UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

EXPERT SYSTEM PROTOTYPE FOR HYDRAULIC SYSTEM DESIGN FOCUSING ON CONCURRENT ENGINEERING ASPECTS

TESE SUBMETIDA À UNIVERSIDADE FEDERAL DE SANTA CATARINA
PARA A OBTENÇÃO DO GRAU DE
DOUTOR EM ENGENHARIA MECÂNICA

JONNY CARLOS DA SILVA

FLORIANÓPOLIS, MARÇO 1998

EXPERT SYSTEM PROTOTYPE FOR HYDRAULIC SYSTEM DESIGN FOCUSING ON CONCURRENT ENGINEERING ASPECTS

JONNY CARLOS DA SILVA

ESTA TESE FOI JULGADA PARA OBTENÇÃO DO TÍTULO DE

DOUTOR EM ENGENHARIA

ESPECIALIDADE ENGENHARIA MECÂNICA E APROVADA EM SUA FORMA FINAL PELO CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA.

Prof. Abelardo Alves de Queiroz, PhD Coordenador do Curso	Prof. Nelson Back, PhD Orientador

BANCA EXAMINADORA

Prof. Nelson Back, PhD Orientador	Prof. David Dawson, MSc, Ceng, Co-Orientador
Prof. Victor Juliano De Negri, Dr. Eng Co-Orientador	Prof. João Lirani, PhD Relator
Prof. João Carlos E. Ferreira, PhD	Prof. Franco Giuseppe Dedini, PhD

“Knowledge is as wings to man's life, and a ladder for his ascent.
Its acquisition is incumbent upon everyone.”
Bahá’u’lláh (1817-1892)

I dedicate this work to my Family for their priceless
help throughout this endeavour.

Acknowledgements

Many people helped me in this project, though here only few are mentioned. I wish to thank the viva members João Carlos Espíndola Ferreira, João Lirani and Franco Giuseppe Dedini for their dedication and comments on this work which greatly enhanced its quality, especially thanks to Nelson Back and Victor De Negri for their invaluable support, mainly during the project last stages.

To David Dawson, who accepted the project proposal from an unknown source, had patience with my "Brazilian English", helped in the work development as supervisor and domain expert in England, provided facilities in Lancaster and revised the manuscript, my deepest thankfulness. To Lancaster University Engineering Department for providing the material resources to the project execution and dissemination.

To Lancaster University staff members Jan Anderson, Ray Cheung, David Bennett and Marcus Duffy, who, several times, made me feel at home with their friendship and important help during my term in England.

To all experts who freely provided their time and efforts for this project, specially to Bud Trinkel (fluid power consultant; President of Hydra-Pneu Consulting, Inc., Newburgh, IN, United States of America), who, even without knowing me, played a vital role on the prototype expansion and validation. Also to Derek Seward, Senior Lecturer at Lancaster University, for his support on evaluating the prototype.

To Mechanical Engineering Department of UFSC for allowing me the 18 months term in England, which was a fundamental part of project accomplishment. To CAPES, Brazilian Research Agency who sponsored me during this work.

To all who directly or indirectly supported this initiative and to those who challenged the project in several stages of its progress, for they were a great impelling force that made me accomplishing more.

To Nader Manoutchehri who aided me as personal trustee in Brazil during my stage in Britain, my genuine gratefulness.

To the Bahá'í friends in Britain and in Brazil for their Spirit of Unity in Diversity which made us understand a bit more about World Citizenship wherever I went, my most great thankfulness.

INDEX

Chapter 1- Issues on Sequential and Concurrent Engineering

1. Introduction... 1

1.1- Sequential Engineering…………………………………………………......1

1.2- Concurrent Engineering………………………………………………...... 3

1.3- The Importance of Engineering Design for the economical success…............. 5

1.4- Life Cycle Concept…………………………………………………….... 5

1.5- Concurrent Engineering Benefits……………………………………........ 7

1.6- Main Pitfall of Concurrent Engineering Implementation……………........... 8

1.7- Work Structure…………………………………………………………...... 9

Chapter 2- Design Methodologies

2.1- Multidisciplinary Characteristics……………………………………........ 12

2.2- Computational System Aspects for Concurrent Engineering…………............ 13

2.3- Contacts abroad and Schemebuilder Project Introduction…………............. 16

    2.3.1- Functional Modelling……………………………………………....... 18
    2.3.2- Conceptual Design…………………………………………………....... 19

2.4- Computational Agents……………………………………………………....... 20

2.5- Advantages of Simulation Techniques……………………………........ 21

Chapter 3- Expert Systems & Object Oriented Approach

3.1- Expert System Main Definitions………………………………………….....24

3.2- Knowledge Engineering in This Context……………………………....... 25

3.3- Knowledge Elicitation Techniques in Hydraulics…………………….......... 27

3.4- Expert System Applicability…………………………………………....... 30

3.5- Expert System Development Process…………………………….......... 32

3.6- Expert System Choice……………………………………………………..... 33

3.7- Knowledge Representation Techniques………………………………….......... 34

3.7.1- Object-Oriented Properties……………………………………....... 38

3.7.2- Knowledge Organisation using Object-Oriented Modelling…............. 40

3.8- Knowledge Systematisation……………………………………………........ 46

Chapter 4- Hydraulic System Design Issues

4.1- Justification for Hydraulics……………………………………………….... 48

4.2- General Issues on Hydraulic System Design……………………………........... 50

4.3- Knowledge Engineer and Domain Expert Interaction………………............ 53

4.4- Computational System for Hydraulics……………………………………........ 56

4.5- Consideration about Electrohydraulic Systems…………………………........... 60

4.6- Hydraulic System Modelling Techniques…………………………….......... 62

4.7- Hydraulic System Design Packages…………………………………............ 65

4.7.1- HydroAnalysist.. 65

4.7.2- AutomationStudio... 67

4.7.3- CircuitWorks... 69

4.8- Simulation packages……………………………………………………........ 70

Chapter 5- Prototype Description

5.1- Initial Prototype Definition…...……………………………………………........ 72

5.2- Prototype Evolution……………………………………………………….... 72

5.3- System General Structure……………………………………………........ 77

5.3.1- Knowledge Base- The Classes………………………………….......... 78

5.3.2- The Rules…………………………………………………………..... 83

5.3.3- The Messages…………………………………………………...... 88

5.4- Generation of HTML Files as Output…………………………………......... 90

5.5- Generation of Functional-Dynamic Models……………………………........ 93

5.6- Concurrent Engineering directly addressed in the Initial Prototype…............... 99

5.6.1- Power Supply Circuit Aspects……………………………......... 100

5.6.2- Weighting Tool…………………………………………........ 103

5.6.3- Size Circuits and Components……………………………......... 106

5.7- Knowledge Base and User Interface Separation…………..…………........... 110

5.8- Fluid Selection Module…………………………………………....... 113

Chapter 6- Prototype Validation

6.1- Definition of Validation…………………………………………...... 116

6.2- Validation Metrics……………………………………………...... 116

6.3- Validation Methods……………………………………………..... 117

6.4- Validation Tests……………………………………………...... 119

6.4.1- Prototype Expandability Description…………………..……......... 120

6.4.2- Usability Tests……………………………………………..... 127

6.4.3- Experts' Formal Evaluation..……………………………....... 130

6.5- Final Considerations on Validation………………………………......... 137

Chapter 7- Conclusion & Future Issues

7.1- Main Contributions…………………………………………..... 138

7.2- Future Issues……………………………………………....... 142

7.2.1- Direct Increment of the current knowledge base.............................. 142

7.2.2- Greater Changes in the Knowledge base... 147

7.3- Final Remarks……………………………………………...... 155

Appendix 1- Prototype Graphical Presentation.………………………....... 157

Appendix 2- Prototype Output and Additional Experts' Comments….............. 168

Abstract

The work presents the main phases in the research and prototyping of an expert system (i.e. conceptualisation, implementation and validation). The project is based on a prototype development to demonstrate the feasibility of integrating the following areas: Expert System Application; Object-Oriented Techniques; and an industrially significant domain, Hydraulics. The system integrates a computational tool to support the design of hydraulic systems focusing on concurrent engineering aspects.

The project explicitly defines the areas in which the prototype application can improve the design process for hydraulic systems. It describes the functional framework and documents in a chronological form the development process of an expert system prototype. Within a defined scope, the project considers, as much as possible, robustness, expandability and modularity throughout its elaboration. The prototype attracted an intensive participation of users in its implementation which was also a key factor in the whole development. The development also greatly benefited from the use of the Internet for knowledge harvesting and validation of the system.

Finally, the work discusses future issues related to the prototype expansion and a potential development of a commercial package.

List of Figures

Figure 1.1. Time-to-market comparison between sequential and concurrent engineer........... 3

Figure 1.2. Cost influence at the design phases.. 4

Figure 2.1 Conventional Product Design Cycle vs. Product Design Cycle with CE............. 14

Figure 2.2 Work Scheme Diagram.. 17

Figure 2.3. Example of a Function-Means Tree for speed control function.......................... 19

Figure 2.4. Knowledge Representation and Modelling... 21

Figure 3.1- Expert System Development Process.. 32

Figure 3.2. Rules chaining examples.. 36

Figure 3.3. Subject level diagram.. 41

Figure 3.4. Class-&-Object diagram for some modelled entities.. 42

Figure 3.5. Structure diagram relationship between system and circuit................................ 42

Figure 3.6- Generalisation-Specification relationship among circuit classes.......................... 43

Figure 3.7- Attribute and Service representation.. 44

Figure 3.8- Message connection diagram... 45

Figure 4.1- Typical modules for a hydraulic system... 50

Figure 4.2. Valve and Pump Control Strategies.. 52

Figure 4.3. Flow chart used in choosing a particular type of pump...................................... 60

Figure 4.4. General Servo System Structure... 61

Figure 4.5. Trapezoidal Speed Profile 62

....

Figure 5.1- Description of the Prototype Modules... 78

Figure 5.2- Central control rule and blocks of rules.. 84

Figure 5.3. Central Control Rule Structure... 84

Figure 5.4. Create power supply circuit rule... 86

Figure 5.5. Create a bleed-off circuit rule structure... 86

Figure 5.6- Simplified structure of rule Combine Options.. 88

Figure 5.7- Connection diagram to create components... 89

Figure 5.8- HTML Generator Agent structure.. 92

Figure 5.9. Example of DYMOLA cuts for a cylinder.. 95

Figure 5.10. Functional Structure of the Computational Agent... 96

Figure 5.11. An example of the agent output in DYMOLA format..................................... 97

Figure 5.12- User, Prototype System and Internet Browser Interaction............................... 100
Figure 5.13 Power Supply Circuit (default option) 101
Figure 5.14. Power supply circuit options, presented for comparison.................................. 102
Figure 5.15- Ranking process.. 104
Figure 5.16- Relationship between load and circuits in the sizing process............................ 107
Figure 5.17- File Structure with system core as intersection among all versions.................. 112
Figure 5.18- Fluid selection module- Interaction with the Main Knowledge Base................ 114
Figure 6.1- Rule added as result from Pittsburgh feedback... 126
Figure 7.1. Paradigm shift as a result of the incremental approach....................................... 140
Figure 7.2- Rule diagram for selecting a gear pump type.. 143
Figure 7.3- Rule diagram for selecting a vane pump type... 144
Figure 7.4- Rule diagram for selecting a piston pump type... 145
Figure 7.5. Description of Troubleshooting Module of Hydraulic System........................... 150
Figure A1.1- Prototype Introduction window... 157
Figure A1.2- User information window.. 158
Figure A1.3- Definition of a directory for each session.. 159
Figure A1.4- Qualitative attributes input... 160
Figure A1.5- System browser window.. 162
Figure A1.6- Power supply selection window... 163
Figure A1.7- Weighting facility.. 163
Figure A1.8- Supply pressure definition.. 164
Figure A1.9- Definition of parameters for a linear load... 165
Figure A1.10- Group loads window.. 166
Figure A2.1- Hydraulic system diagram for the first problem... 169
Figure A2.2- Graphical representation of a actuation circuit... 170
Figure A2.3- Alternative for power supply unit... 171
Figure A2.4- Power supply unit. 172
Figure A2.5- One alternative system diagram for a moulding machine................................ 175

List of Tables.

Table 1.1- The Consumer-to-Consumer Process.. 6

Table 1.2- Concurrent Engineering Return.. 7

Table 3.1- Expert System Development Phase Definition... 32

Table 4.1- Comparison between valve and pump control.. 53

Table 4.2- Some computational systems for Fluid Power simulation............................... 71

Table 5.1- Default values of *has components* slot for circuit sub-classes.......................... 81

Table 5.2- Summary of WEB growth. 90

Table 5.3- Example of a component list generated by the prototype............................... 101

Table 5.4- Results from an expert ranking comparing circuits... 104

Table 6.1- Comparison between two versions of a hydraulic system............................... 123

Table 6.2- Students' validation criteria set. 128

Table 7.1. Relationship among sources, effects and causes... 148

Table 7.2. Description of some rules of knowledge base. 151

Table A2.1 Load qualitative inputs.. 170

Table A2.2- Example of component list.. 171

Table A2.3- component list of power supply unit. 173

Chapter One
Issues on Concurrent and Sequential Engineering

1- Introduction

Due to the needs originated from intense market transformations, i.e. increasing competitiveness, technological advances as well as economic globalization, more companies have been applying Concurrent Engineering techniques. In this context, the time to market is as important as the product quality, some researches show that 80% of a market for a new product is shared with the first two companies which launch the product (BRAZIER & LEONARD,90).

In order to cope with this environment, multidisciplinary teams grouping, for example, engineers, designers and market analysts are being integrated to search for ways of creating quickly and more efficiently new products through collaborative Concurrent Engineering efforts (ASHLEY,92).

This work describes the implementation of a prototype computational system to support the design of hydraulic systems focusing on concurrent engineering aspects, particularly related to this product type. Moreover, the project applies, as much as possible, guidelines regarding concurrent engineering implementation in the development of the computational system itself. Therefore, in this manner, the prototype is also considered a potential product.

Concurrent Engineering is defined as a systematic approach to integrate the product design process with its correlated processes, including manufacturing and logistic support. This approach aims to consider as many as possible the attributes involved during the life-cycle of the product from conceptual design stage to recycling or discharge of the product, including quality, cost, time schedule and users requirements (CORBRIDGE,92). Through a concurrent engineering approach, a company intends to integrate from the first development phases, knowledge, resources and experience gathered from marketing, design, manufacturing and sales areas, for a successful product design, with high quality, low cost and customers' satisfaction. The most important outcome of applying this approach is the time-to-market reduction when comparing to a more traditional sequential engineering approach (RADHARAMANAN,93).

1.1- Sequential Engineering

A drawback of sequential engineering is that usually the product conceptualisation is carried out without an estimation of manufacturing feasibility, and unless a feasibility study is done it becomes difficult to conceive the inherent problems related to a partial product definition, for instance, in a mechanical design, tolerances can be specified that are not according to the available machining precision.

In several cases, the functional requirements definition lacks further specification, and even if a complete specification is done, to direct the effort to satisfy only the functional requirements, assuming that the other requirements (manufacturing, operational, maintenance, transport etc.) are less important or can be achieved later, might prove extremely wrong and uneconomical. Ironically, some design engineers are reluctant to modify the design at the conceptual stage. However, several months or even years later, when the product design is almost finished, designers do accept design modifications, which is exactly the opposite of what the company had planned (HARTLEY,91).

In some cases, experts from other areas, specially manufacturing, add information into the conceptual stages, only suggesting few smaller changes as a form to adapt the existing design to manufacture. This situation is an outcome of the segmented way companies operate as well as the great responsibilities placed on the departmental level. Usually the attempts made by vanguard thinkers in order to break down the barriers among the departments face resistance from others. When a concurrent engineering team is firstly established, some design engineers can consider it as a threat. An adequate design management is necessary to allow those engineers to concentrate on their particular skills at the same time of receiving inputs from other team members (HARTLEY,91). Some of the weakest aspects of sequential engineering, when analysing the design process from conceptual to prototype stages, are:

- Insufficient product definition;
- No manufacture or assembly study is made;
- Insufficient cost estimation;
- Few guidelines about the detailed design during the conceptual stage;
- Great potential for late and expensive changes in the design.

In hydraulic system design, some typical problems originated from the sequential engineering approach can include:

- Lack of maintenance guidelines during the design phase;
- Poor involvement of component suppliers in the design process;
- Insufficient search for alternative solutions for power supply and actuation circuits;

- Incomplete load requirement description which might lead to an inadequate component and control selection.

These and other issues will be considered throughout the project via guidelines and alternative solution generation provided by the prototype system.

1.2- Concurrent Engineering

As a result of the team work concept, in which every member is important, and through the application of certain techniques, concurrent engineering aims to overcome the aforementioned barriers. The contests that occur among different experts in a company can jeopardise the whole design. The loyalty must be towards the company and its product in the first place. The functions and skills of each expert must be explored, which points a need for the team to analyse the design from different perspectives, before committing cost or making irreversible or difficult decisions. Obviously, the design engineers are responsible for the functional design, but in a concurrent engineering environment each member can (or should) make suggestions from his/her own viewpoint (HARTLEY,91).

The planning definition considering the Product Life Cycle is a need in a concurrent engineering context. Then, one of the requirements is to plan the activities in such manner that the involved people can be managed based on their specific results as well as on the team global achievement. Without a formal design procedure there might exist several uncertainties in the group performance assessment. This formal procedure aims to facilitate the concurrent engineering approach in achieving its main goal, i.e. reduce the time-to-market as shown in figure 1.1.

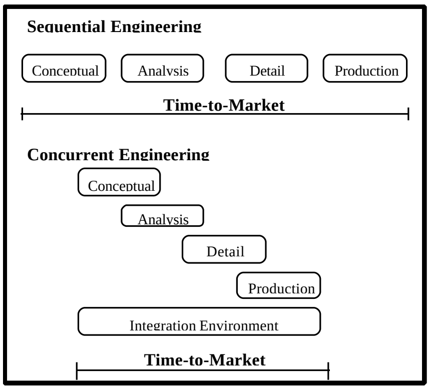


Figure 1.1. Time-to-market comparison between sequential and concurrent engineering, adapted from (CARLSON, KEMSER & ALLEN,97).

As far as cost reduction is concerned, some researches show that the first two phases of the design process, i.e. the conceptual stage (requirements definition, feasibility studies and alternative generation) and preliminary stage (product and process modelling), spend about 20% of the allocated budget. However, these phases define approximately 80% of the total product cost, as shown in figure 1.2.

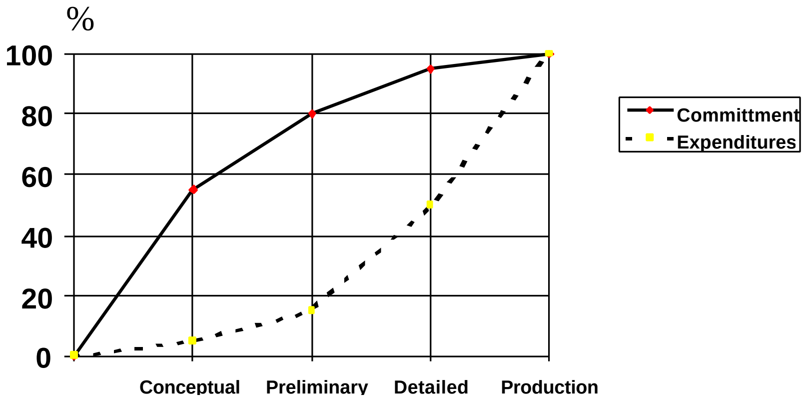


Figure 1.2. Cost influence at the design phases (FABRYCKY & BLANCHARD,91).

From figure 1.2 it can be deduced that the reduction cost programmes that are concentrated on the detailed design or production phases have small impact in terms of life cycle costs. Thus, initial consideration from manufacturing planning, production, assembly and even customers' support can greatly improve the maintainability and manufacture of the product as well as help to attenuate the need for tardy changes in the design, with a consequent cost reduction.

In order to apply a concurrent engineering approach it is necessary to develop a model of the design process. Thus to manage such an integrated environment it is fundamental to consider the product functional modelling as well as the process modelling, with its sub-tasks that are related to concurrent engineering perspective.

The necessity for a design oriented to manufacture was a direct consequence of the fast changes of the users' needs as well as improvements on the state-of-art manufacturing and design methods available to satisfy those needs. Considering that approximately 70% of the production decisions are directly determined by the product design, there is less freedom of choice in terms of decisions after the conceptual and preliminary design phases.

1.3 The Importance of Engineering Design for the economical success.

The increasing occurrence of interdisciplinary product development has not only removed many of traditional constraints to design but has now given the designer or design team a much wider freedom of choice as to the best solution to a particular design problem. It must be noted that interdisciplinary design does not necessarily imply the presence of a group, although in practice this is often the case and particularly when the product is large and complex in nature. It is possible that as designers and engineers gain experience and become exposed to areas of expertise beyond their normal regime- hence complementing their existing knowledge- they are able to bring more to bear upon their design problems, and possibly bringing about a more innovative design solution (OH & SHARPE,95).

The entire product development process can be seen as a series of decisions on interrelated issues which modify information describing the product being developed. Further, a decision structure based on noting the issue to be resolved, criteria associated with the issue, alternatives developed, comparison between alternatives and criteria, and the rationale for the decision is a good model for tracking and evaluating the product development process (ULLMAN,97). Therefore, there is a great need to provide an adequate support for decision making throughout the design process, mainly during the first two phases, when the most important decisions, in terms of life cycle, are made. In these phases, most of the information is qualitative and subjective (OH & SHARPE,95).

The great number of qualitative and quantitative options commonly found in virtually every design makes this area appropriate to investigate the application of decision making supporting systems. Moreover, a specific application of computational systems which has already started, and which will continue to grow, is in decision support systems, for decision making constitutes a significant part of work that is left to humans, partly due to the increasing automation (HOLLNAGEL,90).

1.4- Life Cycle Concept

The life cycle concept is universal in its applicability. Between its extremes, i.e. the users' needs perception and the product discharge (or recycling), there are two main phases. The first phase relates to the product development itself, including design, manufacturing, test, etc. The second phase corresponds to the product application, it involves operation, maintenance, discharge or recycling of the product (FABRYCKY & BLANCHARD,90). The different phases involved in the life cycle are depicted in table 1.1.

Table 1.1. The Consumer-to-Consumer Process (FABRYCKY & BLANCHARD,90).

CONSUMER	**Identification of Need**	Wants or desires for systems because of obvious deficiencies or problems are made evident through basic research results.
PRODUCER	**System Planning Function**	Marketing analysis; feasibility study; advanced system planning (system selection, specifications and plans, acquisition plan, research, design and production, evaluation plan, system use and logistic support plan); planning review; proposal.
	System Research Function	Basic research, applied research (need oriented); research methods; results of research; evolution from system research to system design and development.
	System Design Function	Design requirements; conceptual design; preliminary system design; detail design; design support; engineering model/ prototype development; engineering test; transition from design to production.
	Production and/or Construction Function	Production and/or Construction Function requirements; industrial engineering and operational analysis (plant engineering, manufacturing engineering, production control); quality control; production operations.
CONSUMER	**System Evaluation Function**	Evaluation requirements, categories of test and evaluation, test preparation phase; formal operational test and evaluation; data collection, analysis, reporting and corrective action; retesting.
	System Use and Logistic Support Function	System distribution and operational use, elements of logistics and life cycle maintenance support; system evaluation, modification, product phase-out: material disposal, reclamation and/or recycling.

As table 1.1 presents, life cycle concept is a very comprehensive definition, it includes all phases of the product design. Although table 1.1 depicts the concept in a sequential form, in a concurrent engineering environment consideration of later stages (for instance, logistic support and system evaluation) must be given attention as early as possible in the product development.

The adequate definition of the product attributes must be the focus to improve design, manufacture and logistic support. The development process integration is the key issue for this goal, it should involve the following areas:

1- Conceptual process that defines the product attributes which will satisfy customers' values and the producer's objectives;

2- An advanced product design morphology which embraces more functions concurrently and considers more alternatives in an accelerated development cycle;

3- Infrastructure improvements that are based on target product attributes as a basis for justifying the investment in new technologies.

The importance of life cycle is due to the fact that a design oriented only to accomplish the primary functions, usually would create side effects as operational problems. This is due to exclusive consideration of the primary functions rather that involving other aspects (manufacture, maintenance, assembly, etc.). Although there exists specific knowledge to address each one of these aspects, the main drawback to do so is related to the integration of these different types of expertise in a systematic form. The capacity to integrate them is, or at least should be, an intrinsic activity of engineering. The contest for market leadership will be won by those companies who can make effective use of both design methodology and technologies (FABRYCKY & BLANCHARD,90).

1.5- Concurrent Engineering Benefits

Some researches present the interaction between Concurrent Engineering and Quality Control on Manufacturing. As the Quality Control deals with the manufacture of physical products which can be measured more easily, it can be said that the quality control refers to “how to make things correctly”, on the other hand, the quality control on design, implemented through concurrent engineering, refers to “how to make the correct things”, i.e. to be sure that the products will satisfy the customers’ needs (CHARNEY,91).The return gained from concurrent engineering is variable, as depicted on table 1.2, however it is proved from experience, as well as reinforced through the present trends, that this approach has a positive influence on several aspects of product development.

Table 1.2. Concurrent Engineering Return (CHARNEY,91)

Concurrent Engineering Benefits	**Percentage**
Development Time	30-70 shorter
Engineering Changes	65-90 fewer
Time-to-market	20-90 shorter
Overall Quality	200-600 higher
Office Productivity	20-110 higher
Investment Return	20-120 higher

Even accepting that the information on this table is very generic, and perhaps imprecise, the point to be emphasised here is not the quantitative aspects of the return from concurrent engineering effort, but rather the potential impacts of this approach in several aspects of the design process. Moreover, an investigation with a German company proved that the concurrent engineering can be successfully applicable. This study showed that because of the close cooperation of quality assurance, manufacturing specialists as well as job scheduling nearly from the beginning of development, the design optimising interventions can be considered when the effort for changes is still small. The result is a clear reduction of time from receipt of order till dispatch of the individual product while the product quality rises (AHRENS & BEITZ,97).

1.6- Main Pitfalls for Concurrent Engineering Implementation

Despite the previous mentioned advantages or benefits and present increasing trends towards concurrent engineering effort, the implementation of concurrent engineering does involve difficulties which deserve to be, even briefly, addressed here. One of the main difficulties relates to the ineffectiveness of the design team, this is mostly due to the lack of experience in the field of group working. In order to minimise the risks, some considerations must be taken into account (WILSON,90):

- The team should include a wide variety of skills and control the most relevant aspects of the design, from the conceptual stage until the first six months of manufacturing;
- It is necessary an effective participation of all members during the design requirements definition, if possible even involving the customers;
- Establishing common goals and plans for all members since the first design phases;
- Provide an adequate location for members, at least the representatives from design, process planning and supplier;
- Management should determine that the functional structure (organisation) of the company does not interfere with the team objectives.

Another aspect to be considered as a likely difficulty is the definition of an impractical time scale. In order to diminish this effect, it should be taken into consideration:

- To know the complexity of product development and relative percentage between new and conventional technologies;

- To examine the performance, in terms of time scale accomplishment, of projects with different complexity levels;
- To keep a detailed documentation of each project, relating complexity and time scale.

There is no more harmful aspect to keep a development time schedule than to change product requirements during the process. On the other hand, in order to obtain a new competitive product, it is necessary to satisfy not only the present users' needs but also the future ones, including the analysis of potential demands. Thus, to avoid requirement changes during the design, the team should try to predict the future needs based on technological and marketing trends. This process increases product complexity and possibly costs during the first stages, but it should benefit later phases (WILSON,90).

The characteristics of the present productive processes are heavily based on suppliers. Usually, there might exist also difference among suppliers in terms of delivery times, quality standards and costs. Those aspects make the involvement of the supplier sector a critical factor in concurrent engineering implementation. In order to help to overcome this difficulty some points can be considered:

- The team should control the interaction with the suppliers. Thus, if possible, the suppliers should be selected at the beginning of the design process. They should work directly with the designers to verify that the components will accomplish with the specification, delivery times and costs;
- Reduce the numbers of suppliers, keep them as members of the design process.

1.7- Work Structure

As can be noticed from the previous sections, the understanding of concurrent engineering and its proper application involve a great deal of knowledge from different areas. However, such knowledge is spread among different people in an organisation. Therefore, a computational tool for supporting the designer in applying those particular types of knowledge can largely improve the design process. This assertion is one of the main motives for the work described in the next chapters.

Presently, there exist specific CAD packages tailored for hydraulic systems, some of them discussed in chapter 4. Furthermore, different researches have been made regarding the application of Concurrent Engineering through computational tools, some of them are introduced in chapter 2. However, complementarily to those packages and researches, the present project describes the development of a prototype expert system to integrate those two aspects, i.e. design of hydraulic system with a concurrent engineering focus. The system described in this work accomplishes the following functions:

- Prompts the user to respond interactively to determine the machine requirements in a friendly form, i.e. without requiring specific knowledge on hydraulics.
- Automatically generates a set of feasible circuits for consideration by the design engineer, based on well proven principles of circuit design.
- Allows the preliminary ranking among the alternative solutions, considering general attributes.
- Allows the change of Power Supply unit, redefining the component lists.
- Calculates the power supply demand based on the load attributes (force, speed, torque, etc.).
- Handles servo-hydraulic circuits as feasible alternatives.
- Generates topological dynamic models for simulation tailored for a specific simulation package.
- Displays the circuit schematics and descriptions through automatically generated HTML pages which can also be viewed via the Internet.

The developed system has two targeted types of user: For a noviciate engineer (or student) without expertise in hydraulics, the system provides a better understanding of the design process and some guidelines in hydraulics, and for experienced users, the system would empower them with more freedom of choice and a quicker start-up in terms of generating alternative solutions for the design problem, a facility to balance in a much broader way general criteria for the design at the conceptual design stage and also provide a tool to automatically develop the basic calculations. Next, a description of the work structure applied to develop such system is presented.

The next three chapters constitute the theoretical background upon which this work is based, while the others compose the project description. Chapter 2 concentrates on the design methodologies, tools and techniques applied and/or related to this project. In chapter 3, the main concepts and definitions on the expert system area are discussed, it includes the basis on which the choice for applying expert system approach, rather than other techniques, was made. This chapter also involves the conceptualisation and implementation of the knowledge acquisition in this project. Chapter 4 focuses on the domain knowledge, i.e. hydraulics, it discusses issues related to modelling, definitions and computational systems used on hydraulic system design. It also presents more specifically the reasons which support the choice of hydraulics for this development. In chapter 5, the structure of the prototype is given, it describes the system with its computational agents. Chapter 6 points out some validation aspects on expert system in general, as well as outlines the validation tests carried out with the prototype. Finally, chapter 7 summarises the main contributions and potential expansions of the prototype. The prototype graphical description is given in the appendix 1, and some of its results are shown in appendix 2 along with additional comments on the prototype application.

In order to assist the reader, throughout this text the computational system is referred to as the prototype and the author is identified as knowledge engineer. These concepts, directly related to Expert System, are properly defined in chapter 3.

Chapter Two
Design Methodologies

This chapter concentrates on the design methodologies, tools and techniques applied and/or related to this project. It points out the multidisciplinary characteristics of this project, discusses some necessary aspects for a computational system whose main objective is to support a Concurrent Engineering (CE) implementation, presents the concepts used as background for the development, and describes some advantages of applying simulation during the design process.

2.1- Multidisciplinary Characteristics

As defined in the previous chapter, this work describes the development of a computational system to support the design of hydraulic systems focusing on CE aspects. Therefore, from the analysis of this scope, it is clear to recognise the multidisciplinary profile of such task, whose main bulk is to describe the knowledge engineering process involved in developing the prototype. This process is outlined in detail in chapters 3 and 5.

Other important issues in this project include: design methodology concepts; advantages of hydraulics as a well defined area for expert system development; and the CE aspects that are considered throughout the project.

The multidisciplinary profile of this task also agrees with the trends in the engineering and business activities. These trends demand from the modern engineers the ability to: make connections among specialised areas of knowledge; understand the relationships among seemingly disparate discoveries, events and trends; and integrate them in ways that benefit the world community (FRANCIS,93).

Based on the project amplitude as well as on the time scale limit in terms of implementation, the knowledge engineering task (defined in chapter 3) is the paramount point during the development process. Therefore, in order to have a rapid system prototyping, one responsible for such project is required to have a basic comprehension about the knowledge domain, i.e. hydraulics, without necessarily being an expert. This avoids the necessity for an early indoctrination in the domain to be conversant in it, which is sometimes a drawback for knowledge acquisition (GONZALEZ & DANKEL,93).

Although the project involves the application of several knowledge sources, as described in the next chapters, in some stages of the project, key decisions were taken by the knowledge engineer on his own. This points out the importance of the design methodology background during the process, for as in every product development, the earliest decisions cause the greatest impacts on the development process. Here, those decisions were: choice for expert system approach as an implementation paradigm; domain knowledge selection; knowledge representation technique; and choice for incremental approach as an expert system development process model. The reasons which support these decisions are explained in detail in the next chapters.

2.2- Computational System Aspects for Concurrent Engineering

Although Concurrent Engineering has changed manufacturing philosophy and practice in several organisations, until recently there has been a dearth of support tools that effectively provide interface and information exchange among the members of interdisciplinary teams. This dearth is evident for the comprehensiveness required to develop those tools as shown later. Presently, there is a crucial need for adequate communication tools which, for example, can provide the designer with manufacturing constraints at the early stage of the product design.

An automotive manufacturer after a recent study attributed up to 30% of the total cost of the project and over 60% of elapsed time to additional design modifications that were necessary to make the product manufacturable. In another internal study by an automotive manufacturer, each part was subjected to an average of 20 engineering changes after release to production, at an average of $1500 for each change, not including tooling. Retooling costs varied from $20K to over $2.0 million for each change. The same study indicated that problems that required customer recall or dealer repair averaged an order of magnitude higher cost than the cost for a factory change. The solution to the problem of greatly increased production cost due to late design changes can be addressed by applying CE concepts (ORADY & SHAREEF,93).

Due to the inherent complexity of CE efforts, performance of the CE team could be substantially improved by using computer-assisted tools that are fully integrated and can provide proper communication between the team members. In order to determine the needed tools, the role of the CE team needs to be fully understood. The computer-assisted environment should be designed to act as a consultant for a designer and other members of a CE team. The environment should allow input from individual members of the team to the designer and other members (ORADY & SHAREEF,93).

As mentioned before, the ultimate goal of a CE team is to design a high quality high efficiency product that can be developed and manufactured at lowest cost, highest profit and in minimum amount of time. This can be achieved only if the designed part (system) can be manufactured without major modifications in all stages of the product design cycle. Therefore, the CE team must be structured, as shown in figure 2.1, to have representative participation from engineering and non-engineering departments.

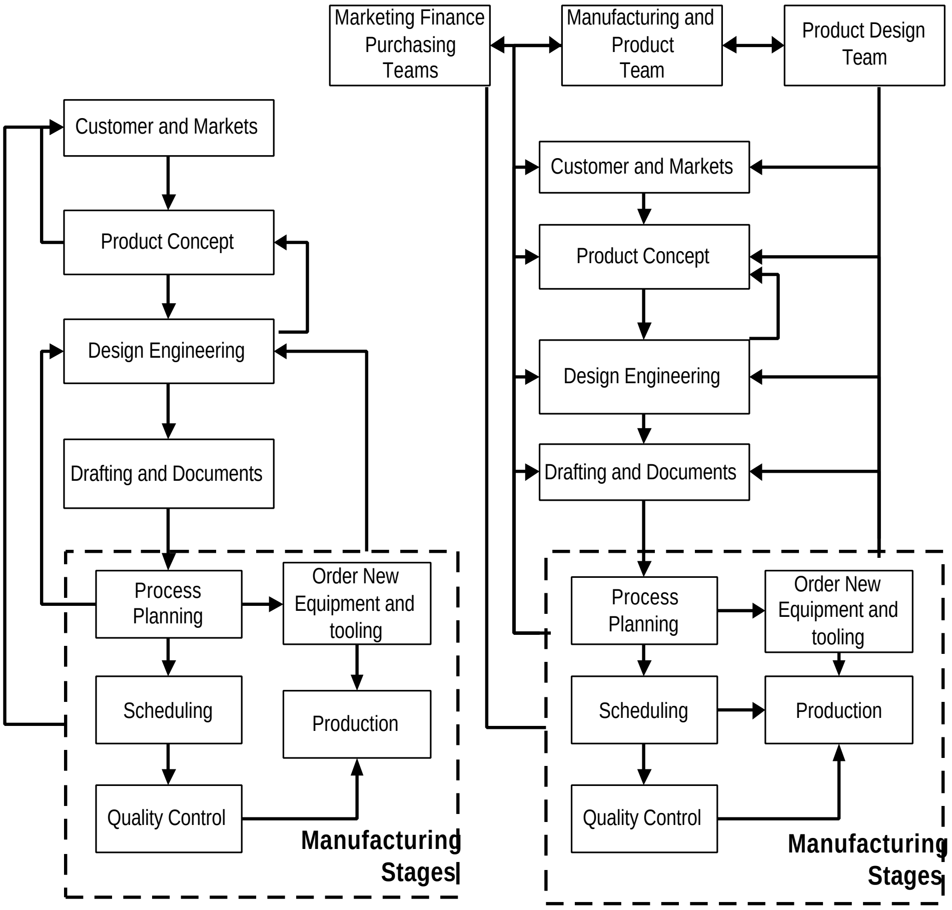


Fig. 2.1 Conventional Product Design Cycle Vs. Product Design Cycle with CE (ORADY & SHAREEF,93)

As depicted on figure 2.1, the flow of information during the CE product design cycle is parallel. All of the participants members in the product design cycle team share information and provide input right at the beginning and in the early stages of product design. This not only decreases costly design changes, but also reduces lead time and production cost. Application of CE practice during the product design cycle also leads to changes in the organisational structure of the companies as well.

Some desirable characteristics for a computational environment to support a CE team include (ORADY & SHAREEF,93):

1- Providing the designer with a manufacturing knowledge base that assists him/her throughout the design stages;

2- Facilitating communication between the design engineer and other members of the CE team in all design stages;

3- Comparing between possible production methods and selecting the most economical manufacturing procedure;

4- Determining the manufacturability of the designed part in relation to the selected manufacturing process or processes;

5- Being capable of determining the inspection procedure to achieve the desired quality;

6- Providing basic knowledge related to design for assembly and after sales serviceability, during all the above stages;

7- Linking the computer assisted tools such as CAD/CAM/CAPP/MRP etc. and all data bases ;

8- Establishing a permanent knowledge base for inputs and feedback between members of the CE team; and

9- Providing information about alternative suppliers.

From this list it is clear to understand the complexity and comprehensiveness of an environment to support CE team efforts. Add to these factors the impact on the company organisational structure and it becomes evident that to develop such a huge environment is a multi-task effort which would require years, or even decades, rather than months to be completed. Therefore, it would be very naïve to claim that the prototype, with an implementation time scale of months and limited human resources, could accomplish the whole functionality described in the above list. Another aspect which enhances this difficulty relates to the expert system development that in its own brings some specific challenges, as described in the next chapters. Thus, the very objective of this project is to demonstrate that the computational structure developed here, involving design concepts, tools, development approach and prototype system, is sufficiently robust to:

-Model some of the main entities involved in the design of hydraulic system;

-Supply a basic means of communication between the designer and the other participating members;

-Provide guidelines to the designer about issues such as: maintenance, operation, cost effectiveness and safety related to hydraulics, during the conceptual design stage;

- Offer an expandable environment to include other CE aspects.

Those must be the criteria on which the verification and evaluation of this project should be based.

Next, the main design concepts used in this development will be presented together with project applied as a basic structure to elaborate the prototype.

2.3- Contacts abroad and Schemebuilder Project Introduction

From the above mentioned aspects, some desirable characteristics for a computational system included in this project are: possibility of generating alternative functional models and solutions for a design problem and integration with other computational systems. From the very beginning of the project, it was noted that the author would require to interact with different sources for this development.

As in every product development, the users or customers play a very important role, therefore in the early stages of this development contacts were made with the main industries who supply hydraulic solutions in Brazil, to try to establish some linking scheme between the industries and the developer. Unfortunately, no reply was obtained from those contacts; besides, there was a necessity of having a platform on which the system could be developed. These aspects were the reasons that caused the author to contact institutions abroad.

In searching for potential partner institutions to develop this project, important references were found in the proceedings of Computer Aided Conceptual Design (SHARPE & OH,94), 1994 Lancaster International Workshop on Engineering Design. Through these and other references, it was possible to know about existing research projects in the United Kingdom which had similar approach to the one this project was originally based on.

Two British universities were contacted and accepted to collaborate with this project through an exchange programme scheme. These were Lancaster and Bath universities. The first was carrying out a project called Schemebuilder, whose objective was to develop computational systems to support the conceptual design of mechatronic systems in general. The second university was directly involved on developing computer tools, mainly simulation, to assist the design of fluid power systems.

Despite the importance of Bath university in the world scenario as far as research on fluid power system is concerned, the choice was made for the Engineering Design Centre- (EDC) Lancaster University, mainly because of the research profile of this institution which was more concentrated on design methodology in general, rather than simulation for fluid power system. It is important to mention that, before this proposal had been accepted, the Schemebuilder project had not focused on hydraulics, but the methodology was being applied to bio-engineering and mechatronic systems in a very generic approach (OH et al,94). Next, the design concepts related to the Schemebuilder project and the project proposal description are presented. Figure 2.2 shows a diagram of phases involved in the prototype development.

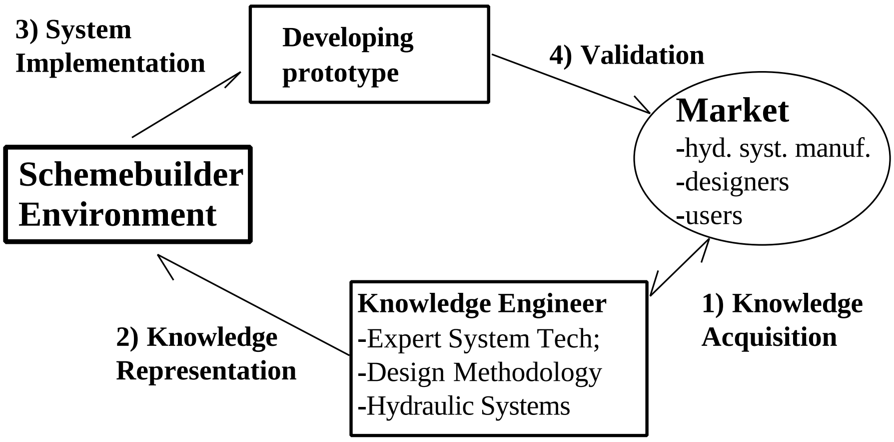


Figure 2.2 Work Scheme Diagram.

As shown on figure 2.2, the work scheme accepted by EDC is composed of four major phases: Knowledge Acquisition; Knowledge Representation; System Implementation and Validation. These parts embrace basically all activities in the development of an expert system, and they will be given more explanations in chapter 3. However, here the design concepts related to the Schemebuilder project are pointed out.

The first important definition is related to a scheme. According to (FRENCH,85) "*a scheme is an outline of a solution to a design problem, carried to a point where the means of performing each major function has been fixed, as have the spatial and structural relationships of the principal components. A scheme should be sufficiently worked out in detail for it to be possible to supply approximate costs, weights, and overall dimensions, and the feasibility should have been assured as far as circumstances allow. A scheme should be relatively explicit about special features or components but need not go into much detail.*"

Other concepts of great importance to this development are related to Working Principle and Means. A working principle consists of one or more required sub-functions, which may have certain attributes defined to accomplish one specific task. A means consists of at least one generic component and if necessary, one or more required subfunctions, which may have certain required attributes (LANGDON et all. ,95).

Furthermore, general design concepts include functional requirements and design parameters, that are defined as: Functional requirements are explicitly-stated design targets, such as force, weight, power, speed, etc. They are the most important features that must be present in a designed system. The design parameters are the independent variables or the "givens" that we must work with, such as material density, motor speed, heat dissipation properties, etc. (FRANCIS,93). The manner these concepts are represented in the computational environment is explained through the description of the prototype general aspects in chapter 5.

2.3.1- Functional Modelling

The functional modelling in the Schemebuilder environment applies the methodology known as Function-Means Tree which depends on the Law of Vertical Causality. This law states that the decomposition of a particular function into subfunctions is only possible, when a means has been chosen to realise the function (BUUR,90). There is causality in the sense that once a function is formulated, then it is possible to designate a number of alternative means, which may all carry out the desired function. Every means will however need the realisation of a set of subfunctions in a lower level. This methodology is best described by the function-means tree as depicted on figure 2.3.

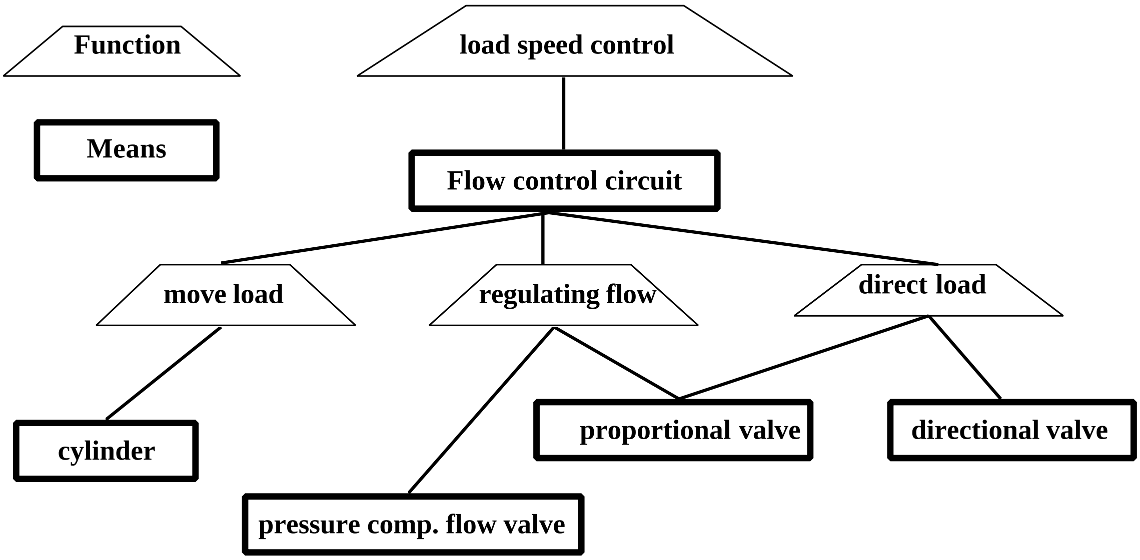


Figure 2.3. Example of a Function-Means Tree for speed control function.

The example presented on figure 2.3 involves only the domain of fluid power, but a similar structure could be applied for other energetic domains such electro-mechanical actuation. It can be noticed that there is a causality between function and means for the main function *speed control* is only decomposed after the means to accomplish it (*flow control circuit*) is defined. Each subfunction can be further decomposed once its corresponding means is defined, for example a proportional valve means needs measure, control and actuate functions. This example also demonstrates one of the key properties of the fluid power area that facilitates the design process, i.e. the close relationship between the functional modelling and the physical system.

A functional description of the design process applied to hydraulic systems is given in (LÜCKE et alii,95). The description clearly indicates the conceptual, preliminary and detailed stages (PAHL & BEITZ,88) involved in designing a electro-hydraulic system. This well defined description of hydraulic systems, in both product and process modelling, was one of the reasons to support the choice of this area for the project, it will be explored in further details in the next chapters.

2.3.2- Conceptual Design

The conceptual design in the Schemebuilder environment is a process of structured logical thinking in which engineering techniques are the basis for operation. This phase of design is generation intensive with evaluation being limited to gross heuristic concepts. It involves taking the problem statement or expression of need and generating broad solutions to it in the form of design concepts (OH et al.,94). The focus during conceptual design is that of synthesis where separate design elements in the form of propositions, concepts or facts are built up into a cohesive and inherent whole, resulting in a finite set of potential solutions from a collection of decisions describing the finished product function and form. Thus, conceptual design looks at generating the specifications, objectives and solutions which can potentially satisfy the desired purpose.

As can be noticed from the above definition, conceptual design is a very comprehensive and important stage of the whole design process. The form in which the design concepts are defined in the proposed system had to be adapted according to the requirements of the expert system approach as well as the knowledge domain area, i.e. hydraulics. The knowledge structure to represent design requirements, propositions, concepts and facts is directly related to the Expert System Approach and Object-Oriented technique application which are detailed in chapter 3.

2.4- Computational Agents

In every knowledge domain there are some concepts that represent concrete entities (such as, pumps, valves and actuators in the fluid power area) and others related to more abstract items (such as designer, customer, etc.) whose properties are not as clear and well defined as the previous ones.

Among the most important activities in developing an expert system is to study those entities in order to define which are the most relevant characteristics to be modelled. For example, the concept related to a hydraulic pump embraces several characteristics, such as maximum power, flow, pressure, efficiency, assembly instructions, application range, etc., in the context of an expert system whose objective is to model the functionality of a pump, without necessarily analysing its assembly, the assembly guidelines are not requested to be represented (in a first approach), nevertheless it is a relevant concept.

A powerful concept to help in this study refers to ontology, i.e. branch of philosophy that deals with the nature of existence (OXFORD,93). In Artificial Intelligence, ontology deals with how the concepts must be expressed and related among each other (RICH & KNIGHT,91). Through this concept it is possible to define different types and levels of knowledge modularization which helps to develop several integrated systems to handle different properties of the same entities, from a high abstract level of representation to a very concrete level, through the development of computational agents, whose context is presented in figure 2.4.

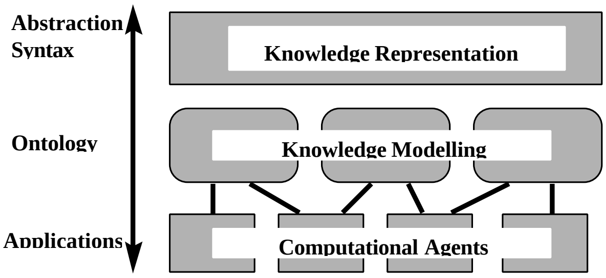


Fig.2.4. Knowledge Representation and Modelling (TOMYIAMA et al.,94).

In this architecture, there is a system for knowledge representation at the syntax level. However, the knowledge modelling level which describes ontology is more critical, for it represents what to be modelled and defines the basis of communication for intelligent agents.

The Intelligent Agents level represents application systems that integrate knowledge according to existing needs. For instance, intelligent agents can be modules for integrating a dynamic simulation package; a CAD system to perform geometric representation; a CAE system to model stress and strain analysis or another application module. The concept of intelligent agent modules is intensively used throughout this development. It will be further explored in the next chapters.

2.5- Advantages of Simulation Techniques

The complexity of fluid power systems has increased mainly due to the trend towards automation and high performance requirements. Likewise, the design of systems using more recent techniques has demanded greater effort during the design phase. In other hand, a shorter time-to-market has become a decisive factor. These conflicting aspects only can be solved through the application of more sophisticated techniques during the design (ELLMAN et al.,93).

In this context, simulation techniques have been increasingly applied to study the behaviour of fluid power systems during the preliminary design phase. Due to the nature of fluid power components, the models must take into consideration the non-linear characteristics such as backlash and friction forces, which are very commonly found during transient phases, e.g. end stops and valve switches. Moreover, sampling frequency and saturation of the amplifier are some of the nonlinearities to be found in the control system. It is worth to mention that simulation, as presented in this topic, refers to analysis of the components dynamic characteristics.

The application of simulation techniques can be defined in two phases. The first includes system model conceptualisation and the second one refers to simulation analysis. In the model conceptualisation, the designer must take into account data related to the main parameters and be capable of establishing a trade-off between model required precision and acceptable computational work. A good deal of experience is also necessary in this stage, for the designer must determine which inputs are needed to study the system behaviour and be able to interpret simulation results (ELLMAN et al.,93).With simulation technique applications, the following advantages can be obtained:

-Some physical prototyping phases can be avoided, which reduces the material, design and manufacturing costs;

-Optimisation techniques can be applied in the fluid power as well as control system design;

-Manufacturers become more familiar regarding the dynamic properties of their products and better capable of explaining them to users. Therefore, the existing problems can be more quickly understood and solved;

-The design team can test new concepts, it becomes easier to implement innovations;

-Designers have access to valuable information concerning the effect of actual properties on the system operation.

Instead of creating a simulation package tailored for hydraulics, which in its own is a very comprehensive task, the simulation aspect on this project has been considered through the development of computational agents to integrate the knowledge base with existing software systems. This aspect will be explained in chapter 5. The advantages of this approach are:

-It uses existing simulation packages, whose application has been verified by many users;

-In principle, it would be possible to develop integration modules for different simulation languages, which would be defined according to the user preferences;

-Comparative studies of simulation language performance and applications can be done.

As a final point in this chapter, it is important to mention that although Schemebuilder project researches have been carried out for several years in the academic environment, they pointed out the importance of building a computational system on its foundations and extending them to a level where a designer in industry feels comfortable and confident with the system application (OH et al. ,94). Thus, the present project has sought for involving as early as possible feedback from potential industrial users both on the project proposal conceptualisation and system evaluation, as described in the next chapters.

Chapter Three
Expert Systems and Object Oriented Techniques

In this chapter, the main concepts and definitions related to the expert system area are presented. It includes the basis on which the choice for applying expert system approach, rather than other Artificial Intelligence (AI) Methods, was made. This chapter also involves the conceptualisation of knowledge engineering in this project. As key aspects of the developed system, knowledge representation techniques, in general, and the application of Object Oriented Modelling in particular are also introduced in this chapter.

3.1- Expert System Main Definitions

What is an expert system? Prof. Edward Feigenbaum of Stanford University has defined an expert system as "… an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solution." (GIARRANTANO & RILEY,94). That is, an expert system is a computer system which emulates the decision-making ability of a human expert. The term emulates means that the expert system is intended to act in all respects like a human expert. An emulation is much stronger than a simulation which is only required to act like the real thing in some respects.

An expert system is composed of key parts which make its functionality. These parts are common to many knowledge-based systems (GIARRANTANO & RILEY,94):

- User interface- the mechanism by which the user and the expert system communicate;
- Explanation facility- explains the reasoning of the system to a user;
- Inference engine- makes inferences by deciding which rules are satisfied by facts or objects, prioritises the satisfied rules, and executes the rule with the highest priority;
- Knowledge base- a set of all rules, representing the knowledge in the system;
- Agenda- a prioritised list of rules created by the inference engine, whose patterns are satisfied by facts or objects in the working memory;
- Knowledge acquisition facility- an automatic way for the user to enter knowledge in the system rather than by having the knowledge engineer explicitly code the knowledge.

The knowledge acquisition facility is an optional feature on many systems. In some expert system tools like KEE and First Class, the tool can learn by rule induction through examples and automatically generate rules. However, the examples are generally from tabular or spreadsheet type data better suited to decision trees. General rules constructed by a knowledge engineer can be much more complex than the simple rules from rule induction (GIARRANTANO & RILEY,94). Depending on the implementation of the system, the user interface may be simple text-oriented display or sophisticated high-resolution, bit-mapped display.

Both issues above, i.e. knowledge acquisition facility and user interface were considered in the developing project. Concerning the knowledge acquisition facility, as implied, this type of tool is usually developed for including a spreadsheet type data, which can be a result of shallow knowledge description, as defined below. In the developing project, it was noticed that to try to allow a knowledge acquisition tool to embed high level cognitive knowledge, such as the design process know-how, would be beyond the project, considering its time-scale and the expert system tool state-of-the-art. However, a preliminary module for trouble-shooting analysis was implemented which allows the user to include new facts in the knowledge base, relating possible problem causes with their sources and effects. A description of this module is given in chapter 7.

As far as the user interface is concerned, the approach adopted in the project was to start the development using a standard text-oriented input, provided in CLIPS (C Language Integrated Production System, a shell tool developed by NASA) (GIARRANTANO & RILEY,94) and a graphical output which would represent the hydraulic system diagrams, thus mapping the way the designer in hydraulics usually approaches the area. In order to define a clear interface with CLIPS Input-Output (I/O) functions, a set of key functions (such as "*ask an open question*", "*ask closed question*" and their counterpart answers) was generated. The idea was that all calls for standard CLIPS I/O functions would be grouped in only few files. In this way, it would be possible to change these files in case of other interface platforms, this decision proved to be one of the key aspects of the developing system, for it allowed the generation of the system core (i.e. knowledge base and class definition) in a platform independent form as well as a straightforward integration with different interface systems, a more completed description of this aspect with its implementation is given in chapter 5.

3.2- Knowledge Engineering In This Context

The term knowledge engineering (WATERMAN,86) was coined to describe the activities of computer programmers eliciting rules from experts and coding them in some language into a knowledge base. The task of knowledge engineering has several milestones, such as: the definition of the expert(s), the knowledge representation techniques, the feasibility study, etc.. Among them, one that also deserves attention is the choice of the knowledge engineer, for this is the element who translates the expertise into a computer environment. This activity requires general technical abilities, i.e. background knowledge about expert system technology and mastering in computer techniques along with non-technical skills, for example, friendliness and interpersonal communication skills (GONZALEZ & DANKEL,93). Also the profile must embrace as much as possible an understanding of knowledge-based systems and of the domain application, for this balance greatly helps the process of knowledge elicitation.

Considering the large amount of references in AI that give several definitions of knowledge, it is important to conceptualise knowledge according to the present work. Therefore, some key aspects are presented, followed by their relationships in this context.

The types of expertise in knowledge-based systems can be classified into three distinct categories (GONZALEZ & DANKEL,93). Knowledge-based systems have had varying degrees of success when representing knowledge from each of these categories. These categories are (1) associational (black box), (2) motor skills, and (3) theoretical (deep) knowledge.

- Associational knowledge- This level of skill reflects heuristic ability or knowledge that is acquired mostly through observation. The expert may not understand what happens inside a black box, but he can associate the inputs with specific outputs. In this type of problem, an expert may have excellent associational understanding of some devices, and based on the experience, may be able to fix almost any problem encountered. This type of knowledge is also known as shallow knowledge. It is symptomatic in nature and does not attempt to understand the underlying principles of the problem, but only to solve it using its external features. Usually, it relates to diagnostic, classification and interpretative tasks.

- Motor skill knowledge is physical rather than cognitive-oriented, therefore knowledge-based systems cannot easily emulate this type of expertise, because humans learn these skills by repeatedly performing them.

- Theoretical knowledge, also known as deep knowledge, is acquired through formal training and hands-on problem solving. Typically, engineers and scientists who have many years of formal training possess this type of knowledge. Because of its theoretical and technical nature, this expertise is very easily forgotten unless continually used. At present, it is difficult to duplicate this kind of knowledge in conventional knowledge-based system. An attempt to model this type of knowledge applies model-based reasoning systems to encapsulate this deep knowledge and reason with it.

Although the model-based reasoning strategy appears to be powerful, the above reference presents example only using this technique for system diagnosis problems. In the context of hydraulic system design, the developing system aims to emulate the associational and theoretical knowledge. The first one applies to the troubleshooting type of problem which is an important feature for maintainability, while the theoretical knowledge relates to the main task, i.e. conceptualising a hydraulic system based on the user requirements in a high-level language. Here, high-level language means the definition of general attributes, such as position, reaction, type of control, etc., without requiring specific knowledge about hydraulics.

In the present context, the knowledge engineer played the role of an “assistant expert” in the first stages of the system structure definition (IGNIZIO,91). This aimed to develop a rapid prototype system to present the main underlined concepts as well as to define the next steps for the industrial and more experienced experts. This approach proved useful, as usually the domain experts lack general knowledge about AI techniques, also the presentation of a rapid prototype enhanced the interaction between the domain experts and the knowledge engineer.

The incremental approach considered in this work demonstrated its benefits, mainly during the stage of knowledge acquisition as will be shown later. Recent interest in design knowledge capture shows that the knowledge acquisition field is heading in the direction suggested by Aristotle's remark: *"the best manner to observe things successfully is to see them evolving from their origins"*. Knowledge acquisition is thus an incremental process of achieving, refinement and generalisation (BOY,91).

3.3- Knowledge Elicitation Techniques in Hydraulics

One of the potential outcomes of the knowledge engineering process for a company is to preserve its corporate memory. (KLEIN,92a) presents five aspects of knowledge engineering that should be taken into account for this purpose, they are given by the quotations in *italic* below. These aspects are used here as a template to explain some aspects of how the process of knowledge engineering was carried out in the developing system.

a)-Locating sources of expertise in an organisation.

Sometimes, the question "who is an expert?" is replaced with the question "Where is the expertise in a given organisation?".

In hydraulics, the expertise can be found in different sources, in fact this is one of the reasons which support hydraulics as an appropriate field to develop an expert system. In this work, knowledge sources were obtained from industrial associations, technical literature, consultants and academic institutions.

b)-Assaying the cost/benefit of engineering the expertise.

If knowledge is a resource, its valuation is driven by the cost of the elicitation and the benefits it confers. The cost of elicitation is determined by how scattered the expertise is and by the knowledge elicitation procedures needed.

In any work requiring the elicitation and codifying of knowledge, sponsors must be realistic about the magnitude of the task. Furthermore, an early payback in financial terms is unlikely and hence the exemplary research work will normally be carried out by a research agency or university, as in this case. Once this stage has been accomplished, the demonstration of benefits, for example in the shortening of the design cycle, will attract more focused commercial sponsorship. In fact, this aspect has been already verified, for through the prototype demonstration to industrial contacts their interest in a future (commercial) version was evident. This issue is more explored in chapters 6 and 7.

c)-Acquiring the knowledge.

This aspect includes several strategies: unstructured and structured interviews, analysis of familiar tasks, limited information tasks (using familiar tasks but omitting information that is typically available), constrained processing tasks (watching how an expert performs a familiar task under time pressure or other constraint), and the method of "tough cases" (analysis of how experts handle difficult tasks).

In the developing system, unstructured and structured interviews were applied, which are discussed later, with some aspects of analysis of familiar tasks. They are explained below in the section Knowledge Engineer and Domain Experts Interaction, which is part of chapter 4.

d)-Codifying the knowledge.

It appears that the large-scale intelligent system, such as XCON, that was once put forward as the sign of things to come is not very practical. We may see more applications that use limited intelligent system components, perhaps with a small rule base of 300 to 400 rules, to handle limited tasks embedded within larger system.

So far, the incremental approach adopted in the present work has been successful (SILVA & DAWSON,97a). It implies that the increasing functionality of the expert system depends on developing specific modules such as Troubleshooting Analysis and Dynamic Modeller that are considered as computational agents tailored for defined tasks (HUANG et al.,93). In this way, the system is kept modular and the modules are developed with fewer rules, which eases the validation process.

e)-Applying the engineered knowledge.

Three applications for engineering knowledge are: decision making, technology transfer and training. In decision making, the system user usually has to answer a set of relevant questions, which describe the features of the decision task and the system will generate a recommended action.

Here, the application of the engineered knowledge has been carried out through tests by some experts, as early as possible. The feedback from these tests was used to enhance the knowledge base and I/O interface. The main tests involve contacts through the Internet, which means that the level of detail contained in the interface and files descriptions were tested at the best level available, because all information relied upon email messages and associated transferred files, without face-to-face contact. The other applications, i.e. technology transfer and training, were considered in the later stages of the project, mainly in the process of validation.

Besides the above mentioned aspects, the knowledge engineering process to capture the general designer’s rationale may bring other benefits to a company (KLEIN,92b):

- Explicitly represented design rationale can help individual designers clarify their own thinking about the design;
- Ensuring that all relevant issues and requirements have been addressed;
- Detecting flaws in one's reasoning;
- Tracking the consequences of changes in requirements and design decisions and so on.

The benefits become even greater when one considers that the design is typically undertaken by multiple agents over time. The reasoning used to make a decision becomes available for others to critique and augment from their perspectives. The participants whose contributions are affected by a change in the requirements or design can be identified readily. Designs developed by others which addressed similar requirements and difficulties can be retrieved, understood and then modified to meet current needs. Documentation production is simplified. Design rationale can help new members in design teams quickly familiarise themselves with a project and can serve as a basis for training new designers. These issues have been addressed in the developing system, for its objectives is not only to model the design product (i.e. hydraulic system) but also the design process.

3.4- Expert System Applicability

Before presenting some aspects directly related to the development of expert systems, it is valid to define in which conditions this type of project is conceivable. In order to accomplish this evaluation, there are three main characteristics that must be firstly defined: possible, justifiable and appropriate (WATERMAN,86). Each of these attributes demands specific issues.

What is necessary to consider that the development is possible?

- There are reliable experts in the knowledge domain;
- Those experts agree with the choice and precision of solutions;
- The experts must be capable of explaining the methods applied to derive the solutions;
- The task must require cognitive skills;
- The task can not be very difficult, that means the expert should be able to teach the task to a beginner. If the expert is only able to solve the task through an intensive cognitive process, the knowledge required to this task is hard to be captured in an expert system;
- The task must be precisely understood, without requiring for example pure science research. The activity should not require an intensive manipulation of commonsense knowledge, this is a type of generic knowledge, that virtually everyone has and it is difficult to be modelled.

What are the conditions to be justifiable?

- The solution should have a high return;
- The expertise is in danger of being lost;
- The experts are becoming rare, as a result of changes in an organisation or due to other factors;

- The expertise may be requested in different places at the same time;
- The expertise may be necessary in hostile or difficult access environments;

What are the conditions for an appropriate development?

- The task must be done through symbolic manipulation, except pure mathematical problems (number crunch), which usually are not appropriate to expert systems;
- The task must require knowledge from expertise, if the problem can be solved only through an algorithmic manner, it is not appropriate to be described in an expert system approach;
- The task should be sufficiently difficult to require the investment, typically it must demand few years of experience in the specific knowledge domain for someone to be considered an expert;
- The task must deal with a problem sufficiently restricted to be manageable and sufficiently broad to have practical interest. The definitions of manageable and broad depend on the problem domain and are vital to the development progress.

The considerations involved in the three above conditions are not always explicit before the development. Thus in order to minimise the uncertainties regarding the process of interaction with experts, it was justifiable to apply the incremental approach. This approach determines that the expert system itself can help in its development. For this purpose, it is necessary that as soon as the knowledge engineer has “captured” part of the knowledge enough to implement a simple system, this should be done, and then apply the experience obtained from testing the simple system to guide its expansion (WATERMAN,86).

The incremental model has been used very successfully in large conventional software projects. The incremental model is also useful for expert systems development in which the addition of rules increases the capabilities of the system from assistant, to colleague, and finally expert level. The primary advantage of this model is that the increases in functional capability are easy to test and validate. Each functional increment can be tested immediately with the expert rather than trying to do the entire validation at the end (GIARRANTANO & RILEY,94).

As can be verified in chapters 5 and 6, the capacity to provide a rapid feedback to experts during the prototype development confirms the applicability of the incremental approach to model the knowledge base in this project.

3.5- Expert System Development Process

Although the development of an expert system requires a good deal of empirical tasks, mainly due to the interaction with experts, the selection of the domain area and validation activity, several models have been developed to map this specific type of project. Here, the development process model adopted in the prototype is presented.

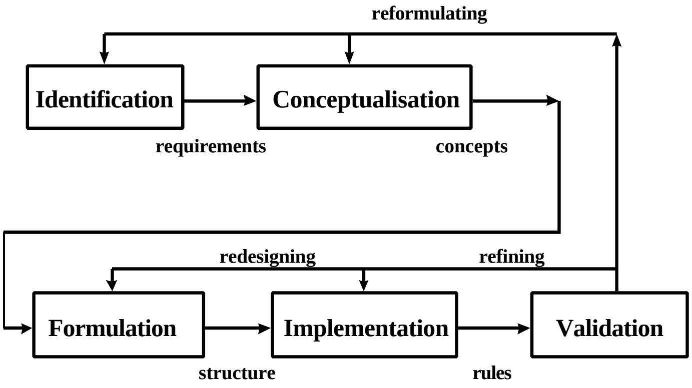


Figure 3.1- Expert System Development Process (WATERMAN,86).

Table 3.1- Expert System Development Phase Definition

Phase	**Definition**
Identification	Type and scope of the problem, choice of the experts, necessary resources and system objectives.
Conceptualisation	Knowledge engineer and experts decide which concepts, relations, strategies, sub-tasks and constraints are necessary to solve problems in the specific scope.
Formulation	Express key concepts and relations according to the structure of the implementation tool to be used. In this phase, the knowledge representation technique (rules, frames, semantic nets, etc.) to be applied become fundamental.
Implementation	System coding, it involves knowledge domain content, the specific implementation tool, the integration of different knowledge sources (to avoid contradiction) that can occur among the rules and data structure.
Validation	Test of the performance and usefulness of the system. The experts evaluate the prototype system and support the knowledge engineer on improving it. This phase can reveal errors in the knowledge representation, and consequently originate iterations for refining, redesigning, or reformulate the previous phases.

Despite the fact that the phases presented on figure 3.1 are depicted in a sequential pattern, with the application of the incremental approach, those phases are considered at the same time in different levels of complexity. This means that while some “parts of knowledge” are being formalised, others previously implemented are being validated. In this way, it is possible to co-ordinate the system development in an organic growth. However, in order to achieve this purpose it is important to establish key points, such as: which AI approach is applied; which knowledge representation technique allows a modular development; which modules are chosen to be implemented at first. These and other aspects will be explained in the next sections. The choice for this development model, together with the incremental approach, was justified for the need to clearly distinguish between the phases, as shown in prototype chronological evolution presented in chapter 5.

3.6- Expert System Choice

Considering a computational system as a product, based on the Concurrent Engineering aspects previously mentioned in chapter 2, the decisions made in the early stages of the system development are the most important ones. Among them, the choice of the implementation approach can be seen as of paramount importance. For depending on this choice, the whole project will more likely succeed. Therefore, here it is explained why the choice was for Expert System approach rather than other AI techniques.

Firstly, the general requirements for the system are defined and then how the requirements have been met through the chosen approach. One of the most important aspects to be accomplished is to achieve a rapid prototype for the system, this is due to the short development time available. Therefore, the approach to be chosen must offer a reliable way to solve a design engineering activity, rather than to create an excessively challenging problem from the computer science perspective, for the design activity on its own already presents a considerable challenge. In other words, the theoretical foundation of the approach should be well established to tackle similar problems.

The computational system must provide a clear explanation about the solutions it proposes. This feature is fundamental to the design task due to its comprehensiveness. Also because the type and number of entities involved in the design process increases the task complexity, the explanation facility becomes one of the key aspects to be taken into account.

The availability of a reliable implementation tool for the system is also an important point for this project. The tool must allow for ease of expansion of the system; modular development;

compatibility with different operating systems and a friendly learning process in terms of programming. For a non-reliable tool would jeopardise the whole development and potential use.

In the conceptual design stage, the designer deals, at first, more with qualitative attributes than quantitative parameters, therefore the chosen approach should provide straightforward symbolic manipulation.

Based on the above four points, (i.e. rapid prototype, explanation facility, implementation tool and symbolic manipulation), it was decided to implement this AI system through an expert system approach, using CLIPS as the implementation tool. Having defined this point, it was clear the great complexity of a design task could not be modelled using only the Rule-Based paradigm (If A Then B), therefore a decision was made to use the COOL module (CLIPS Object Oriented Language) which allows the application of fundamental properties, such as inheritance, abstraction and assembly relationships (SILVA & CHEUNG,97). Although the decision to use the expert system paradigm was made much earlier, i.e. during the specification phase (described in chapter 5), a survey done through a set of questionnaires posted to WEB newsgroups, in the implementation phase, reinforced that the choice was applicable for the defined knowledge domain.

3.7- Knowledge Representation Techniques

In Artificial Intelligence, the term knowledge refers to the information that a computational system must manipulate in order to behave in an intelligent form, or similar to a human expert in the same area.

One of the techniques used to represent the knowledge is known as Production Rules that are based on heuristics. The term heuristics originates from the Greek “heuriken”, which means discovery. It is also the origin of the word “eureka” mentioned by Archimedes while discovering a method to determine the gold purity (RICH & KNIGHT,91). A heuristics is a technique which enhances the search for a solution. Thus, by applying heuristics, it is possible to obtain adequate solutions for difficult problems in a considerable short period of time. Besides this, it is possible to formulate specific heuristics for particular types of problem domains, such as hydraulic system design. Some of these heuristics will be discussed in chapter 4.

As previously mentioned, the knowledge base is a fundamental part of an expert system, usually it contains different knowledge representation forms. Therefore, it is important to explain which representation techniques were applied to this project. In the present context, it will be discussed only Rule and Frame Techniques, despite other representation forms have been developed (GONZALEZ & DANKEL,93). It is important to emphasise that only the rule technique is not sufficiently powerful to formulate and solve complex engineering problems, mainly because only rules do not allow a great capacity of representing complex entities (DYM & LEVITT, 91), typically when these entities have several attributes and aggregated procedures.

Despite its limitation, the rule technique to model the domain knowledge is an important aspect, for the systems based on this representation technique have had a significant role in the evolution of Artificial Intelligence, from pure laboratory researches to commercial applications (RICH & KNIGHT,91).

The rules express the knowledge through a set of conditional statement of type If-Then, these statements are composed of two parts: antecedent (also known as conditional or pattern part) and conclusion, which define the procedures that will be taken depend on the condition satisfactions. In order to explain the rule concept, it is presented an example from hydraulic system design (SARGENT et al.,88).

- **If**: Speed control is required in the forward direction. (fact A)
- **And If**: The load is resistive in the forward direction. (fact B)
- **Then**: Flow control should appear before the actuator in the forward line.

An important factor regarding the application of rules is related to the inferencing methods, i.e. how the problem solving strategies are representing in the system. Two general methods of inferencing are commonly used: forward chaining and backward chaining. Forward chaining is reasoning from facts to the conclusions resulting from those facts. Backward chaining involves reasoning in reverse from a hypothesis, a potential conclusion to be proved, to the facts which support the hypothesis (GIARRANTANO & RILEY,94). These strategies are exemplified through figure 3.2.

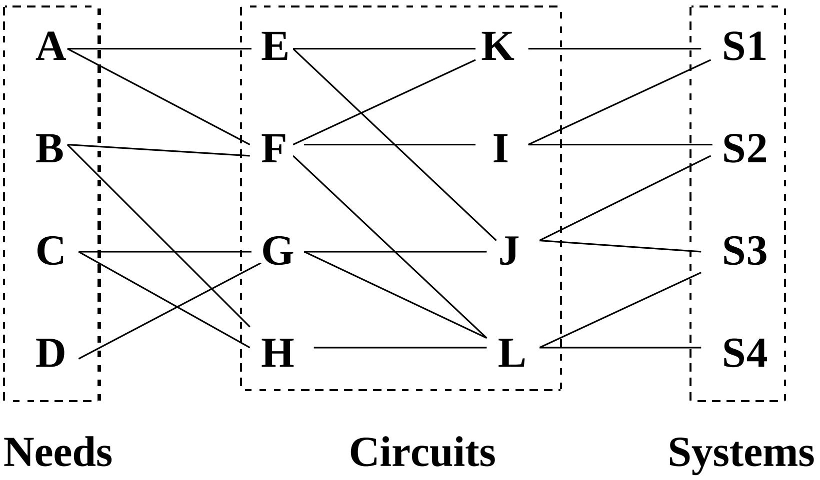


Figure 3.2. Rules chaining examples.

In the context of hydraulic system design, the facts can be represented as the user needs, as stated through the machine requirements. Those requirements in their own, based on the modular approach explained in chapter 4, demand specific circuits (or subsystems) which define functional blocks of a hydraulic system. Hence, based on figure 3.2 the forward chaining would be, for example, from the statement of needs **C** and **D** (facts), which hydraulic systems would satisfy those needs? Considering that all the links represent OR conditions, it is easy to verify that those needs can be satisfied by systems S2, S3 and S4. On the other hand, the backward chaining can be exemplified by the problem: Which needs could match the system S1? From the previous diagram, it would be clear that needs A and B would be the answer.

An important issue concerning the chaining process relates to what is the best approach to be implemented for a particular type of problem. There are some guidelines to support this decision (RICH & KNIGHT,91):

1. Are there more possible start states or goal states? It should be preferred to move from the smaller set of states to the larger (and thus easier to find) set of states.

2. In which direction is the branching factor (i.e. the average number of nodes that can be reached directly from a single node) greater? The option should be to proceed in the direction with the lower branching factor.

3. Will the expert system be asked to justify its reasoning process to a user? If so, it is important to proceed in the direction that corresponds more closely with the way the user will think.

4. Which kind of event is going to trigger a problem-solving episode? If it is the arrival of a new fact, forward reasoning makes sense. If it is a query to which a response is desired, backward chaining is more natural.

Although the two first guidelines can conclude a preference for backward chaining, in the present context, the option was to apply forward chaining, considering the following aspects:

- The explanation facility is fundamental for this project, for this facility will serve to explain the decision making process throughout the design process.
- Usually, the user of this type of system thinks in terms of needs as starting point to search for alternative solutions. It is typical for every design task.
- The implementation tool used in this project allows a more direct modelling of forward than backward chaining.

In order to permit the representation of more complex concepts, such as, entities with several characteristics, grouping of entities, generalisation and specification, pertinence relationships, etc., it is necessary, in expert systems, to use the *Frames* representation. A Frame is a set of attributes (usually denominated as slots) and their associated values (with possible constraints) to describe the entities in the real world (RICH & KNIGHT,91).

In the present project, the knowledge representation through frames is composed of a set of frames, interconnected among themselves via relationships described among their attributes. Each frame can represent a set of entities with common properties (in this case it is known as class) or a specific element of a class whose attributes have defined values (known as object or instance of such class) (RICH & KNIGHT,91).

Another definition considers frame as a representation technique which associates concepts to nodes, where the characteristics of each node are described by its attributes and values. Thus, a representation via frame is a net of nodes organised in a hierarchical form, whose higher level nodes represent more abstract concepts and the lower level ones model specific concepts. Therefore, the frame representation is considered as a semantic net (WATERMAN,86).

There is a great similarity between *Frame* and *Object-Oriented* representations, in fact the later is considered a specificity of the former one, for the Object-Oriented technique considers that all concepts communicate among themselves through message sending (WATERMAN,86). Due to this fact and also because this representation technique is applied in this project, the Object-Oriented modelling will be more explored below.

3.7.1- Object-Oriented Properties

Although several knowledge techniques (such as rules, frames, semantic-nets, and so forth) are presented in the literature, one decision in this project was the option for Object-Oriented Methodology, combined with Rule-Based Programming. In order to explain the rationale upon which this decision was made, some concepts related to Object-Oriented Methodology are presented below.

In order to develop an expert system for the design task it is crucial to understand how the designers think. In fact this understanding defines part of a specific area known as Cognitive Research. Although the complete understanding of such complex process is beyond the present work, there was an attempt to search for some key aspects to use as guidelines in the project and these aspects are discussed as follows.

Concerning the issue about how the designers think, some researchers suggest that experts, in an area very close to hydraulic system, tend to think in terms of Conceptual Chunking (COOKE,92). They replicated the traditional expert-novice recall of results with electronics technicians and the reconstruction of symbolic circuit drawings. Results indicated that experts attempted to recall drawings (i.e., a drawing with random placement of circuit symbols) in terms of units that were functionally related. In addition, the experts were faster than the novices on between-chunk transitions and often characterised the entire display in a matter of seconds. Although Conceptual Chunking, as presented in the reference, was not applied directly in the developing system, it supports the concept of a circuit as a working principle and of a hydraulic system as a scheme. These definitions are the key concepts of the Schemebuilder Project, as shown in chapter 2.

Other research postulates that a world of design information consists of many different sized groupings. This can be modelled by an object-oriented technique which possesses the representational adequacy to create a model containing separate hierarchical representations for distinct empirical concerns in a product engineering domain. Thus, object-oriented software systems are advantageous for modelling engineering design activities because of their support for complex relationships and evolutionary processes (ZUCKER & DEMAID,92).

Further research on design area, near to the present work field, which supports the application of Object-Oriented techniques presents the following remarks (YALIF,94): The data used during airplane design can be very complicated. Each part of the plane, such as the engine, has its own specifications. Each part also contains subparts, just as the engine has a turbine, compressor, and fuel pumping system. Each of these subparts has its own specifications in addition to sub-subparts. Therefore, design data needs to be arranged in an ordered manner that is readily accessible and understandable to the user. Object Oriented Design (OOD) is useful for storing the complex, voluminous and hierarchically arranged data produced during airplane design. The usefulness of OOD has been recognised elsewhere in the aerospace industry. In a study entitled “Managing Engineering Design Information” (FULTON & YEH,88), ten different data storage methods were examined. The conclusion: “The object-oriented data model was found to be a better data modelling method for modelling aerospace vehicle design process” than any of the others studied.

The application of OOD requires the understanding of key properties, which compose the theoretical foundation of this technique. These properties are abstraction, encapsulation, inheritance and polymorphism. They provide a powerful modelling strategy for complex systems. Next, these properties are explained (GONZALEZ & DANKEL,93).

1. *Abstraction: To ignore aspects of some entity that are not relevant to the current problem so that it is possible to concentrate more fully on those aspects that are.* In the present context, the complete machine functionality, which defines the design problem, is accomplished by a hydraulic system in a very abstract form. The abstraction takes place through a definition of set of circuits, in other words, the problem functionality represented by different loads (or points of actuation) is broken down into sub-problems and the functionality of each load is accomplished by a specific circuit.

2. *Encapsulation: Each part of a program should hide a single design decision with each program interface revealing as little as possible about its inner workings.* This aspect is achieved by the definition of methods, that is a set of procedures related to the behaviour and properties of a specific class represented by a message-handler. Several methods were defined for each class, for example, procedures to create files; to size circuits and components, to rank alternatives, and so forth. Therefore, the complexity of these procedures is encapsulated into each class, this makes the system more manageable and expandable.

3. *Inheritance: It allows to express the common characteristics possessed by a collection of different classes of objects once.* In the present system, some generic characteristics (such as, ID to identified each instance; name, cost, weight, etc.) are defined in a upper component class, while other more specific attributes (displacement for pumps and motors; area for cylinders; pressure drop for valves) are defined in the sub-classes.
4. *Polymorphism: It allows each class to respond to the same message in its own way.* An example of implementation of this principle in the developing system is: both System and Circuit classes have the same create-components message-handler, which means that those objects can receive the same message, however the procedures activated by this message depend on each class. The system class processes the message in a more abstract form, i.e. it forwards the same message to a set of circuits, which handle the message more specifically.

The application of the above properties makes Object-Oriented technique a powerful knowledge representation methodology, providing a great flexibility for the development of expert system in different areas. Further detail on how the properties were implemented into the developing system are given in the system description, chapter 5.

3.7.2- Knowledge Organisation using Object-Oriented Modelling

In the previous section the properties and usefulness of Object-Oriented Techniques were discussed. However, to structure the knowledge applying those techniques demands a methodological approach, for different levels of complexity must be manipulated and several relationships among the concepts should be modelled. Different methodologies were defined to propose forms of modelling concepts through Object-Oriented techniques. A comparative study of all available techniques is beyond the scope of the present project, a comprehensive analysis can be found in (BOUZEGHOUB et al., 97). However, a specific technique for this project was studied during the specification phase. The Yourdon/Coad methodology was chosen for its enough comprehensiveness, hierarchical modelling and because this technique was an expansion of the structured approach used at the EDC (BRACEWELL et al.,95).

The Yourdon/Coad method for Object-Oriented Analysis is directed for software engineering, it aims to support the development of a generic system architecture. Its main objectives are to improve the productivity, enhance the quality and maintainability of the system.

The method proposes modelling the system in five levels: Subject, Class-&-Object, Structure, Attribute and Service. These levels are implemented through the following diagrams: Object State, State Transition and Object Interactions. The specific definitions for the terms related to this method are according to the pattern defined by (YOURDON & COAD,92). In order to demonstrate the method and simultaneously introduce some concepts related to the present project, a sequence of diagrams follows. To illustrate the technique, the diagrams are related to the hydraulic system design.

The first level corresponds to the subject level, figure 3.3. Here one tries to define in a broad way the problem domain and the general system functionality. In this context, the system functionality relates to how the system elements must be organised in order to form a coherent environment.

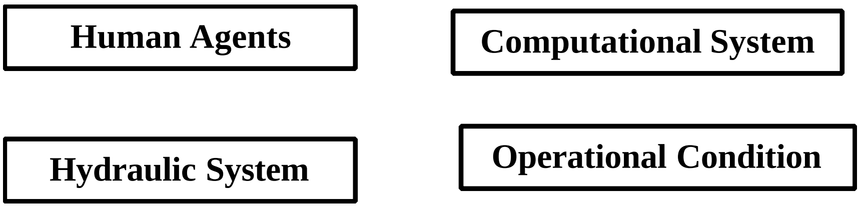


Figure 3.3. Subject level diagram.

Figure 3.3 represents the four main areas which were considered to plan the present project. The Human Agent area defines some entities that will interact with the system, typically the user (a hydraulic system designer) and possibly his customers and suppliers. The computational system block relates to the structure of the expert system in itself (knowledge base, user interface functions, etc.). The hydraulic system represents the design solution, that is the expert system outcome for a particular set of operational conditions.

In order to explain the Class-&-Object diagram, figure 3.4 depicts a simplified representation for some entities modelled by the expert system. To simplify the example, only few attributes are presented for each class. In this diagram, a clear distinction is made between Class and Object. It should be pointed out that this structure, with its classes is a key aspect to understand the underling knowledge representation implemented in the prototype.

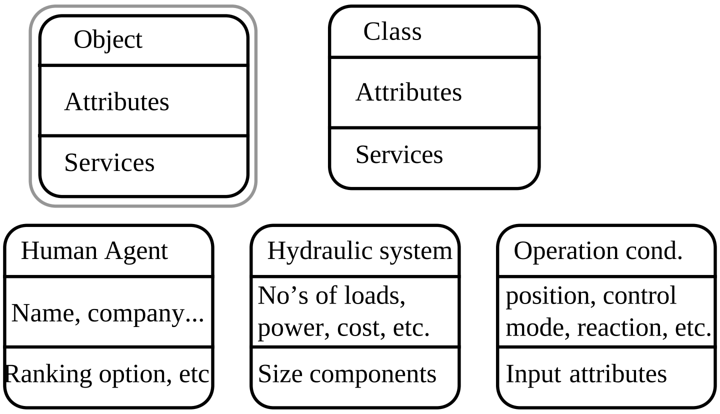


Figure 3.4. Class-&-Object diagram for some modelled entities.

In order to describe the other levels, the hydraulic system class was chosen. This was also due to its more practical and broad scope, most of it already implemented. Figure 3.5 shows a typical diagram relating some entities commonly found in a hydraulic system.

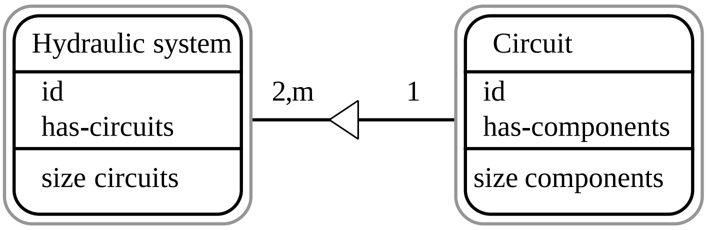


Figure 3.5. Structure diagram relationship between system and circuit.

Figure 3.5 depicts the relationship between a hydraulic system and a circuit, or more generically a scheme and a working principle respectively. Again for clarity, figure 3.5 only presents few attributes for each object. The triangle means that there is an assembly relationship between these entities, i.e. one hydraulic system is composed of (at least) two circuits, one power supply and one actuation circuit in case of only one load. In order to represent this type of relationship, the attribute ID which identifies specific instances (system1, system2, circuit1, circuit2, etc.) is used to create a network among the several generated objects. A similar relationship exists between circuit and component instances. It should be noted that this type of relationship occurs between instances and not between classes. Obviously, in the classes the attributes to allow this description are defined, but not their values which really perform the assembly relationship. Further details on this implementation will be explored in chapter 5.

Another type of relationship found in modelling complex entities is the Generalisation-Specification relationship, represented on the next figure. This association exists among different classes.

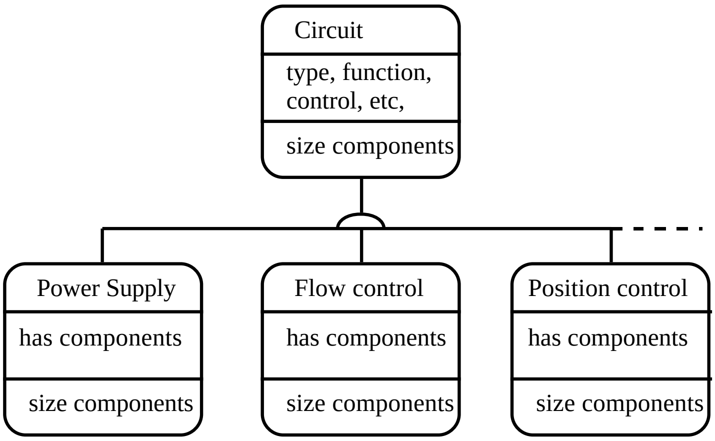


Figure 3.6- Generalisation-Specification relationship among circuit classes.

As figure 3.6 shows, there is an upper class circuit on which generic attributes such as type, control, function, etc. are defined and inherited by the sub-classes. Although the figure depicts only few classes in a lower level and with the same attributes, two points must be emphasised: firstly, there are several sub-classes with their specific attributes. Secondly, the attribute *has components* depicted above has different default values for each one of the classes, in other words, the functionality of each circuit, represented by its component set, is modelled according to each sub-class. This facility enables the generation of component sets for all circuits in a very easy form. A similar type of representation is used to define the component class structure, for in this domain there are also several generic and specific attributes.

At the attribute level, one has to define all the relevant characteristics of an object in the context of the system functionality. For example, considering a hydraulic pump, in a system aiming to model concurrent engineering aspects, properties such as cost, efficiency and availability should also be modelled. In other words, besides technical properties the knowledge representation should consider the operational and economical properties. At this level, all the possible relationships among the objects must be represented according to the system functionality. In a system whose main objective is conceptual design, the cost aspect can be modelled as an attribute. However, if one computational system has economical analysis as basic function, cost should be considered as a class whose several elements (maintenance cost, operational cost, transportation cost, etc.) are relevant to the analysis.

Furthermore, through Object-Oriented Modelling it is possible to explicitly define the system granularity, i.e. what is the most detailed level in terms of knowledge representation (WATERMAN,86). In the present project, the least functional unit to be represented on the expert system is the hydraulic component. However, it should be emphasised that in case of a system for modelling hydraulic component (ex.: pump, motor, cylinder or valve), its internal structure must be represented, in other words a component is itself a functional system, which embraces hydraulic, mechanical and some times electrical sub-systems.

At service level, figure 3.7, the specific object behaviour should be modelled, i.e. it specifies the operations that can be performed on each object and the messages exchanged between objects. Thus, it is relevant the definition of an Object State as the identification of the values of its attributes. Therefore, a change in a value(s) of an attribute(s) reflects an object change.

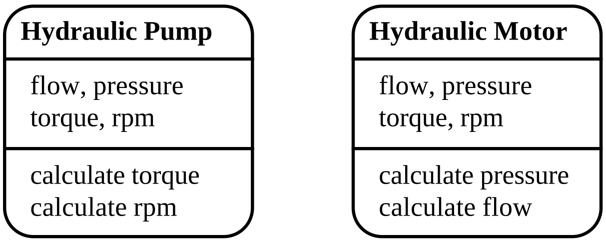


Figure 3.7- Attribute and Service representation.

Considering the classes represented on figure 3.7 as an example, it is easy to verify that although they share the same attributes (due to the similarity between a motor and a pump), their services are different, for to size a pump the basic input parameters are pressure and flow required, which are directly related to torque and rpm through relationships involving efficiency and displacement. This procedure is basically reversed to size a motor. Thus, a hydraulic motor class should model other services (methods or procedures). In order to achieve comprehensiveness, a computational system should model all relevant states for every object according to the system functionality.

According to the design theory for mechatronic systems (BUUR,90), the system state is a consequence of its component states, which establish a behavioural model for the system in several levels of complexity. In the Yourdon/Coad Method, the states are modelled through the following diagrams (YOURDON & COAD,92).

Object State diagram- it presents all different states of an object throughout time. This diagram identifies states and transitions from one state to another. Through this diagram the detailed behaviour of an object is defined with its service specifications.

Service diagram- it specifies all services related to a class. Each diagram should be establish per service. It includes attribute specifications, with their corresponding input-output.

Message connection diagram (shown on figure 3.8)- it models interactions among objects. Each message deals with a set of values sent to a particular object and the outcome of this method.

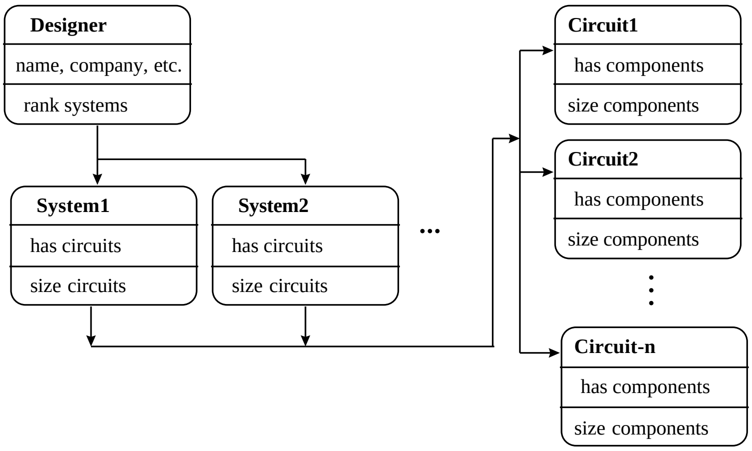


Figure 3.8- Message connection diagram.

Figure 3.8 exemplifies a connection message diagram, involving designer, system and circuit objects. As can be seen, the designer object sends messages to system objects (for example, create components or size components), the system objects pass similar messages to their corresponding circuits, which process the messages according to specific lists of components defined in the **has components** slot. This process repeats for other procedures implemented in the prototype, as explained in chapter 5.

It should be noticed that the designer object is only an attempt to model some properties of an actual designer, examples of such properties are: name, company, email, ranking option, different weights for the selection criteria, customer name, etc.. Due to the complexity of the design activity and also because of different profiles for designers (conservative, creative, team-oriented, etc.) a complete modelling of the designer is beyond the scope of this application.

As mentioned before, this section only aims to demonstrate the Yourdon/Coad method applying concepts related to the present project, not to describe the whole system, which is the objective of chapter 5. Therefore, here only few aspects, in a more generic form, were represented. Based on the comprehensiveness and simplicity obtained from its hierarchical approach (the five levels) as well as in the experience gained from the prototype implementation so far, it can be said the Yourdon/Coad method has proved to be applicable to modelling the task of hydraulic system design.

3.8- Knowledge Systematisation

Besides the choice of representation technique, the knowledge base definition also includes processes directly related to knowledge systematisation, such as (TOMIYAMA et al.,94):

- Setting up a view: To define a background theory which dictates conceptual relationships among its own vocabulary and to define the scope and focus of attention.
- Articulation: To identify instances of concepts that belong to the view and to give them representations of the background theory.
- Codification: To find out structural relationships among instances articulated in the previous stage based on the background theory by codifying them. This process results in generating pieces of factual knowledge.
- Crystallisation: To generate general, abstract knowledge from purely factual knowledge, which can be called a theory. This theory will be tested against the background theory that can be improved, abandoned or taken as it is.
- Reusing and Sharing of Knowledge: These are the goals of systematising knowledge and can be achieved by having not only a common knowledge description format, but also terminological, taxonomical and ontological level standardisation.

The present project involves very much the first three processes, for it directly models concepts based on a background theory, i.e. hydraulic system design. It identifies and articulates those concepts (load, system, circuit, component and designer) according to a knowledge representation technique. This project also aims to codify the concepts in an expert system prototype in order to validate the approach.

Crystallisation and knowledge sharing, as presented above, are beyond the scope of this project. However, the prototype application as a working platform in an industrial environment can bring another perspective to the design of hydraulic systems. Mainly, because the project involves a direct handling of concepts, such as alternative solution generation and concurrent engineering aspects, that despite of being known as theory by the design methodology community for a long time are not always applied by the designers of hydraulic system. This aspect has been noticed throughout the project development through the intensive interaction that the project has had with market sectors (i.e. hydraulic system suppliers and designers), which have emphasised the usefulness of those concepts for them represented in a computer environment. Further aspects related to this market feedback are pointed out in chapters 5, 6 and 7.

Chapter Four
Hydraulic System Design Issues

The previous chapters dealt with concurrent engineering, design methodologies and expert system issues to establish a common background to understand the project development. In order to prepare for the prototype description, it is necessary to point out some aspects directly related to the knowledge domain used in this project, i.e. hydraulic system design, therefore the present chapter will address some issues on this domain. The chapter also discusses some modelling techniques related to hydraulics. It presents the modular approach to hydraulic system design as a means to facilitate the design task; it also describes the most desirable characteristics for computational systems in general, with specific points about expert systems, applied to hydraulics.

4.1- Justification for Hydraulics

Next, some justifications which support the adequacy of applying hydraulic systems design for expert system development and object-oriented approach are presented (SILVA & CHEUNG,97):

- Hydraulics is a very broad field of research and has some aspects, that are explained later, which make it appropriate for the development an expert system.
- Hydraulic systems modelling and analysis have a well established theory, and this aspect makes the area applicable for Concurrent Engineering Projects, which are more suitable for so-called conventional technologies (WILSON,90).
- Hydraulic systems are defined by their components, each one has a specific function in the system. This aspect makes a close relationship between the system functional structure and its physical model (KIAN & CHEONG,93), and also it eases the application of Object-Oriented Techniques in computer support tools, explained in chapter 3.
- Due to the great extent of this field, the most important product aspects (design for cost, assembly, safety, reliability, and so forth) can be directly included in the support system, or they can be considered through guidelines. Those general aspects are found in whatever class of product (WAY,93).
- There is a close analogy among hydraulic, pneumatic and electrical systems. Thus a computational environment for hydraulic systems may be relatively easy to adapt to design other kinds of system.

- As a branch of mechatronics, hydraulics is a product oriented area, thus the market issues also play an important role in this field (BUUR,90).

Besides the points that make hydraulics an useful application area for this development, it is important to mention the advantages of hydraulics compared to other means of actuation. The principal advantage is the greater force available for a given actuator size and weight which often allows a more compact system with greater dynamic response particularly where large inertias have to be moved. Properly designed systems which are well manufactured will give notably reliable and leak free service. The following characteristics of electrohydraulic systems can often be used to provide performance advantages and lower system costs (BFPA,96):

- Higher output force compared to electro-mechanical drives, often eliminating the need for gear reduction or lead screws and the associated backlash problem;

- Higher output stiffness compared to electric drives giving higher dynamic response in direct drive systems;

- Lower system weight and size, particularly if one flow source serves a number of actuators, or if the actuators can be part of the system structure;

- Simple inclusion of check valves or solenoid operated valves to maintain actuator position or ensure specific actuator behaviour if electrical power or other prime mover power is lost;

- Ease of heat removal from the point of operation.

Despite the above advantages, hydraulics also presents drawbacks, some of them are (OHHASI,91):

- Hydraulic systems require pump units as “flow generators” and associated devices. Generally, pump units are the primary sources of noise which is sometimes significant for high pressure systems;

- Leaks should be protected by careful attention to the selection of seals and couplings, otherwise oil leak soils surroundings;

- Usually, hydraulics requires cooling water or cooling air;

- Maintenance is necessary in maintaining cleanliness of oil and smooth movement of actuators more frequently than those for electrical systems.

Based on the above justifications and on the points regarding expert system applicability defined in the chapter 3, it is clear that hydraulic system design is appropriate to the present project. In addition to those aspects, the modular nature of hydraulic systems makes a close relationship between the physical representation and its functional modelling (BACK,83; BURROWS,93). A modular representation of a hydraulic system is depicted on figure 4.1.

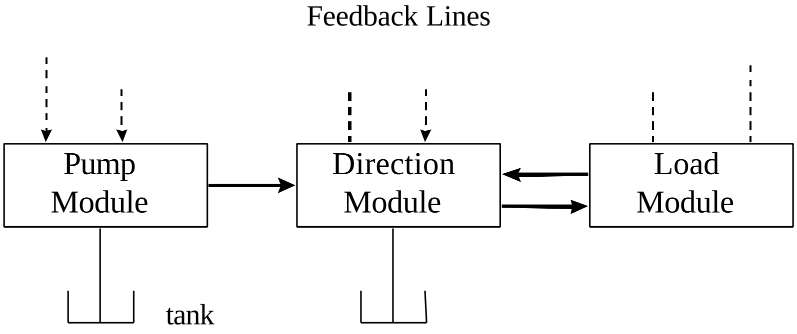


Figure 4.1- Typical modules for a hydraulic system (SARGENT et al., 88)

Within each module depicted on figure 4.1 exists a series of submodules related to different components. The design process essentially consists of establishing through knowledge of the load requirements, the absence or existence of a module and component (SARGENT et al., 88). Therefore, the computational system for supporting design is requested to query the user in a satisfactory manner to deduce the need for each submodule. Figure 4.1 is a functional description of a hydraulic system and the feedback lines represent the signal flow between the modules.

4.2- General Issues on Hydraulic System Design

Hydraulic systems are designed to perform one or more tasks which define a duty cycle, using hydraulic power to move the actuators. As already mentioned, these systems are made of components, such as, different types of pumps, valves, cylinders, hydraulic motors, prime movers, reservoirs, etc., being each one of them specifically chosen for a well defined function. This fundamental characteristic makes the product functionality explicitly represented down to the component levels, once their functions are clearly understood.

In order to define a higher level of functionality, and thus simplify the design task, the present development applies the concept of circuit, which maps directly the definition of working principle presented in chapter 2, and also agrees with the ways hydraulic system design is taught in most courses. Hydraulic circuits are set of components assembled to perform a specific function, i.e. a sub-function of the complete system. As examples of hydraulic circuits there are: power supply circuit, speed control circuits, force control circuits, position control circuits, etc.. With the circuit definition, it is possible to establish a hierarchical structure among system, circuit and component, which greatly facilitates the design task.

From a conceptual perspective, hydraulic system design can be broadly divided in two phases. First, it is necessary to functionally represent the load requirements in order to reason about the need for specific actuation circuits. Then, it is required to handle the quantitative load parameters and their interaction during the duty cycle. In other words, the design process must embrace the functional and behavioural descriptions of a hydraulic system (LÜCKE et al. ,95). The behavioural description can take place through graphs which define the power variables (force/speed or torque/rpm) for each load in a time scale, hence describing their interaction throughout the duty cycle.

An alternative approach is to represent the loads via the description of maximum values for the power variables and the operational sets which roughly model how the loads are activated during a machine operation. Certainly, this approach is simpler but more limited than the previous one. Usually, the first one is more applicable when there is a clear specification in terms of sequence among the loads, e.g. the majority of automatic industrial equipment. However, in areas such as mobile hydraulics, a potential area for this project development, there is no clear definition of actuation time sequence. Moreover, considering that the present project primarily aims at the conceptual design phase, the second approach was chosen for representing the loads activation. This procedure will be more explained with the system description in chapter 5.

Besides these points, it is necessary to consider: pipe sizing, pressure losses, component selection and other requirements (OHHASI,91). In order to systematise the design process, a conventional design procedure for hydraulic systems will follow a typical sequence:

-System general definition, including load requirements to define the need for different circuits;

-Determination of operational and environmental conditions, including tolerances in terms of type of errors (position, speed, force) and efficiency range;

-Definition of maximum required pressure based on design guidelines;

-Determination of the overall system with circuit definitions;

-Actuator sizing and selection;

-Sizing and selection of power supply circuits, including filtration range;

-Valve and pipe sizing; reservoir sizing and system filtration;

-Fluid selection;

-Maintenance planning.

Another aspect to be considered on hydraulic system design is to provide information to support the selection of the control strategy. In hydraulic systems, the direct control parameter is flow, for pressure at the actuator is an outcome of the movement resistance. Therefore, through the flow control to an actuator is possible to control its speed, acceleration, position, force (or torque) and the resulting pressure. These parameters are difficult to control simultaneously and different control strategies are selected depending on which parameter has to be controlled. There exist two general concepts to apply for flow control, they are pump control and valve control, as depicted on figure 4.2.

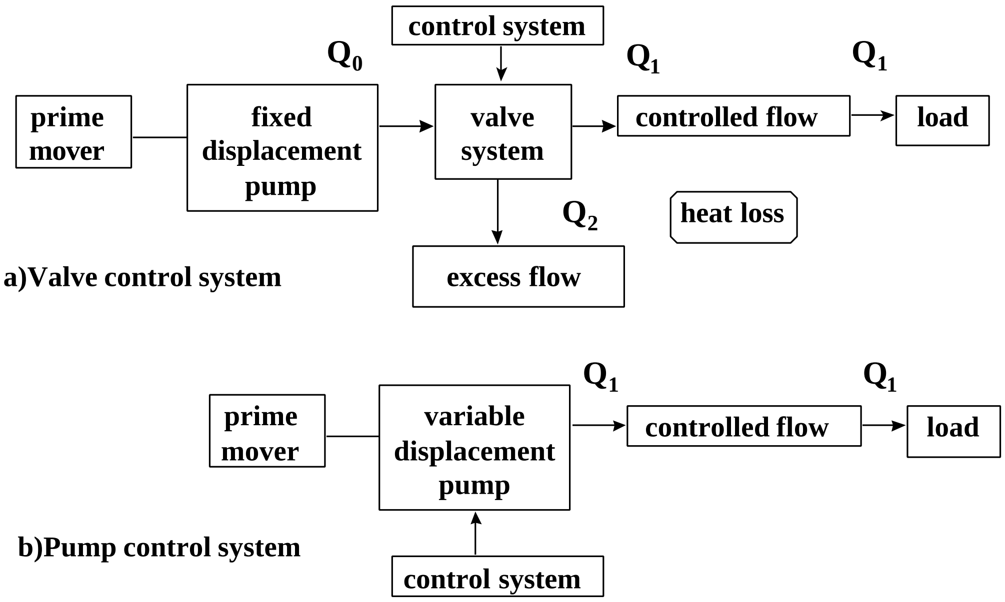


Figure 4.2. Valve and Pump Control Strategies (OHHASI,91).

As depicted on figure 4.2, in a valve control system, Q1 is a result of flow diversion from a fixed displacement pump, as a consequence of such strategy, wasteful heat generation takes place, whose value is proportional to the product of the diverted flow Q2 by the corresponding pressure. In other hand, in the pump control strategy, resulting from a variable displacement flow source, the generated flow is directly dependent on the required linear or angular speed. This strategy generates much less power loss and thus offers a better efficiency. Table 4.1 presents the comparative analysis between these strategies (OHHASI,91).

Table 4.1. Comparison between valve and pump control.

Strategy	**Valve control**	**Pump control**
Pump cost	Usually lower.	High
Efficiency	Low, due to excess flow.	Better, no excess flow.
Pressure	Limited by the specification of valves and generated heat.	Suitable for higher pressure systems because of less valves.
Flow rate	Limited by the valve capacity and efficiency.	Suitable for larger flow rate, no limitation by valves.
Controllability	Usually, the response is fast, suitable for high-speed use. Valves can be arranged very close to the load.	Limited by response of control system of variable pump and the distance between the pump unit and the load.
Simultaneous operation of plural loads.	Possible by using plural control valves and control signals.	Difficult in principle, the system has only one control parameter in the pump unit.
Characteristics	Usually suitable for smaller size hydraulic systems of approximately under 30 kW or systems having plural loads.	Usually for large capacity system or systems requiring power efficiency or less heat generation as machine tools.

Despite the apparent advantage of the pump control strategy, there are cases that limit its applicability, as depicted on table 4.1. As this comparison is based on information from a specific manufacturer, the rule of thumb regarding 30 kW can be variable depending on different suppliers. The type of information in this table is represented in a very simple manner which makes it easier to be implemented in an expert system. In fact, the ability to analyse and change interactively the power supply circuit and control strategy is one of the main characteristics implemented in the developing system, as shown in the chapter 5. Obviously, hydraulic system design involves further consideration which will be also explained in greater details during the system description.

4.3- Knowledge Engineer and Domain Expert Interaction

In order to develop an expert system focusing on the Concurrent Engineering aspects of hydraulic system design, one of the paramount features concerns the identification of the main drawbacks usually found when applying the Sequential Engineering Approach in hydraulics. The earlier these points can be identified the better and faster will be the design process, for most of the detailed reworks that occur in the Sequential Engineering approach will be avoided. In order to define the drawbacks, a series of structured interviews was performed with different experts in hydraulics (SILVA & DAWSON, 97b). Some issues from these interviews are reported next, where KE and DE stand for Knowledge Engineer and Domain Expert respectively.

KE- What are the main BOTTLENECKS in the hydraulic system design process? And Why?

DE- *Getting good information from machine designer due to his lack of fluid power knowledge.*

Getting the machine designer to make decisions about the circuit.

Finding the right kind of equipment to efficiently meet the needs of the circuit that are readily available and not over-priced.

Based on this aspect it was made clear that one of the applications of the developing system will be as a front-end for an end user, who usually has a limited knowledge about fluid power. It also provided an insight to create an advice tool, which could offer support in some decisions, thus facilitating the design process.

In the process of knowledge acquisition, conflicts usually appeared when using different experts for building the knowledge base. In hydraulic system design, the definition of the supply pressure is considered one of the first compromising or constraining decisions, for it will directly influence the power consumption, heat generation, component sizing, and so on. Next, some aspects are presented that related to this issue, i.e. different experts' viewpoints concerning the pressure supply definition. The quotations relating application areas and supply pressure were extracted from a handbook (GÖTZ,84) and submitted to the **DE**. The next part presents some closed questions, to which the answer must be very specific.

KE- Some references suggest that the operating pressure (bar) is dependent on the application area, see below:
Do you agree with such definition? Why?
(application (area agricultural-machinery) (range 150 200))
DE- *I would think this was low more like to be 150-300 bar.*
(application (area construction-machinery) (range 100 250))
DE- *Likewise here also 150-300 bar.*

After analysing other experts' opinions, it became clear that consensus would not be achieved in this matter, reflecting on the influence of market segmentation. Another expert pointed out that the supply pressure should be limited to 215 bar to avoid excessive noise. Therefore, the decision implemented is that the expert system must be capable of providing as much information as possible, keeping an allowable range for the supply pressure (for example between 100 and 315 bar) and leaving the final decision for the user.

Usually, hydraulic systems are applied in areas that require high performance in terms of precision and power/weight ratio. However, the specific knowledge required in hydraulics is generally dependent on the supplier or designer (components supplier or consultant) rather than on the user (machine builder, OEM- Original Equipment Manufacturer, process company, etc.). Therefore, the aspects concerning the performance of a hydraulic system should be included in the knowledge base. Some of these aspects are presented in the next dialogue with the same expert.

> **KE**- What are the most common points found that define an unsatisfactory performance? And what are your proposals to increase the efficiency based on each point?
>
> ***DE**- a- Over-heating even with large heat exchangers.*
>
> *Prop.: Reduce or eliminate wasted energy.*
>
> *b- Short component life. I see pumps that should last years require changing in months. Prop.: Filtration, cooling, different pump design, better component application.*
>
> *c- Excess and/or unnecessary components that only complicate the system and trouble-shooting. Prop.: Eliminate them.*
>
> *d- Undersized fluid conductors that may also be poorly installed.*
>
> *Prop.: System repipe.*
>
> *e- Little or no setup or operation information on the schematic.*
>
> *Prop.: Update schematic.*
>
> *f- No use of circuits such as regeneration to reduce maximum system flow.*
>
> *Prop.: Use any and all techniques to reduce system flow without losing cycle time .*
>
> *g- Using valves that work but are not the best for the application.*
>
> *Prop.: Replace the wrong or obsolete components.*

Although not all these issues have been considered yet, some of them had impact in the decision making process related to the prototype. For example, as far as the point *c* (above) is concerned, it brought more specific questions related to the Circuit Objects definition, a key concept in the developing system (SILVA & CHEUNG,97), such as *How are the circuit objects defined?*, in terms of their components, graphical representation and explanation. The other points are more applicable in the development of quantitative load attributes. Other aspects, discussed in more details in the next chapters, are:

a) The system must provide general guidelines to select a reputable supplier, for this is a relevant point in the design process. End users must be able to configure a hydraulic system to aid selections based on their actual supplier base. Although contacts with suppliers have been made, this aspect remains to be implemented, but through the contacts and tests the system structure has been proved to be capable of expanding, as shown in chapter 6.

b) Redefinition of the load requirement inputs interface based on the user's knowledge profile. This aspect was a consequence of the first feedback from the Domain Expert, who despite having experience in hydraulics, was not familiar with the Bond Graph terminology (KARNOPP et al.,90), previously used in the phase of load requirement definition. Therefore, the development of the interface was adapted according to the user's technical background.

c) The output interface of the prototype was also influenced by an expert's feedback, for in the earlier version of the system only a graphical output for a proprietary simulation package, DYMOLA, was provided (ELMQVIST et al.,93). Because the above mentioned expert did not possess this package, he was not able to visualise the hydraulic system diagrams, and thus reported a disappointed reply. This feedback caused the search for alternative solutions for the graphical outputs. This search generated the development of a computational agent to create outputs in HTML format readable via an Internet browser (SILVA & DAWSON, 97c). This topic as well as other issues related to the knowledge acquisition process are expanded in the next chapters.

In the context of an expert system, an important aspect regards the definition of granularity, i.e. "at what level of detail should the world be represented?" (RICH & KNIGHT,91). As mentioned before, in this development the smallest functional unit is the hydraulic component, for each component in itself is a relatively complex system, whose analysis involves handling of mechanical, hydraulic and sometimes electrical parameters (LINSINGEN et al. ,91; LINSINGEN et al.,92). Therefore, in the present work only the component external properties, necessary to represent their connectivity will be modelled.

4.4- Computational System for Hydraulics

From the above points, it becomes clear how complex and comprehensive the development of computational system for hydraulics is. In addition, several researches have been done by different institutions and some commercial packages have been developed to explore this potentiality. Most researches have concentrated on the methodological aspects of computational tools for hydraulics and on the detailed points related to dynamic modelling and simulation of hydraulic systems. Specially this second point has been receiving greater attention from the research community. This project, however, presents detailed implementation aspects related to a design tool and more preliminary points as far as simulation is concerned, and thus, it does complement previous researches.

In order to demonstrate the vast potentiality of computer tool applications on the field of hydraulics, next it is quoted a comprehensive list of the possible applications (BURROWS,93):

> A user-friendly interface: windows, icons, menus and pop-ups; Rapid production and reconfiguration of circuit layout; Expert assistance; Physical/mathematical models of proprietary components; Large database of components models; Facility for users to add new models; Component model selection from pop-up menus; Component parameter data-realistic default values; Rapid dynamic and steady-state solutions (time and frequency domains); Thermal simulations; Definable fluid properties; Interchangeable units (SI, Imperial); Automatic optimisation; Compatibility of other packages (controls, etc.); Real-time simulation- ability to incorporate hardware; Hybrid system simulation- hydraulic, pneumatic, mechanical and electrical; Expert help in identification of potential malfunction; Life, safety and reliability- automated FMEA and FTA; Assistance in condition/health monitoring; Safety; Spatial (three-dimension) layout; Integrated draught/ part list/ Bill of materials; Component availability; Prediction of noise levels and methods of reduction; Selection and siting of filters.

It is important to reinforce that such list shows more the potentialities of applications than what is actually already available in terms of computational tool. In fact, as that author presented it, this list is “the ideal fluid power simulation package- a wish list of features”.

Similarly to the list of desirable characteristics for a computational environment to support a Concurrent Engineering team, presented in chapter 2, to accomplish the above listed features is a multi-task project which would require a team effort of years, or even decades, to be satisfactorily concluded. Therefore, correspondingly as pointed out in chapter 2, although the present work directly addresses some of the above features (e.g.: User-friendly interface; Rapid production and reconfiguration of circuit layout; Expert assistance; and Rapid dynamic and steady-state solutions), it does not complete all points, rather the criterion here is to establish a framework sufficiently expandable and modular to serve as a basis for integrated projects in this area, but at the same time supporting the solution of key sub-problems listed above.

The importance among the listed features depends on the needs and abilities of the system user. For example, if the user has a considerable understanding of computational modelling, he/she can wish an open environment to include more sophisticated component models. In other hand, if the user is not so familiar with hydraulics, he/she might prefer the availability of a comprehensive library of standard components, with most of computational modelling being transparent to him/her.

Based on the aspects necessary to design hydraulic systems, it becomes evident that such activity involves a considerable degree of expertise, which is not always available in an acceptable cost and time scale. This means that a decision making supporting tool could facilitate the interaction with the expert, or even educate an inexperienced engineer on the field. Therefore, there are clear benefits of applying the expert system approach to hydraulic system design, some of which are presented as follows (SARGENT et al., 88):

1. More efficient, consistent, predictable circuit designs, including appropriate applications of special-purpose and/or energy efficient components;
2. Total integration of computer technology into hydraulic circuit design and analysis precluding the necessity (in some situations at least) of prototype hardware and laboratory testing (if used in conjunction with existing CAD programs);
3. Efficient use of the designer’s time freeing him to concentrate on more esoteric and unusual designs;
4. Better understanding by both the expert and the neophyte designer of the human decision-making process used in circuit design (educational tool);
5. Preservation of design knowledge that otherwise would be lost through attrition, retirement or death;
6. The ability to modify the knowledge base as new components or circuit design techniques are developed.

In the context of this project, the prototype has been developed using as much as possible a modular approach to consider different aspects of the design process, such as hydraulic system configuration, system sizing, component selection, guidelines about maintenance, safety and so on. Thus the design process can be systematised based on the different load attributes used for different modules. The limitations of a system developed with this method is virtually defined only by the knowledge base size, not its structure. As the knowledge base evolves in an incremental pattern (defined in chapter 3), the present limitations due to time constraint for implementation are only temporary and not fundamental, i.e. they do not compromise further enhancements. Chapters 5 and 6 demonstrate how the knowledge base increased in terms of rules, message-handlers and classes throughout the development process.

In the component selection topic, a typical application of an expert system can be to choose among different types of pumps, taking into consideration technical, economical and operational aspects, as described on figure 4.3. This figure presents the interaction among different aspects necessary to select a particular type of pump. These guidelines are well known and based on experience gathered from practical designs. The manner in which the decision making process for a key component is depicted demonstrates a close relationship to an expert system approach.

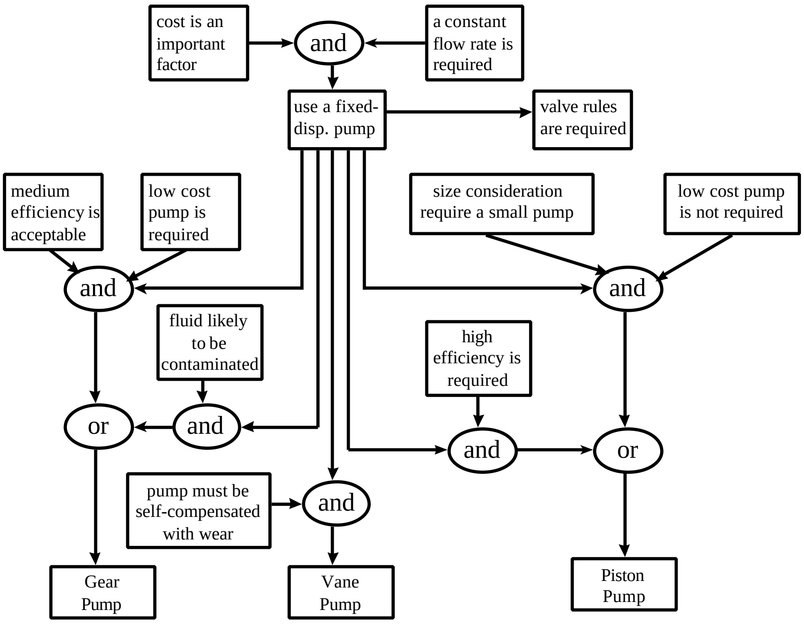


Figure 4.3. Flow chart used in choosing a particular type of pump (SARGENT et al., 88).

Considering that a similar process can be applied to other types of components, it becomes clear the usefulness of an expert system for automating this task and at the same time it demonstrates that the complexity for component selection can be managed through this approach. Although the prototype deals more with the general system configuration, it also addresses the component selection issue in terms of implementation for some components and conceptually for others.

4.5- Consideration about Electro-hydraulic Systems

With the increasing demand for high performance and automatic systems, the topic of proportional or servo hydraulic circuits has gained a special attention in the design context. Thus, it is important to point out what their specific properties are, how they are represented, and which objectives in terms of performance can be obtained from the application of this type of circuit.

A closed-loop servo can either be a regulator or a follow-up system. In a regulator type of system the control loop objective is to maintain the output at a given value independent of all system disturbances. The thermostat loop on a gas oven is a regulator system where the function of the loop is to maintain the oven setting at the desired value independent of the oven load and external cooling. A follow-up system is one in which the input function is constantly changing and the output is controlled to follow the input. A typical example of a follow-up control is a copying lathe in which the cutting tool position (the system output) follows a path traced from a template (PINCHES & ASHBY,89). Figure 4.4 illustrates a servo system generic structure.

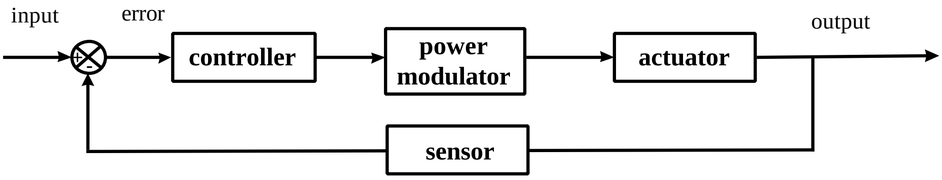


Figure 4.4. General Servo System Structure.

As depicted on figure 4.4, the servo system conceptual structure consists of the following elements: controller, power modulator, actuator and sensor. Similarly to a hydraulic system, each component has a specific function, thus the system physical model maps exactly the functional structure. It should be noted that the above structure is domain independent, this means it is applicable for servo mechanisms in general, regardless of the energy domain used, i.e. hydraulics, pneumatics or electro-mechanics. Although each of these domains has some particular aspects, they share the same conceptual structure.

Because of the above mentioned characteristics, in servo hydraulic systems the dynamic aspects play a much more important role than in conventional hydraulics. This means that the dynamic properties are as important as the general functional properties to accomplish the system performance.

Over the years many control strategies have been developed ranging from simple proportional control, through PID control (Proportional, Integral and Derivative) to advanced digital control methods which may be self setting or adaptive. Despite this, the basic principle is straightforward; the controller sets the demand, that is the required position, velocity, load or pressure; a sensor or transducer feeds back the current value of the variable; the controller compares the demand with the actual value, determines the error and applies a signal to the control valve (power modulator), in the appropriate direction, such that the system moves to minimise the error. By actively monitoring where the system is, comparing it with the position the controller wants it to be, the loop is closed and errors can be minimised (BFPA,96).

Here, some points directly related to electro-hydraulic systems are presented, some of them were addressed on the developing system, as described in chapter 5, while others are considered for future developments.

Designing an electro-hydraulic position control system is often considered difficult but for lightly loaded systems the approximate performance may be calculated with relative ease (CLEASBY,96). This reference suggests a trapezoidal speed profile, see figure 4.5, to establish a relation between the amplifier gain and the acceleration and deceleration times.

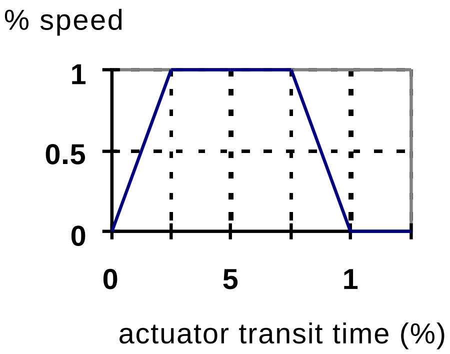


Figure 4.5. Trapezoidal Speed Profile (CLEASBY,96).

Although the reference does not define exactly what a lightly loaded system means, in a personal letter to the knowledge engineer, replying to some questions about the paper, that author pointed out:

> *"the mathematics of the framework system (what little is) ignores the compressibility of the oil and the forces required to accelerate the load on the actuator. If these two items were significant the velocity of the actuator would not be proportional to the position of the spool in the electrohydraulic valve and over-shoot may occur when the valve is closed and possibly inoperative. … I would define a lightly loaded system as*

> *one in which the pressures required for acceleration were less than 1/20th of the pump pressure."*

In this comment the framework system represents the mathematical model of a position control system, i.e. valve, controller, actuator and sensor. This type of heuristics can be either implemented as an automatic rule to define about sizing procedures, based on the pressure values, or it can be used as a design guideline. In this project both approaches will be discussed. Although (CLEASBY,96) is recognised among the companies related to hydraulics in the UK as one source of reliable experience, it should be pointed out that, in order to implement such heuristics in the knowledge base, a more comprehensive analysis should be done, for this heuristics represents the experience of one expert, which sometimes contradicts others.

Another heuristics defined in the above mentioned reference states that: "For systems with asymmetrical cylinders, design the framework system using the valve flow and cylinder velocity for movement in the direction of the lowest cylinder velocity. Then add a sign dependent attenuator at the input to the valve so that the same velocity is obtained in the opposite direction." As asymmetrical cylinders are very common, this specific knowledge must be available during the design task. The reference also presents guidelines to predict and improve actuator position accuracy; and specify the valve and system gains. The manner in which these guidelines can be implemented for circuit definition as well as for component parameter selection will be described in the next chapters.

4.6- Hydraulic System Behavioural Modelling

Although the prototype is more concentrated on the functional description of a hydraulic system, as mentioned before, the behavioural modelling is also a key aspect. Therefore, this section presents a technique applied to model hydraulic systems in this context, i.e. behaviour modelling.

The section roughly describes the GRAFCET technique (MARCÉ & LE PARC,92), addressing the following issues: general definition, its application, why the technique has not been implemented in the developing system and its potential for implementation.

GRAFCET- Historical Background and General Description

GRAFCET (the acronym of GRAphe Fonctionelle de Commande Etape/Transition or, in English, Step Transition Function Charts) is a graphical method for specifying industrial automation. Sometimes referred to as sequential function charts (SFC), GRAFCET was invented back in seventies as a documentation tool to allow easy communication about control systems for people of various backgrounds (mechanical engineers and control specialists for example) (PERRON,96).

The basic concepts of this discrete system modelling were, and remain today, quite clear and simple: the step, the action, the transition, and the condition associated to transition.

- Step represents a partial state of system, in which an action is performed. The step can be active or idle. The associated action is performed when the step is active or idle.
- The associated action is performed when the step is active and remains asleep when the step is idle.
- The transition which links the previous step (one or several) of any transition and the following step (one or several) represents the fact that the action(s) of the previous step(s) is followed be the action(s) of the following one(s) and figures a decision of changing system state. Nevertheless, changing is under control of two conditions:
- every step to the transition must be active;
- a boolean condition associated with the transition must be true.

As can be seen, due to its characteristics, GRAFCET became a common programming language for Programmable Controllers which are devoted to control of industrial processes, such as control of nuclear plants, chemical reactions, metallurgy and so on. Most of these applications have to be very secure, and so it is necessary to develop tools to verify programs correctness (MARCÉ & LE PARC,92).

Implementation Issues

Despite its modelling capabilities and industrial applicability, GRAFCET has not been directly considered in this developing system, for the following reasons:

1. As stated above, this technique was originated, and is mainly applicable, to sequential systems. Although there are several hydraulic systems applied to sequential operations, this feature reflects more one additional attribute of a specific sub-set of hydraulic systems as far as their behavioural characteristics are concerned. These characteristics can be considered during the

detailed design stage, providing that the decisions implemented on the conceptual phase allow one to do so.

2. The developing system is more focused on general hydraulic system attributes needed during the conceptual design phase. It also provides a preliminary facility to model the interrelations (parallelism and sequencing) among the different actuation circuits through the operational set definition, as described in chapter 5. This alternative approach, although more limited than GRAFCET, is enough to model the basic sizing of all hydraulic circuits and components. Moreover, the present approach does not impede, in the way the system has been developed, further implementations as described in chapter 7.

3. As described in chapter 3, a key aspect required to develop an expert system is the experts' availability. Although some experts were available regarding the GRAFCET modelling (mainly in Brazil), most of the closest experts were unfamiliar with this technique. Perhaps this reflects a market segmentation in hydraulics or even a more geographical aspect, for the closest experts in this project were from USA and UK, and the technique was mainly developed in France. Therefore, due to a limited time scale, the knowledge engineer had to strike a balance regarding the GRAFCET implementation. Another point is that this issue was brought into the project in a much later stage in its development, when the basic prototype structure, functionality and scope had already been defined. However, the next chapters will discuss alternatives to model this technique in further developments.

Finally, it is important to mention that GRAFCET is a formal flowchart method supported by international standards IEC-848 and IEC-1131. This technique will become more popular among fluid power technologies since its use is recommended in the ISO 1219-2 (Fluid power systems and components- Graphics symbols and circuit diagrams). ISO 1219-2 refers to the use of GRAFCET to supply sequence descriptions along with the design documentation (PERRON,96).

4.7- Hydraulic System Design Packages.

From the experience gathered during this project, mainly through an intensive sequence of tests and demonstrations to industry and academy as well, it appears that the approach, methodology and scope of the prototype are quite original for this application area, i.e. fluid power system design. However, several computational systems to support the design task of fluid power systems do exist, though with different applicability and approach. The next section is an attempt to describe, albeit in a limited form, some of these systems functionality. The reader is invited to search for more explanations directly within the corresponding references.

In order to represent in a precise form, as much as possible, the general descriptions are taken directly from the systems material and homepages, be it help files or marketing leaflets. Hence, in no way does this section try to undermine the specific systems or judge their usefulness, rather it tries to offer a general view about the available tools and comment on them.

4.7.1-HYDRO ANALYST

Description taken from the Help file of the system, version 2.0 (FLOTRON,96). The system offers three modules: Motion Control, Force Control and Power Efficiency. These modules are indeed standalone systems, which can work separately: There are three basic elements to the Motion control module:

1. HYDRAULIC SYSTEM DESIGNER

This element of the program performs two distinct functions which can be summed as follows : Investigating the integrity of the hydraulic transmission and formulating a transfer function which describes the dynamic behaviour of the hydraulic transmission.

The basic elements making up the transmission are as follows :

The power source, normally a positive displacement pump.

The control device, either a proportional or servo valve or hydrostatic transmission. The actuator i.e. a single or double ended cylinder, rotary actuator or hydraulic motor. The control device falls into one of two categories : Flow controls and Pressure controls

2. ELECTRONIC SYSTEM DESIGNER and FREQUENCY RESPONSE ANALYSER (FREQUENCY DOMAIN)

The three basic functions performed by this module can be summarised as follows :

i) Derivation of the overall system transfer function

ii) Analysis and synthesis of system stability

iii) Prediction of system dynamic performance

The information required to formulate the complete system transfer function describing the dynamic behaviour of the hydraulic transmission is automatically obtained using the 'Hydraulic

System Designer' transfer function, and a transfer function for the components e.g. valve and feedback transducer from the 'Component Database'.

3. TIME DOMAIN ANALYSER.

As the name implies, this part of the program investigates system performance in the time domain. It is a single purpose module whose objective can be summed up as "The prediction of system transient response to a given input stimulus". The input stimulus can cover a wide range from a simple step demand to a compound duty cycle with up to eight transitions. An impulse either generated as an additional input signal or caused by an external load disturbance can be superimposed at any time interval.

Elements of the dynamic system identification are automatically transferred from the frequency domain, and the required input stimulus is now specified. The effect on response time and overshoot of applying either a ramp-step or triangular input can be displayed by selecting the appropriate graphic.

Alternatively any duty cycle can be entered by specifying the time period and polarised change of amplitude. The program includes a continuous function generator to input the following waveforms Sinusoidal, Triangular, Reverse Triangular, Square, Reverse Square, Saw Tooth, Inverted Saw Tooth.

The **Force Control** module offers the following features:

This program is confined to systems controlled by a proportional pressure control valve and assumes a dead headed system. The structure of the program is very similar to that for flow control but the user is faced with a different set of system identification parameters appropriate to a pressure control system. FORCE CONTROL can be achieved by one of three methods :

Method 1: By using a proportional pressure control valve. Can be either a closed or open loop system.

Method 2: By using a proportional flow control valve. Can only operate as a closed loop system.

Method 3: By using shunt leakage force control.

The **Power Efficiency and Dissipation module** accomplishes the following functionality:

This program is selected by clicking the appropriate icon in the program group. It is fully integrated with the main program and will give at a glance the theoretical power efficiencies and power dissipation of the system being reviewed under a series of selected circuit modes i.e. Fixed displacement pump, Pressure match system, Pressure compensated pump, Power match system, and Bleed-off system. The program also shows effective power output for the chosen system.

Comments: Based on tests of this system, the knowledge engineer was able to analyse some of its features. As can be noticed, the system is very comprehensive, for it embraces basically all steps necessary for a fully functional design of a hydraulic system. It has several pre-defined circuits, with default values and a formal description. The software presents information based on an one-to-one circuit, i.e. only one actuation circuit is presented individually connected to the power supply circuit. Therefore, it is not very clear how to size the entire hydraulic system, considering that in general a system is composed of many actuation circuits. The program has a well organised help system, which is very necessary due to the requirements in terms of technical background in hydraulics needed to use the system. The program handles either linear or rotational circuits at once, this means there is no facility to manipulate in the same hydraulic system different types of loads. As mentioned before, these comments are only a limited attempt to define some system functionality to the reader, and as such they should be considered.

4.7.2-AutomationStudio

Automation Studio[1] allows users to create designs integrating more than one diagram. This is helpful to separate diagrams by function and category. During simulation, all diagrams interact with each other.

Quick simulation operation. The simulation mode can be triggered by the simple click of an icon. You can further control the simulation pace with functions such as Full-speed, Slow-motion, Step-by-step and Pause.

Easy and integrated editing. With Automation Studio, there is no need for additional editing software, it provides editing toolbars and pull-down menus that are easy to learn and use.

Full colour simulation. During simulation, circuits come to life! Components become animated and lines are colour-coded according to their states. On-line pictures and full colour cross-section animation further enhance the simulation features.

Modular libraries

Automation Studio allows to choose from a variety of optional library modules. Additional component libraries may be purchased and added to your package to meet your expanding needs. This flexibility allows you to purchase only what is needed, thus limiting your initial investment. You can select from the following libraries: Ladder Logic, Pneumatics, Hydraulics, Digital Electronics, Grafcet (SFC), Electric Power, Function Blocks. The last three libraries are not available yet. A comprehensive and modular library. The library displays component categories in a comprehensive fashion. Simply browse through the list and see the graphical component representation in the lower part of the window. Then select and drag the component onto the schematic.

Comments: the knowledge engineer had an opportunity to see a demo of this system during the Hydraulics & Pneumatics show in Pittsburgh, November 1997, and to interact with some of its developers. This software is very comprehensive, it really integrates different libraries and presents information in a very user-friendly interface, it also presents schematic information about components. Although the description explicitly mentions simulation, in fact, this system processes an animation, which causes a great impact in terms of acceptability by the user. However, the system does not address the issue of dynamic modelling and simulation formally defined, i.e. to study the dynamic properties of the hydraulic systems and components. The software also requires an understanding of fluid power in order to use it, for the task of assembly the hydraulic circuits is an user’s responsibility. This task is carried out component by component using drag and drop facilities. From this brief description, it seems that this software also concentrates on the hydraulic system functional specification, this means that it does not address concurrent engineering approach as aimed in the prototype.

Incidentally, it is important to mention that this software does allow the modelling of fluid power systems using GRAFCET. Considering that the developer company FAMIC is based in Quebec (the French-speaking part of Canada), this may confirm the author’s view pointed out before that the application of this technique greatly depends on a cultural influence, i.e. GRAFCET might be much more spread among French speaking engineers.

4.7.3- CircuitWorks.

---

[1] Description taken from http://automationstudio.com/

CircuitWorks[2] is a specialized, completely self-contained drawing package designed exclusively and optimized for fluid power schematics. The program has been streamlined solely for the fluid power designer and really is the easiest to learn and use. Training time involves a couple of hours instead of weeks or months that conventional drafting programs require.

With the InsertKey utility program, symbols can be easily tagged with key information including part number, category, pressure, flow, etc. This information can be saved in a custom parts database for easy access later.

Use the bill of materials program. For quick and easy automatic documentation for the schematic. This information can be shown on the drawing or hidden as required. The easy to customize libraries have well over 600 symbols. A new symbol selector utility program allows easy viewing of multiple symbol libraries. It includes additional libraries from Vickers.

The output can be sent to printers, plotters, postscript printers, DXF format, etc. With the new Windows shell program, it can print using Windows' printer and plotter drivers, so printing problems disappear.

Comments: the author also had opportunity to see an explanation and interact with some of the system’s developer during the previously mentioned show. A great emphasis is placed on the Bill of Material, compared to the previous system. In terms of usability, this system also requires an understanding of fluid power in general, for the hydraulic system assembly task is carried out by the user component by component. This system does not address dynamic simulation, nor does it involve concurrent engineering aspects as considered in the present development.

Despite the limited descriptions and contacts that the author had with the above software systems, it is sufficiently clear that their scopes in the current versions have been addressing different issues compared to the prototype. It is important to emphasise that the above described systems are commercial packages, which have been available in the market for some years. The author also received a positive feedback from the later two software developers in the personal contacts during the show.

4.8- Simulation packages.

---

[2] The description can be found in http://www.techteamusa.com/cw.html

Although the prototype does not directly focus on dynamic modelling and simulation of hydraulic systems, it addresses this issue through the concept of computational agent, as described in chapter 5. However, due to the importance of this aspect for the design of fluid power systems, table 4.2 presents a list of computational systems specifically developed for this activity. This list was extracted from the site of Tampere University of Technology, Finland (http://matwww.ee.tut.fi/~piche/ fluidpower/).

Table 4.2- Some computational systems for Fluid Power Simulation.

Name	**Operating Systems**	**Library**	**Institution Web site**
BATHFP	UNIX	yes	http://www.bath.ac.uk/
EASY5	Unix, Macintosh, Windows NT, Windows95	yes	http://www.boeing.com/
HOPSAN	Unix, Macintosh, Windows	yes	http://hydra.ikp.liu.se/
HYDRO ANALYST	Dos, Windows	yes	http://www.flotron.co.uk/
SINDA/ FLUINT	Unix, Macintosh, Windows	yes	http://www.webcom.com/ ~crtech/sfcap.html
20-SIM	Unix (Sun), Windows	no	http://www.rt.el.utwente.nl/20sim/
VISSIM	Unix, Windows	no	http://www.vissim.com/
WINSIMU	DOS, Windows	yes	http://horsma.me.tut.fi/winsimu/

The list is placed here solely to demonstrate how many research and commercial institutions have been working in this area, and as such, it should be considered as a starting point for those who want to work in the field. However, the author has not analysed the presented systems, and thus, has no responsibility on the information obtained from them and on the links presented here, which were updated by the time the list was extracted, November 97.

It can be broadly understood that this chapter presented specific issues related to *why* hydraulics is a feasible area for an expert system for design; and *which* aspects can be covered by such system that were not considered in the already available systems for this area. However, the chapter does not explain *how* an expert system can be developed to address those issues. Therefore, the next chapter presents a description of the developing system, focusing on its evolution, functionality and demonstrating how the prototype directly provides a concurrent engineering perspective to the process of designing hydraulic systems.

Chapter Five

Prototype Description

The previous chapters offered a general view on concurrent engineering, design methodologies, expert system aspects and hydraulic system design, some of their sections also presented specific points on the developing system. Based on this background, here a more detailed description of the expert system to support the design of hydraulic systems is given. This chapter is defined in two major parts. First, a chronological description of the development process is given, which demonstrates the evolution in terms of system functionality as result of an intensive interaction with users throughout the project implementation. The reason for this part is to illustrate and document the process as well as to show potential for further enhancements. Secondly, the system modules are described in terms of their functionality and implementation details.

5.1. Initial Prototype Definition

The development of an Initial Prototype (IP) was considered part of the incremental process, whose definition is given in chapter 3. The main objective of the IP is to learn firsthand about the knowledge in the domain. Thus, its most important component should be a prototype knowledge base. The prototype knowledge base should be enough to solve some complete sub-problems from input to output, but be restricted so as not to present too great a development effort (GONZALEZ & DANKEL,93).

In the present stage, the developing system has an IP which is capable of designing several hydraulic systems based on the user’s specifications of the qualitative and quantitative load attributes. It handles the requirements to generate alternative solutions, combining 29 types of actuation circuits and 5 types of power supply circuits, each one with a specific functionality, description and set of components. This chapter points out the main aspects of this system focusing also in its potential for expansion.

5.2- Prototype Evolution

The purpose of this section is to demonstrate the system evolution throughout its conceptualisation and implementation phases. In order to so, a chronological description of the major milestones during the development is presented. This type of description is quite common in the AI literature (HART,92; RICH & KNIGHT,91).

At this point, the proposal for developing an expert system for

Project proposal	hydraulic system design was defined in a generic form.
August 95 First contact with EDC	The proposal was sent to the EDC Lancaster University and other UK institutions, as mentioned in chapter 2.
November 95 Proposal acceptance	This point marks the official acceptance of the proposal by the EDC. It also started a more detailed definition of the project.
April 96 Qualifying presentation	The qualifying project describing in details the proposal was accepted by UFSC. This document explicitly defined the link with EDC for the implementation phase.
September 96 Implementation phase start-up	This date defines the beginning of the implementation phase, for it corresponds to the start of work at EDC.
(26th)	In this day, the proposal was formally presented to all EDC staff. During the presentation the aspects regarding the interface between the knowledge engineer and other EDC members were also presented.
October 96 CLIPS start-up	The concepts presented on the proposal were generic, and thus independent of the implementation tool, providing that the shell tool supported Object-Oriented Modelling. However, it was clear that the implementation of a rapid prototype was a fundamental point in the project, therefore at this point, the knowledge engineer started to study CLIPS.
First industrial trip	The first industrial trip was to Denison, at this contact the proposal was presented and discussed with two managers (marketing and engineering). This contact was of vital importance, for it opened a gateway with the BFPA (British Fluid Power Association) for a future proposal presentation.
December 96 Main class definition (load, circuit and system). Explanation facility implemented.	At this step, the system was able to generate alternative hydraulic systems, handling basic circuits and load attributes. The system also offered a simple explanation on the reasoning process, i.e. why a specific circuit could accomplish a defined load.

January 97 Graphical representation	As one of the key points in the design task, the graphical representation of the system and circuit attributes was achieved. Initially, it was tailored according to the tool Dymodraw, defined later.
February 97 System documentation	Two papers on the developing systems were submitted in this step. One of them defined the basic points regarding the Object-Oriented Structure and the other related to the knowledge base structure and concurrent engineering issues.
Proposal presented to BFPA as a result from a previous contact.	As defined in the proposal, this Expert System development should involve as much and as early as possible the manufacturer market, therefore the proposal was presented at the BFPA technical committee. This opened an opportunity to discussion with an expert and broad audience.
Proposal summary published in the Internet	This point brought a new dimension to the development. As soon as the proposal was presented to BFPA, its summary was published in some specific sites in the Internet.
First contact from a domain expert	A consultant on the area of hydraulic system design from US (www.comsource.net/ ~budthyd/) who had a large industrial experience in hydraulic and pneumatic equipment design came across the proposal summary and decided to answer some points on it. This initiative originated a formal knowledge elicitation process through emails, by the end of 97 more than 70 messages were exchanged.
March 97 The first version of the system sent for expert's analysis.	As mentioned before, the development approach adopted in this project was based on an intensive involvement with different experts. Therefore, the first prototype version was sent, through email, for the US expert's analysis.
First feedback from the expert.	The first version was in standard text input and text output. It also had a graphical output tailored for the package DYMOLA. As the expert did not have this system, his feedback was at first quite

	disappointing.
Development of a HTML agent.	The disappointing feedback triggered the search for alternative ways to display graphical information. This search brought the idea of using the HTML language as an output for the system. The idea was almost immediately implemented and a new version with graphics was sent to the same expert, who replied in a very positive form. This started an on-going relationship and opened a new frontier for the system development, for it enabled the use of an Internet browser and other facilities.
May 97 Implementation of basic sizing for circuits and components	This included the coding of the equations to define the circuit attributes values, i.e. to implement the calculations according to the general literature in hydraulics, in order to define the basic values for variables, such as: valve pressure drop, actuator dimensions (cylinder or motor), valve rated flow, oil spring natural frequency, etc..
June 97 Inclusion of alternative power supply circuits	This facility allowed the user to analyse alternative circuits for power supply, including aspects about safety, cost and efficiency. Accordingly, the user could also freely exchange this circuit. This option greatly enhanced the usability of the prototype, as well as allowed a concurrent engineering approach during the conceptualisation of the hydraulic systems.
July 97 Handling of different systems of unit.	Due to the increasing involvement with the users, and because of their request, an option to input quantitative load attributes with different systems of unit was implemented.
Session option included	In order to provide a comparison among different load requirement inputs simultaneously, a directory structure was defined. Therefore, results from different inputs were placed into specific directories.
One expert decided to show the prototype.	As a consequence of the system development and due to intense interactions between the knowledge engineer and a specific domain

expert (i.e. the American consultant [1]), he spontaneously decided to demonstrate the prototype during one of his seminars at the Hydraulics & Pneumatics show- Pittsburgh, USA. Obviously, this initiative greatly boosted the interest and confidence in the prototype development.

Note: Until this date, this expert had tested only the standard CLIPS text-oriented version with graphical output in HTML. However, a graphical interface was being developed for few months. This point will be discussed later.

August 97
Component handling

At this time, the system was capable to handle some basic sizing properties down to the component level. Thus, a hierarchical design approach system-circuit-component had been fully demonstrated.

Graphical input interface version was sent to the expert.

The way the system was designed allowed a complete separation between the system knowledge base (i.e. rule and object definitions) and its interface. This point will be further explained during the description of the system functionality.

System documentation and demonstration

In this month the system was demonstrated as a poster session at ICED. Also three other papers were prepared, the knowledge acquisition process, the dynamic modelling agent and the HTML agent. They were presented in the next three months.

September 97
Designer information

By this time, the prototype had been demonstrated to different contacts and it was natural to expand it for handling generic information regarding the designer (name, company, email, etc.). Therefore, this new class was created for this purpose, initially with few attributes.

October 97
Ranking option implemented

As the proposal had stated, the developing system is focussed on the concurrent engineering approach. In order to enhance this feature, a weighting tool was implemented through which the designer can rank among alternative solutions based on general

---

[1] Bud Trinkel is a Certified Fluid Power Engineer with more than 30 years experience in hydraulic and pneumatic circuit design, trouble-shooting, and training. (http://www.comsource.net/~budthyd/)

	criteria, in terms of degree of importance, considering different factors.
November 97 System demonstration at Pittsburgh show.	The prototype system was fully demonstrated over a three day period as a workable tool to in excess of 20 industrial and academic participants at Pittsburgh Hydraulics and Pneumatics Show, November 11-13th 1997. During this event, the system commercial potential was clearly emphasised both for teaching and industrial applications.

As this chronological description shows, the modular development of the knowledge base, in terms of classes and rules, has allowed a gradual expansion as far as the system functionality is concerned. Not mentioned in this description is the development of a preliminary trouble-shooting tool and an oil selection facility. Both were generated separately, but are fully capable of integration with the main module. The next sections will explain how the concepts introduced in the previous chapters were harmoniously implemented in order to achieve the current functionality. Appendix 1 presents a graphical description of the prototype application through its several steps. Some results generated by the prototype are shown in appendix 2.

5.3- Prototype General Structure

As mentioned in chapter 3, the system core is developed in CLIPS. This comprises the knowledge base, i.e. rules and classes which handle the input from the user in a very generic form, i.e. without requiring knowledge about hydraulics. In the knowledge base, the definition of load, system, circuit and component objects is a key aspect to understand the system functionality. Figure 5.1 presents the system structure, defining in a general manner the knowledge base and the different agents in the prototype.

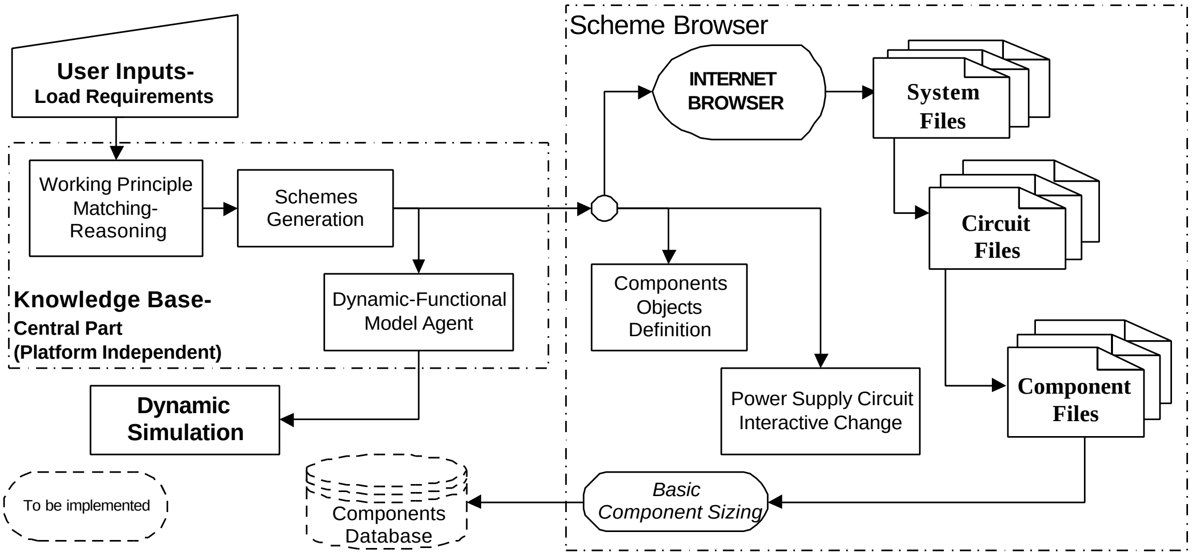


Figure 5.1- Description of the Prototype Modules.

Figure 5.1 presents the structure of the current stage of the Initial Prototype (IP). It can be seen that the knowledge base is platform independent, i.e. it is common to the different operating system versions. This aspect is one of the key points in the development, for it allowed to separate the parts only related to knowledge manipulation from those functions which deal with input and output operations.

Although still in its preliminary stages, compared to commercial packages, the IP has been gradually tested during its development. It runs under UNIX as well as WINDOWS95 and also includes the interfaces for the other modules, such as the Dynamic Modeler and Internet Browser. In this development, the agent-based paradigm is applied (TOMIYAMA et al.,94), in which each agent, an expert system, a computer program or an human expert module, interacts with the others to solve a complex task (HUANG et al.,93). The agents will be explained throughout this chapter.

5.3.1- Knowledge Base- The Classes

This section presents the main classes represented in the prototype system, with their functionality and some attributes. Despite the brief description, the knowledge representation structure can be readily expanded as shown in the next classes.

Class: load

Functionality: This class embraces the machine functions broken down according to its several defining criteria of actuation. It is domain independent, i.e. its definition is regardless of the energetic domain used to move the machine parts.

Main attributes:

- *Id*- it distinguishes each individual load, for a machine can have different loads with the same functional attributes.
- *Reaction*- defines the mechanical relationship between the load and its actuator.
- *Domain*- defines if the load is linear or rotational.
- *Control*- this features describes in a general form which function the load is designed to accomplish.
- *Position*- outlines the influence of the gravity in the load.
- *Description*- it allows the user to include a text to identify each load.

Class: Circuit

Functionality: It represents a working principle, this means a set of required sub-functions, defined through a specific set of means necessary to accomplish each load function.

Main attributes: The circuit class has all the main attributes of the load class. In addition, the circuit class has the following attributes:

- *Has components*: it represents a specific family of components (means) for a defined circuit. This attribute contains number, class and type (sub-class) of each component. It is vital for the process of component generation and sizing, as well as to define the circuit functionality.
- *Components-ids*: it stores the ids for the component objects that are generated for each circuit.
- *Model*: it provides an indicator corresponding to a dynamic model, which is defined in a library of working principle.
- *Circuit name:* labels a circuit according to its common denomination in the area, for example, bleed-off, meter-in, meter-out, etc..
- *Connected load*: this attribute defines to which load a circuit is “connected”, in other words, which load is generated by the circuit functionality. Each actuation circuit corresponds to one specific load, but the same load can be accomplished by different circuits in different systems. This feature allows the manipulation of alternatives for the whole design.
- *Sized*- this attribute defines if a circuit has been sized.

- *Description:* although this attribute is common to the load class, here it has a double functionality. First, it has a default value depending on the sub-class of circuit, which generically describes the circuit. Second, it is used to explain the reason why a specific circuit was selected for such load. Different from the description attribute in the former class, where it is totally specified by the user, here this feature has its value defined by the system.
- *Numerical attributes*: the circuit class also contains specific attributes related to size the hydraulic components, such as: cylinder area, stroke, displacement, flow, pressure, power, maximum dynamic and static efforts (torque or force), maximum speed, mass, inertia, etc..

It can be noticed that not all attributes are applied to all components, and not every circuit handles all the attributes. In fact, each type of attribute has a specific component related to it. For example, stroke, cylinder area and mass are directly related to linear circuits, while displacement and inertia are related to rotational ones. One can argue that the attribute definition should be divided among the component classes, and in fact that is **also** the case, however the inclusion of all attributes in the circuit class makes it easy to represent the properties in terms of Object Oriented Modelling (mainly it provides abstraction, see corresponding section in chapter 3), for it allows to handle the attributes in a higher level.

- *Ranking attributes*: in order to allow a comparative analysis between alternative circuits for the same load, the following attributes were defined: power efficiency, cost effectiveness, easy maintenance, easy operation and precision. These properties were selected based on an expert's advice, they have values defined as fuzzy sets. By no means is this set complete, rather its objective is to demonstrate the methodology for ranking among the alternative solutions. In order to facilitate the ranking process, each of those attributes has an equivalent numerical property.

The circuit class has several sub-classes with their own component lists. As mentioned before, the IP combines 29 types of actuation circuits and 5 types of power supply circuits. Although this aspect could be modelled using exactly the same number of sub-classes, this is not advantageous for in some cases the only difference among classes would be in a few attribute values. Therefore, in order to keep an elegant and manageable knowledge representation only a small set of circuit types was defined. Table 5.1 presents some values for the *has components* slot for some circuit sub-classes, the table shows only few circuits that are implemented.

Table 5.1. Default values of *has components* slot for circuit sub-classes

Circuit Sub-class	Quantity	Component Class	Type
Pressure-control	1 2 1	directional-valve pressure-control-valve cylinder	four-way reducing double-action
Flow-Control	1 2 1	directional-valve flow-control-valve cylinder	four-way non-compensated double-action
Flow-Supply	1 1	directional-valve cylinder	four-way double-action
Power-Supply	1 1 1 1 1	rotational-transformer pressure-control-valve filter reservoir motor	piston-pump relief low-pressure small-tank combustion-engine

Although the values in table 5.1 are set as default, they can be manipulated by the system according to rules and messages necessary for each application. Examples of such manipulation will be given in this chapter. In the above table, type represents a component specific attribute, the rationale for implementing this attribute is explained later.

Class: System

Functionality: It represents a scheme to solve the design specification defined according to the load set. It includes all actuation circuits and the power supply circuit. In fact, although this class embraces the whole design, it is less complex than the previous one, for the system object functionality is broken down in its corresponding circuit objects.

Main attributes:

- *System_id*: depending on the combination of load set attributes, the prototype can generate alternatives for the design. Therefore, it is important to distinguish among them. This function is performed by this attribute, for it specifies each system as an unique entity.

- *Has circuits*: similarly to the relationship between circuits and components, mentioned in the previous class, there is an assembly relation between system and circuit objects. This feature is defined by this attribute.

- *Operational sets*: in order to size a hydraulic system, it is necessary to represent how the actuation circuits relate to each other. For in case they are required to operate simultaneously, this should be considered for sizing the whole power supply circuit. Hence, the definition of operational sets provides this facility.

- *Rank*: each hydraulic system has singular properties. The prototype offers an option to rank the alternative solutions according to general criteria. Thus, after this process, each system receives a ranking value that is stored in this slot.

Class: Component

Functionality: this class involves the generic properties of hydraulic components. It defines a physical means required to accomplish a specific function. Due to the great variety existing among hydraulic components, this class contains only few attributes, because in order to keep a more manageable knowledge representation it is convenient to represent the more specific attributes in their corresponding sub-classes rather than define all features in only the main class. For each component family, represented as a sub class, has its own features.

Main attributes:

- *Component_id*: similar to the previous classes, each component object receives an identifier which is stored in this slot.
- *Kind*: This attribute was created in order to represent a specific property, without having to refer to a sub class, for example, component: pump, kind: piston-pump. This eases the manipulation of the component definition in a higher level in terms of detail.
- *Model*: it refers to a symbolic representation of the component.
- *Description*: standard explanation about the general functionality of a specific component. It enhances the component presentation for the user.
- *Sized*: this attribute sets if a specific component was sized.

As mentioned before, the above classes and their attributes embrace the great part of the system functionality. However, it should be noticed that here only the main attributes were discussed. In addition, the present knowledge representation structure allows the inclusion of other classes and other attributes for the above classes. Providing that the knowledge required to manipulate the new entities (attributes) is clearly and explicitly defined, the knowledge base can be enhanced. Further points on expansion will be discussed later in this chapter and in chapter 7.

5.3.2- Knowledge Base- The Rules

Together with classes, rules define the knowledge base core, for they represent the reasoning process from the load set definitions to the system generation. This section explores the structure of the rules, the type of control matching process applied, the integration between rules, functions and message-handlers.

An expert system is not a procedural program, for the distinguishing feature of the procedural paradigm is that the programmer must specify exactly *how* a problem solution must be coded. On the other hand, the goal of non-procedural programming is to have the programmer specify *what* the goal is and let the system determine how to accomplish it. Because of this feature, one of the key points in building an expert system is to define how the information flow control will be coded in the system, i.e. how the system will accomplish its goals. Among the different methods for this purpose, the developing system applies the **Control Pattern** approach for rules matching (GIARRATANO & RILEY,94). This approach has the following advantages: it allows the control knowledge to be separated from the domain knowledge, for splitting these types of knowledge makes maintenance and development easier; it provides better control on the matching of each rule or rules block. In this context, control knowledge specifies the technique for controlling rules execution.

Another powerful alternative approach is the **Modular Control**, in which each module of the system has it own agenda, the definition of agenda is given in chapter 3. Although this approach is more adequate for large knowledge base than the **Control Pattern** approach, providing that the size of the agenda is kept manageable, this means few rules are satisfied in a specific time, it is also possible to handle a large knowledge base with the control pattern method.

In the developing system, a central control rule was defined to handle the control in terms of the sequence of firing the rules. The central control rule determines the control facts which are used to control the rule matching process. It controls which block of rules whose patterns are satisfied by the active facts or objects in working memory, in other words which rules will be listed in the agenda. Figure 5.2 presents the central control rule surrounded by blocks of rules defined here according to their control facts.

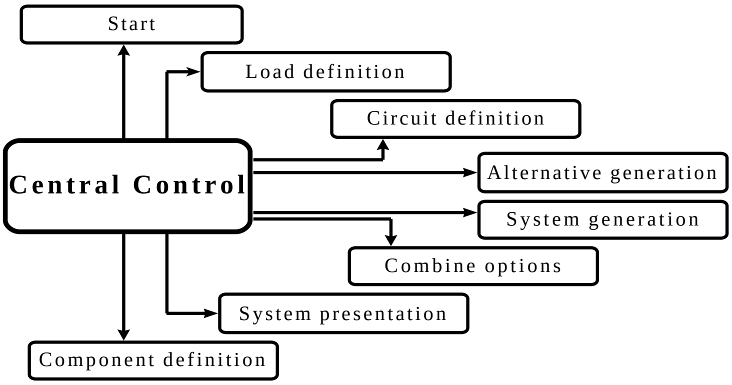


Figure 5.2- Central control rule and blocks of rules.

As figure 5.2 depicts, the rule blocks do not communicate directly among each other, rather once the rules in each block are fired, the inference engine (a built-in facility in every shell tool, as defined in chapter 3) fires the control rule which defines the next block of rules to be fired. The structure of the central control is represented in figure 5.3. In order to avoid the prefix notation common in CLIPS syntax, a more informal schematic representation is applied to describe the rule structure.

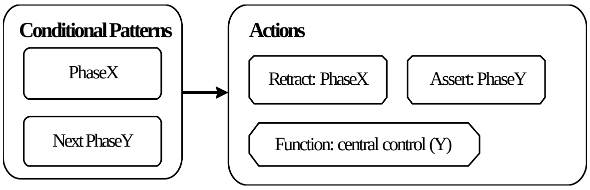


Figure 5.3. Central Control Rule Structure.

As can be seen, the central control rule has two conditional patterns (in this case, facts that are required to fire this rule) that are (PhaseX) and (Next PhaseY). The first pattern refers to the current phase in terms of the information flow (for instance: start, load definition, circuit definition, etc.) while the second pattern represents the next phase in the defined sequence. In addition to this structure, there is a set of predefined facts that determines the first fact (Phase Start) as well as a phase sequence, defined on figure 5.2. A typical fact defining this sequence is for example: (phase-after start loads-definition).

The central control rule processes three actions, the first two functions are predefined CLIPS functions (GIARRATANO & RILEY,94) and common to the majority of shell tools.

1. Retract: PhaseX - it removes PhaseX from the fact list.
2. Assert: PhaseY - it adds PhaseY to the fact list.

3. Central control Y- it processes some actions required before the next phase. For instance, it deletes all load instances (created in the previous session) before defining a new load set. It presents some information to the user about the next phase to be processed, etc. This function is platform dependent, that means two versions were implemented, one for the standard text-oriented input, another for the Graphical User Interface version. By keeping the same functionality in the two versions and defining linking functions the system in terms of knowledge manipulation is platform independent.

Although an expert system is a non-procedural program, in every expert system there are some procedures for solving a problem, therefore a sequence in terms of steps should be specified, shown in figure 5.2. In the present context, this sequence was very well defined and hence implemented. The information flow is defined according to the following blocks of rules:

Start- it allows the user to enter personal information (name, company, email, etc.) as well as define a directory where the resulting files from each session (one whole design) are stored. Basically in this rule, only the designer object is manipulated.

Load definition- this block of rules specifies the qualitative load attributes, generates the **load objects** and if there is a contradiction in some load definition, it redefines some of its attributes according to some specific knowledge. For instance, in the present version, the system corrects the input of the load definition if the reaction was defined to be positive (i.e. the actuator always drives the load) but at the same time the load is set up as vertical. Having done that, the system adds to the load description defined by the user a note regarding the change. Another example of contradiction refers to defining a torque-rpm combined control for a linear circuit (or vice versa), in this case the system also processes a similar change. Although these examples are very simple, the importance here is rather to demonstrate that a more complex reasoning can be modelled and also to emphasise the explanation facility.

Circuit definition- once the **load objects** are specified, the next block of rules corresponds to the definition of circuits. This block is the knowledge base core for it is in these corresponding rules that the **circuit objects** are generated. Presently, this block contains fourteen rules which deal with knowledge manipulation to define all the required circuit objects with their qualitative attributes. The number of rules presented here is not a fixed value, and does not represent any specific point, rather it only reflects the comprehensiveness of this block compared to the others. In other to exemplify this block of rules, the next figures present two of its rules.

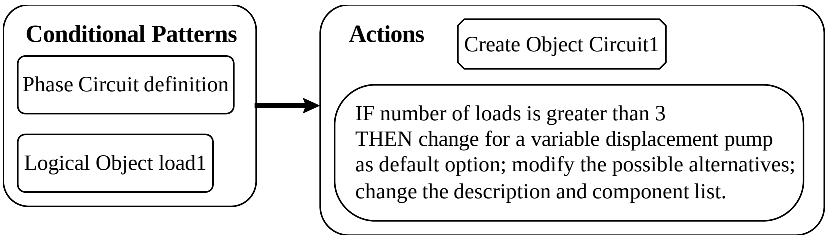


Figure 5.4. Create power supply circuit rule.

As depicted on figure 5.4, the rule for creating the power supply circuit has two conditional patterns. The first specifies the corresponding block, or phase in terms of information flow. The second pattern creates a logical dependence between circuit1 and load1 objects. Although this pattern is redundant, for the rule would fire exactly once only with the first pattern, the logical dependence guarantees the consistency of the system in terms of truth maintenance. This means that once the load1 object is deleted (before starting the next session, for example) the circuit1 object is automatically deleted. This rule has two actions:

1. It creates the Circuit1 object, with default values for component list, description, model, possible alternatives, and so on. Those values are defined in the class structure.

2. Depending on the number of loads, the rule changes the attributes in order to emphasise the importance of a more efficient choice.

The load1 object was chosen because it is the minimum option, i.e. a hydraulic system has at least one load. There is a common structure among the remaining rules of circuit definition block, therefore here only one more rule of this block is presented.

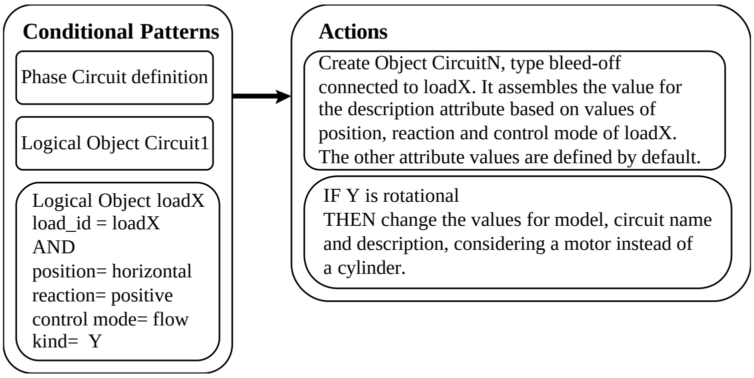


Figure 5.5. Create a bleed-off circuit rule structure.

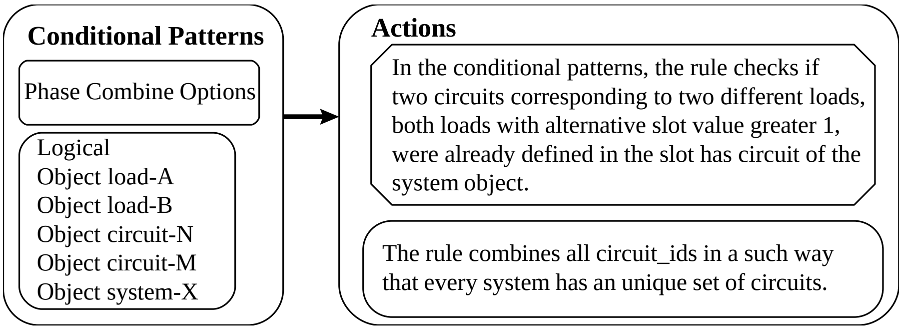
The first two patterns of this rule have a similar function to the patterns in the previous rule. However, here the logical dependence is between the new created circuit and circuit1 objects. As there is a dependence between load1 and circuit1 objects, the logical dependence in relation to load1 is also maintained. The circuit ID is defined based on a counter implemented as global variable. There are similar rules for other circuits.

From the description of the previous rules, it is clear that they could be decomposed into more rules, this means one for creating the circuit objects and another only for changing its attributes depending on the load attributes. Both approaches have the same outcome, in fact the existence of this type of “nested rules” is quite common in the developing system, because of the facilities CLIPS provides to implement such approach. Hence, the common practice of measuring the complexity and comprehensiveness of an expert system **only** based on its number of rules does not reflect the whole knowledge embedded on the system. Furthermore, this issue becomes more complicated when using the power of class structure and messages handling, the last will be discussed in more detail later.

Alternative generation- at this phase, the **load** and **circuit objects** are created, however no definition in terms of alternative solutions is made. Therefore, before combining the circuit objects to provide alternative hydraulic systems, it is necessary to define which loads (if any) have more than one possible working principle. Hence this block of rules, whose output is to increment the slot **alternative** of each load according to how many circuits are “connected” to it and include the corresponding circuit_id in the slot **options**.

System generation- once the alternative slot is defined, the **system objects** are created, initially with only the power supply circuit.

Combine Options- in terms of conditional pattern, this rule has the greatest complexity for it involves the load, circuit and system objects to combine the alternative solutions for each load. The rule is responsible for functionally building the hydraulic systems combining the options for each load, in such a way that for every load in each system there exists a specific circuit object, and every system has an unique set of circuits which accomplishes the whole design problem. Figure 5.6 presents a simplified description of this rule, emphasising its conditional patterns rather than describing all the action details.



In the conditional patterns, the rule checks if two circuits corresponding to two different loads, both loads with alternative slot value greater 1, were already defined in the slot has circuit of the system object.

The rule combines all circuit_ids in a such way that every system has an unique set of circuits.

Figure 5.6- Simplified structure of rule Combine Options.

From figure 5.6 it is important to mention that the rule needs a system object to be fired. Hence, this explains why the system objects were created before, even as an incomplete form. Obviously, there must be other forms of achieving the combination, the purpose here is only to describe how the prototype was developed, and therefore by no means is the rule unique nor is it optimal. However, it does provide the satisfactory output.

After the combination of alternatives has taken place, the expert system includes in all hydraulic system objects the ID’s of those circuits connected to loads which have only one option.

System presentation- at this phase, all system and circuit objects were fully defined, i.e. their qualitative attributes necessary to functionally achieve the machine specification have been determined. However, the resulting knowledge in terms of alternative solutions is distributed throughout the objects.

5.3.3- Knowledge Base- Messages

Despite of those above mentioned objects were created, no readable output file (in any format) was generated. Therefore, it is necessary to activate different processes to create these files, based on several formats, and handle the information distributed in the objects according to the system requirements. Also it is important to define the list of components which are the most basic functional units to be established. In order to execute these tasks, that basically are done by handling of object properties, the choice was to use message passing or methods. Object and message passing also provide a way to specify concurrent, asynchronous operations (WATERMAN,86). By definition, methods manipulate the instance objects to provide or derive specific data used in reasoning about the objects (GONZALEZ & DANKEL,93).

Similarly to the properties defined in the classes, the methods defined in a higher class are also inherited by its sub-classes. This feature makes it easier to manipulate objects of different sub-classes with a very generic message sent to the higher class. Moreover, if the same message is redefined, by a new set of procedures, in a sub-class, this new set takes precedence over the procedures described in the higher class. For instance, each hydraulic component type has its own set of equations and guidelines that should be used to size it, therefore these procedures should be specified for each sub-class, e.g. pump, motor, directional valve. However, the **size** message to trigger the procedures can be also specified in the highest class, i.e. the component class. Therefore, when a component object receives the **size** message, it will use specific procedures according to its sub-class. With this approach multiple procedures can be triggered using an unique message. This property is called polymorphism, defined in chapter 3, and it was largely applied in the prototype.

To illustrate the application of polymorphism in a very important part of the prototype, the next figure shows how messages are applied to create the component lists and how the entire process is triggered by one instruction.

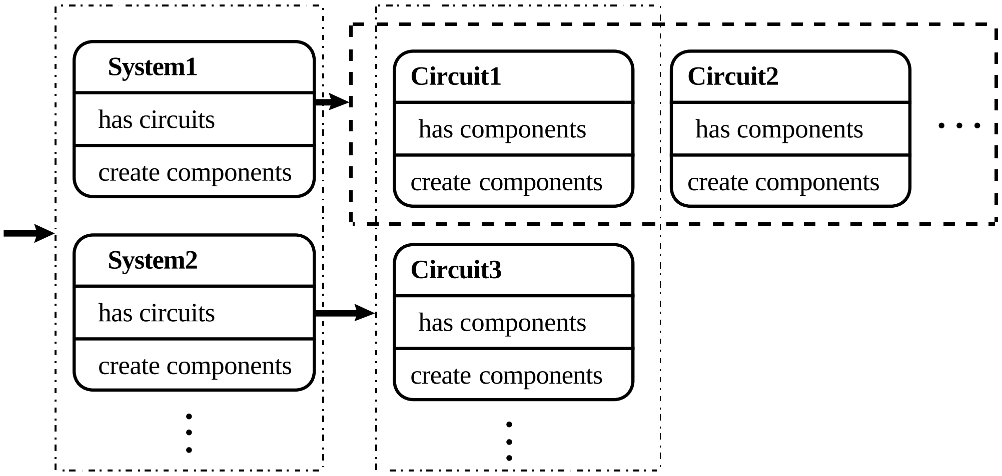


Figure 5.7- Connection diagram to create components.

On figure 5.7 the arrows represent the passing of the same message (create components), which is handled by different objects in a particular form. As the figure depicts, the whole set of system objects receives the referred message. As their attributes values were already defined, by the previous rules, each system object has a specific set of circuits related to it. Thus, each system object passes the same message to its circuits, for simplicity only two systems and three circuits are described here but the process itself has no such limitation. By definition, only circuit1 object is common to all system objects. Having received the message create components, each circuit object checks the number, class and sub-class of its components specified in the *has components* slot. According to this data the circuit objects activate a function to make their specific lists of components. Once a component is created, its ID is placed in the slot *components-ids* for each circuit. This slot is used to avoid sending the message again to a circuit object whose component list has already been created.

From this process, it can be seen that the flow of information, which involves a large amount of knowledge manipulation, is activated by only one instruction, i.e. message passing to the system objects. Similar approaches were developed to create the agents that correspond to generate HTML files, define functional dynamic models, size circuits and rank systems. These agents are described below.

5.4- Generation of HTML files as output

As explained in the beginning of this chapter, although the specific use of Internet technology was not explicitly defined since the project start-up, this application became a major area in terms of development in the prototype system.

The rate of the web's growth has been and continues to be exponential, but it is slowing in it's rate of growth. For the second half of 1993, the Web had a doubling period of under 3 months, and by January 97 the doubling period was still under 6 months. Only to show in a very simplified form the growth of this technology, table 5.2 presents some statistics about the rapid growth of the WEB, mainly the increasing number of commercial sites connected to it ^{2}.

Table 5.2. Summary of WEB growth

Month	**# of Web sites**	**% .com sites**
6/93	130	1.5
12/93	623	4.6
6/94	2,738	13.5
12/94	10,022	18.3
6/95	23,500	31.3
1/96	100,000	50.0
6/96	230,000 (est.)	68.0
1/97	650,000 (est.)	62.6

The increasing number of commercial sites has allowed the development of applications which can speed up the design process. In this new environment, tasks such as suppliers comparison can be done in a much shorter time, therefore contributing to shorten the time-to-market.

Based on the above information and considering the facilities provided by the Internet, the potential benefits for virtually all areas become obvious. Next, some of these benefits are presented along with their potential application for the design area:

- Servicing Customers- In this aspect, a design system connected to the WEB would facilitate the interaction with the customers, which is one of the key characteristics to be taken into account in Concurrent Engineering projects;
- Updating information regularly- With the use of more efficient communication systems, the updating information need, inherent in almost all design activities, would be possible;
- Using the Internet within their own wide area (Intranet)- As more companies are been connected to the WEB, the communication among the several departments (design, manufacturing, suppliers, etc.) becomes easier. Thus this technology provides a platform to implement a better team work;
- Looking forward to future markets- Although the present reality in terms of computational systems is far from establishing a complete virtual environment, with the improvement of satellite communications, there is no doubt about this trend. Therefore, the earlier the Internet technology is explored for design application the greater the benefits will be.

In addition to these benefits, the World Wide Web with its HTML language has quickly become a standard means for hypertext document delivery (TANSKANEN,97). Because of numerous advantages, Web tools are soon expected to be found on each engineer's desktop. Even today, some speak of 85% of the workplace computers as already connected to the Internet and using some of the Web services. The spreading of Web technology had also the interesting side-effect that many programs without an user-interface have emerged, which rely wholly on a web browser for interaction. Web tools run on all major platforms and have a high degree of compatibility with all kinds of applications. Web technology is widely known and standardised and offers good communication performance. The resulting tools have already gained the acceptance even of engineers that belong to the "late adopters" of computing and network tools (DRISIS,97).

---

[2] Credit to "Matthew Gray of the Massachussets Institute of Technology" (http://www.mit.edu/

Following this trend, and as a consequence of an intense interaction with an US expert user since early phases of this development, the expert system adopted HTML as its main means of communication, in terms of presenting outputs generated from the knowledge base, i.e. system, circuit, and component objects in an user-friendly form. In this context, the prototype system automatically creates a set of HTML files, which present all objects with textual and graphical information. Each object has its file, which is linked to files of related objects. Hence, with this facility the user can quickly and naturally navigate through the whole set of options in terms of design. Figure 5.8 depicts in a simplified form the module structure defined to create the HTML files (SILVA & DAWSON, 97c).

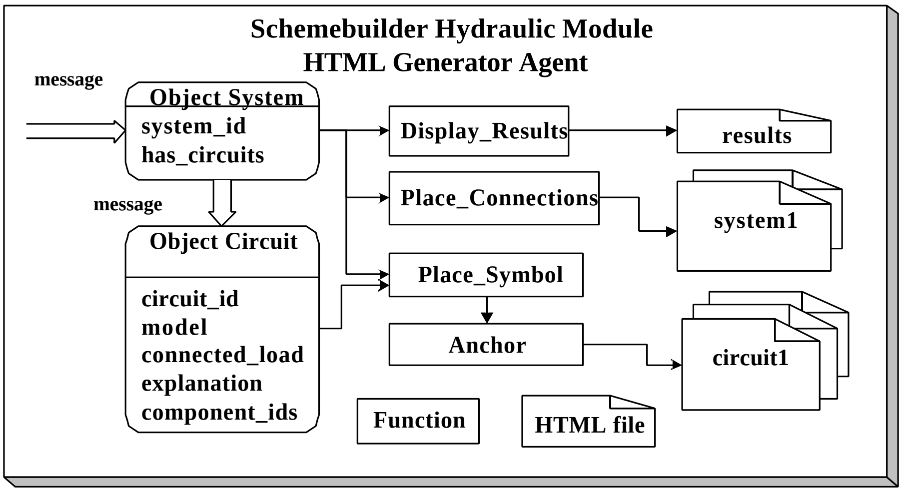


Figure 5.8- HTML Generator Agent structure.

Similarly as described above in the process of creating components, the process of created HTML files is based on the application of messages. It also uses the linking attributes (e.g.: *connected load*, *has circuits* and *has components*) to define hyper-links among the several files automatically generated. This important aspect is accomplished by the function anchor, which writes in HTML format the specification of the link and its label.

As depicted on figure 5.8, the agent generates a set of HTML files, indexed according to the circuit and system ID's. The structure of the files allows the user to:

- Browse among the several alternative hydraulic systems that were generated depending on the load specifications;

---

people/mkgray/net/). This link was accessed in Dec 97.

- Identify each load with a specific circuit (sub-set of the whole systems). This includes an explanation about the circuit and its application;
- Check the list of components for each circuit;
- Compare between the available circuits to accomplish each load;
- Verify the system objects order according to the set of degree of importance for the general comparative attributes defined in the circuit class.
- Study alternatives for power supply circuit and dynamically modify the default option. Once the change is processed, the expert system redefines all component lists, and the files for system, circuit and components.
- Check the general information about the design, including the load set specification.
- Access through the WEB relevant references used to develop the expert system, and potentially this allows the user to contact supplier companies to send the HTML files or their addresses to estimate the cost of the proposed hydraulic systems.

Although the expert system, in the present version, allows only output for the Internet format, there is a plan to develop a full Internet version. Since the prototype was developed with a clear separation between the knowledge base and the input/output functions, it is expected that a full version for Internet is feasible, further comments on this issue will be given later. Another relevant area considered in this prototype is the dynamic simulation. The next section presents how this point was handled in this development.

5.5- Generation of Functional-Dynamic Models

As mentioned before, in the present context, a computational agent is defined by a set of functions, that can be implemented through several forms tailored for specific tasks. In this section the dynamic modelling agent will be discussed, its task is to format outputs from an expert system according to a specific simulation language package. In the development of this agent, the knowledge engineer interacted with an expert who has experience in control engineering and in this specific language application [3].

---

[3] Dr. David Bennett is currently a Research Associate at EDC Lancaster University in the area of control engineering.

Similarly to other points already described, here the computational agent also consists of a set of messages, relating the specific hydraulic circuits to create a file which defines the set of classes for an Object-Oriented Modelling, (OOM) Language, e.g. DYMOLA (ELMQVIST et al.,93). Although the agent was defined specifically for DYMOLA, for it was the tool chosen previously chosen by EDC, the underlining methodology should be applicable for other OOM simulation systems. This language allows the modelling of complex dynamic systems using the Object Oriented Techniques already mentioned, e.g. inheritance, abstraction and polymorphism.

In DYMOLA, a complex system is broken down in submodels which contain the system dynamic equations. Next, this language is described in more details.

DYMOLA is an object-oriented modelling and simulation language for large mixed non-linear continuous and discrete systems (ELMQVIST et al.,93). DYMOLA terminology classifies a final scheme as a model, and individual components/sub-components as model classes. Models are hierarchically decomposed into submodels (instances of model classes) that allow reuse of modelling knowledge by the formation of libraries. A generic model class can be developed, defining a type of interface (cut) structure allowing inheritance by common components.

Model classes are connected via cuts which model the physical coupling. Therefore unlike Block Diagram Modelling (e.g. SIMULINK) ports, many variables can be passed through a DYMOLA cut, for instance modelling a double-ended cylinder, see Figure 5.9. When creating a model, cuts are not defined as inputs or outputs, causality is automatically defined during compilation. Cut definitions are based on bond graph terminology (KARNOPP et al.,90). Variables are segregated into two categories, effort and flow, or across and through. Some standard components have a variety of cuts that allow a combination of modelling domains to be connected, thus allowing multi-disciplinary design and simulations. Mixed domain modelling extends to the development of domain specific class-libraries which can be combined to form multi-disciplinary application models.

The underlying structure of the component models are a series of differential and algebraic equations. Discontinuities are represented by if….then statements, translated to discrete events as required by the numerical integration routines.

The CLIPS-OOM agent automatically generates the appropriate DYMOLA model files, declaring instances, connectivity and parameter values. Once read by DYMOLA the system of differential equations (defined by the model class libraries and connection statements) are symbolically converted to state-space format. A set of algorithms operates to produce a set of minimal system of equations which are solved simultaneously. Among other features, the formula manipulation techniques, implemented in DYMOLA, also automatically determine causality and solve algebraic loops.

The computational causality of a model determines how the physical laws that are encoded in the model equations must be interpreted in order to obtain a program that can be executed on a sequential machine using existing numerical algorithms. In other hand, an algebraic loop occurs when, for example, a variable x must be known before y can be computed, but y must be known to compute x from a separate equation in the same model. Algebraic loops among variables within a model sometimes mean bad modelling, or rather, a bad choice of variables. However, algebraic loops that are the result of interconnections between different objects occur frequently and are unavoidable. Based on Formula Manipulation Techniques, DYMOLA handles the algebraic loop and causality during the modelling phase, i.e. before the simulation stage (CELLIER & ELMQVIST,93).

The concept of cut is illustrated in figure 5.9, using a hydraulic cylinder. This component is modelled with four cuts (2 hydraulic and 2 mechanical cuts). The variables pressure and flow (p/q) are defined for the hydraulic cuts, while the mechanical cuts include position, velocity, acceleration and force (x, xd, xdd/f).

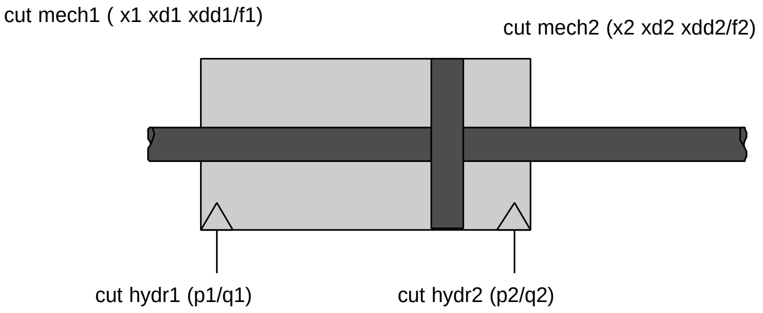


Figure 5.9. Example of DYMOLA cuts for a cylinder.

The same concept of cuts is applied to model the connections among different hydraulic circuits. Here, the well defined structure of hydraulic systems facilitates the modelling of connections. Because, in most hydraulic system applications, the actuation circuits, directed linked to the loads, are connected in parallel to the power supply unit.

At present, there exists a comprehensive hydraulic component library (BEATER,97) written in DYMOLA [4]. This library was freely offered to EDC to serve as a basis to develop the library of circuits, which are the submodels in DYMOLA terminology of a hydraulic system. The library of working principles has been partially developed, this means not all circuit models were already built. However, the present stage is sufficient to demonstrate the underlining methodology. The next figure depicts the agent structure.

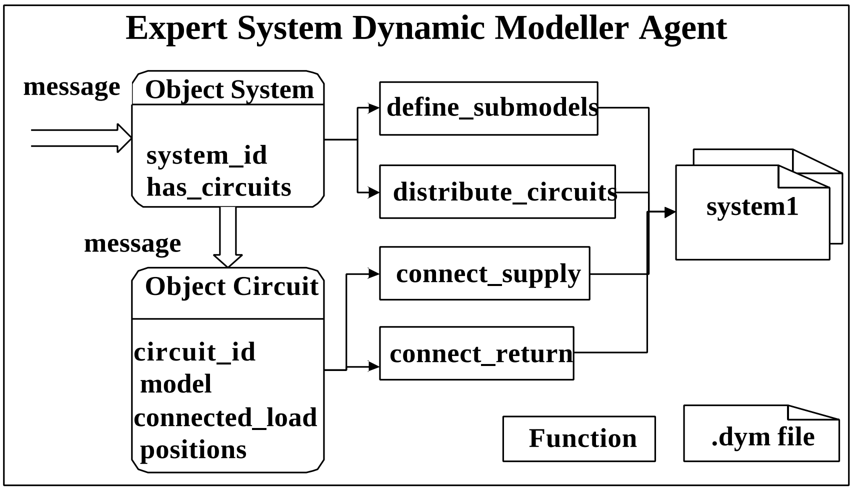


Figure 5.10. Functional Structure of the Computational Agent (SILVA et al.,97).

Figure 5.10 depicts the functional entities related to this computational agent. As mentioned, its objective is to generate a set of files, one for each generated hydraulic system object, in an OOM format. Similar to the previous processes, this procedure is triggered by a message sent to a **System Object** that identifies which are the **Circuit Objects** that belong to it, whose ID values were defined by the rules and are stored in the *has circuits* slot for each system. With this procedure, the agent defines the submodels and the connect statements required to run, compile and simulate a DYMOLA model.

In order to obtain a proper DYMOLA model, the agent handles the following circuit attributes:

- MODEL- it defines the type of circuit, according to a library of models in DYMOLA that must be loaded together with the complete system model;

---

[4] During the European Simulation Symposium, Passau-Germany, October 97, it was made public that this library would be available in the DYMOLA version 3.1.

- ID- it identifies the circuit instance, for a hydraulic system can be composed of some circuits of the same type, but with different properties such as flow, actuation effort, etc.;
- COORDINATES- it is used to position each circuit in a DYMODRAW window (the graphical front-end of DYMOLA).

The next figure presents one file automatically generated by the expert system, according to the DYMOLA format. It can be seen that this file presents the circuit types and Ids, as well as their statements necessary to construct a DYMOLA model.

```
model system1  {* (-100, -100) (100, 100)}
{* window 0.8 0.8 0.8 0.8 }
submodel(RPC)  circuit3 {* at (85,50) (135,110) }
submodel(VMO)  circuit2 {* at (-135,50) (-85,110) }
submodel(PS)  circuit1 {* at (-25,-130) (25,-70) }
connect circuit1:supply at circuit3:supply {* via (-5,-130) (-5,5) (105,5) (105,110)}
connect circuit1:return at circuit3:return {* via (5,-130) (5,-30) (115,-30) (115,110)}
connect circuit1:supply at circuit2:supply {* via (-5,-130) (-5,5) (-115,5) (-115,110)}
connect circuit1:return at circuit2:return {* via (5,-130) (5,-30) (-105,-30) (-105,110)}
{* using
using /home/../power.lib
using /home/../pressure.lib
using /home/../flow.lib
using /home/../supply.lib
using library/block/io.lib      }
{* ellipse (-6,6) (-4,4) fill_color=3 fill_pattern=1 }
{* ellipse (4,-29) (6,-31) fill_color=3 fill_pattern=1 }
end
```

Figure 5.11. An example of the agent output in DYMOLA format.

In figure 5.11, the terms RPC, VMO and PS stand for Rotational Pressure Control Circuit, Vertical Meter-Out Control Circuit and Power Supply Circuit, respectively. These attributes are selected through the knowledge base and define an OOM model for each corresponding circuit. The "submodel" and "connect" statements also contain the coordinates of each circuit and their connection ports.

The submodel declaration instantiates objects from classes, while the connect statement is used to describe the interconnection between the objects (CELLIER & ELMQVIST,93). For example, in the above file, circuits 2, 3 and 1 are instances of the classes VMO, RPC and PS respectively. Moreover, the connect statements define that the supply and return ports (cuts) of circuits 2 and 3 are physically linked to the corresponding circuit1 cuts. As can be seen, this procedure takes advantage of the fixed topology which is a characteristic of the majority of the hydraulic systems, i.e. actuation circuits connected in parallel to the power supply unit. According to this definition, by entering the previous model, DYMOLA assigns that the pressure at the supply port of the power supply circuit has equal value to the corresponding ports at the actuation circuits. Being the flow at circuit1 supply port a result of the sum of the flow for each actuation circuit. It considers that no friction takes place and neglects the pipe lines, though these effects can be included in a more sophisticated model. Note: if the circuits are sized, a parameter list for each circuit can be defined after each instance declaration. The parameters can also be manipulated directly in DYMOLA.

Upon entering a model, DYMOLA immediately instantiates all submodels (objects) from the model types (classes). It then extracts the formulated equations from these objects (defined in the model libraries), and expands them with the coupling equations that are being generated from the description of the interconnections between objects (CELLIER & ELMQVIST,93).

It is important to notice the difference between a top-level model, which essentially defines the topology of a system, and a low-level model that contains the detailed equations. Due to the complexity of low level models, it is essential that they can be reused (ELMQVIST,93). In the present context, the top-level models represent the hydraulic systems, while the low-level ones describe the several circuits.

Although the structure of the agent is already defined and its implementation (as part of the initial prototype) has taken place, the complete usefulness of this module depends on other factors: the validation of the Working Principle- Circuit library in DYMOLA; the dissemination of this simulation tool and the validation of the expert system as a whole.

Regarding the complete validation of the circuit library, this task is more related to the field of dynamic simulation, rather than to the area of knowledge engineering, the main topic of this project. Therefore, although dynamic simulation is of importance for hydraulics, this point will not be fully accomplished in the current project. However, as the knowledge engineer also has some experience in research of dynamic modelling and simulation of mechanical (SILVA,90) as well as hydraulic systems (LINSINGEN et al.,91; LINSINGEN et al. ,92), it was judged to be more important to emphasise the agent underlining methodology rather than the full validation of the model libraries. This decision was also based on the following points:

- Despite its features, which greatly facilitated this development, this modelling language has some drawbacks, such as: it is not widely spread (i.e. it has few users) compared to other simulation tools. This conclusion is entirely based on the author's observation throughout the project, during the contacts with engineers who were working in hydraulics, hence no comparative study among the different tools, in terms of usability, was made. This modelling language was not specifically developed for modelling fluid power systems (differing from some systems mentioned in chapter 4, tailored for hydraulics), therefore, more tests in this field may be needed;

- The task of validating the circuit library would also require a team of researchers with experience in this specific field, as well as laboratory facilities to verify the model parameters based on hardware tests. Neither of these two points was available in the project time scale, nor were they requested for this research;

- Due to the great variety of simulation packages for fluid power (see list in chapter 4) the most useful application of this methodology will be defined by the users' (industries and/or universities) chosen tools. For in some cases, they may already have validated libraries for which they may want a specific agent.

With the methodology described in this section, a great part of the modelling task in terms of equation definitions, parameter settings, etc. is executed by the developer of the model libraries, in this context working principle libraries. The expert system handles the model in a top-level approach, i.e. generating the system files. Hence, the user's task is facilitated for it will be more concentrated on performing the simulation and analysing the results.

Even considering the present stage, the methodology explained in this section has been capable of inducing more research in this area and attracting experts' attention to this approach.

5.6- Concurrent Engineering directly addressed in the Initial Prototype.

As pointed out in the previous chapters, the developing system has among other objectives to provide a better understanding and application of concurrent engineering concepts throughout the activity of hydraulic system design. Being this point one of the main criteria to support its potential use either in academic or industrial environments, it is paramount to describe what and how the prototype can provide for this feature. Hence, in this section, the concurrent engineering aspects directly addressed by the prototype will be discussed.

5.6.1- Power Supply Circuit Aspects

The heart of any hydraulic system is the power supply unit which transforms the mechanical input into fluid power and the machine used is a positive displacement pump which provides a flow proportional to the input speed. The pressure is dependent upon the external resistance of the circuit and the load on the actuator (GREEN,85). Due to its inherent importance, in the prototype, the power supply circuit deserved a special attention compared to the other circuits.

As mentioned in the previous sections, the object generated to model the power supply function "belongs" to all design alternatives. By definition, it has default values for its attributes, such as, component list, description and model. However, considered the great variety of possible circuits to accomplish this function (i.e. circuits with different types and numbers of pumps, with or without accumulators, etc.) and because of its relevance for the hydraulic system design, it is necessary to provide means to modify the default option offered by the expert system. This procedure is explained as follows.

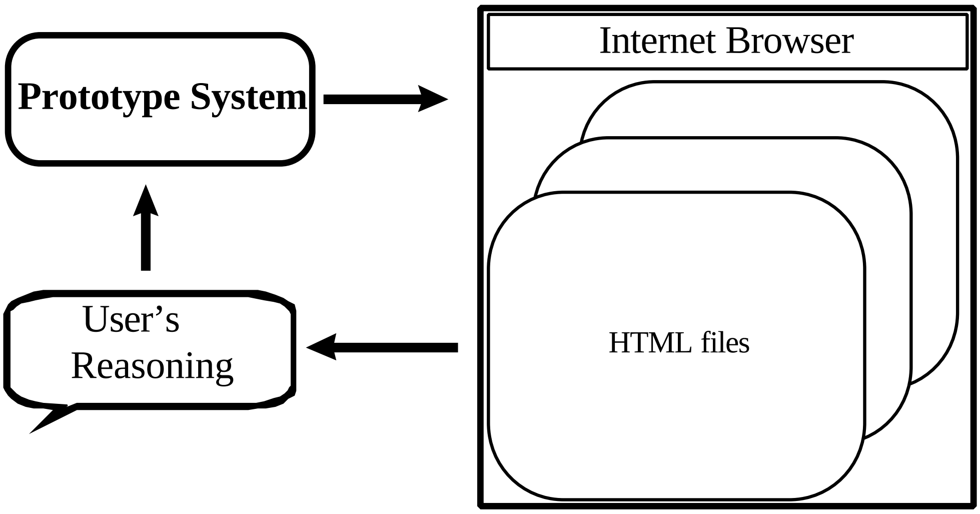


Figure 5.12- User, Prototype System and Internet Browser Interaction.

As figure 5.12 shows, there is a cycle in terms of interaction between the user, the prototype system and the Internet browser. As follows, the user specifies the load information to the prototype system, i.e. qualitative and quantitative attributes of the load set. Based on its knowledge base, the prototype reasons on the load attributes, instructs the user to launch the browser, which automatically displays all the HTML files regarding the results from the reasoning process. While navigating through the files, the user can browse all alternative solutions from the level of systems to the level of circuits (and if loads are sized, also down to some component details). Next, it is presented a typical HTML file for circuit1.

--circuit1.html file--

Circuit ID: CIRCUIT1:

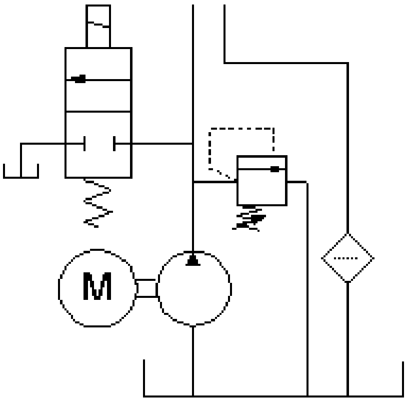


Figure 5.13 Power Supply Circuit (default option).

Circuit Description:

(Circuit1 has the POWER AND SAFETY FUNCTIONS, and it is ALWAYS created.).

It has the following components:

Table 5.3- Example of a component list generated by the prototype

Component	**Class**	**Type**
Component13	pump	piston_pump
Component14	pressure_control_valve	relief
Component15	directional_control_valve	unloading
Component16	filter	low_pressure
Component17	reservoir	small_tank
Component18	prime_motor	combustion_engine

Check the following alternatives for this circuit:

1.psv

2.ps_a

3.ps_a1

-------------------------------------end of file-------------------------------------

As can be seen, this file presents the information regarding the default values for the power supply circuit. The component-id numbers are resulted from the process of creating components, explained before, and not directly related to the circuit-id. The file also suggests that the user checks alternatives for this solution, these alternatives presented here as psv, ps_a and ps_a1 (see next figure) are hyper-links to predefined files that explain each alternative separately. Each of the files contains the graphical representation of the corresponding circuit as well as notes regarding general aspects, such as efficiency, cost, safety and so on. The options are also exposed in the comparative analysis file as described on next figure, being the first circuit the present choice.

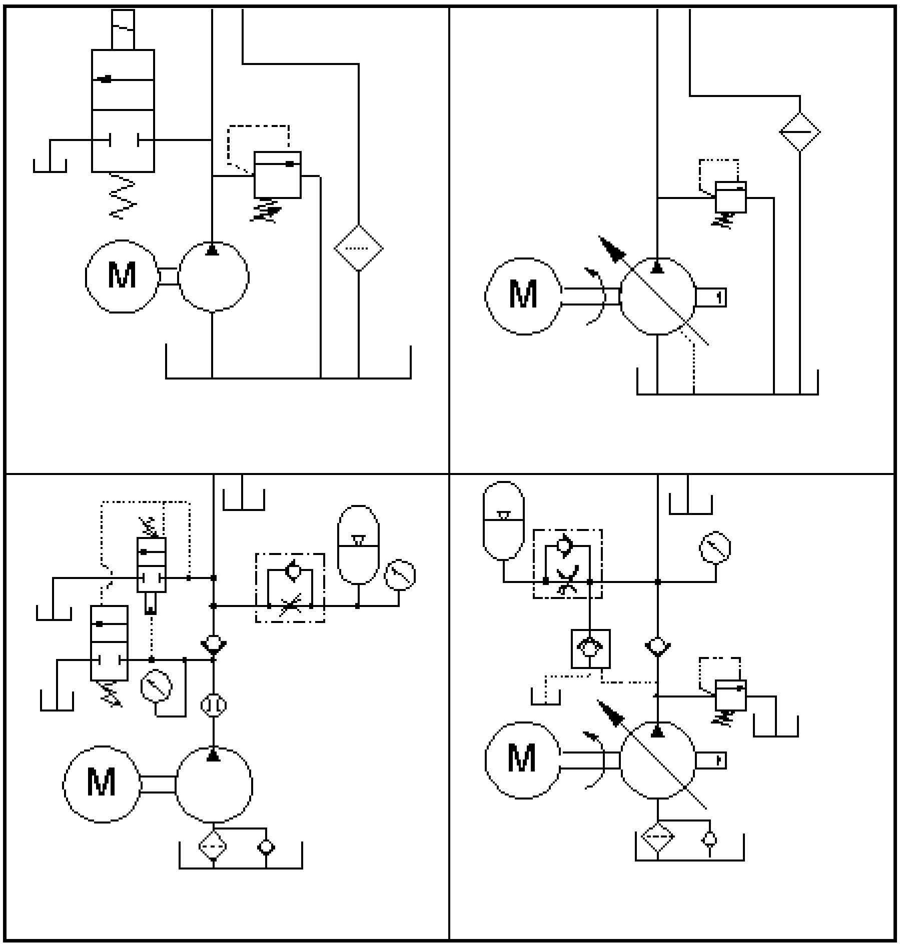


Figure 5.14. Power supply circuit options presented for comparison.

After studying the options, the user can change according to his/her decision on which option is more appropriate to that specific application. Once the choice is made, the expert system provides the facility to change the power supply circuit, automatically redefines all slots for the circuit1 object, as well as the entire component list for all systems, in other words, all component objects are recreated for the above alternatives differ also in the number of components for each one of them. By providing this facility, the prototype system empowers the user, even an experienced engineer, with a greater freedom of choice, at the conceptual design stage. Clearly, this approach brings to the design activity some concurrent engineering aspects, that are not too common in the field of hydraulics.

5.6.2- Weighting Tool

Other relevant aspect related to concurrent engineering addressed in the prototype is the capacity of providing to the user, even in a very basic form, one comparative analysis among the alternative hydraulic systems generated. In order to perform this task, the system offers a ranking scheme in which the user can input, in a friendly form, degrees of relative importance among five attributes. As mentioned before, in the definition of circuit class, these attributes are: power efficiency, cost effectiveness, easy maintenance, easy operation and precision. The types of attributes were defined based on an expert's advice. The chosen number is just for the sake of implementation, and in terms of methodology there is no limitation.

To implement such facility in the prototype system, the knowledge engineer submitted a comparative table, considering all the circuits presently modelled in the knowledge base to an expert [5], the circuits were grouped according to their functionality in a way that they were considered possible alternatives. The expert was requested to define score, using a fuzzy set (very high, high, medium, low, very low) for representing the values in each attribute. Table 5.4 presents that expert's choice regarding a comparison among possible alternatives, divided in the different sections of table.

Table 5.4- Results from an expert ranking comparing circuits

Circuit Type	**power-efficiency**	**cost-effectiveness**	**easy-maintenance**	**easy-operation**	**precision**
bleed-off	very low	very high	high	very high	very low
meter-in	medium	high	medium	high	medium

---

[5] In this case, the expert was David Dawson, EDC Deputy Director and project advisor.

meter-out	medium	high	medium	high	medium
Proportional	medium	low	low	medium	high
pressure- control single set-up	medium	high	high	high	low
pressure-control proportional	high	medium	medium	high	medium
Power Supply (PS) fixed displacement pump	low	high	high	high	low
PS Hydro. Transmission	high	medium	medium	medium	medium
PS variable disp. Pump	very high	medium	medium	high	high
PS fixed displacement pump and accumulator	medium	medium	medium	low	low
PS variable disp. Pump with accumulator	high	medium	medium	medium	medium

The table was an attempt to define metrics that could be used to support the analysis activity, i.e. the decision among the alternative solutions generated by the expert system. It is important that, in order to ease the ranking procedure, all definitions are in a single direction scale, that is the higher is better. In the present version, only the procedure comparing the actuation circuits was implemented, this relates to the first two sections of the table. The above values were implemented in the knowledge base for each circuit, every attribute had also its corresponding numerical value (1 to 5). This double scale, fuzzy and numerical values, had two functions, first the numerical scale allows an adequate ranking, second the fuzzy values allow a better explanation. Next figure shows a scheme relating the three elements in the ranking process.

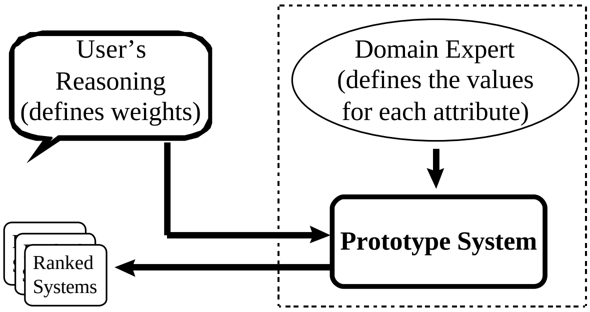


Figure 5.15- Ranking process.

Figure 5.15 demonstrates the relationship among the expert, the prototype system and the user in the context of the ranking process. As shown, the expert's knowledge appears embedded in the prototype system, this symbolic representation defines that in an user's session these two elements are considered as an unique entity (dashed line). However, they are in fact distinct elements, for the prototype can be implemented to model different expert's opinion, in other words different values for the attributes. In the ranking process, the user, internally represented by the designer object, defines weighting values for each of the five previously mentioned characteristics. These aspects are generic, i.e. they are primarily related to the design as a whole rather than to energetic domain applied to solve the problem. Therefore, the user, who is supposed to have general understanding in terms of the machine requirements, should be able to balance in a comparative scale the degrees of importance of all attributes. Once the weights are defined by the user as attributes of the designer object, the prototype executes a weighting sum for each hydraulic system, considering only those circuits related to loads with more than one option. This process sets a rank for each system object. After this definition, the prototype creates a ranking array, with all system-ids whose ranks are in a decreasing order. This is another designer's attribute which is displayed as a guideline to the user in a HTML file.

Although the above presented methodology is quite simple, it does present the opportunity to take into account a comparison of alternatives based on general attributes. By no means is it an absolute comparison, for different experts and users can provide different values for the attributes and weights respectively. However, this only reflects the inherent comprehensiveness of the design task, mainly when considered in a concurrent engineering perspective.

Once implemented, this facility was submitted to experts' opinions, through free tests and demonstrations, and so far it has received a general approval. More aspects related to this comment will be discussed in chapter 6.

Despite its empirical development, i.e. based only on the knowledge engineer's background and experts' inputs, this facility does agree with trends which present the importance of performing evaluation at the early stages of the design process (CZIULIK & DRISCOLL,97). Moreover, the adequate prototype expandability, discussed in chapter 6, allows further enhancements in the evaluation tool.

5.6.3- Size circuits and components

As the previous chapters emphasised, in a concurrent engineering context, the time-to-market is as important as the product quality. Therefore, one of the primary functions of the prototype system is to provide a facility to speed up the design process. In the hydraulic system design area, a great amount of time is spent in performing calculations for sizing the several parts on the system, e.g. components, pipes, etc.. Thus, if those calculations were managed by a computational tool, this would strongly benefit the whole design activity. Hence, this section describes how the process of sizing circuits and components, albeit available in a very basic form, was implemented.

The process of sizing makes a direct use of the object oriented techniques presented throughout this work. Here, the property of abstraction was a key aspect to implement the process, for the overall sizing of different components, each type with its specific set of equations and guidelines, is abstracted in a higher level, i.e. in the circuit class. The basic principle is that if a circuit representing a sub-function of the system is preliminarily sized, this means the ideal numerical values for its general attributes are defined even without choosing its real components, it is possible to “send” those values to its corresponding components in a later stage. In order to model such process, another important property applied was polymorphism, for different classes were modelled to handle the same message with different procedures.

The first step to size a hydraulic system is to define the supply pressure set-up, for this value directly affects the dimensions of components and pipes, as well as other aspects of the system, such as filter life and maintenance degree.

As pointed out in chapter 4, this issue was considered very carefully in the implementation of the prototype system. Different knowledge sources, i.e. technical literature and experts, were examined. Although there exists some common ground, a true consensus among the sources was not achieved. Therefore, the approach implemented here takes into account all information gathered from the sources, displays it to the user, offers an available range for the value (between 100 and 315 bar) and transfers the decision to the user.

For some analysts, a system which “transfers” the decision to the user cannot be considered an expert system. However, based on the complexity of the design task, on the profile of the potential user (a design engineer, who usually wants to have some degree of control in terms of decision making) and on the great diversity of areas for hydraulic system (e.g. from agricultural to industrial applications), the approach adopted in this project is quite satisfactory. Furthermore, even a human expert when submitted to this task may offer a range of design options depending on the supply pressure. Thus, with this approach the expert system is just emulating a human expert, which is exactly what it is supposed to do, based on its definition in chapter 3.

The sizing process starts converting all loads, if they were defined according to the Imperial System to SI, this is defined for each load. As mentioned before, to size a hydraulic system, which includes its power supply unit, it is important to understand how the loads interact with each other. Therefore, the system requests the user to define a set of operational sets, this set represents all combinations of load sets defined by the user. For instance, the set ("load1 load2" "load2 load3" "load1 load4") determines that there are three operational sets to be considered when sizing the power supply. In other words, the power supply sizing must take into consideration the requirements, in terms of flow demand, of (load1 & load2) and (load2 & load3) and (load1 & load4) separately. Besides this, the loads are also considered individually, a load (e.g. load5) may require a greater flow than all the previous combinations.

Once the loads and sets are determined, the prototype sends a specific message to a proper set of circuits, depending on the load type to which they are connected, as depicted in figure 5.16.

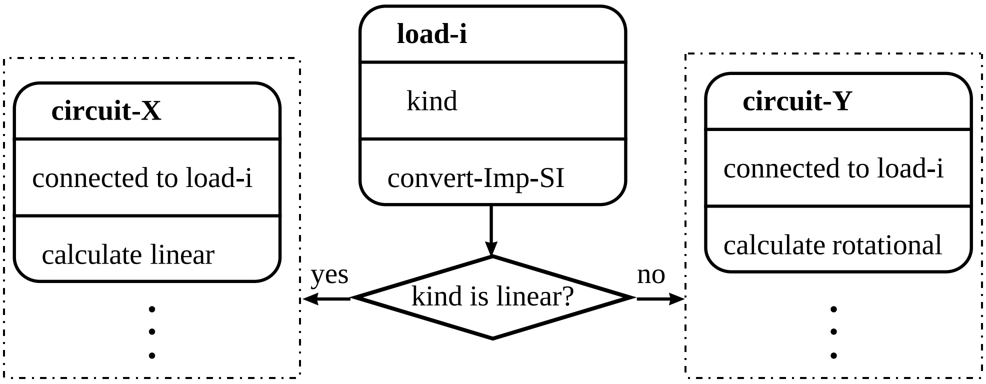


Figure 5.16- Relationship between load and circuits in the sizing process.

Although the relationship shown on figure 5.16 is very simple, the aspect here is to emphasise that several circuit objects receive the *size* message from the same load object. Despite of analysing only the *kind* attribute to set the message, a more complex process reasoning with more attributes can be modelled, providing that this knowledge is available and based on the load attributes that are also modelled. Basically, sizing here means to define the main parameter values for the actuator, the valves and pipe sizes for each circuit. Next, the process of sizing a linear circuit is explained.

All actuation circuits must be able to withstand the static and dynamic loads. In the case of translational circuits, due to the area ratio, the sizing process must also considered the values in both directions. Although there are some cylinders with equal areas, here in order to simplify the process, a default area ratio of 0.5 was used. The prototype applies the definition of load pressure to establish an optimum operational condition (MARTIN,95). The equations shown in this section are implemented for all linear circuits, however, in a detailed design they may require some modifications, for the pressure drop through the directional valves depends on each manufacturer. This enhancement is feasible in the current computational structure.

$$p_l = \frac{2}{3} \cdot p_s \tag{5.1}$$

Where: p_l- load pressure

p_s- supply pressure

Based on this approach, an usual design practice is to establish the maximum actuator load including friction, then determine a piston area based on the load pressure criterion. The remaining part of the supply pressure is then available for designing the valve. The approach considers that the seal friction is about 15% of the stall force. Therefore, the area equations are:

$$area_{d1,2} = \frac{Fd_{1,2}}{(p_l - 0.15 p_s)} \tag{5.2}$$

$$area_{S1,2} = \frac{Fs_{1,2}}{p_s} \tag{5.3}$$

Where: $area_{d1,2}$; $area_{S1,2}$ represent the piston areas (cap and rod ends), calculated based on the dynamic and static forces respectively.

$Fd_{1,2}$ and $Fs_{1,2}$ are the dynamic and static forces in both directions.

The previous equations generate four values. They are compared one to each other, considered the area ratio of 0.5, and the highest value defines the ideal area for the cylinder. Although this is typically a component attribute, as discussed before in the definition of circuit class, here it is modelled as a circuit attribute. Once the area is defined, then the system determines the flow based on the required speed. The ideal situation would be to define two attributes for speed, extending and retracting. However, the system assumes that the maximum speed is required in the forward direction.

$$q = area.v \tag{5.4}$$

The system also provides a figure for the natural frequency of the specific circuit, based on the following equation.

$$w = \sqrt{\frac{4b.A^2}{M.V}} \tag{5.5a}$$

Where: β represents the oil bulk modulus, typically defined as 15×10^8 Pa;

M- represents the load mass;

A- cylinder area (previously defined);

V- the volume in the cylinder chambers, V= stroke x area.

Although the calculation presented here is very simple, the principle is to demonstrate that all those attributes are handled in the circuit object level. Once these values are determined, the system sets the value for the *sized* slot accordingly. This process is repeated for all actuation circuits (there exist similar equations for rotational circuits).

After sizing all actuation circuits, the system calculates the total flow required, considering the operational sets, previously defined, and the individual loads. Thus the basic parameters, pressure and flow, for the pump unit are also determined.

Once all circuits are sized, the *size* message is sent to those component objects that “belong” to a specific circuit, but only for the objects whose *size* message was also defined. The message flow through different objects used here is similar to the approach applied to create components as described on section 5.3.3.

For the purpose of implementation, each circuit has at least one component class, whose sizing procedure has been implemented. This principle keeps uniform growth of the system, this means more component sizing procedures can be added without a great change in the system code. Due to great deal of detail necessary for sizing each different type of component, it was not possible to implement the procedures for all of them. However, as emphasised in other parts of this work, the criterion was to develop a robust structure, in terms of software, to provide for ease of expansion, i.e. without major changes in the code, rather than to develop a complete single module, for example implementing all types of components in detail, but without the other modules already presented.

5.7- Knowledge Base and User Interface Separation

In every expert system, the user interface can be used for several purposes such as: enabling the expert system to pose questions to the user about the problem at hand; providing explanations about why the system is asking particular questions; display the derived results; providing graphical output for the derived results, etc.. Knowledge-based system projects have been known to fail due to inadequate design of their user interface. The emphasis in the development of a knowledge-based system has traditionally been with the expert and trying to represent his/her knowledge (GONZALEZ & DANKEL,93).

As pointed out earlier, at the beginning of the interaction process with one of the experts the user interface issue was risen, when in the first version the expert system was fully dependent on textual facilities for input and explanation output, and on a graphical output tailored for a specific package. At that stage, although the prototype had its basic structure in terms of rules and classes established (which is virtually the same structure valid at its "final" stage) the expert replied was very disappointed, for he did not possess the graphical package required for analysis the output. This feedback emphasised that for this knowledge domain, fluid power design, the engineer does depend mostly on the graphical analysis of the problem, through a system diagram. This conclusion was already expected, but the degree of dependence was reinforced. Therefore, a search for an alternative, more wide spread, means to present the output was done and the HTML interface was developed, as described in this chapter.

The HTML interface is as a standard output mode for presenting graphical and textual information, and it was demonstrated to be much more powerful, in terms of user acceptability, than the previous graphical mode. However, even with the HTML interface for output, the prototype versions submitted to the expert, for a period of six months (from March to September 97), relied always upon standard text input. This approach was taken based on the following points:

- Despite of the user interface relevance, the most important part of the prototype was **always** considered to be its knowledge base, rule structure and class attribute definition. Thus, during the period when the expert was testing evolving versions with standard text input, it was possible to test how clear the questions to the user were, as well as how representatives the load attributes and result descriptions were for fluid power design engineers.

- At the same period, the knowledge engineer was supervising the implementation of a Graphical User Interface (GUI), in development at EDC [6]. Although the details of this implementation, in terms of programming, were not the knowledge engineer's direct responsibility, the specification and validation of the GUI were directly under his control. Therefore, the inputs

---

[6] Ray Cheung is currently a Research Associate at Engineering Design Centre, working in the area of Graphical User Interface.

received from the expert during his tests of the standard text versions were being used to improve the GUI during this period.

- During that period, decisions related to the project architecture were being made. Mainly, there was no clear definition on which should be the basic platform, i.e. PC or Workstation, and for the PC platform, the developing language (Visual C++) was being learned at EDC.

Based on literature guidelines and on the analysis of foreseeable aspects, which were clearly proved with the above points, the decision was, since the implementation start-up, to develop the prototype system with a clear separation between the knowledge base and the user interface. Hence, the knowledge engineer had always complete control on the system expansion, being capable of validating its knowledge base even without a complete GUI version. This fundamental decision confirmed to be one of key aspects of the system development process, for the evolving versions had a gradual acceptability by an expert, up to the point that he spontaneously decided to demonstrate the prototype (even with a standard text input) in one of his seminars. Another aspect to be mentioned is that with the expert's increasing understanding and acceptability of the standard version, he became much more interested and involved when submitted to a GUI version. All contacts referred to in this section, in terms of expert's feedback, were established through the Internet with a constant exchange of messages and system files. Besides the advantages already mentioned regarding the Internet, those contacts meant that the level of detail contained in the interface and files descriptions was being tested at the best level available, because all information relied upon email messages and associated transferred files, without face-to-face contact (SILVA & DAWSON, 97b).

In order to provide a system with clear separation between knowledge base and user interface, it was necessary to define a file structure and a set of functions to guarantee that the system core, i.e. the knowledge base, could gradually expand and different platform versions had the same functionality. This file structure and set of functions are represented on figure 5.17.

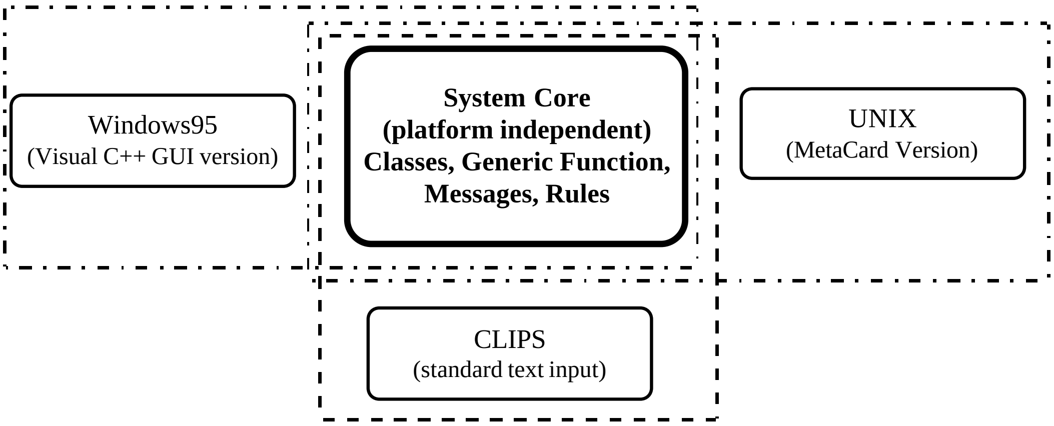


Figure 5.17- File Structure with system core as intersection among all versions.

As this figure shows, the system core is composed of set of files with the main definitions common to all versions. This structure was achieved by encapsulating all calls for standard CLIPS I/O functions in only one file, which contained the basic I/O functions (for instance: to read a field input, to present a sentence on the screen, to format data in a table, etc.) and its corresponding user-defined functions. Another file was created, only for better distribution and program structure, with more elaborated functions regarding the system output in textual version which contained calls solely to the user-defined functions located in the previous file. In figure 5.17, generic functions mean those related to object attribute and knowledge manipulation, e.g. set load values, get supply pressure, etc., and they are platform independent. Based on this structure, it is possible with the implementation of only these two files in other versions (or platforms) to have the same functionality. This structure also makes easier to develop versions for other tools, thus, as mentioned before a full version for Internet is among the plans for future projects. The flexibility obtained from this structure confers a great capacity for expansion of the system, considering this project as a product, it can be mentioned that such key decision taken in the first stages of the project development proved to be a concurrent engineering perspective brought to the system evolution.

5.8- Fluid Selection Module

Another aspect to consider in the design of a hydraulic system applying the life cycle concept, defined in chapter 1, is the issue of maintainability. Among several points to be taken into account for providing a satisfactory maintainability is the fluid selection. Due to its importance in the present context as well as the great variety in terms of available fluids, the expert system incorporates a module to assist the designer in selecting the hydraulic fluid.

The purpose of describing the Fluid Selection Module here is dual. At first, to demonstrate how the Object-Oriented Techniques and system structure presented throughout this work were equally applied to model, albeit very basically, the process of fluid selection. Second, the description will exemplify the expandability of the expert system, for this module was initially developed as a separate knowledge base and easily integrated in the main system.

Using the same knowledge representation technique presented so far, i.e. classes and rules, this module has the oil as the basic class, whose attributes were modelled as follows:

> *Oil-id*- an identifier of the oil object, it is used also to count how many types of fluid satisfy the searching criteria.
>
> *Kind*- denominates a fluid according to its standard classification, i.e. H, HL, HLP, etc.
>
> *Application*- describes a set of key words all the application areas or features for a specific fluid.
>
> *No-applicable*- defines the limitations of a specific fluid, for example a certain type of fluid may not be applicable for using with fire risk.
>
> *Additives*- specifies the family of additives contained in the fluid.
>
> *Pressure range*- determines for which range of pressure a type of fluid is applicable.
>
> *Description*- presents a general description of the fluid.

As described in the main system, here the structure also allows a future inclusion of more attributes to this class. The control pattern approach, defined before in this chapter, was similarly applied to this module. A basic description of this module is presented in figure 5.18.

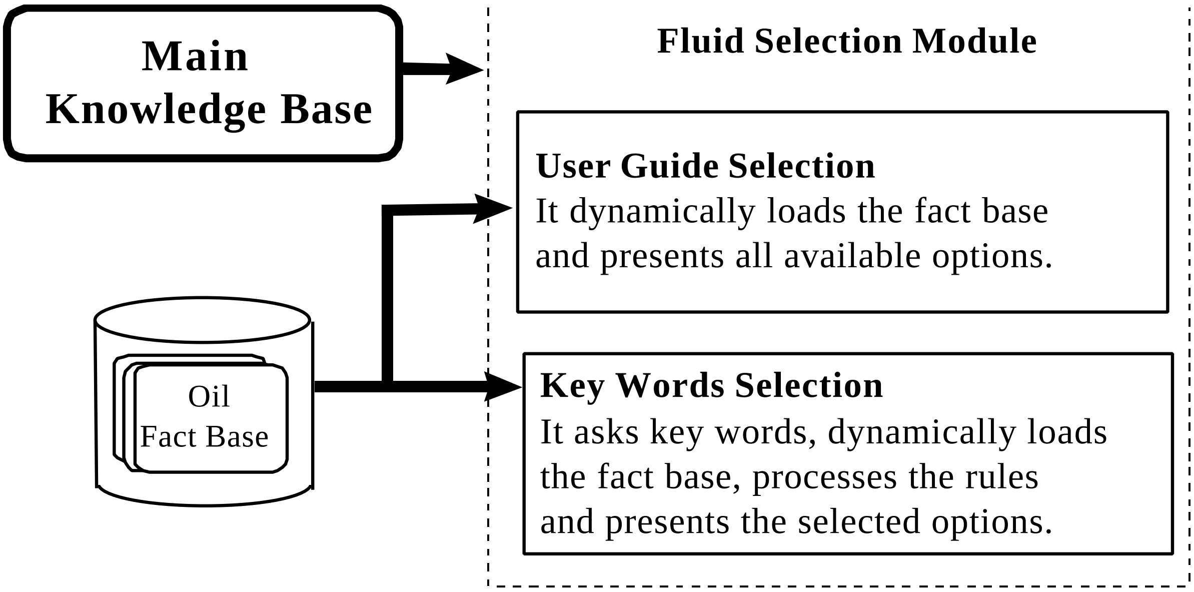


Figure 5.18- Fluid selection module- Interaction with the Main Knowledge Base.

As this figure depicts, the fluid selection module offers two alternatives, in the first one, User Guide Selection, the user is presented with all available options in the fact base. Thus, based on their attributes (application, no-applicable, pressure range, etc.) the user selects the best option for the design. In the second alternative, Key Words Selection, the user is requested to enter a set of key words (e.g. fire, environment, anti-corrosion, etc.), then the prototype searches for the options according to this set.

In both alternatives, the fact base (a database structure in CLIPS fact format) is platform independent and open to inclusion of new types of oil. In the present version, for the sake of implementation, solely eight types are available. Also in both alternatives, for each session the expert system creates oil objects, with all attributes of those facts that are compatible with the selection. As objects are created, they can be also manipulated by the main module entities, i.e. system or designer objects. The reason for creating objects is that this type of entity confers a great flexibility in terms of manipulation, when compared to facts that can be only applied to fire rules.

The integration between the fluid selection module and the main knowledge base is done by calling a single function, which executes two instructions. First, it adds the corresponding control pattern fact (defined before) to the fact list, similarly to the procedure in the central control rule. The second instruction starts the execution of the corresponding rule, which asks the user to select between the User guide selection module and key words module. With this integration in the main knowledge, the expandability of the prototype is made clear. Therefore, a similar approach can be taken to include other knowledge base modules of relevance for the design of fluid power systems, such as, Failure Module Effect Analysis, Cost Analysis, Design for Assembly, and so forth. These modules can be developed to handle different entities and/or other attributes of the entities already implemented in the expert system. Further aspects on the expandability issue will be discussed in the next chapters.

Chapter Six
Prototype Validation

This chapter presents general issues on expert system validation, including the theoretical background required for the successful conduct of this activity and its different methods. It also documents the test and validation results carried out throughout the prototype development.

6.1- Definition of validation

In simple terms, validation is the final quality control step of knowledge-based systems. Validation ensures that the output of the system is correct (however that is defined) and that the developed system is what the users want and need (GONZALEZ & DANKEL,93).

In the context of the developing system, this definition provokes the question: What is a correct output? Since the system aims to support the design of hydraulic systems, a correct output would require that the user (a designer) agrees with the quality of the knowledge base and feels comfortable with the results generated by the system.

Considering the complexity and comprehensiveness of a design task, the best way of accessing this information was to expose the prototype to key users during the development process. As can be verified in chapter 5, with the chronological description of the evolution process, this was the adopted approach. Therefore, the task of validation was considered throughout the development as a fundamental aspect of the incremental model chosen for the project, defined in chapter 3. This validation process can be classified as informal validation, which is valid but not sufficient for developing expert systems. Hence, a more formal validation method through tests was also applied in this development, as reported below.

Before dealing with specific issues on validation, it is worth to mention that, in Expert Systems, validation is directly related to other aspects. For example, the maintenance and evolution of an expert system is more of an open-ended activity than with conventional programs. Because expert systems are not based on algorithms, their performance is dependent on knowledge. As new knowledge is acquired and old knowledge modified, the performance of the system improves. In a very real sense, a commercial expert system may never really be finished- it only keeps getting better (GIARRATANO & RILEY,94). Hence, the aspects considered in this chapter deal with the validation of the prototype system, in terms of its user acceptability and knowledge base usefulness, in its present stage, and by no means does this represent a final commercial version.

6.2- Validation Metrics

In order to perform an adequate validation, it is necessary to have the objectives of the computational system clearly defined, for they must serve as the metrics based on which the achievements of this project should be measured. The project objectives, specified in chapters 2 and 4, are summarised as follows:

To demonstrate that the computational structure developed in the project, involving design concepts, tools, the development approach and prototype system, is sufficiently robust, expandable and modular to:

1. Model some of the main entities involved in the design of hydraulic system;
2. Supply a basic means of communication between the designer and the other participating members in a project team, or between client and supplier;
3. Provide guidelines to the designer about issues such as: maintenance, operation, cost effectiveness and safety related to hydraulics, during the conceptual design stage;
4. Offer an environment to include other Concurrent Engineering aspects;
5. Serve as a basis for integrated projects in the fluid power design area, but at the same time supporting the solution of key sub-problems.

Based on the above definition, three features should be considered in the validation, they are robustness, expandability and modularity. The first aspect, robustness, is related to the system capacity of providing correct and complete outputs given correct input, consistent output given the same input on further occasion, reliable use so that it does not crash due to bugs, etc. (GIARRATANO & RILEY,94). As can be seen, this issue involves the recognition of several other objectives, e.g. correct, complete, consistent and reliable output. The robustness aspect will be considered in this chapter through the description of feedback from different users (experts and non-experts). The expandability feature is basically analysed with the study of how the knowledge base was increased and the user interface improved in a chronological pattern, as a consequence of constant feedback throughout the project development. The third feature, modularity, relates to how the system structure and knowledge representation (i.e. Object-Oriented Techniques and Rules) used in the project allowed satisfactory robustness and expandability to be accomplished.

6.3- Validation Methods

The fact of the matter is that there is no clear procedure to be used to validate an expert system, though the Turing Test is known as the best attempt to do so. This method was proposed in 1950 by Alan Turing, the test consists of an experience in which a person plays the role of the interrogator, who is in a separate room from the computer and another person. The interrogator can ask questions of either the person or the computer by typing questions and receiving typed responses. However the interrogator knows them only as A and B and aims to determine which is the person and which is the machine. The goal of the machine is to fool the interrogator into believing that it is a person (RICH & KNIGHT,91). Even the theorists in Artificial Intelligence have not achieved a consensus whether a machine will someday be able to fully pass in such test, though adaptations of this test have been used to validate some expert systems, for example MYCIN (GONZALEZ & DANKEL,93). Hence, despite its historic importance in the context of AI, the Turing test is more recognised by its philosophical challenge than by its use in practice. Based on this aspect, due to the inherent comprehensiveness and complexity of the design activity and because of the incremental approach adopted in this project, no attempt was made to reproduce the Turing test, perhaps to achieve such objective could be considered the "Holy Grail" of any expert system- especially in the area of design.

Therefore, in this project a more pragmatic approach was taken, that means always the evaluators were conscious that they were dealing with a prototype expert system. Some drawbacks of this approach are that experts or users can have some misconception, or even prejudice, towards computers and thus interact with them in an uncomfortable manner. On the other hand, the designer can feel threatened by an expert system, which could in future be used to replace his/her position.

Some measures were taken to overcome the above drawbacks. First, a clear definition of the potential applicability of the prototype was made. The evaluators were informed that the prototype was not tailored to replace the designer but that rather it would have two major applications. For a noviciate engineer (or student) without expertise in hydraulics, the system would provide a better understanding of the design process and some guidelines in hydraulics, and for the experts, the system would empower them with more freedom of choice, a quicker start-up in terms of generating alternative solutions for the design problem, a facility to balance in a much broader way general criteria for the design at the conceptual design stage and also provide a tool to automatically develop the basic calculations. Therefore, more time would be available for other activities, e.g. a better analysis, for the experienced designer.

The second measure taken was to adopt an attitude of co-operation between the knowledge engineer and the experts directly involved in the project, by making them aware of the relevance, uniqueness and consequent challenge of their contributions for this endeavour. Although the knowledge engineer was not able to convince all the experts [1], at least the majority of them (some mentioned in chapter 5) were pleased to collaborate throughout the project. Truly, in an expert system development, the human issues may become even far more complicated than the technical aspects (BERRY & HART,90) and a certain degree of psychological insight is necessary to deal with them (COOKE,92). Further specific aspects on knowledge acquisition can be found in (HART,92).

---

[1] In a key area for the project, the knowledge engineer even faced an expert who was apparently interested to co-operate, but in fact almost jeopardised the project. This expert was a typical "High Priest" profile (GONZALEZ & DANKEL,93).

Since the beginning of the project, despite the above mentioned reluctant expert, the knowledge engineer had plenty of knowledge sources to explore, be they from general literature, expert interviews or the market. This aspect proved the validity of one of the main claims of the project, that hydraulic system design would be a feasible area to be used for an expert system development, for the well established theoretical foundation of hydraulics, its comprehensiveness, and the knowledge engineer's background in this field would overcome possible difficulties due to human issues.

One common point of debate among AI researchers is when the validation process should start. A common argument is that to validate anything but a nearly complete system against criteria is foolish because it is only then that the system has sufficient knowledge to make accurate decisions. Other researchers argue that the validation process should begin with the specification of the system and continue throughout the development process. This last point of view has become the more widely accepted of the two, resulting in validation becoming ingrained in the development process, rather that being simply a phase within it. Nevertheless, the broader implication of the first statement is that there still needs to be some final validation process that evaluates the performance of the system against its most stringent criteria (GONZALEZ & DANKEL,93).

The present project followed the second argument, i.e. the validation really began in the system specification, which was done by submitting the work proposal, as the qualifying project, to a panel of examiners. The proposal was also the linking point between the knowledge engineer and EDC. Even after starting the system implementation, in the UK, the proposal was constantly exposed to the experts' community through visits, technical papers, participation in events and the use of Internet, as described in the previous chapter. This constant exposure was fundamental for two reasons, to define that conceptually the project was feasible and to start a network of potential users and contacts. Both points proved to be decisive during the development.

The next section will describe some of the tests carried in the project, and how the knowledge base was enhanced as a result of the continuous feedback from the users. The theoretical aspects of validation will be included throughout the section.

6.4- Validation Tests

This section is divided into three parts; first follows a report format of some interactions between the knowledge engineer and the system users throughout the project, emphasising the increasing development of the knowledge base to accomplish with some users' requirements. Secondly, a set of formal tests is documented, relating general acceptance from users without expertise in hydraulics. Finally, some issues generated from experts' formal evaluations are discussed. As can be seen, based on the three aspects previously defined (i.e. robustness, expandability and modularity), the first part of this section covers, mainly, the expandability aspect, but also presenting points on the other features, while the two last sections are more concentrated on the robustness aspect.

Each technological area has its profile in terms of people that work on it. Despite increasing technological change, mainly due to the influence of computer applications, the area of hydraulics is compared to other areas composed of members whose expertise comes from several years of work in industry rather than in research, and who have been gradually become familiar with computer technologies. As the knowledge engineer had to search for expertise in this market, it became necessary to interact with potential experts with little or no involvement or background in the area of Artificial Intelligence. Therefore, it was predicted that in order to transmit key concepts, deeply related to AI, for this market, a prototype should be used. Hence, as early as possible a prototype was developed to demonstrate the concepts, as presented in the previous chapters, this turned out to be one of the most important aspects of the development process.

In order to illustrate how the prototype evolved as a result of its own application, next some parts reporting an expert's feedback will be discussed along with their implication in terms of AI and the impact in the increase of the knowledge base.

6.4.1- Prototype Expandability based on a Chronological Description

Among the several pitfalls in building an expert system, a special one may arise when trying to develop a system in a domain where the experts are blue-collar workers. These experts tend to be skeptical of the academic approach to problem solving and even more skeptical about the value of AI and computers (WATERMAN,86). Although this is not exactly a reflection of experts in hydraulic system design, Trinkel [2], an expert who became a major contributor to the development, showed a certain degree of scepticism at the beginning of his involvement in the project. This is presented in the next quotation.

I would be happy to work with you on what I think is a monumental project. In the past there have been several articles in Hydraulics & Pneumatics magazine on others who have attempted to do the same thing. So far I have seen no further developments from these undertakings. (16/02/97)

At that point, Trinkel had had contact with the work proposal, and spontaneously decided to respond some general points on design. Based on this, he was invited to informally collaborate in the project. This invitation was accepted with the above quotation. The point to be emphasised

---

[2] The way Bud Trinkel had contact with this project was reported in the previous chapter.

here is that despite of his practical expertise in designing hydraulic systems he had no deep contact with AI before. The date[3] is placed to document the chronological sequence of tests.

The next step was to send technical papers to Trinkel to present more theoretical aspects related to the project and also suggesting that he could test a very "embryonic version". This would require a good deal of personal interest from him, but in case of a positive feedback it could establish a collaboration scheme, in which the expert would identify himself as an important agent of the process.

The first contacts also served to analyse if the expert was really appropriate for the task, thus the expert's traits (such as competence, articulate, self-confidence, availability, open-mindedness and enthusiasm) were being evaluated (GONZALEZ & DANKEL,93).

As result of this contact, Trinkel accepted to test the system, and only using guidelines from a readme file was able to run the first version. However, the knowledge engineer knew that, as the literature points out, initial perceptions of the system (that of an early limited function version of the final system) could lead to confusion about what to expect and be difficult to change in later stages. This means where there are high expectations, any signs of poor performance of the prototypes can lead to a lack of confidence in the development (EDMONDS et al.,90). The next quotation reflects the first feedback.

I had a little trouble getting the program to run but finally got there… After going through a couple of made up problems today I could start to see where you are headed I think.

I was probably expecting a lot more development than you have so far. I got lost in some of the questions and explanations…Also I did not fully understand the section that asked for "Effort", "Flow" or "None"…To me it seemed too simple with much less questioning that I had imagined.

I did not have "dymodraw" so the end results were not clear. Does it put out a schematic or is it just a block diagram output?

I hope this information is some help to you and not too negative. I'm not sure where I might start on a project like this so I don't want to appear to know it all. (06/03)

Although this first feedback seemed very disappointing, in fact it became a turning point in the system development. It proved that it would be much easier to interact through the prototype tests. As the literature points out, criticising an existing model is a very productive way of refining the knowledge base (HART,92). The following key measures were taken based on this feedback.

- The former used terminology for inputs, based on bond graph description, where effort and flow respectively represent force/torque and speed/rpm, would not be accepted by a general

---

[3] All references to months in this section relate to 97, unless different stated.

user, even with expertise in hydraulics. Hence, the input description was replaced by a more familiar form.

- As mentioned in the previous chapter, the graphical output that was only available for dymodraw (the graphical front-end of DYMOLA) also could not be accepted as an unique option. This point triggered the search and further development of the HTML agent as an alternative for textual and graphical output.

An important aspect in the knowledge acquisition process is the speed and frequency on which the knowledge engineer replies to expert's feedback. There should not be an excessively long delay in between acquiring knowledge and giving feedback (HART,92). Hence, throughout the project, the knowledge engineer tried to give a high priority to the tests carried out by Trinkel, implementing and sending new versions as soon as possible.

The implementation of the HTML **module** was almost immediately after the above reply, however to produce the graphical files corresponding to the circuit diagrams took some time. After testing the new version, Trinkel gave the following reply.

…I did notice in working with the program that I can open my browser with a START.HTML file in the display folder and go directly to the screen that starts the SYSTEM.HTML.*

I got it to work and have tried a couple of circuits with fantastic results as far as the schematic it turned out and speed of the program…

I was impressed with the fact that I could print out the complete circuit and sub-circuits on my ink-jet printer…(23/03)

Based on this reply, a new dimension for the development was established for it boosted the confidence in the output chosen option. The system diagrams along with the textual explanation were accepted by the user. It took less than three weeks between the two previous replies. This fact demonstrated a mutual interest from the knowledge engineer and the expert.

Besides the evident qualitative expansion obtained with this improvement, the fact that this facility was implemented without replacing any of the already implemented features (in terms of knowledge representation) but only adding new attributes and message handlers to the defined classes, as pointed out in chapter 5, clearly demonstrates the modularity of the system structure.

After the above feedback, it was clear that the modelling of the basic functionality of a hydraulic system was achieved, for this feature is almost fully represented by the system diagram description. However, as the prototype also aimed to map the design process, the next steps in the knowledge acquisition process were concentrated on this sub-area, i.e. the design process. In order to accomplish this task, Trinkel received a series of questionnaires relating the main phases and possible bottlenecks of the design activity. By the end of May, he was convinced that the approach and pace of the development was in the right direction, just as a demonstration of his interest on the project, the next quote was published in his company web-site by that time, *"See the work being done on a program called Schemebuilder that will take information about a circuit and translate it into a completed schematic."* The point to emphasise here is the change from a sceptical position to an integrated participation in the project.

Concurrently to the interaction with the above expert, the knowledge engineer was also interviewing other experts, one of them, Seward [4], was from an academic background. In this case, the interaction was being carried out through the reverse engineering of the hydraulic system of an excavator. This hydraulic system had been redesigned to modify the machine from a standard manual operated excavator to a servo-operated computerised system, known as LUCIE project (Lancaster University Computerised Intelligent Excavator) (BRADLEY & SEWARD,92).

In this context, the face-to-face contact was possible and knowledge acquisition took place through a series of structured/unstructured interviews. Because of this more familiar approach and with the experience of the structured interviews already done, this part was more concentrated on the unstructured interviews. In one these unstructured interviews, the expert was asked to distinguish between the weak and strong points comparing the manual and servo versions of the excavator system. An example of the results is presented in table 6.1.

Table 6.1. Comparison between two versions of a hydraulic system.

Aspect	Manual Version	Servo Version
Strong	*-Because of there were three pumps feeding separately the sections of the system, the main functions of the excavator were independent.* *-Cheaper and more reliable.*	*-It proved to be a useful research field for robotic excavators, which will be applied in dangerous situations.* *-Smoother operation, independent on the operator's skills.*
Weak	*-Due to the nature of the manual operated valves, this version constrains the cabin layout.* *-Greater noise level in the cabin, due to the manual operated valves.*	*-The position feedback sensors located in the joints of the excavator proved to be a design problem, in terms of cost and reliability for this application area.* *-Because of the absence of an unloading valve in the power supply and with a close-centre configuration for the directional valve over-heating took place.*

The points listed in table 6.1 were used to include a servo circuit option for each conventional one (on-off operated valve circuits). Therefore, by the end of May, the prototype had included circuits with proportional valves as alternatives for each conventional circuit. This enhancement was considered by the experts of great importance, both from the design methodology and hydraulics perspectives. From the observation of Seward in a talk-through situation

---

[4] Derek Seward is a Senior Lecturer at the Engineering Department- Lancaster University and the co-ordinator of the LUCIE project.

(HART,92), i.e. during the knowledge acquisition, it was possible to discover important missing points in the design of the retrofitted excavator. For example, the absence of the unloading valve in the power supply circuit which caused over-heating in the system. This aspect was considered to refine the design itself and to expand the knowledge base, for this power supply circuit was incorporated in the knowledge base, including an unloading valve in the component list, as described in chapter 5.

When developing systems with knowledge-based components it is essential to adopt a design and development model which recognises and accommodates requirement and design changes as an inherent part of the development process. It is also important to recognise that the design and development model itself is only a framework which will be adapted to the particular circumstances of a project (EDMONDS et al.,90). The present project adopted exactly this concept. Considering again the interchange with Trinkel, as the prototype developed it was being submitted to his continuous evaluation and consequently additional requirements had to be considered. By mid of July, the system was able to handle also alternatives for power supply and allowed the input of some quantitative parameters. This facility relates to a specific **module** implemented to handle parameters and equations corresponding to size hydraulic circuits and components, as presented in chapter 5. In this context, similarly to the HTML module development, no replacement was made in the knowledge base, rather solely more attributes and message handlers were implemented to manipulate, for a different purpose, the already existing objects, i.e. load, circuit, system and component objects.

Both above mentioned facilities, i.e. handling of alternatives for the power supply unit and of quantitative parameters, are key aspects to bring a concurrent engineering approach to the design of hydraulic systems. First, the freedom of choice among alternatives for the power supply, as described in chapter 5, guides even an experienced designer to consider more options, and therefore possibly a better selection, for the part which is regarded as the heart of any hydraulic system (GREEN,85). Consequently, the decisions made in this choice are among those of greatest impact in the life of a hydraulic system, affecting among other aspects its maintainability, safety and cost. In the other hand, the manipulation of quantitative parameters in a hierarchical form, i.e. from circuit to components, as presented in chapter 5, permits the calculation of basic variables (e.g. cylinder areas, motor and pump displacements, flow rates,, supply and return pipe line diameters, etc.). This facility frees the designer from having to execute the calculations, thus it speeds up the design process and provides more time for the designer to analyse the options generated by the expert system in a much broader way. For example, the designer can rank the options based on power efficiency, cost effectiveness, easy maintenance, easy operation and precision, which are the criteria already implemented in the weighting tool, as shown in chapter 5. Clearly, the design process speeding up and the comparative ranking offer a concurrent engineering approach to the designer.

In order to demonstrate the increasing interest and requirements from Trinkel as the system progressed, and another consequent expansion in the system, the next quotes are presented.

It would be excellent for the United States to have the inputs for force, pressure etc., with an option for the imperial system. They are normally PSI, pressure, Pounds, Force, IPM Inches per Minute, pistons speed, Inches for stroke, and Inch or Foot pound for torque. (18/07)

It appears the new circuit information replaces old information so it is permanently lost. Is there a way to save individual sessions to a file so it can be looked over later?… (27/07)

The first point relates to user acceptability towards a specific unit system, while the second comment states the importance of providing comparison among results from different sessions. Although these facilities had not been anticipated, with the Object-Oriented Techniques used in this project, it was clear how to include and manipulate additional attributes. Therefore, these requests were implemented and new features were defined according to the market needs, for instance, an alternative unit system.

In relation to the user acceptability, until end of July all tests were carried out using the standard text version for inputs, however, a Graphical User Interface (GUI) was being developed. At that time, the system development was further encouraged by the interest of Trinkel to demonstrate the system- even with the standard version- during one of his seminars.

Despite the priority given to the knowledge base development, a great attention was also being given to the prototyping of the user interface, which was in constant enhancement through the interaction process, leading to refinement of the input instruction in the text version and the output quality. Some reasons are given for the necessity of prototyping the user interface design (EDMONDS et al.,90). These are as follows:

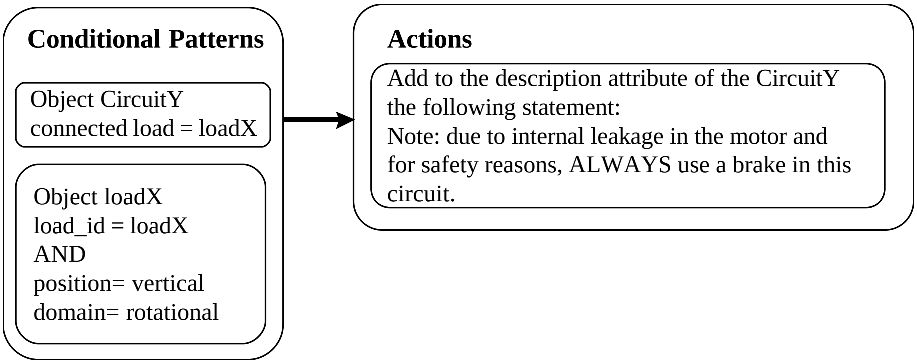
- User interface formal specification can be very difficult (e.g. a written document does not enable users to visualise the system in use);
- There is a variation in the types and styles of users for any system;
- The complexity of requirements often leads to conflicting design goals which cannot be detected or resolved by written documentation;
- Properties such as user-friendliness and ease of use are highly subjective; the ‘look and feel’ is revealed only when the system is ‘live’.

Based on the feedback obtained so far, by mid July, and considering the mature status of the GUI version, a new set of files with the GUI front-end was sent to Trinkel. The next statement shows his opinion on this version.

I set up the program and tried it out. This version is much more appealing and is easy to use since I had experience with the original offering. The things I see that need work are probably due to you not having time to implement them… (08/08)

As expected, the acceptance of the GUI version was much greater, mainly due to his understanding of the previous version. Once again, the statement declares the confidence in the approach taken and progress made.

In order to emphasise the system robustness and its expandability, it is important to mention that the developing prototype was fully demonstrated over a three day period as a workable tool to in excess of 20 industrial and academic participants at Pittsburgh Hydraulics and Pneumatics Show[5], November 11-13th. Considering that this event corresponds to the main commercial exposition in the area of Fluid Power in the United States, and as such gathers the main industries and academic institutions in this field, including some software suppliers as mentioned in chapter 4, it was a great challenge to expose for such an expert public a research prototype which had, at that date, only about one year in terms of implementation. However, despite its academic basis and embryonic stage, there was a great deal of interest in the prototype, and only one comment of substance was made to question the knowledge base. This important point, that had not been perceived by the users so far, related to fact that for every rotational load in a vertical position a mechanical brake is required due to internal leakage in the motor. In spite of the fact that this was a very important missing aspect, it was easily taken into consideration through the addition of the following rule.



Add to the description attribute of the CircuitY the following statement:
Note: due to internal leakage in the motor and for safety reasons, ALWAYS use a brake in this circuit.

Figure 6.1- Rule added as result from the Pittsburgh feedback.

The above rule demonstrates that the interaction between load and circuit entities, expressed through their attributes modelled in the corresponding classes, was sufficient robust to easily implement the comment. It also shows the relevance of the description attribute, predicted since the beginning of the project, for complex issues in the design area can only be explained through a combination of graphical and textual descriptions as applied in the prototype HTML

---

[5] This event is yearly organized by the Hydraulics and Pneumatics Magazine (http://www.fpweb.com/).

files. It is also important to emphasise that, similar to the previous enhancements, here only an inclusion was done, i.e. no replacement was necessary.

As can be seen, the reported aspects in this section defined clearly that, so far, the system structure has been expandable and robust to include the requirements resulted from the interaction with different users.

As can be verified in this section, and in some parts of chapter 5, the experts had an invaluable contribution to the system evolution. They were key participants for defining on points to improve the system. However, a clear distinction should be made between defining *what* to do, which was a consequence of the experts' feedback, and defining *how* to do and implement it. This later part was a complete responsibility of the knowledge engineer in all modules and phases of this project.

6.4.2- Usability Tests

Although the responses gained from the above tests were very supportive to the development, it should be considered that they cannot be valid as the only type of tests as far as validation of the prototype is concerned, rather the above answers reflect an evolution of the knowledge base and user interface throughout the process. To consider experts as main users is not the ideal case for testing the complete usefulness of an expert system. Hence, a new type of tests was defined aiming to validate the usability of the prototype with non-expert users.

In this context, usability is defined in terms of the effectiveness with which a system allows a task to be carried out, its learnability, flexibility and attitudes it engenders in users (EDMONDS et al.,90).

For the usability validation, a group of ten 4th year students of mechanical engineering at Lancaster University was selected to test the prototype in a laboratory experiment. All students had attended a course in fluid power, but they had neither true experience in designing a hydraulic system nor background in AI. This profile matches the description of an user type, presented in the section Validation Methods, as one of the potential applications for the developing system, i.e. a student or a novitiate engineer. The tests with the students also highlighted several comments, made by expert users who tested the system, as well as by others for whom the system had been demonstrated, that the prototype certainly would have an educational role in engineering.

In order to be introduced to the task of validating the prototype system, the students were given one session (talk) about AI applied to design in general, a demonstration of the prototype and one hour of a free trial (going through) with it. At that time, no user manual was provided and all information relied upon the understanding of the system input/output.

Besides the validation of the prototype system itself, the students had to use it to design two hydraulic systems according to the exercises defined by the lecturer [6]. This task had a time constraint of four hours (one morning session), including a report elaboration (regarding the exercises) and the evaluation form of the system, some exercises are presented in appendix 2. The students were divided in five groups, each group had a PC for the tests whose minimum specification was: Pentium 133 MHz with 16 MB Ram and 256K Cache.

In terms of validation criteria, in this test the intention was not to obtain feedback about the knowledge base completeness, for the users were not experts. Rather, other aspects of the prototype were being validated, thus a specific set of criteria was defined for this task. The students were asked to give their considerations, expressed according to a score scale (strongly agree, agree, accept, disagree and strongly disagree) in a corresponding decreasing order (4 to 0) for different criteria (GONZALEZ & DANKEL,93). It should be mentioned that in order to keep the test fidelity as much as possible, the students were allowed to remain anonymous in the evaluation report, which was separated from the exercises.

For each criterion that received less than 2, in other words, that the student disagreed, the student was asked to specify why the criteria had not been met and what should be done to enhance the system in that specific criterion.

The criteria were defined by the knowledge engineer, considering the metrics presented at the beginning of this chapter. The criteria set was also discussed with the lecturer. Table 6.2 presents the criteria with the results from the students' evaluation.

Table 6.2. Students' validation criteria set.

Criterion	**Score(%)**
1- Ease use in terms of qualitative input	82.5
2- Clear information flow	75
3- Adequate explanation output	77.5
4- Clear understanding of quantitative input	60
5- Capacity to rank the alternatives	85
6- Ability to propose alternative designs	77.5
7- Clear understanding of power supply alternatives	82.5
8- Satisfactory response speed	100

As can be seen, the highest ranked criteria (satisfactory response speed; capacity to rank the alternatives; clear understanding of power supply alternatives and ease of use in terms of

---

[6] David Dawson, Senior Lecturer and EDC Deputy Director, was also the lecturer responsible for the previous course that the students had in hydraulics.

qualitative input) emphasised the main feature of the system, which is to provide a decision making supporting tool for the design task in the conceptual stage.

Despite the general positive feedback, table 6.2 clearly presents a specific point for improvement, which is the quantitative inputs description offered by the system. This aspect can be a result of different reasons. In this phase, quantitative input, all inputs require a better understanding of the mechanical definitions, e.g. maximum static force extended, maximum dynamic force extending and retracting, etc.. Also this phase, compared to the other parts of the system, requires the greatest amount of input information, this means eight attributes for a linear load, four attributes for a rotational load, plus the definition of supply pressure and operational sets. Considering that even a simple hydraulic system can demand three actuators, the number of quantitative parameters can vary from 13 to 33 inputs, plus the definition of operational sets. The low score for this criterion can also be due to a lack of experience with the system, since the users had a very short time to learn the prototype. However, the knowledge engineer did recognise that a better description of the quantitative inputs, compared to other parts, was required. Thus, this part became the highest priority to be focused on the development of a help-online facility.

Beyond the aspects analysed by each student through the score table, their comments brought some important contributions as far as future works are concerned, for example: ‘it would be good to see the system linked to a database for actual component prices’ and ‘it needs an option to alter existing systems’. These comments, together with other issues, will be discussed in the next chapter.

As with any other software product, it is impractical and infeasible to provide 100 percent performance guarantees on any knowledge-based system. This problem is further compounded by inaccuracies within the knowledge itself on which such systems are based. Therefore, validation of knowledge-based systems is really a quest to determine if the system is performing acceptably, rather than perfectly. The determination of acceptable performance, furthermore, may vary widely depending on the particular application (GONZALEZ & DANKEL,93). Two important system parameters to be consider here are: accuracy and adequacy.

- Accuracy can be defined as the proportion of acceptable answers that the knowledge-based system generates. Likewise, acceptability can be defined as those solutions that agree with those proposed by an expert faced with the same problem.
- Adequacy, on the other hand, measures how much of the problem domain is covered by the system.

Taking into account the profile of those users (i.e. the students), the amplitude of the design task, the time constraint they had to use the system, the prototypical stage of the system, and the inherent complexity of evaluating an expert system, it should be considered that, in terms of its robustness and potentiality, the prototype had a satisfactory acceptance among the students’ group.

The issues of accuracy and adequacy can be further explored considered the evaluations carried out by experts, which are partially discussed in the next section.

6.4.3- Experts' Formal Evaluation

The previous sections documented the validation based on system evolution and its acceptability by non-expert users. In order to define an evaluation of the prototype usefulness and potentiality, in its present stage (i.e. in December 97), a group of experts, who had tested the system, was chosen to comment on the system features, e.g. its strongest and weakest points, general applicability, future developments, and so forth.

As a result of the system expansion, experts with different profiles and backgrounds were selected. This point also emphasised the system breadth, namely its potential areas of use.

Next, those experts' comments will be presented together with the knowledge engineer's arguments to explain, justify and/or expand on them. As in the previous section, the experts' comments will be in *italic*.

First Expert: Bud Trinkel

Profile: Hydraulic system design consultant in USA, his emphasis is on designing systems for industrial application. His web site address was given in the previous chapter.

Although the knowledge engineer had the opportunity to meet him, in the Hydraulics & Pneumatics show (in November), most of the contacts and his evaluation comments were carried out through the Internet.

Evaluation:

1- How long have you been testing this system?

> *Since the first program was presented, approximately one year ago. Before that I helped with the information gathering that went into the program.*

Comment: as can be seen, this expert carried out several tests throughout the development, described in section 6.5. His attitude changed from a certain scepticism to a co-operation posture, resulting from the system expansion between February and December.

2- How do you describe the system in terms of its user interface? (Input information, explanations, help facilities, etc.)

> *Early on I had some problems with some of the terminology. This has mostly been taken care of with the exception of one or two areas. Several of the corrected areas were handled by "information" notations at the particular area in question. There are*

> *some areas that I still have to think over before entering another steps that could be more user friendly.*

Comment: This item pointed out two aspects. First, it clearly showed an evolution of the system acceptability. Mainly, this was due the change from the Bond Graph terminology to a more familiar form used to provide system input. Second, it also defined key areas for further development, which are related to a better user support in terms of help on-line.

3- How do you consider this system would be useful for you (or your organisation)?

> *I envision using the software for preliminary design of most hydraulic circuits and adding refinements based on experience. The program does almost all the math work so more time is available to do fine tuning on the schematic. I also see this software as a training tool for new engineers and maintenance persons and a way for experienced persons to lessen their work load when it applies hydraulic circuit design.*

Comment: This point is in accordance with the potential applications of the software, defined in the section 6.3, this means the system can be used by noviciate and experienced engineers. It is also very clearly stated that the facility of executing the calculation was well accepted by the expert and seen as a way of proving “more time to do fine tuning on the schematic”. This possibly implies more time to consider other important aspects on the project, i.e. to involve concurrent engineering perspective in the design process.

4- In your opinion, what are the strongest point in this system?

> *Having all the math work performed with minimal input of system parameters. Offering options and weighting outcomes based on user input and preferences. Having a schematic drawing almost immediately to look at and discuss.*

Comment: In addition to the aspect of performing calculation, previously emphasised, this point also reinforces the acceptance of the weighting tool and fast response of the system, *“schematic drawing almost immediately”*, as a means of establishing communication, *“to look at and discuss”*, between the designer and the other participants in the design process, e.g. clients and component suppliers. Therefore, this point addressed in a positive way several validation metrics defined in the section 6.2.

5- What are the weakest points in this system?

> *Not being able to specify special circuits like regeneration, flow dividers, etc.. Not having a drawing editor that will interface with DXF files or others so the engineer can integrate it into his standard work environment.*

Comments: This answer is a basis for future developments in two distinct areas. First, a direct expansion of the system knowledge base with the inclusion of more *"special"* circuits. This can be done with the same system structure, i.e. classes, rules and messages, including only more options, and attributes, to define the loads with its complementary circuits. The second additional requirement is related to the development of a new computational agent, possibly to *"interface with DXF files"*. Although the knowledge engineer believes that this requirement is also achievable, it will demand a more comprehensive analysis, compared to the previous one, for it would link the system to a commercial standard (DXF) and, of course, it would require a complete understanding of this standard in order to develop the agent. So far, it is not clear that this is the best way to proceed, in terms of investing time and effort, perhaps concentrating on developing a full Internet version and other agents (cost, maintainability, etc.) can prove to be more effective. These issues are further discussed in chapter 7.

6- Would you be interest in collaborating with the system enhancement? If yes, how?

Yes. In the same way as before and does not interfere with my consulting work.

Comments: although this expert had a great contribution to the system development, all his inputs were without any financial charge, despite the fact that he usually charges $60 per hour plus expenses. This again demonstrates a great interest from this expert and it points out the ability of the knowledge engineer to interact with him.

7- State your general comments about this system:

Having seen some preliminary work by others and where they were stopped. I am impressed with the quality of what I see so far. I would like to see the work continue to completion and beyond. This program could not only design a working schematic but could give part numbers, pricing, start up recommendations and more so anyone could design and build reliable, efficient and long life hydraulic circuits.

Comment: Based on his participation throughout the project and his relevance in the hydraulics and pneumatics market in USA (he was one of the instructors in the Hydraulics and Pneumatics show in Pittsburgh), this expert should be considered a key participant for future enhancements, which, according to his comments, he is willing to do so.

Second Expert: Victor Juliano De Negri

Profile: Differently from the previous expert's background, this expert has an academic profile, having obtained his doctor degree with a work on methodology for testing hydraulic systems (DE NEGRI,96). He is also the supervisor of the Laboratory of Hydraulic and Pneumatic

Systems (LASHIP) of Federal University of Santa Catarina (UFSC), Brazil. Even considering that this expert is from the same institution that the knowledge engineer, during the testing phase only contacts through the Internet were made, for in this time the knowledge engineer was working in the UK, the questionnaire is based on the version of December 97.

1- How long have you been testing this system?

Approximately three months.

Comments: differently from the previous expert, who participated throughout the system implementation, the first version this expert received was in a much more advanced stage. However, this expert was one of the members of the panel who analysed the project proposal as a qualifying project, in April 96, as mentioned in the section 5.2. Therefore, even at his first contacts with the prototype system, he had a better understanding of the underlining concepts in this project compared to the previous expert. Thus, the early phase to convince the expert about the system potential and/or applicability, necessary with the previous expert, was not needed in this contact.

Considered the comprehensiveness of this expert's answer related to the next question, it is commented in parts.

2- How do you describe the system in terms of its user interface? (Input information, explanations, help facilities, etc.)

In general terms, the user interface is adequate, requiring only elementary information to facilitate the designer's task.
In my opinion, small corrections are necessary, such as:
In the first screen:
The main menu has no direct correspondence to the phases presented on the first screen. I believe that it would facilitate the understanding of the system use, mainly in the first times. Despite of this, presently after two or three sessions, the steps to follow are clearly understandable.

Comment: Although this is an important point, which can be easily corrected, it is not related to the system structure and knowledge representation, but only with details that are relevant for a final (full) commercial package.

At phase START, the great limitation is the impossibility to keep the information from the previous session, you have already explained this point.

Comment: as can be deduced from the answer, the knowledge engineer already replied about this point to the expert, but it is pertinent to document it also here.

When the prototype processes the information described in the load objects, it automatically generates the corresponding HTML files to explain all the objects generated (i.e. circuits, systems, components, designer information, comparative table, etc.). All these files are placed into a directory defined by the user at the beginning of the session and they can be accessed whenever wished. However, when a new session starts, all the previous created objects, not the files, are deleted. This is necessary in order to define other load objects. Despite the fact that CLIPS does offer an option to save all the objects, this function was not implemented because the file generated from it allows a complete mapping of the objects tree, with their classes, attributes and values. Then with a concentrated effort on reverse engineering this file the system underlining knowledge structure could be exposed, of course, this is not desired. Therefore, to implement a similar function, it is necessary to better evaluate other options and balance their consequences.

> *STAGE 1: Load Definition:*
> *2)Speed or RPM- It would be better to present speed and angular velocity.*
> *7)High precision position control- The word 'high' is too generic, whose interpretation can greatly vary from each one designer. Would it not be better to define in another way?*
> *8)Simple movement without feedback control- I think it is premature to state on the control with or without feedback. The fact of having feedback does not guarantee a minimum error of position, force or speed. If it is the intention, it could be defined as 'use of position switches, without displacement transducer'.*

Comment: This part of the answer refers to the improvement of the user input presentation in the first phase (load definition), as such it is very relevant. The numbers are options presented to the user's choice in a pull down menu. As can be verified, the expert is concerned with a clear definition, in terms of technical specifications presented to the user. In the prototype, the form of the input does not explicitly define the symbols that will be used to manipulate the concepts (for example: the option 2 is internally represented by the symbol *flow*), therefore the suggestions offered by the expert can easily implemented.

> *BROWSER- In the future, it would be good to rigorously follow the ISO 1291-1 and 1219-2 that deal with the standard symbols, including the component identification. This is a minor detail.*

Comment: Although the point was referred to as *"a minor detail"*, it is important to be mentioned. The knowledge engineer does acknowledge the fact that when the HTML agent was developed, as previously documented, the symbols were drawn with a commercial system, which claimed to be in accordance to the standards. As far as the component identification is concerned, the prototype presents a component list per circuit, as described in chapter 5, in the same file.

Probably, to add a number on the graphical representation of the circuit can require further considerations.

> *CIRCUIT ID: Circuit1*
> *Why to specify the type of component? Would it not be enough to specify the functional principle?*
> *For example: Pump- Piston Pump (I think it is premature to establish the type of pump in a generic power supply circuit);*
> *Controller- PID (This is not necessarily the answer, there are many control strategies that can be applied).*

Comment: In principle, the knowledge engineer agrees with this response. However, as stated in the definition of the component class (section 5.3.1), the attribute ***kind***, which is referred to by the expert in the above answer, was created in order to represent a specific property of the component without having to refer to a sub-class. It also eases the manipulation of a component object in a higher level. Moreover, this attribute is not fixed, rather it is only a default option which can be manipulated by the system based on some rules. In fact, as an example of this facility, in the present version, if the power supply circuit is changed from an unit with fixed displacement pump to a variable displacement one, this attribute reflects the change.

3- How do you consider this system would be useful for you (or your organisation)?

> *Besides the applicability in the design, I perceive that it can be applied as a didactic tool, it only requires to expand the explanation texts and, perhaps, an user manual for the student.*

Comment: as the expert is an university lecturer in undergraduate and graduate levels, working also in research related to fluid power, this response greatly boosted the confidence in a future exploitation of the tool as a didactic support, which was also clear when applying the system with the students in Lancaster, as described in the previous section.

4- In your opinion, what are the strongest point in this system?

> *One of the strongest points is the possibility of executing a design from a problem specification with little knowledge in hydraulics. It is a compact tool (few menus and subdivisions) which provides two concrete results to the designer (from the load viewpoint): 1) the hydraulic system diagram; 2) the sizing of the system.*

Comment: This feedback agrees with the previous expert’s response, i.e. the quick generation of the system diagram as well as its sizing are clearly stated as the main benefits. Considering that the two experts have distinct profiles, in terms of background and area of work, these two facilities are definitely USP (Unique Sale Points) of this prototype.

5- What are the weakest points in this system?

> *One of the aspects that should be implemented is the inclusion of behavioural models in discrete events as a form of establishing the system behaviour, i.e. to analyse the circuits as a whole.*
>
> *As it was not possible to execute the Dymola, it was difficult to comment on the continuous behaviour of the system, considering time response. I am curious about the potentiality to apply Bond Graphs on modelling hydraulic systems, for it applies basic parameters such as capacitance, resistance, etc. and not parameters obtained from experimental identification like natural frequency, etc.*

Comment: the first part of this answer, *"the inclusion of behavioural models"*, relates to a development of an agent to model the system considered, for example, a technique such as GRAFCET. The reasons for not including this type of technique were stated in chapter 4. The knowledge engineer believes that with a clear understanding of this technique, and providing that experts on it are properly available, behavioural models can be implemented. This would require the definition of new attributes (and their values) to the load and circuit objects. However, an overall analysis of the potential users is required before investing effort to develop this module, for it could demand a great deal of effort without a proper return if the market does not accept it. As mentioned in chapter 4, GRAFCET application, for example, possibly involves strong cultural aspects which should be taken into account in future projects.

The answer second part considers the dynamic modelling and possible Bond Graph application for simulating hydraulic systems. As described in section 5.5, Dymola had been previously chosen by EDC, and even for this tool there still exists work to be done to validate the circuit libraries. So far, the knowledge engineer is neither fully convinced that this is the best tool nor that Bond Graph is the most applicable technique. Rather the point to emphasise is the evident potential that the system structure has to generate dynamic models which can speed up the design process.

6- Would you be interest in collaborating with the system enhancement? If yes, how?

> *I hope that we can continue this work at LASHIP.*

Comment: as the knowledge engineer is also a lecturer at UFSC, this answer points out the potential continuity of this project. Although this project was mostly implemented in the UK, the knowledge engineer firmly understands that the system progress can be carried out at UFSC, in Brazil, for not only has this institution developed research projects and consultancy services in hydraulics, which form a considerable knowledge base, but also because the knowledge engineer has full control regarding the project, in terms of its structure, implementation issues and interaction with experts.

7- State your general comments about this system:

> *Complementing the above remarks, it is worth to mention that as a concurrent engineering tool, the technical solution had a much greater attention compared to the other aspects: safety, cost-effectiveness, etc.*
>
> *The module 'VIEW SYSTEMS/WEIGHTING TOOL' establishes a means to consider these other aspects, but it is not evident from the explanations presented along with the solutions the influence from each aspect. Maybe this was a strategy!.*

Comment: in this last item, the expert addresses the concurrent engineering issue, stating that the prototype handles the technical aspects (functional requirements) with much more detail than the others. The knowledge engineer does agree with the above remarks. A form to take into account other requirements would be to develop specific agents (cost, maintenance, etc.) and/or even to improve the circuit descriptions including attributes to store more specific explanations. This second alternative can be easily implemented while the former one involves a deeper analysis.

6.5- Final considerations on Validation

Considering the intrinsic complexity involved in validating an expert system (which can be noted when two experts in a same field disagree about a specific solution or proposition), the resource limitation in terms of personnel and time directly related to this research project and the comprehensiveness of the design area, the knowledge engineer considers that the reported feedback from expert and non-expert users concludes in a satisfactory form about the prototype robustness, expandability and modularity. This conclusion is also based on the fact that the development shows only few fundamental changes throughout the implementation, which took place at the beginning of the project. This issue as well as the future steps in the prototype enhancement are discussed in the next chapter.

Chapter Seven
Conclusion & Future Issues

The previous chapters presented the fundamental aspects related to concurrent engineering, design methodology, expert systems, object oriented techniques, fluid power system design and a functional description of the prototype system as well as points regarding its validation.

This chapter is an attempt to summarise the main conclusions drawn from this project as well as to describe potential future works as expansions or complementary researches regarding this work. In a certain sense, considering the scale, complexity and present stage of integration among the areas involved in the project (i.e. artificial intelligence, design methodology and fluid power) it seems that this chapter could be as large as, or even larger than, some of the previous chapters together. However, due to practical constraints regarding time and scope of this research project, the chapter limits itself to comment only on the main aspects of the future issues.

The chapter is divided in two major sections. First, the aspects are discussed as general conclusions about the system usefulness, potential, continuing validation and expansion. Considering that some issues related to those aspects were sufficiently presented in the previous chapter, here solely the main points only are summarised. In the second section, the future issues are highlighted.

7.1- Main Contributions

The work pointed out the main areas of usefulness of the prototype system as being:

-The prototype presented a clear means of exposing concepts regarding the application of expert system approach for the design area in general, and fluid power more specifically. As a result of its exposure to an expert public, through direct tests, visit demonstrations and presentation of its technical background at conferences and workshops[1], it was evident that the system can be applied as a way of attracting interest from the public, i.e. industries and academic institutions, for testing the system in-house as well as contributing for future developments, as described in the next sections.

-In its present stage, with some improvements regarding help on-line and an user manual, the prototype system can be applied to support teaching activities, for demonstrating the use of a methodological approach to design hydraulic systems, considering key concepts, besides the purely functional aspects, that have great impact in the life cycle of the system. Therefore, the

---

[1] As a whole, aspects of the prototype system were presented at five international conferences/workshops during 97, along with demonstrations of evolving versions.

prototype can be used to convey a concurrent engineering perspective to the students in one area in which, traditionally, the functional requirements are much more emphasised than the other aspects.

- The features introduced in chapter 4, justifying the choice for hydraulics as an area for this development, proved to be of great relevance for the project. Here, the important points of those features are presented:

a) Hydraulics demonstrated to be a very broad field of research and appropriate for this project. Developing an expert system for a much narrower field could be functionally easier, but probably the issues related to the knowledge acquisition and market feedback would be regarded as a system for a “toy domain”. In this type of system, the problem is usually a gross simplification or unrealistic adaptation of some complex real-world problem (WATERMAN,86). The distinction between real-world and toy domains is important to understand because defining an appropriate problem scope for an expert system is absolutely crucial for its success.

b) Despite of the researches carried out in hydraulics, the methodology for hydraulic system design has a well established theoretical foundation. This was verified throughout the project development, for always the knowledge engineer had plenty of reliable sources, i.e. experts and literature, to investigate and model. Therefore, the project never suffered from stagnation or even great disagreement among the knowledge sources, which sometimes can jeopardise a whole project.

c) In hydraulics, the close relationship between the system functional structure and its physical model was recognised to be a key factor to allow the application of Object-Oriented Techniques to represent the main entities involved in the design process, i.e. loads, circuits, systems and components, as described in chapters 3 and 5.

d) The comprehensiveness of hydraulics made it possible to include in the developing system, either directly modelled or through guidelines, aspects other than the purely functional features (this means cost, safety, maintenance, etc.) which opened a gateway for a concurrent engineering perspective and, at the same time, showed the prototype potential and expandability.

e) The close analogy among hydraulic, pneumatic and electrical systems was clearly confirmed to be an important point, for even when the prototype system was demonstrated for a public not directly related to hydraulics, remarks were made about the development of an equivalent system for the other domains. This aspect also unfolds a field for future projects.

f) Hydraulics certainly presented itself as a product oriented area (BUUR,90), where the market issues played a pertinent role, for the system development received feedback from the market agents, i.e. consultants, teachers and component suppliers.

In spite of their similarities, this project confirmed the differences between knowledge engineering (defined in chapter 3) and software engineering, which involves representing well-known and well-defined algorithmic procedures that are typically known by many individuals (GONZALEZ & DANKEL,93; HART,92).

Besides the clear distinction in terms of their definitions, another difference between knowledge and software engineering involves the nature and quantity of the knowledge. While the nature and quantity of knowledge required to solve a traditional algorithmic problem can be estimated reasonably well, such is not the case for knowledge-based systems. Typically, the nature and quantity of problem-solving knowledge required within a knowledge-based system is not well known even by the experts themselves. This makes it difficult to predict the total effort required to develop a knowledge based system. But more important, it can also make it difficult to arrive at a suitable design in the early stages of the project. This last situation can lead to what is commonly called a **paradigm shift**, as represented in figure 7.1 (GONZALEZ & DANKEL,93). This is most common when the implementation phase is carried out without a well defined conceptualisation. A paradigm shift may place the project seriously behind schedule. A paradigm shift that takes place early in the development process, however, can be beneficial since mistakes can be corrected before great investment of time and money are put into the prior paradigm.

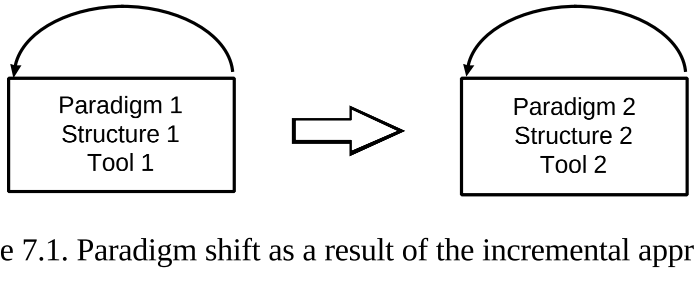


Figure 7.1. Paradigm shift as a result of the incremental approach.

Based on the definition presented in chapter 3, and as figure 7.1 shows, the incremental approach allows to utilise the feedback from the tests in the developing system to its own development. If a paradigm shift occurs, a change in the adopted models to: represent knowledge; interact with the knowledge sources; organise the system structure; or even to configure the implementation tool will be required.

In the prototype development, as mainly described in chapters 5 and 6, the **only paradigm shift** that occurred during the project took place early in the process, when the use of the Bond Graph terminology to represent the qualitative inputs was replaced by a more familiar description, and the application of a proprietary software as the main tool for communicating the system outputs was modified to use the HTML browser for displaying textual and graphical results. Both shifts occurred simultaneously as a result of exposing the system, even in a very embryonic stage, to an expert. This interaction required an open-minded attitude from the knowledge engineer, that means to be able to receive and cope with critics and present improvements based on their comments. As the progress shows, this attitude, kept throughout the system implementation, proved to be invaluable to its expansion.

Regarding the AI approach chosen for this development, the system requirements (i.e. necessity for a rapid prototyping, relevance of explanation facility, existence of a reliable implementation tool and symbolic manipulation feature), as defined in chapter 3, along with the hindsight obtained from the project, firmly supported the applicability of the Expert System approach for this research. This choice together with the selection of Object-Oriented techniques plus Rules, as a knowledge representation paradigm, and the Incremental approach as development model, proved to be the reliable bedrock upon which this project was based.

As far as the initial prototype use for developing a final system is concerned, some researches recommend that the prototype be discarded upon the completion of its evaluation with the development of the final system starting from scratch, unless *all* of the decisions made in the preliminary design stage are shown to be justified by the initial prototype (a rather unlikely scenario) (GONZALEZ & DANKEL,93). Based on the conclusions presented in this chapter as well as on the description of the validation process, in chapter 6, the knowledge engineer would recommend a comprehensive analysis of the future requirements to investigate how much of the experience and software structure gathered from this project can be applied to accomplish the new requirements.

Although, the prototype is not yet a complete (commercial) package, up to this stage, the knowledge engineer has not found any limitation, in terms of knowledge representation techniques and the elicitation approach used in this project, which could make it inappropriate to apply the prototype as a basis for expansions. In order to support this claim and to give a more detailed perspective in terms of future projects, the next section sets out aspects related to these potential areas for expansion.

7.2- Future Issues

As can be concluded from this project, there are many areas for complementary researches. In this section, a separation is made among these potential projects. First, it presents the expansions considering a direct increment in the present knowledge base, it means without needing a great change in the system structure. Second, projects involving greater changes or increments in the knowledge base are discussed, including some possible paths to follow based on the experience obtained from the prototype. These comments also introduce issues related to the system application, prototype or final package, in an industrial environment.

7.2.1- Direct Increment of the current knowledge base

Although, it is known from the literature that the increment of a knowledge base is far from a simple process, for sometimes a single added rule can greatly enhance the system complexity, based on the experience gathered in this project, this section presents possible increments in the current knowledge base. These increments, if properly implemented, will take full advantage of the present system structure, without adding a great complexity to it. This claim is mostly based on the fact that the increments presented in this section are mostly related to the addition of new attributes, rules or functions on the already implemented classes.

It should be noted that the expansions proposed in the next sections are only some options, and by no means do they reflect the full spectrum of possibilities to achieve the same objectives. Considering that the options can be seen as recommendations, that they will not be necessarily implemented by the same knowledge engineer who developed the prototype, and that some AI sources suggest a team rather than an individual to perform this activity, the next sections present the word *team* to represent the knowledge engineering group responsible for enhancing the system.

Component Specification

As a natural expansion of the present knowledge base an increment to provide a component specification module seems to be the next step to enhance the prototype system. As described in chapter 5, there is a block of rules, presented as ‘component definition’, which deals with the process of creating the component objects. Chapter 5 also presents the mechanism to allow the component sizing that provides the calculation of basic parameters (e.g. cylinders areas, motors and pump displacements, supply and return pipe line diameters, flow rates, etc.). These parameters can be used to select a component size from an industrial catalogue. However, in order to implement a more comprehensive specification process, and thus provide an adequate concurrent engineering perspective, other parameters should be considered.

As an example of specification process a flow chart, figure 4.3, is shown in section 4.4 to implement a selection of pump type. That chart describes the parameters (object attributes) required for choosing a pump type and how those parameters relate among themselves (rules) to perform the selection.

The parameters given on the chart are: efficiency, cost effectiveness, contamination resistance, wear self-compensation and size consideration. The knowledge engineering task related to this module can be defined as how to represent those parameters, and their relationships, in the current knowledge base structure for choosing the most appropriate type of pump. As can be shown, those parameters belong to a general system description, not to a specific load, therefore they should be included in the system class, or even in the designer class, since they also reflect design decisions. The inclusion of the attributes in the designer class would be a similar approach to the weighting tool implementation described in chapter 5.

In order to demonstrate how the knowledge base can be enhanced to include the information from the previous mentioned chart, next, some rule diagrams are presented. The structure used here is similar to the rule presentation in chapter 5.

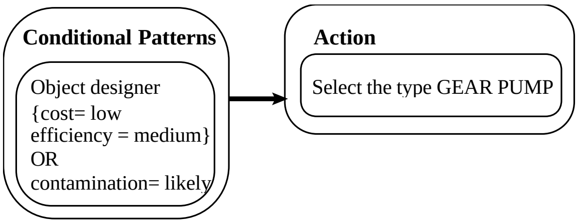


Figure 7.2- Rule diagram for selecting a gear pump type.

In figure 7.2, the conditional patterns are based on the designer object. Despite the fact that they refer to general aspects related to the hydraulic system, the definition of their values are decisions that should be taken by the designer. Usually, these are not straightforward decisions and most of the designer’s task is to gather information on which the conclusion should be drawn. Another rule to be implemented refers to the selection of a vane type pump, as follows:

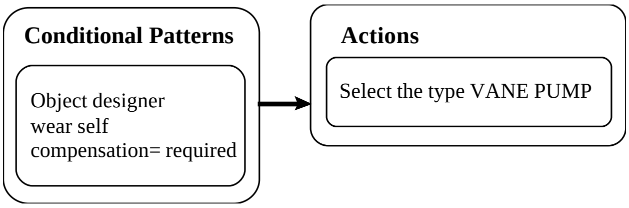


Figure 7.3- Rule diagram for selecting a vane pump type.

Although the rule presented on figure 7.3 is simpler than the previous rule, the main knowledge engineering task in this context is to define the preference among the rules. As can be seen, there is no conflict in terms of the conditional patterns between the two previous rules, i.e. an actual situation can occur when both rules are satisfied. However, their conclusions differ between a gear or a vane type option. Therefore, a measure has to be taken in order to establish the preference among these rules. If they are coded in the above order, it means that first, the gear pump rule is fired, and thus an instruction to specify the slot *kind* of pump as gear type takes place. Subsequently, the vane pump rule is fired and the same slot has the value changed to vane. This means that in case all above conditional patterns are satisfied the option vane pump takes precedence. In terms of design process, this decision implies that the necessity for a self-compensation to wear overcomes the issues on efficiency, cost and contamination resistance. Though not simple, this definition requires a trade-off among several factors which influence the life cycle of a hydraulic system, hence a comprehensive interaction between the *team* and the knowledge sources, for example system designers and pump manufacturers, should exist in order to model the most appropriate decision.

A similar analysis can apply to the rule for selecting a piston type pump, figure 7.4, for although this rule does not conflict with the gear pump rule, it does contradict the vane pump rule.

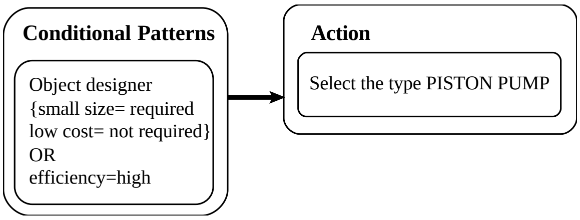


Figure 7.4- Rule diagram for selecting a piston pump type.

Even based on a preliminary analysis of the three above mentioned rules, besides the necessary trade-off to decide on the preference, another important issue rises, this is related to uncertainty. In other words, what do the qualitative terms (such as small size, low cost, high efficiency, likely contamination, etc.) really mean?

The way uncertainty is dealt with in Expert Systems greatly varies from each domain application. For example, two systems equally recognised to be successful expert systems in their own areas, R1 and MYCIN, treat this issue quite differently (RICH & KNIGHT,91). R1 is a commercially used program that configures DEC VAX systems, while MYCIN is a system to provide medical diagnosis, developed by Stanford University, in the early 70’s.

In MYCIN, the rules are implemented using the Bayesian approach for uncertainty management. This method is based on probability theory, it models that if a condition X is true, then it concludes that Y exists with a specific probability, called certainty factor or CF (GONZALEZ & DANKEL,93).

However, unlike MYCIN, the rules in R1 contain no numeric measures of certainty. In the task domain with which R1 deals, it is possible to state exactly the correct thing to be done in each particular set of circumstances (although it may require a relatively complex set of antecedents to do so). One reason for this is that there exists a good deal of human expertise in this area. Another is that since R1 is doing a design task (in contrast to the diagnosis task performed by MYCIN), it is not necessary to consider all possible alternatives; one good is enough. As a result, probabilistic information is not necessary in R1 (RICH & KNIGHT,91).

Incidentally, despite of the fact that the prototype system had no influence from the structure of R1, it follows quite a similar approach as far as uncertainty management is concerned. This means, in the prototype no probabilistic information was modelled. It is interesting to note the similarity, when comparing to the diagnosis task of MYCIN, between the task of R1, configuring computers, and the prototype main objective, i.e. to support the design of hydraulic systems. Therefore, not surprisingly the prototype applies an equivalent approach for uncertainty. Moreover, although in the above quotation mentioned “doing a design task…, it is not necessary to consider all possible alternatives; one good is enough” the prototype does offer a set of feasible alternatives for the designer.

Based on the above points, it is recommended that a facility to explain the meaning of qualitative terms (such as: low, high, small, etc.) in the context of the design is used rather than to represent those concepts through certainty factors. This approach seems much more logical than asking the designer points related to probability to define the qualitative terms. The explanation facility may have a help on-line with design examples which can greatly enhance the designer’s understanding about the decision making process, thus it will be also part of the educational process. Certainly, with this approach there still is a deal of subjectivity, but this aspect is considered in a more methodological form. In fact, the knowledge engineer believes that a certain degree of subjectivity will always exist in the design task.

Certainly, this brief commentary does not conclude about the selection process of all types of components. In fact, this task can be shown to be as complex as to obtain the preliminary design of a hydraulic system, for each component type has its own features to be taken into account in the selection process. Therefore, rather than trying to cover all components, which is beyond this research project, next, an attempt to determine a methodological approach for the team to accomplish this activity is given.

■ Define the most relevant component types to start the implementation. It is more likely that not all types will be implemented at once. In fact, it is better to break down the implementation in, for example, three groups of components. In this way, the learning curve obtained from the experience in the first group, in terms of development phases (defined in chapter 3), will greatly enhance the process in the next groups. Obviously, the definition of the number of groups, as well as the size of each group, depends on the resources available for the whole task, i.e. team size, its experience in knowledge engineering and domain, the quality of the interaction among the team members, as well as their interaction with the knowledge sources.

■ For each component type, draw a flow chart of the selection process, similar to figure 4.3, determining the relevant attributes and their relationships. This definition depends on the system scope, i.e. preliminary or detailed design, on the priorities established by the users, and on the design methodology applied. In other words, the expert system will reflect a concurrent engineering perspective or a more traditional sequential engineering approach. In order to perform a proper definition, it is fundamental to analyse how much of each approach (concurrent or sequential engineering) has been used by the experts in their activities, for this will be echoed in the flow charts they will present. With this step, the knowledge granularity, defined in chapter 3, that will be handled by the expert system is determined.

■ Implement the rules for selecting each type, considering the possible conflicts among the rules, and dealing with the aspects related to uncertainty that will be put before the user in order to perform the selection process. In other words, try to develop the knowledge base concurrently with the user interface.

- Validate the block of rules for each component type defined in a group before starting the process in the next group. The validation should involve not only the tests by experts, to prove the system robustness, but also tests by non-expert users, to evaluate the system acceptability, for these will more likely be the actual users.

Even this methodological approach does not guarantee the system success, but rather it will increase the likelihood of obtaining a robust, modular and expandable system. Furthermore, a constant and fruitful interaction between the team and knowledge sources will definitely be the most important aspect to develop the system.

7.2.2- Greater Changes in the Knowledge base

The next point to be discussed as a possible knowledge base increment refers to the development of agents related to different aspects of the life cycle concept, more specifically to maintenance and cost.

A computational tool to support the maintenance analysis could be developed using various combinations of approaches. For example, one possibility would be to expand the guidelines for each applicable object represented in the expert system, i.e. hydraulic circuit and components. These guidelines would help the designer to find the fault in parts of the hydraulic system as well as to correct them. The implementation of this approach is quite straightforward in the current structure, for the computational framework already models the necessary objects, this means that it is only required to include more attributes as specific guidelines, rules as mechanisms to define them, and message handlers to present them in a proper way to the user. Conceptually, this procedure is very similar to the one previously presented in the component specification section.

Another possible perspective for tackling the maintenance aspect is to provide a module in which the designer may question the expert system in terms of possible failures, their sources, causes and potential solutions. As illustration a conceivable structure for dealing with troubleshooting type problems in hydraulic systems is given. This structure has been implemented as a module using a standard CLIPS input/output (i.e. without graphical interface), however, due to time and resource limits, it has not been integrated with the prototype as described in chapter 5. Despite this limitation, it is believed that in the same way the fluid selection module was easily integrated in the prototype, this module can also be properly merged with the prototype.

The idea in this module is that during the design process the user of a support tool can have an easy access to information regarding potential problems in the hydraulic system. Thus if the design is carried out with some Concurrent Engineering perspective, those problems will be more thoroughly dealt with and therefore the life cycle of the system will be improved. In order to accomplish this purpose, it is necessary to model the main entities involved in a fault and their attributes. The implemented module maps a well known structure gathered from a component supplier manual (GÖTZ,84) and this structure is partially presented in table 7.1.

Table 7.1. Relationship among sources, effects and causes (GÖTZ,84).

Effect\Source	**pump**	**pressure-valve**	**flow-valve**	**...**
excessive noise	-turning too fast -maximum pressure exceeded -control system oscillating ...	-valve chatter due to dirt on seat , valve worn -unsuitable characteristic curve ...	-control system oscillating -valve oscillates and excites other control elements ..	...
insufficient power and torque at the actuators	-unsuitable type -internal leakage due to wear -end of control pressure too low ...	-operating pressure too low -internal leakage due to wear -setting range too low	-excessive pressure losses - false setting - unsuitable type ...	...
excessive operating temperature	-reduction in efficiency due to wear - rotational speed and/or delivery excessive ...	-constant delivery flow too high -pressure setting too high ...	-through-flow set too low , excessive pump delivery through relief-valve ...	...
...	....	....	....	....

As mentioned before, the description presented here, as implemented, represents only one alternative to tackle the problem, and by no means is it the best or unique solution. The reference given above was taken as just an example for implementation, and no comparative analysis with other supplier catalogues was done. However, the applied methodology should be sufficiently generic to implement information from other sources. Table 7.1 shows the relationships regarding components, potential failure effects and possible causes relating each effect to a component type.

As previously mentioned, table 7.1 presents only part of a catalogue table. In fact, the source table contains more than 240 possible faults, relating nine effects and eleven sources. In order to represent this information in a CLIPS format, a template/fact structure was adopted (GIARRATANO & RILEY,94) which relates a fault to its effect as follows:

(fault (*source* x) (*cause* “cause-description”) (*code* y)) (*effect* (*description* "a set of string to describe the problem") (*code* y1))

Whereby, *source* (table column) means one element in the hydraulic system (for example: pump, cylinder, motor, mechanical-drive, fluid, etc.) and *effect* (table row) represents a specific type of problem. The *cause* and *effect* descriptions are input of strings without size limitation. *Code* is an integer, that permits to relate an effect to a source, in other words, to identify a possible cause. The slot *code* also permits to increase the knowledge base, including other effect-source-cause relationships. The module provides the user with the following options:

1) Describe the effect(s), with or without the source, and the system presents the probable causes.
2) Define the source(s) and the system will present the potential problems.
3) Type key words to search if they can be potential causes for trouble.
4) Increase the knowledge base.

In some extent, the approach adopted in this module is similar to the one applied in the fluid selection module. This means, a file is written to represent a specific table as a fact list, with the above fault and effect statements for the whole table, according to CLIPS syntax. During the module execution, this file is dynamically loaded and thus used to fire the rules. The facility to dynamically load a fact base increases the computational efficiency since the size of the fact list is kept as smaller as possible during the execution. Figure 7.5 demonstrates the relationships between the rules and fact base.

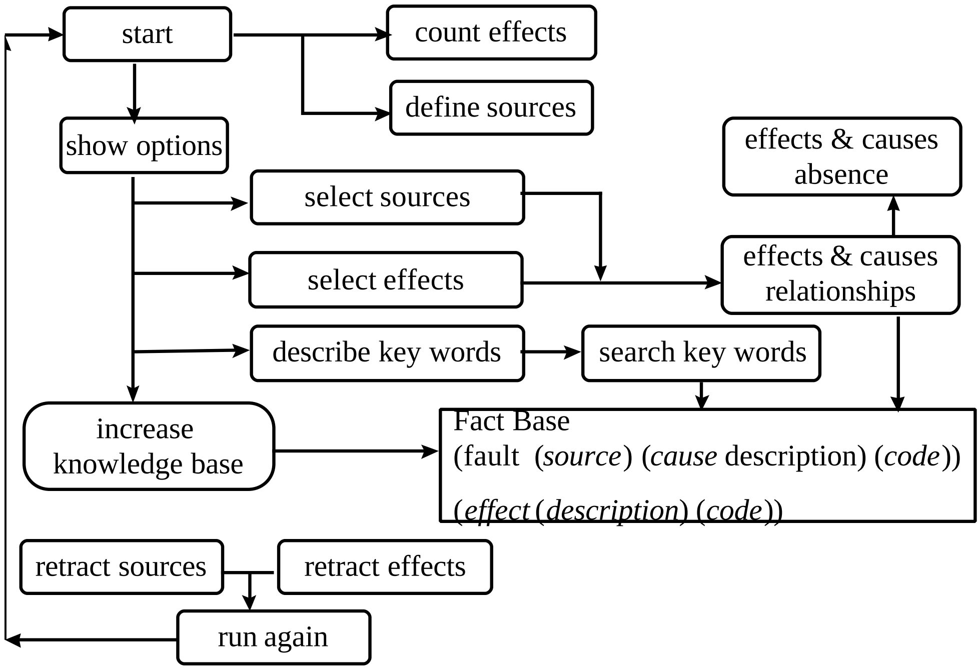


Figure 7.5. Description of Troubleshooting Module of Hydraulic System.

Figure 7.5 is an attempt to describe the information flow in an expert system using the tools for data flow diagram. As mentioned before, an expert system is ***not*** a conventional program, that is the information flow of the system is not controlled by a statement order, rather the rules execution is controlled by the Inference Engine (GIARRATANO & RILEY,94). This is the reason why the block of three rules at the bottom appears to be disconnected from the rest. This facilitates the understanding of the figure (since this block follows the execution of either four options), also it better represents the module structure, for this block of rules is functionally separated from the previous rules, i.e. while the four options relate to different forms to approach the problem, the last three rules (*retract sources, retract effects* and *run again*) deal with the preparation for allowing another search.

The rules presented as count effect and define sources on figure 7.5 are fired immediately after the phase start. This is necessary to allow the system to update the information provided to the user. In other words, all effects and sources defined in the fact base are presented to the user in the next rules.

For the option *select sources*, the module asks the user if any of the available effects are present, but the user is also allowed to search for all the effects related to a specific source, an equivalent procedure takes place with the option *select effects*.

It should be noted that, in an expert system for diagnosis task, which defines a problem from the effect perspective in order to search for a possible sources, implies the implementation of backward chaining, defined in chapter 3. Hence with the approach adopted in this module, this chaining process is emulated, even though in CLIPS backward chaining is not directly supported, compared to the available structure for forward chaining (GIARRATANO & RILEY,94).

In order to describe the module functionality, table 7.2 presents the description of the *conditions* and *main actions* for some rules in the knowledge base.

Table 7.2. Description of some rules of knowledge base

Rule	Conditions	Main Actions
Start	No condition, it fires at first	Asserts: (show options)
show options	(show options)	Asserts: (describe effects) or (describe sources) or (key words) or call: (increase knowledge base)
describe effects	(describe effects)	Asserts: (troubles description) and (sources all) or (sources some)
describe sources	(describe sources)	Asserts: (sources description) and (troubles all) or (troubles some)
key words	(key words)	Asserts: (key words some)
troubles-causes relations	(troubles description) (sources description) (fault (source) (cause) (code)) [fact base] (effect (description) (code)) [fact base]	Asserts: (trouble source defined) Presents explanation of relationship trouble-cause, for each specific source found. Asserts: (run again)
troubles-causes absence	(sources description) (troubles description) (not (trouble source defined))	Presents an explanation of the absence. Asserts: (run again)
search key words	(key words some)	Presents explanations for each key word or set of key words found, presenting source-cause-trouble related accordingly.
retract sources	(sources description)	Removes the fact (sources description) from the fact-list.
retract effects	(troubles description)	Removes the fact (troubles description) from the fact-list.
run again	(run again)	Asserts: (show options) Or presents an Exit message.

In table 7.2, the action *Asserts* means to add a fact to the fact-list, while *Retracts* relates to remove a fact from the fact-list. This process relates to the Inference Engine which defines the sequence of rules execution. The fact-list represents all facts that can be used as conditions to execute the rules. As presented in chapter 3, the Inference Engine controls that a specific set of facts is used only once for each rule.

The fourth option in figure 7.5, *"increase knowledge base",* allows the user to add more effects in the fact base, which can be related to the already defined sources indicating new causes, or even to include other sources of potential faults. In some extent, this option permits the expert system to acquire knowledge in a direct form. It provides the user with an input facility without requiring knowledge about CLIPS syntax. Once the input of each source-cause-effect relationship is specified, the system formats it according to the template previously presented, counts the effects in the fact base and includes the new relationship in the corresponding file. Therefore, if a new search is executed the added fact will be used in the quest and the knowledge base is increased. Considering that learning can be defined as: the improvement in the performance of a specific task (intellectual or physical) after previous exposure to that task or a related one (GONZALEZ & DANKEL,93), the option *increase knowledge base* would imply a certain degree of learning capacity, which is, of course, very limited, for the system does not check about inconsistency in the added knowledge.

Although this is mentioned here, proper learning, which includes understanding, was not addressed in the project. In fact, to develop an expert system with learning features in the task of design remains still to be done, for learning advanced concepts, such as those related to deep knowledge (defined in chapter 3) applied in a design task, always depends on the mastery of more simplistic ideas (GONZALEZ & DANKEL,93). Hence, it can be suggested that for an expert system to be able to actually learn about design in a domain, it needs to manipulate more simplistic concepts in that domain. It is important to mention that despite the concepts in this section have been implemented in CLIPS they are generic. In other words, other shells provide facilities to represent the same concepts, a comprehensive list of tools for developing expert system is presented in (WATERMAN,86).

The above option to consider a trouble shooting problem represents the knowledge in a very simplistic form. Though still useful, it also needs a better analysis, in terms of knowledge representation, and evaluation, considering the user's acceptability as well.

Another potential area for increments in the knowledge base is to model fault finding charts using functional block diagrams and troubleshooting charts (PINCHES & ASHBY,89). In this method, the complete hydraulic system should be broken down into sections which can be considered individually, for example an actuator or groups of actuators and the associated control valves. This reference presents that there is considerable work involved in designing and drawing up the appropriate charts, so this method may only be viable if a number of similar machines are involved. In the fault finding charts, the maintenance engineer is required to investigate a problem through a set of simple questions, type yes/no answers, considering each component.

In the above remark, two aspects are of greater interest. First, a clear analysis of the method functional structure, in which a hydraulic system is broken down into sections that are considered individually, makes it evident that this structure maps exactly the functional representation of a hydraulic system as modelled in the prototype, i.e. system, circuit and component objects, as defined in chapter 5. Therefore, the current prototype structure should be applicable for modelling such approach. The second aspect is that, although the method is recommended, the reference suggests that because "there is considerable work involved in designing and drawing up the appropriate charts, the method may only be viable if a number of similar machines are involved", hence it is indisputable that there exists a potential demand for applying computational tools in this task.

Because of the AI development history, which had at the beginning a great influence of diagnosis type problem (EMYCIN had a historical impact in the evolution of expert systems) (WATERMAN,86), it is very possible that research or commercial software systems have been developed for tacking troubleshooting in hydraulic systems, although the knowledge engineer had not carried out an investigation in this field. However, the point to be emphasised here, as a great potential advantage of the maintenance module increment in the prototype knowledge base, is that in case this module is developed to integrate with the prototype, the resulting support tool would allow maintenance to be considered as a requirement during the conceptual design phase, and not as a stand alone issue.

Cost Agent Aspects

In a computational environment aiming a concurrent engineering perspective, it is clear that the cost aspect should be given adequate attention. In the prototype, an attempt was made to consider cost effectiveness as one of features in a comparative analysis among the design alternatives generated by the system. Although this approach was accepted by the general public, i.e. expert and non-expert users (as described in chapter 6) or engineers to whom the system was demonstrated, the cost aspect may deserve a greater attention. For to predict the overall cost of a system in its conceptual design stage must prove to be an invaluable tool.

However, unlike the previous comments on the maintenance issues, where most of information is available in public sources (text books and supplier catalogues), to implement a module for dealing with cost will greatly depend upon the team interaction with industries, system designers and component suppliers.

Although there has been an interest from industrial agents throughout the prototype development, the knowledge engineer concludes that this interest does not necessarily mean that the industrial agents had been keen to provide key information to the development, rather it manifests the prototype potential to enhance the design activity in an industrial environment.

As a common case in any computational system for industrial application, in the cost analysis module, it seems that either one industry directly demands this specific system or the team has to prove, most likely through a workable prototype, the potential benefits of the tool to the industries. Unless the team is able to convince at least one industrial partner to invest the time of its experts, and possibly some financial support, to develop a cost analysis module, the knowledge engineer considers that it is not worth to embark in this endeavour.

The above claim is supported by the fact that, mainly in the cost area, the team will need access to the records of the organisation and these may contain commercially sensitive information. They may also contain embarrassing information with respect to the past competence of the client organisation and its personnel. Thus the client organisation may be reluctant to release what may be the most informative records and the team must possess a considerable degree of tact when dealing with such cases. The team will also need access to the current tasks and operations of the client organisations and these too are likely to be sensitive. Perhaps the major expense to the client organisation is the disruption that is likely to be caused to their current operations by the loss, even for limited period of time, of their expensive and valuable domain expert or experts (DIAPER,90). Thus, if a successful expert system is to be developed then the client organisation must be a willing partner in the enterprise.

7.3- Final Remarks

This thesis covered the main phases regarding a knowledge based system (i.e. conceptualisation, implementation and validation). The project was based on a prototype development to demonstrate the feasibility of integrating the following aspects:

- Expert System as a specific approach;

- Object-Oriented Techniques and Rules as knowledge modelling paradigm;
- Hydraulics as knowledge domain;
- Incremental Approach as a development model.

The thesis presented the functional framework and documented in a chronological description the development process of an expert system prototype. The prototype proved to be sufficiently reliable to explore the potential for a computer aided design environment for the conceptual stage, focusing on a concurrent engineering perspective. Albeit limited in its scope, the project was implemented to consider, as much as possible, robustness, expandability and modularity throughout its elaboration. As can be concluded from the system validation, the project had an intensive participation of users in its implementation that confirmed to be also a key factor in the whole development.

As mentioned at the beginning of this chapter, due to the scale, complexity and present stage of integration among the areas involved in the prototype, there still exist many researches to be carried out. Hence, this text, regardless of its extent, would not be able to comment on all guidelines about possible alternatives to undertake these researches.

Besides the already mentioned expansions, it is worth to register other projects that can directly or indirectly benefit from the prototype application. A brief list of projects may include:

- Implementation and validation of circuit libraries of dynamic models;
- Equivalent prototypes for other domains;
- Full Internet version;
- Agents for CAD packages;
- Inclusion of behavioural models for representing a hydraulic system;
- Training for direct users as well as for those whose activities will be affected by the system use;
- Modelling of other participants in the design process, e.g. suppliers and clients, in a similar form as the designer was represented.

Some of these areas have been discussed in the chapters 5 and 6, while others, such as the last two items, transcend the technical scope of this research. In order to accomplish these items, it is paramount to involve, in a much more intense form, members with experience in training and management, for these projects clearly embrace several human aspects.

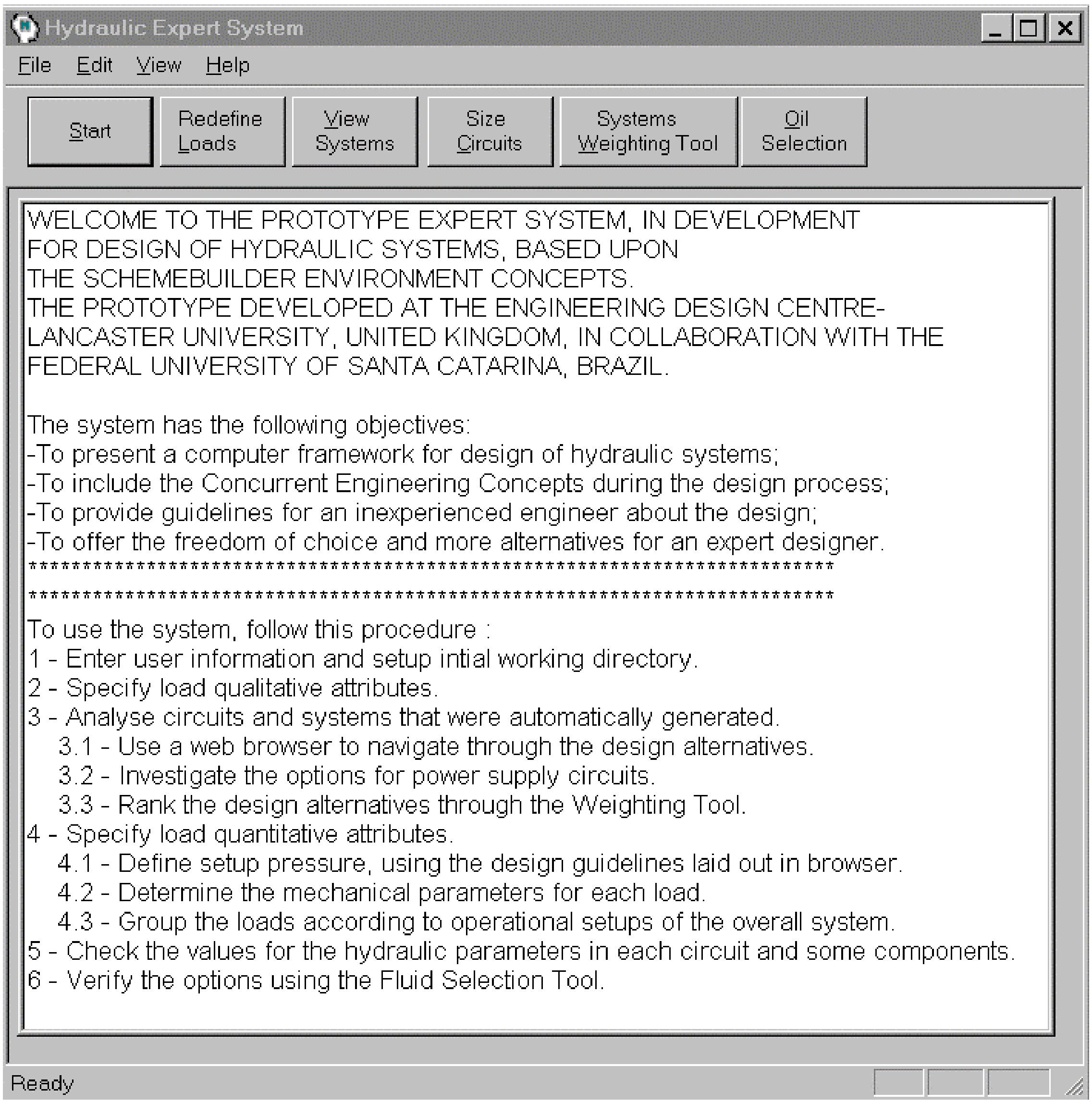
The attention given to the different aspects of system development varies according to the circumstances and the objectives of the power agents in any given project. Very often a key individual with enthusiasm and drive is to be found behind the successful scheme. If a system is to become more than someone’s pet project and instead perform a productive commercial function, it goes without saying that it should be used (EDMONDS et al.,90). Thus, based on the feedback obtained from the prototype, the knowledge engineer has a firm conviction that this system clearly demonstrated its capacity of performing a productive commercial function.

Although the present project was focused on the technical aspects, based on the experience gathered from it, the knowledge engineer believes that the greatest challenges for obtaining an useful expert system reside on the human issues, for they far outweighed the technical aspects found in the project. Hence, as a final remark, it is important to emphasise that, despite of the great effort given to technical points in the general AI literature, the development of an expert system is beyond the task of coding some instructions in a computer language. Rather it embraces other aspects, such as economical, psychological, ethical (BELOHLAV et al.,97) and cultural, that deserve equal, or even greater, attention for a successful accomplishment.

Appendix One
Prototype Graphical Presentation

This appendix sets out the main steps by which a user will employ the prototype. To a certain extent, this is a basis for an on-line help facility. It is also a direct complement of chapter 5, where the system underlining concepts were discussed.

As presented in the previous chapters, the prototype is an Expert System which assists the designer in rapidly creating design alternatives for Hydraulic Systems. It also supports a noviciate engineer to understand some aspects on the design of a hydraulic system, considering a methodological approach with some concurrent engineering perspective. Figure A1.1 presents the first window available to the user when applying the prototype.

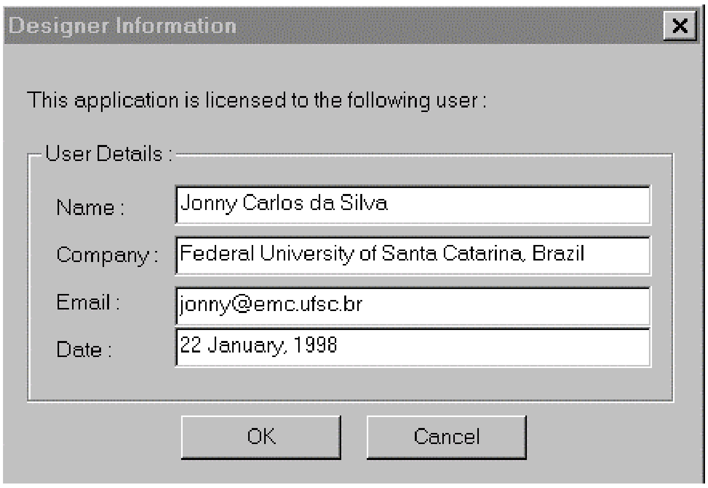


WELCOME TO THE PROTOTYPE EXPERT SYSTEM, IN DEVELOPMENT
FOR DESIGN OF HYDRAULIC SYSTEMS, BASED UPON
THE SCHEMEBUILDER ENVIRONMENT CONCEPTS.
THE PROTOTYPE DEVELOPED AT THE ENGINEERING DESIGN CENTRE-
LANCASTER UNIVERSITY, UNITED KINGDOM, IN COLLABORATION WITH THE
FEDERAL UNIVERSITY OF SANTA CATARINA, BRAZIL.

The system has the following objectives:
-To present a computer framework for design of hydraulic systems;
-To include the Concurrent Engineering Concepts during the design process;
-To provide guidelines for an inexperienced engineer about the design;
-To offer the freedom of choice and more alternatives for an expert designer.
****
****
To use the system, follow this procedure :
1 - Enter user information and setup intial working directory.
2 - Specify load qualitative attributes.
3 - Analyse circuits and systems that were automatically generated.
    3.1 - Use a web browser to navigate through the design alternatives.
    3.2 - Investigate the options for power supply circuits.
    3.3 - Rank the design alternatives through the Weighting Tool.
4 - Specify load quantitative attributes.
    4.1 - Define setup pressure, using the design guidelines laid out in browser.
    4.2 - Determine the mechanical parameters for each load.
    4.3 - Group the loads according to operational setups of the overall system.
5 - Check the values for the hydraulic parameters in each circuit and some components.
6 - Verify the options using the Fluid Selection Tool.

Figure A1.1- Prototype Introduction window.

The previous figure is an attempt to summarise the main attributes of the prototype as well as to provide a guideline on the use of the prototype. Thus, one of the attributes of the prototype is to include concurrent engineering concepts during the design process. Therefore, it is important to offer a means of documentation in the process. Hence, the next step in the prototype, figure A1.2, is an input window regarding user information.



This application is licensed to the following user :

User Details :

Name : Jonny Carlos da Silva

Company : Federal University of Santa Catarina, Brazil

Email : jonny@emc.ufsc.br

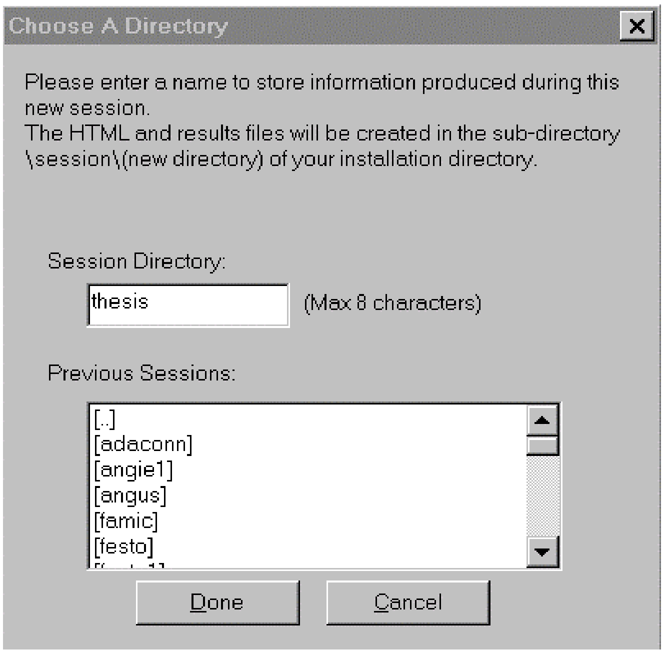
Date : 22 January, 1998

Figure A1.2- User information window.

The attributes displayed on this window are defined by default, but they can be changed (and in a future version improved). These features are handled as attributes of the designer object and are displayed in the result files together with the characteristics of the loads later specified. An attention should be given to the inclusion of email, for with the use of an Internet browser as the main means of displaying the results, there is a facility, even in the current version, to utilise email to transfer the files generated by the prototype to other people involved in the design process. The files can be transferred, for example, to members in the same company (Intranet) or even to different participants, e.g. suppliers or clients. This easy access among the members in a design process is one of the main aspects in a concurrent engineering environment.

Other aspect in this section is that even the default attributes can be modified by the developer (knowledge engineer) before sending the prototype to other users, in fact that was the way the tests were undertaken. Therefore through a similar approach, information regarding user licensing can be defined. Furthermore, the fact that the user finds his/her data shown as a main entrance in the prototype might bring him/her closer to the system application.

Regarding the documentation of the design, in each session of the prototype application, it is important to offer a means to store the results generated by the system so as to allow, for example, a comparison with other sessions. Hence a directory for each session can be created or chosen from the previous defined directory as shown in the next figure.



Choose A Directory

Please enter a name to store information produced during this new session.
The HTML and results files will be created in the sub-directory \session\(new directory) of your installation directory.

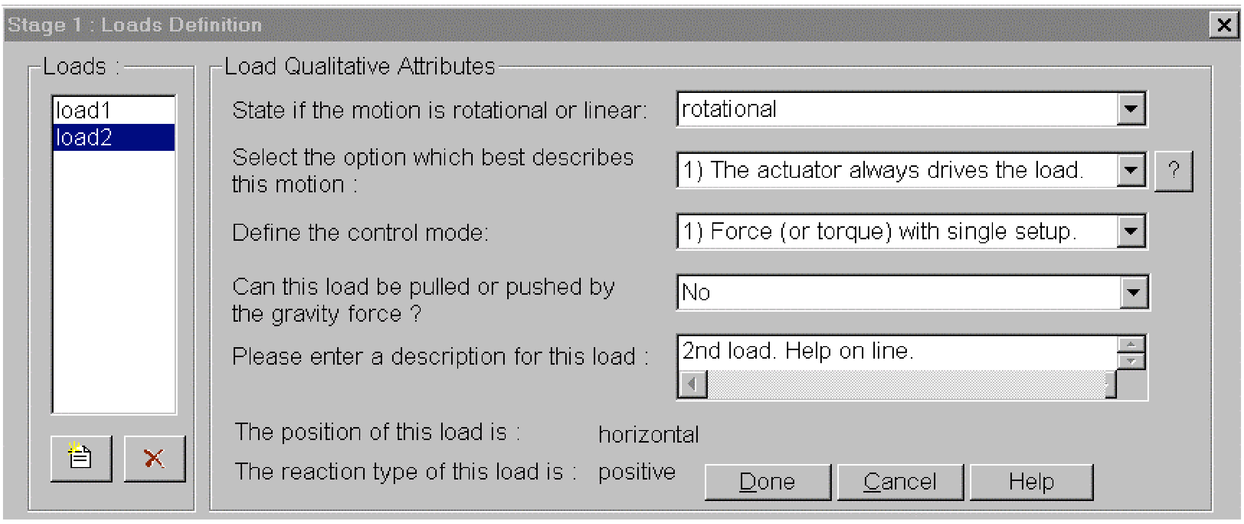
Session Directory: thesis (Max 8 characters)

Previous Sessions:
[..]
[adaconn]
[angie1]
[angus]
[famic]
[festo]

Figure A1.3- Definition of a directory for each session.

This figure shows directories previously defined in demonstrations for some companies. In fact, a practical use of this facility was evident at the Pittsburgh show (mentioned in the previous chapters), for on that occasion several demonstrations took place, and the manner in which the prototype stores information proved to be invaluable to retrieve all contacts made during the show.

One aspect the prototype provides is a focus on a methodological approach to design hydraulic systems. Therefore, instead of specifying the whole machine system at once (defining qualitative and quantitative attributes), the prototype guides the user to determine first the qualitative features of each load. In other words, the user is advised to concentrate on the preliminary design of the system, before dealing directly with details. As discussed in chapter 4 the load specification is a basis to design an actuation system, be it hydraulic, pneumatic or electro-mechanical. The next figure presents the input window corresponding to the definition of the qualitative attributes for each load.



State if the motion is rotational or linear: rotational

Select the option which best describes this motion : 1) The actuator always drives the load.

Define the control mode: 1) Force (or torque) with single setup.

Can this load be pulled or pushed by the gravity force ? No

Please enter a description for this load : 2nd load. Help on line.

The position of this load is : horizontal

The reaction type of this load is : positive

Figure A1.4- Qualitative attributes input.

Although the attributes displayed in the above figure are very few, only four functional attributes plus a general description, as can be seen through the prototype development, these attributes and their values embrace a satisfactory scope in terms of possibility for the design. In case of enhancements in the knowledge base, new attributes will be added to this section or new values included in the already implemented attributes. Next follows some comments on the attributes with the values:

-State if the motion is rotational or linear. Although this option was defined as a binary form (rotational or linear), it can be expanded to include for example oscillatory behaviour.

-The feature “Select the option which best describes this motion” has the following alternatives:

1)The actuator always drives the load.

2)The actuator can be pulled by the load.

3)The actuator must drive heavy loads.

As can be noticed, there is a button [?] beside the option. This button pops up a dialog box with this explanation:

1. The actuator (cylinder or motor) always drives the load, i.e. the load is always resistive.
2. The load can drive the actuator as it moves, for example running away load.
3. The actuator must drive heavy load, for example large mass or high frequency load.

Although this is a limited explanation, it proved to be useful since the reaction attribute demonstrated to be a decisive feature for selecting among different circuits, mainly flow control circuits.

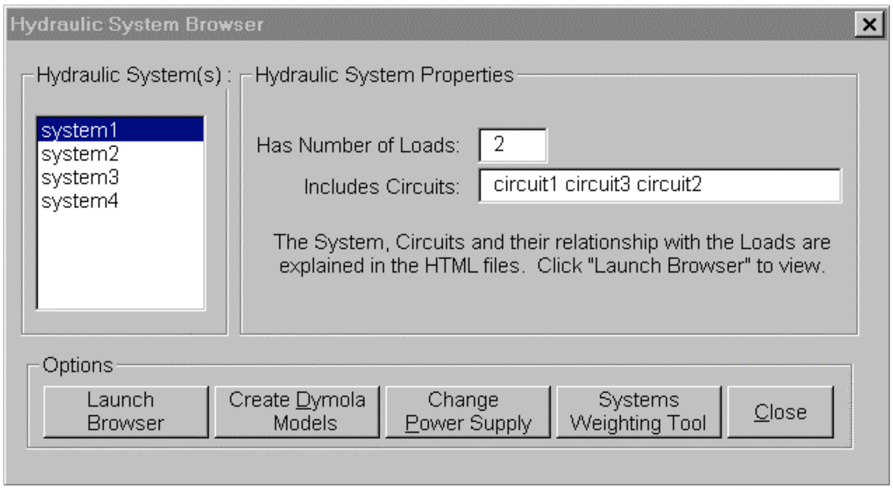
-The next point is: Define the control mode, whose options are:

1. Force (or torque) with single setup.
2. Speed (or rpm) with single setup.
3. Force (or torque) with multiple setup.
4. Speed (or rpm) with multiple setpus.
5. Force and speed combined.
6. Torque and rpm combined.
7. Position control.
8. Simple movement without requiring feedback.

The comments on these attributes were discussed in chapter 6 regarding feedback from different experts. The point to be emphasised here is that the current version can be easily improved, i.e. without modifying the knowledge base, in terms of presenting these attributes in a better way to the user.

-The next point on figure A1.4 refers to the influence of gravity on the load. As with the previous attributes, this feature can be enhanced to include more alternatives (e.g. unknown influence) and/or to improve its presentation.

-In the attribute description, the user is asked to enter a free style set of strings, which is used to identify a specific load. This attribute is also manipulated by the prototype to inform the user in case that inconsistent inputs were defined, for example: a vertical and always positive load. Once the user finishes the input for all loads, the prototype processes the information and the user is advised to check the resulting system through the window given in the next figure.



system1
system2
system3
system4

Has Number of Loads: 2

Includes Circuits: circuit1 circuit3 circuit2

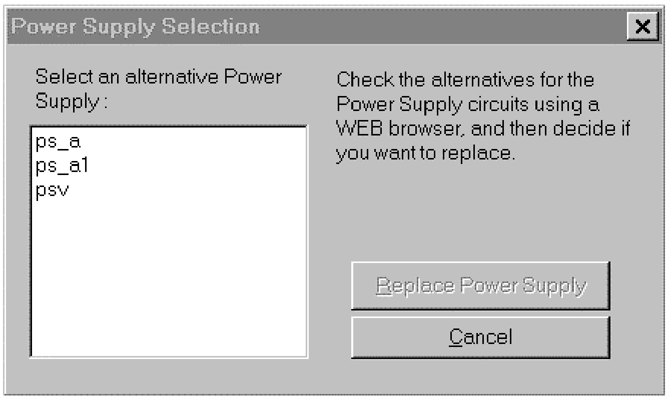
The System, Circuits and their relationship with the Loads are explained in the HTML files. Click "Launch Browser" to view.

Figure A1.5- System browser window.

The system browser window opens options to: *launch browser*, *create dymola models*, *change power supply* and *systems weighting tool*. The option ‘launch browser’ automatically runs the default Internet browser and presents an introduction window with links to navigate through the resulting system diagrams, to verify the design information (i.e. designer’s data, load specification, and circuit generation), to analyse all alternative circuit diagrams.

The button ‘create dymola model’ executes a function which generates the corresponding dymola files, e.g. figure 5.11, for all the hydraulic systems created. Comments regarding the development and application of this option were given in section 5.5.

As shown in chapters 5 and 6, the prototype directly addresses concurrent engineering aspects in the design of hydraulic systems. Among these aspects one of great importance is the option to analyse alternatives for the power supply unit. In this context, the prototype offers a set of options, defined in figure 5.14 section 5.6.1. The options are identified in the HTML files with a diagram for each functional alternative plus guidelines regarding cost, safety, etc.. In the option ‘change power supply’ the user is presented the next dialog box.



Select an alternative Power Supply :

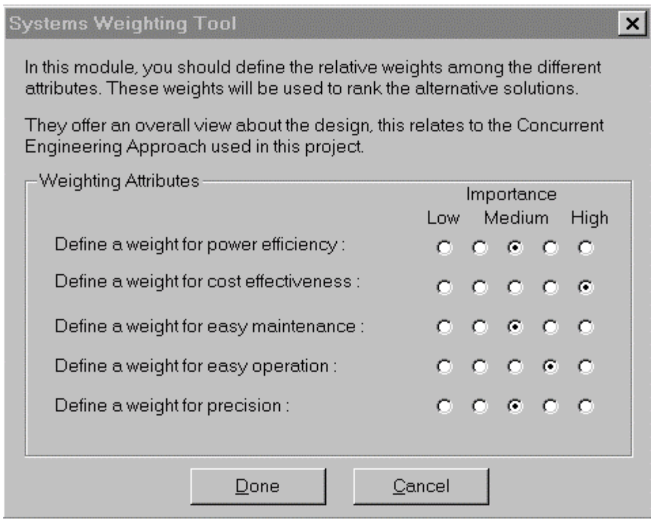
ps_a
ps_a1
psv

Check the alternatives for the Power Supply circuits using a WEB browser, and then decide if you want to replace.

Figure A1.6- Power supply selection window.

After analysing the guidelines for the alternatives in HTML files, the user selects one option to substitute and clicks the button *'Replace Power Supply'*. With this action the prototype changes the configuration of the power supply unit according to the user's choice and redefines all component lists, it also modifies all system and circuit files. The user can repeat this process as much as it is required.

The prototype thus provides flexibility in terms of design options, for the actuation and power supply circuits, and a quick redefinition of the system diagrams based on the user's decision. As the prototype offers design alternatives, it is also important to assist the user in selecting among them, for each alternative has its own characteristics which can reflect on the whole design. In order to accomplish this purpose, a weighting tool, explained in section 5.6.2, was implemented to aid the designer in this selection process. This tool is shown in next figure.



Systems Weighting Tool

In this module, you should define the relative weights among the different attributes. These weights will be used to rank the alternative solutions.

They offer an overall view about the design, this relates to the Concurrent Engineering Approach used in this project.

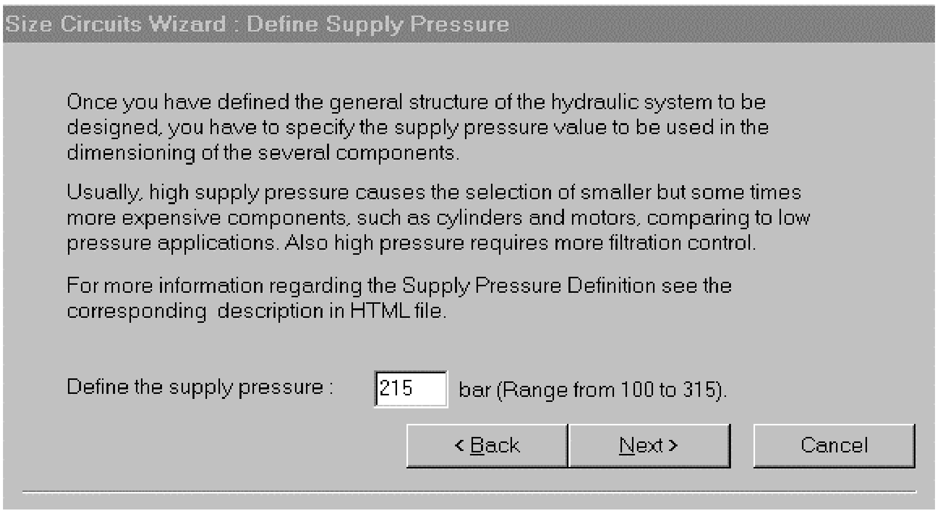
Weighting Attributes

	Importance				
	Low		Medium		High
Define a weight for power efficiency :	○	○	●	○	○
Define a weight for cost effectiveness :	○	○	○	○	●
Define a weight for easy maintenance :	○	○	●	○	○
Define a weight for easy operation :	○	○	○	●	○
Define a weight for precision :	○	○	●	○	○

Figure A1.7- Weighting facility.

As figure A1.7 depicts, only five general attributes were implemented in this facility. The purpose is to demonstrate a methodology not to satisfy all possible design criteria. The tool allows a great deal of flexibility and embraces a normal degree of fuzziness inherent in the design process. Albeit limited and based on a direct interaction with experts, this tool proved to be well accepted by both experienced and noviciate users. This module provides to the user a table with all systems (with ID's as hyperlinks) ordered according to their ranks, obtained as shown on section 5.6.2.

Once the designer has defined on the hydraulic system functional structure, i.e. system diagram and ranking criteria, then a next phase is to determine the quantitative parameters for each load. Before taking this step, the designer is advised to choose the supply pressure set-up, for this decision has a great impact in the overall design. The issues considered to implement this phase were thoroughly discussed in chapter 4, therefore here only a brief description is presented. The dialog box for defining the pressure setup is given on figure A1.8.



Size Circuits Wizard : Define Supply Pressure

Once you have defined the general structure of the hydraulic system to be designed, you have to specify the supply pressure value to be used in the dimensioning of the several components.

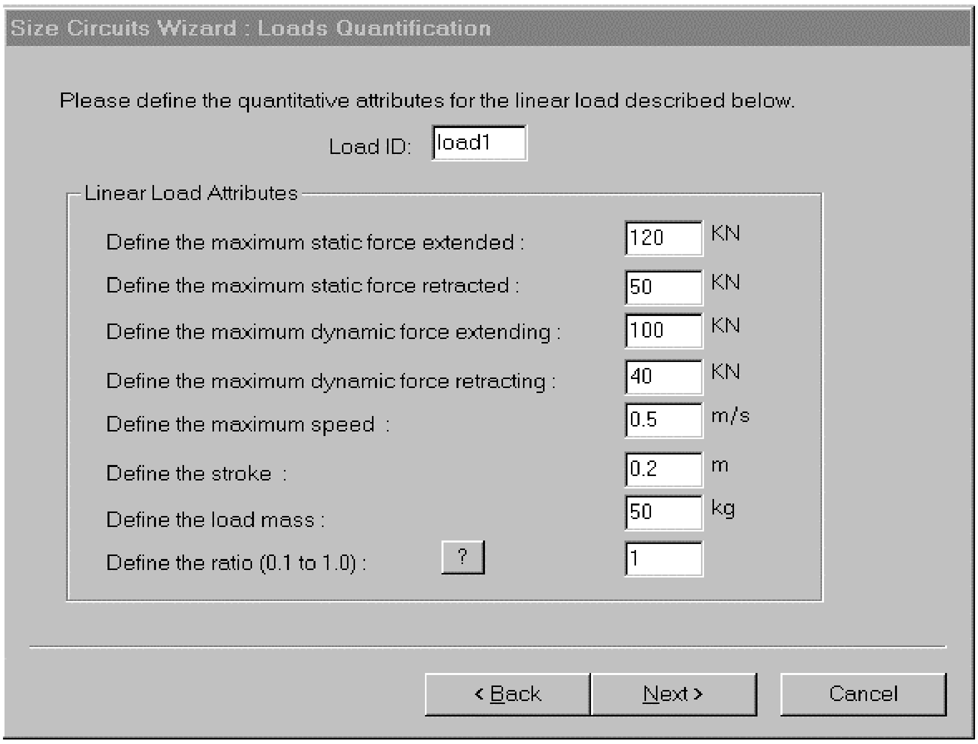
Usually, high supply pressure causes the selection of smaller but some times more expensive components, such as cylinders and motors, comparing to low pressure applications. Also high pressure requires more filtration control.

For more information regarding the Supply Pressure Definition see the corresponding description in HTML file.

Define the supply pressure : 215 bar (Range from 100 to 315).

Figure A1.8- Supply pressure definition.

The prototype offers a range between 100 and 315 bar for the supply pressure. As mentioned in chapter 6, the prototype also provides the same range in the Imperial System. Besides the brief guidelines on the previous figure, the user can access more information through the HTML files. Having decided on the supply pressure, the user is guided to specify numerical parameter per load, according to the next window.



Size Circuits Wizard : Loads Quantification

Please define the quantitative attributes for the linear load described below.

Load ID: load1

Linear Load Attributes

Define the maximum static force extended : 120 KN
Define the maximum static force retracted : 50 KN
Define the maximum dynamic force extending : 100 KN
Define the maximum dynamic force retracting : 40 KN
Define the maximum speed : 0.5 m/s
Define the stroke : 0.2 m
Define the load mass : 50 kg
Define the ratio (0.1 to 1.0) : 1

Figure A1.9- Definition of parameters for a linear load.

The first two features in the previous figure refer to output forces required when the cylinder is in a standstill position. The distinction between extended and retracted is due to the difference in terms of cylinder cap and rod areas which directly affects the output in each direction. Indeed, there are cylinders, known as double-rod cylinders, where this distinction does not exist. However, most of applications use single-rod cylinders, therefore this was the implemented choice. If a double-rod cylinder option should be added, a new load attribute must reflect it, and a change in the procedure to size a linear circuit should take this into account.

The next two attributes on the previous window, i.e. maximum dynamic forces extending and retracting, refer to the output forces required when the load is moving. Thus, these calculations consider the flow through the directional control valves with the corresponding pressure drops. With the above mentioned four attributes plus the maximum speed, the prototype defines the cylinder areas and the required flow to the actuation circuit, this procedure is explained in section 5.6.3.

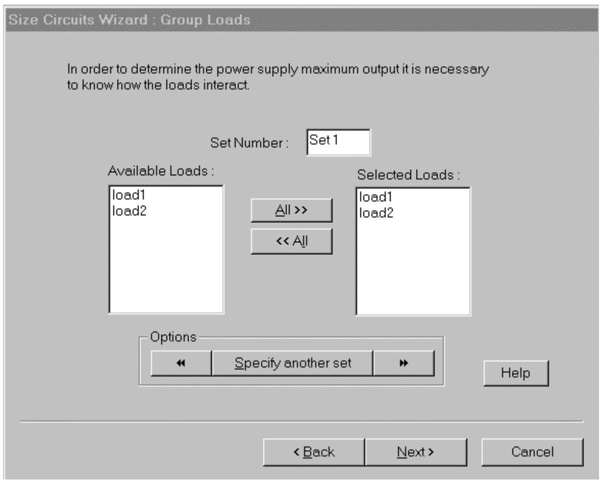
The stroke and mass attributes are used to calculate a ballpark value for the natural frequency of the corresponding circuit. In this option, a standard bulk modulus was applied which can be properly modified if conditions required.

The last option, ‘define ratio’, is an attempt to figure out the demanded power for that circuit. It offers the following explanation: this attribute means the ratio between the force (or torque) at maximum speed (or rpm) and the maximum force (or torque). The demanded hydraulic power of each circuit is the product of this ratio by its corresponding flow and the maximum pressure. This feature along with the operational set definition provides the overall demand for the power supply circuit.

As defined, the previous figure refers to a linear load. For rotational loads there is an equivalent input window, but without the distinction between extending and retracting (since a hydraulic motor is a symmetrical device) and the option stroke. The option ‘mass’ is replaced by load inertia.

It should be emphasised that all load attributes are domain independent, it means they are specified regardless of the energetic domain applied as actuation means, i.e. hydraulic, pneumatic or electro-mechanical. Moreover, this clearly distinguishes two parts of the prototype knowledge base, i.e. load definition (domain independent) and actuation system generation. Such structure undoubtedly will benefit the expansion to incorporate or develop a similar prototype for other domains.

After defining parameters for each load, the user should determine how the loads interact among themselves, for this is important to consider in order to calculate the requirements for the power supply unit. As explained in chapters 4 and 5, there are different forms to specify this interaction, and because the prototype aims the preliminary design of hydraulic system, it adopts a more simple approach, which can be seen in the next figure.



Size Circuits Wizard : Group Loads

In order to determine the power supply maximum output it is necessary to know how the loads interact.

Set Number : Set 1

Available Loads :
load1
load2

Selected Loads :
load1
load2

Figure A1.10- Group loads window.

Here, the user defines different operational sets that contain information on which loads are required to act simultaneously. In other words, the user does not include information on how the loads interact in terms of their characteristics (force and speed) throughout the cycle time.

As mentioned on the comments regarding the GRAFCET methodology, this method or others, such as state diagrams, should be implemented to describe more thoroughly the loads behaviour (SHAKERI & BRÆK,97). This will require further load attributes to map a comprehensive description of the overall system to specify the mechanical parameters, i.e. force (or torque) and speed (or rpm), through the time cycle

A reasonable perspective would be to use the current description to obtain a rough estimate on the power demand and a more specific approach, which could be implemented as detailed design option, would complement the design in case the application demands, since for some simple applications just a rough estimate of power demand might sometimes be enough for the user. Regarding the implementation of a more specific approach, it is worth to emphasise that a careful analysis of the user's profile, in terms of design practice, is to be done before committing a great deal of effort on implementing a methodology which is unknown or not well accepted by the user.

The next appendix presents results generated by the prototype, as well as more comments based on feedback from other experts in addition to the ones mentioned in section 6.7.

Appendix Two

Prototype Output and Additional Experts' Comments

A.2.1- Prototype results based on users' tests.

This appendix presents results generated by the prototype, as well as more comments based on feedback from other experts in addition to the ones mentioned in section 6.7.

As discussed in chapter 6, in order to evaluate the system performance in terms of user's acceptability, tests were conducted with non-expert users, in this case the tests involved 4th year engineering students at Lancaster University. The prototype was tested to accomplish engineering exercises defined by the students' lecturer[1]. For each group of two students, the exercises embraced the execution of a preliminary design for two applications, as whole four applications were defined. In order to prepare the students for the task, the lecturer defined the following situation:

***Scenario**: You are the Design Engineer in a contractor's organisation whose primary technical focus is in fluid power systems. You design and supply systems for a wide range of actuation applications, industrial, marine and off-road machinery. Your Technical Director has recently been given a b-release of a new "expert system" which is claimed to assist design engineers in your field. So far the system is configured for hydraulics applications only. He asks you to run a few problems through it from your experience and give an evaluation. He has been given some formats for responding to this (here they are !!) Being very experienced you are a bit sceptical, but maybe this tool will help discussions between your Company's sales engineers and your clients, especially if all the design information can be so easily exchanged over the Internet. Also, you are only too aware that companies like yours and system builders generally can't recruit people who know anything about fluid power systems, so maybe there's an educational market for software like this ?*

You rummage through your files and make a few summary sketches and these are the examples you come up with:

Here are presented two of the four examples given to the students. In each case is shown first the problem statement and then the results obtained by the students with a commentary.

Example 1- Problem Statement: A trawler winch has to exert a maximum pull of 40 kN on a cable running on to a drum of mean diameter 0.7 m, at which time the cable speed on to the drum is 0.5 ms^{-1}*. Under light loads, for reeling in the cable, the speed required is up to 2.5* ms^{-1}*. In order to pay out the cable as the trawler moves ahead, the drum must be able to rotate freely, pulled by the resistance to motion of the net in the sea. When trawling at a steady ship speed, the winch drum is locked using a band brake applied by a short stroke actuator on*

---

[1] David Dawson, Senior Lecturer at Lancaster University Engineering Department.

retract mode. This exerts a force of 20 kN at the end of a stroke of 100mm. The system maximum operating pressure is 200 bar. Based on this information, the load data are:

For the drum: T_1 = 14 kNm and ω_1= 13.641 RPM

T_2 = 2.8 kNm and ω_2= 68.205 RPM

The value for T_2 was figured out by the students considering that the mechanical power is the same in the two conditions presented above.

For the brake: F_1= 20 kN and stroke = 0.1m

Results and Comments: As can be noted, the value for brake speed is not defined, therefore an assumption had to be made for this variable, an assumption could be v=1 m/s. With the above values as input, the prototype produced the system according to figure A2.1.

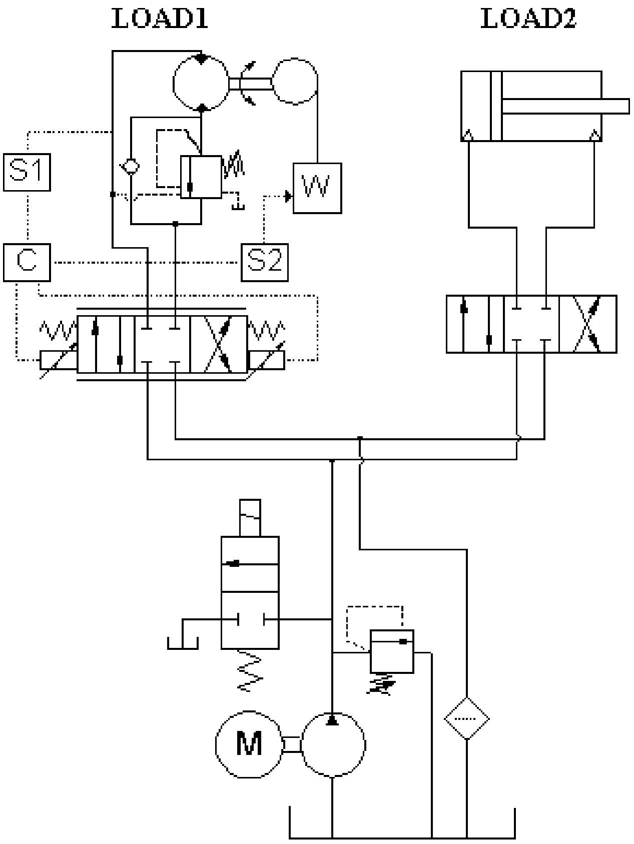


Figure A2.1- Hydraulic system diagram for the first problem.

This diagram is automatically generated by the prototype and presented in a HTML file. The definition of the above system as the unique option was based on the values that describe the whole system. These values, given on table A2.1, are determined by the system designer based on his/her analysis of the problem statement. The prototype does not check the overall consistency of this data, therefore the designer still plays a paramount role in the process, for a misinterpretation of the problem statement may lead to a different set of solutions.

Table A2.1 Load qualitative inputs

Load_id	**Mode**	**Reaction**	**Position**	**Domain**
load1	torque_rpm	negative	vertical	rotational
load2	no	positive	horizontal	linear

Table A2.1 is available to the designer in a HTML format, thus the user is able to check the values and, if necessary, revise them in another session of the prototype.

In figure A2.1, by selecting the circuit corresponding to load1, the user is guided to verify the details of this circuit, presented also in a HTML file. This file gives the circuit identification (in this case, circuit3), its graphical representation (figure A2.2), its description and its component list (table A2.2), as follows:

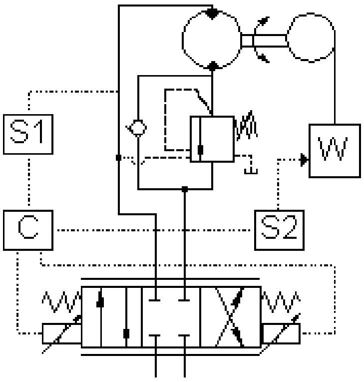


Figure A2.2- Graphical representation of a actuation circuit.

Circuit Description:

(This circuit is an Electrohydraulic Torque and RPM Control circuit. This circuit uses the valve to control both flow and pressure. This arrangement can provide advantages in installation space, cost and control accuracy compared to two independent valves. To implement such a circuit, pressure must be measured with a transducer and the amplifier controlling the valve must switch to pressure control mode at the appropriate time. Valves used to control both pressure and flow are sometimes referred to as PQ valves [Ref.1]. This circuit matches load1 due to: control mode selected torque_rpm and position vertical. Note: due to internal leakage in the motor and for safety reasons, ALWAYS use a brake in this circuit.).

As can be noticed, the above description has three parts. The first explains the general characteristics of this configuration, indicating a reference that can be used for searching more information. The second part justifies why the configuration was chosen for load1. The third part, mentioned in chapter 6, refers to a specific note regarding the safety of the load due to internal leakage in the motor.

In the process of designing a hydraulic system an important aspect to speed up the process is to define the list of components. In this context, the prototype produces for each circuit its component list whose example is given in table A2.2.

Table A2.2- Example of component list

Component	Class	Type
component9	proportional_valve	directional
component10	controller	PID
component11	sensor	pressure_transducer
component12	sensor	position_transducer
component13	motor	axial_piston
component14	pressure_control_valve	counterbalance

This list along with the graphical representation and description are placed in the same file, therefore allowing the user to understand the circuit functionality and its components. As discussed in section 6.7, in response to one of the experts, the types listed in the third column are default values of possible solutions. These values can be manipulated. The components identification, in the first column, are generated by the prototype in a sequential order, which includes the components of all circuits generated.

Although the prototype provides only one hydraulic system diagram for the above problem (in terms of actuation circuits), the user still has alternatives regarding the power supply circuit, as given in figure A2.3.

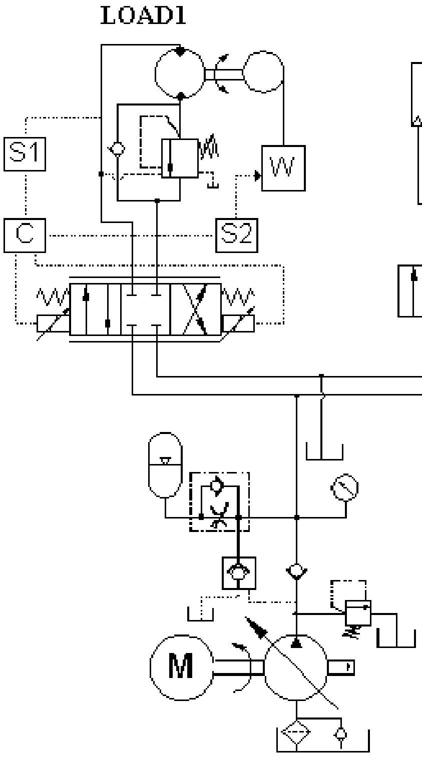


Figure A2.3- Alternative for power supply unit.

The above figure is created by the prototype, once the user selects ps_a1 in the option *"change power supply"* depicted on figure A1.6. The option identification as ps_a1 stands for "power supply with an accumulator", the number one refers to a second alternative, i.e. with a variable displacement pump. As discussed in section 5.6.1, once the option for changing the power supply unit is made, the prototype not only redefines all system diagrams, but also recalculates the component lists for all circuits. The selection of an alternative power supply is based on analysis of the available options, depicted on figure 5.14, through their corresponding HTML files which include a graphical description of each alternative plus guidelines regarding safety, cost, maintenance, etc..

The possibility of configuring the system with alternatives in a quick way definitely brings a concurrent engineering perspective into the design process, for the reconfiguration is automatically done and the designer has access to different features beyond the strictly functional requirements, such as safety, cost and maintenance.

Another aspect that reinforces the concurrent engineering approach of the prototype application is the facility to perform a rapid sizing of the circuits. In order to demonstrate this facility, once the quantitative parameters given above are entered in the prototype, the system produces the automatic sizing of all circuits and some of their components. To illustrate the result of this process and the format of outputs, figure A2.4 presents the values for the power supply circuit whose component list is given in table A2.3.

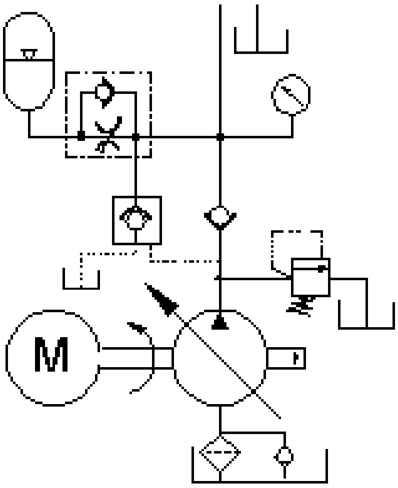


Figure A2.4- Power supply unit.

Circuit Description:
(In this case, the accumulator absorbs pressure spikes and prevents shocks, improving the life of the system, circuit1 has the POWER AND SAFETY FUNCTIONS, and it is ALWAYS created. The variable displacement pressure compensated pump makes this circuit more expensive, although it can be cost effective.)

Table A2.3- component list of power supply unit.

Component	Class	Type
component1	pump	variable_displacement
component2	accumulator	gas_charged
component3	flow_control_valve	compensated
component4	pressure_control_valve	relief
component5	valve	check_dump_piloted
component6	valve	check
component7	gauge	manometer
component8	gauge	flow_meter
component9	filter	low_pressure
component10	reservoir	small_tank
component11	prime_motor	combustion_engine

Beyond the component list, the above table also points to other files related to the components 1 and 4, presented as underlined. These files contain specific information (graphical and numerical values) to size respectively the pump and the relief valve. Although this facility was implemented for only few component types, it can be easily expanded to other types. This is a typical application of the polymorphism property (defined in chapter 3), for different types of component require specific sizing procedures according to their particular characteristics. The values generated by the prototype are presented as follows:

```
Supply pressure: 200.00 [bar], ( 2899.8115 [psi])
Flow rate: 509.97 [lpm], ( 112.1938 [GPM] )
Maximum Power: 50.00 [kW], ( 67.0486 [HP] )
The internal diameter ranges were defined based on [Ref.4] .
Supply Line: Dmin= 39.32 ,Dmax= 42.47 [mm]
Return Line: Dmin= 60.06 ,Dmax= 73.56 [mm]
```

These values are shown in the same HTML file that includes the graphical and textual descriptions. Thus, it provides an adequate access to information from different sources. The mark regarding Ref.4, shown underlined, is a hyperlink which points to a file with the reference used to calculate the values. Hence, the user has opportunity to check the knowledge sources in case more information is required.

In order to demonstrate the competence the prototype has to generate multiple alternatives for a same design problem, another example tested by the students is given as follows.

Example 2- Problem Statement: A moulding machine has four actuators(n) *which act in sequence within a cycle thus:*

Mould CloseRam(1)*:* *Speed* $1ms^{-1}$*. Force 10 kN to accelerate tool. Stroke 0.5m*
(advance) *Time: t = 0 to t = 1s*
Tool Clamp Ram(2)*:* *Speed* $0.5\ ms^{-1}$ *. Force 500 kN at end. Stroke 0.2m*
(advance) *Time: t = 1 to t = 2 s*
Injection Screw (3)*:* *Speed 300 RPM. Torque 2.0 kNm*
(motor rotate fwd) *Time: t = 2 to t = 20 s*
Injection Screw Ram: *Speed max* $1\ ms^{-1}$*. Force 60 kN. Stroke 0.5m*

(advance) (4) *Time: t = 21 to t = 25 s*
Injection Screw(3)*: Speed 350 RPM. Torque 0.5 kNm*
(motor rot. reverse) Time: t = 75 to t = 79 s
Injection Screw Ram: Speed max 1 ms^{-1}*. Force 20 kN. Stroke 0.5m*
(retract) (4) *Time: t = 75 to t = 79 s*
Tool Clamp Ram(2)*: Speed 0.5* ms^{-1} *. Force 5 kN. Stroke 0.2m*
(retract) Time: t = 80 to t = 81 s
Mould Close Ram(1)*: Speed 1* ms^{-1}*. Force 10kN to accelerate tool. Stroke 0.5m*
(retract) Time: t = 82 to t = 83 s
The system maximum operating pressure is 300 bar.

Results and Comments: This specification presents a typical sequential system for a moulding machine. For this example, if all actuators are specified for a combined control model, i.e. torque/rpm or force/speed, the prototype generates only one combination for the actuation circuits, similar to the previous example. However, if the rams are requested to control only speed (with single setup) and the motor controls speed with multiple setups, the prototype offers 27 alternative combinations for actuation circuits which can still be analysed with four different power supply units. As this second assumption was the one considered by the students, here it is presented.

In order to demonstrate the ranking option application, defined according to figure A1.7, the students decided to use the following scale for the comparative weights: cost 4; maintenance 4; precision 5; ease of operation 3 and efficiency 3. These weights are entered in a fuzzy scale. With these values, the prototype performs the ranking among alternatives and suggests "system18" as the most adequate functional option. In this application, because the machine is composed of more than three loads, the prototype offers as default a power supply unit with variable displacement pump. Nevertheless the user can still change for a fixed displacement pump.

The diagram corresponding to system18 is depicted in next figure, considering a variable pump with accumulator. According to the students' decision this power supply unit was chosen for the following reasons:

- Safety against power failure;
- Shocks due to jamming are tolerated;
- Cost over time reduced, due to reduce size and maintenance system.

Those conclusions were reached after the students studied the guidelines for each power supply circuit.

As mentioned before, the fact that the students paid attention to aspects beyond the solely functional requirements and were able to analyse different power supply units clearly demonstrates the didactic application of the prototype, providing at the same time concepts related to design methodology and hydraulic system design in a computer environment. It is important to mention that, besides the 27 system objects, this simple example involves the generation of 11 circuit objects and 51 component objects, each one of these entities with specific properties interrelated in a coherent manner required to form all combinations.

On the top of the graphical representation, figure A2.5, the prototype places information with links for each circuit. Thus, the user has chance to easily analyse the circuits individually.

Description: SYSTEM18

SYSTEM18 consists of

Load Circuits: circuit3, circuit5, circuit7, circuit11

Supply Circuit:circuit1

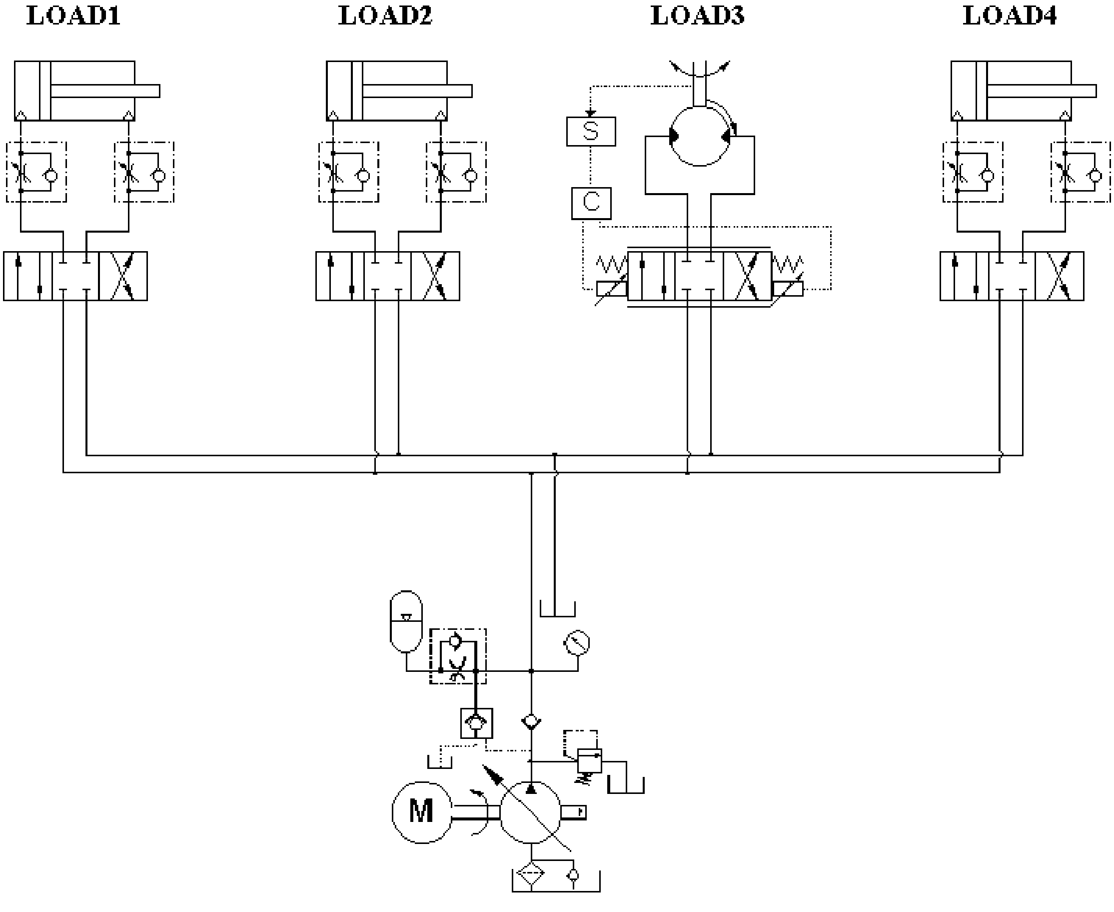


Figure A2.5- One alternative system diagram for a moulding machine.

The examples given in this section are only illustrations of the prototype potential and its output. Hence, they do not try to prove the robustness of system, for in an expert system project this aspect is much better examined through intensive tests carried by experts.

Due to general constraints regarding the extent of this text, no more examples are presented, though the students performed more tests with the prototype. However, the following section discusses additional comments on the prototype from experts' viewpoints.

A2.2- Additional Comments from Experts

In this section, complementarily to the validation comments given in chapter 6, other remarks are included regarding the prototype usefulness and potential.

Here, points raised by one of the exhibitors[2] at the Pittsburgh show, previously mentioned, are discussed. These points were formally expressed via letter addressed to the project supervisor in England. As adopted in the previous chapters, the expert's comments are given *in italic*.

The software that you and Jonny demonstrated is obviously due to a large amount of work and creativity on his part. The interface as I remember was quite easy to understand and operate. The only concern I have would be in the presentation of the product to the user. The user will undoubtedly have a very broad skill level and this will be a possible detriment to the success of the software if an individual uses the information improperly.

This engineer was presented with a full demonstration of the prototype, during the show, at that time no help on-line facility was available. He seemed to be very active in his comments, more specifically about the component sizing and selection module. This reflects his company profile, i.e. component designer. His comments also emphasised that the help facility should provide clear information to the user.

When we discussed the use of a vented versus non-vented counterbalance valve (load lowering valve), Jonny did not take into consideration the potential for misapplication of these valves. I feel that the system must be able to identify these types of problems or warn the user that there are pressure levels that may cause failure of the system.

In this comment the engineer demonstrated his deep understanding on the component application, which had not been considered into the knowledge base. However, he also showed his confidence on the system expandability, to include guideline rules for the user. The previous comment also makes evident the importance of having a knowledge engineer with some technical background in the domain application for discussing the underlining concepts with experts. Although this is not always the case, for example often a computer scientist is responsible for an expert system project in medicine, in this project the technical background proved to be a relevant aspect for the whole development.

The only other approach would be to provide a model of every type of component. I am familiar with the SPICE software used in electrical engineering and have found its capability to allow modifications of models to be very useful. As with any type of computer software, the user must understand what he is trying to do and be able to correct interpret the information he

---

[2] John Reckard, Engineer of Sun Hydraulics Corporation, Designer & Manufacturers of fluid power control valves. Florida, United States.

obtains. I believe your software will assist system designers with their work, but could be easily used inappropriately by some persons.

Although the approach suggested in the above comment has not been fully implemented, the current knowledge representation (Object-Oriented Modelling plus Rules) can support it, for in this structure each component is modelled via through a specific class. Therefore, providing that adequate attributes and values are specified for each component, they can be represented in the current structure.

Good luck with your software. I will be waiting for the announcement of its release.

Despite its embryonic stage, i.e. only about a year of implementation, the prototype raised a great attention from the market. This comment is just one of several positive feedback obtained from the prototype exposure. This feedback greatly enhanced the possibility for the development of a complete commercial package.

References

1. AHRENS & BEITZ,97- AHRENS, Gritt and BEITZ, Wolfgang. A method for the introduction of simultaneous engineering in a Product development process. ICED97, International Conference on Engineering Design, Tampere, Finland, 19-21 August 97. Vol. 3, pp 771-778.
2. ASHLEY,92- ASHLEY, Steven- *"DARPA Initiative in Concurrent Engineering"*, Mechanical Engineering, April 1992.
3. BACK,83- BACK, Nelson. Metodologia de projeto de produtos industriais. Guanabara dois, 1983.
4. BEATER,97- BEATER, P.. Modeling of Hydraulic Systems: Tutorial Manual for the Hydraulics Library for use with Dymola, 1997. Dynasim AB, Lund, Sweden.
5. BELOHLAV et al.,97- BELOHLAV, J.; DREHMER, D. and RAHO, L.. Ethical Issues of Expert Systems, The On-Line Journal of Ethics, Institute for Business and Professional Ethics, 1997, http://www.depaul.edu/ethics/.
6. BERRY & HART,90- BERRY, Dianne and HART, Anna. Expert Systems- Human Issues, Chapman and Hall, 1990.
7. BFPA,96- Guidelines to Electrohydraulic Control Systems, British Fluid Power Association, BFPA/P49, 1996.
8. BOUZEGHOUB et al., 97- BOUZEGHOUB, M.; GARDARIN, G. and VALDURIEZ, P.. Object Technology- Concepts and Methods. International Thomson Computer Press, 1997.
9. BOY,91- BOY, Guy A., *Intelligent Assistant Systems*, Academic Press Limited, Series Knowledge-Based Systems, 1991.
10. BRADLEY & SEWARD,92- BRADLEY, David A. and SEWARD, Derek W., *Artificial Intelligence and Robotic Construction Plant- the smart artisan*, IEE Computing and Control Division- Colloquium on Artificial Intelligence in Civil Engineering, pp 8/1-8/3, London, January, 1992.
11. BRACEWELL,94- BRACEWELL, R.H. and SHARPE,J.. "A Computer Aided Methodology for the Development of Conceptual Schemes for Mixed Energy-Transforming and Real-Time Information Systems". 1994 Lancaster International Workshop on Engineering Design.

12. BRACEWELL et al.,95- BRACEWELL, R.H., CHAPLIN, R.V., LANGDON, P.M., LI, M., OH, V.K., SHARPE, J.E.E. and YAN, X.T..Integrated platform for AI support of complex design part I: Rapid development of schemes from first principles. Lancaster International Workshop on Engineering Design, 1995.

13. BRAZIER & LEONARD,90- BRAZIER, David and LEONARD, Mike, "Concurrent Engineering Participating in Better Design", Mechanical Engineering, January, 1990.

14. BURROWS,93- BURROWS,C.R., Computer tools for the design of fluid power systems. Third Scandinavian International Conference on Fluid Power, 1993.

15. BUUR,90- BUUR, J. A theoretical Approach to Mechatronic Design. Institute for Engineering Design 1990, IK Publication, Technical University of Denmark.

16. CARLSON et al.,97- CARLSON -Skalak, S.; KEMSER, H.P. and ALLEN, K.. Concurrent Engineering applied to Product Design in Small Companies. ICED97, International Conference on Engineering Design, Tampere, Finland, 19-21 August 97.

17. CELLIER & ELMQVIST,93- CELLIER, F.E. and ELMQVIST, H.. Automated Formula Manipulation Supports Object-Oriented Continuous-System Modeling. IEEE Control Systems, Vol. 13, No. 2.

18. CHARNEY,91- CHARNEY, Cyril, Reducing Product Lead Time, Society of Manufacturing Engineers, 1991.

19. CLEASBY,96- CLEASBY, Ken. “How to plan an electrohydraulic system”. Fluid & Air Magazine, Nov/Dec 1996.

20. COOKE,92- COOKE, Nancy J., *Modeling Human Expertise in Expert System,* The Psychology of Expertise, Cognitive Research and Empirical AI, edited by Robert R. Hoffman, Springer-Verlag, 1992, pp 29-60.

21. CORBRIDGE,92- CORBRIDGE, Kim, "Integrated CAE/CAD/CAM for Concurrent Engineering", CICOMGRAF 92.

22. CZIULIK & DRISCOLL,97- CZIULIK, Carlos and DRISCOLL, John. Conceptual Design Evaluation based on functions metrics. ICED97, International Conference on Engineering Design, Tampere, Finland, 19-21 August 97.

23. DE NEGRI,96- DE NEGRI, Victor Juliano, Estruturação da modelagem de sistemas automáticos e sua aplicação a um banco de testes para sistemas hidráulicos. Tese de Doutorado. Curso de Pós-Graduação em Engenharia Mecânica da Universidade Federal de Santa Catarina, abril 96.
24. DIAPER,90- DIAPER, Dan. An Organizational Context for Expert System Design. Chapter 12- Expert Systems- Human Issues. Editors Dianne Berry and Anna Hart., Chapman and Hall, 1990.
25. DRISIS,97- DRISIS, Leonidas. Wide Area Design Teams- Tools for Collaboration. ICED97, International Conference on Engineering Design, Tampere, Finland, August 97, Vol.2, pp 255-260.
26. DYM & LEVITT,91- DYM, C.L. and LEVITT, R.E.. Knowledge-based Systems in Engineering, McGraw-Hill, 1991.
27. EDMONDS et al.,90- EDMONDS, E.; CANDY, L.; SLATTER, P. and LUNN, S.. Issues in the Design of Expert Systems for Business. Chapter 6- Expert Systems- Human Issues. Editors Dianne Berry and Anna Hart., Chapman and Hall, 1990.
28. ELLMAN,93- ELLMAN, A.U.; LINDBERG, I.I. and VILENIUS,M.J.. Simulation in the design of hydraulic-driven machines: new approach and aspects of application, Third Scandinavian International Conference on Fluid Power, 1993.
29. ELMQVIST,93- ELMQVIST, H.. Object-Oriented Modeling and Automatic Formula Manipulation in Dymola. SIMS’93, Scandinavian Simulation Society, Kongsberg, Norway, June 9-11, 1993.
30. ELMQVIST et al.,93- ELMQVIST, H.; CELLIER, F.E. and OTTER, M.. *Object-Oriented Modeling of Hybrid Systems*, ESS'93, European Simulation Symposium, Delf, The Netherlands, Oct 25-28, 1993.
31. FABRYCKY & BLANCHARD,90- FABRYCKY, W.J. & BLANCHARD, B.S., *Systems Engineering and Analysis*, Second Edition, Prentice-Hall, Inc.,1990.
32. FABRYCKY & BLANCHARD,91- FABRYCKY, W.J. and BLANCHARD, B.S., Life-cycle cost and Economic Analysis, Prentice Hall, Inc, 1991.
33. FLOTRON,96- Hydro Analyst- Copyright Flotron Ltd. Written by R.B. Walters & D. Harrison, October 96.

34. FRANCIS,93- FRANCIS, Philip H.. Progress toward a science base for design and manufacturing. The National Design Engineering Conference, ASME, Design for Manufacturability. DE-Vol.52, pp. 1-7.

35. FRENCH,85- FRENCH, M.J., *Conceptual Design for Engineers*, 2nd. Ed., Design Council, London, 1985.

36. FULTON & YEH,88- FULTON, R. E., and YEH, Chao-Pin. *Managing Engineering Design Information*. Proc. of the AIAA/AHS/ASEE Aircraft Design, Systems and Operations Conference. 7-9 Sept. 1988. Atlanta: AIAA paper No. 88-4452, 1988.

37. GIARRATANO & RILEY,94- GIARRATANO, J. and RILEY, G., *Expert Systems-Principles and Programming*, Second Edition, PWS Publishing Company, 1994.

38. GONZALEZ & DANKEL,93- GONZALEZ, Avelino J. and DANKEL, Douglas D., *The Engineering of Knowledge-Based Systems- Theory and Practice*, Prentice-Hall, Inc., 1993.

39. GÖTZ,84- GÖTZ, Werner, *Hydraulics, Theory and Applications from Bosch*. Robert Bosch GmbH Hydraulics Division, 1984.

40. GREEN,85- GREEN, W.L.. Aircraft Hydraulic Systems. John Wiley & Sons Ltd, 1985.

41. HART,92- HART, Anna. *Knowledge Acquisition For Expert Systems*, McGraw-Hill, Inc., 1992.

42. HARTLEY,91- HARTLEY, J.R. Concurrent Engineering, Productivity Press, Cambridge, 1991.

43. HOLLNAGEL,90- HOLLNAGEL, Erik. Responsibility Issues in Intelligent Decision Support Systems. Chapter 13- Expert Systems- Human Issues. Editors Dianne Berry and Anna Hart., Chapman and Hall, 1990.

44. HUANG et al.,93- HUANG, Guo; SHELDON, Derek and PERKS, Roger. *Concurrent Engineering by Cooperating Expert Systems*. The National Design Engineering Conference, Chicago, Illinois, ASME, 1993, pp 51-56.

45. IGNIZIO,91- IGNIZIO, James P., *Introduction to Expert Systems*, McGraw-Hill, Inc. 1991.

46. KARNOPP et al.,90- KARNOPP,D.; MARGOLIS, D.L. and ROSEMBERG, R.C., System Dynamics: a unified approach. John Wiley & Sons, Inc. 1990.

47. KIAN & CHEONG,93- KIAN, O.E. and CHEONG, F.S. Conceptualizing hydraulic system design, Third Scandinavian International Conference on Fluid Power, 1993.

48. KLEIN,92a- KLEIN, Gary A.. *Using Knowledge Engineering to Preserve the Corporate Memory*. The Psychology of Expertise- Cognitive Research and Empirical AI, Springer-Verlag, Edited by Robert R. Hoffman, 1992.

49. KLEIN,92b- KLEIN,M., DRCS: An Integrated System for Capture of Designs and Their Rationale, Artificial Intelligence in Design'92, Sydney, Australia, Kluwer Academic Publishers, 1992.

50. LANGDON et al.,95 LANGDON, P.M., BRACEWELL, R.H., DUFFY, M.J.W. and SHARPE, J.E.E.. An integrated design platform for the conceptual design of complex mechatronic systems. EDC 15, Internal Report, Lancaster, 1995.

51. LINSINGEN et al.,91- LINSINGEN, Irlan von; SILVA, Jonny C. e CASTRO, Nelson .J.A.. "Análise teórico-experimental do comportamento dinâmico de válvulas direcionais". IX Seminário de Instrumentação do Instituto Brasileiro do Petróleo, Nov. 1991.

52. LINSINGEN et al.,92- LINSINGEN, Irlan von; SILVA, Jonny C. e VIEIRA, Agnelo D.. "Análise teórico-experimental do comportamento dinâmico de válvulas reguladoras de vazão". V Congresso Nacional de Engenharia Mecânica do Chile, Nov. 1992.

53. LÜCKE et al. ,95- LÜCKE, Hermann A.H.; DE NEGRI, Victor J. e MADEIRA, Mauro N. "Emprego da Técnica de Orientação a Objetos na Automação Industrial", CONINFO'95.

54. MARCÉ & LE PARC,92- MARCÉ,L. and LE PARC, P.. Defining the semantics of languages for programmable controllers with synchronous processes. IFAC - Workshop on Real Time Programming WRTP'92 - Bruges - June, 1992.

55. MARTIN,95- MARTIN, Hugh, The Design of Hydraulic Components and Systems, Ellis Horwood Limited, 1995.

56. OH et al., 94- OH, V., LANGDON, P. and SHARPE, J.. *Schemebuilder: an integrated computer environment for product design*. Computer Aided Conceptual Design, 1994, Lancaster International Workshop on Engineering Design.

57. OH & SHARPE,95- OH, Vincent and SHARPE, John. Conflict management in an interdisciplinary design enviroment, EDC14, Internal Report, EDC, Lancaster University, 1995.

58. OHHASI,91- OHHASI,Y.. Basic Hydraulic Systems, JICA Training Course, "Oil Hydraulic and Its Application", Kyushu, Japan, Nov. 1991.

59. ORADY & SHAREEF, 93- ORADY, Elsayed A. and SHAREEF, Iqbal. Expert System Design Philosophy For Application of Simultaneous Engineering In Industry, The National Design Engineering Conference, ASME, Design for Manufacturability. DE-Vol.52, pp. 151-158.

60. OXFORD,93- OXFORD Advanced Learner's Dictionary of Current English, Fourth Edition- Oxford University Press, 1993.

61. PAHL & BEITZ,88- PAHL, G. and BEITZ, W.. Engineering Design- a Systematic Approach, Springer-Verlag, UK, 1988.

62. PERRON,96- PERRON, Stephen. Designing Fluid Power Controls with Grafcet. NFPA Proceedings, April 23-25, 1996.

63. PINCHES & ASHBY,89- PINCHES, Michael J. and ASHBY, John G., Power Hydraulics, Prentice Hall International UK Ltd., 1989.

64. RADHARAMANAN,93- RADHARAMANAN, R. "Concurrent Engineering and Design for Manufacture", I Congresso Latino-Americano de Engenharia Industrial, Florianópolis,93.

65. RICH & KNIGHT,91- RICH, E. and KNIGHT, K.. Artificial Intelligence, McGraw-Hill Book Company, 1991.

66. SARGENT et al.,88- SARGENT, C.M.; BURTON, R.T. and WESTMAN, R.W. "Expert Systems and Fluid Power", Fluid Power 8, BHRA, Birminghan, England, April, 1988.

67. SILVA,90- SILVA, Jonny C.. Desenvolvimento de um sistema para análise e modelamento dinâmico de robôs industriais. Master Dissertation in Mech. Engineering, aproved by the Post-Graduate Course in Mechanical Engineering Federal University of Santa Catarina, September 90.

68. SILVA & CHEUNG,97- SILVA, Jonny C. and CHEUNG, H.F.. Computer Support Environment Applying Object-Oriented Methods for Hydraulic Systems Design, Lancaster International Workshop on Engineering Design Computer Aided Conceptual Design, CACD'97, 14-15 April,1997.

69. SILVA & DAWSON,97a- SILVA, Jonny C. and DAWSON, David. The development of an expert system for hydraulic systems design focusing on Concurrent Engineering Aspects. ICED97, International Conference on Engineering Design, Tampere, Finland, 19-21 August 97.

70. SILVA & DAWSON, 97b- SILVA, Jonny C. and DAWSON, David. Knowledge Acquisition In The Context Of Hydraulic Systems Design- A Case Study. The Tenth Bath International Fluid Power Workshop. University of Bath, 10-12 September 97.

71. SILVA et al.,97- SILVA, Jonny C.; BENNETT, David and COUNSELL, J. M. Computational Agent to Integrate Expert System and Simulation Language. Paper to be presented at the 9th European Simulation Symposium, Passau, Germany, October 19-23, 1997.

72. SILVA & DAWSON, 97c- SILVA, Jonny C. and DAWSON, David. Computational Agent to Integrate An Expert System for Design and The Internet. Paper to be presented at WebNet97 World Conference of the WWW, Internet and Intranet. Oct. 31-Nov. 5, 1997, Toronto, Canada.

73. SHAKERI & BRÆK,97- SHAKERI, A. and BRÆK, R.. Modelling the Behaviour of Mechatronic Systems- ICED97, International Conference on Engineering Design, Tampere, Finland, 19-21 August 97.

74. SHARPE & OH,94- SHARPE, John and OH, Vincent. Computer Aided Conceptual Design Editors, 1994, Lancaster International Workshop on Engineering Design.

75. TANSKANEN,97- TANSKANEN, Kari. On-Site Component Selection Support using WWW. ICED97, International Conference on Engineering Design, Tampere, Finland, 19-21 August 97, Vol.2, pp 243-248.

76. TOMIYAMA et al. ,94- TOMIYAMA, T.; KIRIYAMA,T. and UMEDA,Y.. "Toward Knowledge Intensive Engineering. 1994 Lancaster International Workshop on Engineering Design.

77. ULLMAN,97- ULLMAN, David G.. Metrics for Evaluation of the Product Design Process. ICED97, International Conference on Engineering Design, Tampere, Finland, 19-21 August 97.

78. WATERMAN,86- WATERMAN,D.A., *A Guide to Expert Systems*, Addison-Wesley Publishing Company, 1986.

79. WAY,93- WAY, N. *The Future for Hydraulics*, 10th International Conference on Fluid Power, April 1993, Belgium.

80. WILSON,90- WILSON, C.C. *Potential pitfalls of Concurrent Engineering*. Concurrent Engineering, 1990.

81. YOURDON & COAD,92- YOURDON, E. &, COAD, P. Análise Baseada em Objetos, Editora Campus, Série Yourdon Press, 1992.

82. YALIF,94- YALIF, Guy U.. *The Computer Aided Aircraft-Design Package CAAP*, Third Conference on CLIPS, Proceedings Electronic Version, September 12–14, 1994, Lyndon B. Johnson Space Center.

83. ZUCKER & DEMAID,92- ZUCKER, J. & DEMAID, A.. *Modelling Heterogeneous Engineering Knowledge As Transactions Between Delegating Objects*, Artificial Intelligence In Design'92, Sydney, Australia, pp. 141-149.