plate 16

Wheat

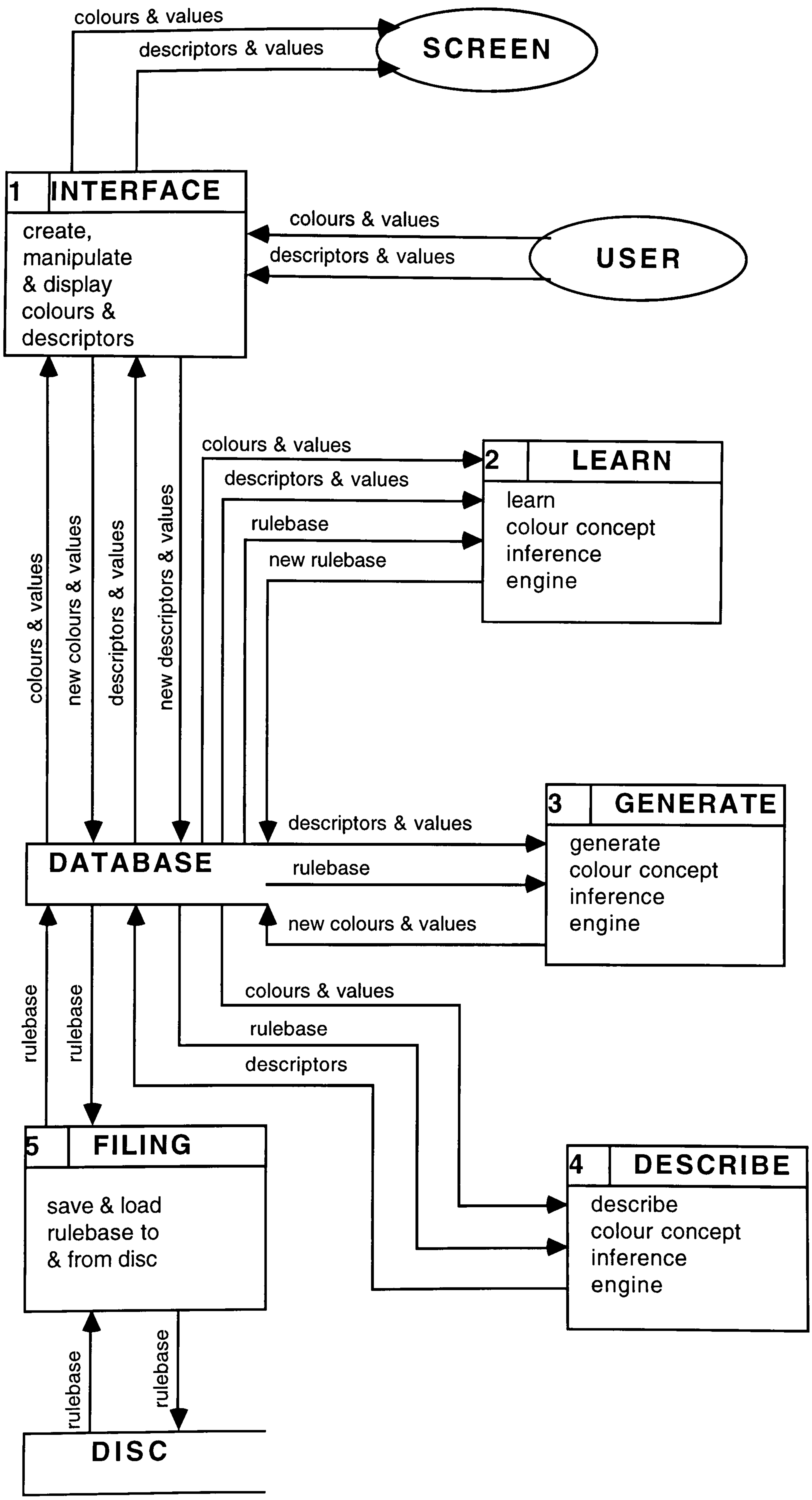
nutrition
vitamin
wholesome
cereal



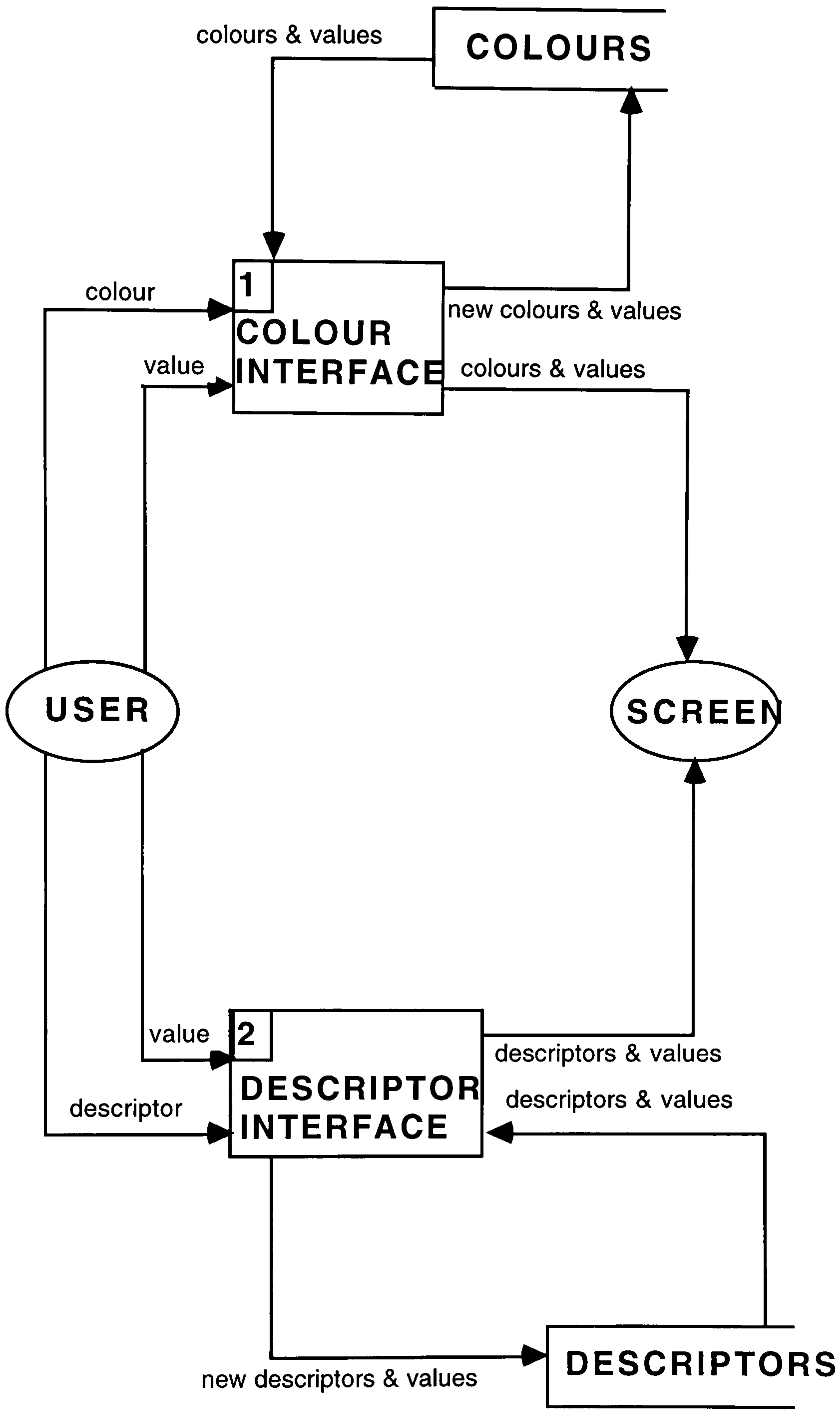
Appendix G

Data Flow Diagrams

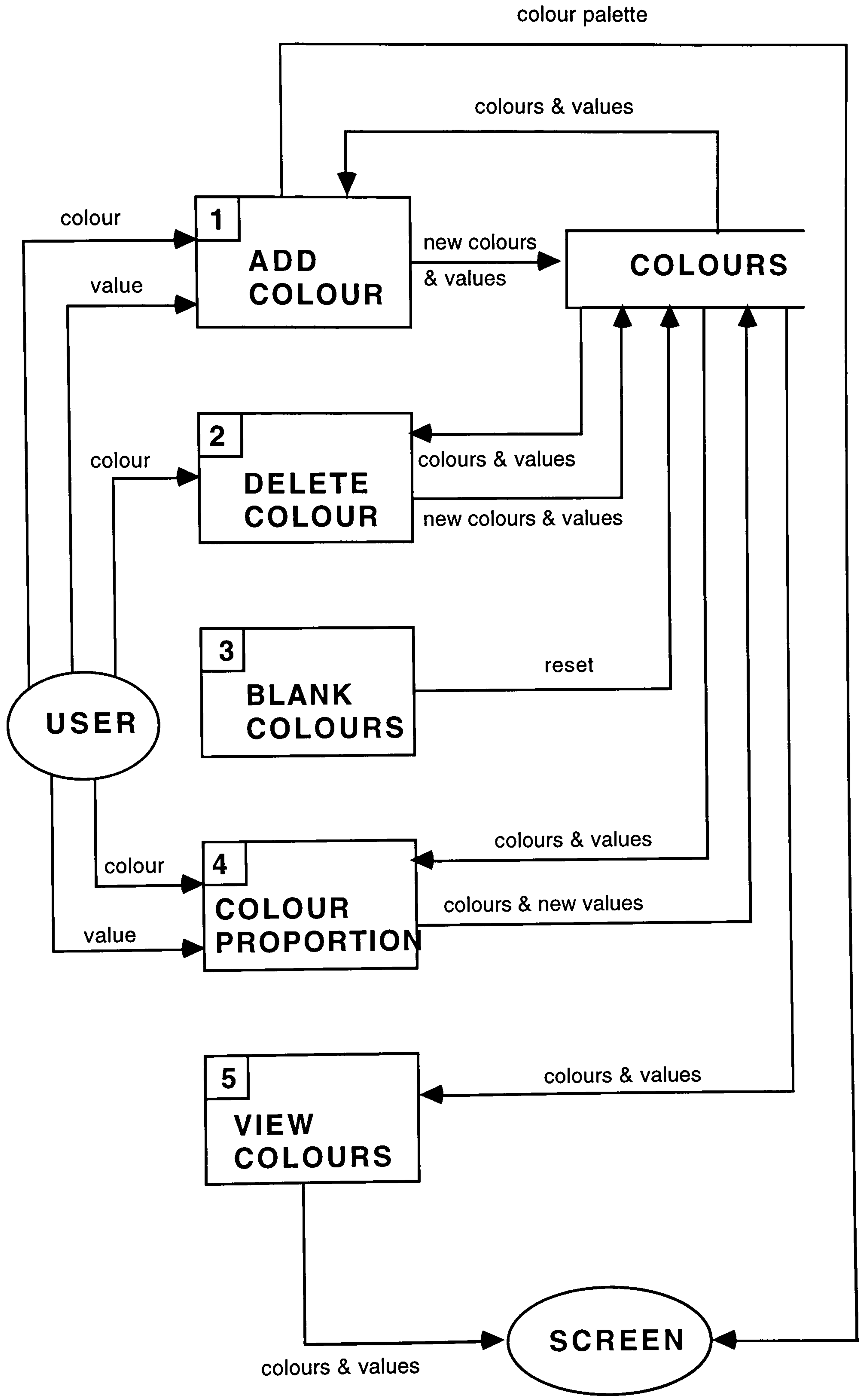
DFD 0 Colour Concept Generator



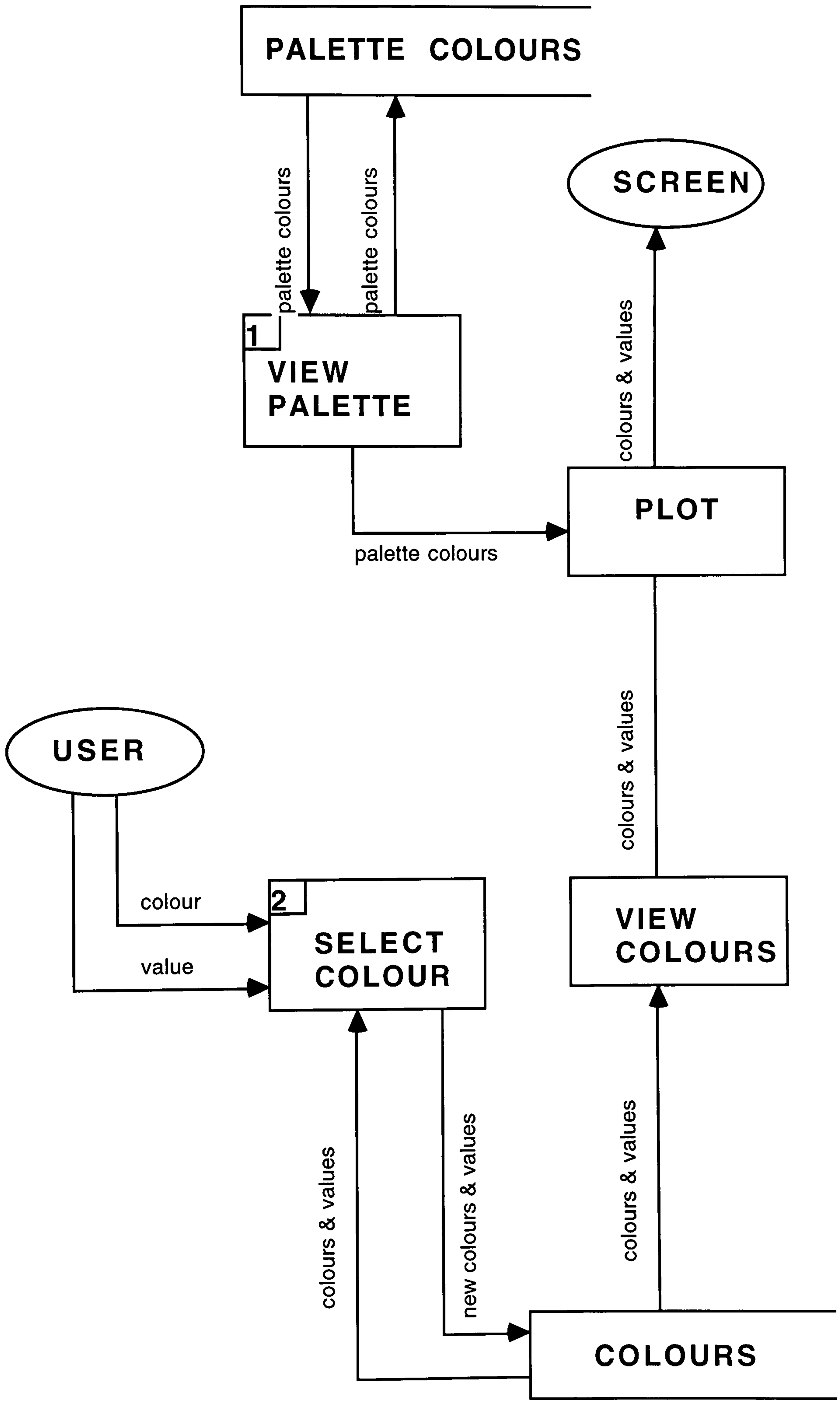
DFD 1 Interface



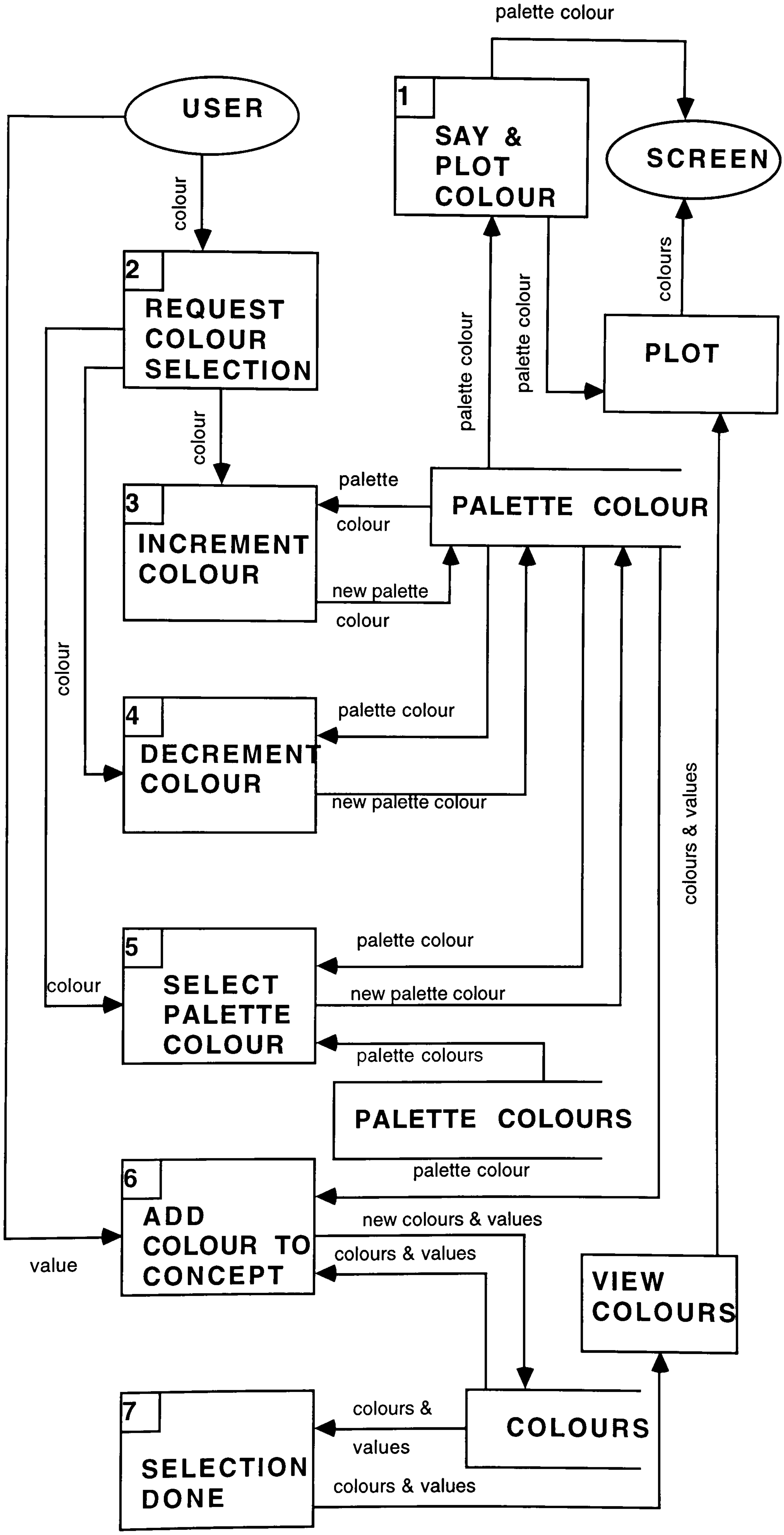
DFD 1.1 Colour Interface



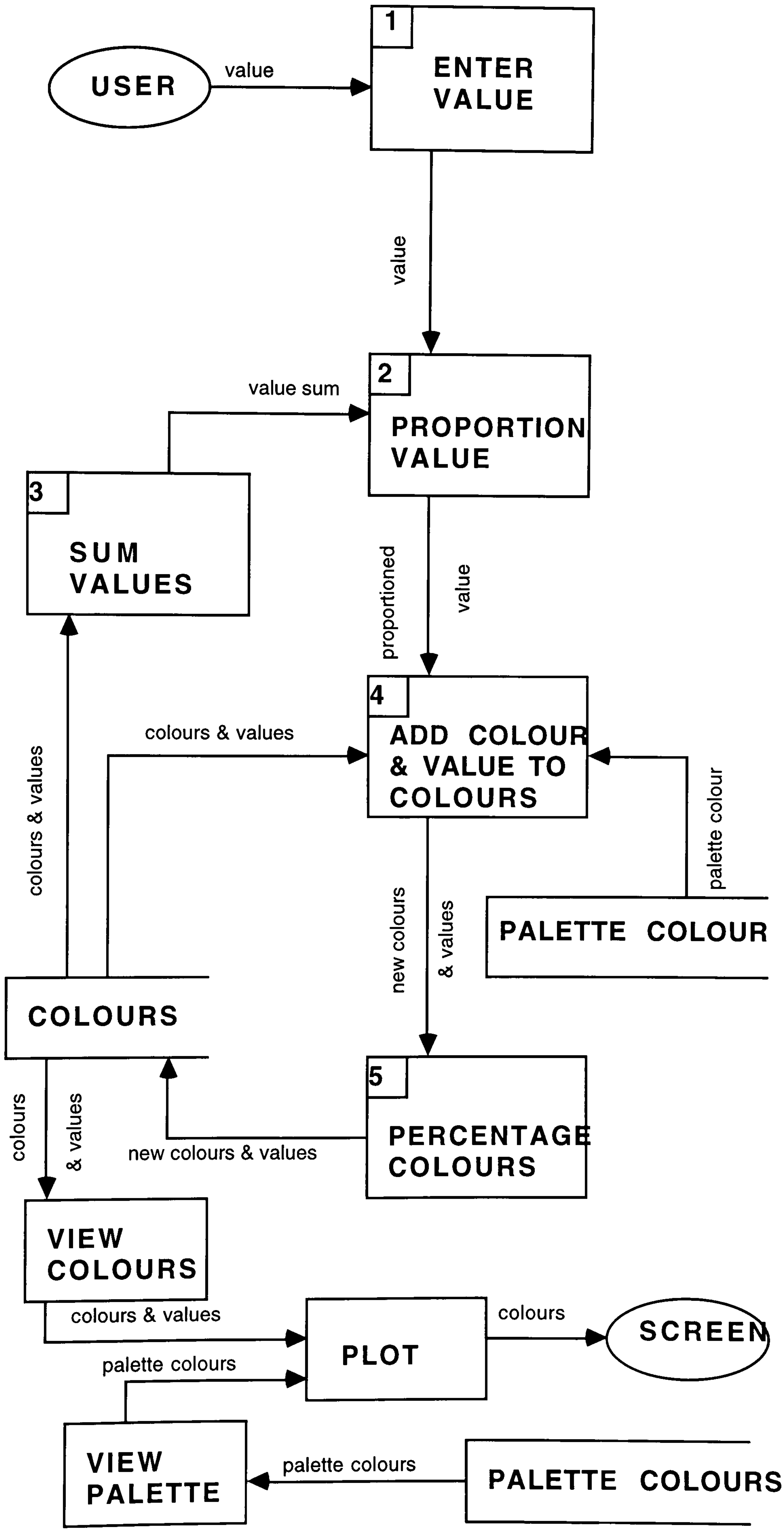
DFD 1.1.1 Add Colour



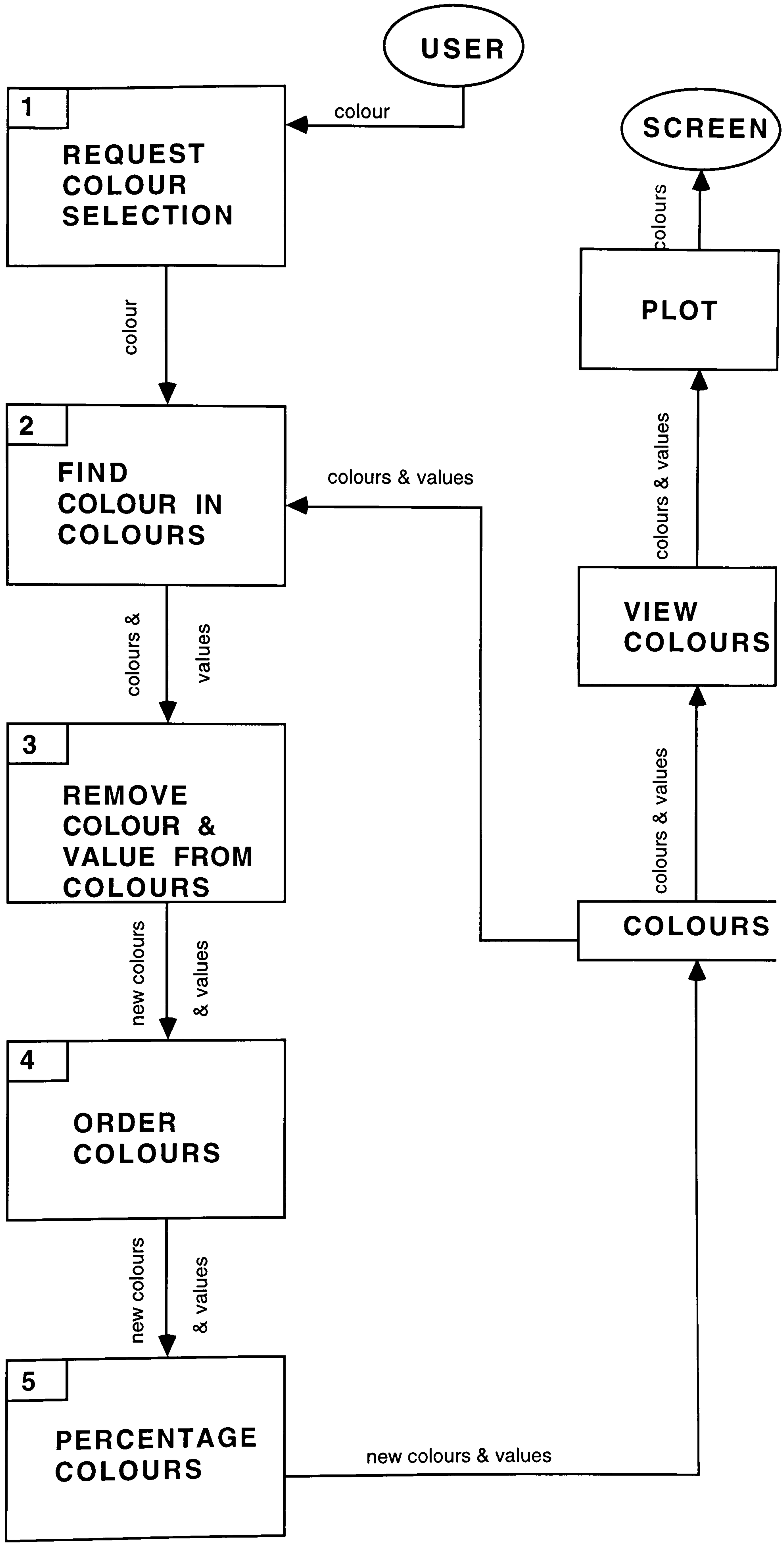
DFD 1.1.1.2 Select Colour



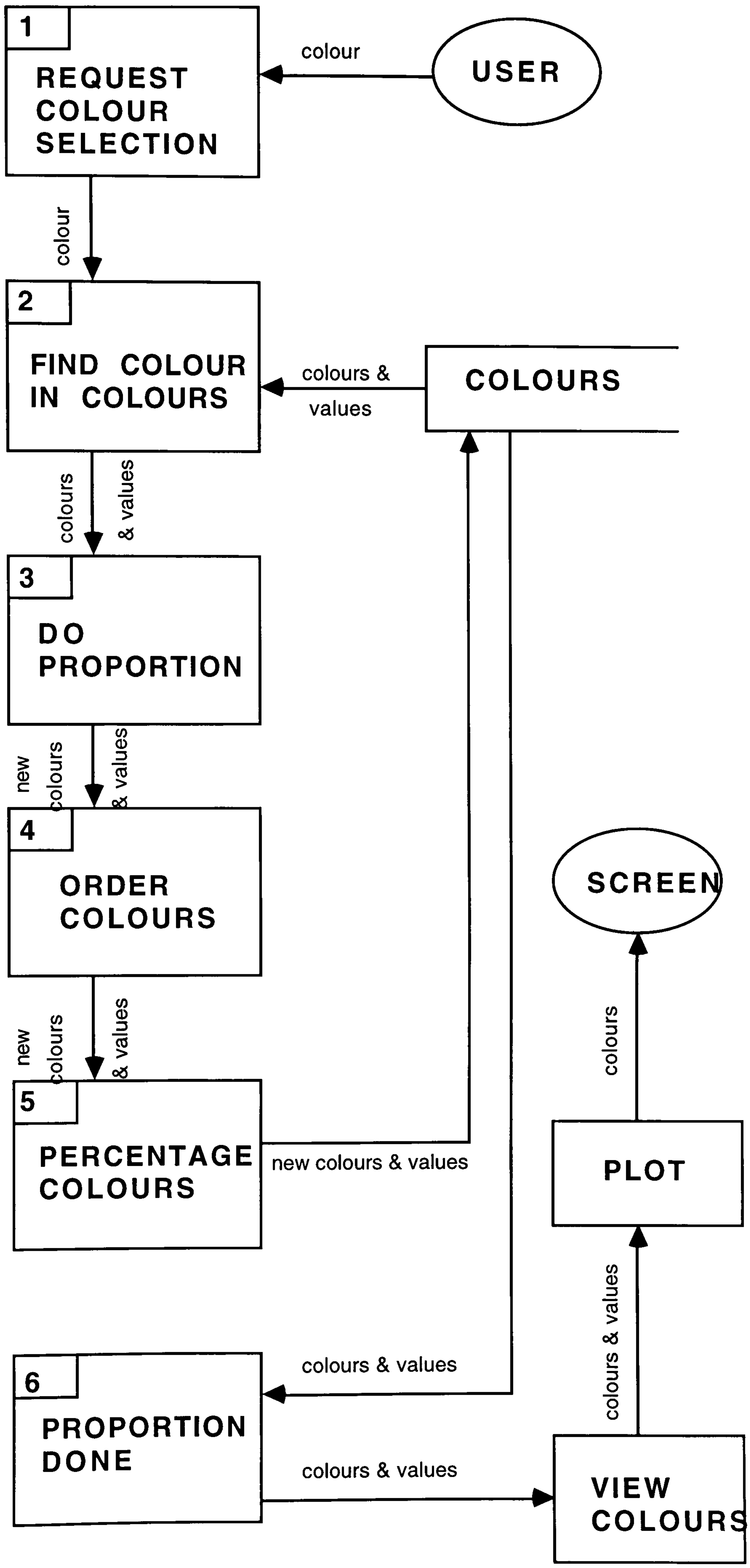
DFD 1.1.1.2.6 Add Colour to Concept



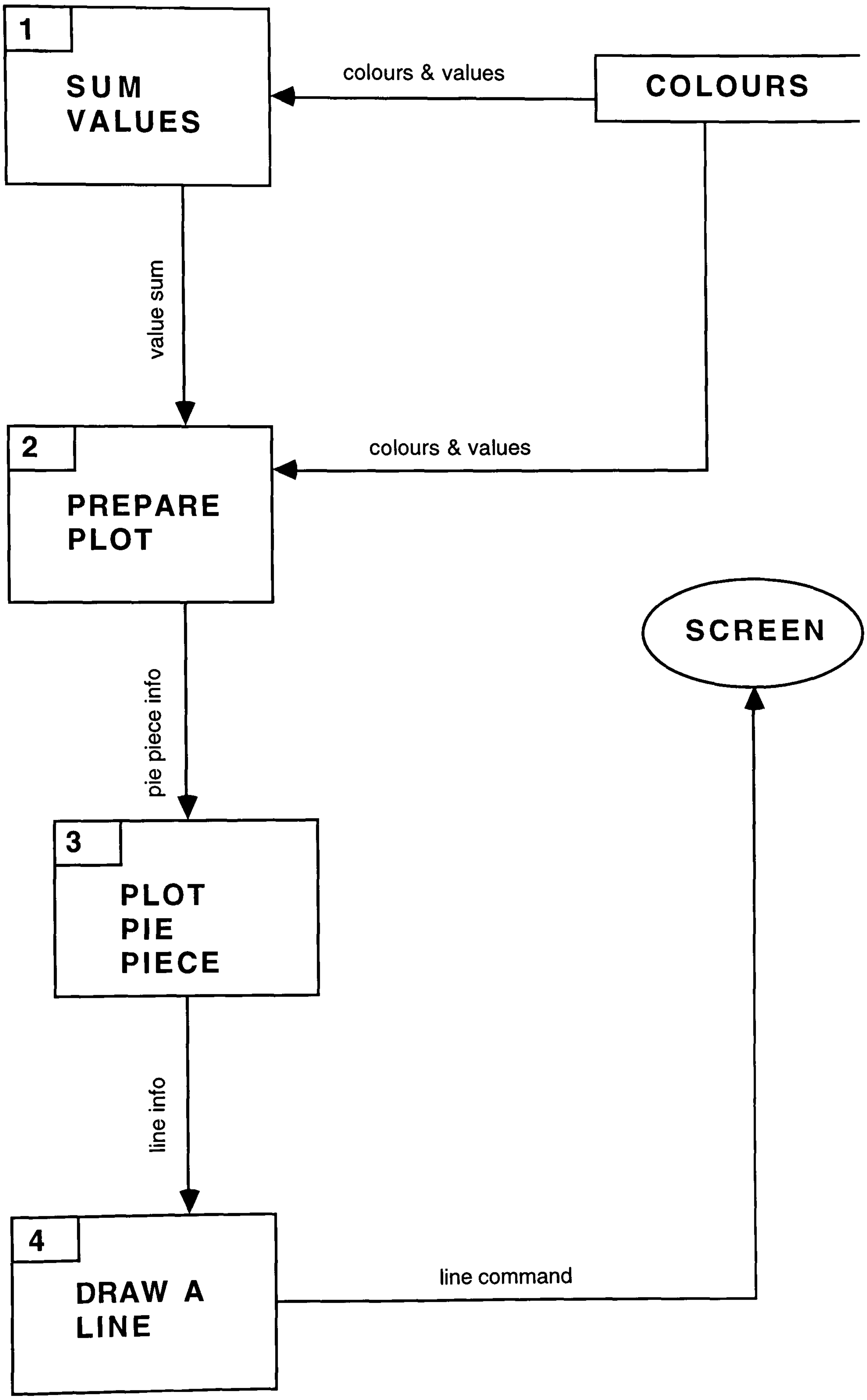
DFD 1.1.2 Delete Colour



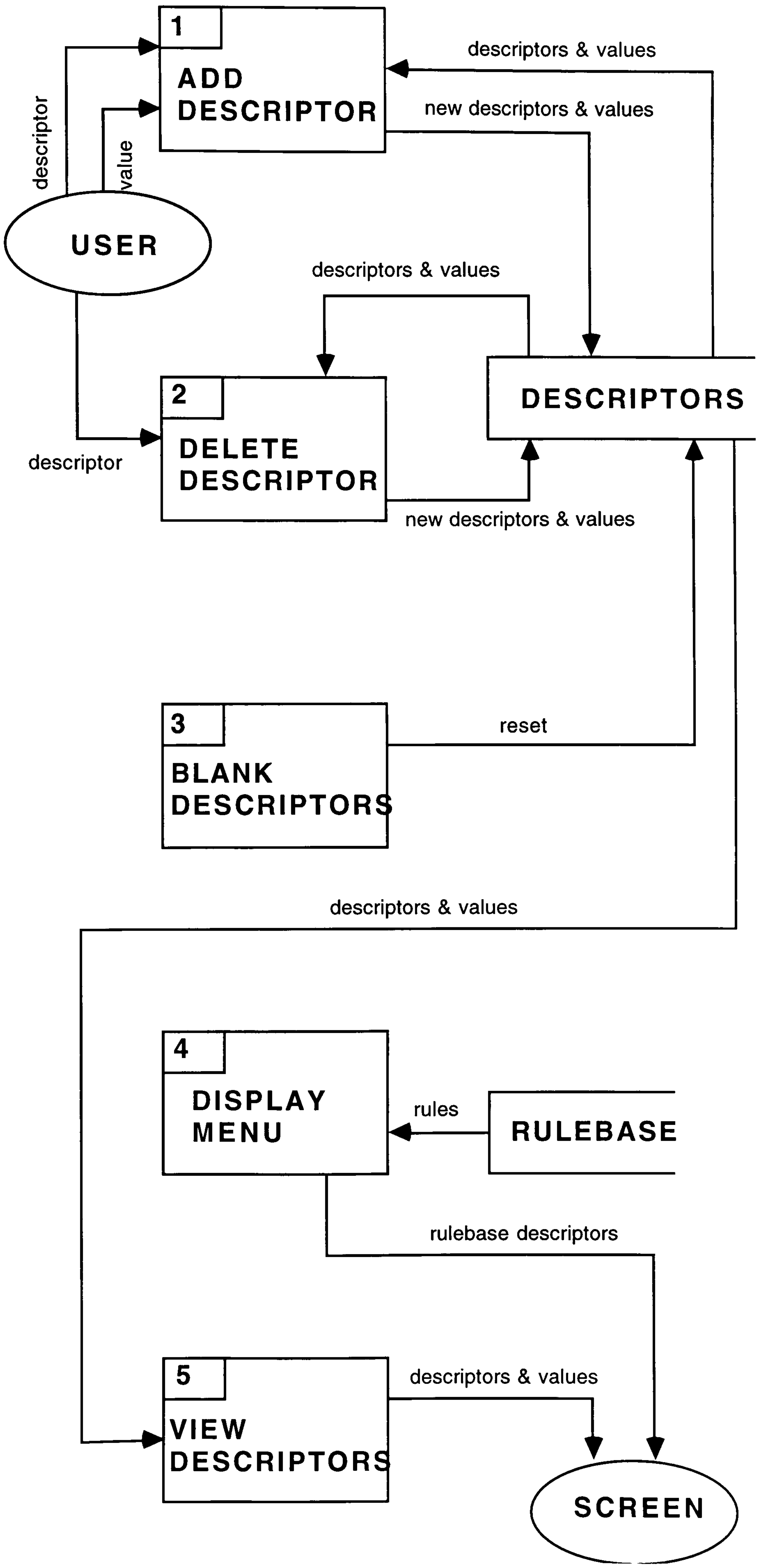
DFD 1.1.4 Proportion Colour



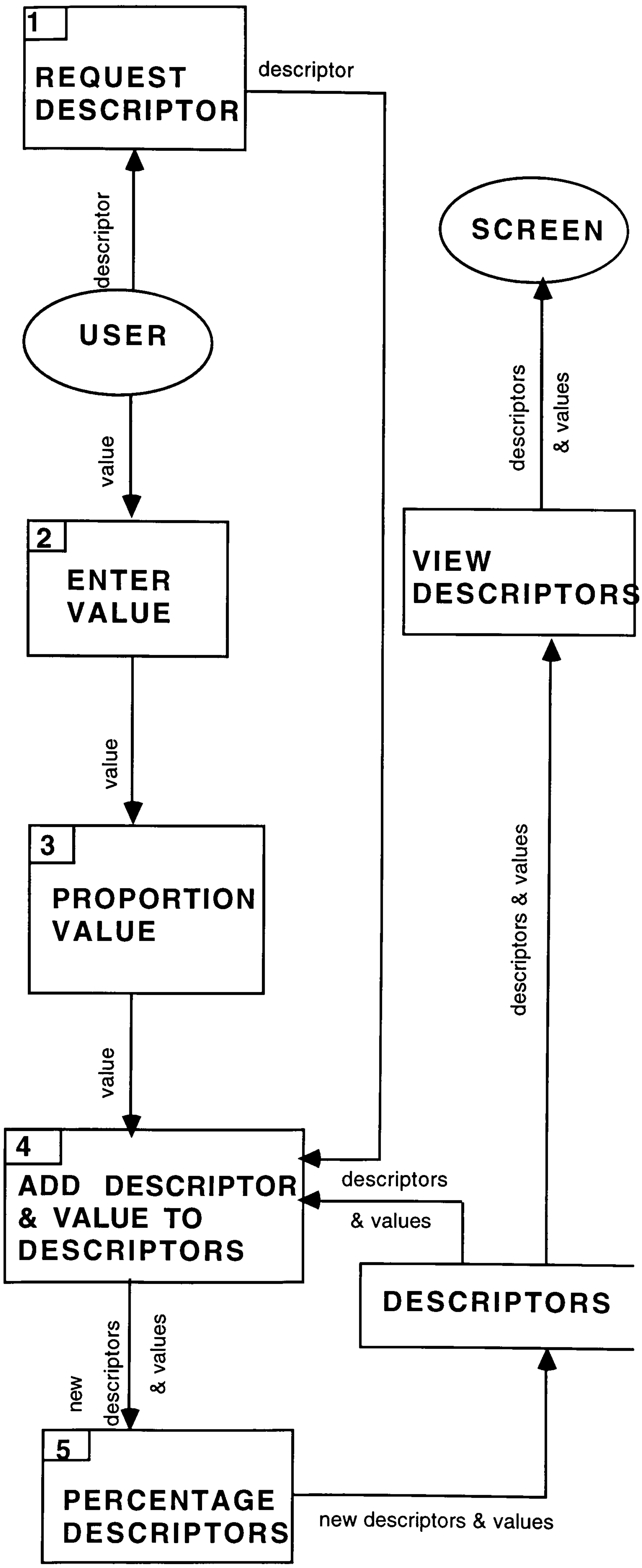
DFD 1.1.5 View Colours



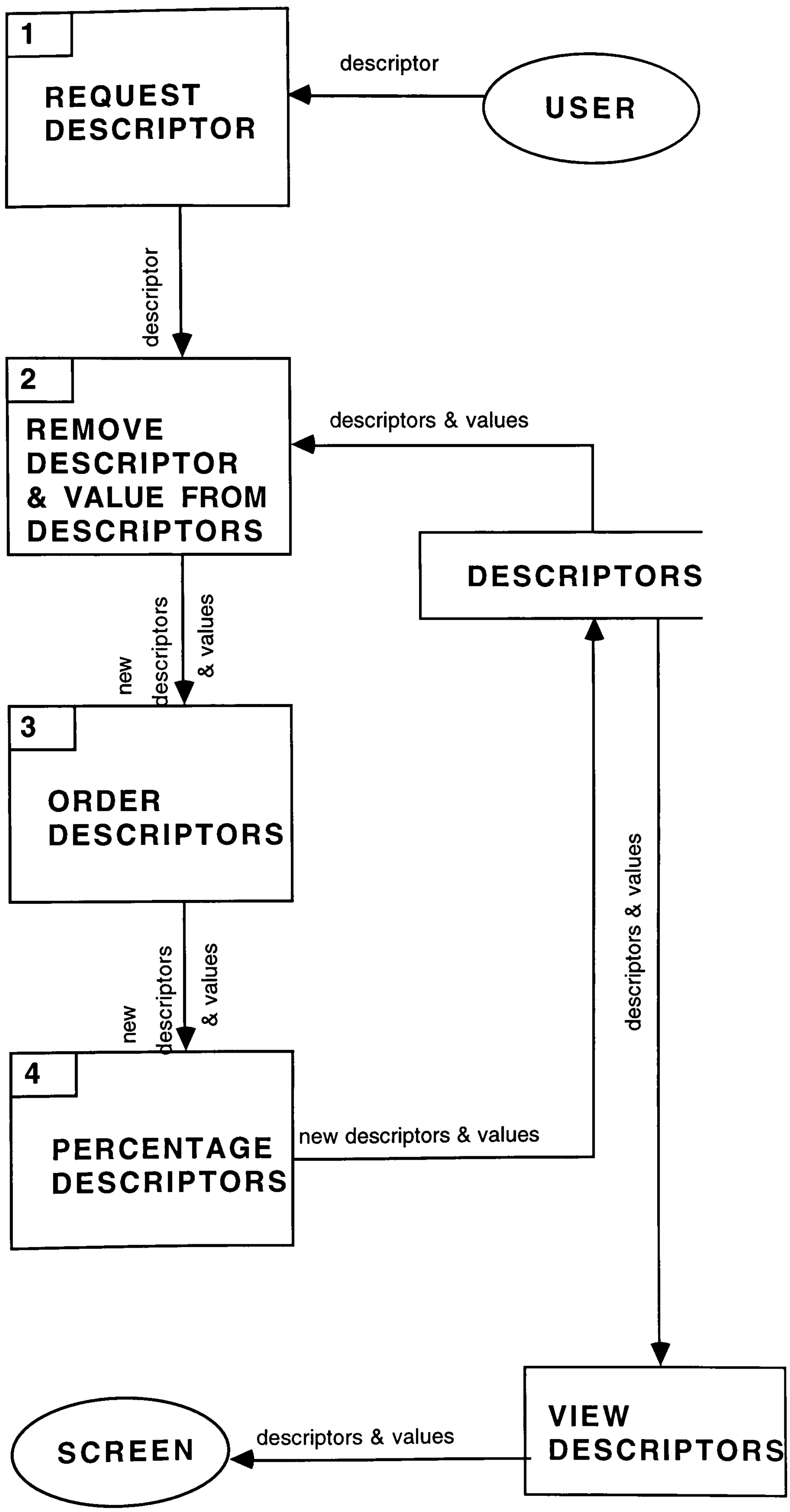
DFD 1.2 Descriptor Interface



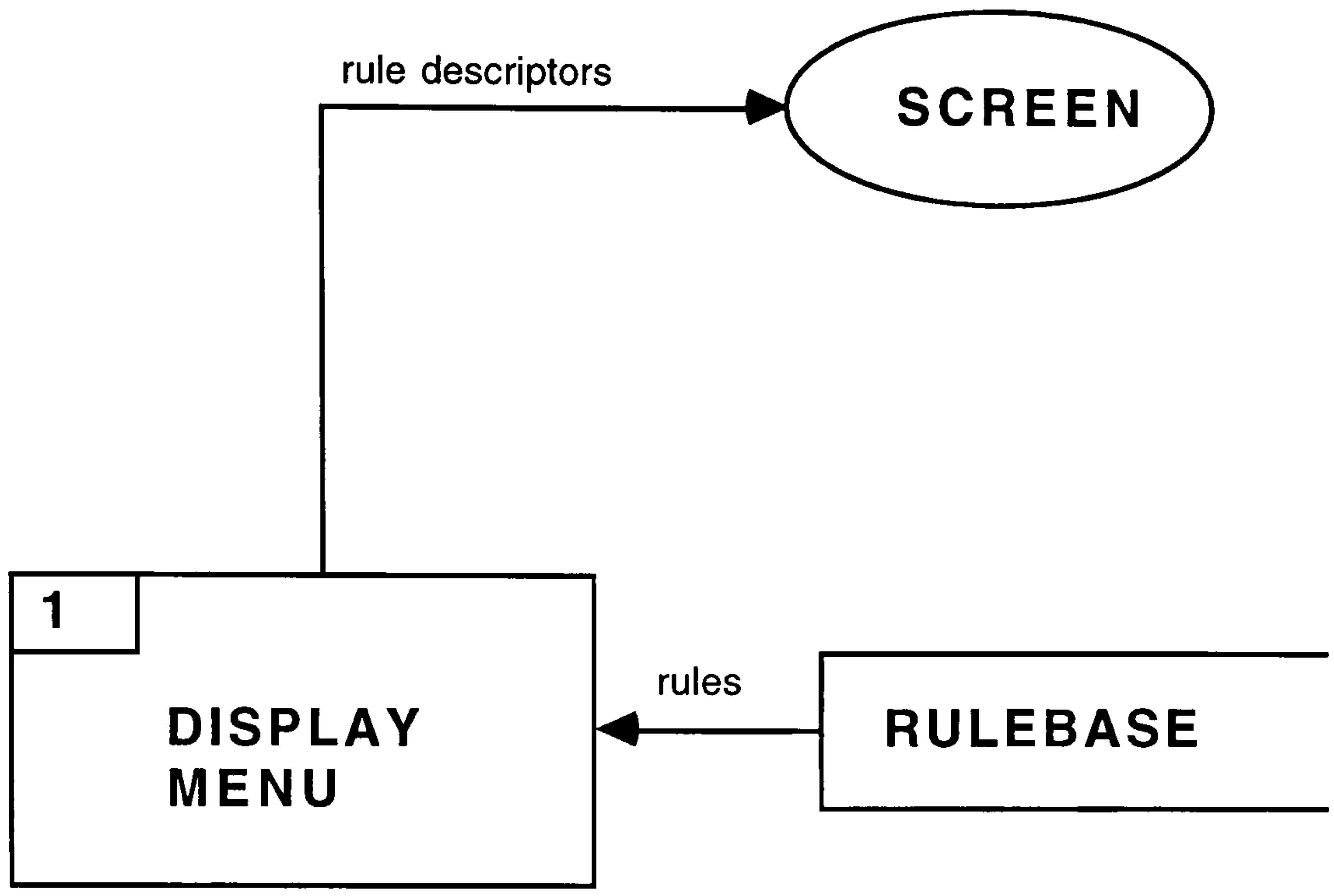
DFD 1.2.1 Add Descriptor



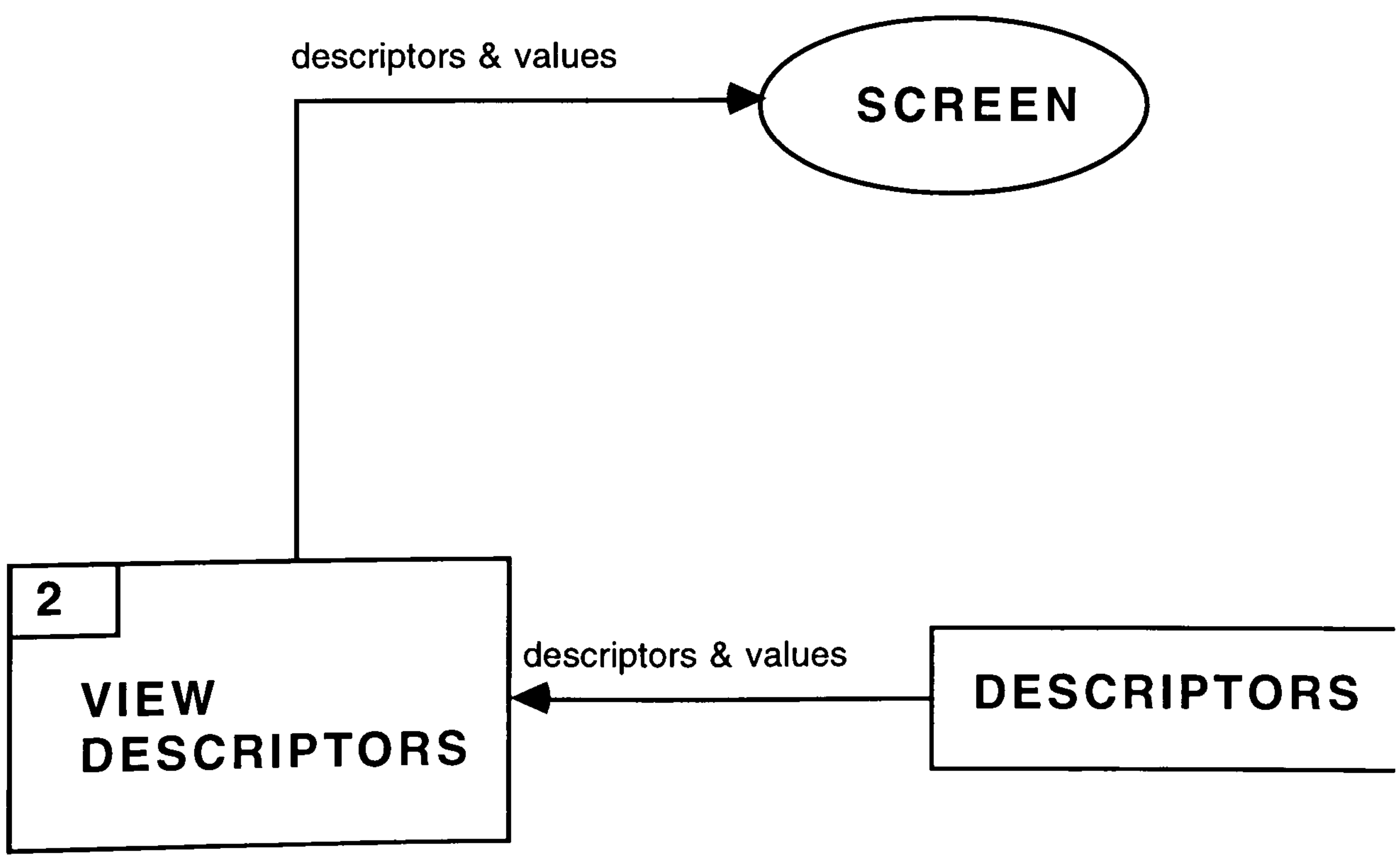
DFD 1.2.2 Delete Descriptor



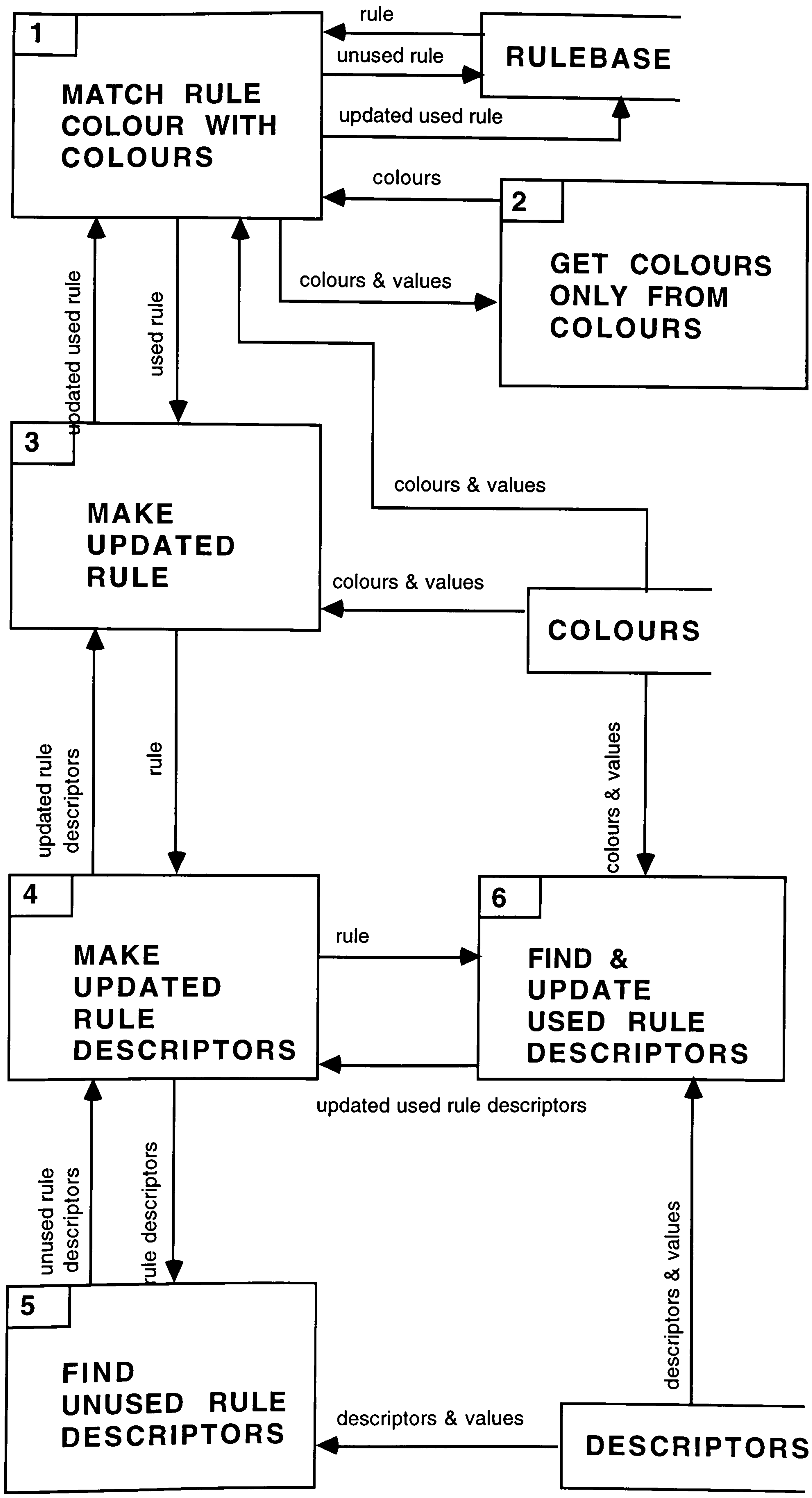
DFD 1.2.4 Display Menu



DFD 1.2.5 View Descriptors

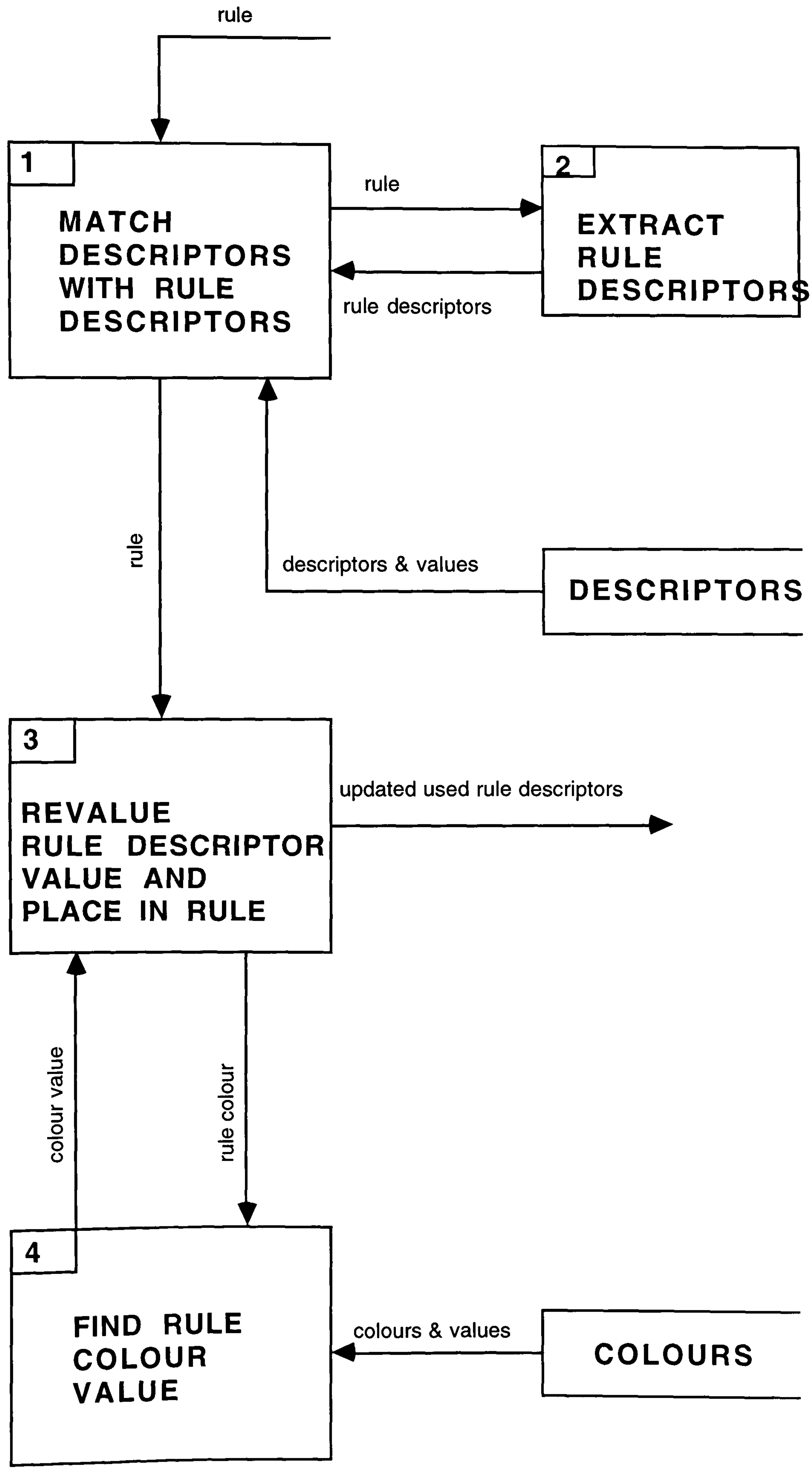


DFD 2 Learn

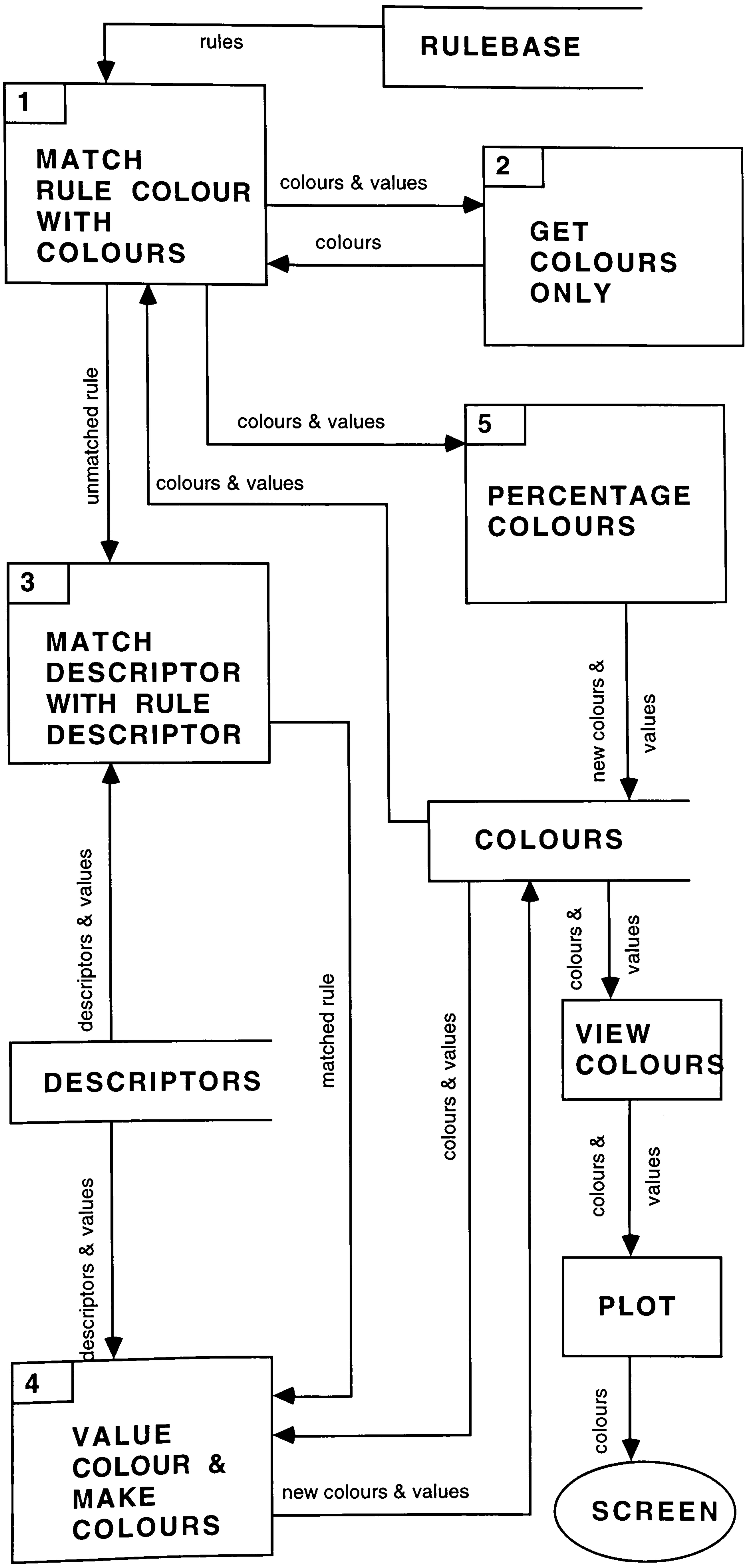


1
MATCH RULE COLOUR WITH COLOURS
rule
unused rule
RULEBASE
updated used rule
colours
2
GET COLOURS ONLY FROM COLOURS
colours & values
updated used rule
used rule
3
MAKE UPDATED RULE
colours & values
colours & values
COLOURS
updated rule descriptors
rule
colours & values
4
MAKE UPDATED RULE DESCRIPTORS
rule
6
FIND & UPDATE USED RULE DESCRIPTORS
updated used rule descriptors
unused rule descriptors
rule descriptors
descriptors & values
5
FIND UNUSED RULE DESCRIPTORS
descriptors & values
DESCRIPTORS

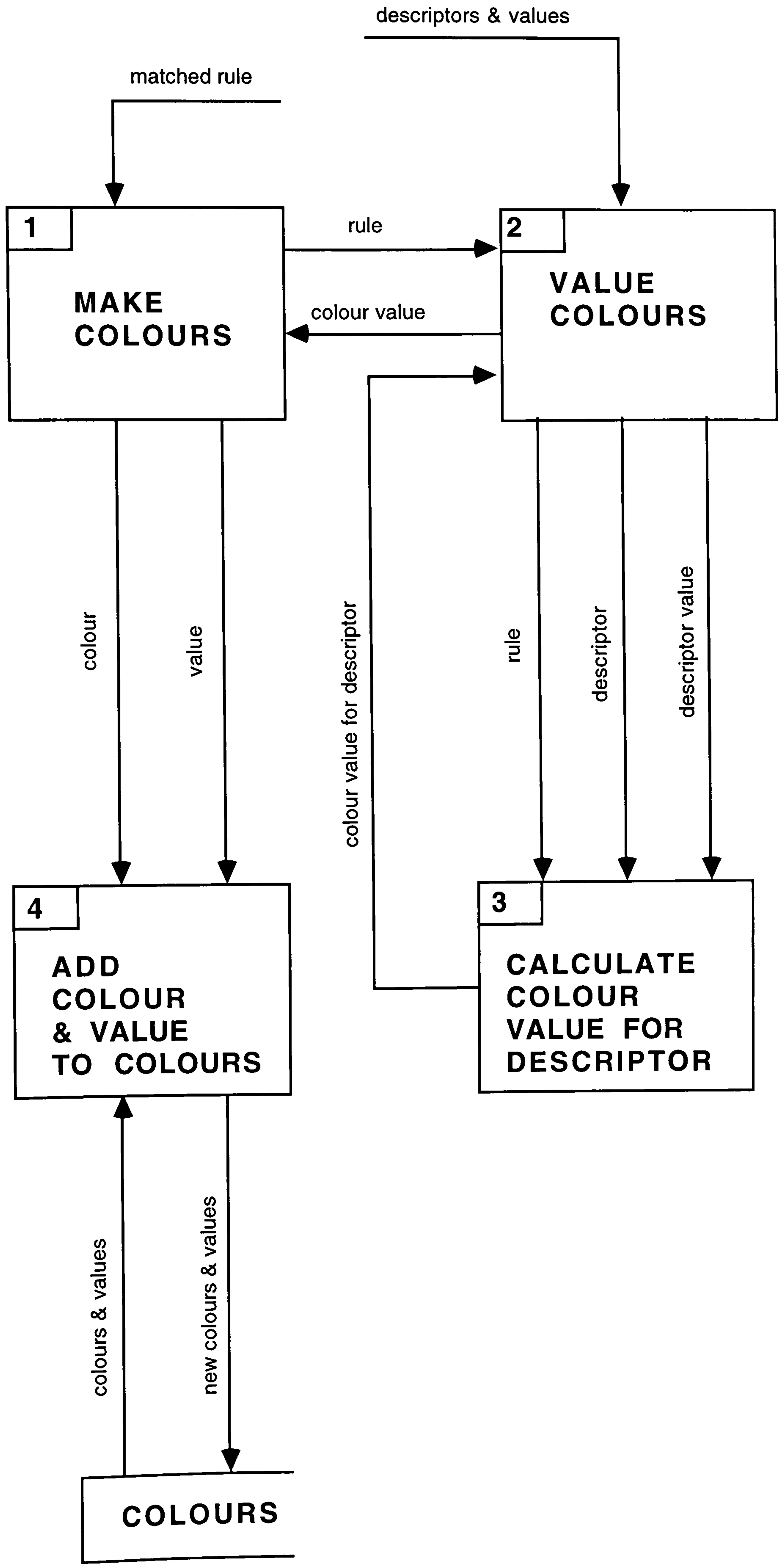
DFD 2.6 Find & Update Used Rule Descriptors



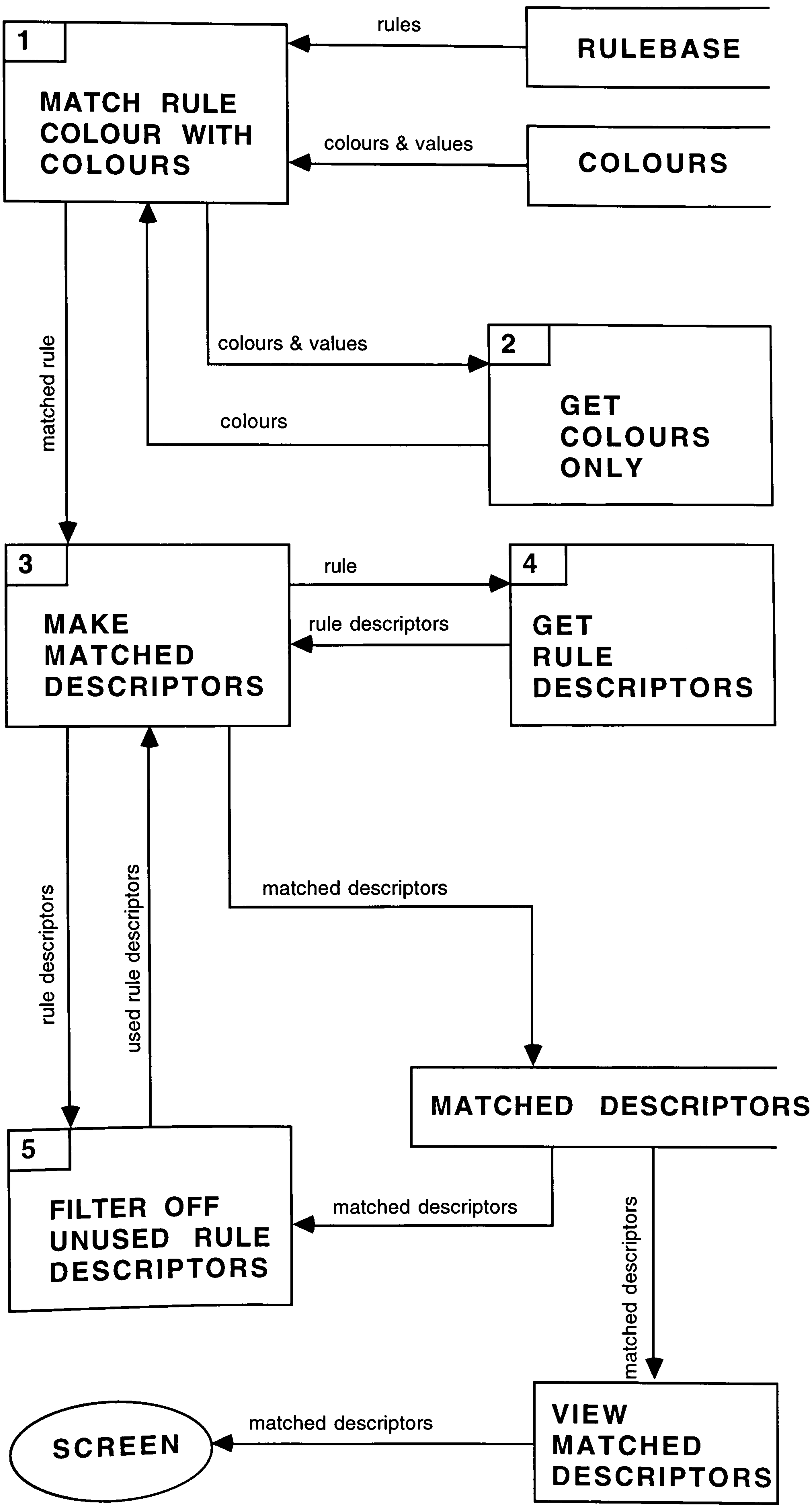
DFD 3 Generate



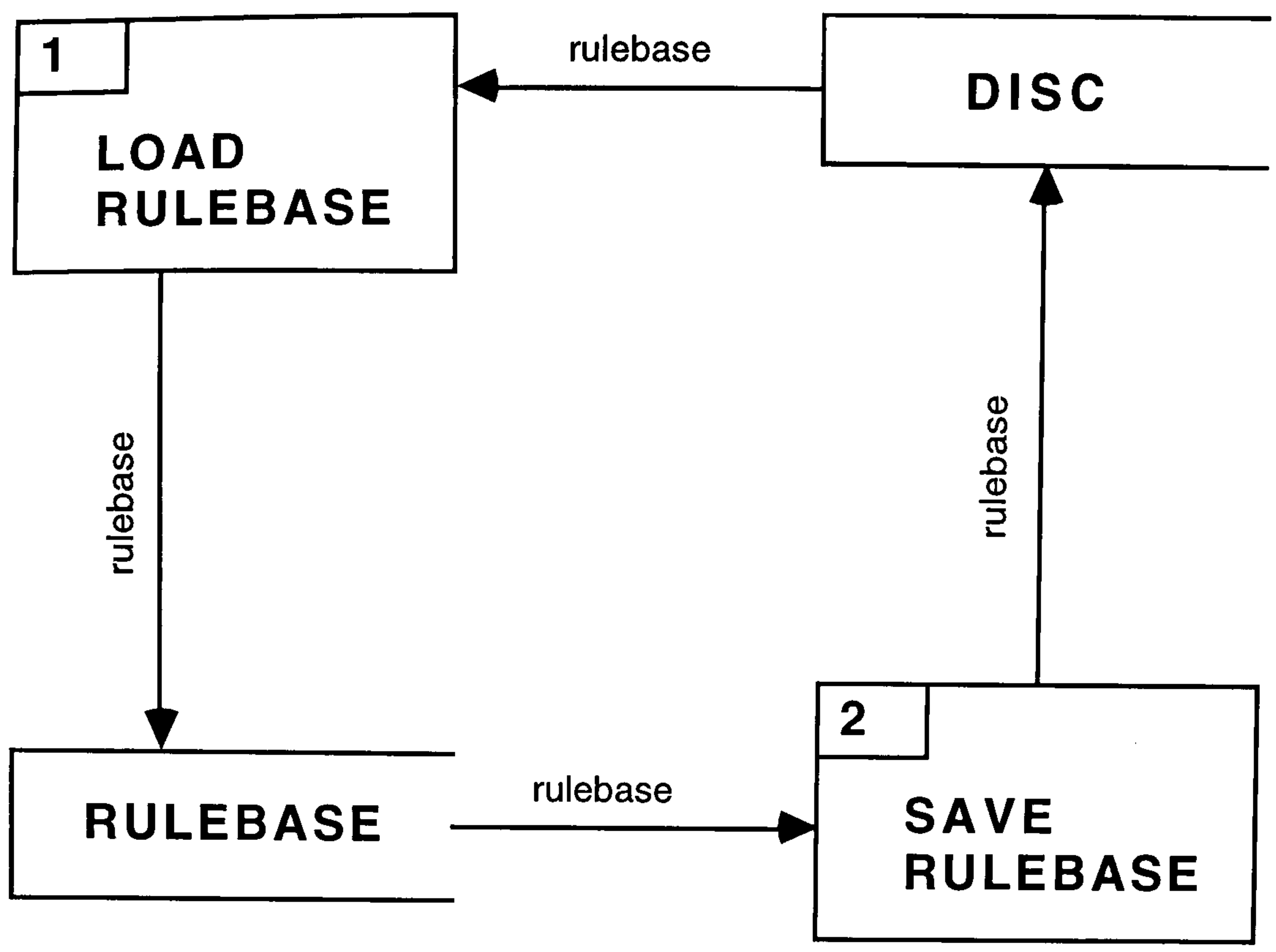
DFD 3.4 Value Colour & Make Colours



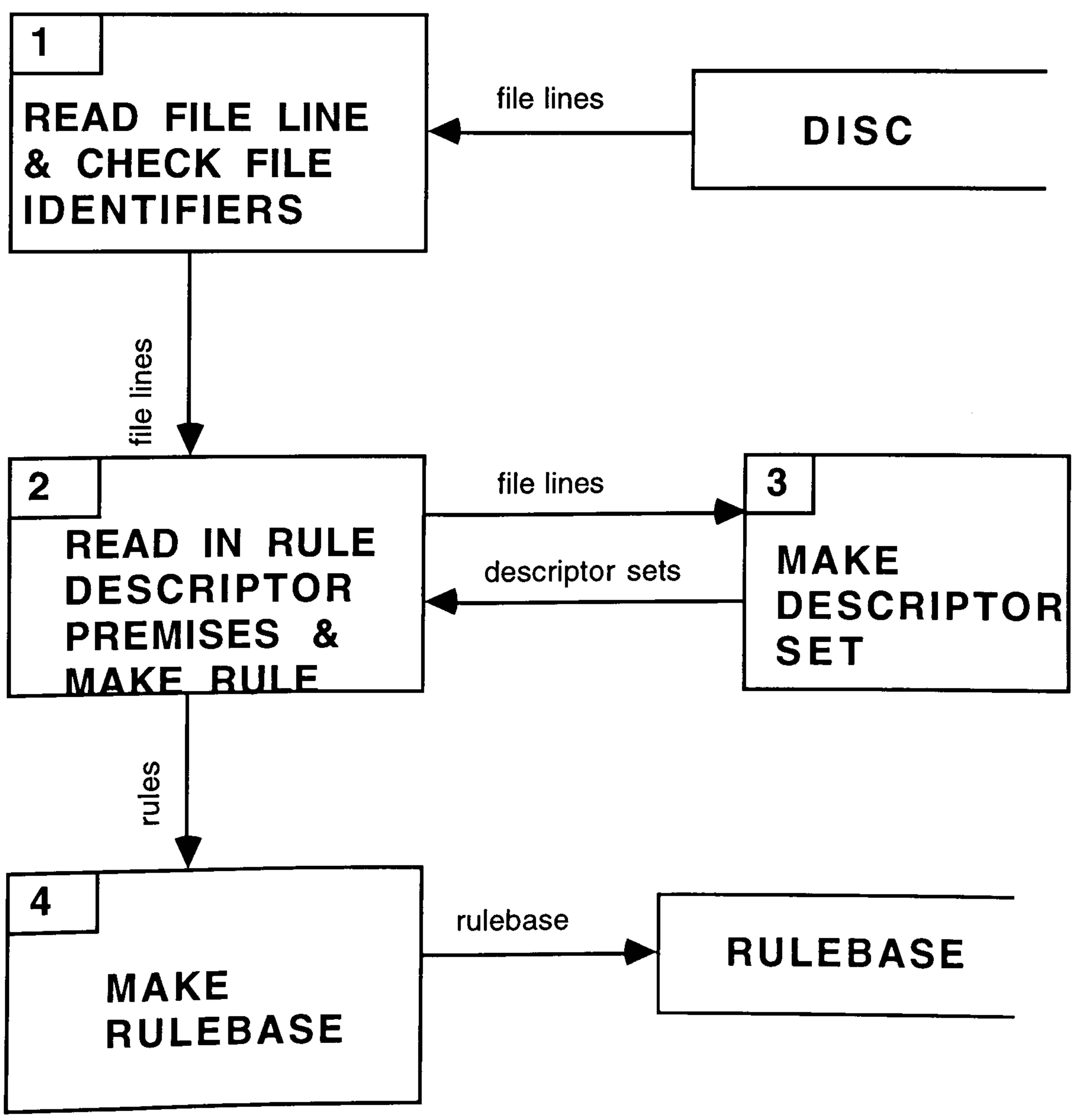
DFD 4 Describe



DFD 5 File Handling



DFD 5.1 Load Rulebase



DFD 5.2 Save Rulebase