

**UTILISATION OF EXPERT SYSTEMS IN  
CONTAINER SHIP DESIGN: ACCOMMODATION  
LAYOUT DESIGN EXPERT SYSTEM (ALDES)**

**Ph.D. Thesis by  
Şebnem HELVACIOĞLU, M.Sc.**

**(508932019)**

**T.C. YÜKSEKÖĞRETİM KURULU  
DOKÜMANTASYON MERKEZİ**

112226

**Date of submission : 16 February 2001**

**Date of defence examination: 3 July 2001**

**Supervisor (Chairman): Assoc. Prof. Dr. Mustafa İNSEL**

*M. İNSEL*

**Members of the Examining Committee: Prof.Dr. Yücel ODABAŞI (İTÜ.)**

*Y. ODABAŞI*

**Assoc. Prof. Dr. Şakir KOCABAŞ (İTÜ.)**

*Ş. KOCABAŞ*

**Prof.Dr. Ercan ÖZTEMEL (Sak.Ü.)**

*E. ÖZTEMEL*

**Asist. Prof.Dr. Ercan KÖSE (KTÜ.)**

*E. KÖSE*

**JULY 2001**

**UZMAN SİSTEMLERİN KONTEYNER GEMİSİ  
DİZAYNINA UYGULANMASI: ÜST YAPI YERLEŞİM  
DİZAYNI UZMAN SİSTEMİ (ALDES)**

**DOKTORA TEZİ**  
**Y. Müh. Şebnem HELVACIOĞLU**  
**(508932019)**

**Tezin Enstitüye Verildiği Tarih : 16 Şubat 2001**

**Tezin Savunulduğu Tarih : 3 Temmuz 2001**

**Tez Danışmanı : Doç.Dr. Mustafa İNSEL**

**Diğer Jüri Üyeleri : Prof.Dr. Yücel ODABAŞI (İTÜ.)**

**Doç. Dr. Şakir KOCABAŞ (İTÜ.)**

**Prof.Dr. Ercan ÖZTEMEL (Sak.Ü.)**

**Yar.Doç.Dr. Ercan KÖSE (KTÜ.)**

**TEMMUZ 2001**

## ACKNOWLEDGEMENTS

I would like to say thank you to those who have given me their love, time, support and shared their knowledge with me. I would like to thank my supervisor *Dr. Mustafa Insel* who is a friend more than a supervisor. This thesis became more valuable because of his intelligence and skills. I also wish to thank *Prof. Dr. A. Yucel Odabasi* for dealing with me and tolerating my bad temper sometimes, suggesting to me this very interesting and difficult subject and supporting me in all stages of my academic life. A special indebtedness is also owed to my former student and good friend, *Mr. Gokhan Goke* with whom I have had many constructive discussions and to *Dr. Osman Turan* for giving me the opportunity of working at Strathclyde University and for being especially supportive during our time in Glasgow. I would like to thank *Omer Dayi* who has been a great father for sharing his experiences with me as an ocean-going master.

I must also thank the following:

Mr. Altan Demirsoylu and Mr. Haluk Sismanyazici, two of the experienced ship designers in Turkey; Mr. David Watson for his advice on this work and in particular his valuable discussion in ship design; Dr. Martin Welsh whose thesis was an inspiration and a valuable resource also for being very generous with his valuable time in helping to explain some of the difficulties with Expert System; Dr Aykut Olcer, my friend and colleague for his support and encouragement; Mrs. Meryem (Kavak) Dikili for typing the manuscript and Handan Oztimer for her help in the preparation of drawings and diagrams; my students who are friends, for their respect, love and friendship. I also would like to thank Ms. Paula Dixon for proof reading my thesis in English with endless patience.

I would like to express my sincere thanks to my family who gave me endless support and encouragement throughout this study. Also I would like to thank my friends who have tolerated erratic mood changes. Finally I would like to express my deepest gratitude to *Dr. Ismail Helvacioglu* who is my best friend and my husband. This thesis is dedicated to him and to my daughter *Elif* to show my love and respect to them.

February 2001

Sebnem Helvacioglu

## 1. TABLE OF CONTENTS

|                                       |       |
|---------------------------------------|-------|
| ACKNOWLEDGEMENTS                      | iii   |
| LIST OF ABBREVIATIONS                 | x     |
| LIST OF TABLES                        | xi    |
| LIST OF FIGURES                       | xii   |
| NOMENCLATURE                          | xiv   |
| SUMMARY                               | xvi   |
| ÖZET                                  | xviii |
| 1. INTRODUCTION                       | 1     |
| 1.1 Introduction                      | 1     |
| 1.2 Problem Definition                | 1     |
| 1.3 Method of Study                   | 3     |
| 2. GENERAL DESIGN AND SHIP DESIGN     | 6     |
| 2.1 Introduction                      | 6     |
| 2.2 The Nature of Design              | 6     |
| 2.3 Design Stages                     | 8     |
| 2.4 Design Characteristics            | 9     |
| 2.5 Models of Design                  | 10    |
| 2.6 A Knowledge Based Model of Design | 11    |
| 2.6.1 Representation                  | 11    |
| 2.6.1.1 Facts                         | 12    |
| 2.6.1.2 Knowledge                     | 12    |
| 2.6.1.3 Control                       | 13    |
| 2.6.2 Reasoning                       | 14    |
| 2.6.3 Syntax                          | 16    |
| 2.6.4 Design Space                    | 16    |
| 2.6.4.1 A Design Syntax               | 18    |
| 2.6.4.2 A Design Interpreter          | 19    |

|                                                              |           |
|--------------------------------------------------------------|-----------|
| 2.6.4.3 Representing Design Knowledge                        | 20        |
| 2.6.4.4 Design Processes                                     | 20        |
| <b>2.7 Ship Design</b>                                       | <b>21</b> |
| 2.7.1 Ship Design Process                                    | 22        |
| 2.7.2 Methods of Ship Design                                 | 24        |
| 2.7.2.1 Design Spiral Method                                 | 24        |
| 2.7.2.2 Mothership Method                                    | 24        |
| 2.7.2.3 Optimisation Method                                  | 25        |
| 2.7.2.4 Decision Based Method                                | 25        |
| 2.7.2.5 Knowledge Based Method                               | 26        |
| 2.7.3 Overview of Ship Design                                | 26        |
| <b>2.8 Conclusion</b>                                        | <b>32</b> |
| <b>3. ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEMS</b>         | <b>33</b> |
| <b>3.1 Introduction</b>                                      | <b>33</b> |
| <b>3.2 Definition and History of Artificial Intelligence</b> | <b>33</b> |
| <b>3.3 Expert Systems</b>                                    | <b>36</b> |
| 3.3.1 The Structure                                          | 36        |
| 3.3.2 Elements of An Expert System                           | 37        |
| 3.3.2.1 Knowledge Base                                       | 38        |
| 3.3.2.2 Inference Engine                                     | 44        |
| 3.3.2.3 User Interface                                       | 48        |
| 3.3.2.4 Explanation Facility                                 | 48        |
| <b>3.4 Knowledge Acquisition and Elicitation</b>             | <b>49</b> |
| 3.4.1 Knowledge Elicitation Techniques                       | 50        |
| <b>3.5 Expert Systems Versus Conventional Programming</b>    | <b>51</b> |
| <b>3.6 Advantages and Disadvantages</b>                      | <b>51</b> |
| <b>3.7 Selecting the Appropriate Problem</b>                 | <b>52</b> |
| <b>3.8 Expert Systems Applications and Domains</b>           | <b>53</b> |
| <b>3.9 Expert Systems Tools</b>                              | <b>54</b> |
| <b>3.10 Conclusion</b>                                       | <b>55</b> |

|                                                                          |           |
|--------------------------------------------------------------------------|-----------|
| <b>4. LAYOUT DESIGN</b>                                                  | <b>58</b> |
| <b>4.1. Introduction</b>                                                 | <b>58</b> |
| <b>4.2. Systematic Layout Planning (Muther)</b>                          | <b>59</b> |
| 4.2.1 Information Gathering                                              | 60        |
| 4.2.2 Flow Analysis and Activity Analysis                                | 60        |
| 4.2.2.1 Type of flow Patterns                                            | 61        |
| 4.2.2.2 Flow-analysis Approaches                                         | 62        |
| 4.2.2.3 Activity Relationship Analysis                                   | 63        |
| 4.2.2.4 The Relationship Diagram                                         | 64        |
| 4.2.3 Space-relationship Establishment                                   | 64        |
| 4.2.4 Developing Layout Alternatives                                     | 65        |
| 4.2.4.1 Automated Layout Design Procedure - ALDEP                        | 66        |
| 4.2.4.2 Computerized Relationship Layout Planning - CORELAP              | 67        |
| 4.2.4.3 Computerised Relative Allocation of Facilities Technique - CRAFT | 69        |
| 4.2.5 Evaluation of Layout Alternatives                                  | 70        |
| <b>4.3. Ship General Layout Problem</b>                                  | <b>71</b> |
| <b>4.4. Ship Accommodation Design</b>                                    | <b>73</b> |
| <b>4.5. Conclusion</b>                                                   | <b>74</b> |
| <b>5. KNOWLEDGE BASE OF A CONTAINER SHIP DESIGN</b>                      | <b>76</b> |
| <b>5.1 Introduction</b>                                                  | <b>76</b> |
| <b>5.2 Knowledge Elicitation in the Ship Design Domain</b>               | <b>76</b> |
| <b>5.3 Knowledge Representation</b>                                      | <b>77</b> |
| <b>5.4 The Knowledge Base</b>                                            | <b>78</b> |
| 5.4.1 Container Ship Design                                              | 79        |
| 5.4.2 Case Based Design Step                                             | 80        |
| 5.4.3 The First Approach to Main Dimensions                              | 80        |
| 5.4.4 The First Approach to Hullform                                     | 89        |
| 5.4.5 General Arrangement                                                | 89        |
| 5.4.5.1 Set Main Bulkheads and Double Bottom                             | 90        |
| 5.4.5.2 Engine Room Size and Location                                    | 92        |
| 5.4.5.3 Tanks                                                            | 94        |
| 5.4.5.4 Cargo Hold Container Arrangement                                 | 95        |

|                                                                             |            |
|-----------------------------------------------------------------------------|------------|
| 5.4.5.5 General Arrangement of Weather deck                                 | 99         |
| 5.4.6 Crew Number Determination                                             | 103        |
| 5.4.7 Accommodation                                                         | 103        |
| 5.4.7.1 Crew Space                                                          | 103        |
| 5.4.7.2 Galley, Messrooms, Dining Rooms, and Store Spaces                   | 105        |
| 5.4.7.3 Relation Between Spaces and Access                                  | 106        |
| 5.4.7.4 Accommodation Design Knowledge                                      | 106        |
| <b>5.5 Application of Facility Layout Tools</b>                             | <b>109</b> |
| 5.5.1 Human and Material Flow of a Container Ship                           | 109        |
| 5.5.2 Relational Chart For A Container Ship                                 | 112        |
| <b>5.6 Conclusion</b>                                                       | <b>115</b> |
| <b>6. ACCOMMODATION LAYOUT DESIGN EXPERT SYSTEM APPLICATION</b>             | <b>116</b> |
| <b>6.1. Introduction</b>                                                    | <b>116</b> |
| <b>6.2. Application of The Conceptual Model</b>                             | <b>117</b> |
| <b>6.3. Programming Tools</b>                                               | <b>120</b> |
| 6.3.1 Delphi Part                                                           | 120        |
| 6.3.2 CLIPS Part                                                            | 120        |
| 6.3.2.1 1 <sup>st</sup> Module: Calculate Main Dimensions And Set Bulkheads | 123        |
| 6.3.2.2 2 <sup>nd</sup> Module: Crew Number Calculation                     | 126        |
| 6.3.2.3 3 <sup>rd</sup> Module: Living Compartment                          | 129        |
| 6.3.2.4 4 <sup>th</sup> Module: The Other Compartment                       | 132        |
| 6.3.2.5 5 <sup>th</sup> Module: Relational Chart:                           | 133        |
| 6.3.2.6 6 <sup>th</sup> Module: Deck Dimensions:                            | 133        |
| 6.3.2.7 7 <sup>th</sup> Module: Layout Design:                              | 134        |
| 6.3.2.8 8 <sup>th</sup> Module: Calculation Of Cost:                        | 134        |
| <b>6.4. Case Studies</b>                                                    | <b>135</b> |
| 6.4.1 Case Study I                                                          | 135        |
| 6.4.2 Case Study II                                                         | 139        |
| 6.4.3 Comparison of The Results                                             | 143        |
| <b>6.5. Conclusion</b>                                                      | <b>143</b> |
| <b>7. CONCLUSIONS AND THE FUTURE WORK</b>                                   | <b>144</b> |
| <b>7.1 Conclusions</b>                                                      | <b>144</b> |

|                                                  |            |
|--------------------------------------------------|------------|
| <b>7.2 The Future Work</b>                       | <b>147</b> |
| <b>REFERENCES</b>                                | <b>149</b> |
| <b>APPENDIX A. CLIPS EXPERT SYSTEM SHELL</b>     | <b>155</b> |
| <b>A.1 The History of CLIPS</b>                  | <b>155</b> |
| <b>A.2 Interacting With Clips</b>                | <b>155</b> |
| A.2.1 Integration with Other Languages           | 156        |
| <b>A.3 Basic Programming Elements</b>            | <b>156</b> |
| A.3.1 Data Types                                 | 156        |
| A.3.2 Functions                                  | 157        |
| A.3.3 Constructs                                 | 158        |
| <b>A.4 Data Abstraction</b>                      | <b>158</b> |
| A.4.1 Facts                                      | 158        |
| A.4.1.1 Ordered Facts                            | 158        |
| A.4.1.2 Non-ordered Facts                        | 159        |
| A.4.1.3 Initial Facts                            | 159        |
| A.4.2 Objects                                    | 159        |
| A.4.2.1 Initial Objects                          | 160        |
| A.4.2.2 Global Variables                         | 160        |
| <b>A.5 Knowledge Representation</b>              | <b>160</b> |
| A.5.1 Heuristic Knowledge – Rules                | 160        |
| A.5.2 Procedural Knowledge                       | 161        |
| <b>A.6 Basic Cycle Of Rule Execution</b>         | <b>161</b> |
| <b>A.7 Conflict Resolution Strategies</b>        | <b>162</b> |
| A.7.1 Depth Strategy                             | 163        |
| A.7.2 Breadth Strategy                           | 163        |
| A.7.3 Simplicity Strategy                        | 163        |
| <b>A.8 LHS Syntax</b>                            | <b>163</b> |
| A.8.1 Pattern Conditional Element                | 164        |
| <b>APPENDIX B. REGULATIONS</b>                   | <b>165</b> |
| <b>B.1 Rules and Regulations For Ship Design</b> | <b>165</b> |
| B.1.1 Stairways                                  | 165        |
| B.1.2 Companion Ways                             | 166        |

|                                                      |            |
|------------------------------------------------------|------------|
| B.1.3 Emergency Escapes                              | 167        |
| B.1.4 Sick Room                                      | 168        |
| B.1.5 Habitability                                   | 168        |
| B.1.6 Lighting                                       | 170        |
| B.1.7 Living Compartments                            | 170        |
| B.1.8 Crew Bunks/Beds                                | 173        |
| B.1.9 Crew Lockers                                   | 174        |
| B.1.10 Floor Area                                    | 175        |
| B.1.11 Mess Rooms                                    | 177        |
| B.1.12 Pantry                                        | 178        |
| B.1.13 Recreation Rooms                              | 179        |
| B.1.14 Galleys                                       | 179        |
| B.1.15 Provision Rooms                               | 180        |
| B.1.16 Sanitary Equipment                            | 180        |
| B.1.17 Structure                                     | 182        |
| B.1.18 Emergency                                     | 184        |
| B.1.19 Life Saving                                   | 185        |
| <b>APPENDIX C. KNOWLEDGE ELICITATION (INTERVIEW)</b> | <b>189</b> |
| <b>APPENDIX D. DELPHI</b>                            | <b>191</b> |
| <b>D.1 Delphi</b>                                    | <b>191</b> |
| <b>D.2 Data Representation in ALDES</b>              | <b>192</b> |
| <b>CIRRICULUM VITAE</b>                              | <b>194</b> |

## LIST OF ABBREVIATIONS

|          |                                                                        |
|----------|------------------------------------------------------------------------|
| AI       | : Artificial Intelligence                                              |
| AKO      | : A-Kind-Of                                                            |
| ALDEP    | : Automated Layout Design Procedure                                    |
| ALDES    | : Accommodation Layout Design Expert Systems                           |
| CAD      | : Computer Aided Design                                                |
| CLIPS    | : C Language Integrated Production System                              |
| CORELAP  | : Computerised Relationship Layout Planning                            |
| DSP      | : Decision Support Systems                                             |
| ES       | : Expert System                                                        |
| GRT      | : Gross Tonnage                                                        |
| HVAC     | : Heating, Ventilation, and Air Conditioning                           |
| ILO      | : International Labour Organisation                                    |
| IMO      | : International Maritime Organisation                                  |
| ISO      | : International Standards Organisation                                 |
| KBS      | : Knowledge Based Systems                                              |
| MARPOL   | : Marine Pollution                                                     |
| REL      | : Relationship Chart                                                   |
| SBG      | : See-Berufsgenossenschaft                                             |
| SLP      | : Systematic Layout Planning                                           |
| SOLAS    | : Safety of Life at Sea                                                |
| STCW     | : Standards of Training, Certification and Watchkeeping for Seafarers  |
| TEU      | : Twenty feet Equivalent Unit                                          |
| TL       | : Turkish Lloyd                                                        |
| VRML     | : Virtual Reality Modelling Language                                   |
| MARIN    | : Maritime Research Institute Netherlands                              |
| INCODES  | : Intelligent Concept Design System                                    |
| QUAESTOR | : A Knowledge Based System For Computations in Preliminary Ship Design |
| SWATH    | : Small Waterplane Area Twin Hull                                      |
| CR       | : Closeness rating                                                     |
| TCR      | : Total Closeness Rating                                               |
| ITU      | : Istanbul Technical University                                        |

## LIST OF TABLES

|                                                                                   | <u>Page Nr.</u> |
|-----------------------------------------------------------------------------------|-----------------|
| Table 2.1 Three types of reasoning in relation to the syllogism                   | 16              |
| Table 2.2 Evolution of ship design                                                | 28              |
| Table 3.1 History of Artificial Intelligence                                      | 33              |
| Table 3.2 Typical differences between conventional programs versus Expert Systems | 50              |
| Table 3.3 Broad classes of Expert Systems                                         | 53              |
| Table 5.1 Some sample container ships                                             | 79              |
| Table 5.2 Number of watertight transverse bulkheads                               | 90              |
| Table 5.3 Order numbers for relational chart                                      | 112             |
| Table 5.4 Relational chart for the sample ship                                    | 113             |
| Table 6.1 Calculated Main Dimensions                                              | 127             |
| Table 6.2 Crew members according to the program                                   | 129             |
| Table 6.3 The other compartments in the accommodation part                        | 133             |
| Table 6.4 Case I output data                                                      | 139             |
| Table 6.5 Case II output data                                                     | 143             |

## LIST OF FIGURES

|                                                                                                                                                    | <u>Page Nr.</u> |
|----------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| <b>Figure 2.1</b> : A Syllogism Represented Graphically                                                                                            | 15              |
| <b>Figure 2.2</b> : A space of design                                                                                                              | 17              |
| <b>Figure 2.3</b> : Generative and interpretative knowledge                                                                                        | 18              |
| <b>Figure 2.4</b> : Design spiral                                                                                                                  | 27              |
| <b>Figure 2.5</b> : Andrew's spiral                                                                                                                | 27              |
| <b>Figure 3.1</b> : Principal components of an Expert System                                                                                       | 35              |
| <b>Figure 3.2</b> : The elements of a typical Expert System                                                                                        | 37              |
| <b>Figure 3.3</b> : A simple semantic net                                                                                                          | 40              |
| <b>Figure 3.4</b> : Example of a frame                                                                                                             | 41              |
| <b>Figure 3.5</b> : Example of forward chaining                                                                                                    | 44              |
| <b>Figure 3.6</b> : Example of backward chaining                                                                                                   | 45              |
| <b>Figure 3.7</b> : Map Travelsal problem                                                                                                          | 46              |
| <b>Figure 3.8</b> : A dept-first search                                                                                                            | 47              |
| <b>Figure 3.9</b> : Breadth-first search                                                                                                           | 47              |
| <b>Figure 4.1</b> : Systematic Layout Planning procedure                                                                                           | 58              |
| <b>Figure 4.2</b> : Basic Horizontal Flow Pattern (a) Straight or I Flow; (b) L Flow; (c) U Flow; (d) Circular or O Flow; (e) Serpentine or S Flow | 60              |
| <b>Figure 4.3</b> : Vertical flow pattern                                                                                                          | 61              |
| <b>Figure 4.4</b> : Activity relationship chart                                                                                                    | 62              |
| <b>Figure 4.5</b> : Activity relationship diagram                                                                                                  | 63              |
| <b>Figure 4.6</b> : Space relationship diagram                                                                                                     | 64              |
| <b>Figure 4.7</b> : Zig-zag pattern                                                                                                                | 66              |
| <b>Figure 4.8</b> : CORELAP evaluation                                                                                                             | 67              |
| <b>Figure 5.1</b> : The basic steps in container ship design                                                                                       | 77              |
| <b>Figure 5.2</b> : Diagram for first approximation of number of container rows and tiers under deck                                               | 80              |
| <b>Figure 5.3</b> : Engine speed versus power                                                                                                      | 81              |
| <b>Figure 5.4</b> : Deadweight versus capacity                                                                                                     | 82              |
| <b>Figure 5.5</b> : Deadweight versus displacement                                                                                                 | 82              |
| <b>Figure 5.6</b> : Engine power versus Admiralty coefficient                                                                                      | 83              |
| <b>Figure 5.7</b> : Low speed                                                                                                                      | 84              |
| <b>Figure 5.8</b> : Medium speed                                                                                                                   | 84              |
| <b>Figure 5.9</b> : Ship profile                                                                                                                   | 85              |
| <b>Figure 5.10</b> : Ship depth and breadth                                                                                                        | 86              |
| <b>Figure 5.11</b> : Deadweight versus grosstonnage                                                                                                | 87              |
| <b>Figure 5.12</b> : Determining the vertical location of the main engine                                                                          | 92              |
| <b>Figure 5.13</b> : Required clearances at the engine aft                                                                                         | 93              |
| <b>Figure 5.14</b> : Container standards                                                                                                           | 95              |
| <b>Figure 5.15</b> : Typical twistlock                                                                                                             | 96              |

|                      |                                                                  |     |
|----------------------|------------------------------------------------------------------|-----|
| <b>Figure 5.16</b>   | : Typical cell-guide system                                      | 97  |
| <b>Figure 5.17</b>   | : Container spaces                                               | 98  |
| <b>Figure 5.18</b>   | : Hatch                                                          | 99  |
| <b>Figure 5.19</b>   | : Hatch                                                          | 99  |
| <b>Figure 5.20</b>   | : Code of container position                                     | 101 |
| <b>Figure 5.21</b>   | : Departure from a port                                          | 109 |
| <b>Figure 5.22</b>   | : Dinner Time                                                    | 109 |
| <b>Figure 5.23</b>   | : Arrival to a port                                              | 110 |
| <b>Figure 5.24</b>   | : Leaving a ship in an emergency                                 | 110 |
| <b>Figure 5.25</b>   | : Fire fighting with an A type fire                              | 111 |
| <b>Figure 6.1</b>    | : Conceptual Model of ALDES                                      | 119 |
| <b>Figure 6.2</b>    | : Decomposition                                                  | 120 |
| <b>Figure 6.3</b>    | : Flow Chart of ALDES                                            | 121 |
| <b>Figure 6.4</b>    | : A CLIPS window with facts and agenda windows after a reset     | 122 |
| <b>Figure 6.5</b>    | : Watch Facility of CLIPS                                        | 123 |
| <b>Figure 6.6</b>    | : Explanation facility of CLIPS                                  | 123 |
| <b>Figure 6.7</b>    | : Derived facts of the first module                              | 126 |
| <b>Figure 6.8</b>    | : A screen image from second module to show explanation facility | 129 |
| <b>Figure 6.9</b>    | : Getting the compartment template                               | 132 |
| <b>Figure 6.10</b>   | : Modifying floor area attribute                                 | 132 |
| <b>Figure 6.11 a</b> | : Deck 1 and Deck 2 for the first case                           | 136 |
| <b>Figure 6.11 b</b> | : Deck 3 and Deck 4 for the first case                           | 137 |
| <b>Figure 6.11 c</b> | : Deck 5 and Deck 6 for the first case                           | 138 |
| <b>Figure 6.12 a</b> | : Deck 1 and Deck 2 for the second case                          | 140 |
| <b>Figure 6.12 b</b> | : Deck 3 and Deck 4 for the second case                          | 141 |
| <b>Figure 6.12 c</b> | : Deck 5 and Deck 6 for the second case                          | 142 |
| <b>Figure D.1</b>    | : Class hierarchy of ALDES                                       | 193 |
| <b>Figure D.2</b>    | : Typical ownership relation of test case                        | 193 |

## NOMENCLATURE

### Chapter 4

|                                |                                                                                                  |
|--------------------------------|--------------------------------------------------------------------------------------------------|
| <b>A</b>                       | : Absolutely necessary                                                                           |
| <b><math>A_i</math></b>        | : Environmental loading.                                                                         |
| <b><math>a_{ij}</math></b>     | : Association of activity $i$ and $j$                                                            |
| <b><math>a_{ik}</math></b>     | : Cost of location of facility $k$ at a site $i$                                                 |
| <b><math>B_{klj}</math></b>    | : Cost of relationship between facilities $k$ and $l$ , with $k$ located at $i$ and $l$ at $j$ . |
| <b><math>C</math></b>          | : Cost function                                                                                  |
| <b><math>C_{kl}</math></b>     | : Cost of transportation between facilities $k$ and $i$ per unit of quantity and distance.       |
| <b><math>C_L</math></b>        | : Cost parameter such as cost of exterior walls, of fence etc.                                   |
| <b><math>COST_{MIN}</math></b> | : Cost of circulation and environmental factors                                                  |
| <b><math>D_{ij}</math></b>     | : Transportation distance between sites $i$ and $j$ .                                            |
| <b><math>d_{ij}</math></b>     | : Distance between activity $i$ and $j$                                                          |
| <b>E</b>                       | : Especially important                                                                           |
| <b>I</b>                       | : Important                                                                                      |
| <b>L</b>                       | : Weighting factor                                                                               |
| <b>m</b>                       | : Possible locations                                                                             |
| <b>n</b>                       | : Functional centres                                                                             |
| <b>O</b>                       | : Ordinary Importance                                                                            |
| <b><math>Q_{kl}</math></b>     | : Quantity of materials to be transferred between facilities $k$ and $i$ per period of time.     |
| <b>U</b>                       | : Unimportant                                                                                    |
| <b>X</b>                       | : Undesirable                                                                                    |
| <b><math>X_{ik}</math></b>     | : Layout alternative factor.                                                                     |

### Chapter 5

|                             |                                                                 |
|-----------------------------|-----------------------------------------------------------------|
| <b>B</b>                    | : Greatest moulded breadth                                      |
| <b><math>C_B</math></b>     | : Block coefficient                                             |
| <b>DB</b>                   | : Depth of double bottom                                        |
| <b>h</b>                    | : Depth of centre girder                                        |
| <b>H</b>                    | : Height containers                                             |
| <b>KM</b>                   | : Metacentric height from keel                                  |
| <b><math>l</math></b>       | : Length of a container                                         |
| <b>L</b>                    | : Ship length                                                   |
| <b><math>L_{BCC}</math></b> | : Length between most forward cargo hold and collision bulkhead |
| <b><math>L_{BP}</math></b>  | : Length between perpendiculars                                 |
| <b><math>L_c</math></b>     | : Container portion                                             |

|          |                                             |
|----------|---------------------------------------------|
| $L_c$    | : Length of engine room and after peak tank |
| $L_{ER}$ | : Length of engine room                     |
| $L_f$    | : Length of fore peak tank                  |
| $N_x$    | : Number of containers in length            |
| $N_y$    | : Number of tiers of containers in holds    |
| $N_z$    | : Number of tiers of containers in holds    |
| $V$      | : Ship velocity                             |
| $T$      | : Draught                                   |
| $\Delta$ | : Displacement                              |



## **UTILISATION OF EXPERT SYSTEMS IN CONTAINER SHIP DESIGN: ACCOMMODATION LAYOUT DESIGN EXPERT SYSTEM (ALDES)**

### **SUMMARY**

Although the term Artificial Intelligence (AI) was first used in the early 1950's, its origin is referred back to 17<sup>th</sup> century when Pascal invented the first calculator. AI and its sub-fields were widely used in all scientific and industrial fields after the development of computer technology in the late 1980's. One of the sub-fields of AI, Expert Systems (ESs) has found a wide application area along with the development of computer technology.

Ship design is very complex, based on expertise and iterative in nature. It also requires a wide dynamic knowledge base and database. Every year rules and regulations are updated, so are the knowledge base. Due to these complex characteristics of ship design, it is very suitable for an ES application. There are not many ES applications in ship design used in Turkey. The domain experts in ship design and production industry in Turkey have been changing their jobs very often, as there are some very productive times followed by recessions. When they transfer between jobs, the domain knowledge is lost in the design office. If ESs in ship design are developed and introduced to the industry, expertise of the experts can be preserved even if they leave the environment. Thus, the dynamic knowledge base can be updated more easily.

In this work, applicability of ESs to ship design is examined by reviewing the topics including engineering design, ship design, AI, ESs, plant layout methods, and by developing a pilot application in ship design. Feeder type of container ship type was chosen for the application, because this type of ship is commonly built in Turkey.

AI is a science that it is used to teach computers to think and decide like human beings. ESs are computer programs to solve a domain problem based on a knowledge base. There are several programming languages and shells available to develop ESs. CLIPS shell was preferred for the application in this work. There are two basic inference methods in ESs. The first one is the forward chaining which uses facts to achieve the goal. The other one is the backward chaining which starts from a goal to achieve facts. This work aims to achieve a design solution starting from design requirements. Therefore the forward chaining method was found to be more suitable for the purpose.

In this thesis, general design is examined and ship design techniques are summarised, since this work establishes an ES application into ship design. A knowledge based design system was developed and the knowledge was represented by production rules in the ES.

An ES should be developed by collaborative work of a domain expert and a knowledge expert. The most important parameter in an ES is the knowledge base. One of the most difficult problems in the development of an ES, is to acquire the knowledge from experts. Interviewing is one of the techniques for knowledge acquisition. A knowledge base is developed by interviewing the experts, using the reference books and the technical rules for container ship design.

Several techniques have been developed for layout design and optimisation in architecture and industrial engineering. However, these techniques are not commonly used in ship design. Some plant layout techniques used in industrial engineering for factories and plants were examined and applied to general arrangement of accommodation spaces of ships.

Interviews with ten experts were carried out and SOLAS (Safety Of Life At Sea), ILO (International Labour Organisation), SBG (See-Berufsgenossenschaft) rules were added to develop the knowledge base. The heuristic knowledge acquired from experts and other technical knowledge are represented as rules. An ES program package called ALDES was developed consisting eight modules. Preliminary main dimensions of a container ship were calculated with formulas which were derived from a database of similar ships, the main transverse bulkheads were set and the general arrangement design was completed using ALDES. A living compartment was defined as an object with several attributes and the heuristic knowledge about the compartments were coded by using logic programming to generate living compartments in the package.

ALDES was run to calculate the distance for evacuation in an emergency for different scenarios. The effect of various compartment door locations to the evacuation distance were also examined.

This work is hoped to be useful to design safer ships, especially the safety of ships and human lives after collisions is the main concern nowadays.

The AI and the ES have not been widely used in the Naval Architecture field. However, following this work, the application of ESs in ship building is strongly suggested since the knowledge base is so large and dynamic to handle.

Considering the complex nature of ship design the study showed that under the certain assumptions and constraints, a computer model can be developed by using Expert System.

# UZMAN SİSTEM METODUNUN KONTEYNER GEMİLERİNİN DİZAYNINA UYGULANMASI: UZMAN SİSTEM İLE YAŞAM MALLİ DİZAYNI (ALDES)

## ÖZET

Yapay Zeka (YZ) terimi ilk olarak 1950'lerin başında kullanılmaya başlamış olmakla beraber, temelleri 17. yüzyılda Pascal'ın ilk hesap makinasını bulmasına dayanır. 1980'lerin ikinci yarısında bilgisayar teknolojisindeki gelişmelere paralel olarak YZ ve bunun alt bilim dalları tüm bilimsel ve teknolojik alanlarda daha yaygın olarak kullanılmaya başlamıştır. YZ alt bilim dallarından biri olan Uzman Sistemler (US) bilgisayar teknolojisindeki bu gelişme ile beraber yaygınlaşmıştır.

Gemi dizaynı oldukça karmaşık, uzman bilgisine dayalı ve iteratif bir yapıya sahiptir. Aynı zamanda dizayn için geniş ve dinamik bir bilgi tabanı ve veri tabanı gerekmektedir. Her yıl uygulamaya konulan yeni kurallar ve gelişmeler nedeni ile bilgi tabanı değişmektedir. Bu özellikleri nedeni ile gemi dizaynı US uygulamaları için oldukça elverişli bir alandır. Türkiye'de gemi dizaynı konusunda Uzman Sistem uygulamasına pek rastlanmamaktadır. Türkiye'de gemi dizayn ve inşaatı sektöründe, çok verimli dönemleri çok durağan dönemler takip ettiğinden, konunun uzmanları sürekli yer değiştirmektedirler. Uzmanlar yer değiştirirken konu ile ilgili bilgileri de beraberlerinde götürmektedirler. Uzman Sistemlerin geliştirilip piyasaya kazandırılması sayesinde uzmanların ortamdaki ayrılması halinde bile bilgileri saklanabilecek ve zamanla değişen bilgi tabanı kolaylıkla güncelleştirilebilecektir.

Bu çalışmada, Uzman Sistemlerin gemi dizaynına uygulanabilirliği incelenmiş, genel dizayn ve gemi dizaynı, Yapay Zeka ve Uzman Sistemler, tesis yerleştirmesi konuları araştırılmış ve bunların uygulamaları verilmiştir. Uygulama için örnek gemi olarak Türkiye'de çok fazla sayıda inşa edilmesi nedeni ile, feeder konteyner gemileri seçilmiştir.

Yapay Zeka bilgisayara insan gibi düşünmeyi ve karar vermeyi öğretme bilimidir. Uzman Sistemler; bir bilgi tabanına bağlı olarak, belli bir alandaki problemleri çözmek amacı ile yazılmış bilgisayar programlarıdır. Uzman Sistem programlarının yazılabilmesi için çeşitli programlama dilleri ve paketleri vardır. Bu çalışmadaki uygulama için CLIPS paketi seçilmiştir. Uzman Sistemlerin temel iki çalışma prensibi vardır. Bunlardan birincisi eldeki bilgiden sonuca ulaşma; ileri zincir yöntemidir. Diğer de sonuçtan bilgiye ulaşma, geriye zincir yöntemidir. Bu çalışma eldeki gereksinim tanımından dizayna erişme amacını taşıdığı için, sadece ileri zincir yöntemi kullanılmıştır.

Bu çalışma Uzman Sistemlerin gemi dizaynına uygulanması olduğu için, dizayn genel anlamda incelenmiş ve gemi dizaynında bugüne kadar kullanılan

yöntemler kısaca özetlenmiştir. Bilgi tabanlı dizayn sistemi geliştirilmiş ve bilgi kurallara dönüştürülerek bir Uzman Sistem programı yazılmıştır.

Bir Uzman Sistemin geliştirilmesinde en önemli parametrelerden birini bilgi tabanı oluşturmaktadır. En büyük problemlerden bir tanesi de uzmanları kullanarak bilgi tabanını oluşturabilmektir. Bu amaçla kullanılan çeşitli yöntemlerden biri de mülakat tekniğidir. Bu çalışma çerçevesinde uzmanlar ile yapılan mülakatlar, kaynak kitaplar ve teknik kurallar kullanılarak konteyner gemisi dizaynı için bir bilgi tabanı oluşturulmuştur.

Mimarlık ve Endüstri Mühendisliğinde, yerleştirme yapmak ve yerleştirmenin optimizasyonu için çeşitli yöntemler geliştirilmiştir. Ancak bu yöntemler gemi dizaynında yaygın olarak kullanılmamaktadır. Bu çalışma çerçevesinde endüstri mühendisliğinde kullanılan tesis yerleştirme yöntemleri incelenmiş ve gemi yaşam mahalli yerleştirmesine uygulaması yapılmıştır.

Bilgi tabanının oluşturulması için on uzmanın görüşü alınmış ve buna SOLAS (Safety of Life At Sea), ILO (International Labour Organisation) ve Alman Liman kuralları (SBG) eklenmiştir. Uzmanlardan alınan sağduyu (heuristic) bilgisi ve diğer teknik bilgiler kurallara dönüştürülerek, sekiz bölümden oluşan bir CLIPS programı yazılmıştır. Bu program ALDES olarak adlandırılmıştır. ALDES kullanılarak bir konteyner gemisinin ön ana boyutları, veri tabanındaki gemilerden elde edilen parametrik formüllerle hesaplanmış, ana bölmeleri yerleştirilerek yaşam mahalli dizaynı gerçekleştirilmiştir. Kamara, çeşitli özellikleri olan bir obje olarak tanımlanmış, ve bu özellikler sağduyu bilgisinin mantık programlaması sayesinde tek tek oluşturulmuştur.

ALDES programı, herhangi bir acil durumda kaçış mesafesi bulmak amacı ile, çeşitli senaryolar için çalıştırılmış ve kamara kapılarının yerlerinin değiştirilmesinin kaçış mesafesine etkisi incelenmiştir.

Deniz kazalarında can güvenliğinin en güncel olduğu şu günlerde, yapılan bu çalışmanın daha optimum ve güvenli gemi dizaynında faydalı olması umulmaktadır.

Yapay Zeka ve Uzman Sistemler gemi inşaatında çok yaygın olarak kullanılmamaktadır. Ancak bilgi tabanının bu kadar büyük ve dinamik olduğu bir bilim dalında, bu çalışmanın devamı olarak Uzman Sistem uygulamaları arttırılmalı ve gemi endüstrisinin kullanımına sunulmalıdır.

Gemi dizaynı probleminin karmaşıklığı gözönüne alındığında, bu çalışma göstermiştir ki, çeşitli kabuller ve kısıtlar altında Uzman Sistemler gemi dizaynına uygulanabilir.

# **1. INTRODUCTION**

## **1.1 Introduction**

Scientific and engineering disciplines have continuously been in search of new techniques to solve problems which cannot be solved effectively by the currently available methods. A transition from deterministic to stochastic methods, or from procedural to heuristic methods is recently observed to tackle these problems. Consequently, Artificial Intelligence (AI), Expert Systems (ES), and Fuzzy Logic techniques etc. have attracted worldwide interest and have been applied to various subjects.

Ship design applications have been carried out mainly by human expertise and deterministic analysis techniques. Application of stochastic methods and formalisation of heuristic methods in ship design have not been widely utilised yet. This work is intended to investigate the applicability of a heuristic method, namely ES approach, in ship design. A method consisting of three steps; reviewing available techniques, deriving a suitable approach, and assessing it with a case study, has been utilised on the subject.

## **1.2 Problem Definition**

The current work aims to solve “A Container Ship Design Problem”. There are many approaches such as case based, decision making, multi-criteria decision making etc. applied to the problem. In this work a knowledge based method, supported with a database, was developed to examine the suitability of the method to the problem domain. As ship design problem has a large domain, few specific problems of the domain such as calculating main dimensions, locating main transverse bulkheads and generating accommodation part etc. were selected for the current application. The main focus was on the accommodation design of a container ship. The assumptions of the problem domain can be summarised as follows:

- capacity and speed of the ship are known,
- the principle dimensions can be calculated from a database,

- hullform is assumed to be known,
- height of the double bottom and location of the main transverse can be calculated,
- height of the bridge and number of the decks can be found by using heuristic rules,
- number of the crew can be calculated according to STCW (Standards of Training, Certification and Watchkeeping for Seafarers) regulations,
- the compartments can be located by an Expert System program.

A thorough review of ESs, with an aim to assess the advantages, disadvantages and limitations for ship design application, indicated that ship design can be suitable for an ES or knowledge based system application. Ship design requires a large dynamic knowledge base which mainly consists of heuristics modelling of such knowledge in computer applications can be dealt effectively with an ES.

As ship design is a broad application area, a case study approach has been utilised with a limited knowledge base. Feeder container ship design has been chosen as case study with a payload range of 650 to 1500 TEU. Extension of the current approach into all container ship types shall not be persuaded in this work and shall be considered as a future study topic.

ES development intends to improve CAD capabilities to collaborate intellectually with the designer during the preliminary design stage.

In addition to the main goal of investigation of ES suitability into ship design domain, following sub-goals are defined:

- i. to assess some of the contemporary ship design approaches,
- ii. to present an approach in the utilisation of ESs which allows improved collaboration in the design stages,
- iii. to acquire expertise about container ship accommodation and represent knowledge using a rule-based knowledge representation technique,
- iv. to develop or derive a formal method in the layout design,
- v. to support the system by visualising the end product by using a CAD program,

- vi. to make recommendations for any required developments in order to improve further ES use in ship design.

### **1.3 Method of Study**

This thesis consists of seven chapters in addition to four appendices. The state of the art of design, ship design, AI and ESs, and plant layout techniques are discussed in Chapter 2 through Chapter 4. Chapter 5 and 6 depict the structure of ALDES (Accommodation Layout Design Expert System), the knowledge base and the application of the method.

In Chapter 2, the design process is explained by presenting the nature of design, design stages, and models of design. Knowledge based design model concepts; knowledge representation, reasoning and syntax are reviewed. Differences between generative knowledge (syntax) and interpretative knowledge are outlined to demonstrate a knowledge based design model.

Ship design can be defined as an iterative, complex, multidisciplinary, highly technical and analysis oriented engineering problem. The historical development of ship design is also presented in Chapter 2 by reviewing ship design methodologies from Evan's spiral to contemporary computer methods. Ship design stages are also examined to clarify the application of these methods. Finally, knowledge based design work is evaluated to assess the feasibility of the current work.

Chapter 3 describes the area of computer science known as AI. An aspect of AI Expert or Knowledge Based System is dealt with in particular. A collection of the literature concerning AI and ESs is presented in order to introduce the historical development of AI. AI promises that computers will be able to provide design help instead of being limited to clerical and numerical work.

An ES approach is utilised in the current ship design method. The structure and components of ES, i.e. knowledge representation, inference, explanation facility, knowledge elicitation are described and analysed by ship design point of view. Detailed discussion of the knowledge elicitation is presented, as this task is the main bottleneck in establishing an ES.

Formal layout design methods have been already developed in Architecture and in Industrial Engineering. In Chapter 4, the plant layout problem is presented by reviewing plant layout techniques and methods. Systematic Layout Planning technique and its procedural steps are examined to develop a method for ship layout

application. Computerised facility layout techniques are also described to demonstrate the logic. Layout design of a ship follows a hierarchical decomposition of ship hull into forepeak, cargo holds, engine rooms, aftpeak and super structure. Each hierarchical part is then analysed independently by using topological relations. This analysis is usually performed by using past design cases by abstraction, and generalisation. The approach is outlined to be utilised in an ES application.

In order to compare the success of a layout, a cost function must be described. Definition of cost functions due to layout and improvement of layout by use of cost functions are also given in Chapter 4. Ship design cost functions are examined in details.

The overall capability of any ES is determined by the quality of the associated knowledge base. The main problems associated with the development of ES applications are mostly concerned with the process of obtaining the information required to construct the application of a specific knowledge base. This process of extracting knowledge from either human or non-human sources is often called knowledge elicitation, and is of fundamental importance in the development of ES applications.

Chapter 5 describes the knowledge base which has been developed for assisting container ship layout design. The content and structure of the knowledge base are described in detail to present the methodology. Knowledge base is focused on two aspects; firstly hierarchical decomposition of a container ship is established by presenting methods for main attributes of each division. Secondly, the accommodation block has been analysed in detail and both generative and interpretative knowledge base are presented. A study of human and material flow has been presented and a relational chart has been provided to establish topological relations of accommodation block.

An ES called ALDES for container ship accommodation layout has been developed to investigate the applicability of this approach into ship design as a pilot study. ALDES is developed on production rule based ES approach using CLIPS ES shell. Details of ES utilisation is presented in Chapter 6. The tools used during the development of ALDES, which include ES, plant layout and visualisation, are explained in Chapter 6 as well.

The last chapter, conclusions of the current work, and recommendations, is divided into three sections: firstly the experience gained in developing ALDES is presented, secondly the advantages, disadvantages and limitations of ALDES are

presented to establish a critical assessment of the current approach. Finally some future work is pointed out to improve ES utilisation in ship design.



## **2. GENERAL DESIGN AND SHIP DESIGN**

### **2.1 Introduction**

The result of a design process is a specification of a proposed object to fulfill a predefined set of requirements within a set of environmental, regulatory, technical, economic, social, ethic and physical constraints (Duffy, 1986). Design of an engineering artefact is usually carried out by an analysis-synthesis-evaluation cycle. In the design of integrated systems, an overall analysis is usually not possible and the solution is divided into a number of manageable parts. These parts can then be analysed individually and combined to give an overall solution. If the individual parts are not fully independent, integration stage is performed in an iterative method. As ship design involves a wide range of technologies, the overall task of integration is very demanding. Another difficulty of the ship design task is caused by the design requirements and constraints that are virtually unique to the ship design.

Evan's design spiral was the first structured methodology in the ship design (Evans, 1959). As the technology is improved, the limitations of this method were realised. The computer assisted techniques were consequently developed and utilised as universal tools by engineers and scientists. At first, computer technology was used for simple applications facilitating tedious manual tasks in the ship design. But lately, researchers have been developing models based on AI techniques and domain knowledge as well as the procedural analysis approaches by the use of computers.

The aim of this chapter is to present an understanding of the design process and its application into ship design. A brief history about the methods, which have been applied to ship design is given and their characteristics are discussed to present the current state of the art in ship design.

### **2.2 The Nature of Design**

In approaching and solving a problem, one of the initial challenges which the designer faces is to understand the nature of the design process of the problem in a specified domain. Systems have many components, each of which is related in different ways to the other components through their characteristics. The designer's

task is to create a specification for such a system, for a given set of required functional objectives to be achieved in a given environment.

Design is a purposeful activity and it involves a conscious effort to arrive at a state of affairs in which certain characteristics are evident. Design is also the process of originating systems and predicting how these systems will fulfil objectives. The designer has an end in mind at the beginning of the process but this end does not constitute the final artefact. It is merely a description of it from which predictions of performance can be made. Therefore, a design is an abstraction (Coyne et al., 1990). It produces a description of an artefact able to be interpreted by some other agent for purposes of manufacture or construction. A design could therefore consist of a set of instructions or some other abstraction suitable for interpretation and realisation as a physical artefact.

More specifically, design has been usefully described as a goal-directed activity that involves taking decisions. It can be translated into a process of producing a set of descriptions of an artefact that satisfy a set of given performance requirements and constraints. Design begins with the realisation of a need (a dissatisfaction with existing conditions) and a realisation that some course of actions must be taken. A goal is set, which is for the attainment of a state of satisfaction in response to this need. The ability to set goals and plan for their attainment is a powerful and essential human capability.

For a start, it is difficult to specify the nature of the desired state. Goals themselves are the products of processes that are little understood. Different designers would set different objectives. Given an agreement on objectives, fewer would agree on the way to reach these objectives.

Furthermore, goals appear to be a product of what is possible. Goals rarely behave as autonomous entities that can each be realised independently. Goals interact hence all the design goals can not be achieved where the satisfaction of one obviates the satisfaction of another. This is not simply an operational problem. It cannot be known what is possible until some kind of exploration has been undertaken. Design sometimes appears as an activity of exploration, in which partial responses lead to the redefinition of goals.

Design also appears to be ill-structured in the sense that there is no straightforward process to be followed. A designer may start with goals and work towards the design objects that will satisfy these goals, or the designer may begin with the objects and check whether the properties of these objects match those of the

requirements. There also appears to be no fixed starting point. A designer may follow a certain line or may attack the problem at different points.

Goals are not the only means of deciding on the form of an artefact. Designs often appear to be a response to some invisible system of conventions as much as to stated objectives. Some conventions may be peculiar to individual designers, and save them from a great deal of reinvention. These conventions are often referred as style.

### 2.3 Design Stages

The design has various stages; e.g. concept, preliminary, analysis and detailed design. The beginning and end of these stages are not clearly defined. They overlap and gradually build and fade in levels of activity. The duration of each stage varies between designs, depending upon the product and the progress through the design process. The design stages can be explained as follows:

- a) **Concept Design:** this stage of design includes rough ideas, which may satisfy a set of requirements. The designer gets the idea from a market review or from discussion with an owner. The concept design stage is concerned with the generation and choice of these ideas. When the design requirements have been detailed, the designer should have identified the types of solutions, e.g. containership or a new innovative design (Duffy, 1986).

At the end of the concept stage a rough indication is given for possible design parameter values, in which the final solution is likely to lie, may be known, and prospective solutions reasonably defined for the next stage.

- b) **Preliminary Design:** this stage concentrates upon the further development and evaluation of the solutions from the concept design stage. During this stage, feasible design solutions are generated and a best design selected to proceed to the detailed design stage. Design parameter values for a ship such as length, beam, and depth are varied and their effects are checked against specified requirements, such as number of containers and stability. Depending on the results the design parameter values may be modified again. The process is repeated until a satisfactory approximate solution is reached (Duffy, 1986). Comparison with previously defined solutions results in a single design being chosen to carry into the detailed design stage. At the end of the preliminary design stage an approximate design solution is defined.

- c) Analysis: The information from the preliminary design is investigated by means available analysis methods satisfy design goals and constraints. The solution is refined in detail until a final solution is reached.
- d) Production design: Production analysis is carried out from product definition, obtained in previous step, to produce manufacturing methods and continuous refinement is carried out until final build stages.

The design process is iterative in its nature. This iterative process requires multiple decisions and computations to allow the progression, control and checking of the prospective solution.

## **2.4 Design Characteristics**

The design process is complex and requires knowledge and understanding of the requirements, the design domain, the intended environment, production techniques and many other related topics.

Design can be considered to have the following significant characteristics:

- Creative: it requires imagination to build models. As a result of creative activities, the form and structure of these models may change or develop as the design proceeds.
- Multiple solutions: there can be many answers to a given design problem, all of which may achieve the objectives, and may thus be technically and economically feasible. Thus the design problem is not deterministic.
- Empirical: the process of creating and evaluating a model does not always follow well formalised rules with a good theoretical background. Very often relationships are of an empirical nature.
- Approximate: since the design is a modelling process, which uses empirical relationships, the results obtained are generally approximate. Accuracy increases as the design proceeds and greater levels of detail are included. However, the concept of expected and acceptable accuracy is important to the designer.
- Requires expertise: the designer uses his/her expertise in many different ways during design with respect to relationships used, in the actual design process, and even in the judgement of the acceptability of proposed solutions.

- Measure of merits: in a design problem, the designer may be required to access the feasibility of a proposed design or evaluate alternatives that are under consideration. Basically, this involves an examination of costs, benefits, and consequences of the design or the alternatives over a period of time.

## 2.5 Models of Design

There are two major approaches to increase understanding in a particular discipline: case studies and models. Case studies are particularly important in discussion of architectural and engineering design. Some design phenomenon such as design of a particular ship, is observed and analysed, in an attempt to extract general principles.

In the second approach, models as limited abstractions of particular phenomenon, less ambitious than theories, aimed to explain and predict within a subset of connected phenomenon.

The simplest model of design is based on the observations of the phenomenon and deriving the models based on empirical method (empirical model).

Problem solving model starts with the problem statement, a hypothesis or conjecture is generated and tested by trial and error analysis (generate and test). Hence problem solving model can be characterised as a process of search.

Another model of design process is optimisation aimed to control search process. This method starts with definition of design goals such as suitable size, economy, referred as objectives, objectives must be satisfied for the goals to be met. Variables such as size, cost, called performance variables, must acquire values within ranges called constraints in specific terms, or criteria in general directional terms. A designer can assign values to decision variables, such as material thickness, types of components which collectively describe the design. By making decisions, the designer controls the values of performance variables to satisfy the objectives. The relationships between the performance variables and the decision variables are not always obvious which may be a type of black box. Optimisation seeks not only a design but the best possible design.

Meanwhile simulation is a problem solving approach to design in which a design solution is postulated. That is the decision variables are posited and their values are given. The performance variables are obtained by calculations or by other means. Their values are examined or evaluated against a given criteria to determine

whether they are satisfactory. If they are not satisfactory the decision variables are modified and the process is repeated until a seemingly satisfactory result is obtained. Hence in simulation there is no way of feedback from performance variables to decision variables.

Logic can also be considered in formulating design. Design can be assumed essentially reasoning process by using logic to provide insight into design process. Hence design in a logical scheme can be assumed as a process of making logical deductions about a set of requirements. Design starts with design requirements, axioms and proceed along the course of reasoning to the design, a theorem. The procedure is incremental and understanding of design is improved on every stage. This approach can be called design by deduction.

Another model of the design is to use an analogy to natural language which is a configuration of components governed by rules. The linguistics can be used as an operational model relevant to the generation of design.

## **2.6 A Knowledge Based Model of Design**

Knowledge based design is a model of design process in computer environment in an attempt to capture and render operable human knowledge about the domain. The goal is to represent knowledge in such a way that it is comprehensible to both human and the computer.

A knowledge based design model is based on three concepts (Coyne et al., 1990):

1. Representation- how we represent information in a computer system.
2. Reasoning- what is understood by design reasoning.
3. Syntax- the role of syntactic knowledge in design and computer systems.

### **2.6.1 Representation**

Representation of information in computers can be based on representation of components and relations, i.e. facts, generative and interpretative statements to manipulate the facts, i.e. knowledge and lastly strategy to guide reasoning process, i.e. control (Coyne et al., 1990).

### 2.6.1.1 Facts

There are two types of information about components of an artefact, which are objects and relations. These can be used to organise information pertaining to design in a computer system.

Objects are simply units of information which may correspond to physical or non-physical components. Relations are labels attached to groupings of information units. Relations may also map onto relationships between physical objects. Relations can refer to topology as well as grouping of objects. There are several ways of expressing this type of organisation of information into objects and relations using symbols. One useful method is to enclose the names of objects that belong to a group in parentheses:

NextTo(MasterCabin, OwnerCabin).

These objects and relations grouping will be regarded as facts. They are translated into natural language as follows:

The master's cabin is next to the owner's cabin.

Although these facts refer to physical objects and the physical relationship between them, facts can also be about other information units and their relationships, such as objects and attributes of various kinds. Numbers are allowed as objects in this notation. Also, a fact can have a value as true or false. A fact is true because it is declared to be true, or it can be true because it is consistent with some knowledge that determines its truth status.

This model serves as a method of database organisation (Coyne et al., 1990). It forms the basis of a relational database. It can also be argued that it underlies all other methods of organisation. Essentially these statements are in a computer system which represents facts about the world. Facts can be described in terms of units of information and the relationships between these units.

### 2.6.1.2 Knowledge

Objects and relations can be regarded as the primitives of an information processing system. Although objects and relations are very simple notations, they are powerful enough to provide a lever into understanding of knowledge (Coyne et al., 1990). Knowledge can be characterised as statements about mappings between facts. These mappings enable the system to derive new facts from existing ones.

There are many ways of representing these mappings. One of the most useful methods is to define a series of special relations that denote logical connections such as and, or, if between facts. An example of representing knowledge with this mapping is as follows:

IF (GRT  $\geq$  25000) AND (R = LONG-DISTANCE)

THEN INPUT

(OCEAN-GOING-MASTER 1)

(OCEAN-GOING-CHIEF-OFFICER 1)

(OCEAN-GOING-WATCH-KEEPING-OFFICER 2)

Knowledge as mapping between facts is the same idea as the idea of knowledge as generalisation. A generalisation is a statement about a class of objects that pertains also to members of that class. Generalisation represents an invaluable economy in the storage of information. As example of generalisation as a clausal statement is:

Ship(x) If Containership(x)

This says that an object x belongs to the class of objects called Ship if it belongs to the class of objects called Containership (Coyne et al., 1990).

Knowledge has been discussed as logical clauses. The advantage of the clausal form is that it makes explicit the idea of knowledge as statements about relationships between facts, and avoids the issue of how knowledge is processed. The processing of knowledge can be seen as an issue independent of the domain to which the knowledge belongs. It is, of course, possible to consider knowledge in terms of computational procedures - as programs containing control structures. The clausal form serves to highlight the essential nature of knowledge apart from the control component of computer programs.

### **2.6.1.3 Control**

Facts have been characterised as statements about relationships between objects, and knowledge as concerned with the relations between classes of facts. Control is essentially concerned with mappings between items of knowledge. Control can be explained in terms of rules. Rules are explicit statements mapping facts.

Control is an important aspect of design knowledge that can help to make design task tractable. The built-in control of automated reasoning systems provides control for a language such as CLIPS. Coyne et al. (1990) identified several important control tasks that are desirable to the model:

- Search strategies: design is a process that attempts to encounter a description in the design space that maps onto some intended performance in the interpretation space. The process of producing design involves progression from an incomplete, unresolved, and contradictory description of a design to a complete description. This process can be regarded as search.
- Goal satisfaction: the generation of design is an artefact description. The generative component of design is goal directed. It means that design starts with some statement and attempts to prove if it is true or false. That approach is called top-down strategy. Sometimes it is required to start from a design goal and end up with a true statement, which is called bottom-up strategy.
- Multiple knowledge sources: knowledge can be acquired from many sources in design such as external storage media in the form of texts, codes, and diagrams, as well as teams of experts. In accounting for control knowledge in design systems, the knowledge which a designer organises and gains access to knowledge from dissimilar sources must be modelled.
- Multiple control levels: design is a process in which certain actions are brought to bear in manipulating and assembling components. There are control actions for manipulating and assembling design actions. This suggests a hierarchy of control abstraction in which the actions of one level become the objects of the controlling level above.

### **2.6.2 Reasoning**

There are three types of logical reasoning; deduction, induction, and abduction. Logical deduction is the principle mode of reasoning with certainty. This type of reasoning lends itself to verification and can be readily formalised. The syllogism provides a convenient vehicle for talking about the essential characteristics of reasoning. An example of syllogism is demonstrated in Figure 2.1.

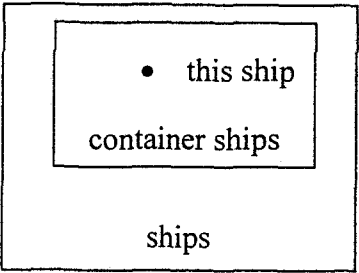


Figure 2.1 A syllogism represented graphically

- 1. This is a containership
- 2. All containerships are ships
- 3. Therefore this is a ship.

The components of syllogism have been variously labelled as 1-case, 2- rule, and 3- result. In this case it is convenient to label them 1- fact, 2- knowledge, and 3- inferred fact (Coyne et al., 1990).

The third statement can be regarded as an interpretation of the first two statements. The interpretation of a set of logical statements is simply that which can be logically inferred from them. In terms of knowledge based systems it can be said that an inferred fact is a statement that is implicit in the statements constituting the facts and the knowledge of the system.

Deduction is the main method of formal reasoning systems (Coyne et al., 1990). The other two modes of reasoning are induction and abduction. Table 2.1 describes these three types of reasoning.

Table 2.1 Three types of reasoning in relation to the syllogism (Coyne et al., 1990)

|           |      |   |        |   |        |
|-----------|------|---|--------|---|--------|
|           | (1)  |   | (2)    |   | (3)    |
| Deduction | Case | + | Rule   | → | Result |
|           | (1)  |   | (3)    |   | (2)    |
| Induction | Case | + | Result | → | Rule   |
|           | (2)  |   | (3)    |   | (1)    |
| Abduction | rule | + | result | → | case   |

Induction is the method to derive the rule from case and result, i.e. knowing "This is a containership" and "It is a ship" a rule indicting "All container ships are ships". Hence induction is regarded as the process in science by which theories are derived from observations of certain phenomena.

A third mode of reasoning is called abduction. Abduction is the derivation of statements about the world given logical rules and some logical consequences. In syllogism, abduction would be operating if the rule "all containerships are ships" and the conclusion "this is a ship" were known but did not have the original statement "this is a containership". This type of reasoning is not certain.

### **2.6.3 Syntax**

As a consequence of reasoning process, new facts are created, or relations are defined. Additionally rules can be rewritten by the reasoning process. According to computational theory, rewrite rules form the basis of a grammar system that defines the syntax of a language. This similarity between language and design can be used as an analogy. Components, such as rooms, furniture, stairs form vocabulary, rules form the grammar, and the design description forms the sentence. The meaning of a sentence is not a concern in syntactic system. Hence the performance criteria are not utilised in generation step.

Syntactic system utilises the rules to generate the design system by using a deductive reasoning process.

### **2.6.4 Design Space**

Knowledge in design process can be categorised into generative and interpretative knowledge.

Generative knowledge is concerned with forming a design description from components and relations by utilising syntax rules. Such syntactic system must make sure that the components can be assembled by using topologic relations and actions such as moving, sizing, and rotating.

Interpretation of a design is the process of obtaining performance description. This step is usually performed in a deductive style although the rules may be very complex (such as finite element techniques, boundary element methods, or an empirical model). Interpretation is similar to meaning of a sentence in a language system, and may be called semantics.

This division between semantics and syntax (or interpretation and generation) forms the mapping between space of designs and space of interpretations.

One simple representation of design can be described as generation of space of designs by using generative knowledge and interpretation of all this space by using interpretative knowledge to choose which part of space of designs satisfies the design requirements and criteria (Figure 2.2).

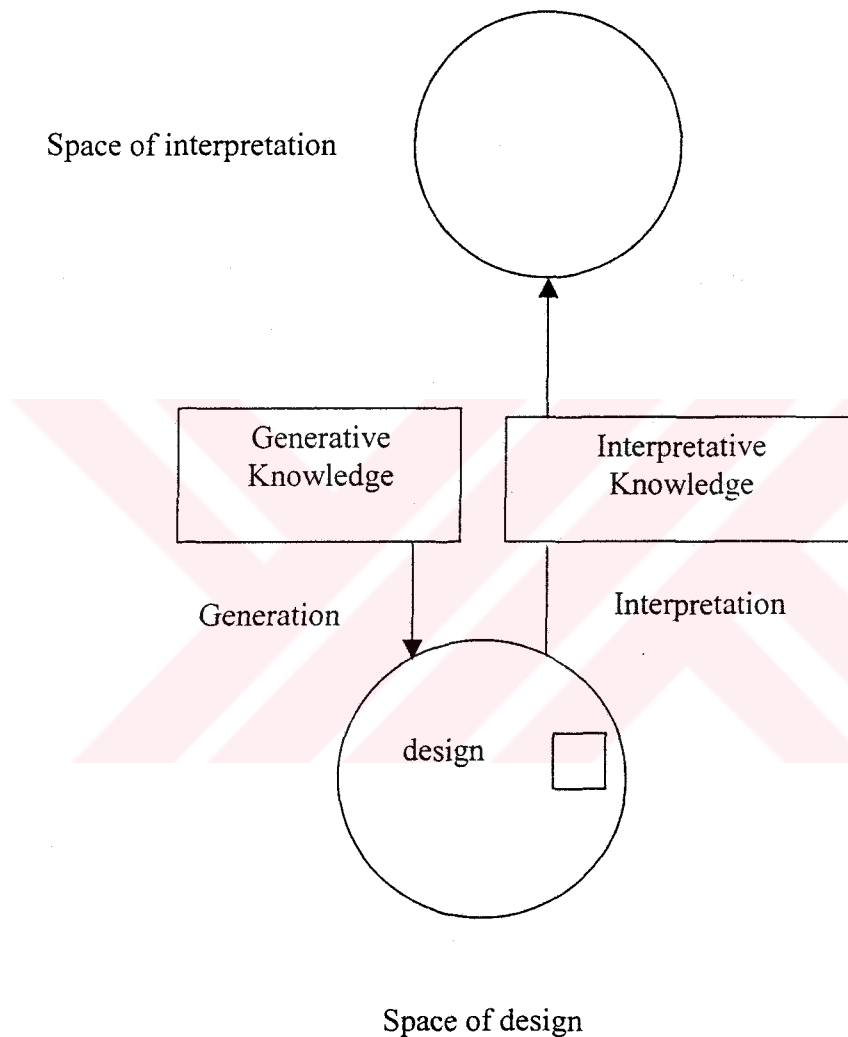


Figure 2.2 A space of design (Coyne et al., 1990)

This generate and evaluate model of the design is unpracticable as space of designs will be very large to evaluate. Both generation and interpretation process uses deductive reasoning in this model.

Alternative model of design should employ other logical reasoning mechanisms other than deductive process. Figure 2.3 shows such a model. Space of designs is an intersection of design descriptions generated by generative knowledge through deduction, and design descriptions obtained by interpretative knowledge from performance specifications by abduction.

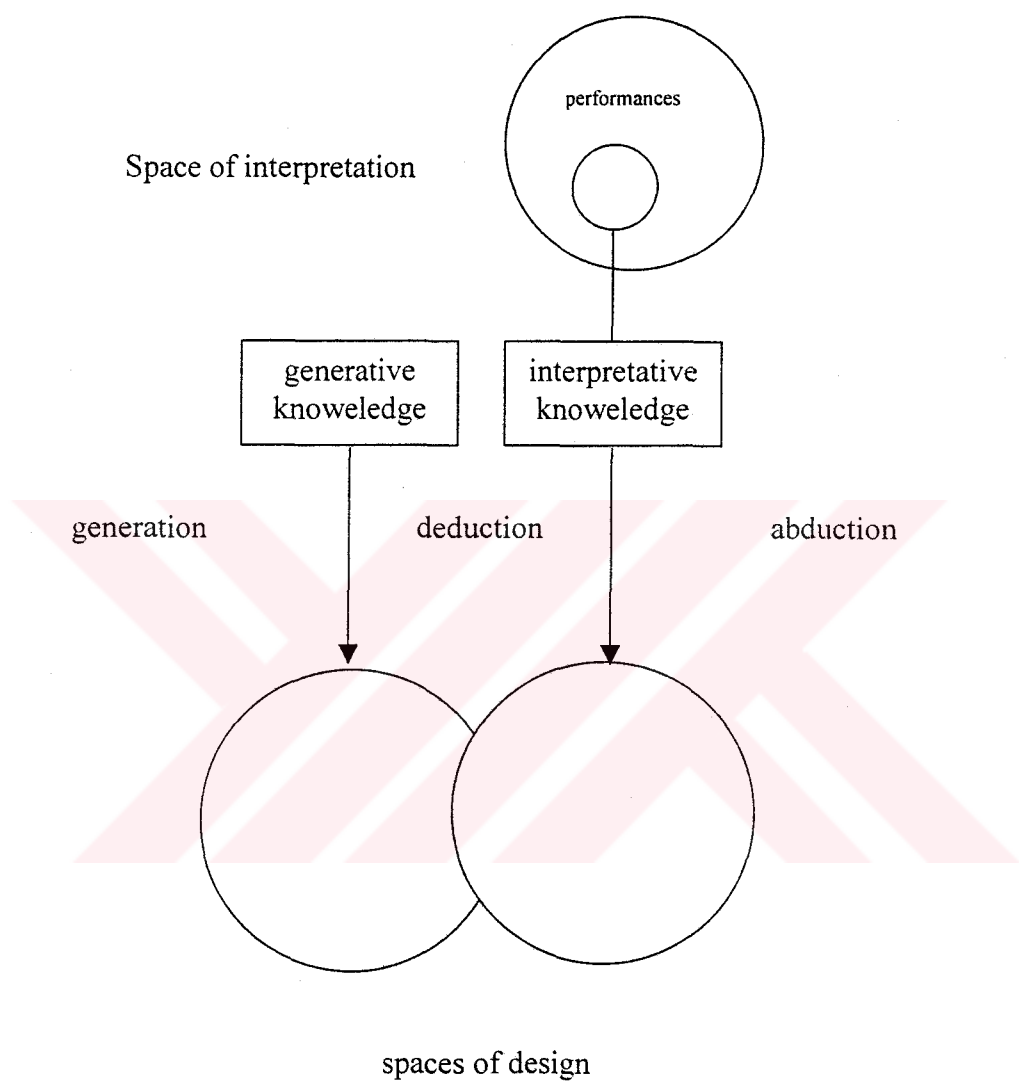


Figure 2.3 Generative and interpretative knowledge (Coyne et al., 1990)

**2.6.4.1 A Design Syntax**

The syntax of a design process can be described as assembling components extracted from a computer database. This database would contain representations of facts about the design. These facts will be such that they can be translated into a final artefact. The attributes of components such as sizes, materials and spatial relationships should be defined. However not everything will be known about the candidate components of the beginning.

There are also rules, or knowledge, about how these components normally fit together. It can be described as generative knowledge. Generative knowledge can be represented in terms of actions for putting components together. Such actions might include various geometrical transformations, or more specifically actions which connect components together. It is possible to represent actions as statements about objects and relations.

The objects may also be represented as variables. Actions serve to facilitate the generation of configurations of components. They therefore define a space of designs. They constitute the grammar of a syntactic system.

For example a house design is considered as components such as walls, windows, and the spaces bounded by walls constitute a loosely-defined vocabulary. The components can also be considered to be primitives of lines, points and labels appearing in space, or composites of these forming rectangles, triangles, parallelopipeds and spheres. The syntactic knowledge about design may be characterised as that concerned with the configuration of these components. The full range of geometrical operations can be considered as defining the syntax of a design system - involving the union, intersection and difference of components - but such a syntax is likely to produce a design space that is too large for practical consideration. Legal operations may also define all possible ways in which rooms can be combined. In order to be useful, a designer's knowledge would appear to be restricted to a subset of those operations - that is, a more specific syntax.

#### **2.6.4.2 A Design Interpreter**

The knowledge is considered pertaining to interpretation. This is the knowledge by which the performance of an artefact can be inferred given some description in a computer database. Such knowledge can be represented in many ways including the use of logical clauses.

The performances of the artefact may be known. There may be certain attributes which the artefact must exhibit, such as compliance with a cost threshold, dimensional and weight constraints, speed of operation, tolerances in calibration, accuracy of the display and ease of use. The interpretative knowledge is that which enables these behaviours and attributes to be inferred from the design description.

A design system is expected at least to contain the knowledge concerned with the generation of designs and that concerned with their interpretation. The design task is then to produce a description of an artefact that conforms to the syntactic

(generative) rules and which also facilitates the required interpretation. The design process can be modelled as one in which the syntactic rules serve to generate configurations of elements that are then interpreted to see if they comply with the intended interpretations. This constitutes a generate and test approach. With design tasks of any complexity this approach cannot be of much assistance. Clearly, further knowledge is required.

Design codes are a pervasive form of interpretative knowledge. A particular performance we may be interested in is that a design is safe and habitable. They do not generally tell how elements may go together to produce a design (syntax), but rather embody the knowledge by which a design is to be interpreted as being 'safe and habitable'.

#### **2.6.4.3 Representing Design Knowledge**

Syntactic knowledge is essentially concerned with the configuration of elements. Grammatical rules, or actions have been discussed to delineate syntactic knowledge. But design knowledge can also be depicted in terms of generic descriptions, as prototypes. Syntactic knowledge can be represented in terms of type descriptions and some body of knowledge by which this generic description can be instantiated-translated into the description of a design instance. This knowledge could simply be embedded in the prototype description, in terms of acceptable ranges of possible values. Such generic design descriptions are used informally in design manuals and guides.

Interpretative knowledge can be represented in terms of generic design descriptions. Descriptions can be used as diagrams, and diagrams imply configurations of elements. The difficulty with translating interpretative knowledge into generic descriptions is that interpretative knowledge generally embodies many options.

#### **2.6.4.4 Design Processes**

Design knowledge is characterised in terms of statements about mappings between facts and as generic descriptions. There are classes of design processes and procedures about design processes. It is possible to treat operations in the design process as objects and bring knowledge to bear in reasoning about these objects.

Knowledge of the control of a logic system can be represented in terms of relationships between rules. Rules are treated as objects that are manipulated by

means of meta-rules. Generative rules can be seen as actions which change design descriptions in some way. Facts about actions may constitute 'vocabulary' elements in a design control system.

It is possible to devise a system which controls a design system as one in which the objects are actions, and vocabulary elements are manifested as statements about the relationships between actions. There is therefore knowledge by which sequences of actions can be interpreted and there is syntactic knowledge which defines a language of such actions. The sequence of actions generated by such a system would constitute a plan of actions. This system effectively serves as a controller for a generative system.

Designers appear to possess a ready 'vocabulary' of permissible actions: for example, to put objects in place, to move objects, erase them, and make them larger or smaller. In fact the full range of geometrical transformations is available. But actions can also be described in relational terms: for example, to place an object next to another object, or to locate an object inside another object. There are, of course, further transformations by which objects change their meanings, such that abstract geometrical entities become design components.

The control of actions becomes important if design can also be seen as the selection and sequencing of appropriate actions to produce descriptions of artefacts with particular attributes. The ordering of actions inevitably involves the resolution of conflicts between competing actions. Designers seem to adopt strategies which enable them to actively avoid situations where conflicts occur, or to detect conflicts at an early stage when they are relatively easy to rectify.

## **2.7 Ship Design**

The basic ship design refers to determination of major ship characteristics affecting cost and performance (Taggart, 1980). Basic design includes the selection of ship dimensions, hull form, power, preliminary arrangement of hull, machinery arrangement, and major structure. Proper selections assure the attainment of the mission requirements such as good seakeeping performance, maneuverability, the desired speed, endurance, cargo capacity, and deadweight. Furthermore, it includes checks and modifications of the achievement of required cargo handling capability, quarters, subdivision and stability standards, freeboard and tonnage measurements.

A number of major developments were observed during late 1960's and 1970's which had impact on the general basic design problem. Use of computers in

the design was among the most significant developments. While the computer effected how basic design is performed, other changes have impacted on what constitutes the basic design problem. For example, one revolutionary development was the change from breakbulk to containerised cargo in the liner trades. The developments in ship design are explained in a chronological manner in the following section.

### 2.7.1 Ship Design Process

The ship design process closely follows the pattern described in the nature of design. The shape of the ship is important to its structural and hydrodynamic performance and to its aesthetic appeal. The internal and external layout of the ship in both two or three dimensions are of major importance to a successful design. Design of a ship layout is more difficult than the land based structures due to hostile environment, the need for mobility and the possible need to contain both fire and flood. Human factors are also major concerns, not only in the operational and ergonomic sense but also in ship motion effects. The successful designer will try to minimise the incidence of seasickness by paying careful attention to the motion frequencies.

Ye (1991) explains some of the key features of ship design as follows:

- Multiple criteria: there would generally exist several measures of merit for achieving a design objective. Those measures may not be of equal importance in the design. Their relative importance should be properly evaluated so that the feasible design solutions could be balanced to obtain a best compromised one.
- Uncertainties: some important uncertainties, which could have a significant influence in design, must be assessed quantitatively before the final design is decided.
- Iteration: owing to the complexity and uncertainties some design values can only be determined often after the several iterative steps.
- Creativity: creativity is the basic element of ship design. Design requires intelligence and imagination as well as mechanical skills.

There are different steps in ship design. The four steps are amplified further below according to Taggart (1980):

- 
- a) **Concept Design:** the very first effort, concept design, translates the mission requirements into naval architectural and engineering characteristics. Essentially, it embodies technical feasibility studies to determine such fundamental elements of the proposed ship as length, beam, depth, draft, fullness, power, or alternative sets of characteristics, all of which meet the required speed, range, cargo cubic, and deadweight. It includes preliminary light-ship weight estimates usually derived from curves, formulas or experience. Alternative designs are analysed in parametric studies during this phase to determine the most economical design solution or whatever other controlling parameters are considered determinant.
  - b) **Preliminary Design:** a ship's preliminary design further refines the major ship characteristics affecting cost and performance. Its compilation provides a precise definition of a vessel that will meet the mission requirements; this provides the basis for the development of contract plans and specifications.
  - c) **Contract Design:** the contract design stage yields a set of plans and specifications, which form an integral part of the shipbuilding contract document. It encompasses one or more loops around the design spiral, thereby further refining the preliminary design. This stage delineates more precisely such features as hull form based on a faired set of lines, powering based on model testing, seakeeping and manoeuvring characteristics, the effect of the number of propellers on hull form, structural details, use of different types of steel, spacing and type of frames. A weight and centre of gravity estimate takes into account the location and weight of each major item in the ship design. The final general arrangement is also developed at this stage. This fixes the overall volumes and areas of cargo, machinery, stores, fuel oil, fresh water, living and utility spaces and their interrelationship, as well as their relationship to other features such as cargo handling equipment, and machinery components.
  - d) **Detail Design:** The final stage of ship design in the development of the detailed working plans are the installation and construction instructions to the ship fitters, welders, outfitters, metal workers, machinery vendors, pipefitters etc. As such they are not considered to be a part of the basic design process. One unique element to consider at this stage of design is that up to this point, each phase of the design is passed from one engineering group to another. At this stage the interchange is from engineer to craftsman, that is, the engineer's product at this point is no longer to be interpreted, adjusted, or corrected by any other engineer.

---

## **2.7.2 Methods of Ship Design**

In this section some common methods of ship design will be explained. These methods are design spiral, mothership, optimisation, decision based, and knowledge based method, which is used in this thesis.

### **2.7.2.1 Design Spiral Method**

The design process is represented by a design spiral after Evans (1959). The design spiral is shown in Figure 2.4. The spiral representation depicts the very iterative nature of the process. The design starts with the mission requirements and proceeds through the nodes of the spiral towards a detailed design.

In many cases, the typical design spiral becomes over-simplistic and misleading since:

1. design does not proceed in an orderly sequential series of steps,
2. available time limits the possible number of iterations,
3. at different stages of design available information and methods undergo step changes (not continuous changes),
4. sub-system optimisation is encouraged,
5. compatibility of selected equipment and systems cannot be assured.

Ship design spiral approach represents a piecemeal synthesis, it is neither comprehensive nor integrative. Furthermore, during the design evolution and the production and service, demands and constraints tend to derive from the design.

### **2.7.2.2 Mothership Method**

This method is based on a past successful ship or group of similar ships. If satisfactory data on a successful similar vessel is held in the design office, or given in publications available to the designer, sizing can be done fairly rapidly by considering differences between the mothership and the new design. This is simple and reliable provided that the mothership is a reliable real vessel rather than a paper design. However, the method tends to restrict innovation.

---

In this method the most important thing is to find a basis ship which has the minimum differences between the performance requirements of the mother ship and the new ship.

It is a very widely and successfully used method by designers. However, although the end product of the method can be acceptable, most probably it is not the optimal solution. Additionally this method is not suitable to design a novel ship.

### **2.7.2.3 Optimisation Method**

There has been a quite large amount of research on the subject. In this method the design problem is formulated in a mathematical form with an identified objective function, which would be optimised subject to a set of prescribed constraints. The aim of the optimisation process is to minimise or maximise a chosen parameter or parameters.

To use this method, the design problem must be well formulated and be easily understood. Once the mathematical formulation is done, the optimisation method will lead efficiently to an optimum solution. The efficiency of the method is dependent on the power of the computer. The method is carried out usually for one criteria. The design can be optimum for this criteria, but can mislead the designer for the other aspects. Therefore, it can be a very poor approach from an overall design point of view.

### **2.7.2.4 Decision Based Method**

Decision Support Problems (DSPs) provide means for modelling decisions encountered in design. Multiple objectives, quantified using analysis-based “hard” and insight-based “soft” information, can be modelled in the DSPs. For real world, practical systems, in the early stages of design, all the information for modelling systems comprehensively and correctly may not be available. Therefore, the solution to the problem, even if it is obtained using optimisation techniques, cannot be optimum with respect to the real world. However, this solution can be used to support a designer’s quest for a superior solution. In a computer assisted environment this support is provided in the form of optimal solutions for DSPs. Formulation and solution of DSPs provide a means for making the following types of decisions:

- Selection: the indication of a preference, based on multiple attributes, for one among several feasible alternatives.
- Compromise: the improvements of a feasible alternative through modification.

- Coupled or hierarchical: decisions that are linked together - selection/selection, compromise/compromise and selection/compromise decisions may be coupled.
- Conditional: decisions in which the risk and uncertainty of the outcome are taken into account.
- Heuristics: decisions made on the bases of knowledge base of facts and rules of thumb; DSPs that are solved using reasoning and logic only.

#### **2.7.2.5 Knowledge Based Method**

The use of this method is not very common in the field of ship design. The purpose of this model is to enable computers to assist in the design process. The knowledge based model of general design is explained earlier in this chapter. The application of the model into ship design will be given in Chapter 6.

The method provides a central memory for knowledge acquired from a procession of human experts over the years or for knowledge possessed separately by a number of individual human experts at any one time (Ye, 1991). One of the most important advantages is that heuristic knowledge can be modelled by using this method. On the other hand it is quite difficult and time consuming to construct the knowledge base in a clear and logical way. It is difficult to represent the general design knowledge.

### **2.7.3 Overview of Ship Design**

The historical evolution of ship design, briefly summarised in Table 2.2, is explained in this section. In the table some models of ship design is highlighted.

Evans introduced the design spiral in 1959 into ship design. The spiral shows the very iterative nature of the design process (Evans, 1959). Figure 2.4 illustrates the design spiral. Buxton (1972) introduced economics into the design spiral. Andrews represented the ship design process as a helical corkscrew in an attempt to account for the open nature of the design process (Andrews, 1981). Figure 2.5 shows Andrews' spiral.

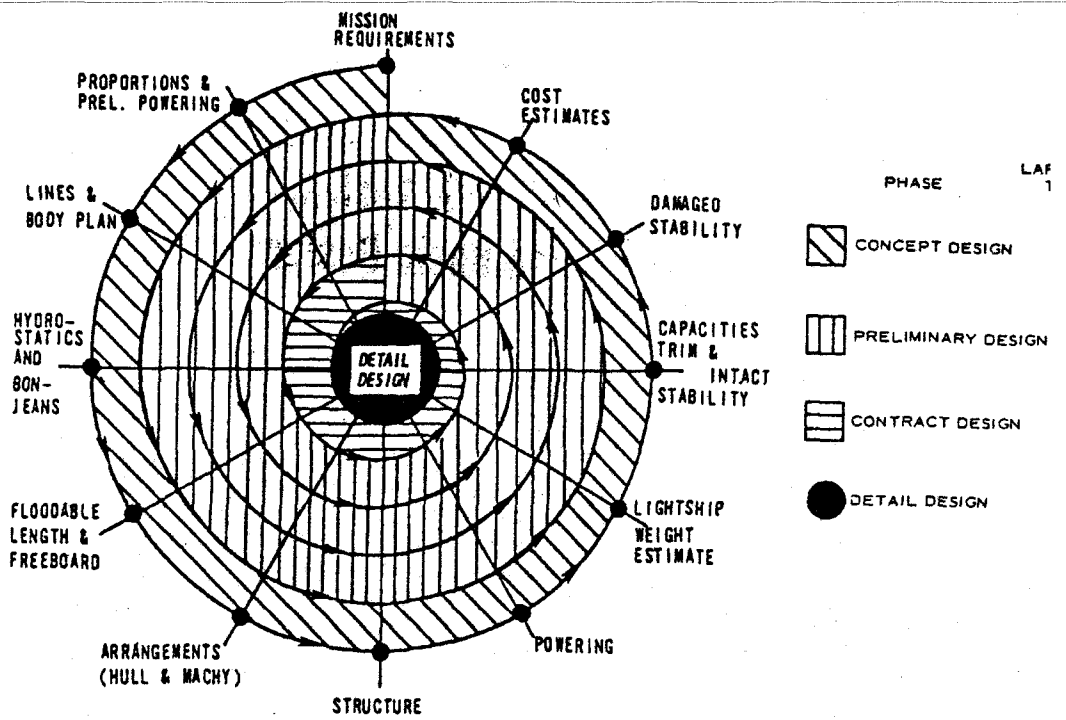


Figure 2.4 Design spiral (Taggart, 1980)

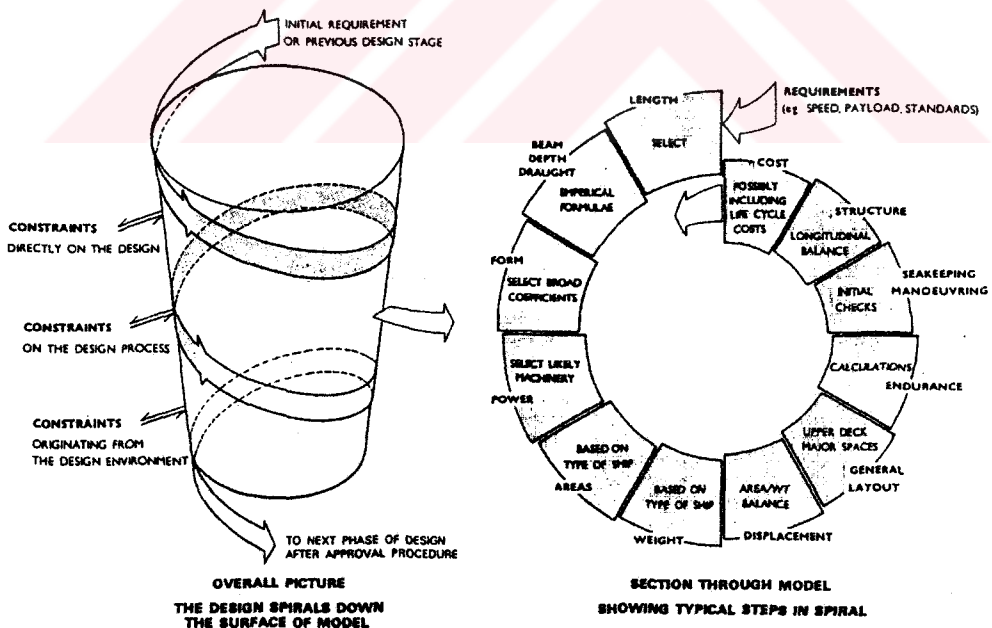


Figure 2.5 Andrews' spiral (Andrews, 1981)

Table 2.2 Evolution of ship design

| MODEL               | WHEN | WHO             | WHAT                                                                |
|---------------------|------|-----------------|---------------------------------------------------------------------|
| Design Spiral       | 1959 | Evans           | Introduced design spiral                                            |
|                     | 1972 | Buxton          | Introduced economics into the design spiral                         |
|                     | 1981 | Andrews         | Represented the ship design process as a helical corkscrew geometry |
| Calculation Forms   | 1969 | Lamb            | Adopted a design procedure by using calculation forms               |
|                     | 1976 | Watson et al.   | Developed calculation sheet approach                                |
| Computer Techniques | 1965 | Murphy et al.   | To determine the optimum combination of L,B,T,C <sub>B</sub> , D    |
|                     | 1978 | Kupras          | Computer graphics                                                   |
|                     | 1978 | MacCallum       | Computer graphics                                                   |
|                     | 1982 | Pattison et al. | GODDESS, CAD related computer program                               |
|                     | 1986 | Kenneth et al.  | Programming technique and ship design logic                         |
|                     | 1987 | Carlson         | GADS, General arrangement design system                             |
|                     | 1987 | Keane           | Computer based hull form concept design                             |
|                     | 1987 | Smith et al.    | Decision support technique using optimisation procedure             |
|                     | 1991 | Koh             | General arrangement design system using CAD                         |
|                     | 1995 | Lee et al.      | Computerised ship design system                                     |
|                     | 1996 | Han et al.      | BASCON, conceptual design system                                    |
|                     | 1997 | Aloson et al.   | Virtual reality                                                     |
|                     | 1997 | Martin          | Virtual reality                                                     |
| Optimisation        | 1966 | Mandel et al.   | Linear optimisation                                                 |
|                     | 1970 | Nowacki et al.  | Non-linear optimisation                                             |
|                     | 1985 | Lyon et al.     | Multi-objective optimisation                                        |
|                     | 1994 | Ray et al.      | Multi-criteria optimisation                                         |
|                     | 1996 | Lee et al.      | Multi-criteria optimisation                                         |
| Expert Systems      | 1982 | MacCallum       | DESIGNER, an ES program                                             |
|                     | 1986 | Duffy           | Added uncertainty in DESIGNER                                       |
|                     | 1987 | Akagi et al.    | An ES using the object-oriented knowledge representation            |
|                     | 1987 | Cort et al.     | Fuzzy sets to develop a space layout design                         |
|                     | 1988 | Sekimoto et al. | An ES using the object-oriented knowledge representation            |
|                     | 1989 | Welsh           | Overall design system                                               |
|                     | 1992 | Van Hees        | QAESTOR, an ES programming                                          |
|                     | 1995 | Akintürk et al. | An ES program in the design of multi hull vessels                   |
|                     | 1996 | Lee et al.      | ES and genetic algorithms in conceptual ship design                 |
|                     | 1996 | Aguliar et al.  | Knowledge-based and object-oriented methodologies                   |
| Other Techniques    | 1990 | Mistree et al.  | Concurrent engineering                                              |
|                     | 1994 | Smith et al.    | Decision-based design                                               |
|                     | 1994 | Erikstad        | Concept exploration                                                 |
|                     | 1997 | Lee et al.      | Case-based design system.                                           |

The use of computers for certain ship calculations became common in late 1960's but most of the programs were incomprehensive and they were usually of the optimisation type for effective control and planning (Lamb, 1969). Lamb adopted a design procedure by using standard calculation forms. Watson (1976) developed another calculation sheet approach after Lamb (1969). He explained the changes in ships from the beginning of the 1960's to the late 1970's.

Murphy et al. (1965) developed a computer technique to determine the optimum combination of length, beam, draft, block coefficient, and depth to satisfy a predetermined "least coast" criterion for a dry cargo ship. In the Late 1970's computer graphics became more common (Kupras, 1978, MacCallum, 1978) and Computer Aided Design (CAD) methods related to CAD started to be used in ship design (Beier, 1978). A number of computer programs were developed like GODDESS in the 1980's (Pattison et al., 1982). The aim of GODDESS was to assist in design formulation, unconventional naval architectural and engineering analysis, and drawing production, for surface warship and submarines, during the concept and feasibility design stages. Kenneth et al. (1986) described programming techniques and ship design logic which allow a ship design computer program to be used for studying impact of using alternative ship design criteria and practices, including different types of hull forms and stability criteria. A General Arrangement Design System (GADS) was developed in 1987 (Carlson and Fireman, 1987). It is a subsystem of the NavSea Computer Support Design system. GADS was derived from the analysis of the general arrangement design process and determines the extent and benefits of applying computer-oriented analytical techniques and information processing capabilities. Keane (1987) presented a computer-based method for hull form concept design. The method is particularly useful for carrying out parametric surveys for new warship classes. Smith et al. (1987) developed a decision support technique, which offers a structured facility for the design of the subsystems and for the modelling of the subsystems. The method, employing optimisation procedures, allows all aspects of the system design to be considered concurrently, to produce the best solution as defined by the specifications. Another application of Computer Aided General Arrangement Design of Ships was carried out by Koh (1991). He used an analysing and merging method for the ship compartmentation. The program is able to do volumetric calculation and the generation of drawings. Lee et al. (1995) developed a computerised ship design system. The system could be used at the basic ship design stages as an efficient design tool. The system consists of a database system, modelling system, application programs and user interface. The application programs perform conceptual design, hull form design, compartment layout design, structural design, resistance and

powering performance prediction, propeller design, seakeeping and manoeuvrability performance prediction, structural analysis and vibration and noise prediction. Han and Lee (1996) developed a conceptual design system which supports the engineering design process and provides not only an integrated design model for co-operative work but also a flexible model to cope with a changing environment (BASCON). Virtual reality tools have started to be used in ship design by late 1990's (Alonson and Garcia, 1997, Martin, 1997).

An optimisation method applied to ship design by Mandel and Leopold (1966). The optimisation procedure relies on relatively crude empirical expressions to make all of the necessary calculations in its various sub areas of the ship design process. The problem of preliminary ship design is of the general form of a non-linear optimisation problem with constraints, in which both the measure of merit, usually some economic criterion, and constraints are non-linear functions of the design variables. Then later, some computer programs, which had used non-linear optimisation techniques, were developed (Nowacki et al, 1970). Lyon and Mistree (1985) criticised the design spiral and they used a multiple-objective optimisation technique. Ray and Sha (1994) described a multicriteria optimisation approach to ship design. The method incorporates naval architecture estimation methods, a decision system handler and a non-linear optimisation tool. Lee and Lee (1996) used a multicriteria optimisation method to overcome the limitation of single objective optimisation.

Owing to the characteristics of ship design, the designers' experience, imagination and analytical skills are frequently needed to make key decisions in the ship design process. So it is very attractive to consider applying the Expert System methods to assist the naval architect with ship design (Ye, 1991). MacCallum (1982) presented an Expert System called DESIGNER, which allows the flexible specification and investigation of numerical relationships in design. Duffy (1986) extended the system to be able to handle uncertainties associated with the use of relationships or with the result generated from a particular model. Akagi and Fujita (1987) presented an ES using the object-oriented knowledge representation concept. The system has the control module for the design process, the processing module for the knowledge base and the user interface module. It contains a set of objects for specific design knowledge, their relationships and the rules for diagnostic evaluation of the results generated. Cort and Hills (1987) developed a space layout design using computer-assisted methods. They used the theory of fuzzy sets to achieve the general layout of space, which allows delineation of the main compartments of a vessel. Another design tool was developed and applied to the preliminary design of ships by

Sekimoto et al. (1988). The tool is a kind of ES building tool and is intended to be used by a designer from the beginning. Welsh (1989) developed an ES tool for the overall design system concerned with the generation of vessel dimensions, hull form and layout design. For this purpose, an ES shell named INCODES (Intelligent Concept Design System) was specified and developed. In 1987 MARIN (Maritime Research Institute Netherlands) started on the development of an environment for parametric knowledge base systems QUAESTOR (van Hees, 1998). van Hees (1995) introduced a generic approach towards design modelling as implemented in the QUAESTOR shell. The design problem is described as a reasoning process. Akıntürk et al. (1995) developed a design tool, which is embedded in an ES environment, to aid naval architects in the design of multihull vessels. The ES environment called ECHIDNA, incorporates a new type of constraint logic programming. Their study shows that the ES environment can be a useful tool for designers in the design of more conformable and safer SWATH vessels. Lee and Lee (1996) investigated the feasibility of applying knowledge based systems, a genetic algorithm, and a numerical optimisation to conceptual ship design. Aguilar et al. (1996) developed a parametric hull form design system utilising combined knowledge based and object oriented methodologies, approaches and techniques.

Andrews investigated ship synthesis (Andrews, 1981) and described the CAD system produced to aid this investigation (Andrews, 1985). The system is applied to service vessels other than simple transportation.

Mistree et al. (1990) reviewed recent development in the field of design and offered a contemporary paradigm, Decision-Based Design, for the design of ships; one that encompasses systems thinking and embodies the concept of concurrent engineering design for the life cycle. Smith and Mistree (1994) demonstrated an integrated total approach to modelling and exploring a ship system in the early stage of decision-based design. They represented the development of top-level specifications and ship characteristics in a ship acquisition process. Erikstad (1994) explained how to improve concept exploration in the early stages of the ship design process. A prototype of the case-based conceptual design system of a ship is developed to support systematically during the design process. The system determines the priorities of the candidates by using the nearest neighbour matching algorithm (Lee and Lee, 1997).

---

## 2.8 Conclusion

In this chapter, general information about design has been given. On the other hand the knowledge based design method has been explained and in some sense detailed information has been reviewed, because this design method is not commonly used or known in the domain of ship design.

It can be seen from Table 2.2 that Expert System applications were introduced into the ship design domain in the early 1980's. This thesis will be a part of study of the application of ES into ship accommodation design.

The knowledge based design method has some shortcomings. Despite this, method can be applied to ship design. Especially if there is a shortage of experts the method can be very helpful.



### **3. ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEMS**

#### **3.1 Introduction**

This chapter describes the area of computer science known as Artificial Intelligence (AI) and offers a detailed description for one of its major components, Expert Systems (ES) or Knowledge Based Systems (KBS). A selection of the literature is given by presenting a general historical introduction to AI.

ES is perhaps the most fertile area for academic research in the field of AI, and this technology has been used on engineering or commercial applications. Expert System solutions are based on reasoning by using problem domain knowledge and heuristics. Hence Expert Systems are the most suitable for simulating human experts.

This thesis aims to investigate ES approach in the ship design more specifically container ship design as a first step of such approach. Therefore the structure and the components of ES i.e. knowledge representation, inference, explanation facility, knowledge elicitation are described and criticised by the ship design points of view. And finally selections of methods have been performed for container ship design.

#### **3.2 Definition and History of Artificial Intelligence**

The antecedents of AI include an intellectual tradition dating back to the Ancient Times. However, the ideas that sparked the formation of the discipline stem from research into human intelligence and formal reasoning that was conducted in the years prior to and following World War II.

The term “Artificial Intelligence” was first adopted by John McCarthy during a Summer Research Project at Dartmouth College (Williamson, 1986). Table 3.1 represents a summary of the works which contributed to AI from 17<sup>th</sup> century (Williamson, 1986).

It has been suggested that there are as many definitions of AI as there are those working in the field, but one of the broadest definitions is that: “AI is a field of study that seeks to explain and emulate intelligent behaviour in terms of

computational process.” From an engineering viewpoint: “AI is about generating representation and procedures that automatically solve problems until now solved by humans” (Schalkoff, 1990). AI has a particular relevance for engineers in that it increases the usefulness of the computer as a tool. One of the clearest definitions was suggested by Marvin Minsky: “AI is the science of making machines do things that would require intelligence if done by men” (Firebaugh, 1989).

Table 3.1. History of Artificial Intelligence

| DATE | WHO                         | WHAT                                                |
|------|-----------------------------|-----------------------------------------------------|
| 1642 | Pascal                      | First calculator                                    |
| 1672 | Leibniz                     | Improved the calculator                             |
| 1832 | Babbage                     | Developed analytical engine                         |
| 1847 | Boole                       | Algebra of logic                                    |
| 1937 | Shannon                     | Described electronic circuit in binary terms        |
| 1944 | Neumann                     | ENIAC (the first computer)                          |
| 1948 | Wiener                      | Published Cybernetics                               |
| 1950 | Turing                      | Defined test for machine intelligence.              |
| 1956 | McCarthy                    | Dartmouth Conference                                |
| 1957 | McCarthy                    | LISP AI language                                    |
| 1958 | Newell and Simon            | Developed general problem Solver                    |
| 1965 | Feigenbaum                  | DENDRAL, one of the first EXPERT SYSTEM application |
| 1966 | Weizenbaum                  | ELIZA, Natural language understanding               |
| 1970 | Intel 4004                  | First microprocessor                                |
| 1972 | Colmerauer                  | PROLOG, programming in logic                        |
| 1975 | Minsky                      | Frames, knowledge representation                    |
| 1977 | Forgy                       | OPS expert system shell                             |
| 1982 | Stanford Research Institute | PROSPECTOR, an ES application                       |
| 1985 | Shortlife                   | MYCIN, an Expert System application                 |
| 1985 |                             | Large memory capability to desktop computers        |
| 1985 | NASA                        | CLIPS Expert System Tool                            |

AI promises that computers will be able to provide assistance to design process instead of being limited to clerical and numerical work. Researchers are beginning to utilise design rules into computers to help with the design process. Computers that understand design rules will serve as electronic apprentices by taking over the simpler design tasks. Once computers take over some of the clerical parts of engineering, humans will have more time for the creative work.

AI is a very broad field of study, which is demonstrable from a consideration of Table 3.1. This table lists some of the more important topics which have been the subject of recent research. It is postulated that it is both necessary and desirable that its study should embrace as many disciplines as diversified as those of engineering and philosophy:

- vision
  - manipulation
  - reasoning about space
  - reasoning about object interaction
- sensor interpretation, including
  - visual sensors
  - acoustic sensors
  - signal-to-symbol conversion
- expert systems, including
  - rule-based reasoning
  - constraint –based reasoning
  - analogy –based reasoning
- language understanding, including
  - speech
  - syntactic analysis
  - question understanding
- learning, including
  - learning by noting patterns
  - learning by exploiting precedents
- common-sense reasoning, including
  - variable, precision reasoning
  - qualitative reasoning about quantities.
  - supercomputing hardware and software
  - massively parallel architectures

### 3.3 Expert Systems

The term ES, knowledge based systems or knowledge based ESs are often used synonymously. Most people use Expert System simply because it is shorter, even though there may be no expertise in their ES, only general knowledge (a rule based system).

The concept of the Expert System arose in the 1970's when AI researchers abandoned the quest for generally intelligent machines and turned instead to the solution of narrowly focused real-world problems.

AI has many areas of interest as presented above. The area of Expert Systems is a very successful approximate solution to the classic AI problem of programming intelligence. Feigenbaum 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” (Giarratano and Riley, 1994). Expert Systems are collected rules of thumb from human experience in a computer. Once an expert's knowledge is captured in a computer program, many copies can be made as required.

Expert Systems are based on the idea that rules are an effective way to tell computers how to do certain kinds of things that people do (Taylor, 1989). An Expert System is based on an extensive body of knowledge about a specific problem area. In general, this knowledge will be organised as a collection of rules that allow the system to draw conclusions from the given data.

#### 3.3.1 The Structure

The basic structure of an Expert System can be identified by its three principal components as shown in Figure 3.1; facts, rules, and control (Duffy, 1993).

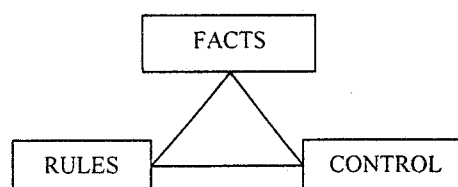


Figure 3.1. Principal components of an Expert System (Duffy, 1993)

**FACTS** refer to simple relations between two items and can be considered to represent knowledge about a specific situation, for example: Length of the ship is 150 m.

**RULES** are the basic operations which allow some kind of inference to take place, and can be considered to represent expertise about a situation. A common form of expression is as a “production rule”, for example.

IF gross tonnage is bigger than 5000

THEN there should be three bridge officers in the ship

**CONTROL** is the process by which facts and rules are managed and can be considered to contain the reasoning process. The control is able to match the symbolic patterns of a fact to a condition of a rule in order that the rule can “fire” and generate new facts. The control process decides when rules should fire, and indeed in which direction, since it is possible for it to determine the truth of condition statements given facts appropriate to actions. A convenient grouping of this structure is to call the facts and rules “the knowledge base”.

The approach of an ES replaces the tradition of: “data + algorithm = program” with a new structure centred around a “knowledge base” (facts and rules) and “inference engine”, so that: “knowledge + inference = expert system”, which is clearly similar, but different enough in approach to have certain consequences (Yazdani, 1986).

### 3.3.2 Elements of An Expert System

The elements of a typical expert system are shown in Figure 3.2.

- User Interface - the mechanism by which the user and the Expert System communicate.
- Explanation facility - explains the reasoning of the system to a user by the rules.
- Inference Engine - makes inferences by deciding which rules are satisfied by facts or objects and executes the rule with the highest priority.
- Agenda - a list of rules by the inference engine, whose patterns are satisfied by facts or objects in working memory.

- Automatic 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.
- Knowledge base: includes two parts, the domain knowledge, represents some rules about the problem domain, and working memory where facts and generated facts are hold.

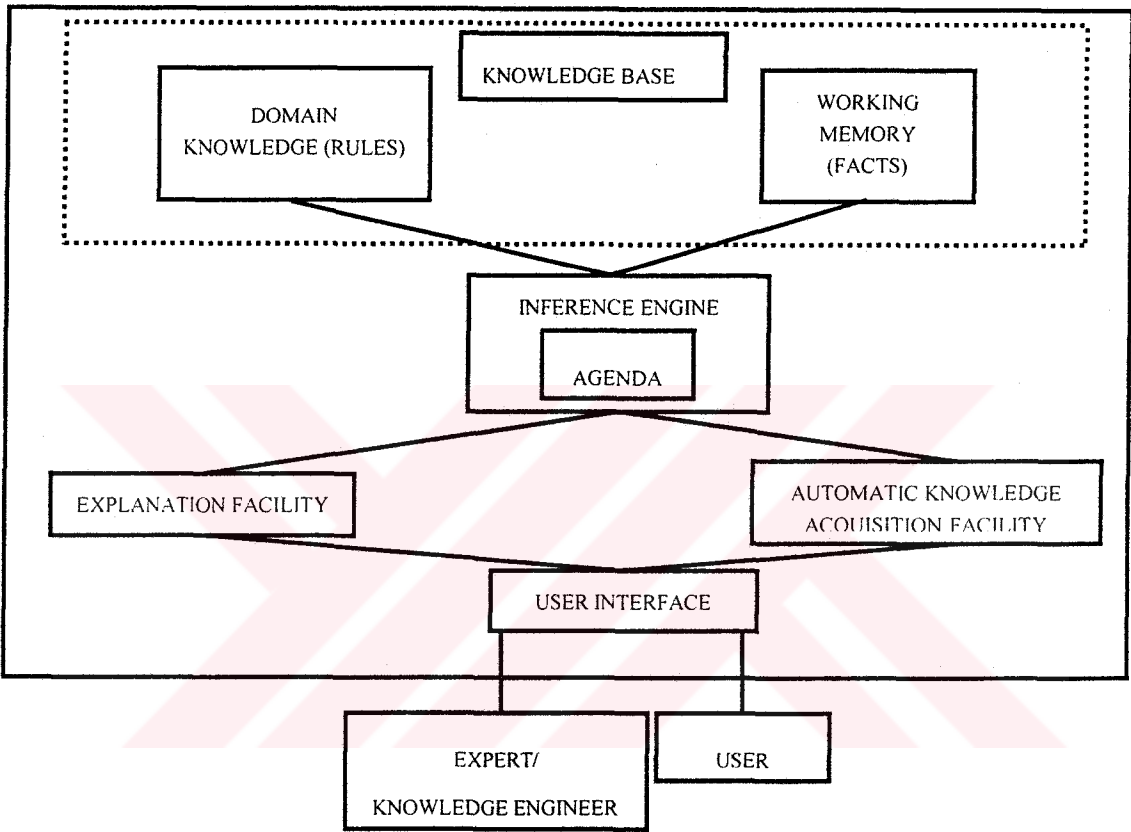


Figure 3.2. The elements of a typical Expert System

### 3.3.2.1 Knowledge Base

The knowledge base contains the knowledge which allows the system to solve the problems for which it was designed (Giarratano and Riley, 1994). This is quite a general definition and in reality can mean different things in different systems. It may include only the information. The information is always applicable to the type of problems in which the Expert System is trying to find the solution. In this situation the information contained is static and does not change during use of

the system. Alternatively, “knowledge base” may be used to refer to the static knowledge about the class of problems according to “working memory”.

In other words, the knowledge base is a part of the Expert System which contains the knowledge and expertise associated with a particular domain. This knowledge will usually be one of two types, factual or rule-based.

- factual knowledge is a type of knowledge that can be derived from text books about the problem area and in the case of marine design, from the requirements of classification societies, IMO regulations and other governing bodies.
- rule-based knowledge comprises the heuristics, good practice, accepted standards, and judgmental reasoning which the human expert applies to the solution of problems within the particular domain.

“Knowledge representation” is another term frequently used in connection with knowledge based systems. It means to organise and store the knowledge which the expert system uses to solve the problem, in the computer. The choice of technique used to represent the knowledge associated with a subject will largely depend on the nature of the problem under consideration and the format and structure of the available knowledge. It is very important to choose an appropriate knowledge representation technique otherwise it will seriously effect the viability of the proposed Expert System.

A number of knowledge representation techniques have been developed and utilised during the history of Expert Systems. These techniques include rules, semantic nets, frames, and logic.

Production rules: A production system has three parts; production rules, a global database which is “working memory” and a control system which corresponds to the inference engine. The production system has a form:

IF [condition] THEN [conclusion]

The condition part of the rule must be true, to use the conclusion part. Conclusion and condition relate to the state of the global database. If the condition is true according to the working memory, then the conclusion may be added to the working memory.

If gross tonnage  $\geq 25000$  and range is LONG-DISTANCE Then assert 3 bridge-officer.

The condition part of the rule is “Gross tonnage  $\geq$  25000 and Range is LONG-DISTANCE” and the conclusion part is “Assert 3 Bridge-officer”. The conclusion part may be added to the working memory. This way allows other rules to be used, adding knowledge to the working memory.

Production rules are of no use unless they can act in conjunction with the working memory, their use is directed by the control structure. Working memory holds all the facts pertinent to the problem currently being solved.

The control strategy of a production system decides which rules should be fired next in the procedure to solve the problem. A very simple control strategy is to check all the rules until a condition part matches the facts in working memory, then fire the rule and add the conclusion to working memory, by changing it. This process continues until the problem has been solved.

To capture this type of “judgmental reasoning”, some measure of the reliability of the conclusion may be stated in the rule. Classical logic is only capable of representing two states, true and false. There may be some degree of uncertainty in the relationship between the conditions and the conclusions. In many real situations, a number of techniques exist for the representation of uncertainty. The most widely used ones are: extended boolean logic, multi-valued logic, and fuzzy logic. Techniques such as those result in production rules being complicated by the addition of a parameter which indicates the level of truth associated with the statement Extended Boolean.

**Extended Boolean Logic:** With this logic representation technique, the two states of classical logic are extended to include a third, unknown. This increases the complexity of the knowledge base, but it has a considerable and beneficial effect on the flexibility of any Expert System.

**Multi Valued Logic:** This time classical logic approach is extended from definitely true through to definitely false with intermediate state such as possible true, probably false and so on.

**Fuzzy logic:** This technique is based on the fuzzy sets approach to incorporating uncertainty in reasoning logic. In this logic, the truth can be measured on a scale from 0 to 1.

Production rules are a natural way to express expertise. Many people can state rules and facts about their particular domain in the form of “If X then Y” statements. This makes the knowledge relatively straight forward. On the other hand the ability to use

uncertainty increases the expressiveness of this technique. Many ES shells use this formalism and production rules are good for diagnostic types of problems, where expert reasoning using the facts and rules of expertise takes place. The rule form is sometimes too rigid to express the knowledge and unless used carefully, uncertainty information is not meaningful and very difficult to set up initially.

Semantic Nets: Semantic nets are well-suited for representing knowledge of a hierarchical nature. In this representation, the knowledge is expressed in terms of a network of nodes and links (arcs) where the nodes represent the objects and the links are the relationships between them. Figure 3.3 illustrates a representation of semantic net kind for ship knowledge. The IS-AN label on an arc indicates that the object from which the arc originates is a member of the set to which the arc leads. An AKO (a-kind-of) link indicates that the set from which the arc originates is a sub-set of the set to which the arc leads.

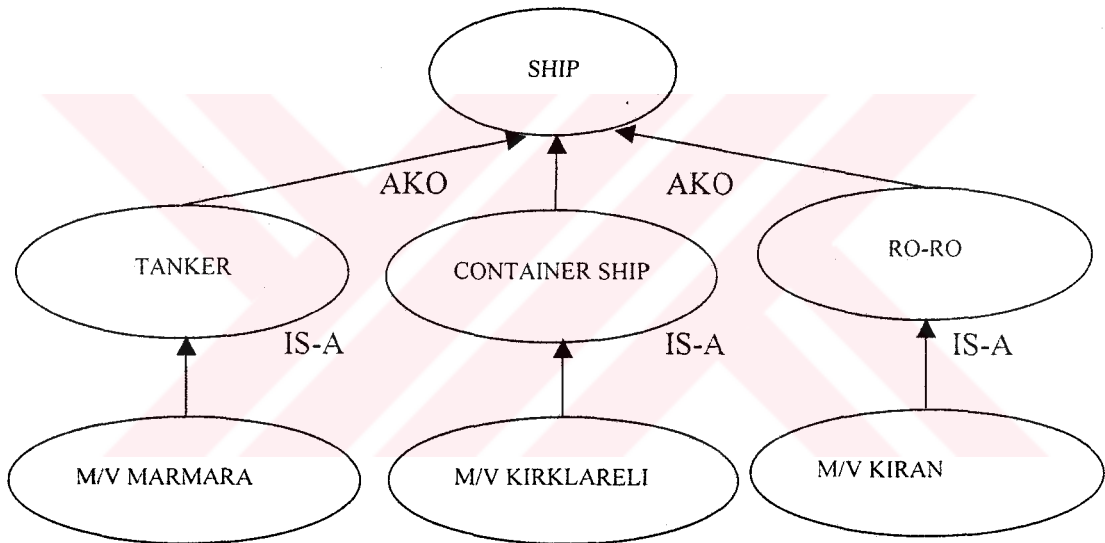


Figure 3.3. A simple semantic net

Figure 3.3 serves to introduce the important idea of inheritance. A particular vessel such as “M/V Kirklareli “ is an instance of the class of all container ships, and naturally has all the attributes of that class. The general attributes of a container ship do not have to be attached to every instance of a container ship that can be inherited from class. Another way of expressing this is that an AKO points from a subclass to a class. Inheritance is a very useful tool in knowledge representation because it eliminates the necessity of repeating common characteristics. Links and inheritance provide a very efficient means of knowledge representation since many complex relationships can be shown with a few nodes and links.

Although semantic nets can be very useful in representing knowledge, they have limitations such as the lack of link name standards. This makes it difficult to understand what the net is really designed for. Another problem is the combinatorial explosion of searching nodes, especially if the response to a query is negative. For a query to produce a negative result, many or all the links in a net may have to be searched. Semantic nets are logically inadequate because they cannot define knowledge in the way that logic can. Another problem is that semantic nets are heuristically inadequate because there is no way to embed heuristic information in the net on how to efficiently search the net.

Frames: Frames provide a method of combining declarations procedures within a single knowledge representation environment. The fundamental organising principle underlying frame systems is the packing of both data and procedures into knowledge structures.

The frame representation allows hierarchical information about object relationships to be stored in a knowledge base. A frame is a knowledge structure which is used to hold information concerning the attributes of an object. An example of a frame representation is given in Figure 3.4 (Welsh, 1989).

|           |                   |
|-----------|-------------------|
| Ship Name | "M/V Kirklareli " |
| CAPACITY  | 328 TEU           |
| SPEED     | 16.5 knot         |
| LENGTH    | 124.1             |
| BREADTH.  | 17                |
| DEPTH     | 9.4               |
| DRAFT     | 7.22              |
| CB        | 0.717             |
| POWER     | 4890 BHP          |
| ENGINE    | 8 ZAL 40 S        |

Figure 3.4 Example of a frame

A frame contains slots that can be filled with values. Frames are essentially composed of their slots. The basic components of knowledge are represented in frames as follows:

---

**Naming** - a unique name is assigned to each frame

**Describing** - the body of a frame is composed of a number of slots or attributes that have values. These slots or attributes can be used to describe the properties of the frame or to link different frames together.

**Organising** – Each frame, except the top level frame in a hierarchy, has one or more parents, providing an inheritance mechanism.

**Relating** - the values of frame slots may be other frames. Frames may thus be related by having one frame as the value of a slot in another frame. Frames may also be related by rules.

**Constraining** - each slot in a frame may have attached predicates, which are invoked whenever the slot is read or modified.

Frames have several inherent advantages over the normal production system approach. For example, in the frame system, knowledge relationships are stored in addition to the knowledge itself, and storage space requirement is accordingly considerably reduced.

Attributes are stored only once at the highest level at which they apply. Another advantage at the frames, is that attributes can be evaluated using procedures. The only disadvantage of the frames is the increased complexity of the inference mechanism required to control the logic.

Logic: In addition to rules, frames, and semantic nets, knowledge can also be represented by the symbols of logic, which is the study of the rules of exact reasoning. An important part of reasoning is inferring conclusions from premises. Logic is also of primary importance in ESs in which the inference engine reasons from facts to conclusions.

There are many systems of logic, including propositional and predicate logic. Propositional logic is a simple and well known system that includes basic concepts such as truth values and logical connectives. Predicate logic is a more powerful form of logic which includes the notations of variables, quantifiers and predicates.

The working memory of the Expert System contains a set of parameters which represent the current situation and all of the attributes which have been established since the start of the consultation session. Obviously the values of the parameters and attributes will change as the session progresses and more rules are

fired and more facts known. Working memory is the dynamic part of the knowledge base.

### **3.3.2.2 Inference Engine**

Inference engine accesses the knowledge base and contains general problem-solving knowledge. It applies to the knowledge base to infer new information. The inference engine is a mechanism in the Expert System which uses the knowledge stored in the knowledge base in order to solve a specific problem.

The inference engine contains two components - the inference and the control. The inference component works on the simple assumption that if the condition of a rule is true then the associated conclusion must also be true. By the constant application of this assumption to the rules in the rule base more and more facts become known and so the working memory is continually updated.

The term 'control' is used collectively to describe the procedure which an Expert System uses to find its next piece of information or to decide to execute the next step. In a normal computer system the GOTO statement is used to arrange the control. The GOTO statement or calling a subroutine is reached, the control mechanism makes the program pass that step (Duffy, 1993).

The inference engine uses one of two methods of reasoning, backward chaining or forward chaining. The major aspect of control was to determine how to choose the next rule to try.

Forward chaining is an inference method where the IF portion of the rules are matched against facts to establish new facts. Another way of describing forward chaining is reasoning from facts to the conclusion which follow from the facts. The way forward chaining works is illustrated in Figure 3.5.

Forward chaining is useful when many facts are available and it is useful to establish what else is true as a result of them. It is not a good way to find out if a particular situation is true - an Expert System with many rules will take a long time to return a satisfactory result in this situation (Duffy, 1993).

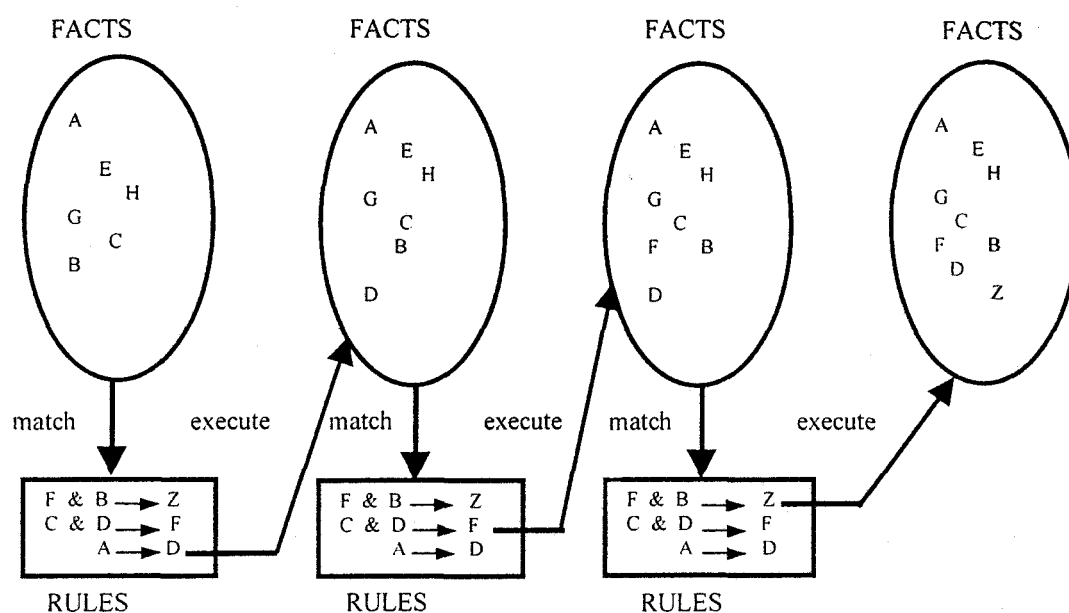


Figure 3.5 Example of forward chaining (Duffy, 1993)

Backward chaining is the reverse process of forward chaining. Backward chaining traverses from a hypothesis back to the facts which support the hypothesis. Another way of describing backward chaining is in terms of a goal which can be accomplished by satisfying subgoals (Giarratano and Riley, 1994).

The best way to understand backward chaining is to look at an example presented in Figure 3.6. This method is useful when the system is being used to try to establish whether particular situations are true or not.

Forward chaining is typically associated with “bottom-up” reasoning, that is, reasoning from facts to goals, while backward chaining is associated with “top-down” reasoning from goals to facts (Jackson, 1999). On the other hand it is possible for a forward chaining production system to perform top-down reasoning, if it is informed by a local control regime, such as an explicit ordering on goals.

From the point view of efficiency, forward chaining is expected more favourable for a fact base and a knowledge base for which there are many facts and few possible goals. A system for which there are few facts any many possible goals would favour the backward chaining approach.

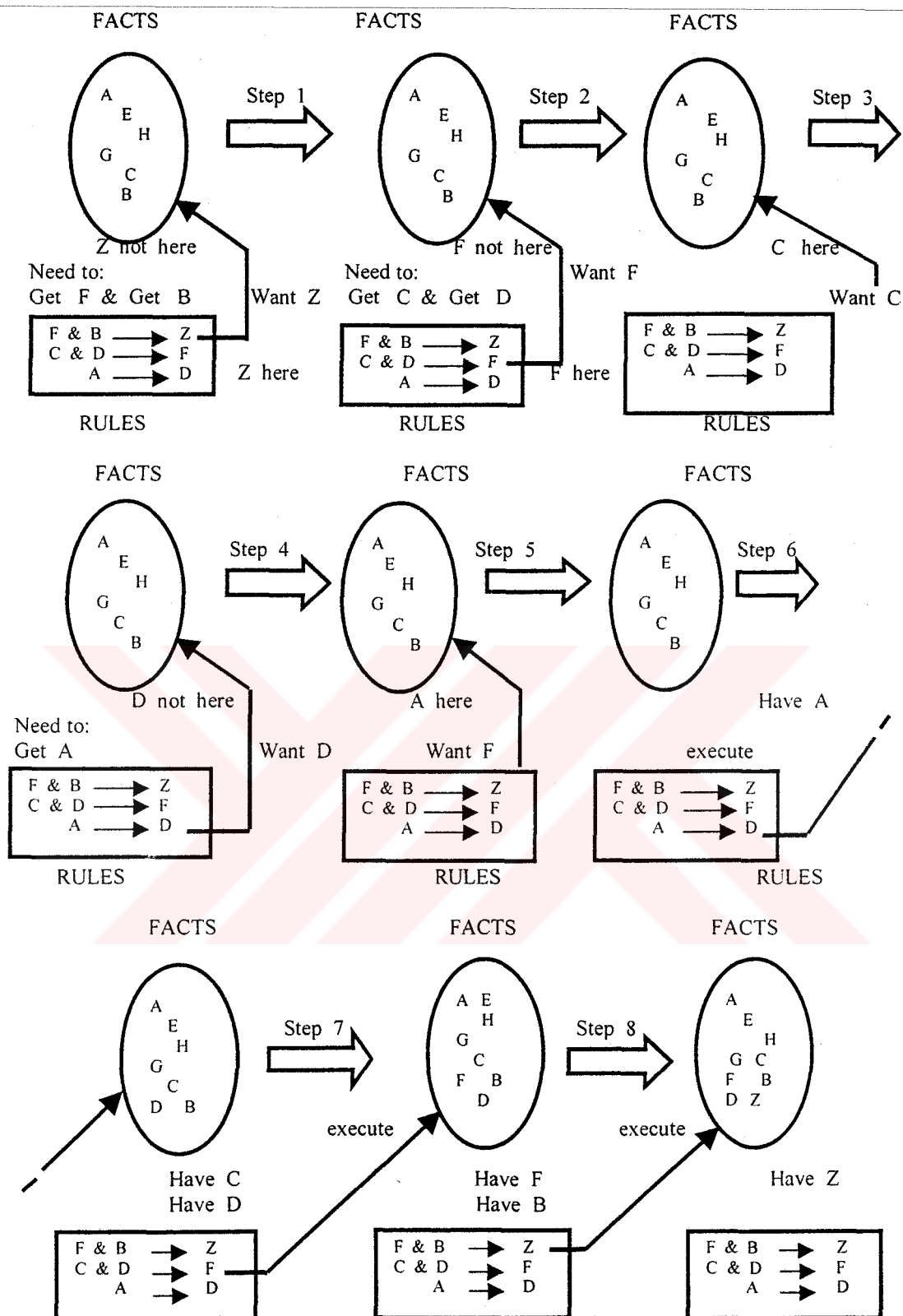


Figure 3.6. Example of backward chaining (Duffy, 1993)

The another facility of inference engine to achieve a goal is the search strategies. There are several common strategies used in Expert Systems. The two main search strategies which will be briefly mentioned are depth-first-search and breadth-first search.

In depth-first-search, the search is conducted from the starting node as far as it can be carried until either a “dead-end” or the goal is encountered. At each node some arbitrary rule is used to direct the search. If a dead-end is encountered, the search is backed up one level, and the next left-most branch is taken.

The logic can be understood by explanation of an example. Figure 3.7 shows a map for travelsal problem. The problem is to find a route from starting state S to goal state G which covers the minimum total distance.

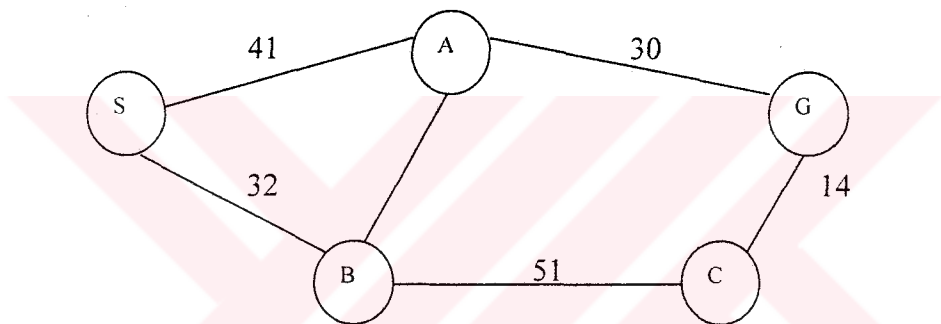


Figure 3.7. Map Travelsal problem (Firebaugh, 1989)

Figure 3.8 shows an example of depth-first-search according to map travelsal problem.

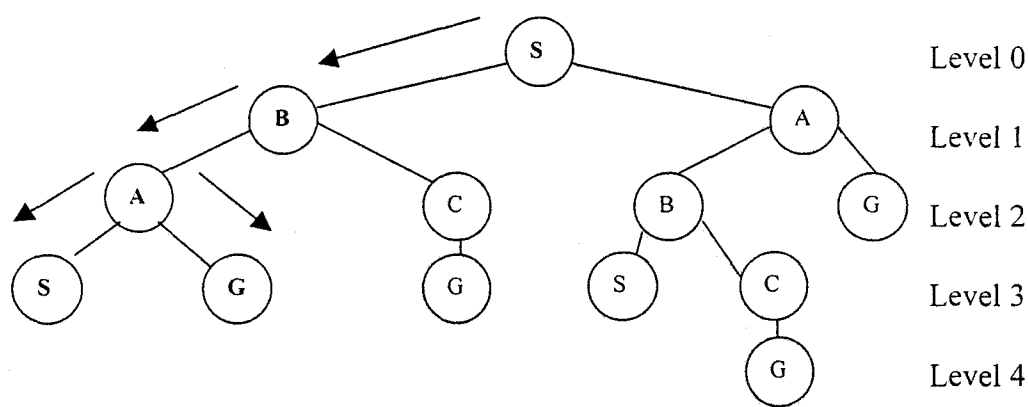


Figure 3.8 A depth-first search (Firebaugh, 1989)

An alternative method is called breadth-first search. This is a form of exhaustive search in which each of the nodes is systematically explored at each level of the search until the goal state is attained. Figure 3.9 gives an example of breadth first search.

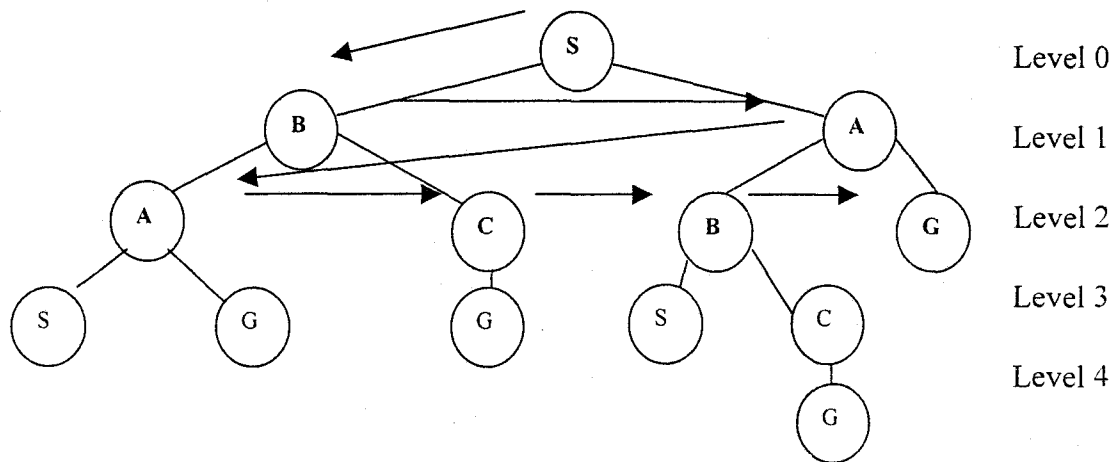


Figure 3.9 Breadth-first search (Firebaugh, 1989)

The breadth first search technique appears to be a more conservative approach which avoids the problem of racing by goal states.

### 3.3.2.3 User Interface

User interface is a mechanism by which the user and the ES communicate. The function of the interface in an ES is the same as in any other computer system. The importance of the interface is emphasised in ES design because of the large variety of information that has to be input into the system and out to the user. The facts about the problem must be described to the system, the reasoning which the system is making is to be explained to the user and the answer to the problem must be given.

### 3.3.2.4 Explanation Facility

An important aspect of ES is explanation. Explanation is generated usually by tracing what the inference engine has been doing, but must be displayed by the interface. In Expert Systems, most explanation facilities are provided by identifying which rules have been used to establish an answer.

The explanation facility also has an important function in that by explaining the current line of reasoning and justifying the assumptions made, it provides the user with the opportunity to either agree with the conclusions or recommendations

made by the system. This ensures that the user remains in overall control of the session and has the power to drive the session in the direction required.

### **3.4 Knowledge Acquisition and Elicitation**

The development of an Expert System involves the mapping of the expertise or knowledge into a computer system. The source of the knowledge may be manuals, published papers, an expert or a group of experts.

The knowledge elicitation is the process of extracting knowledge or expertise from some sources (Jackson, 1999). Knowledge acquisition is the process of eliciting and transferring knowledge or expertise from some source to some sort of computer-based representation. Knowledge elicitation is a stage of knowledge acquisition process. Knowledge acquisition is quite a difficult problem. Jackson (1999) explains some of the difficulties with it:

- it is difficult for experts to communicate their knowledge in everyday language.
- the facts and principles underlying many domains of interest cannot be characterised precisely in terms of a mathematical theory or deterministic model whose properties are well understood.
- eliciting knowledge which is based on personal experience rather than formal training, is much more difficult than eliciting either particular facts or general principles.
- human expertise is set in a broad context that involves a good deal of common sense knowledge about the everyday world.

The acquisition process begins with a basic introduction to the domain. The introduction makes the system developer find out about some basic problems encountered, some basic terminology, about the domain, and a general understanding of the problem domain.

The next stage is to study the background of the domain. The system developer becomes more familiar with the problem. If the knowledge source is an expert or a group of experts, it is necessary for the developer to contact the experts privately. Otherwise the developer must carry out a further study into the problem. Once an adequate understanding of the problem domain has been established then the knowledge elicitation phase can begin.

### 3.4.1 Knowledge Elicitation Techniques

A number of techniques have been developed to assist the system developer in acquiring the necessary knowledge to build the required Expert System. In most cases more than one technique is used. Some of the techniques are:

- historical knowledge: historical knowledge is in forms such as published material, case histories and statistical analysis. This knowledge can be updated and used.
- an expert acting as a system developer: if the expert is familiar to the Expert Systems and computer then he/she can develop a system which contains her/his knowledge. The expert can examine her/his own knowledge.
- interviews: it is one of the most common knowledge elicitation techniques. Interviews can be conducted as compilation of a number of forms. There are a number of questioning strategies available to the system developer. Typical of these are:
  - past cases: the expert can explain interesting past cases. The system developer decides that the knowledge is relevant to the project in hand.
  - distinguishing goals and reclassification: the evidence specified for a particular goal can then be reclassified as sub-goals until the evidence could be directly observable by the expert.
  - symptoms and causes: if the expert can not define his/her knowledge in the form of rules or heuristics, the expert may be able to give a list of symptoms, or characteristics, and possible causes.
  - observational and protocol analysis: the system developer observes the manner in which an expert goes about solving a particular task in the domain of interest.
  - machine induction: This approach removes the system developer and the rules are extracted automatically by the computer from defined examples.
  - prototypes: A prototype system, which is only a rough starting point and will not be comprehensive, can be generated which can then be criticized by the expert who can use it as a basis for modifying or providing additional knowledge.

All these techniques have their merits and drawbacks. The system developer must choose the most appropriate technique according to his/her domain.

### 3.5 Expert Systems Versus Conventional Programming

Table 3.2 represents the differences between Expert Systems and conventional programs.

Table 3.2 Typical differences between conventional programs versus Expert Systems

| Expert Systems                                                                                                                                                                                                                                                                         | Conventional Programs                                                                                   |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>• An Expert System simulates human reasoning about a problem domain</li> </ul>                                                                                                                                                                  | <ul style="list-style-type: none"> <li>• A conventional program simulates the domain itself.</li> </ul> |
| <ul style="list-style-type: none"> <li>• It performs reasoning over representation of human knowledge in addition to doing numerical calculations or data retrieval. The distinct program modules are referred to as the knowledge base and inference engine, respectively.</li> </ul> | <ul style="list-style-type: none"> <li>• It is powerful for numerical calculations.</li> </ul>          |
| <ul style="list-style-type: none"> <li>• It solves problems by using heuristic or approximate methods which are not guaranteed to succeed.</li> </ul>                                                                                                                                  | <ul style="list-style-type: none"> <li>• It works with proper data about the domain.</li> </ul>         |

### 3.6 Advantages and Disadvantages

Expert Systems have a number of attractive features:

- the domain expertise can be available on any suitable computer hardware,
- the cost of providing expertise per user is greatly lowered,
- unlike human experts who may retire, quit or die, the Expert System's knowledge will last indefinitely,
- the knowledge of multiple experts can be made available to work simultaneously and continuously on a problem at any time,
- Expert Systems increase the confidence that the correct decision was made by providing a second opinion to a human expert or breaking a tie in case of disagreement by multiple human experts,

- Expert Systems can explain the reasoning that led to a conclusion in detail,
- fast or real time response may be necessary for some applications. An Expert System may respond faster and be more available than a human expert,
- an Expert System is steady, unemotional, and gives complete response at all times.

While Expert Systems have considerable advantages to offer, they do have their disadvantages or limitations.

- some applications are difficult to use by another person who has not created the knowledge base,
- some systems are very slow when compared to the human expert,
- the knowledge systems' ability tends to end abruptly,
- sometimes it is difficult to extract knowledge from an expert and to put it into a format that the Expert System can deal with,
- the inability of Expert Systems to exhibit common sense limits the effectiveness of present Expert System applications,
- the size of the Expert System domain must be limited,
- Expert Systems can only be applied to those areas in which an expert exists.

### **3.7 Selecting the Appropriate Problem**

An important aspect of an Expert System development is to identify and evaluate an area or domain to which an Expert System can be applied. In this section some clues are given to identify and evaluate suitable application areas (Duffy, 1993):

- the expert knowledge should be applicable and the problem solving process should rely on expert knowledge, experience and judgement,
- if the problem does not primarily involve symbolic reasoning, as for example mathematical analysis, then a conventional approach could be more appropriate,

- Expert Systems represent heuristic knowledge and can reason with uncertain and fuzzy information,.
- the problem area must not be too easy or too difficult,
- the problem should not be too broad trying to solve the problem of an entire domain, and should be self contained. It means the problem domain must not require knowledge from other areas,
- the task should be of a manageable size and one that could be achieved given the available resources,
- if the area is changing too rapidly then the development of the Expert System would be extremely hampered and would not justify the required investment,
- the expertise must be accessible,
- the expertise must be of high enough standard and quality as to justify development work,
- the system can take considerable time, resources and capital to be developed,
- there should not be any alternative solution,
- the system must have proper user interface and should be able to represent any relevant real world expertise,
- the expert must be ready to co-operate, must describe his/her reasoning and explain his/her knowledge. The expert must be easy to get on with and be easy going.

### **3.8 Expert Systems Applications and Domains**

Expert Systems have been applied to virtually every field of knowledge. Some have been designed as research tools, some as important business and industrial functions. Some examples of ES are given in Table 3.3.

There are a lot of examples of applications of Expert Systems. Bertram (1998) gives a variety of examples for maritime applications. Dym and Levitt (1991)

explain a lot of Expert Systems' applications in the field of engineering, design, and planning. A review paper has been presented by Helvacioğlu (1995).

Table 3.3. Broad classes of Expert Systems (Giarratano and Riley, 1994)

| Class          | General Area                                                                                                              |
|----------------|---------------------------------------------------------------------------------------------------------------------------|
| Configuration  | Assemble proper components of a system in a proper way.                                                                   |
| Diagnosis      | Infer underlying problems based on observed evidence.                                                                     |
| Instruction    | Intelligent teaching so that a student can ask as Why, How and What if type of questions just as if a human was teaching. |
| Interpretation | Explain observed data.                                                                                                    |
| Monitoring     | Compares observed data to expected data to judge performance                                                              |
| Prognosis      | Predict the outcome of a given situation.                                                                                 |
| Remedy         | Prescribe treatment for a problem.                                                                                        |
| Control        | Regulate a process                                                                                                        |

### 3.9 Expert Systems Tools

Although Expert systems are a branch of AI, there are specialised languages for Expert Systems that are quite different from the commonly used AI languages such as LISP and PROLOG (Giarratano and Riley, 1994).

The primarily functional difference between Expert Systems languages and procedural languages are the focus of representation. Procedural languages focus on providing flexible and robust techniques to represent data. In contrast, Expert System languages focus on providing a flexible and robust way to represent knowledge. Expert System languages specifically separate the data from the methods of manipulating the data like the separation of facts and rules.

In choosing a language, a basic question should be whether the problem is knowledge or intelligence intensive. Expert Systems rely on specialised knowledge or expertise to solve a problem, while AI emphasises a problem solving approach. Expert systems often rely on pattern matching while AI generally concentrates on searching for paradigms in less restricted domains.

The road selecting an Expert System language is paved with confusion. With the explosive commercial growth in the Expert Systems field since the 1970's the selection of languages is no longer so simple (Giarratano and Riley, 1994). The Expert Systems' tools can briefly be explained:

- language: an Expert System language provides an inference engine to execute the statements of the language. The inference engine may provide backward chaining or forward chaining or both. Programming language written for ESs, such as PROLOG, any other programming language such as LISP, FORTRAN etc. may be used for ESs.
- shell: an Expert System Shell has three components, a formalism for representing a knowledge base, an inference engine and an interface. The user will be able to change the structure or basic components of the system. A shell is an empty program in which domain knowledge may be added. The shell dictates the form in which the knowledge must be structured in the system, and it will dictate the inference method which will be used on it and the way in which the user can interact with the system.
- high level programming languages: languages like OPS5 can be used to develop ESs. The advantage of these languages, it is easy to learn and yet less constraining than a typical shell, free the programmer from low-level considerations of efficiency in the storage, access and manipulation of data (Jackson, 1999).
- multi-paradigm programming environment: which provide a set of software modules that allow the user mix a number of different styles of AI programming. For example LOOPS combine object and rule-based representation (Jackson, 1999).

The disadvantages of a shell are the lack of choice. The system developer cannot introduce new forms for the knowledge or change the interface process.

### **3.10 Conclusion**

Expert System structure has been developed with very powerful knowledge representation techniques and reasoning methods. Such methods have been presented briefly in this section.

As it was stated in the first section, this work aims to develop a container ship design Expert System. Main components of the current system are established from this review to achieve this purpose.

Production rules are the most utilised and the most powerful knowledge representation technique for incomplete heuristic knowledge which is the principle type of knowledge in the ship design. They are also easy to adopt for regulations

governing ship design. Hence production rules have been chosen as knowledge representation technique in the current work.

Design problems are suitable for forward chaining reasoning meanwhile diagnostics, i.e. design control is more suitable for backward chaining. As the current work aims to design or aid to design, forward chaining was chosen as inference method.

A search through available research Expert Systems languages and shells have indicated that CLIPS (C Language Integrated Production System) Expert System shell, by NASA Cosmic, would be suitable for the purpose. CLIPS is available in the public domain, very well documented and is supported on a variety of operating systems including MSWindows. Additionally CLIPS is utilised frequently on research purposes world wide, and CLIPS user group is one of the most active discussion environment in the ES shells.

User interface of CLIPS, as given in Appendix A, allows input of facts, rules or loading facts and rules from disc files, and messages are outputs during runs. However visual capabilities of CLIPS such as tabulated input graphics are very limited. Therefore a pascal language driven visual programming interface was developed in DELPHI as explained in Appendix D. This user interface allows users to input all necessary data, and observe outputs in a visual environment as well as visualisation of graphical outputs. Additionally some of numerically intensive tasks can be performed by DELPHI which cannot be done in CLIPS.

CLIPS provides an automatic basic explanation method by listing fired rules and facts being asserted or retracted. This explanation facility have been utilised and no more explanation facility was foreseen.

Search strategies have also been reviewed in this chapter. CLIPS provides six search strategies including breath first and depth first search techniques. The depth first search which have been utilised is default in CLIPS.

Knowledge elicitation techniques utilised during the work is given in section 5. In summary:

- Historical data from existing ships have been used in statistical analysis form.
- Interviews were utilised to derive heuristics of the experts.

- System developers knowledge have been utilised where no data could be obtained.

By considering all the shortcomings, it has been decided to use CLIPS shell for container ship design.



## **4. LAYOUT DESIGN**

### **4.1. Introduction**

The layout problem is discussed to review the available methods in which the knowledge is mainly about facility layout and location problems, in this chapter. The term “Facility Layout” is used in a very literal sense. In particular it refers to the design of layout for production facilities or for service facilities.

The facility layout problem has geometric and combinatorial aspects, which makes the problem difficult when combined. The facility layout problem can be assumed to be like a jigsaw puzzle problem (Palekar, 1999). A jigsaw puzzle takes a large amount of time when there are a large number of pieces. The problem becomes harder when the picture on the jigsaw puzzle does not give many clues on which pieces should be adjacent. This jigsaw puzzle represents only one part of the facility layout problem. The pieces represent departments and the puzzle-frame represents the building, which houses these departments. There are many ways to solve the problem. If each configuration solution has a score associated with it, and the best score is sought. Then each feasible configuration has to be considered. The amount of time required to solve the puzzle would be large. The layout problem usually involves minimisation of material handling costs, which results in a cost associated with each feasible layout.

Since the problem is so difficult, heuristics are often used to solve the problem. Typically only some aspects of the problem is considered and one or more approximate solutions to the problem are created using this restricted version. The resulting solutions are adjusted to create an acceptable layout and the best resulting solution is implemented.

The interest in facility layout and location problems in this thesis is not oriented toward the layout of production plants; instead, the techniques and methods are explained and adopted for the ship accommodation layout design problem. It is expected that subsequent discussion will be beneficial.

## 4.2. Systematic Layout Planning (Muther)

A high-level approach to the entire process of facility layout was first formulated by Muther (1961). The method is called Systematic Layout Planning (SLP).

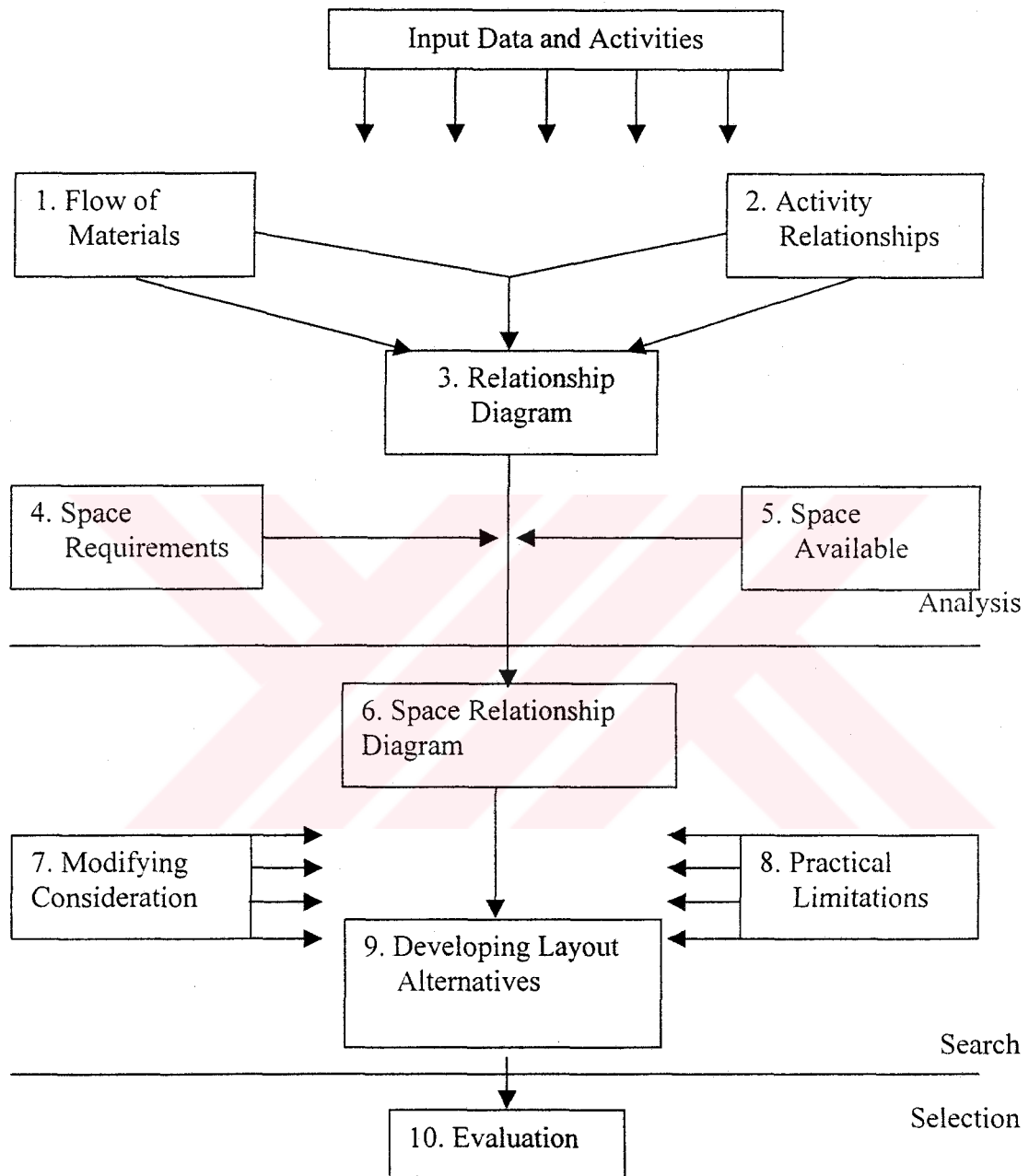


Figure 4.1 Systematic Layout Planning procedure (Mutter, 1961)

The SLP procedure is depicted graphically in Figure 4.1. Once the appropriate information is gathered, a flow analysis can be combined with an activity analysis to develop the relationship diagram. Space considerations, when combined with the

relationship diagram, lead to the construction of the space relationship diagram. A number of alternative layouts are designed and evaluated, based on the space relationship diagram, modifying considerations, and practical limitations.

The first five steps of SLP involve the analysis of the problem. Steps 6 through 9, including the generation of alternative layouts, constitute the search phase of the design process. The selection phase of the design process coincides with step 10 of SLP.

#### **4.2.1 Information Gathering**

Certain information pertaining to the product, process, and schedule must be obtained to perform the facility layout effectively. The data requirement may not coincide with data availability in some cases, but it is assumed that data exists.

Data regarding product design decisions can significantly affect the layout. Therefore it is important to have available data concerning the design of the product. Basic product design data can be obtained from production drawing, assembly charts, parts lists, bills of materials, and prototype of the product.

Process design decisions determine whether a part will be purchased or produced, how the production of a part will be achieved, what equipment will be used and how long it will take to perform each operation. This information is summarised on an operation sheet. A separate route sheet is required for each part. The operation process chart is an analogue model of the operations and inspections involved in producing a product. In some cases the operation process chart is the primary basis for the layout. Assembly charts and operation process charts are given by Francis and White (1974) in more detail. They are not the interest of this work hence they are not given.

#### **4.2.2 Flow Analysis and Activity Analysis**

Flow analysis concentrates on some quantitative measure of movement between department and activities, whereas activity analysis is primarily concerned with the flow of materials.

Once certain basic data have been obtained concerning the process, product and schedule, it is time to analyse the flow of materials, equipment and personnel. The factors that affect the flow pattern are given by (Appel, 1977):

#### 4.2.2.1 Type of flow Patterns

Type of flow patterns is one of the most important subjects in flow analysis. The flow patterns can be classified as vertical and horizontal. There are five basic types of horizontal flow pattern as shown in Figure 4.2. A number of other flow patterns can be developed by combining these basic flow patterns.

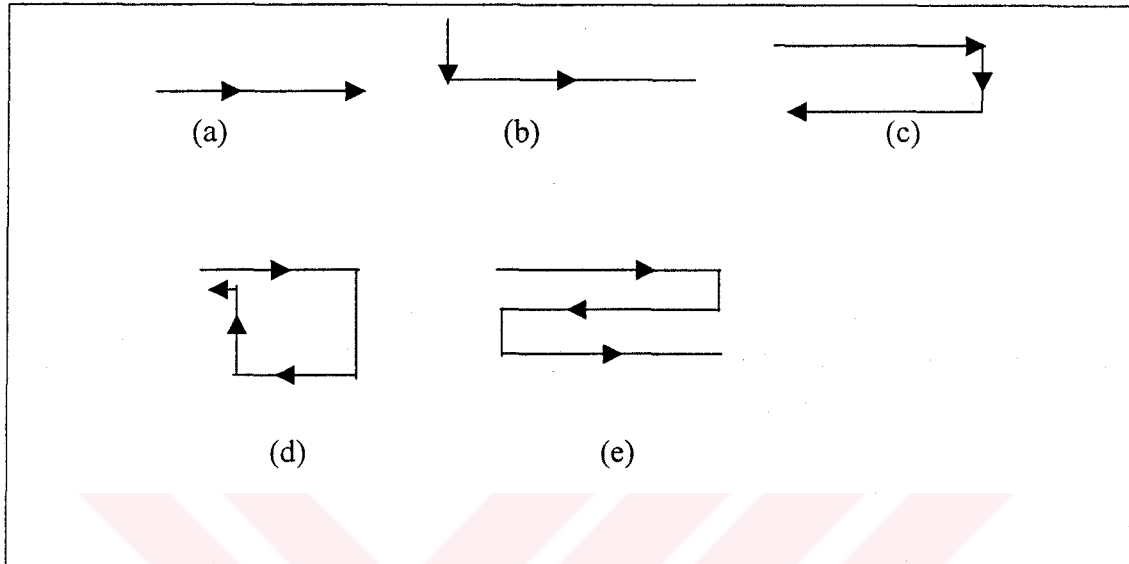


Figure 4.2 Basic horizontal flow pattern (a) Straight or I flow; (b) L flow; (c) U flow; (d) circular or O flow; (e) serpentine or S flow (Francis and White, 1974)

Straight-line flow is the simplest form of flow. However, when employed in a plant, separate receiving and shipping crews are normally required. L-shaped flow pattern is usually adopted when either straight-line flow can not be accommodated in an existing facility or construction cost does not permit straight-line flow. The U-shaped flow is very popular, since it is simple to administer and facilitates a combination of receiving and shipping activities. Circular flow is applicable when it is desired to terminate the flow where originated. The serpentine flow pattern is used when the production line is so long that zigzagging on the production floor is required.

Vertical pattern exists both in single-story and multi-storey buildings. Figure 4.3 shows four different vertical flow types. Vertical flow pattern (a) is often used when there is flow between buildings and there exists an elevated connection between the buildings. Pattern (b) is used when ground level entry and exit are required. In (c), ground level entry and exit occur on the same side of the building. Centralised elevation exists in (d), since travel between floors occurs on the same side of the building.

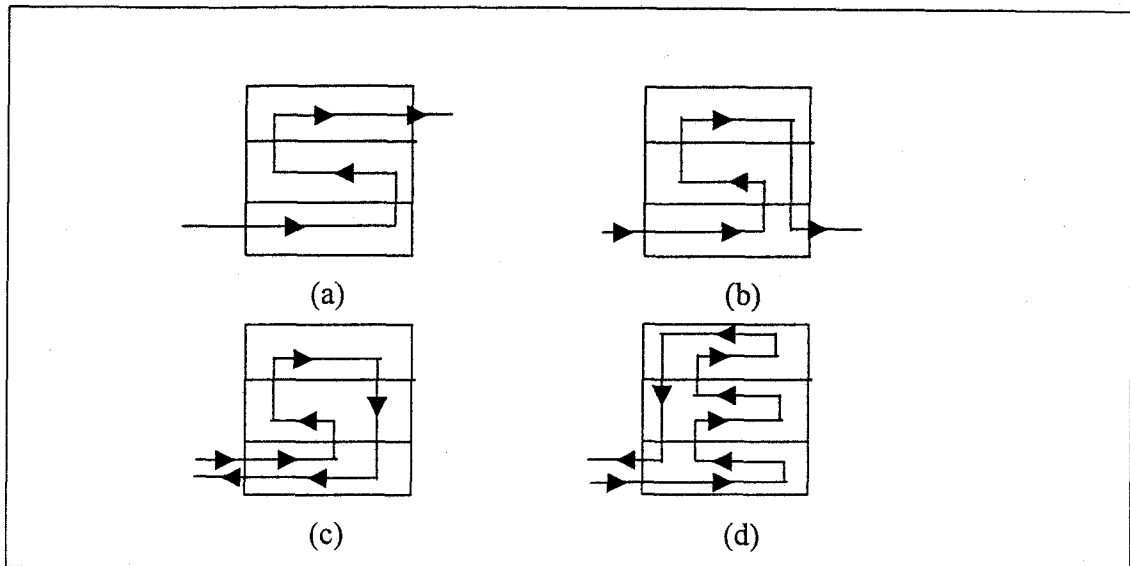


Figure 4.3 Vertical flow pattern (Francis and White, 1974)

#### 4.2.2.2 Flow-analysis Approaches

After discussing the importance of flow analysis in layout design, it is time to consider some techniques that are commonly used in analysing the flow of materials, men, and machines. The most popular method for analysing flow is to use charts and diagrams. Some additional diagrams to the assembly chart and the operation process chart are given as follows:

1. flow process chart,
2. multiproduct process chart,
3. flow diagram,
4. from-to chart.

In this work only from-to charts will be used. Therefore the from-to charts will be explained here. The others are discussed in Francis and White (1974). The from-to chart normally contains numbers representing some measure of the materials' flow between two buildings, departments, or machines. Typically, the from-to chart provides information concerning the number of material handling trips made between two centres of activity and the total material handling distance.

The construction of a from-to chart does not result directly in the solution of a layout problem. Rather, the from-to chart is a convenient mean of reducing a large volume of data into a workable form. It is possible to see from this chart that the

departments have large volumes of item movement and it is also possible to develop a layout design in which these departments are located near one another.

#### 4.2.2.3 Activity Relationship Analysis

The activity relationship chart or REL chart, was designed to facilitate a consideration of qualitative factors. The REL chart replaces the numbers in a from-to chart by a qualitative closeness rating. A typical REL chart is shown in Figure 4.4.

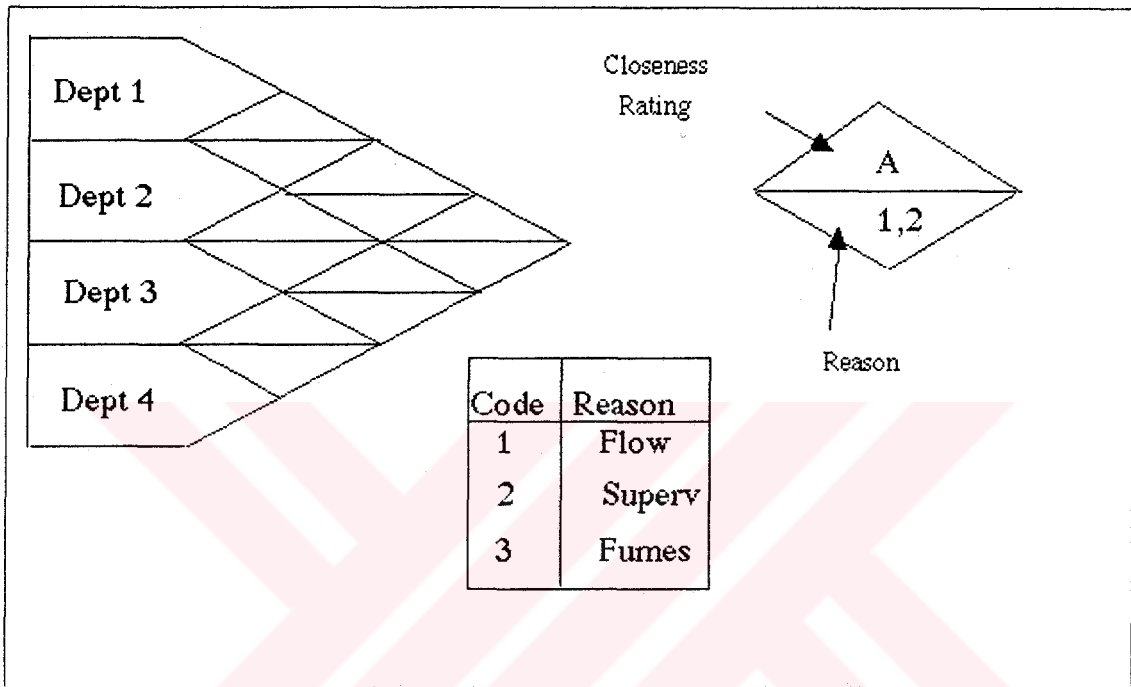


Figure 4.4 Activity relationship chart (Palekar, 1999)

The REL chart represents the importance of adjacency between departments using a six-level rating scheme as follows:

A - absolutely necessary

E - especially important

I - important

O - ordinary Importance

U - unimportant

X - undesirable

It is up to the analyst to determine the exact rating to be assigned to a pair of activities. The REL chart is traditionally represented using a table such as the one shown in Figure 4.4

Each diamond represents the relationship between a pair of departments. The top half of the diamond is used to indicate the rating while the bottom half is used to explain the reason for the rating. A table of the reason codes is also given with the REL Chart.

#### 4.2.2.4 The Relationship Diagram

The REL Chart can be used to determine the adjacency between departments. If material flow is an important consideration or if common supervisory control is important then a high rating between the two departments suggests that these departments should be geographically close to each other. The shape and size of the departments limits the number of departments which can be adjacent to one another. One of the first steps in developing a layout is to determine the adjacency between different departments. Figure 4.5 shows a typical activity relationship diagram.

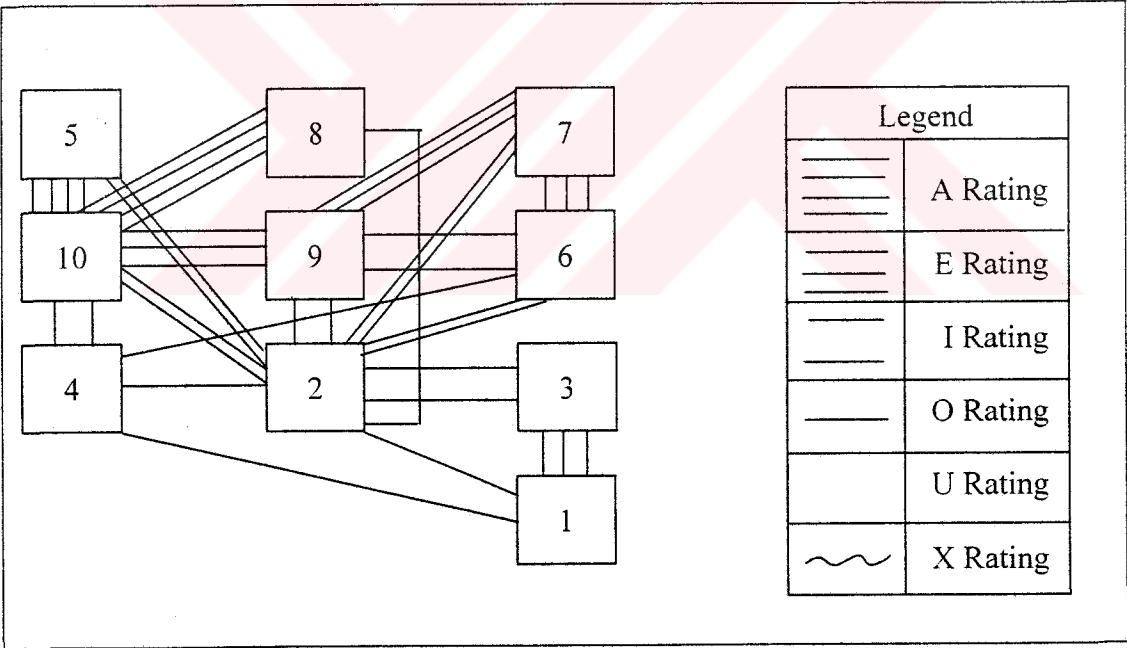


Figure 4.5 Activity relationship diagram (Francis and White, 1974)

#### 4.2.3 Space-relationship Establishment

At this stage, the requirements of space for each department is established and assigned according to space available. This stage can show variety of forms from heuristic space assignment to space assignment according to regulations.

When space assignment is complete, relationship diagram is transferred into space-relationship diagram simply by using correct size for all departments as shown in Figure 4.6. Additionally the blocks can be located according to relationship giving clues for the layout alternatives.

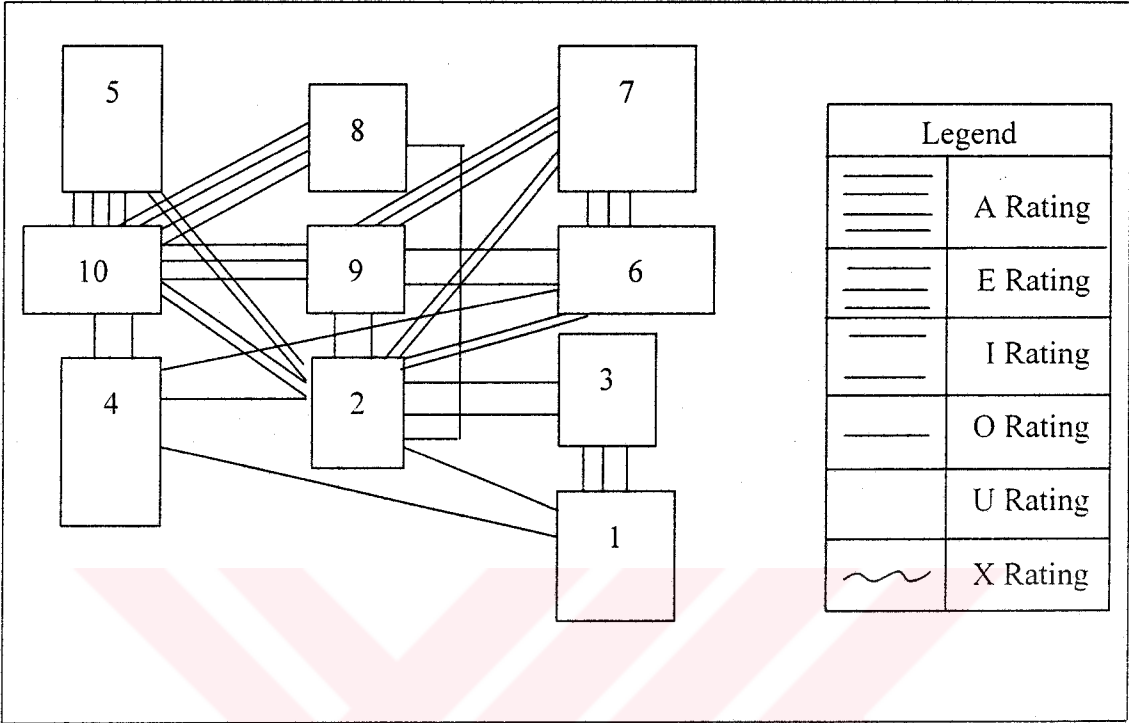


Figure 4.6 Space relationship diagram

#### 4.2.4 Developing Layout Alternatives

Developing alternative layouts consists of two steps. Firstly limitations on the layout and modification strategy is established. Then layouts are sketched with a generative algorithm. This step can either be performed by an expert or by an established computer algorithm. Some of the most common algorithm will be presented here to show computerised systems' logic in developing layout.

Most computerised layout planning construction heuristics can be divided into two phases (Palaker, 1999):

- selection phase - typically involves some greedy rule,
- placement rules - these are designed to be simple and yet ensure that the resulting departments have the following properties:
  - contiguity and connectedness - departments cannot be partitioned,

departments are laid out two columns of grid squares at a time. With a sweep width of 4 the departments are laid out 4 columns at a time. Figure 4.7 shows the zig-zag pattern.

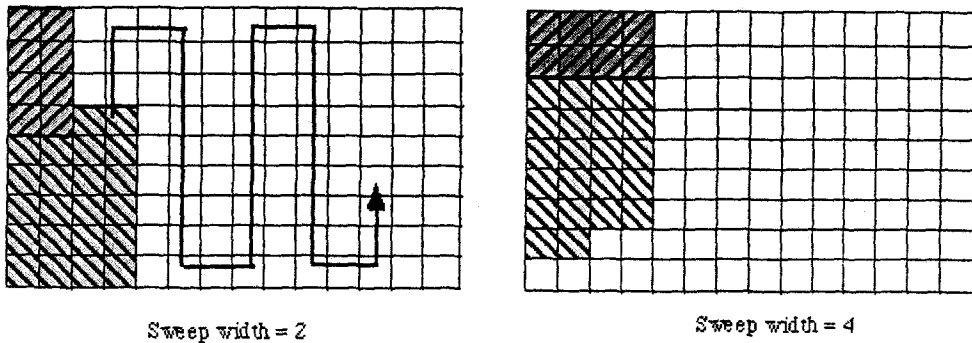


Figure 4.7 Zig-zag pattern (Palekar, 1999)

After a layout is completed, ALDEP determines how good the layout is by giving it a numerical score. To calculate this score ALDEP associates the following weights with the alphabetical ratings of the REL chart.

Weights : A=64; E=16; I=4; U=0; X=-1024.

If two departments are adjacent in the layout then the weight corresponding to the rating between the two departments is added to the score. Notice the large negative weight associated with X ratings. Clearly, if there are a number of departments which should not be next to each other in a problem there is a high chance that some of these departments are going to be adjacent in a layout. This will cause a significant reduction in the layout score. This is important because ALDEP also uses a cut-off score (if not specified by the user this cut-off is zero) to eliminate any layout which has a layout score less than the cut-off score.

#### 4.2.4.2 Computerized Relationship Layout Planning - CORELAP

Like ALDEP, CORELAP uses the REL chart as its input (Palekar, 1999). However, CORELAP asks the user to assign weights to the alphabetic ratings in the REL Chart. These numeric weights are called "closeness ratings" (CR). CORELAP computes a total CR (TCR) for each department by summing all the CRs associated with that department. Since the TCR plays a key role in the procedure, the user should carefully select these values to truly reflect the importance of the A,E,I,O,U,X ratings.

The selection procedure in CORELAP is deterministic and for given values of CR will always return the same department order. The procedure uses the highest TCR, the largest area and the lowest department number (random) rules.

CORELAP uses an elaborate placement and evaluation scheme to create a block layout. As with ALDEP, department areas are converted into grid squares. However, CORELAP does not take the building shape into account. The final shape of the facility created by CORELAP depends on the placement of departments it has selected.

The procedure begins by placing the first department in the centre of the layout. Each subsequent department is placed abutting already placed departments in the position that gives the best placement rating. This can be a cumbersome process as CORELAP will attempt to fit a department so that it is in contact with existing departments at each boundary grid square and for each possible aspect ratio. The arrangement with the best placement rating is selected. If two arrangements have the same placement rating then the one with the smallest boundary contact is selected. If a tie persists then it is broken arbitrarily.

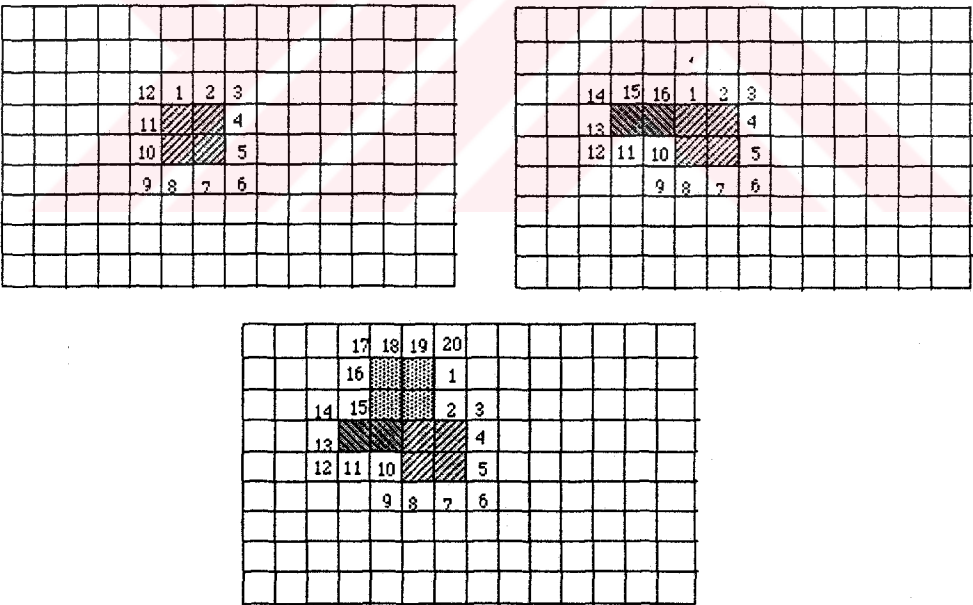


Figure 4.8 CORELAP evaluation (Palekar, 1999)

In the first figure of Figure 4.8 a single department is placed in the centre. The number of positions that CORELAP will evaluate is also shown. For each of these positions CORELAP will try to fit the second department using all possible aspect ratios. The second department in this example occupies 2 grid squares. These

two squares can be arranged either as a 2X1 or a 1X2 department and thus there are 24 possible placements for the second department. Since each position essentially is identical in that it places the two departments in contact with each other the placement rating for each of the 24 positions will be the same and equal to the CR between the two departments. In the second figure of Figure 4.8, the placement that was selected is represented. Notice that the boundary contact is minimised by allowing only one unit of adjacency. There are now 16 positions for the third department, which has 4 grid squares. These can be arranged in 3 ways - 4X1, 2X2 and 1X4. Thus 48 different configurations are to be evaluated. These 48 arrangements are all evaluated to determine the placement rating. Notice that the placement rating will either be equal to the CR between departments 1 and 3 if dept 3 is in contact with dept. 1 only; CR between 2 and 3 if dept 3 is in contact with dept 2 only; or the sum of the two CRs if dept 3 is incontact with both 1 and 2. In the third figure of Figure 4.8 shows the selected placement. If there are only three departments then this will be the final layout. Otherwise the next department will have 20 potential locations.

After the layout is completed CORELAP calculates a numerical score for the layout. The numerical score is calculated using  $d_{ij}$ , the rectilinear distances between the centroids of the department, and the CRs as follows:

$$\text{Layout score} = \sum_i \sum_{j>i} CR_{ij} \times d_{ij} \quad 4.1$$

A small layout score indicates a good layout.

#### 4.2.4.3 Computerised Relative Allocation of Facilities Technique - CRAFT

CRAFT is an improvement procedure, which can be used to improve the layout obtained from another method or improve an existing layout. CRAFT requires an initial layout, cost and flow data. The cost and flow data is in the form of from-to matrices. The initial layout is given in the form of a grid with contiguous grid squares being assigned to departments. CRAFT attempts to improve the given layout by interchanging the locations of departments. Interchanges are only feasible if departments have the same area; or they share a common boundary.

#### 4.2.5 Evaluation of Layout Alternatives

The last stage of SLP is to evaluate the alternative layout to find the most desired layout or one to satisfy the goals. This can be established by an optimisation scheme.

The layout problem aims to locate  $n$  functional centres (machines, storage areas, plant etc.) among  $m$  possible locations, with  $1 < n < m$  and  $2 < m < \infty$ , according to some pre-established constraints of a functional, aesthetic or economic nature.

The location process comprises the generating of feasible location alternatives, which satisfy the basic requirements of the problem, and then choosing the optimal alternative from them.

Layout design involves location of  $n$  interrelated facilities with a given bounded area. The area may be sub divided into  $m$  potential sites ( $m > n$ ), with each site large enough to contain any facility.

The objective function for the system yielding the cost  $C$  for any solution is:

$$C = \sum_{i=1}^m \sum_{j=1}^m \sum_{k=1}^n \sum_{l=1}^n B_{kl ij} X_{ik} X_{jl} + \sum_{i=1}^m \sum_{k=1}^n a_{ik} X_{ik} \quad 4-2$$

under the constraints:

$$\sum_{i=1}^m X_{ik} = 1 \quad k = 1 \dots n \quad 4-3$$

$$X_{ik} = 1 \text{ or } 0 \quad k = 1 \dots n; \quad i = 1 \dots m \quad 4-4$$

which define the decision variables.

$a_{ik}$  cost of location of facility  $k$  at a site  $i$

$B_{kl ij}$  cost of relationship between facilities  $k$  and  $l$ , with  $k$  located at  $i$  and  $l$  at  $j$ .

$X_{ik}$  takes the value 1 if a facility  $k$  is located at the site  $i$ , and a value 0 otherwise. A set of values  $X_{ik}$  for all possible combinations of site and facility, defines one layout alternative.

The location problem as defined (Warszawski, 1972) by 4-2, 4-4 has  $m!/(m-n)!$  different solutions from which the optimal solution must be chosen. The other effects such as cost of exterior walls for architectural buildings should be added to this formula. Then the mathematical model takes the following form with these kind of additions.

$$C = \sum_{i=1}^m \sum_{j=1}^m \sum_{k=1}^n \sum_{l=1}^n B_{kl ij} X_{ik} X_{jl} + \sum_{i=1}^m \sum_{k=1}^n a_{ik} X_{ik} + C_L(L) \quad 4-5$$

with  $L$  the counter pertaining to any particular solution, and  $C_L$  a cost parameter such as the cost of exterior walls, of fence etc.

The total cost as defined by equation 4-5 is composed of the cost of relationship between the various facilities and of the cost of their location. In industrial layout problems, where the facilities to be located are mainly machines or storage areas, the relationship parameter  $B_{kl ij}$  will depend on the flow of materials between the facilities, and is often calculated from:

$$B_{kl ij} = C_{kl} Q_{kl} D_{ij} \quad 4-6$$

$C_{kl}$  cost of transportation between facilities  $k$  and  $i$  per unit of quantity and distance.

$Q_{kl}$  quantity of materials to be transferred between the facilities  $k$  and  $i$  per period of time.

$D_{ij}$  transportation distance between sites  $i$  and  $j$ .

### 4.3. Ship General Layout Problem

For cargo ships the layout should be such that the type of cargo for which the ship is designed can be carried as cheaply as possible in stowage arrangements that ensure it is delivered in good condition and with methods of loading and discharge that are speedy and economical (Watson, 1998).

The layout of a ship consists of the general arrangement which includes arranging the accommodation, anchoring, towing and mooring arrangement design as well as the cargo compartments. Ways of minimising building and operational costs must be considered.

Having generated a set of dimensions and a hull form, it is usually necessary for the designer to examine alternative arrangements of the main compartments in an attempt to optimise the use of the available space. As a starting point, the layout must be accommodated within and be built upon a lines plan developed to meet the deadweight, capacity, and speed requirements and which takes into consideration the needs of stability, trim, seakeeping and manoeuvrability.

The process of resolving such conflicts is the ability of the designer, with the aid of the system, to be able to produce alternative layouts and determine capacities and centroids. During the design of the layout of ships, the scope for innovation is often constrained by operational, regulatory and technical requirements which limit the number of alternatives which need to be examined.

The layout design procedure starts with a concept arrangement sketch and incorporates methods which permit the rapid conversion of the information shown on the concept sketch into a compartment definition, which in turn supports the generation of design information. The layout definition process is considered as the process of deciding on the disposition of main boundaries and structural elements within the confines of the defined hull form. The designer can build up any internal arrangement using representations of preliminary structural elements such as decks, flats, bulkheads, innerhulls and floors. The layout is developed using geometric primitives which are realistic representations of the major internal structural boundaries of actual vessel types (Hills et al., 1989).

The key to layout is convenient and adequate access for men, equipment and services both vertical and horizontal planes. Space is expensive, and hence the need for passageways must be minimised while at the same time giving generous space to those which are essential.

The use of SLP in general arrangement plan design is not possible, as the relationship between departments (compartments in ship arrangement terminology) and practical limitations of a specific ship type are not flexible. The arrangement of compartments are predetermined as the designer has to follow up: fore peak, cargo holds, engine room and aft peak arrangement. So the designer is left freedom only in the size of the compartment and internal arrangement of the compartments. The method is more of a heuristic rather than a formal approach with possible use of an iterative technique.

#### 4.4. Ship Accommodation Design

Ship accommodation area, or service area for the passenger ships and warships, has characteristics to be used in a modified SLP approach. However limitations such as regulatory regulations, social and environmental limitations necessitates the use of a combination of a heuristic and formal layout approach.

The objective function together with various requirements or constraints can be represented by a mathematical model. This mathematical model was explained in the above section and was suggested by Warczewski (1972). The formula 4-2 to 4-6 can be used for ship design. The total cost given by  $C$  is the summation of the cost of the relationship between various facilities and the cost of their location and is a function of the relationship parameter  $B_{klij}$ .  $B_{klij}$  will depend on the flow of personnel and materials between the facilities and can be calculated from formula 4-6. The evaluation of  $B_{klij}$  is obviously of fundamental importance and its evaluation for use in the layout design of shipboard areas can be difficult. In the case of accommodation arrangements, this relationship parameter may be deduced from a consideration of work and residence circulation, service routing, utilisation etc. The parameter  $a_{ik}$  can also represent the cost of environmental factors or functional considerations. Obviously the evaluation of parameter  $a_{ik}$  and  $B_{klij}$  is the most important and most difficult part of the analysis (Cort and Hills, 1987). A great deal of information is available in the form of design manuals etc. which provide data on environmental, operational and ergonomic requirements. The number of factors and relationships which can be considered, is only limited by the computing power available.

Cort and Hills (1987) explained that in practice layouts tend to be dominated by a relatively small number of primary factors including the followings:

1. minimum movement of personnel between functional areas.
2. minimum flow of materials between stores and operational areas.
3. environmental considerations.
4. efficient routing of service systems.

It is obvious that the primary consideration governing ship layout will be operational effectiveness, and the factors influencing layout are usually considered to be part of the design synthesis which facilitates the assessment of that effectiveness.

For the ship design system Cort and Hills (1987) suggested a relatively simple system to measure the cost of circulation and environmental costs.

$$COST_{MIN} = MIN \left[ \sum_{i=1}^n \sum_{j=1}^n a_{ij} d_{ij} + \sum A_i L \right] \quad 4-7$$

The first term is simply the cost of circulation of people and materials between activities  $i$  and  $j$ , a distance  $d_{ij}$  apart where:

$a_{ij}$       association of activity  $i$  and  $j$

$d_{ij}$       distance between activity  $i$  and  $j$

The second term of the performance criteria reflects the environmental cost of each activity where;

$A_i$       environmental loading.

$L$       weighting factor, this is the specified working level of the activity divided by the maximum permitted level for the vessel.

The typical environmental factors considered by Cort and Hills (1987) are:

1. level of machinery noise.
2. propeller induced noise.
3. ship motion.

#### 4.5. Conclusion

In this chapter facility layout problems have been discussed. As a first high level approach SLP was given to understand the layout planning technique. Some computerised layout methods were reviewed to lead the understanding of the computer applications and the difference between these methods and the proposed method.

Planning a facility is dependent on some factors such as flow of material and human and relations between facilities. These headings were reviewed in detail to show their importance in layout design.

Layout and layout optimisation problem for plants were summarised before explaining the ship general layout and ship accommodation layout design. General

ship layout has clearly prototype governed and is suitable for heuristic layout arrangement with highly hierarchical decomposition. Meanwhile accommodation layout is more of a multilevel layout problem with heuristics limitations. Hence two different approaches have been adopted in this work for these two problems.

General arrangement problem is solved with rule based Expert System solution with iterative solution governed by the available data. The iterations are not predetermined as in a procedural language approach. Instead a hierarchical decomposition is applied to ship as fore peak, cargo holds, engine room, aft peak and accommodation compartments. Each compartment has its own calculation procedures for its attributes. The appropriate calculation procedure is chosen according to available data. As rules are fired and more data becomes available different method are applied for the same attribute.

Accommodation block layout design is conducted by a heuristic decomposition dividing accommodation block into different decks. The spaces in each deck is determined by heuristic rule based Expert System approach with a knowledge base assembled from regulations and previous cases, i.e. similar ship database. A relationship approach as in the case of SLP is utilised for this process. Then each level is arranged firstly locating spaces by heuristics if applicable, and an arrangement algorithm is applied for the rest by use of a cost factor.

## **5. KNOWLEDGE BASE OF A CONTAINER SHIP DESIGN.**

### **5.1 Introduction**

This chapter describes the knowledge base which was developed for the purpose of assisting container ship layout design. The content and structure of the knowledge base are described in detail to present the current methodology.

CLIPS, an Expert System Shell, is used to prepare the knowledge base and to form the container ship design Expert System. The general capability of any Expert System is determined by the quality of the associated knowledge base. The main problems associated with the development of Expert System applications are nearly all concerned with the process of obtaining the information required to construct the application of a specific knowledge base. This process of extracting knowledge from either human or non-human sources is often called knowledge elicitation, and is of fundamental importance in the development of Expert System applications.

Presentation of knowledge derived from the experts and rule books in a computer environment is not a straight forward task which has to be conducted in parallel to the knowledge extraction. Two types of knowledge representations must be performed: the attributes of the components which form the ship, and the knowledge to assign values to these attributes.

A knowledge representation model consisting of objects with frames for the attributes and production rules for the generative knowledge was utilised, and syntax of this representation was represented.

### **5.2 Knowledge Elicitation in the Ship Design Domain**

The knowledge elicitation techniques which are commonly in use, were mentioned in Chapter 3.4.1. When the container ship general arrangement design problem is considered, it was decided that the interview technique would be the most suitable approach. The problem domain was divided into a number of sub-problems.

There are several possible sources in generating the container ship design knowledge through interviews as follows:

- naval architects (designers)
- shipbuilders
- ship owners/operators

The naval architects and shipbuilders have the first class expertise to be elicited for the domain. Some shipyard's owners and naval architects from Tuzla, the largest shipyard area of Turkey, have been chosen to apply the interview technique. Ship owners/operators provide an obvious source of expertise concerning aspects of container ship design. A number of ship owners/operators were also consulted to complete the knowledge base on operational aspects.

Several interviews were constructed in which the experts explained their method of reasoning. Some of the interview questions are given in Appendix C. The information obtained from interviews was used to produce the heuristic based production rules contained in the knowledge base.

Three types of questions have been prepared for different domain experts such as ship owners (Aka, 1999), naval architects (Göke, 1998, Şişmanyazıcı, 1999, Demirsoylu, 1999, Börü, 1999, Watson, 2000), researchers (Odabaşı, 1998, Gürsel, 1999, Welsh, 1994) and ship masters (Dayı, 1998) to apply the interview technique.

On the other hand some rule books such as SOLAS'74, ILO, SBG were applied for the knowledge elicitation of containership design. Apart from these sources, Welsh's PhD thesis (Welsh, 1989) was widely used to complete the knowledge base. The main source of knowledge base was, no doubt, the published works in the literature.

### 5.3 Knowledge Representation

In the first stage of knowledge elicitation, a knowledge representation model is required. Knowledge in ship design can be divided into two parts:

- a) attributes of components which are from the ship: this is actually factual database of ship components. In the current work an object oriented approach was adopted. This factual knowledge must be shared in user interface and in the reasoning mechanism. The user interface was programmed in DELPHI in which this type of knowledge was represented by objects and frames attached to the objects to represent the linkage between the objects is also established as given in Appendix D.

Factual knowledge about objects were represented by deftemplates in CLIPS environment for the reasoning process. This representation enables storing of attributes between two process without any difficulty.

- b) generative knowledge: knowledge required for the reasoning process was only modelled in CLIPS and this is completely hidden from the user interface. This knowledge was represented by production rules enabling forward chaining process. The knowledge extracted from the experts was represented mainly by this type.

5.4 The Knowledge Base

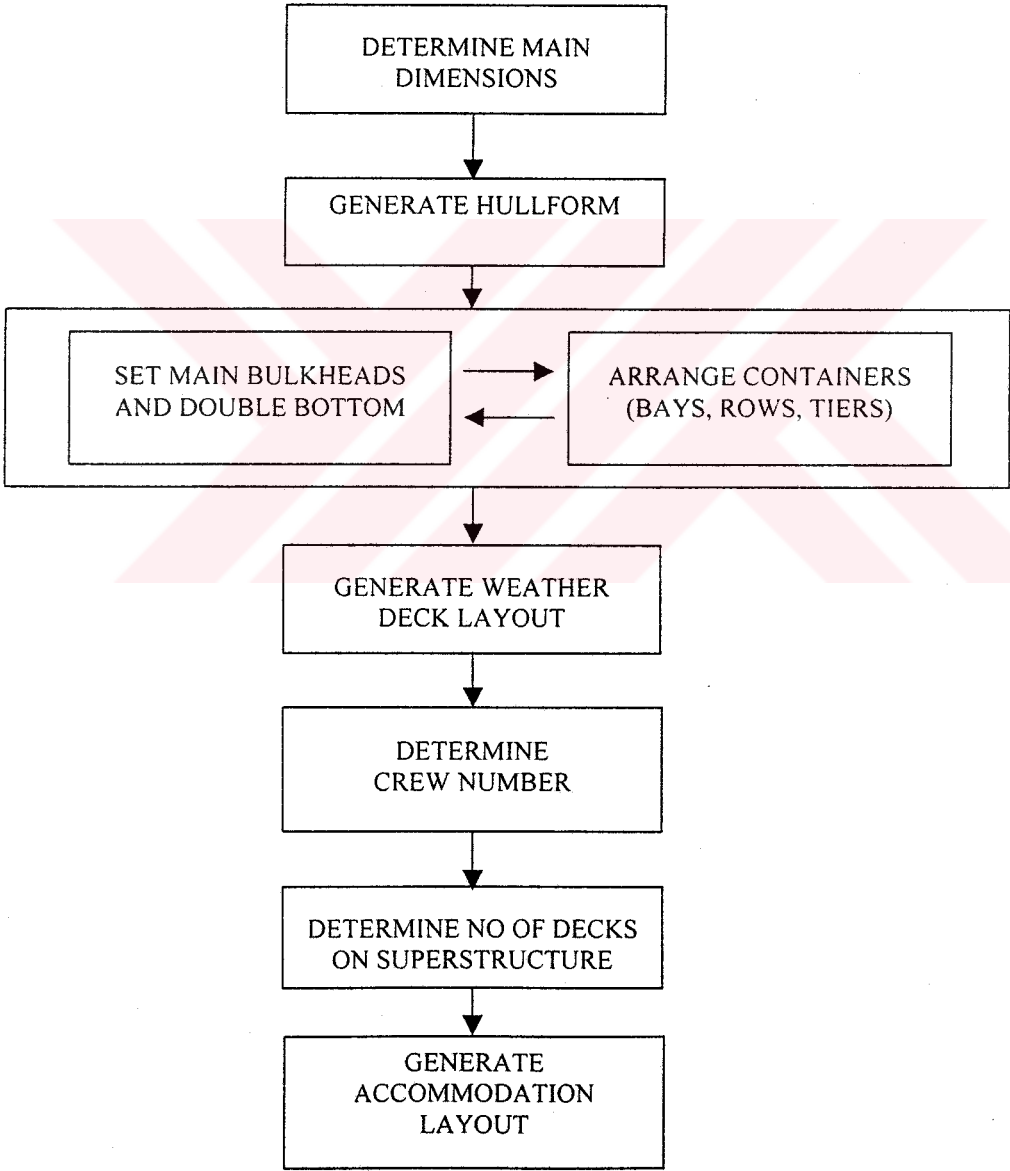


Figure. 5.1. The basic steps in container ship design

The basic steps of container ship design are summarised in Figure 5.1. This figure shows the sub-areas, which compile the scope and structure of the knowledge base. This knowledge base has been prepared for container ships, in which capacity is between 650 TEU and 1500 TEU. They are called feeder container ships and widely utilised in Turkey.

#### **5.4.1 Container Ship Design**

As container ships carry a substantial proportion of their cargo on deck; it is not possible to base the design on the required underdeck cargo volume. In these circumstances, stability considerations take over the primary role in the determination of the main dimensions.

For the maximum economy in the design of any container ship, the dimensions should be such that containers can be stacked up to the limit permitted by the stability requirements. The upper tiers should be reserved for relatively lightly loaded containers, in order to maximise the number of containers on deck.

For each number of tiers of containers carried there is an associated breadth of ship which will provide the KM necessary to ensure stability, whether the tiers are enclosed below hatch covers or carried on deck being a second order effect. Ships designed to achieve a particularly high KM for a given breadth have an advantage in minimised steel weight and ship cost but have a penalty for powering and seakeeping.

Speed has an effect on the number of containers that can be carried in a ship of certain dimensions partly because of its influence on the block coefficient and machinery power and thus on the engine room dimensions. This is then adjusted in association with the length required for the engine room, peak tanks, cell guides, bulkhead stiffeners etc., so that the cargo spaces are tailored to a multiple of container lengths. A first approximation to the dimensions of the ship can be made by predicting the number of containers in longitudinal, vertical and transverse direction both for under deck and above deck containers. If an allowance is made for the loss of containers in the forward part of the hull due to reduced breadth of the hullform. This allowance can be checked when the hullform is determined by allocating space for each container.

### 5.4.2 Case Based Design Step

A database consisting of existing ships is one of the powerful tools in ship design. Hence a database of 60 container ships with capacity range of 500 TEU and 2000 TEU has been compiled by extracting suitable sized ships from Alasehir (2000). Some of these ships are given in Table 5.1 to demonstrate main characteristics of the database. This database is also used to derive some heuristic rules in container ship design as described in this chapter.

Table 5.1 Some sample container ships (Alasehir, 2000)

| Name of Ship      | DWT<br>(design) | TEU<br>(Total) | Loa    | Lbp    | B     | D     | T<br>(design) | Gross | V<br>(service) |
|-------------------|-----------------|----------------|--------|--------|-------|-------|---------------|-------|----------------|
| TROPIC TIDE       | 8304            | 400            | 121.20 | 110.00 | 22.00 | 9.60  | 6.20          | 6536  | 15.70          |
| SLOMAN CHALLENGER | 5665            | 403            | 100.70 | 94.70  | 17.80 | 8.20  | 6.40          | 4489  | 145.00         |
| BARRINGTON ISLAND | 10358           | 434            | 179.90 | 165.00 | 25.20 | 13.20 | 8.00          | 14061 | 22.20          |
| ARKTIS FIGHTER    | 5212            | 444            | 101.10 | 93.60  | 18.80 | 9.30  | 6.00          | 4980  | 15.70          |
| GERMANIA          | 8790            | 645            | 125.20 | 116.95 | 19.00 | 13.50 | 8.00          | 8790  | 16.00          |
| FROTABELEM        | 10640           | 666            | 133.45 | 124.40 | 22.80 | 10.70 | 7.80          | 8801  | 17.50          |
| CARMEN DOLORES    | 11004           | 754            | 135.61 | 125.72 | 20.50 | 10.50 | 8.26          | 7424  | 18.80          |
| SEA ARCTICA       | 8500            | 780            | 134.00 | 118.50 | 24.00 | 15.10 | 7.65          | 11600 | 17.00          |
| JORK              | 11870           | 1122           | 157.30 | 147.00 | 23.50 | 12.80 | 8.30          | 12029 | 19.89          |
| CMBT ENDURANCE    | 13100           | 1129           | 158.75 | 145.00 | 24.00 | 13.90 | 8.50          | 14116 | 18.70          |
| CLIPPER FANTASY   | 28000           | 1130           | 181.00 | 172.00 | 26.00 | 14.40 | 9.67          | 19354 | 14.00          |
| GANTA BHUM        | 15027           | 1228           | 151.20 | 141.20 | 25.00 | 13.60 | 9.00          | 13188 | 19.00          |
| MARWAN            | 18985           | 1388           | 167.24 | 156.71 | 25.00 | 13.40 | 9.50          | 14865 | 18.60          |
| NORDLAKE          | 22450           | 1496           | 179.23 | 167.26 | 25.30 | 13.50 | 9.90          | 16202 | 19.20          |
| SAN LORENZO       | 17205           | 1512           | 166.62 | 156.00 | 27.40 | 13.20 | 8.75          | 15778 | 19.50          |
| WESTERDICH        | 17600           | 1572           | 168.00 | 156.00 | 26.70 | 14.40 | 9.70          | 15908 | 20.00          |
| CONSHIP PACIFIC   | 23276           | 1684           | 164.10 | 153.70 | 27.50 | 14.30 | 10.66         | 16282 | 19.40          |
| CECILIE MAERSK    | 19530           | 1879           | 190.48 | 180.15 | 27.80 | 15.23 | 8.25          | 20842 | 19.00          |

A generalisation method was applied to the case in the database to derive some heuristic rules to be used in the general arrangement of the hull. Some formulas were extracted based on the past cases and these formulas (knowledge) were represented as rules in ALDES system as first approximations.

### 5.4.3 The First Approach to Main Dimensions

The main dimensions of a containership are direct functions of the size of the cargo units to be carried by the vessel. As a result of this relationship, the main dimensions can be determined according to standard container sizes.

Munro-Smith (1975) gave a simple method of estimating ship preliminary dimensions as explained in this section.

The capacity, speed and range of a new ship are known at the start of the design as these are usually the owner's requirement. The first step is to decide and determine the number of under deck containers. The graph in Figure 5.2 extracted from Watson and Gilfillan (1976) can be used to estimate the number of rows and tiers of containers carried under deck according to ship speed and capacity. If the number of rows and tiers are known the number of bays can be determined from the total capacity. The main dimensions length (L), breadth (B) and depth (D) are estimated using data acquired from the under deck number of rows, tiers and bays.

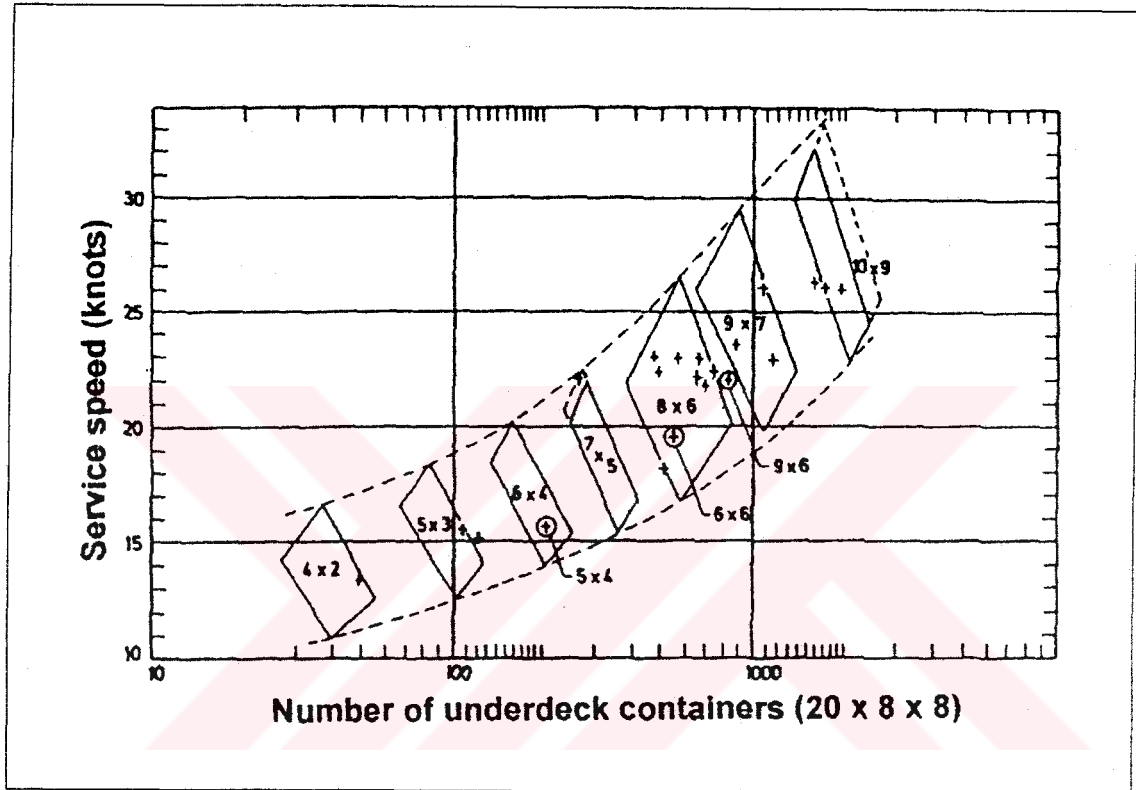


Figure 5.2 Diagram for first approximation of number of container rows and tiers under deck (Watson and Gilfillan, 1976)

For preliminary design work, it can be assumed that the length required for each container is  $(l+1.5)$  m where  $l$  is length of a container (Munro-Smith, 1975) by making allowances for clearance and cross-ties. The  $L_{BP}$  of the ship is the sum of the container portion, engine-sterntube and portion forward. Thus:

$$L_{BP} = L_c + L_e + L_f + L_{BCC}$$

where  $L_c$  = Container portion =  $N_x \cdot (l+1.5)$

$N_x$  = Number of containers in the length.

$L_e$  = Length of engine room and after peak tank

$L_f$  = Length of fore peak tank.

$L_{BCC}$  = Length between most forward cargo hold and collision bulkhead.

As a first approximation the length of the aft peak tank can be taken as 3.5% of  $L_{BP}$ . The length of fore peak tank can be taken as 5%  $L_{BP}$  and length of space forward of container length can be taken as 10%  $L_{BP}$ .

The length of the engine room ( $L_{ER}$ ) is related to the engine length. A review of engine power requirement in existing ship database has indicated the required engine power between 5000 kW and 20000 kW. The utilisation of diesel engines both low speed and medium speed, have been assumed for the current work. The relation between engine speed and power for the database is given in Figure 5.3.

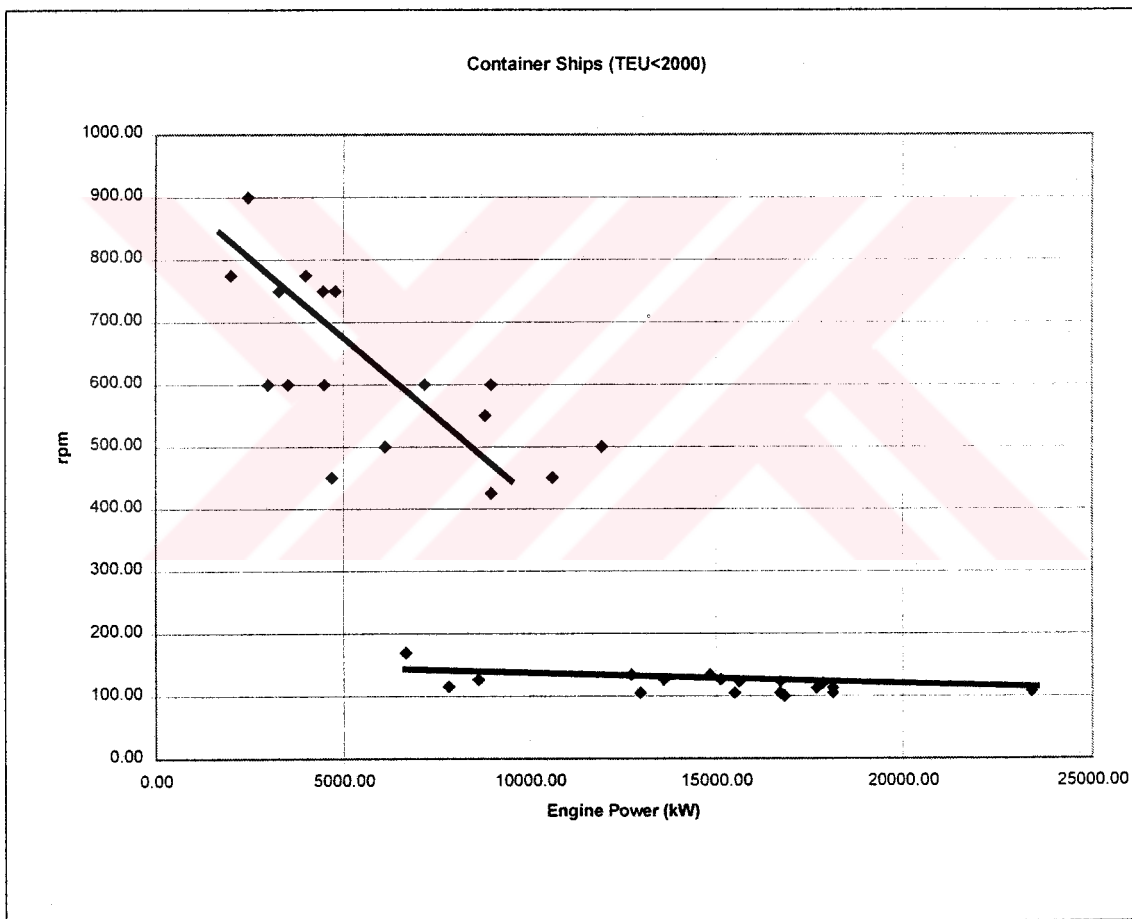


Figure 5.3 Engine speed versus power

The procedure to determine the required engine power by employing the database is adopted as such: approximate deadweight is determined from Figure 5.4 by using container capacity. Secondly a constant deadweight/displacement ratio is adopted from Figure 5.5 to estimate the displacement. And lastly engine power is calculated from Figure 5.6 by using Admiralty coefficient approach.

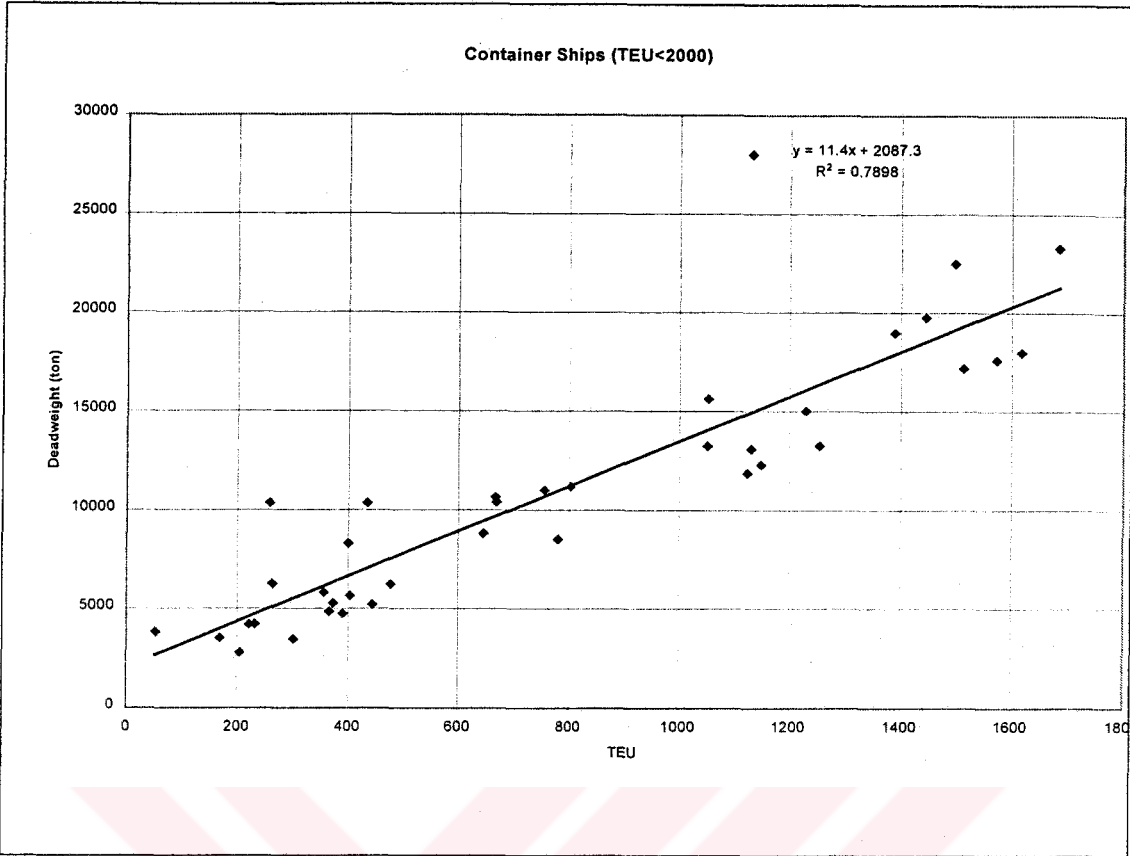


Figure 5.4 Deadweight versus capacity



Figure 5.5 Deadweight versus displacement

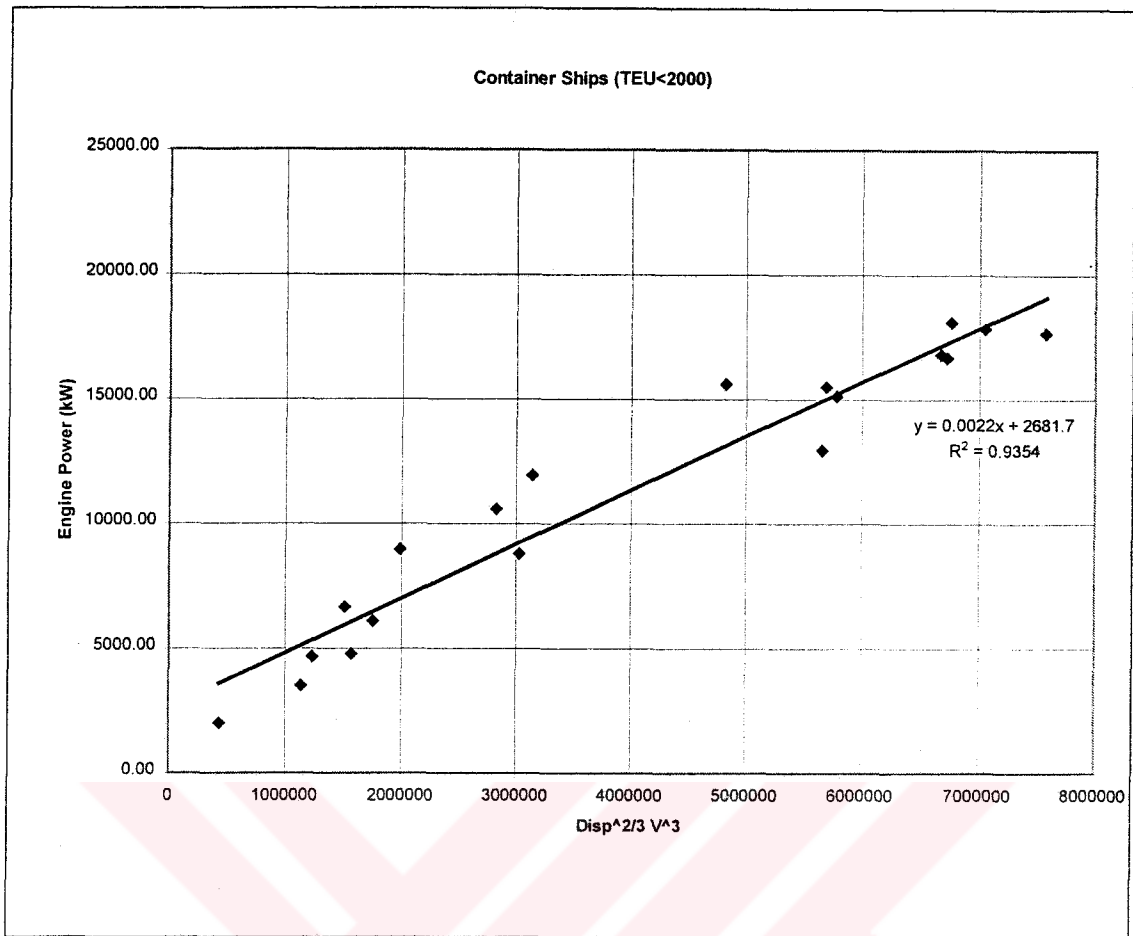


Figure 5.6 Engine power versus Admiralty coefficient

The engine length is dependent on the engine power and engine speed. A review of both medium and low speed engines from 5000 kW to 20000 kW was conducted and a relation between engine length and power is found as given in Figure 5.7 and 5.8 for low speed and medium speed engines respectively.

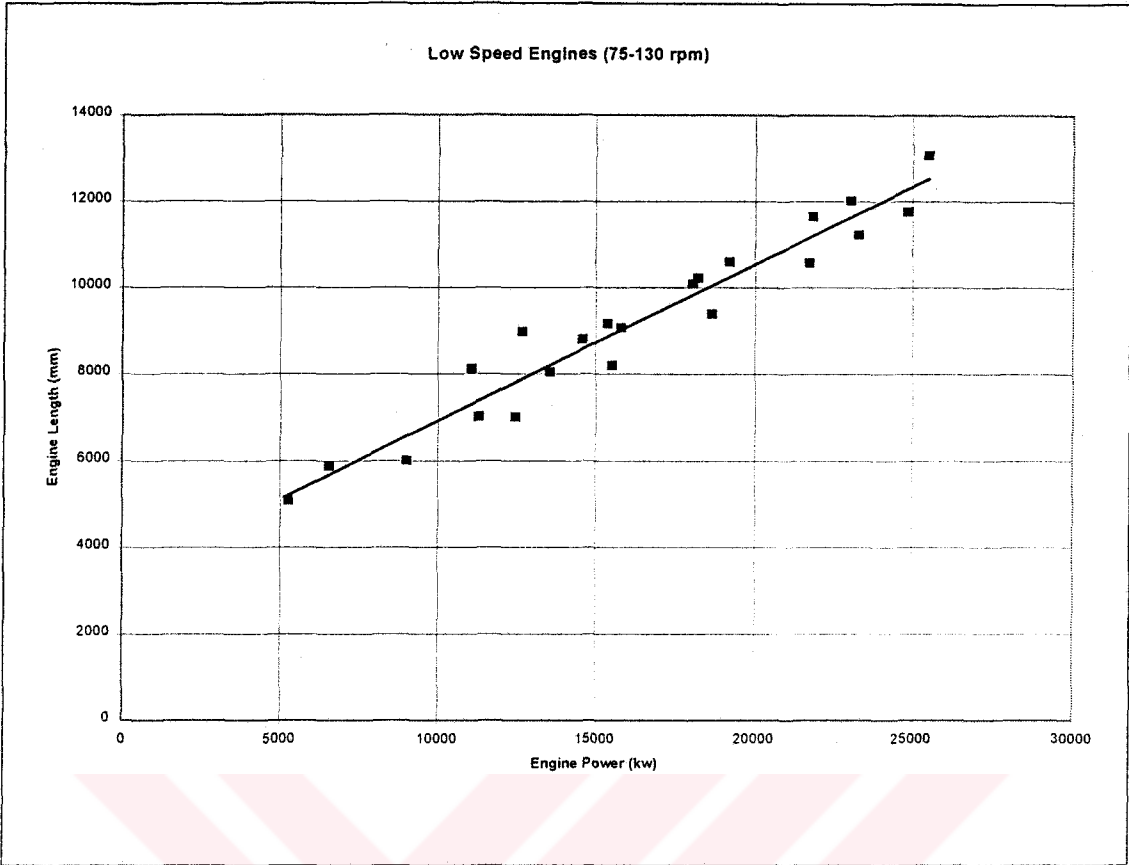


Figure 5.7 Low speed

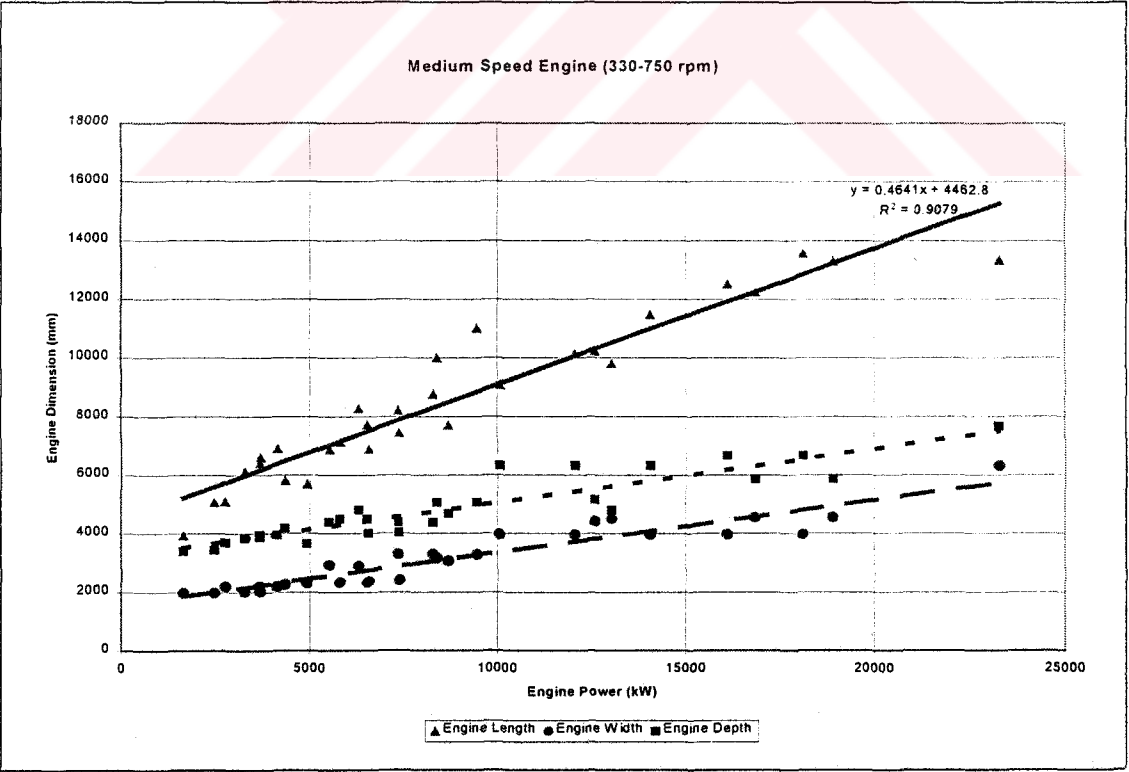


Figure 5.8 Medium speed

After the engine length was determined using the engine power data, the length of engine room can be calculated by assuming the engine room length is equal to length of the engine plus 0.8 metres distance between engine and engine room fore bulkhead and 5 metres distance between engine and stern-tube bulkhead. The length between perpendiculars can now be calculated from the following formula. The lengths which are important to calculate the ship length are represented in Figure 5.9.

$$L_{BP} = 0.185 L_{BP} + N_x (l+1.5) + L_{ER}$$

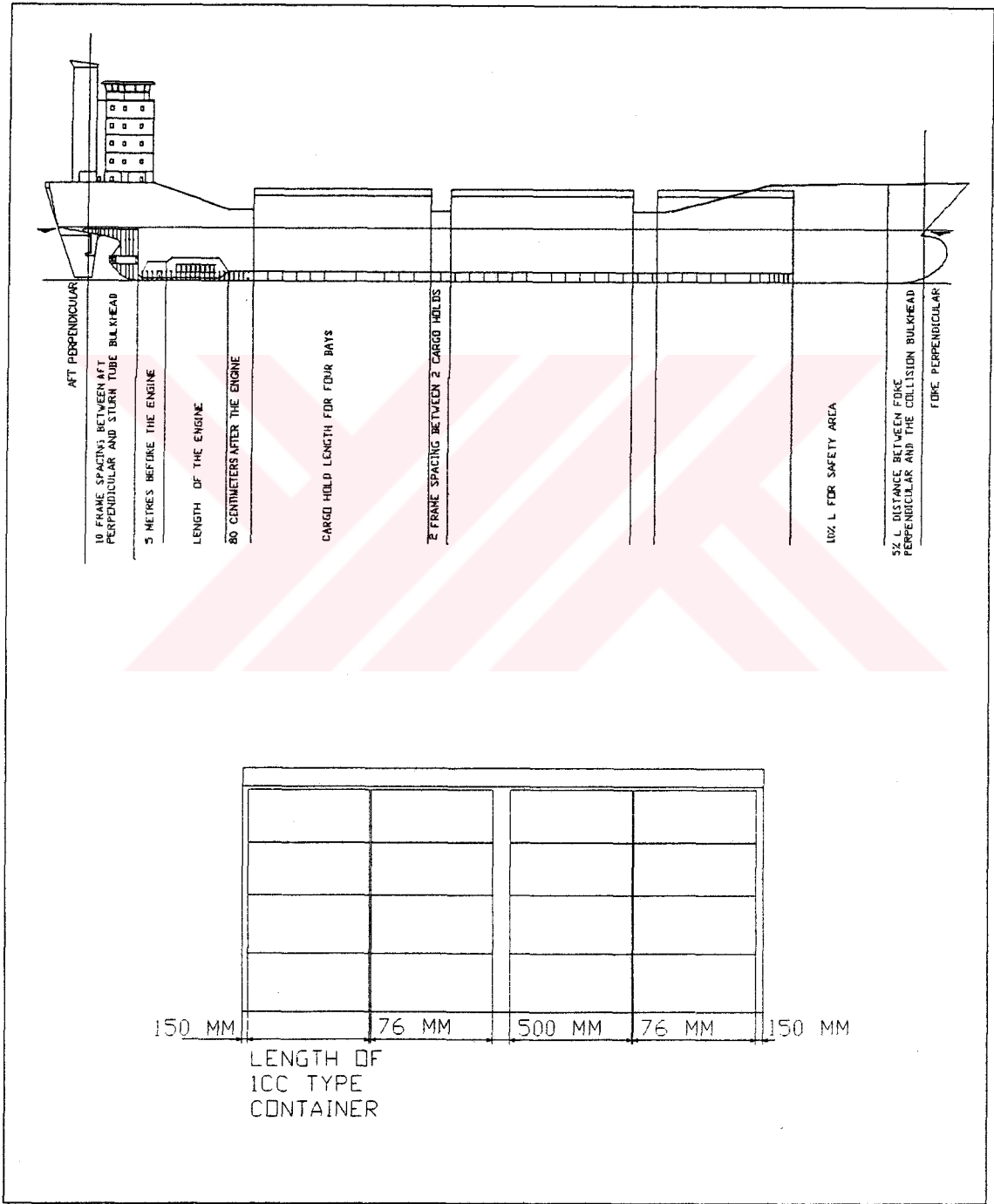


Figure 5.9 Ship profile

Containers have a standard width of 2.43 m. However, each container requires an allowance for clearance, guides, etc. of about 240 mm (Munro-Smith, 1975) so that each container requires a width of 2.67 m. Thus the number of rows ( $N_y$ ) cells located transversely in the ship require 2.67  $N_y$  m. Since the width available for containers is about 80 percent of the ship's breadth  $B$ , then

$$0.80 B = 2.67 N_y$$

$$B = 3.34 N_y$$

$N_y$  : Number of tiers of containers in holds.

The important aspects to calculate the ship depth and breadth are given in Figure 5.10.

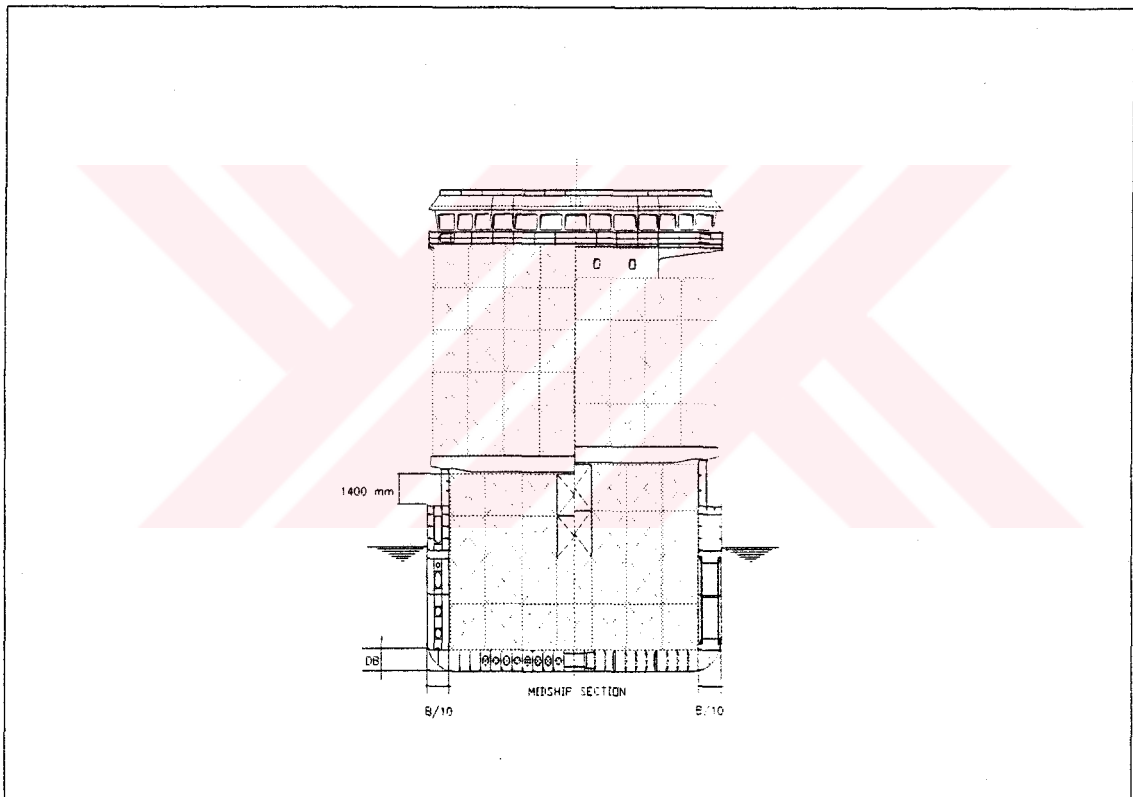


Figure 5.10 Ship depth and breadth

The depth of the ship is in generally controlled by the number of containers to be carried vertically. Thus:

$$D = N_z H + DB - 1.40$$

where  $N_z$  = Number of tiers of containers in holds

$H$  = Height containers

DB = Depth of double bottom.

The main ship dimensions have been calculated related to 20 feet container sizes. It is assumed that one 40 feet container can be taken into account as equivalent to two 20 feet container. No considerations have been given to non-standard containers except for ISO 668 1CC, 1AA size containers (2.59 m in height).

Deadweight and grosstonnage are prime factors in determination of general layout, crew number etc. Hence deadweight and displacement are determined as above from Figure 5.4 and 5.5. Grosstonnage is also determined from the database as given in Figure 5.11

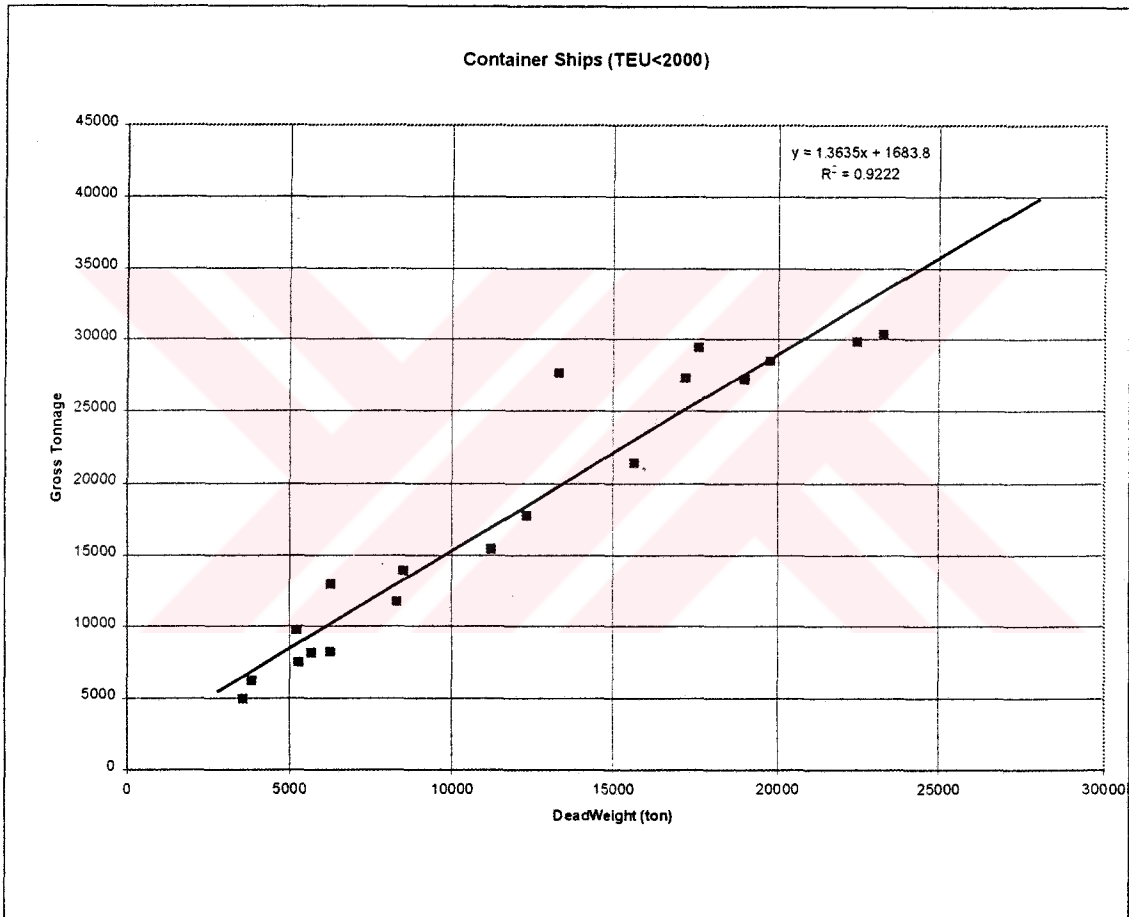


Figure 5.11 Deadweight versus grosstonnage

The block coefficient of the ship can be calculated empirically from the following formula (Munro-Smith, 1975).

$$C_B = 1 - 0.23 \frac{V}{\sqrt{L}}$$

If length, beam, displacement and block coefficient are known then the draft of the ship can be determined from displacement equation as follows:

$$\Delta = L B T C_B 1.025$$

All the principle dimensions have been calculated as a first approximation.

#### **5.4.4 The First Approach to Hullform**

Having defined the dimensions and form coefficients of the hull, the shape of the form can be decided. This has the advantage of facilitating the generation of a database of offsets that can be used in the definition of compartments and boundaries of space (Cort, 1987).

An effective way of generating and developing a hullform is to use a method in corporation with interactive graphics. The procedure to develop a new form would require a large number of Expert System heuristic which is out of the current work scope.

Another well-known and commonly used method is the basis ship hullform input. For this approach a basis ship database is needed. A form is taken from this database. The dimensions of the basis ship are modified to reach the desired result. After that the block coefficient and longitudinal centre of buoyancy are changed by deforming sectional area curve.

In this work, the hullform is assumed to be known and be directly taken from a sample ship.

#### **5.4.5 General Arrangement**

After estimating the main dimensions the next stage is to generate general arrangement of the ship.

The design of the general arrangement of a ship is the most important aspect of the whole design process. The decisions made at this stage have a far reaching effect on the final outcome of the design. The general arrangement of a ship has a significant effect on trim, stability, strength etc. A prototype has been establish in the general arrangement design as variations can reach unacceptable number of alternatives. The description of the prototype shall be given in the following section.

#### 5.4.5.1 Set Main Bulkheads and Double Bottom

The first stage in the generation of a preliminary general arrangement is the positioning of the double bottom and the main transverse sub-division bulkheads to form the holds, engine room, and peak boundaries.

The method for positioning the main bulkheads and the double bottom in the current work is based upon the requirements of Turkish Lloyd (1996) (TL).

Double bottom is assumed to be extending from forward collision bulkhead to engine room forward bulkhead in the current prototype. According to TL rules, on cargo ships: “A double bottom is to be fitted extending from the collision bulkhead to the after peak bulkhead, as far as this is practicable and compatible with the design and service of the Ship” (Turkish Lloyd, 1996). Depth  $h$  of the centre girder is not to be less than:

$$h = 350 + 45 B \text{ [mm]}$$

$$h_{\min} = 600 \text{ [mm]}$$

B: The greatest moulded breadth of the ship in metres.

After the height of the double bottom is determined, the watertight subdivisions will be positioned to find out the cargo hold positions. The watertight subdivisions will be placed by using TL rules and regulations (Turkish Lloyd, 1996).

The length of a container hold and the position of the main transverse bulkheads is a function of the size of the container units to be carried and the number of individual container bays along the length of the hold. Allowance must also be made for structural and stowage clearances between the individual container bays and adjacent hatches and holds. Therefore it is necessary to determine the number of container bays to be accommodated in the hold in order to determine the overall length of a particular cargo hold. It should be appreciated that the number of hatches/holds required to accommodate the total number of container bays must also be considered.

According to TL “All ships are to have a collision bulkhead, a stern tube bulkhead and one watertight bulkhead at each end of the engine room. In ships with machinery aft, the stern tube bulkhead may substitute the aft engine room bulkhead.”

For ships without longitudinal bulkhead in the cargo hold area, the number of watertight transverse bulkheads should, in general, be not less than given, as in Table 5.2. The ships which are longer than 185 m are not included in the knowledge base. The capacity of such ships is out of current work scope, i.e. over 2000 TEU.

Table 5.2. Number of watertight transverse bulkheads

| L* [m]             | Arrangement of aft | Machinery space elsewhere |
|--------------------|--------------------|---------------------------|
| $L \leq 65$        | 3                  | 4                         |
| $65 < L \leq 85$   | 4                  | 4                         |
| $85 < L \leq 105$  | 4                  | 5                         |
| $105 < L \leq 125$ | 5                  | 6                         |
| $125 < L \leq 145$ | 6                  | 7                         |
| $145 < L \leq 165$ | 7                  | 8                         |
| $165 < L \leq 185$ | 8                  | 9                         |
| $L > 185$          | to be specially    | considered                |

The rules for locating the collision bulkhead and stern tube according to TL are the following:

“Cargo ships with  $L_C^* \leq 200$  m. shall have the collision bulkhead situated not less than  $0.05 L_C$  from the forward perpendicular. Cargo ships with  $L_C > 200$  m. shall have the collision bulkhead fitted at least 10 m. from the forward perpendicular”.

“All cargo ships shall have the collision bulkhead located not more than  $0.08 L_C$  from the forward perpendicular. Upon application greater distance may be approved in special cases”

“All ships are to have a stern tube bulkhead which is, in general, to be arranged so that the stern tube and the rudder trunk are enclosed in a watertight compartment”

---

\* L is the distance, on the summer load waterline from the fore side of stem to the after side of the rudder post, or the centre of the rudder stock, if there is no rudder post. L is not less than 96% and need not be greater than 97% of the extreme length of the summer load waterline.

†  $L_C$  is to be then as 96% of the total length on a waterline 85% of the least moulded depth measured from the top of the keel, or as the length from fore side of the stem to the axis of the rudder stock on that waterline, if that is the bigger.

The location of the aft peak bulkhead is less clearly defined with its position being largely determined from consideration of the shaping of the aft end of the vessel and the need to accommodate the propeller shaft and the associated fittings etc. 5-8 times frame spacing distance is required to place a stern tube (Şişmanyazıcı, 1999). Therefore the stern tube bulkhead should be placed to reserve enough distance for the stern tube.

Since the container cells must be located in the most rectangular part of the ship, it is essential to avoid loss of space by the hullform so that the propelling machinery be placed as aft as possible enabling placement of main engine. The space below deck forward of the container portion can be used for water ballast, fuel oil, general cargo.

As the number of cargo holds are determined from number of container bays, then the next step in the development of a general arrangement is to obtain the number of transverse bulkheads to be incorporated in the ship.

Generally the owner requirements and container lengths determine the length of each hold, but this should be checked for the damage stability aspects at the next stage of the design.

#### **5.4.5.2 Engine Room Size and Location**

The engine room is one of the main spaces. There is no general rule to determine the space requirements for the propelling machinery.

The size of an engine room is dependent on the space required for the installation, operation and servicing of the equipment accommodated. The biggest item in the engine room is the main engine. Hence the size of the engine should be known before deciding the size of the engine room (Welsh, 1989). The calculation of engine room length for the preliminary analysis was given in Section 5.4.3 to determine the length between perpendiculars. This arrangement should be re-evaluated when the main engine is selected.

In order to determine the type of main engine to be installed, the power of the engine should be estimated (Welsh, 1989). This can be achieved with different approaches as the parameters are determined such as: past cases approach can be used in the preliminary stage, meanwhile regression method such as Holtrop-Mennen can be utilised when the form parameters are defined. In addition numerical methods can be employed if the hullform is sufficiently determined (Watson, 2000).

The machinery space, for cell guide type container ships, are usually located well aft with generally not more than one cargo hold between the machinery space and the stern. With the aft location there is no interruption of crane movement in way of the container stowage or interference of a deckhouse with a shore crane. There is no shaft alley to interfere with container stowage. Only aft location of engine room is utilised in the current work as the container capacity between 650 TEU and 1500 TEU.

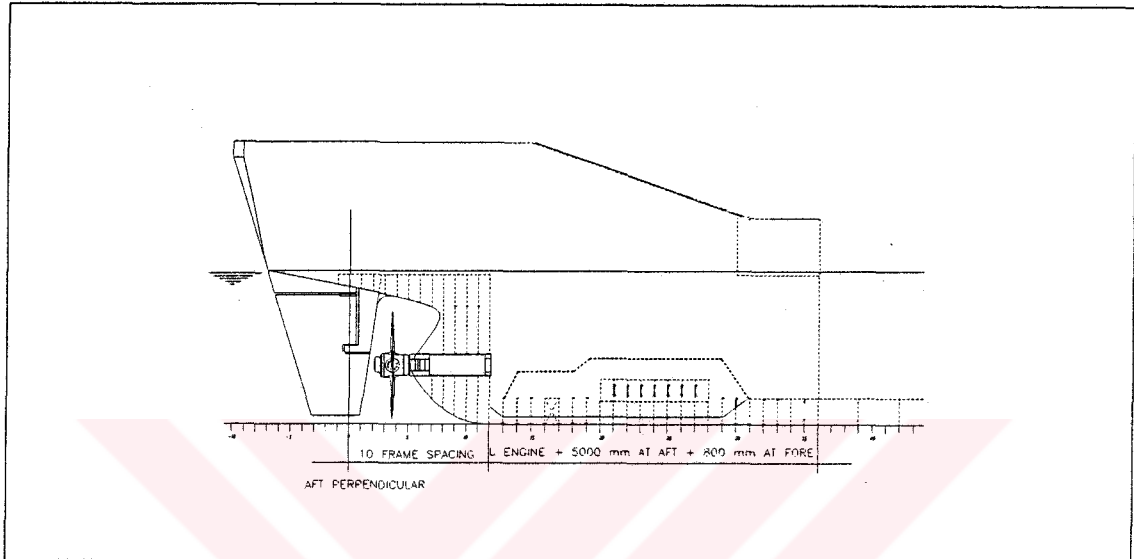
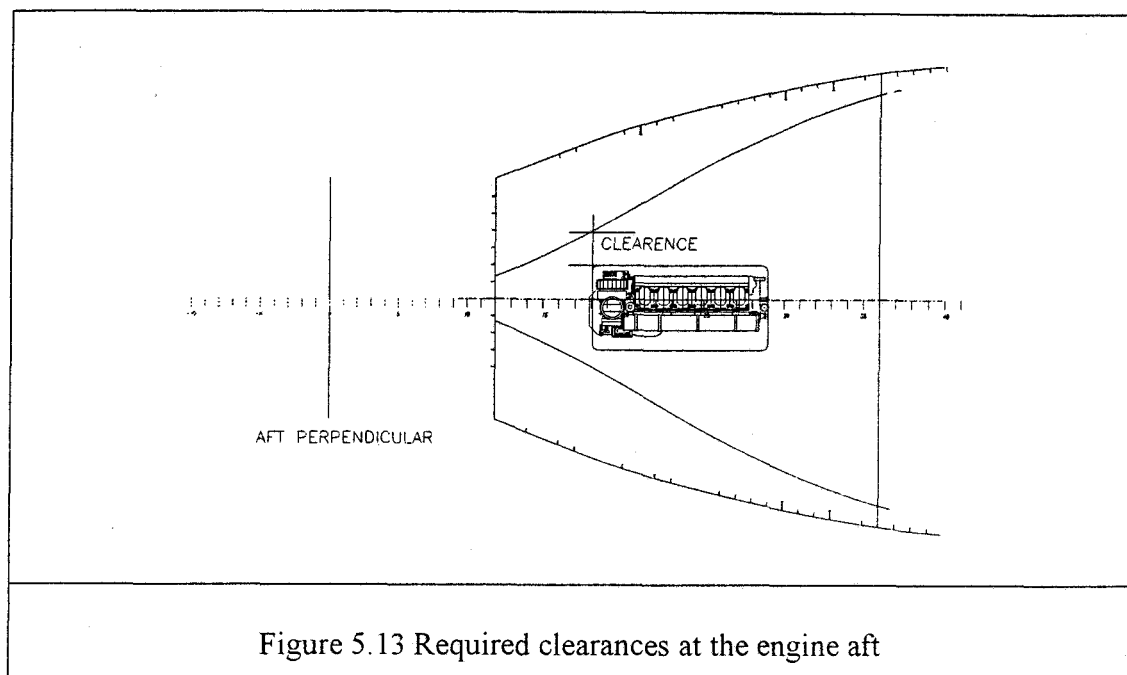


Figure 5.12 Determining the vertical location of the main engine

The general arrangement of the profile of the ship is not only dependent on the length of the engine room but also on the exact location along the length of the ship. This location is primarily dependent on the breadth of engine room tank top required in order to accommodate the engine with its associated minimum clearances (Welsh, 1989). Obviously it is necessary to fix the height of the engine room tank top above the base of the vessel. The main consideration when attempting to determine the height is the alignment of the propeller shaft, with the height being chosen so as to ensure that the shaft is parallel to the baseline of the ship, as shown in Figure 5.12.

At this stage, it is assumed that the diameter and the position of the propeller are known by placing maximum diameter propeller complying with the aft form and required clearances. The required height of the tank top can be obtained by taking the height of the engine output shaft above the engine base from the height of the propeller shaft above the ship base (Welsh, 1989). The required clearances are given in Figure 5.13.



The length of the engine room is a direct function of the length of the main engine. As stated before, engine room length is determined by adding 0.8 metres at the fore end and 5 metres at the end of main engine length. A number of additional factors should be taken into account such as the type and position of power take-off devices, auxiliaries, services etc when the main engine is selected.

#### 5.4.5.3 Tanks

Tanks are one of the main spaces distributed over the ship (Taggart, 1980). The main considerations governing the location of tanks are:

- no access is required except the manholes.
- if the liquid density is relatively high, then the location should be low to improve the stability.

**Fuel Oil Tanks:** these tanks are usually located in the double bottom to reduce risk of fire. The double bottom tanks are subdivided as required to control free surface. Some of the double bottom tanks should be kept separated as ballast tanks to maintain stability when fuel is consumed.

**Fresh Water Tanks:** fresh water is used as potable water, washing water and water for the machinery plant. The latter may be in the form of reserve feed water for the steam plant and cooling water for diesel or gas turbine plants. Just as for fuel oil tanks, the water tanks should be located with regard to both intact and damaged

stability aspects, and are normally placed near the machinery space and accommodation to minimise piping.

Ballast tanks: ballast, either liquid or solid, is carried to maintain stability and sea-kindliness. In general cargo ships, the ballast tank should be distributed over the length of the ship for flexibility in controlling trim. The combined centre of gravity of the ballast tanks should ideally be near or below the combined centre of the fuel tanks in order to maintain proper stability. The carriage of water ballast is an important feature of the container ship concept due to the unusual nature of the cargo carried.

#### **5.4.5.4 Cargo Hold Container Arrangement**

For the determination of the actual required depth, breadth and length of the vessel, generation of container bays, rows and tiers is very important. The depth of a container ship is directly dependent on the size of the container unit to be carried. The heights of individual containers can vary quite considerably with deviations from the standard 2438 mm ISO containers. The container standards are given in Figure 5.14. If a ship is to be designed with the capability of accommodating containers of different heights, then the depth of the ship must be determined with reference to the greatest container height.

As it was mentioned before the number of container rows and tiers under deck is taken from Figure 5.2 in the first stage and the number of bays were calculated according to these numbers.

Having decided upon the maximum height of container, the depth of the vessel can be determined by consideration of the number of tiers of containers to be accommodated. The double bottom has already been calculated in Section 5.4.5.1. The height of the hatch coaming is usually assumed to be 1.40 metres according to designers in Turkey (Göke, 1998). From Figure 5.15 the thickness of a twistlock varies between 25 mm to 30 mm, it is assumed to be 30 mm throughout the thesis (MacGregor-Conver, 1997).

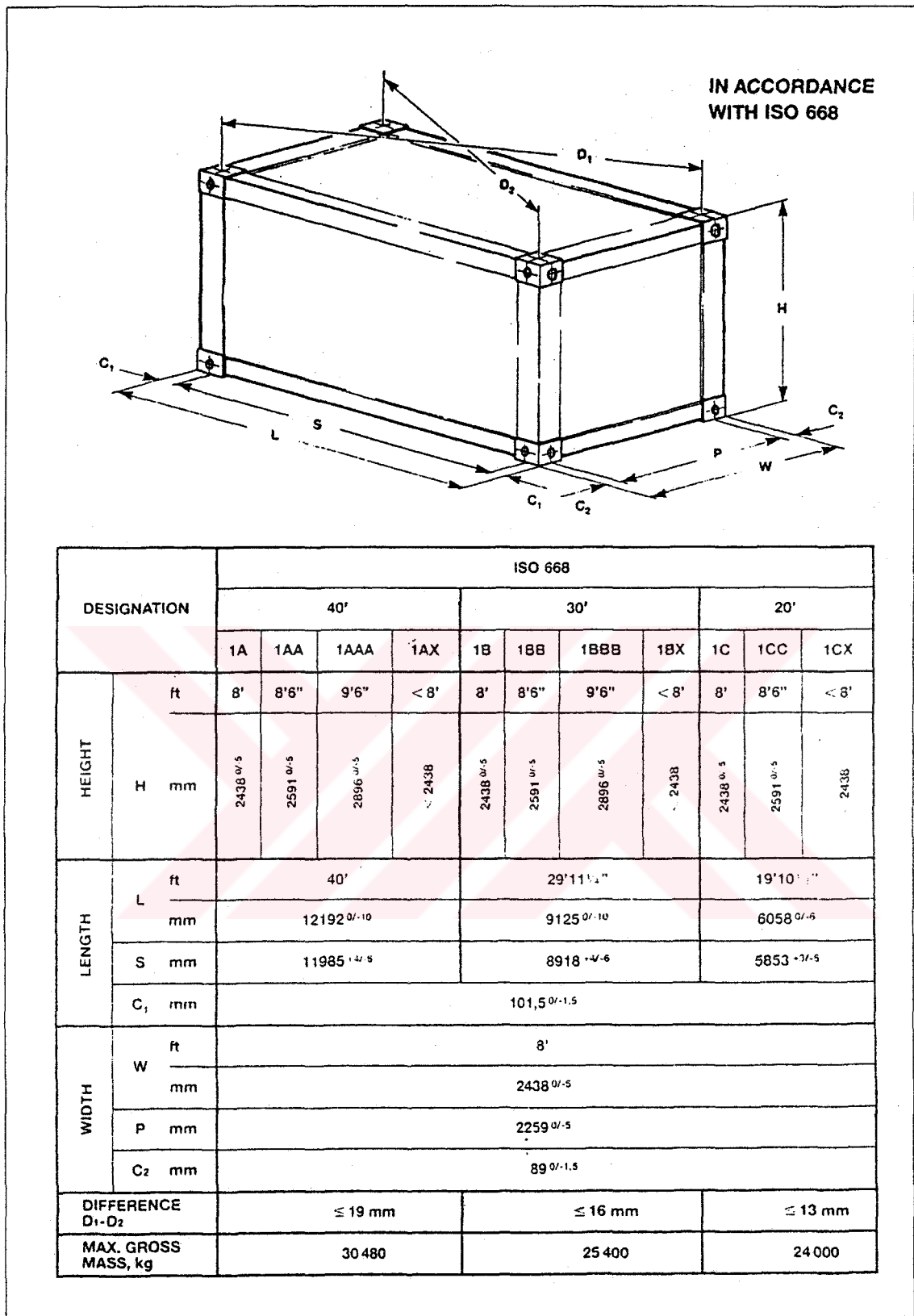


Figure 5.14 Container standards

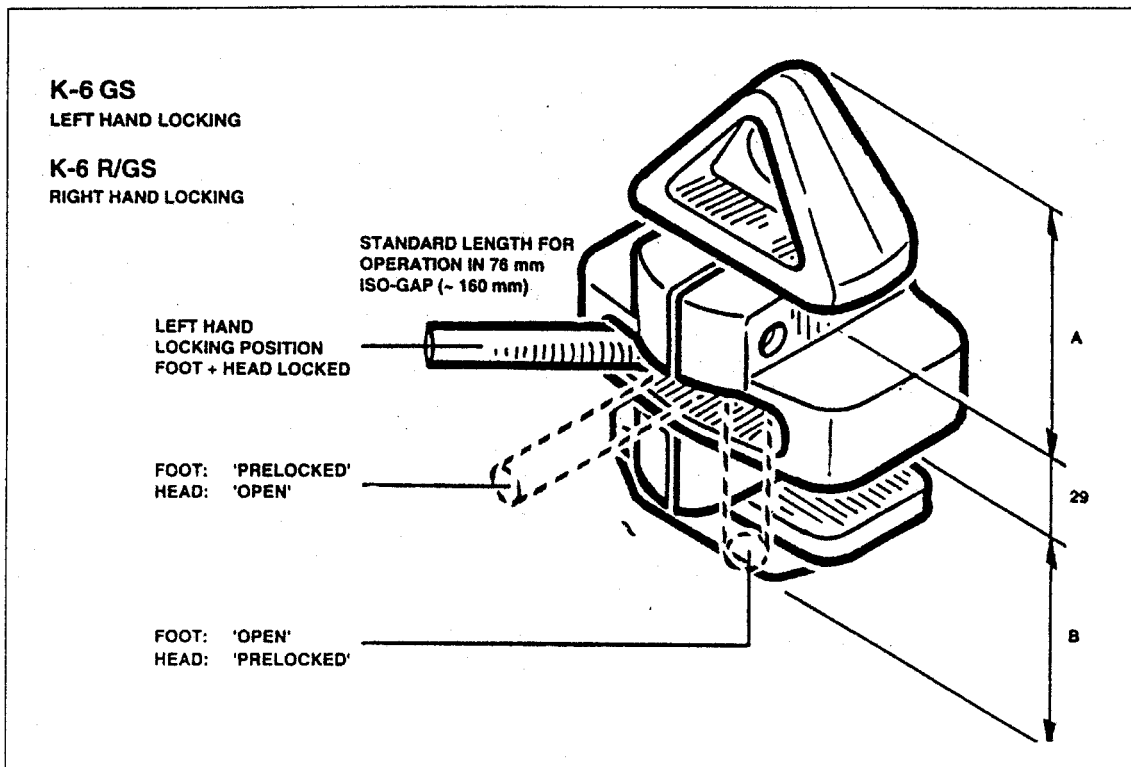


Figure 5. 15 Typical twistlock

There must be a clearance between the container and the guides in order that the containers may slide freely in the cells. The average clearance is about 20 mm at each end and 12.5 mm at each side (Munro-Smith, 1975). Figure 5.16 represents a typical cell guide system according to MacGregor-Conver (1997).

For a container ship it is very important to maximise the number of container slots which can be accommodated within the hatchways, thus the hatchways could be as large as practically possible. Large single hatches can cause problems when considering the operational aspects of container ship design, in that it is more difficult to optimise container stowage arrangements in order to avoid problems such as overstay, where there is only a single hatchway in each hold. Multiple hatch arrangements permit the cargo to be grouped so that individual hatches contain cargo to be discharged at a single port of call, thus avoiding the problem of having to move all of the deck cargo in order to discharge cargo stowed in the holds.

The lengths of individual holds is a function of the clearances between the individual containers, together with the between hatch gaps. The gaps between the individual containers is dependent upon the type of container stowage equipment to be utilised on the vessel, as this will determine the clearances required for the hold cell-guides etc. It is considered reasonable to assume that a gap of 76 mm is sufficient between adjacent container bays as this is the clearance used by most

below-deck cell guides systems, and also permits a forty foot container to be accommodated in the two twenty foot cells Figure 5.17 (MacGregor-Conver, 1997).

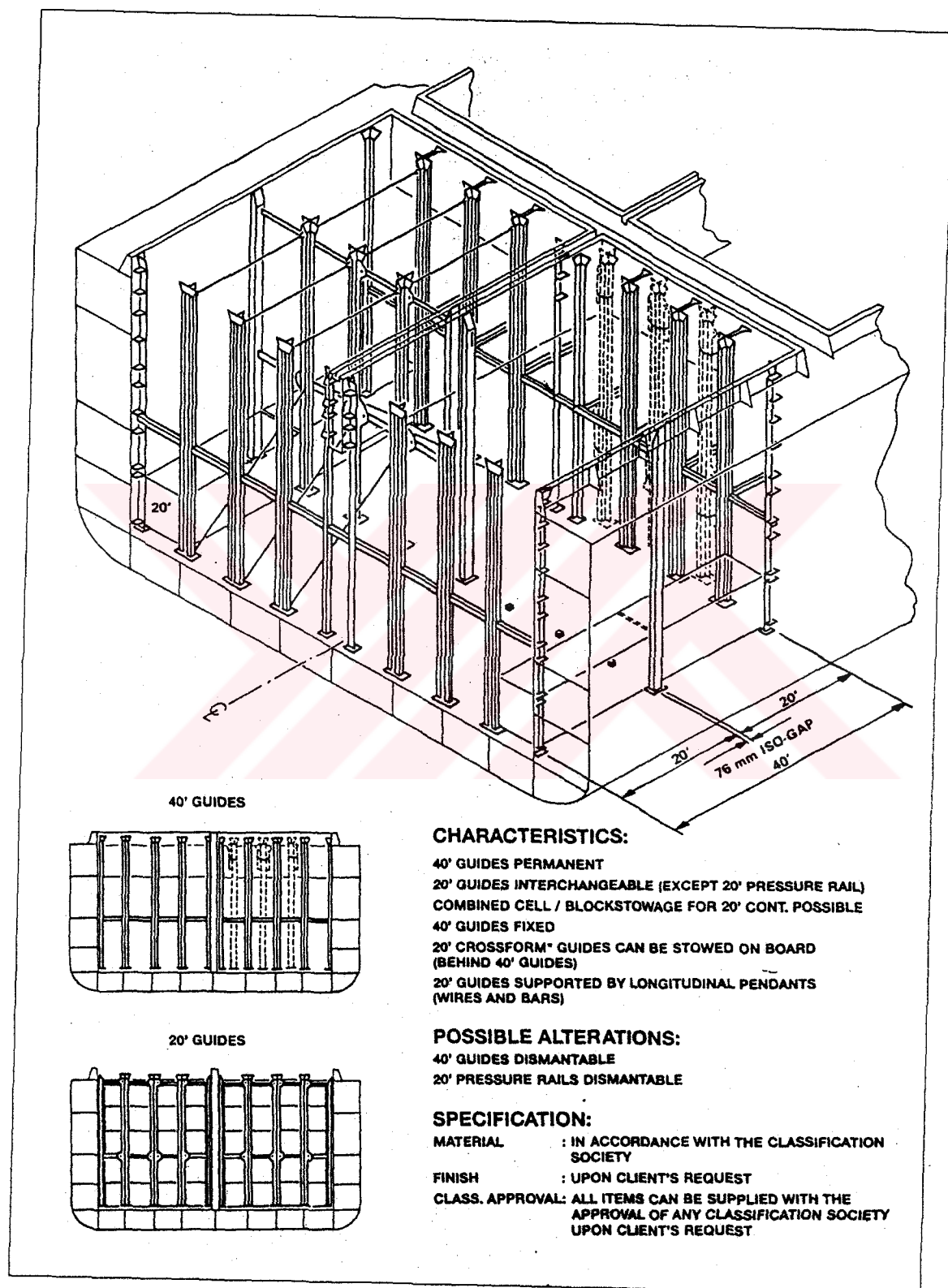


Figure 5. 16 Typical cell-guide system

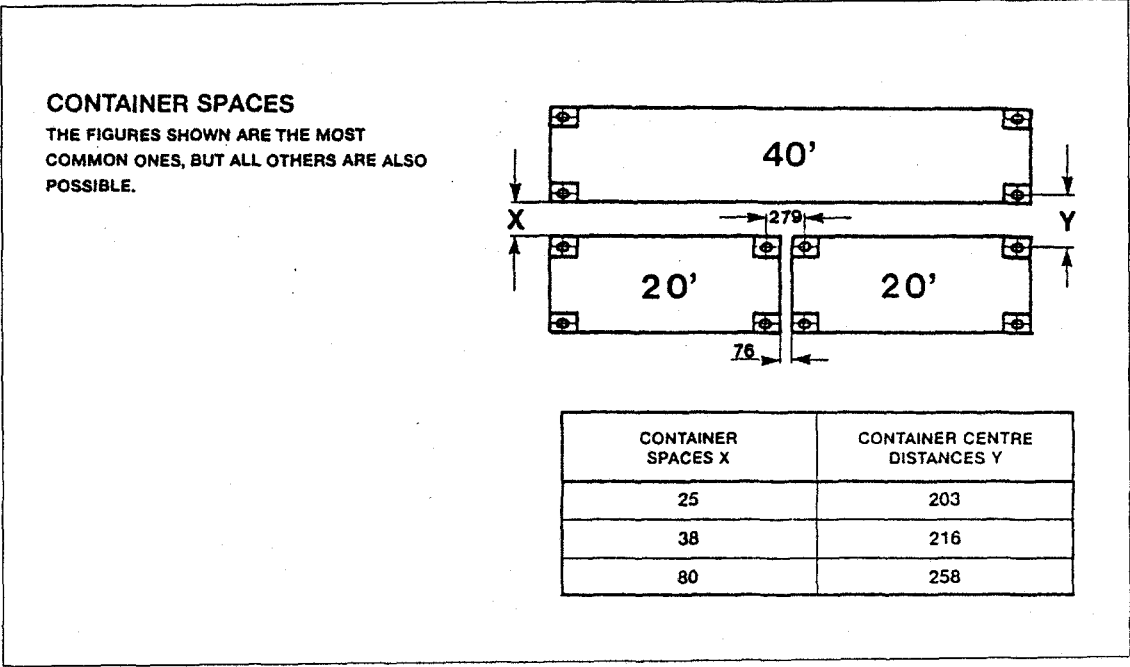


Figure 5.17 Container spaces

It is generally assumed that the frame spacing for a container ship is 700 mm. between forehead collision bulkhead and aft peak bulkhead. The frame spacing is assumed to be 600 mm. from forehead to forward and from aft peak bulkhead to backward (Göke, 1998).

5.4.5.5 General Arrangement of Weather deck

The upper deck layout design is based on the logic involved in ensuring that both strength and operational aspects are satisfied by Classification society rules and standards (Welsh, 1989).

As the hatch opening is normally from 80 to 85 percent of the ship's breadth, longitudinal strength is provided by the fitting of wing tanks and the resulting fore and aft bulkheads. A very important factor in the design procedure for the container ship is the clearance between the containers and the cell guides, as well as the gaps between the guides themselves. It is clear from these that there is a considerable loss of volume between the containers; the container ship has less volume available for cargo than a general cargo ship of the same dimensions. To regain a part of this loss in cargo volume, it is normal practice to carry containers on top of the hatches. The container height on top of the hatches is controlled by the stability characteristics of the ship. The deck containers are lashed in position to prevent movement while at sea and also to prevent racking of the containers. The design of a hatchway must satisfy both the strength and operational aspects.

Experience indicated that it was reasonable to assume the strip of deck outboard of the hatches would provide adequate longitudinal and torsional strength if it was made equal to 8-10% of the overall breadth of the ship, or  $B/15$  of the ship.  $B/15$  or  $0.08-0.1B$  is reserved at each side of the container storage area as wing tanks. In order to calculate the width of the hatches, it is necessary to consider the required gaps between the individual containers in the transverse direction and associated structural clearances as shown in Figures 5.18 and 5.19.

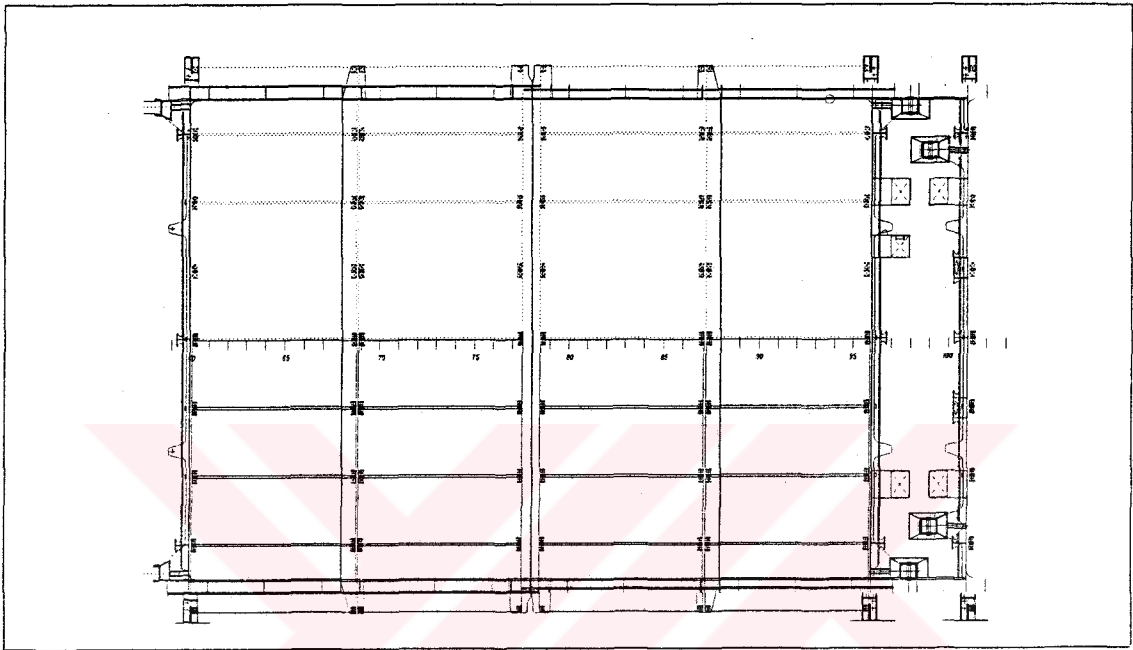


Figure 5. 18 Hatch

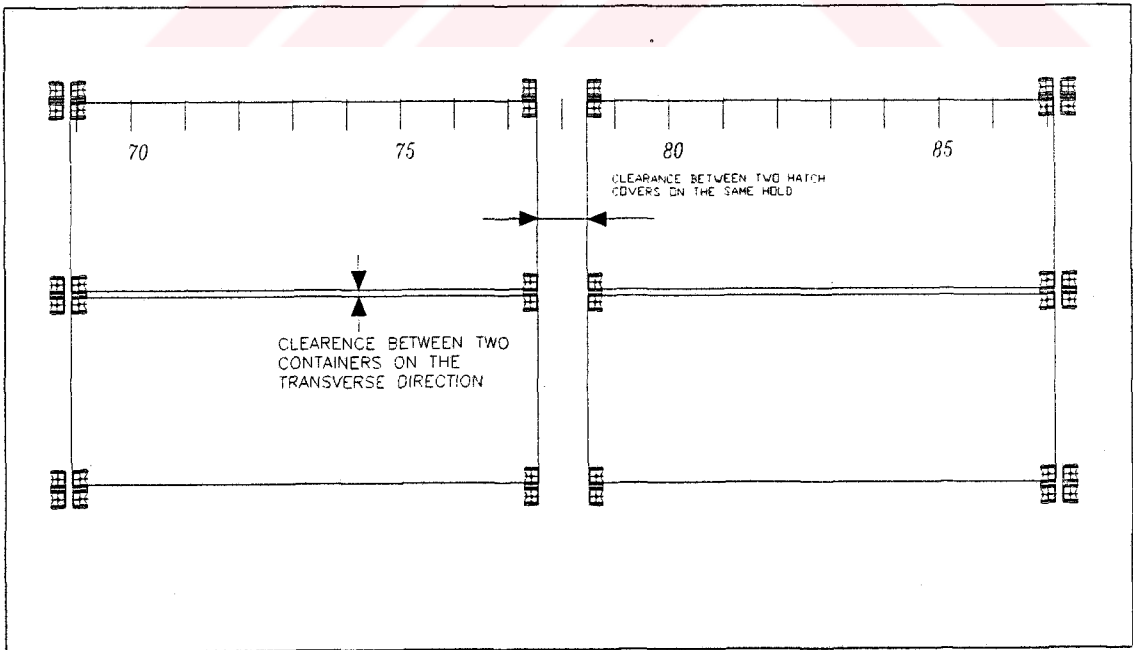


Figure 5. 19 Hatch

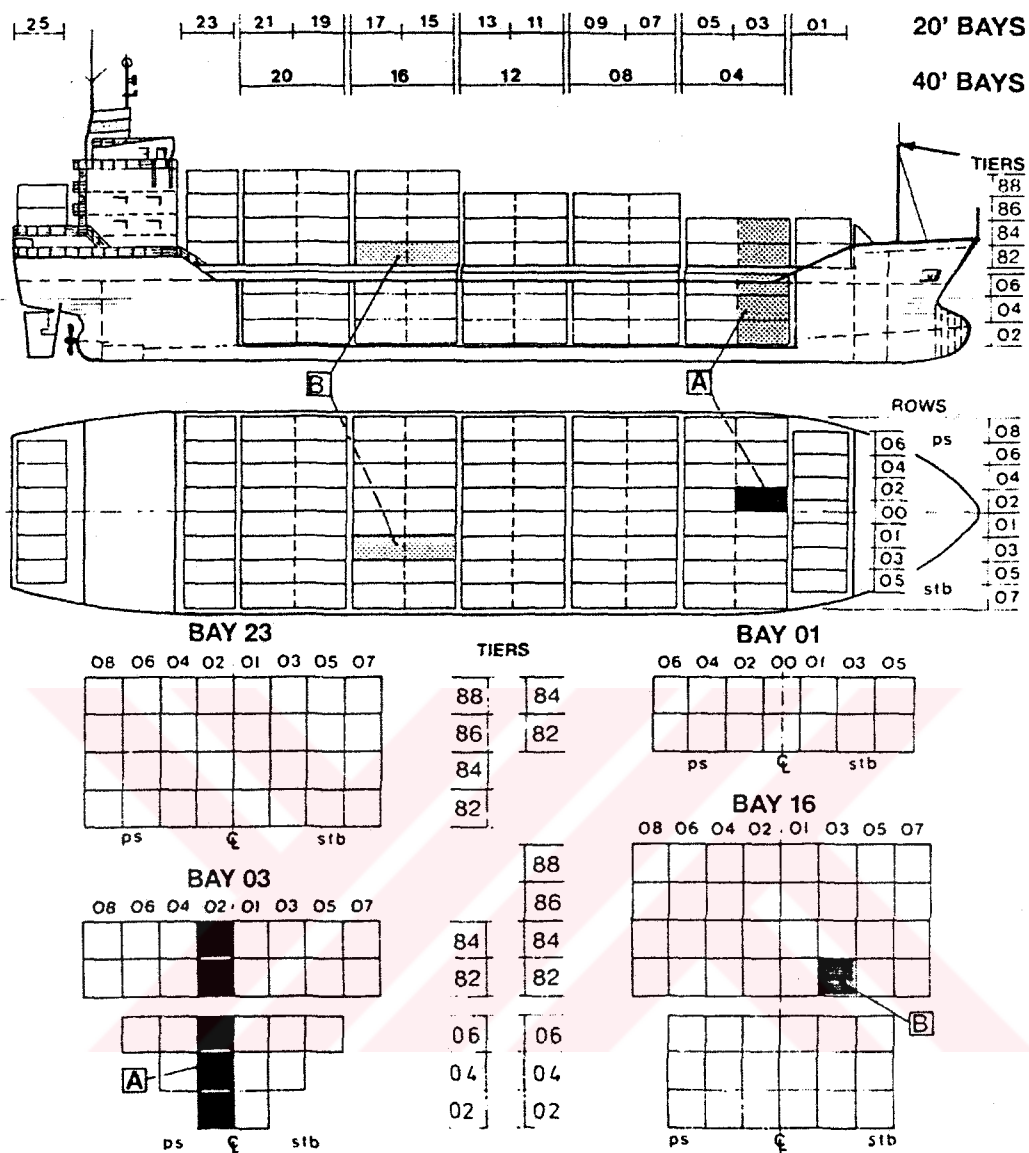
Welsh (1989) assumed that the clearances between the hatch end structure and the end of the adjacent container are according to the container securing system. It is therefore assumed that a gap of 100 mm at the ends of each hatch is sufficient to ensure adequate clearance for the installation of the selected cell guide system. It should be noted that the length of a hatch is a multiple of the frame spacing of the ship. The length of a hatch will be calculated according to container dimensions and the gap at the end of the hatches, but the real length will be bigger than this when the frame spacing is taken into account. The required gap between adjacent hatches is dependent on some factors such as cargo handling equipment (cranes) and frame spacing of the ship. There are two types of cargo hatch-hold arrangement. In the first type the distance between containers and single transverse bulkhead, between two adjacent holds, is two frame spacing if there is no cargo handling equipment or three to four frame spacing if a pedestal type deck crane is fitted. In the second type of arrangement the transverse bulkheads of each hold is aligned with the hatch and two to four frame spacing is left as a deep tank between the holds. The second approach have been adopted in the current work prototype.

Before going into an investigation, it is suitable to mention container slot identification. This method is taken from MacGregor-Conver catalogue (1997). The basis of the convention is that each slot on the ship, above or below the upper deck, is identified by a unique six digit code. The digit code indicates in which bay, tier or row, the slot to be found. The first two digits represent the bay which contains the slot, with the slot being numbered starting at the forward end of the ship and working aft, as indicated in Figure 5.20. The second pair of digits indicates the relevant tier, as shown in Figure 5.20, with the final pair of digits representing the row number.

The arrangement of above deck containers are effected by two factors. The first one is stability which is mentioned before, the second one is the visibility. The height of container tiers on the deck must allow visibility of  $1.25 L_{BP}$  distance forward of fore perpendicular from the navigation bridge.

Hence the number of container tiers on deck can be varied, i.e. increased of the aft end.

The second consideration in the above deck arrangement is the cargo handling equipment. Cranes can be positioned above the transverse bulkheads either at the centre line or on the deck side to facilitate stowage of foldable hatch covers. No cargo handling gear was allocated in the current prototype.



#### DEFINITION

- |                                    |                            |
|------------------------------------|----------------------------|
| <b>[A]</b> POSITION OF A STACK     | ➡ BAY NO. ROW NO.          |
| <b>[B]</b> POSITION OF A CONTAINER | ➡ BAY NO. ROW NO. TIER NO. |
| <b>[C]</b> EXAMPLE 20' STACK       | ➡ 03 - 02                  |
| <b>[D]</b> EXAMPLE 40' CONTAINER   | ➡ 16 - 03 - 82             |

Figure 5.20 Code of container position

Other arrangement alternatives can be summarised as:

- location on the containers in the transverse direction can be adopted between the hatch covers.
- 2 bays of bays of containers can be stowed above the engine room by locating superstructure as aft as possible.
- in the larger ship sizes some containers can be located behind the superstructure then even a single face fall life boat cannot be used in this configuration, and two life boats, one in each side, are required.

#### **5.4.6 Crew Number Determination**

As a first step of the accommodation design, the crew number must be determined. The factors which determine the size of a crew for a ship of Turkish registry are:

1. owners' requirements.
2. STCW, IMO regulations are accepted by Turkish Authority.

The size and standard of accommodation must be reduced in proportion to the size of the vessel, but generally every effort is made to provide a high degree of habitability for vessels which are at sea for long periods.

Navigation, bridge and engine personnel are determined according to regulations, however service personnel, and specialised personnel such as radio officer, electrician is determined with the preference of the owner.

#### **5.4.7 Accommodation**

Superstructure of a container ship is mainly consists of accommodation and service areas. The engine casing and funnel is usually situated behind the super structure to minimise noise, vibration and fire risk.

##### **5.4.7.1 Crew Space**

When the crew number, mentioned above, is calculated by a rule based program, the next step is determination of minimum required crew space. The general knowledge to generate the crew space consists of regulations, standards, and

heuristics which originates from social hierarchy, human factors, ergonomics and safety factors as given in this section.

In this work several standards and regulation books have been used. These are IMO regulations, ILO standards for crew accommodation and German Authority regulations. Beyond these British regulations and US Coast Guard have been mentioned. Among other things, the regulations include the following:

1. location: crew quarters must not be located further forward in a vessel than a vertical plane located at 5% of vessel's length abaft the forward side of the stem at the designed summer load waterline, nor on a deck below the deepest load line (Taggart, 1980). Space for crew is to be positively separated from other type of spaces, such as cargo and machinery spaces.
2. construction: accommodation spaces must be constructed of fireproof materials and in accordance with specified methods of fire protection. This is given by IMO rules in more detailed stairways and corridors, as a means of escape which have special fire protection requirements. The crew spaces are to be insulated from heat, cold, and condensation, and bulkheads are to be odour proof in appropriate locations (Taggart, 1980).
3. sleeping accommodations: the details are given in Appendix B.
4. washroom and toilets: the rules and regulations for these are given in Appendix B.
5. messrooms: these are to be located near galley or suitably equipped serving pantry as is practicable, of a size to seat the number of persons scheduled to eat at one time.
6. hospital: this space can not be used for any other purposes. A hospital is not required when all rooms are for single occupancy, but an emergency treatment room must be provided.
7. miscellaneous: the crew must have a laundry, clothes drying facilities or drying spaces, and recreation area. Heating of crew spaces is to maintain a minimum temperature of 21°C under normal conditions. Crew is to be provided with berthlights. Except in such areas as are considered insect-free, air ports, and doors are to be fitted with insect screen, except air-conditioned spaces.
8. officers' quarters: it should be separated from crew accommodation.

Modern practice is to provide single occupancy rooms with adjoining toilet space for all unlicensed crew. Toilet spaces are usually between rooms and shared by two individuals, except higher ratings such as chief cooks, chief electrician. Area provided is about 10 m<sup>2</sup> clear inside the room, not including the toilet space (Taggart, 1980).

Licensed officer rooms are for private occupancy with a private toilet space and are generally sized at about 14 m<sup>2</sup>. Quarters for captain and chief engineer usually include an office or a day room in addition to a stateroom and may be as large as 37 m<sup>2</sup> or more (Taggart, 1980). Officers should be quartered together by departments to provide access to their place of duty.

Habitability is a measure of adequacy work and off-duty facilities in terms of satisfying those personnel needs, which are dependent on the physical environment. The objective of habitability design is to provide physical working conditions and personnel support facilities which best contribute to total ship effectiveness for both work and off-duty environments (Taggart, 1980). The goal is to provide an environment that allows personnel to perform tasks effectively and insures health and safety. This requires adequate control of temperature, humidity, illumination, noise and vibration. With respect to the off-duty environment, the goal is to satisfy off-duty related needs to allow personnel to return to work physically and mentally prepared to carry out their tasks.

#### **5.4.7.2 Galley, Messrooms, Dining Rooms, and Store Spaces**

The galley, messrooms, and stores must be arranged to provide all the hotel facilities for the crew of the ship.

**Galley:** The galley should contain all the necessary mechanical and fabricated items to prepare food efficiently. It must be as compact as possible, light, airy, and easy to maintain and clean. Attention should be paid to details of this work to insure ease of cleaning and to eliminate pockets and inaccessible spaces. Sufficient drains and gutterways must be provided.

The ventilation and lighting of galley and pantries should be of a very high standard. MARPOL 1973 regulations forbid putting the garbage into the sea, because of this chutes with watertight covers and non-return valves at the shell are generally fitted at the sides of the ship in the galley area.

**Messrooms and Dining Rooms:** Enough space according to crew number should be provided. Separate messrooms are provided for the officers and crew which should

enable seating for all personnel an allowance of 1.1 m<sup>2</sup> per seat is a general guidance. Messrooms should be located near to galley or pantry, if they are served from pantry.

**Store Spaces:** The arrangement for stores on a cargo ship is more or less standard, the consumables being divided into dry and refrigerated stores. In addition there are the usual bosoms' stores, deck gear lockers, and paint and lamp rooms. The food stores are usually directly beneath the galley. Small linen lockers are adjacent to quarters.

The capacity of each store room is dependent on the number of days at sea and crew to be accommodated.

#### **5.4.7.3 Relation Between Spaces and Access**

The functional relationship between spaces is the essence of general arrangement and the prime factor to be considered in arriving at the final selected location of spaces. Aside from the functional relationship among spaces, the general characteristics of a space must be considered in selecting its location on the ship; these include volume, shape, weight of contents, and requirement for HVAC, noise and vibration limits, and fire safety requirements.

A ship would be lacking in comfort and efficiency if access problems have not been solved. Access means the possibility of getting in a simple and direct way from their quarters to their respective places of duty, from their quarters or place of duty to their messing and recreation spaces, from any of the foregoing spaces to the boat embarkation deck or to a weather deck, for emergency, and to any space on the ship for inspection, maintenance, or emergency action.

The crew should be able to travel from their quarters to their place of duty at any time and under any weather conditions without traversing officers' areas. The stewards' crew must traverse officers' areas. Officers and crew should be quartered so as to provide simple access to their place of duty. It is desirable to provide fore and aft crew access below decks when possible. Messes are used three times a day by all crew. Therefore the best access is assumed when these spaces are located near the intersection of the main crew stair with the main crew fore and aft passageway.

#### **5.4.7.4 Accommodation Design Knowledge**

Watson (1998) gave some useful knowledge about accommodation design in his book. The summary of the knowledge is given as follows:

- the end of deck houses should be positioned above bulkheads,
- the sides of deck houses should be positioned to line up with deck longitudinals,
- possibly the next important decision concerns the location of the galley, because of the influence this has on the location of associated spaces which either to serve the galley or to be served from the galley,
- the first decision needed is the number of decks. The basis of this decision is commonly the provision of adequate visibility from the wheelhouse over the forecastle and over the maximum obstruction caused by containers,
- the next step is to put all the required areas for all the accommodation spaces in a sheet. Then the cabins can be allocated to individual decks:
  - to keep related rooms on the same deck. If not possible move on as near a deck as can be arranged,
  - to arrange accommodation of senior personnel on higher decks than junior personnel,
  - to keep the total area on each decks either the same or less than the area of deck below it.
- having decided on the approximate area which is to be given to each type of room together with the furniture to be fitted, the designer should then make a sample cabin layout to try to establish the shape of room which will accommodate furniture in a pleasing way and fit conveniently into the potential deckhouse space,
- an interesting decision concerns the orientation and position of beds. In theory this can lie either fore and aft or athwartships and the former can be positioned either adjoining a passage or at the deckhouse or ship's side. Beds adjacent to passages may suffer some disturbance from those using the passages, whilst the need for the cabin width to be such as will accommodate both the length of a bed and width of a door may not be acceptable,
- beds adjacent to the ship's side are in the most dangerous possible location in the event of a collision,
- the beds at the deckhouse side have a more serious disadvantage of drips from windows or sidelights,

- if the room is to have a couch as well as a bed this may be the next item to be positioned, with a preference for it being located at right angles to the bed,
- thereafter it is a matter of finding wall space for the wardrobe, chest of drawers, desk etc.,
- toilets are important factors. These have to be considered both in relation to the cabin arrangement and general in relation to the toilet of the next cabin,
- one extreme of cabin shape is a long narrow shape in which most of the furniture is arranged along one of the long sides with not much more than an access passage,
- another extreme is a square room. This shape of room will have too much centre space relative to the wall space needed for doors, windows and furniture which is usually wall supported,
- the space allocations are officer cabins  $13.4 \text{ m}^2$ , crew cabins  $9.6 \text{ m}^2$ , excluding toilets which are about  $2.6 \text{ m}^2$ ,
- it is usual to limit the width of passages not much more than a meter,
- if possible stairs serving several decks should be run in a vertical stair well,
- for structural reasons stairs should be arranged (if possible) within the line of openings,
- stairs running fore and aft are probably safer to use in a seaway with rolling motions being more severe than pitching,
- lavatory space as discharge pipes should not be run through refrigerated spaces or over switchboards,
- if possible toilets are grouped side by side in a block so that the discharges can all be directed into one pipe running,
- galleys should be arranged with their length athwartships to minimise the danger of the ship's rolling motion throwing someone working on it onto a hot plate,
- the shelves of storerooms should generally be about 600 mm wide and access passages should be of about the same width,

- the best use is to have shelves on both sides of the passage making 1.8 m a suitable clear width or for larger store rooms, multiples of this width with a connecting passage.

## **5.5 Application of Facility Layout Tools**

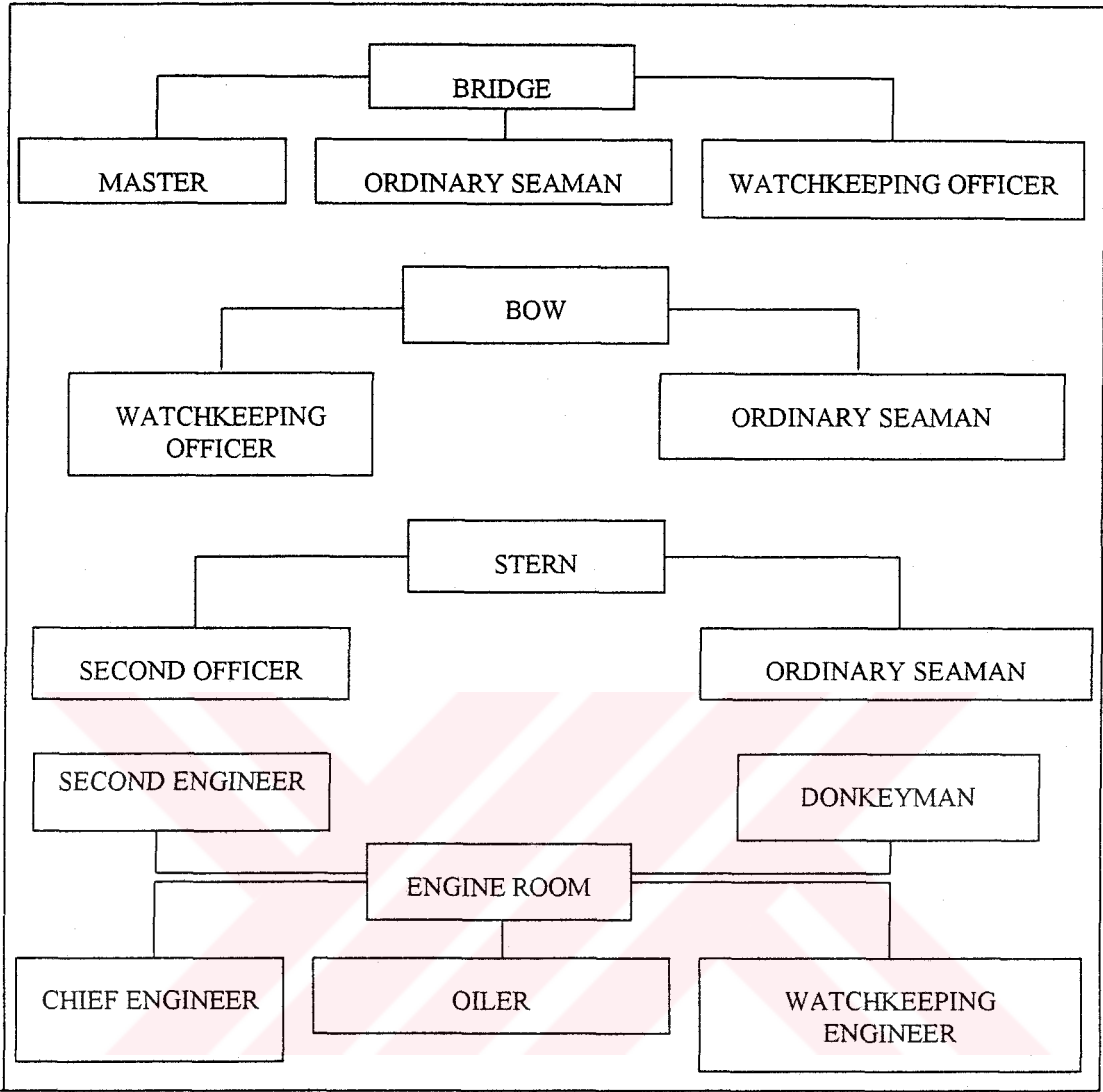
Layout design topic has been explained in Chapter 4. Implementation of the tools for the current work will be explained with all the assumptions in this chapter.

### **5.5.1 Human and Material Flow of a Container Ship**

From-to charts are used to show human and material flow in a facility. There is a number of flow in a ship: the flow of the people for the purposes of:

- going into work places, such as bridge and engine room,
- going into emergency stations such as bridge, engine room, CO<sub>2</sub> room, mooring equipment,
- evacuating the ship,
- going into daily service areas such as messes, pantries.

Additionally there is flow of materials such as fuel, seawater, fresh water, cargo. However this flow is easily managed by using a heuristic approach into ship general arrangement design. Flow of HVAC, fire smoke, noise and vibration is more important for the accommodation design. Some of the human flow diagrams are given below to illustrate most of the important movements in a container ship from Figure 5.21 to Figure 5.25. These figures have been drawn after an interview with a ship master (Dayi, 1998).



5.21 Departure from a port

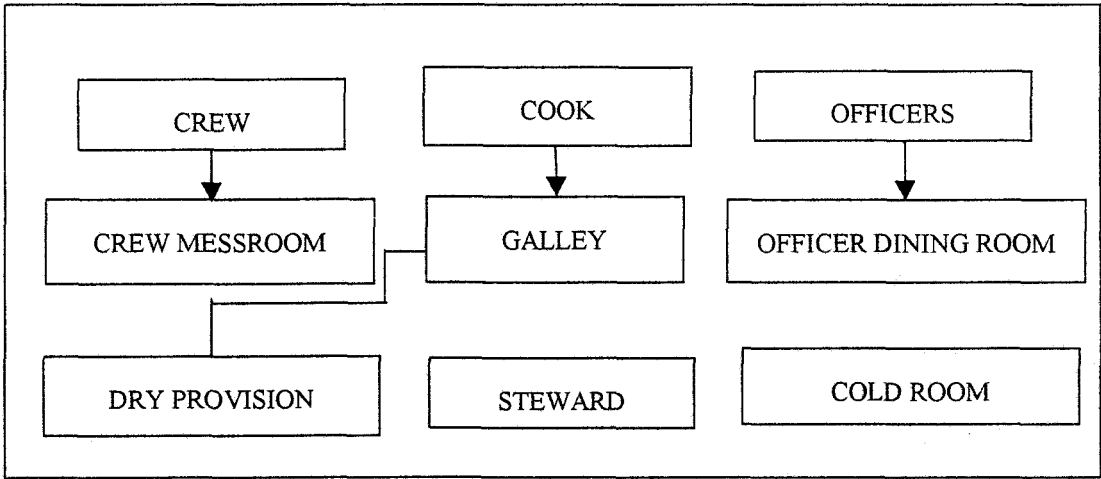


Figure 5.22 Dinner Time

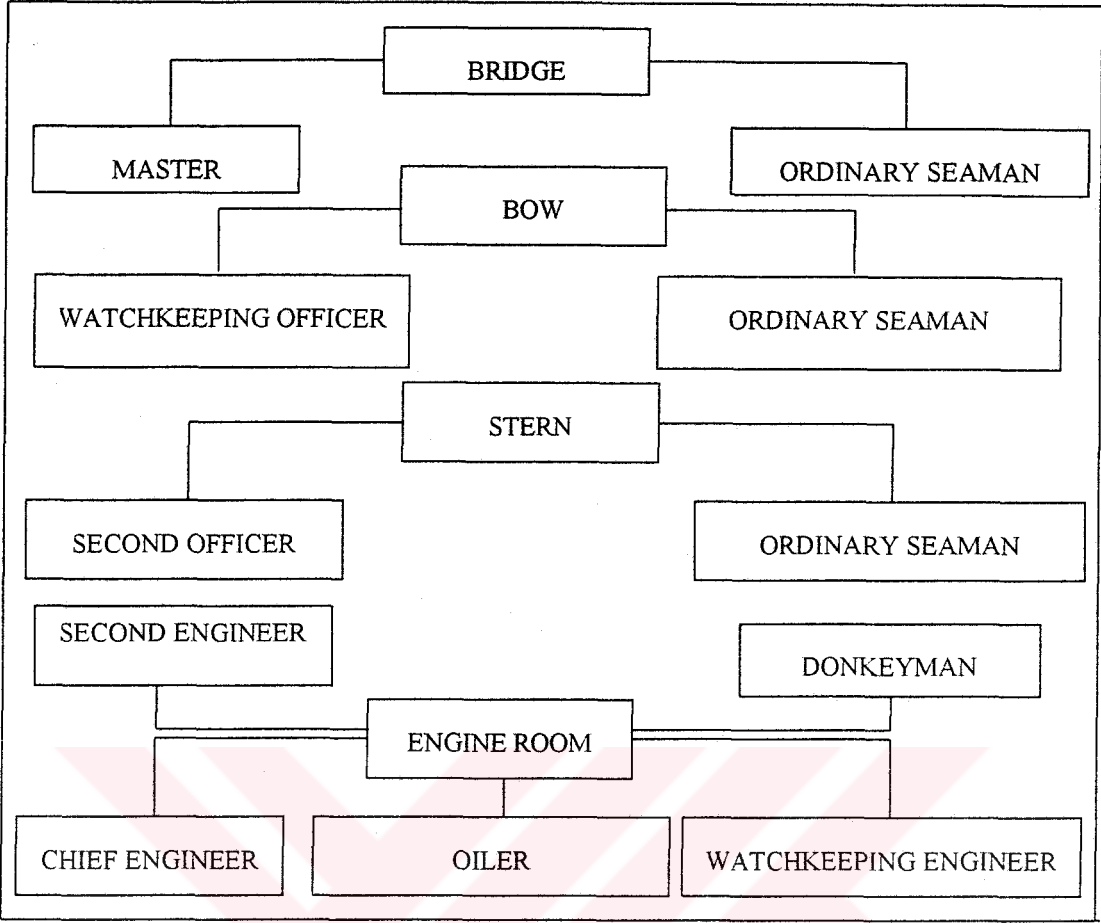


Figure 5.23 Arrival to a port

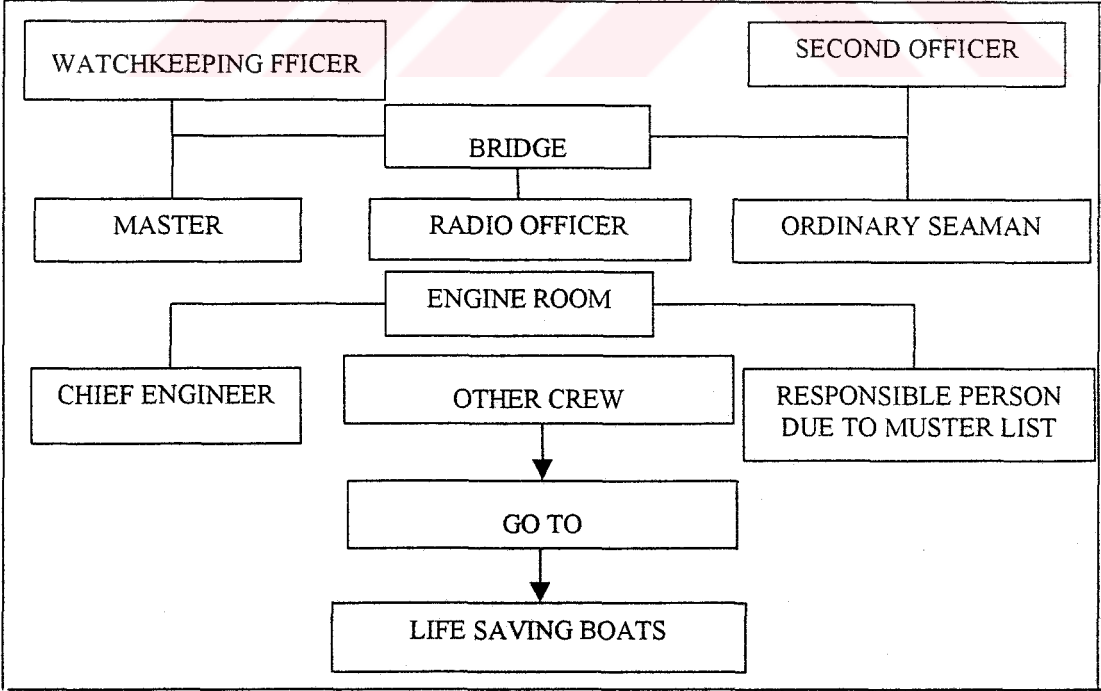


Figure 5.24 Leaving a ship in an emergency

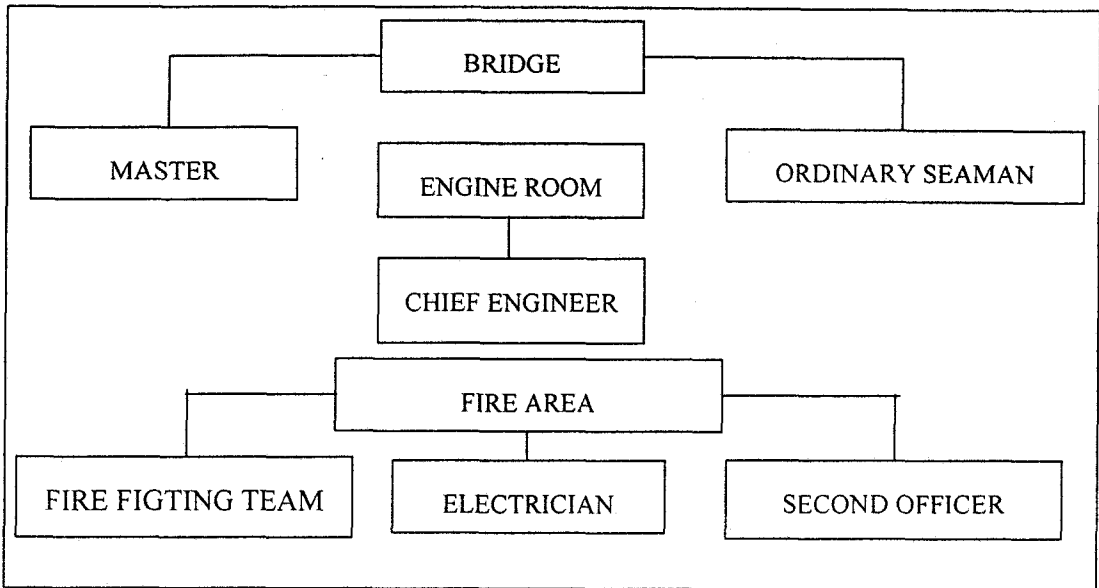


Figure 5.25

### 5.5.2 Relational Chart For A Containership

Relational charts have been explained in Chapter 4 and application for this type of chart will be given in this chapter. In the beginning of this section the number of crew and the number of other rooms are given. As a total number of the above mentioned rooms which is 39 will be located in this work. The rooms for crew members are called LIVING COMPARTMENTS and the others are called OTHER COMPARTMENTS by the author. In Table 5.3 all the compartments are given an order number to be used in the relational chart. In Table 5.4 the relational chart for the chosen crew's rooms and other rooms are given. From-to diagrams are used to prepare this chart for decision of importance rate such as A, E, I etc.

Table 5.3 Order numbers for relational chart

| No | Crew Member Duty                |
|----|---------------------------------|
| 1  | Bridge                          |
| 2  | Oceangoing-master               |
| 3  | Unlimited-chief-engineer        |
| 4  | Oceangoing-chief-officer        |
| 5  | Owner                           |
| 6  | Unlimited-second-engineer       |
| 7  | Radio-officer                   |
| 8  | Oceangoing-watchkeeping-officer |
| 9  | Unlimited-engineer-officer      |
| 10 | Pilot                           |
| 11 | Electrician                     |
| 12 | Engine-office                   |
| 13 | Officer-day-room                |
| 14 | Pantry                          |
| 15 | Crew-mess-room                  |
| 16 | Galley                          |
| 17 | Officer-dining-room             |
| 18 | Hospital                        |
| 19 | Cook                            |
| 20 | Laundry                         |
| 21 | Cold-room                       |
| 22 | Crew's bedrooms                 |
| 23 | Engine room                     |
| 24 | Radio-room                      |
| 25 | Steward                         |
| 26 | Dry-provision                   |
| 27 | Deck-office                     |

Table 5.4 Relational chart for the sample ship

|    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
|----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1  |   | A | U | E | O | U | A | I | U | E  | O  | U  | I  | O  | X  | X  | I  | U  | U  | X  | X  | X  | X  | A  | X  | U  | I  |
| 2  |   |   | E | E | A | O | E | I | O | I  | O  | U  | O  | O  | X  | X  | O  | O  | U  | X  | X  | X  | X  | E  | X  | U  | I  |
| 3  |   |   |   | U | O | E | U | U | I | U  | I  | E  | O  | O  | X  | X  | O  | O  | U  | X  | X  | X  | X  | U  | X  | U  | U  |
| 4  |   |   |   |   | E | U | I | U | U | U  | O  | O  | I  | O  | X  | X  | I  | E  | X  | X  | X  | X  | U  | I  | X  | U  | I  |
| 5  |   |   |   |   |   | U | I | U | U | U  | U  | U  | I  | O  | X  | X  | I  | U  | X  | X  | X  | X  | U  | I  | X  | U  | U  |
| 6  |   |   |   |   |   |   | U | U | I | U  | E  | I  | I  | O  | X  | X  | I  | O  | X  | X  | X  | X  | E  | U  | X  | U  | U  |
| 7  |   |   |   |   |   |   |   | U | U | U  | U  | O  | I  | O  | X  | X  | I  | O  | X  | X  | X  | X  | U  | A  | X  | U  | I  |
| 8  |   |   |   |   |   |   |   |   | U | U  | U  | O  | I  | O  | X  | X  | I  | O  | X  | X  | X  | X  | U  | U  | X  | U  | I  |
| 9  |   |   |   |   |   |   |   |   |   | U  | U  | I  | I  | O  | X  | X  | I  | O  | X  | X  | X  | X  | E  | U  | X  | U  | U  |
| 10 |   |   |   |   |   |   |   |   |   |    | U  | O  | I  | O  | X  | X  | I  | O  | X  | X  | X  | X  | O  | U  | X  | U  | U  |
| 11 |   |   |   |   |   |   |   |   |   |    |    | I  | I  | O  | X  | X  | I  | O  | X  | X  | X  | X  | I  | U  | X  | U  | U  |
| 12 |   |   |   |   |   |   |   |   |   |    |    |    | U  | O  | O  | X  | U  | U  | X  | U  | U  | U  | I  | U  | X  | U  | U  |
| 13 |   |   |   |   |   |   |   |   |   |    |    |    |    | E  | X  | E  | U  | O  | U  | U  | U  | X  | U  | U  | I  | U  | U  |
| 14 |   |   |   |   |   |   |   |   |   |    |    |    |    |    | U  | I  | I  | U  | O  | U  | U  | U  | U  | U  | I  | U  | U  |
| 15 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | E  | X  | U  | U  | U  | U  | I  | X  | U  | U  | U  | X  |
| 16 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    | E  | U  | A  | U  | A  | U  | U  | U  | I  | A  | U  |
| 17 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    | U  | U  | U  | U  | U  | U  | U  | I  | U  | U  |
| 18 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    | U  | U  | X  | U  | X  | U  | X  | U  | U  |
| 19 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    | U  | X  | I  | U  | U  | U  | U  | U  |
| 20 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    | U  | U  | U  | U  | U  | U  | X  |
| 21 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    | U  | U  | U  | U  | U  | X  |
| 22 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    | X  | U  | I  | U  | X  |
| 23 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    | U  | U  | U  | X  |
| 24 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | X  | U  | U  |
| 25 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | U  | X  |
| 26 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | X  |
| 27 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

## 5.6 Conclusion

In this chapter, heuristic container ship design knowledge has been represented. Clips Expert System shell has been used to code this knowledge. The knowledge base includes several methods for ship design methodology subareas and the method, which is used for the ES, is emphasised. The application of the knowledge base will be discussed in Chapter 6 in details.

At this stage a fundamental question should be asked: why the knowledge elicitation is a bottleneck in the building of Expert Systems. Welsh (1989) answered the question as follows:

- there is no way of deciding the best knowledge elicitation techniques for a particular problem. This can be done by trying a number of techniques,
- if the domain expert cannot or does not want to co-operate, the task of the knowledge engineer becomes more and more difficult.

The knowledge engineer can often be misled by the domain experts in their attempts to justify the methodology adapted and decisions made in the problem solving process.

Knowledge representation can be handled by using object oriented programming and production rules effectively. However the choice of objects and rules requires a high level of understanding in the Expert System as the interaction between the objects very difficult to be determined.

There are more than one method to determine attributes of the objects in ship design. In a procedural solution these methods are taken into account by use of an iterative solution which must either be hard coded or left to the user. However the choice of solution method is a powerful aspect of Expert System due to pattern matching capabilities.

Layout design techniques are useful precondition in the accommodation design by use of a relational chart. Once this has been established by a knowledge engineer, layout can be prepared using this relation chart and heuristic rules.

## **6. ACCOMMODATION LAYOUT DESIGN EXPERT SYSTEM APPLICATION**

### **6.1. Introduction**

An Expert System for container ship accommodation layout design has been developed to investigate the applicability of this approach into ship design as a pilot study. This pilot study aims to demonstrate the concepts and methods, but it does not attempt to cover all types described in previous chapters. Hence various assumptions have been made and some of which is given in Chapter 5 and 6.

Three tasks in the container ship design have been selected instead of overall design. Placement of bulkheads, hence division of the ship into compartments, determination of crew number and arrangement of the superstructure. The concept model of these tasks have been presented with a special attention to reasoning logic.

Design knowledge base developed following principles given in Chapter 2 and presented in Chapter 5, is utilised in the current work. The layout methods reviewed in Chapter 4, have been investigated and appropriate methods are integrated to ALDES as given in this chapter.

ALDES (Accommodation Layout Design Expert System) was developed mainly by production rule based Expert System approach using CLIPS Expert System shell except input/output (user) interface, numerical calculations, and visualisation purposes. A Pascal language based object oriented programming environment, DELPHI, is used for these purposes, meanwhile export to a general purpose drawing CAD program, AUTOCAD was provided.

Program modules of ES have been developed by limited knowledge base requirements. Hence knowledge base was subdivided with a logical sequence.

Case study approach has been adopted to assess the program. A container ship with 750 TEU was selected and two case studies were developed. Comparisons of these cases, and conclusions on the development of the method was presented.

## 6.2. Application of The Conceptual Model

A conceptual model for knowledge based ship design was developed in the current work. This model is given in Figure 6.1 on ship design, Expert System and knowledge based design point of views.

ALDES consists of three integrated task modules of ship design. Three sources of heuristic rules have been adopted for these modules:

- a) first heuristic source is the database of previously built ships as given in Chapter 5. The heuristic rules were derived by induction process from the database.
- b) heuristic knowledge have been acquired through knowledge extraction from the interview with the experts.
- c) heuristic rules have been derived from the rule books.

The knowledge derived from these sources have been used in three task modules.

Task Module A aims to generate a general arrangement plan by decomposing ship concept into hierarchical components such as bow, cargo holds, engine room, stern and accommodation space, shown in Figure 6.2. Facts of case, TEU and speed, and heuristic rules are utilised in a deductive reasoning to determine ship dimensions, position of bulkheads, decks, cargo area.

Task Module B is utilised to find minimum required crew number. The facts, engine power and grosstonnage are acquired from Task Module A, rules of interpretative knowledge obtained from regulations were utilised in a deductive process to obtain crew number.

Task Module C deals with a component of ship layout, accommodation block as shown in Figure 6.1. Number of decks and deck areas are obtained from Task Module A. Required rooms, room sizes and topological relations between the rooms are handled by ES from the interviews and rule books. This module outputs decks, rooms on the deck and room sizes.

## 6.2. Application of The Conceptual Model

A conceptual model for knowledge based ship design was developed in the current work. This model is given in Figure 6.1 on ship design, Expert System and knowledge based design point of views.

ALDES consists of three integrated task modules of ship design. Three sources of heuristic rules have been adopted for these modules:

- a) first heuristic source is the database of previously built ships as given in Chapter 5. The heuristic rules were derived by induction process from the database.
- b) heuristic knowledge have been acquired through knowledge extraction from the interview with the experts.
- c) heuristic rules have been derived from the rule books.

The knowledge derived from these sources have been used in three task modules.

Task Module A aims to generate a general arrangement plan by decomposing ship concept into hierarchical components such as bow, cargo holds, engine room, stern and accommodation space, shown in Figure 6.2. Facts of case, TEU and speed, and heuristic rules are utilised in a deductive reasoning to determine ship dimensions, position of bulkheads, decks, cargo area.

Task Module B is utilised to find minimum required crew number. The facts, engine power and grosstonnage are acquired from Task Module A, rules of interpretative knowledge obtained from regulations were utilised in a deductive process to obtain crew number.

Task Module C deals with a component of ship layout, accommodation block as shown in Figure 6.1. Number of decks and deck areas are obtained from Task Module A. Required rooms, room sizes and topological relations between the rooms are handled by ES from the interviews and rule books. This module outputs decks, rooms on the deck and room sizes.

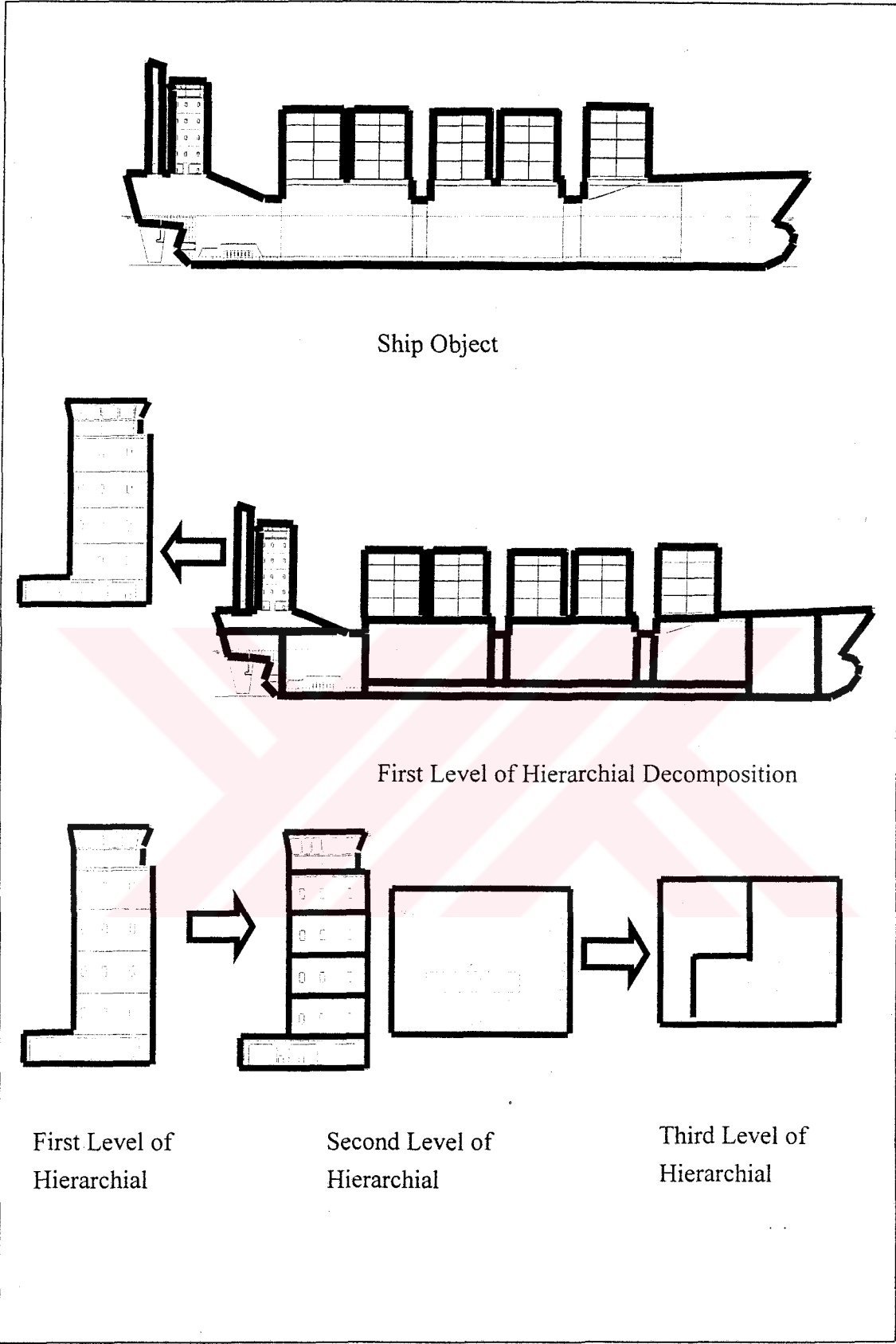


Figure 6.2 Decomposition

6.3. Programming Tools

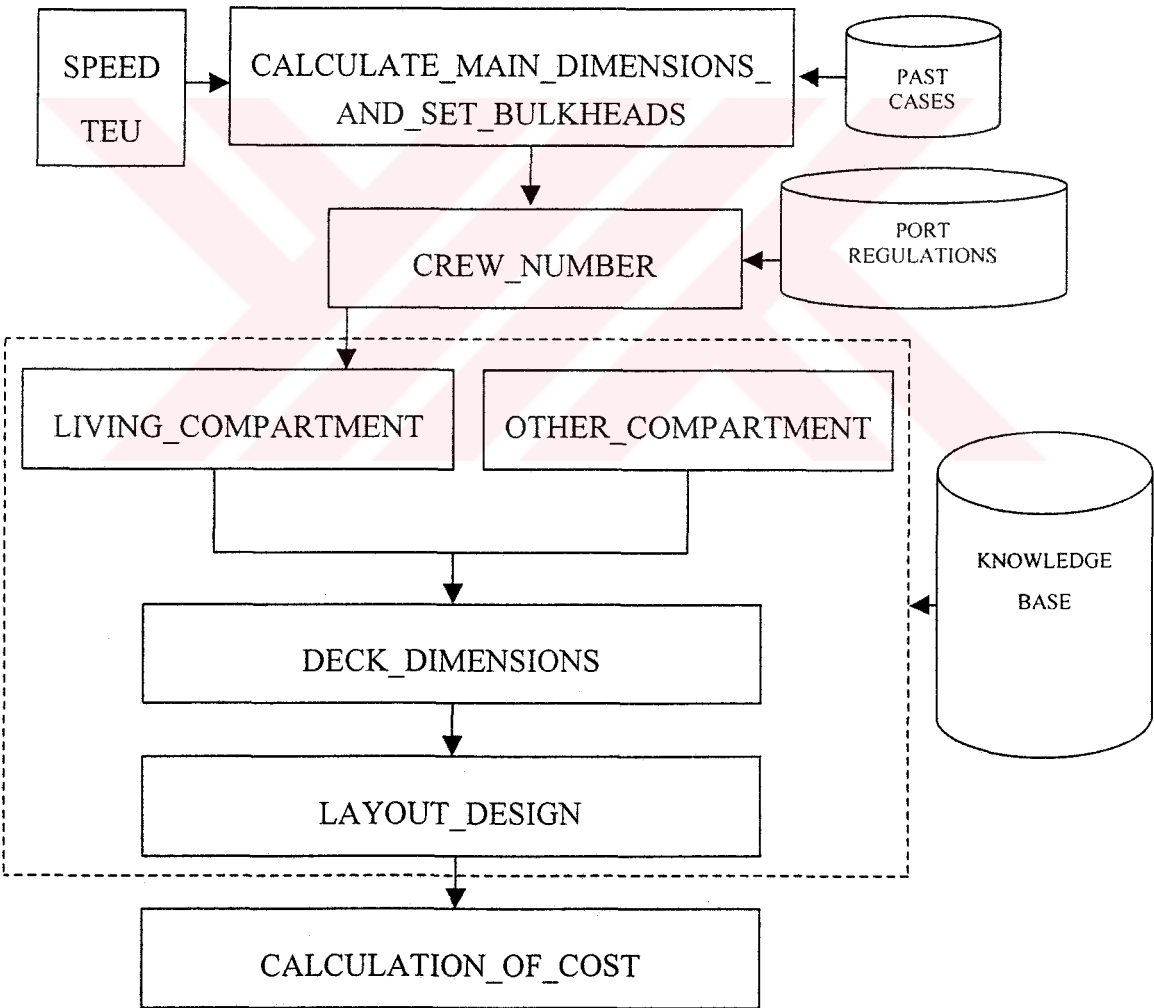
Delpi and CLIPS programming tools were used to code the model.

6.3.1 Delphi Part

User interface, layout visualisation task of ALDES has been performed by an object oriented pascal based visual windows programming language, DELPHI. Details of DELPHI is given in Appendix D.

6.3.2 CLIPS Part

In this section the program construction is given as a simple flow chart in Figure 6.3, and each module of ALDES is explained in detail.



6.3 Flow Chart of ALDES

In Figures 6.4, 6.5, and 6.6 several screen images taken from the first module of ALDES were given to demonstrate the working principle of CLIPS part of ALDES.



Figure 6.4 A CLIPS window with facts and agenda windows after a reset

In the CLIPS environment a construct should be loaded and reset command should be applied to run a program as shown in Figure 6.4. It is possible to watch fired rules and activated facts in CLIPS environment. A screen image is presented to represent “watch facility” of the shell in Figure 6.5. Another screen image is shown to explain the explanation facility of the shell. While the program is running all the steps can be observed. This facility helps to understand which fact was derived from which rule and shows the work order of the rules in the environment, as shown in Figure 6.6.

```

CLIPS 6.0
File Edit Execution Browse Window Help
CLIPS> Defining deftemplate: bulkhead
Defining deftemplate: cargo-hold
Defining defrule: start-main-dim +j
Defining defrule: get-md-input-file +j
Defining defrule: bad-input-result +j
Defining defrule: good-input-result +j
Defining defrule: read-from-file-ac +j
Defining defrule: close-input-ac +j
Defining defrule: read-again-ac +j
Defining defrule: md-assert-data +j
Defining defrule: get-output-file +j
Defining defrule: bad-output-result1 +j
Defining defrule: good-output-result1 +j
Defining defrule: close-main-dim-out1 +j
Defining defrule: bad-output-result2 +j
Defining defrule: good-output-result2 +j
Defining defrule: close-main-dim-out2 +j
Defining defrule: bad-output-result3 +j
Defining defrule: good-output-result3 +j
Redefining defrule: close-main-dim-out2 +j
Defining defrule: row-tier +j+j
Defining defrule: calculate-bay +j+j+j+j
Defining defrule: check-TEU +j+j+j+j+j+j
Defining defrule: deadweight -j-j+j
Defining defrule: grosston +j+j
Defining defrule: main-dimensions +j+j+j+j+j+j+j
Defining defrule: number-of-bulkheads +j+j
Defining defrule: cargo_hold_number +j+j+j
Defining defrule: total_bulkhead_number +j+j+j+j+j
Defining defrule: bulkhead-3 +j+j+j+j+j+j+j
Defining defrule: bulkhead-4 +j+j+j+j+j
Defining defrule: cargo-hold-distance +j+j
Defining defrule: cargo-hold-distance2 +j+j+j+j
Defining defrule: deck-height +j+j+j+j+j+j+j+j+j+j+j+j+j+j+j+j
Defining deffacts: main-dimensions
TRUE
CLIPS> (reset)
==> f-0 (initial-fact)
==> f-1 (container-standards (container-type 1A) (container-height 2438) (tolarence-in-heiht 0 -5) (container-width 2438)
==> f-2 (container-standards (container-type 1AA) (container-height 22591) (tolarence-in-heiht 0 -5) (container-width 243
==> f-3 (container-standards (container-type 1B) (container-height 2438) (tolarence-in-heiht 0 -5) (container-width 2438)
==> f-4 (container-standards (container-type 1BB) (container-height 2591) (tolarence-in-heiht 0 -5) (container-width 2438
==> f-5 (container-standards (container-type 1C) (container-height 2438) (tolarence-in-heiht 0 -5) (container-width 2438)
==> f-6 (container-standards (container-type 1CC) (container-height 2591) (tolarence-in-heiht 0 -5) (container-width 2438
==> f-7 (container-standards (container-type 1D) (container-height 2438) (tolarence-in-heiht 0 -5) (container-width 2438)
==> f-8 (Frame-spacing 0.7)
CLIPS>

```

Figure 6.5 Watch facility of CLIPS

```

CLIPS 6.0
File Edit Execution Browse Window Help
<== f-70 (total-bay 1)
==> f-72 (cargo-hold (bay-no 1) (hold-no 3))
FIRE 27 cargo-hold-distance: f-66,f-64
<== f-66 (cargo-hold (bay-no 4) (hold-no 1))
==> f-73 (bulkhead (type cargo) (no 1 2) (distance-from-aft 46.31077528750829))
FIRE 28 cargo-hold-distance2: f-73,f-69,f-8
==> f-74 (bulkhead (type cargo) (no 2 1) (distance-from-aft 47.71077528750828))
==> f-75 (bulkhead (type cargo) (no 2 2) (distance-from-aft 72.89477528750828))
<== f-69 (cargo-hold (bay-no 4) (hold-no 2))
FIRE 29 cargo-hold-distance2: f-75,f-72,f-8
==> f-76 (bulkhead (type cargo) (no 3 1) (distance-from-aft 74.29477528750829))
==> f-77 (bulkhead (type cargo) (no 3 2) (distance-from-aft 80.65277528750829))
<== f-72 (cargo-hold (bay-no 1) (hold-no 3))
FIRE 30 deck-height: f-6,f-55,f-28,f-27,f-30,f-53,f-51,f-49,f-46,f-48,f-64,f-62,f-58,f-77,f-8
<== f-55 (deck-height-1)
Upper deck con. Tier 4

last container point from AP= 94.2447752875083
last container point from FP= 6.462924451765673
deck length = 5.123522888750536
deck length = 11.25752288875054
first deck point from AP = 13.66877528750829
first deck point from FP= 87.03892445176569
==> f-78 (deck-fore-point-from-AP 13.66877528750829)
==> f-79 (deck-fore-point-from-FP 87.03892445176569)
==> f-80 (deck-length 11.25752288875054)
last deck point from AP= 2.41125239875775
last deck point from FP= 106.2275831844034
==> f-81 (deck-aft-point-from-AP 2.41125239875775)
==> f-82 (deck-aft-point-from-FP 106.2275831844034)
Deck height from main deck = 18.4996160597376
Number of Acc. Decks = 7
==> f-83 (deck-height-from-main-deck 18.4996160597376)
==> f-84 (deck-number 7)
==> f-85 (first-deck-width 17.38)
==> f-86 (other-deck-width 15.38)
FIRE 31 read-from-file-ac: f-36
<== f-36 (read-file-md)
==> f-87 (data EOF)
FIRE 32 close-input-ac: f-87
<== f-87 (data EOF)
FIRE 33 close-main-dim-out1: f-18
<== f-18 (close-main-dim-output1)
FIRE 34 close-main-dim-out2: f-16
<== f-16 (close-main-dim-output3)
CLIPS>

```

Figure 6.6 Explanation facility of CLIPS

### 6.3.2.1 1<sup>st</sup> Module: Calculate Main Dimensions And Set Bulkheads

- Aim: feeder type of container ships are taken into account for this work as it was explained in Chapter 5. The first part of ALDES was designed to calculate ship main dimensions, set the double bottom and main transverse bulkheads, locate the accommodation part, calculate the size and the number of the accommodation decks.
- Inputs: the basic requirements to start the design are speed and capacity of the ship. These are given as facts into CLIPS. The values of the facts in the test case are 16 knot for the velocity, 750 TEU for the capacity. These facts can be represented in CLIPS as follows:

(speed 16)

(TEU 750)

- Assumptions and method: some empirical formulas derived from existing ship database and ISO 668 1CC and 1AA types of container sizes used to calculate main dimensions of the ship. The first step of the program is to choose the number of rows and tiers from a database which was derived from a graph given in Figure 5.1 in Chapter 5. It is assumed that 40% of the total container capacity will be carried under deck for the current test case. So the number of bays was calculated using this assumption. When the under deck capacity is divided by multiplication of number of rows and tiers to calculate the number of bays, the result number will most probably be a real number. Therefore the result is always rounded to the next integer. Container arrangements is not included in the first stage of the current work, then the number of rows in the forward part of hold is likely to be less than the middle part. So it is required to add several number of bays according to the case to achieve the aimed capacity. This addition process makes the model be on the safe side for the result number of total containers. A new capacity is calculated by multiplication of rows, bays and tiers. The calculated new capacity is accepted if it is in the range of 2% less or more than the aimed capacity. Otherwise the number of bays will be enlarged and the procedure will be repeated to find a proper solution.

From the number of rows breadth of the ship and from the number of tiers depth of the ship and from the number of bays length of the ship were calculated by using knowledge base given in Chapter 5.

10% of  $L_{BP}$  is added to length to take into account for the forward part which is not efficiently used to place any container rows. Every cargo hold is assumed to have four 20 feet or two 40 feet containers. The distance between the two block of 20' containers is taken as 50 cm and the distance between bulkheads and the bays is also taken 15 cm. There is five frame spacing distance between two cargo holds. After calculating  $L_{BP}$ ,  $L_{WL}$  is calculated by assuming that  $L_{BP}$  is 96% of  $L_{WL}$ .

It is quite difficult to determine the ship power in the preliminary design stage. In the current work, this calculation was done by using empirical formulas derived from a database (Alasehir, 2000). The length of the engine was calculated according to engine power, so the length of engine room which is a parameter in ship length was calculated from this length.

The other difficulty in the preliminary stage is to set the main transverse bulkheads. The minimum number of bulkheads is known from TL rules. The bulkhead arrangement is done roughly by some basic assumptions and all the distance of bulkheads is given from the aft perpendicular.

DB height is taken from TL rules as given in Chapter 5 to calculate the depth of the ship. Block coefficient  $C_B$  is calculated from Alexandre formula.

The accommodation deck size was calculated with using arrangement of the container bays on the deck. The same number of container bays with the under deck were located on top of the hatches. Two extra bays of containers were added between fore part of the most forward hatch and the collision bulkhead. Another two bays were located between engine room forward bulkhead and accommodation part. On the most aft part of the ship from the most aft of the ship 7.3 metres was reserved to locate a liferaft. A distance from this point to the most aft container border was accepted as accommodation deck length. If the length of superstructure is smaller than 7 metres than one of the upper deck container bay length was added to the deck length. This causes to decrease the upper deck bay number. The height of the bridge from the main deck can be calculated from visualisation rule. This rule suggests a specific distance to be seen from the bridge. This distance is defined as the distance between the bridge and fore perpendicular plus 1.25 times  $L_{BP}$ .

Running the program with these assumptions some of the derived facts is given in Figure 6.7.

```
CLIPS 6.0
File Edit Execution Browse Window Help
f-8      (frame-spacing 0.7)
f-17     (close-main-din-output2)
f-24     (TEU 750)
f-25     (underdeck 300)
f-26     (overdeck 450)
f-27     (tier 5)
f-28     (row 7)
f-30     (bay 9)
f-33     (underdeck 315)
f-34     (TEU 765)
f-38     (U 16)
f-39     (DWT 10637.3)
f-40     (displacement 15675.12528)
f-41     (BHP 8325.738606999104)
f-42     (LER 14.12677528750828)
f-44     (GT 16187.75855)
f-46     (LBP 100.707699739274)
f-47     (LWL 104.9038538950771)
f-48     (LOA 108.6388355831611)
f-49     (B 23.38)
f-50     (double-bottom 1.4821)
f-51     (D 12.1521)
f-52     (CB 8.6332952964533556)
f-53     (T 9.845662454333057)
f-56     (bulkhead-number-TL 4)
f-58     (cargo-hold-number 3)
f-61     (total-bulkhead-number 6)
f-62     (bulkhead (type collision) (no 1) (distance-from-aft 95.46250704452012))
f-63     (bulkhead (type stern-tube) (no 6) (distance-from-aft 7.8))
f-64     (bulkhead (type engine-room) (no 5) (distance-from-aft 21.12677528750828))
f-73     (bulkhead (type cargo) (no 1 2) (distance-from-aft 46.31877528750829))
f-74     (bulkhead (type cargo) (no 2 1) (distance-from-aft 47.71077528750828))
f-75     (bulkhead (type cargo) (no 2 2) (distance-from-aft 72.89477528750828))
f-76     (bulkhead (type cargo) (no 3 1) (distance-from-aft 74.29477528750829))
f-77     (bulkhead (type cargo) (no 3 2) (distance-from-aft 80.65277528750829))
f-78     (deck-fore-point-from-AP 13.66877528750829)
f-79     (deck-fore-point-from-FP 87.03892445176569)
f-80     (deck-length 11.25752288875054)
f-81     (deck-aft-point-from-AP 2.41125239875775)
f-82     (deck-aft-point-from-FP 106.2275831844034)
f-83     (deck-height-from-main-deck 18.4996160597376)
f-84     (deck-number 7)
f-85     (first-deck-width 17.38)
f-86     (other-deck-width 15.38)
For a total of 51 facts.
CLIPS>
```

Figure 6.7 Derived facts of the first module

- Output: after all these assumptions and calculations the values for a containership which has 750 TEU as the capacity and 16 knot as the speed are given in Table 6.1. The values are given according to the calculation order. There are two different output file of the module. The first one includes the output values given in Table 6.1 and the second one keeps the values of power and grosstonnage is used in the next module.

Table 6.1 Calculated Main Dimensions

| Dimension                   | Value |
|-----------------------------|-------|
| Capacity (required) (TEU)   | 750   |
| Speed (V knot)              | 16    |
| Over deck capacity          | 450   |
| Under deck capacity         | 315   |
| Rows ( $N_v$ )              | 9     |
| Tiers ( $N_z$ )             | 5     |
| Bays ( $N_x$ )              | 11    |
| Capacity (calculated) (TEU) | 760   |
| Over deck capacity          | 375   |
| Under deck capacity         | 385   |
| DWT                         | 10637 |
| $\Delta$                    | 15675 |
| BHP                         | 8325  |
| LE                          | 6.09  |
| $L_{ER}$                    | 14    |
| GT                          | 16187 |
| $L_{BP}$                    | 100.7 |
| $L_{WL}$                    | 105   |
| $L_{OA}$                    | 108.6 |
| B                           | 23    |
| DB                          | 1.4   |
| D                           | 12    |
| $C_B$                       | 0.633 |
| T                           | 9.8   |

### 6.3.2.2 2<sup>nd</sup> Module: Crew Number Calculation

- Aim: this module was developed to determine required crew number. The knowledge base of the program was prepared by extracting the knowledge from (Gemiadamlari Mevzuati, 1997). The knowledge base does not include the rules for ships with unmanned engine room.
- Input: crew number module seeks grosstonnage (GRT), engine power and range of the designed ship to fire the rules. GRT and power were calculated in the first module and the range was given as a fact with a value of LONG-DISTANCE in the current module.



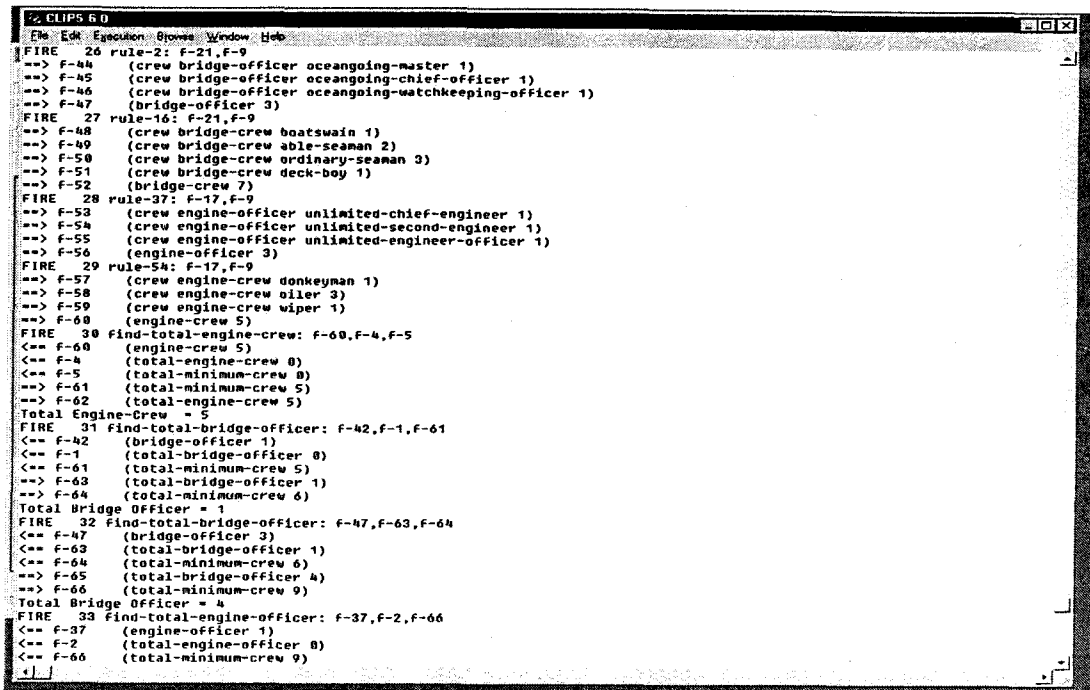


Figure 6.8 A screen image from second module to show explanation facility

- Output: an example of the program output is given in Table 6.2. All these information from the table was written to an output file to be read in the next modules to define the compartments.

Table 6.2 Crew members according to the program

| No | Crew Member Duty                | Number |
|----|---------------------------------|--------|
| 1  | Oceangoing-master               | 1      |
| 2  | Oceangoing-chief-officer        | 1      |
| 3  | Oceangoing-watchkeeping-officer | 1      |
| 4  | Unlimited-chief-engineer        | 1      |
| 5  | Unlimited-second-engineer       | 1      |
| 6  | Unlimited-engineer-officer      | 1      |
| 7  | Radio-officer                   | 1      |
| 8  | Electrician                     | 1      |
| 9  | Boatswain                       | 1      |
| 10 | Able-seaman                     | 2      |
| 11 | Ordinary-seaman                 | 3      |
| 12 | Deckboy                         | 1      |
| 13 | Cook                            | 1      |
| 14 | Steward                         | 1      |
| 15 | Donkyman                        | 1      |
| 16 | Oiler                           | 3      |
| 17 | Wiper                           | 1      |
|    | Total Number of Crew            | 22     |

### 6.3.2.3 3<sup>rd</sup> Module: Living Compartment

- Aim: this module of the program is written to define and modify name, statue and area of crew members' room are modified or defined. Statue attribute is decided by the author to differ compartments from each other as living compartment in which somebody lives in and other compartments which is used for crew any other reason than living.
- Input: living compartment module of ALDES uses the output file of 2<sup>nd</sup> module. Every crew is read from an output file of the module 2 and asserted as a new fact into the CLIPS environment. The format of new fact to define a crew as following:

“(crew ?name ?number ?crew-duty ?number1)”

“?name” is a variable and can be bridge-officer, engine-officer, bridge-crew, engine-crew or crew.

“?number” represents a number which shows quantity of the same crew, such as 3 ordinary-seaman. This variable is used to differ the same crew from each other by coding them with a number. If there are 3 ordinary-seaman they have an order number as 1, 2 and 3.

“?crew-duty” can be oceangoing-master etc.

“?number1” is a number. This number is found if the “?number” is bigger than 1. If this is true “?number1” is equal to (?number-1).

- Assumptions and method: For the compartments a template has been prepared. The template includes as much as possible attributers to define a compartment. During the running of ALDES, these attributes for a compartment are modified or calculated according to the rules which have been acquired from rule books, experts and heuristic knowledge. From this program onward, all the other program modules are run together and they use each others activated facts. It means all these modules are loaded into CLIPS environment together and run at the same time hence they share the memory and the facts base. Compartment member duty which can be watch or dayman is a common sense knowledge. However, there are three different sources as given in Appendix B for accommodation part design of a ship. In this work only ILO rules are coded. ILO rules define three different compartment area for crew, if there is more than minimum required number of crew, if there is only minimum required number of

crew and if two crew are sleeping in the same room. For the sake of simplicity only areas for minimum required number of crew are taken into account. ILO defined different areas for crew and officers as well.

The template is given here as it is written in CLIPS Shell.

```

(deftemplate compartment
(slot compartment-name)
(slot compartment-status (allowed-values LIVING OTHER))
(slot compartment-member (default NONE)
(allowed-values crew bridge-crew engine-crew NONE bridge-officer engine-officer))
(multislot compartment-member-duty (default NONE)
(allowed-values watch dayman NONE))
(slot compartment-quantity (default NONE))
(slot compartment-floor (default NONE))
(slot compartment-floor-area (default NONE))
(multislot compartment-connection (default NONE))
(multislot compartment-direction (default NONE)
(allowed-values BOW STERN
PORT STARBOARD NONE))
(slot compartment-door-face (default NONE))
(slot compartment-window-face (default NONE))
(slot compartment-shape (default REC)
(allowed-values L REC NONE))
(multislot compartment-coordinate1 (default 0 0 0))
(multislot compartment-coordinate2 (default 0 0 0))
(multislot compartment-coordinate3 (default 0 0 0))
(multislot compartment-coordinate4 (default 0 0 0))
(multislot compartment-coordinate5 (default 0 0 0))
(multislot compartment-coordinate6 (default 0 0 0))
)

```

Some of the instance values are given to the attributes and some default values are defined in the beginning not to allow asserting incorrect values during each run. The setting the living compartments and modifying the area attribute in this template are given in Figure 6.9 and 6.10.

```
CLIPS 6.0
File Edit Execution Browse Window Help
FIRE 1 get-living-compartments: f-79
==> f-92 (compartment (compartment-name oceangoing-chief-officer) (compartment-status LIVING) (compartment-member bridge-off
CLIPS> (run 1)
FIRE 1 get-living-compartments: f-78
==> f-93 (compartment (compartment-name oceangoing-watchkeeping-officer) (compartment-status LIVING) (compartment-member bri
CLIPS> (run 1)
FIRE 1 get-living-compartments: f-77
==> f-94 (compartment (compartment-name boatswain) (compartment-status LIVING) (compartment-member bridge-crew) (compartment
CLIPS> (run 1)
FIRE 1 get-living-compartments: f-76
==> f-95 (compartment (compartment-name able-seaman) (compartment-status LIVING) (compartment-member bridge-crew) (compartment
CLIPS> (run 1)
FIRE 1 get-living-compartments: f-74
==> f-96 (compartment (compartment-name able-seaman) (compartment-status LIVING) (compartment-member bridge-crew) (compartment
CLIPS> (run 1)
FIRE 1 get-living-compartments: f-73
==> f-97 (compartment (compartment-name ordinary-seaman) (compartment-status LIVING) (compartment-member bridge-crew) (compa
CLIPS> (run 1)
FIRE 1 get-living-compartments: f-71
==> f-98 (compartment (compartment-name ordinary-seaman) (compartment-status LIVING) (compartment-member bridge-crew) (compa
CLIPS> (run 1)
FIRE 1 get-living-compartments: f-69
==> f-99 (compartment (compartment-name ordinary-seaman) (compartment-status LIVING) (compartment-member bridge-crew) (compa
CLIPS> (run 1)
FIRE 1 get-living-compartments: f-68
==> f-100 (compartment (compartment-name deck-boy) (compartment-status LIVING) (compartment-member bridge-crew) (compartment-
CLIPS> (run 1)
FIRE 1 get-living-compartments: f-67
==> f-101 (compartment (compartment-name unlimited-chief-engineer) (compartment-status LIVING) (compartment-member engine-off
CLIPS> (run 1)
FIRE 1 get-living-compartments: f-66
==> f-102 (compartment (compartment-name unlimited-second-engineer) (compartment-status LIVING) (compartment-member engine-of
CLIPS> (run 1)
FIRE 1 get-living-compartments: f-65
==> f-103 (compartment (compartment-name unlimited-engineer-officer) (compartment-status LIVING) (compartment-member engine-o
CLIPS> (run 1)
FIRE 1 get-living-compartments: f-64
==> f-104 (compartment (compartment-name donkeyman) (compartment-status LIVING) (compartment-member engine-crew) (compartment
CLIPS> (run 1)
FIRE 1 get-living-compartments: f-63
==> f-105 (compartment (compartment-name oiler) (compartment-status LIVING) (compartment-member engine-crew) (compartment-men
CLIPS> (run 1)
FIRE 1 get-living-compartments: f-61
==> f-106 (compartment (compartment-name oiler) (compartment-status LIVING) (compartment-member engine-crew) (compartment-men
CLIPS> (run 1)
FIRE 1 get-living-compartments: f-59
==> f-107 (compartment (compartment-name oiler) (compartment-status LIVING) (compartment-member engine-crew) (compartment-men
```

Figure 6.9 Getting the compartment template

```
CLIPS 6.0
File Edit Execution Browse Window Help
FIRE 1 floor-area-crew: f-85,f-111,f-84,f-86,f-109
<== f-84 (crew crew 1 cook 1)
<== f-86 (compartment (compartment-name cook) (compartment-status LIVING) (compartment-member crew) (compartment-member-duty
==> f-112 (compartment (compartment-name cook) (compartment-status LIVING) (compartment-member crew) (compartment-member-duty
==> f-113 (compartment (compartment-name cook) (compartment-status LIVING) (compartment-member crew) (compartment-member-duty
CLIPS> (run 1)
FIRE 1 floor-area-crew: f-85,f-111,f-83,f-88,f-109
<== f-83 (crew crew 1 steward 1)
<== f-88 (compartment (compartment-name steward) (compartment-status LIVING) (compartment-member crew) (compartment-member-d
==> f-114 (compartment (compartment-name steward) (compartment-status LIVING) (compartment-member crew) (compartment-member-d
==> f-115 (compartment (compartment-name steward) (compartment-status LIVING) (compartment-member crew) (compartment-member-d
CLIPS> (run 1)
FIRE 1 floor-area-officer: f-85,f-111,f-82,f-89,f-110
<== f-82 (crew engine-officer 1 electrician 1)
<== f-89 (compartment (compartment-name electrician) (compartment-status LIVING) (compartment-member engine-officer) (compa
==> f-116 (compartment (compartment-name electrician) (compartment-status LIVING) (compartment-member engine-officer) (compa
CLIPS> (run 1)
FIRE 1 floor-area-officer: f-85,f-111,f-81,f-90,f-110
<== f-81 (crew bridge-officer 1 radio-officer 1)
<== f-90 (compartment (compartment-name radio-officer) (compartment-status LIVING) (compartment-member bridge-officer) (comp
==> f-117 (compartment (compartment-name radio-officer) (compartment-status LIVING) (compartment-member bridge-officer) (comp
CLIPS> (run 1)
FIRE 1 floor-area-officer: f-85,f-111,f-80,f-91,f-110
<== f-80 (crew bridge-officer 1 oceangoing-master 1)
<== f-91 (compartment (compartment-name oceangoing-master) (compartment-status LIVING) (compartment-member bridge-officer) {
==> f-118 (compartment (compartment-name oceangoing-master) (compartment-status LIVING) (compartment-member bridge-officer) {
CLIPS> (run 1)
FIRE 1 floor-area-officer: f-85,f-111,f-79,f-92,f-110
<== f-79 (crew bridge-officer 1 oceangoing-chief-officer 1)
<== f-92 (compartment (compartment-name oceangoing-chief-officer) (compartment-status LIVING) (compartment-member bridge-off
==> f-119 (compartment (compartment-name oceangoing-chief-officer) (compartment-status LIVING) (compartment-member bridge-off
CLIPS> (run 1)
FIRE 1 floor-area-officer: f-85,f-111,f-78,f-93,f-110
<== f-78 (crew bridge-officer 1 oceangoing-watchkeeping-officer 1)
<== f-93 (compartment (compartment-name oceangoing-watchkeeping-officer) (compartment-status LIVING) (compartment-member bri
==> f-120 (compartment (compartment-name oceangoing-watchkeeping-officer) (compartment-status LIVING) (compartment-member bri
CLIPS> (run 1)
FIRE 1 floor-area-crew: f-85,f-111,f-77,f-94,f-109
<== f-77 (crew bridge-crew 1 boatswain 1)
<== f-94 (compartment (compartment-name boatswain) (compartment-status LIVING) (compartment-member bridge-crew) (compartment
==> f-121 (compartment (compartment-name boatswain) (compartment-status LIVING) (compartment-member bridge-crew) (compartment
==> f-122 (compartment (compartment-name boatswain) (compartment-status LIVING) (compartment-member bridge-crew) (compartment
CLIPS> (run 1)
FIRE 1 floor-area-crew: f-85,f-111,f-76,f-95,f-109
<== f-76 (crew bridge-crew 1 able-seaman 1)
<== f-95 (compartment (compartment-name able-seaman) (compartment-status LIVING) (compartment-member bridge-crew) (compartment
==> f-123 (compartment (compartment-name able-seaman) (compartment-status LIVING) (compartment-member bridge-crew) (compartment
```

Figure 6.10 Modifying floor area attribute

- Output: after running this module crew-name such as boatswain or deckboy, crew-member such as bridge officer, crew-member-duty dayman or watch and the floor area of the living compartments are modified and asserted into the working memory as facts.

#### 6.3.2.4 4<sup>th</sup> Module: The Other Compartment

- Aim: Apart from the crew rooms, there are some other compartments in an accommodation part of a ship. This module of the program was designed to define these compartments.
- Input: there is not any additional input of this program. It uses the facts from the working memory. The template of other compartments is the same as living compartment and it was defined in the 3<sup>rd</sup> module. It is used in the current module.
- Assumptions and method: these compartments are given in Table 6.3. All the compartments shown in Table 6.2 and Table 6.3 will be located in the accommodation part, according to the rules which have been acquired for accommodation design. In this program name and statue of other compartments are modified. After Living\_Compartment and Other\_Compartment programs all the names and statues of compartments are installed into the program environment as facts.

Table 6.3 The other compartments in the accommodation part.

| No | Name of the Section | Number |
|----|---------------------|--------|
| 1  | Owner               |        |
| 2  | Bridge              | 1      |
| 3  | Radio-room          | 1      |
| 4  | Deck-office         | 1      |
| 5  | Engine-office       | 1      |
| 6  | Officer-dining-room | 1      |
| 7  | Officer-day-room    | 1      |
| 8  | Pilot               | 1      |
| 9  | Pantry              | 1      |
| 10 | Galley              | 1      |
| 11 | Hospital            | 1      |
| 12 | Crew-mess-room      | 1      |
| 13 | Store-3             | 1      |
| 14 | Store-2             | 1      |
| 15 | Store-1             | 1      |
| 16 | Laundry             | 1      |
| 17 | Cold-room           | 1      |
| 18 | Dry-provision       | 1      |

- Output: there is not any specific output file after this module. The updated or modified facts are used by the following modules of ALDES.

#### **6.3.2.5 5<sup>th</sup> Module: Relational Chart:**

- Aim: Relational\_Chart program includes all relations which are given in Table 5.3. The deck number of the compartments are decided using these relations. Some of the relations are not used at all to decide the deck number of a compartment.
- Input: this module is used only to assert the relations into CLIPS environment as facts. It is designed as a different module because there are large number of relations.
- Assumptions and method: mostly A, E and I relations are used from the table of relations. If all the relations were used the program would be very difficult to code.
- Output: facts are inserted to working memory to be used by the following section.

#### **6.3.2.6 6<sup>th</sup> Module: Deck Dimensions:**

- Aim: the aim of this module is to insert the deck dimensions into the working memory .
- Input: the length and breadth of accommodation decks as well as the number of the decks were read from a file which is the output of the main dimensions program.
- Assumptions and method: the coordinates of the decks were calculated from the input by this module. The first deck over the crew accommodation was assumed as the first deck. The other decks were defined as over and under this deck. A deck template is defined in this section of the package as following:

```
“(deftemplate deck
  (slot deck-name)
  (slot deck-location
    (allowed-values on-main-deck down-main-deck up-main-deck))
  (multislot deck-coordinate-1)
  (multislot deck-coordinate-2)
  (multislot deck-coordinate-3)
  (multislot deck-coordinate-4)
  (slot deck-order-no))”
```

This template is used to define a typical accommodation deck of the ship.

- Output: after running this program, all the decks are asserted into the working memory according to the template. These are used in the next section to calculate the cost function.

#### **6.3.2.7 7<sup>th</sup> Module: Layout Design:**

- Aim: this part of the program does the main reasoning work as an ES. This includes heuristic rules as well as the other rules. All the compartments are set on an appropriate floors according to the rules and relational chart.
- Input: uses the facts from the working memory.
- Assumptions and method: radio room, galley, hospital and engine office are set from common sense rules in the beginning of the section. Then the compartments with A relations are set into proper decks according to pre-set compartments. The compartment direction attribute is modified in this section. The allowed values for this attributes are bow, stern, port and starboard as seen in the compartment template. First part of the package ends with Layout\_Design program. Most of the compartment attributes are defined after this program.
- Output: after running the module, all the compartments are placed into the accommodations decks. It means every compartment has a deck number.

#### **6.3.2.8 8<sup>th</sup> Module: Calculation Of Cost:**

- Aim: the cost factor is defined in Chapter 4. according to the formula 4-7, cost factor is calculated for different scenarios. This formula is simplified by not taking in to account the second part of the formula.
- Input: for this work a simple accommodation design has been developed by the author and this has been illustrated as a basic AUTOCAD drawing. All the coordinates and centres of the compartments have been read from this drawing.
- Assumptions and method: everybody is assumed to be in the centre of their compartments. In an emergency case they will start to move from the centre to escape points. The cost for these movements is calculated. Different scenarios are applied to see different results. This module reads the coordinates from the input file and calculates the cost function for evacuation.

6.4. Case Studies

For this work several case studies have been carried out. In this section these cases will be explained and the results will be compared.

6.4.1 Case Study I

For the first case all the compartments' doors are placed to the nearest position to a evacuation point which is the stairway of the sample ship. The drawings for this case are illustrated in Figure 6.11 a, Figure 6.11 b and Figure 6.11 c. In the case all the crew and officers are assumed to stand at the centre of their rooms. In an emergency case they start from the centre to move to their doors, then the passageway, stairway and the boat deck. For the case the cost factor is calculated by ALDES.

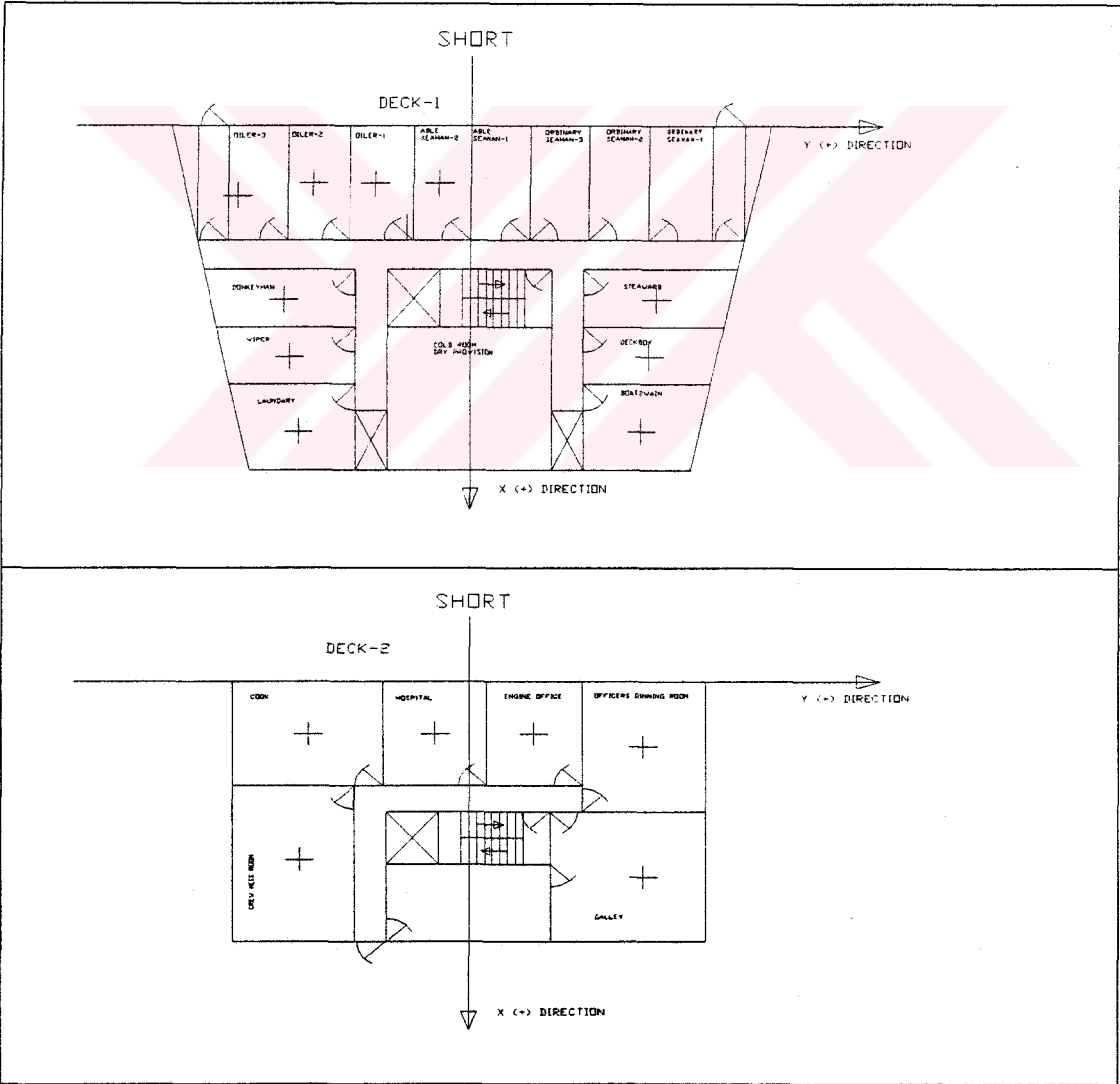


Figure 6.11 a Deck 1 and Deck 2 for the first case

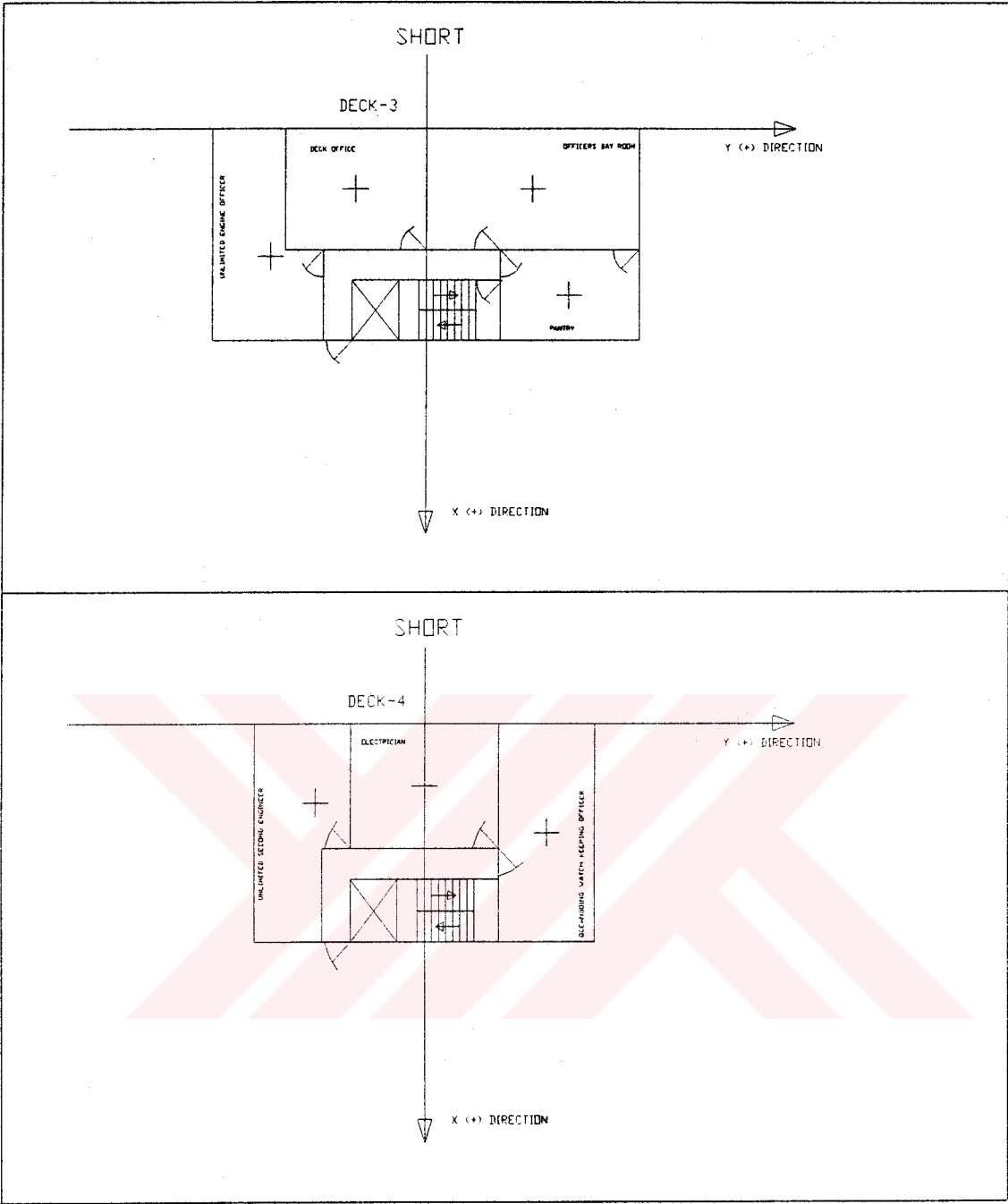


Figure 6.11 b Deck 3 and Deck 4 for the first case

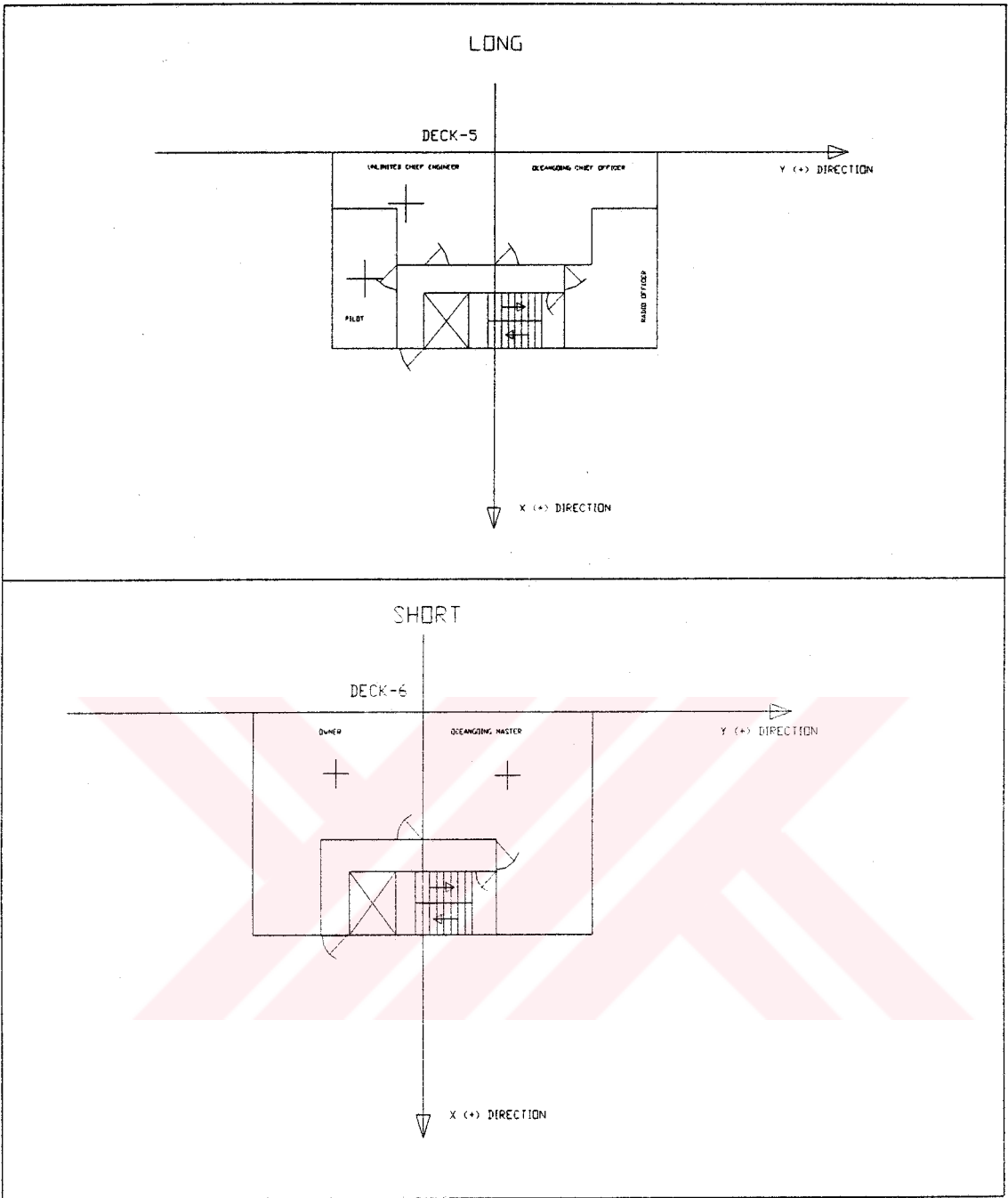


Figure 6.11 c Deck 5 and Deck 6 for the first case

The cost function was defined as the distance for every personnel from their room to a safety point which is assumed to be the third deck as boat deck. The output of this run is illustrated in Table 6.4.

Table 6.4 Case I output data

|                                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                            |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| person : cook<br>distance from centre to door : 4350.0<br>distance from door to stair : 5540.983667184014<br>distance from stair to deck : -2000<br>total distance : 11890.98366718401                                                                 | person : ordinary-seaman<br>distance from centre to door : 450.0<br>distance from door to stair : 2371.181140275876<br>distance from stair to deck : -4000<br>total distance : 6821.181140275876                                                           |
| person : ordinary-seaman<br>distance from centre to door : 400.0<br>distance from door to stair : 4220.189569201839<br>distance from stair to deck : -4000<br>total distance : 8620.189569201839                                                       | person : ordinary-seaman<br>distance from centre to door : 400.0<br>distance from door to stair : 1004.987562112089<br>distance from stair to deck : -4000<br>total distance : 5404.987562112089                                                           |
| person : able-seaman<br>distance from centre to door : 450.0<br>distance from door to stair : 2102.974084481309<br>distance from stair to deck : -4000<br>total distance : 6552.97408448131                                                            | person : oiler<br>distance from centre to door : 1550.0<br>distance from door to stair : 4854.1219597369<br>distance from stair to deck : -4000<br>total distance : 10404.1219597369                                                                       |
| person : able-seaman<br>distance from centre to door : 1550.0<br>distance from door to stair : 2926.174977679906<br>distance from stair to deck : -4000<br>total distance : 8476.174977679906                                                          | person : oiler<br>distance from centre to door : 1550.0<br>distance from door to stair : 6823.672031978091<br>distance from stair to deck : -4000<br>total distance : 12373.67203197809                                                                    |
| person : oiler<br>distance from centre to door : 2200.0<br>distance from door to stair : 8757.28268357257<br>distance from stair to deck : -4000<br>total distance : 14957.28268357257                                                                 | person : oiler<br>distance from centre to door : 2200.0<br>distance from door to stair : 8757.28268357257<br>distance from stair to deck : -4000<br>total distance : 14957.28268357257                                                                     |
| person : wiper<br>distance from centre to door : 4289.813515760329<br>distance from door to point : 2992.072860075436<br>distance from point to stair : 5423.098745182499<br>distance from stair to deck : -4000<br>total distance : 16704.98512101826 | person : donkeyman<br>distance from centre to door : 4665.029474719318<br>distance from door to point : 1073.545527679194<br>distance from point to stair : 5423.098745182499<br>distance from stair to deck : -4000<br>total distance : 15161.67374758101 |
| person : boatswain<br>distance from centre to door : 2550.0<br>distance from door to point : 4975.188438642299<br>distance from point to stair : 943.3981132056604<br>distance from stair to deck : -4000<br>total distance : 12468.58655184796        | person : oceangoing-master<br>distance from centre to door : 3033.562262423502<br>distance from door to stair : 1011.187420807834<br>distance from stair to deck : 6000<br>total distance : 10044.74968323134                                              |
| person : steward<br>distance from centre to door : 1550.0<br>distance from door to point : 1073.545527679194<br>distance from point to stair : 943.3981132056604<br>distance from stair to deck : -4000<br>total distance : 7566.943640884855          | person : deck-boy<br>distance from centre to door : 1550.0<br>distance from door to point : 2992.072860075436<br>distance from point to stair : 943.3981132056604<br>distance from stair to deck : -4000<br>total distance : 9485.470973281096             |
| person : owner<br>distance from centre to door : 6308.922253443927<br>distance from door to stair : 2926.174977679906<br>distance from stair to deck : 6000<br>total distance : 15235.09723112383                                                      | person : oceangoing-chief-officer<br>distance from centre to door : 2150.0<br>distance from door to stair : 1011.187420807834<br>distance from stair to deck : 4000<br>total distance : 7161.187420807834                                                  |
| person : unlimited-chief-engineer<br>distance from centre to door : 3150.0<br>distance from door to stair : 2926.174977679906<br>distance from stair to deck : 4000<br>total distance : 10076.17497767991                                              | person : pilot<br>distance from centre to door : 4250.0<br>distance from door to stair : 4930.770730829005<br>distance from stair to deck : 4000<br>total distance : 13180.770730829                                                                       |
| person : radio-officer<br>distance from centre to door : 2550.0<br>distance from door to stair : 626.4982043070834<br>distance from stair to deck : 4000<br>total distance : 7176.498204307083                                                         | person : oceangoing-watchkeeping-officer<br>distance from centre to door : 2550.0<br>distance from door to stair : 626.4982043070834<br>distance from stair to deck : 2000<br>total distance : 5176.498204307083                                           |
| person : unlimited-second-engineer<br>distance from centre to door : 4250.0<br>distance from door to stair : 4930.770730829005<br>distance from stair to deck : 2000<br>total distance : 11180.770730829                                               | person : electrician<br>distance from centre to door : 4850.0<br>distance from door to stair : 1001.249219725039<br>distance from stair to deck : 0<br>total distance : 5851.249219725039                                                                  |
| Total distance for all personnel is equal to 242426.8069339953                                                                                                                                                                                         |                                                                                                                                                                                                                                                            |

6.4.2 Case Study II

In this case the doors were located the longest distance from the staircase or evacuation point. The drawings are given in Figure 6.12 a, Figure 6.12 b, Figure 6.12 c.

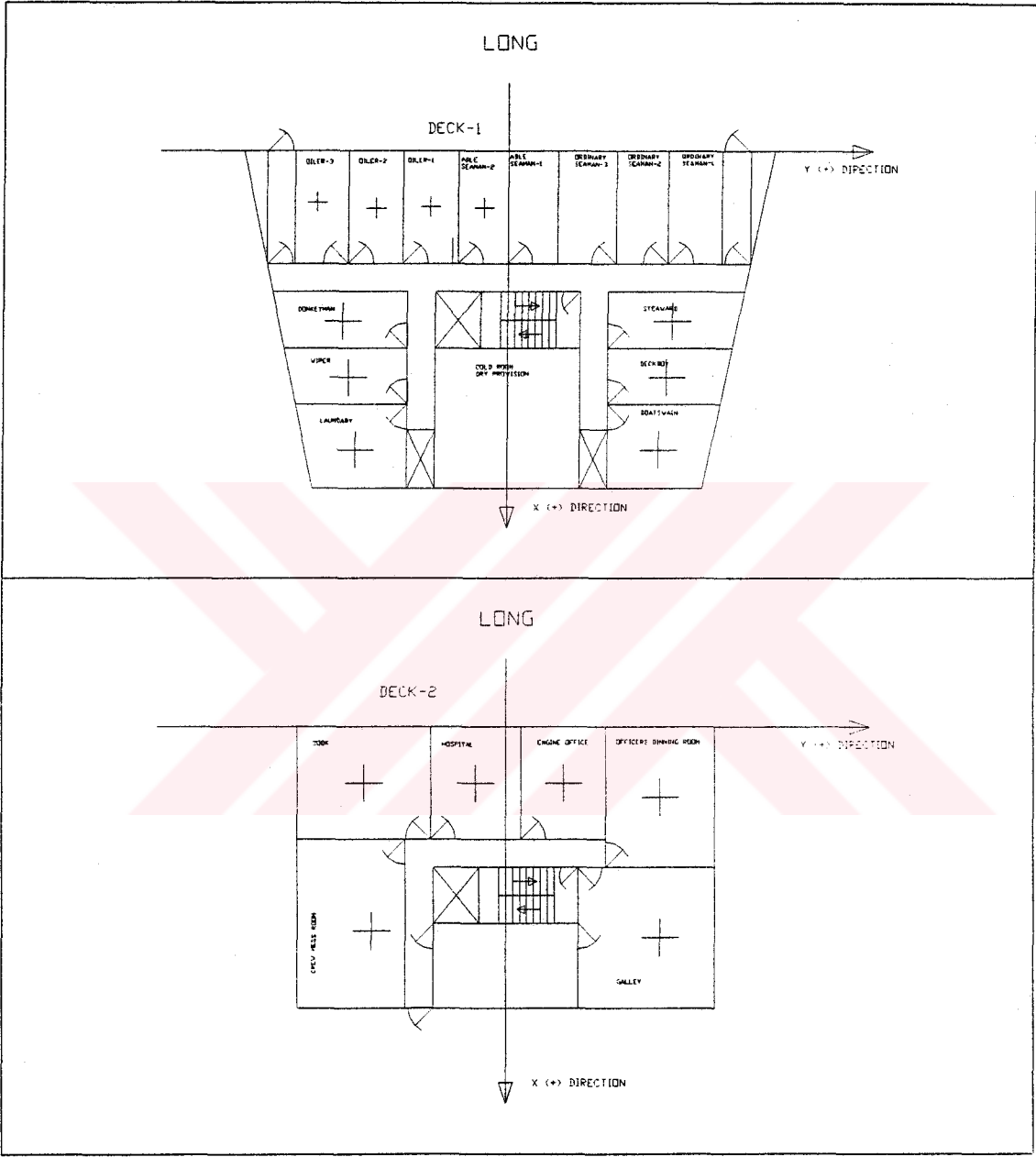


Figure 6.12 a Deck 1 and Deck 2 for the second case

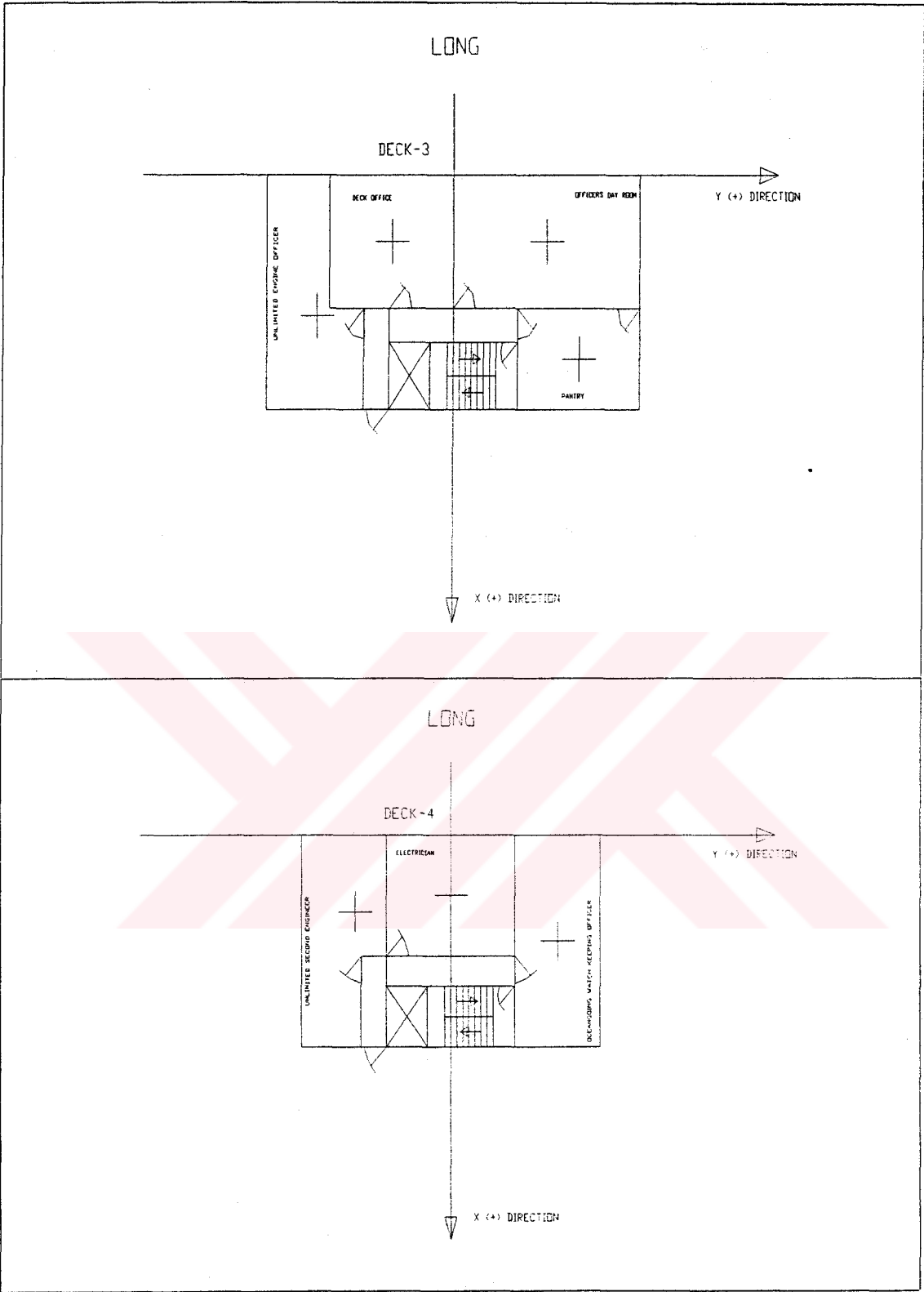


Figure 6.12 b Deck 3 and Deck 4 for the second case

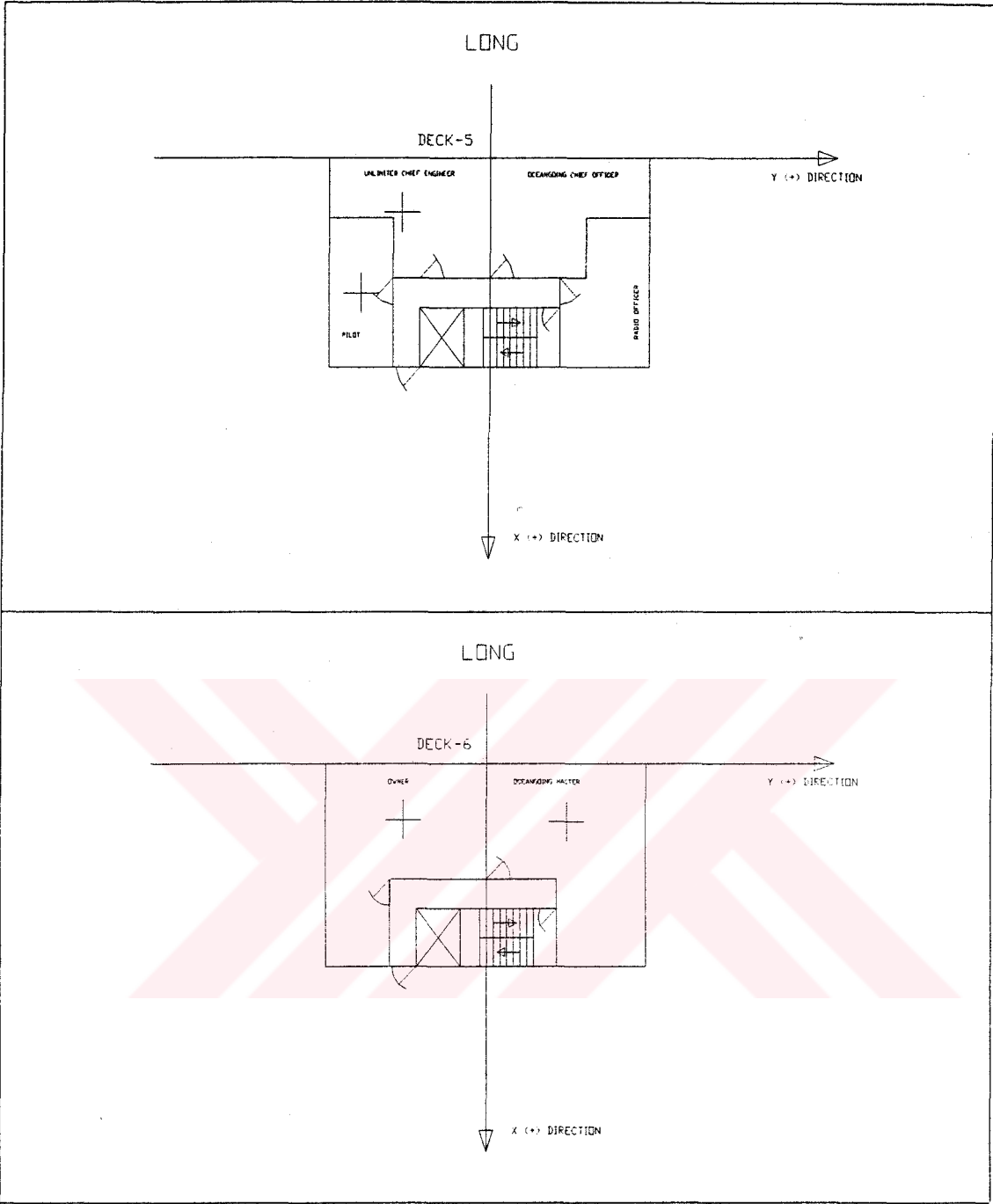


Figure 6.12 c Deck 5 and Deck 6 for the second case

The result of the calculation for the second case is given in Table 6.5

Table 6.5 Case II output data

|                                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                           |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| person : cook<br>distance from centre to door : 4350.0<br>distance from door to stair : 5540.983667184014<br>distance from stair to deck : -2000<br>total distance : 11890.98366718401                                                                 | person : ordinary-seaman<br>distance from centre to door : 400.0<br>distance from door to stair : 4220.189569201839<br>distance from stair to deck : -4000<br>total distance : 8620.189569201839                                                          |
| person : ordinary-seaman<br>distance from centre to door : 1500.0<br>distance from door to stair : 3352.610922848042<br>distance from stair to deck : -4000<br>total distance : 8852.610922848042                                                      | person : ordinary-seaman<br>distance from centre to door : 1550.0<br>distance from door to stair : 1600.781059358212<br>distance from stair to deck : -4000<br>total distance : 7150.781059358213                                                         |
| person : able-seaman<br>distance from centre to door : 450.0<br>distance from door to stair : 2102.974084481309<br>distance from stair to deck : -4000<br>total distance : 6552.97408448131                                                            | person : able-seaman<br>distance from centre to door : 450.0<br>distance from door to stair : 3977.75062063976<br>distance from stair to deck : -4000<br>total distance : 8427.750620639761                                                               |
| person : oiler<br>distance from centre to door : 450.0<br>distance from door to stair : 5934.854673873658<br>distance from stair to deck : -4000<br>total distance : 10384.85467387366                                                                 | person : oiler<br>distance from centre to door : 500.0<br>distance from door to stair : 7863.841300535<br>distance from stair to deck : -4000<br>total distance : 12363.841300535                                                                         |
| person : oiler<br>distance from centre to door : 2200.0<br>distance from door to stair : 8757.28268357257<br>distance from stair to deck : -4000<br>total distance : 14957.28268357257                                                                 | person : boatswain<br>distance from centre to door : 2550.0<br>distance from door to point : 4975.188438642299<br>distance from point to stair : 943.3981132056604<br>distance from stair to deck : -4000<br>total distance : 12468.58655184796           |
| person : deck-boy<br>distance from centre to door : 450.0<br>distance from door to point : 4080.747480548141<br>distance from point to stair : 943.3981132056604<br>distance from stair to deck : -4000<br>total distance : 9474.145593753802          | person : steward<br>distance from centre to door : 450.0<br>distance from door to point : 2110.09478460092<br>distance from point to stair : 943.3981132056604<br>distance from stair to deck : -4000<br>total distance : 7503.49289780658                |
| person : wiper<br>distance from centre to door : 4025.232912515746<br>distance from door to point : 4080.747480548141<br>distance from point to stair : 5423.098745182499<br>distance from stair to deck : -4000<br>total distance : 17529.07913824639 | person : donkeyman<br>distance from centre to door : 4422.951503238534<br>distance from door to point : 2110.09478460092<br>distance from point to stair : 5423.098745182499<br>distance from stair to deck : -4000<br>total distance : 15956.14503302195 |
| person : oceangoing-master<br>distance from centre to door : 3335.416016031583<br>distance from door to stair : 1930.025906561878<br>distance from stair to deck : 6000<br>total distance : 11265.44192259346                                          | person : owner<br>distance from centre to door : 4250.0<br>distance from door to stair : 4930.770730829005<br>distance from stair to deck : 6000<br>total distance : 15180.770730829                                                                      |
| person : oceangoing-chief-officer<br>distance from centre to door : 450.0<br>distance from door to stair : 2102.974084481309<br>distance from stair to deck : 4000<br>total distance : 6552.97408448131                                                | person : unlimited-chief-engineer<br>distance from centre to door : 3150.0<br>distance from door to stair : 2926.174977679906<br>distance from stair to deck : 4000<br>total distance : 10076.17497767991                                                 |
| person : radio-officer<br>distance from centre to door : 2550.0<br>distance from door to stair : 626.4982043070834<br>distance from stair to deck : 4000<br>total distance : 7176.498204307083                                                         | person : pilot<br>distance from centre to door : 4250.0<br>distance from door to stair : 4930.770730829005<br>distance from stair to deck : 4000<br>total distance : 13180.770730829                                                                      |
| i person : oceangoing-watchkeeping-officer<br>distance from centre to door : 2550.0<br>distance from door to stair : 626.4982043070834<br>distance from stair to deck : 2000<br>total distance : 5176.498204307083                                     | person : unlimited-second-engineer<br>distance from centre to door : 4250.0<br>distance from door to stair : 4930.770730829005<br>distance from stair to deck : 2000<br>total distance : 11180.770730829                                                  |
| person : electrician<br>distance from centre to door : 350.0<br>distance from door to stair : 4658.594208556912<br>distance from stair to deck : 0<br>total distance : 5008.594208556912                                                               | person : unlimited-engineer-officer<br>distance from centre to door : 5523.812089490373<br>distance from door to stair : 4930.770730829005<br>distance from stair to deck : 0<br>total distance : 10454.58282031938                                       |
| Total distance for all personal is equal to 247385.7944111032                                                                                                                                                                                          |                                                                                                                                                                                                                                                           |

### **6.4.3 Comparison of The Results**

These two simple case shows that if a ship is defined in the computer once, a lot of aspects can be checked such as in the current cases. It can be seen that only changing the location of the doors changes the evacuation distance. If the resistance factors, other routes etc. for evacuation are added to the cost function many more changes can be checked to achieve better results, in this case shorter evacuation distance.

From the results it can be seen that only the changing of door positions the cost function goes up for 4 metres. Some other changes can effect the evacuation cost more.

### **6.5. Conclusion**

A conceptual model of ALDES was developed in the current work and this was explained in this chapter. This conceptual model is the main contribution of this work.

The application of the model was given in detail here to explain the Expert System logic and its usage in ship design domain. This model proves that Expert System application can be developed successfully in ship design discipline.

Some screen images from the program was given to explain the logic of ALDES system.

The cost function was developed to calculate evacuation distance and applied for two cases to examine the effect of door locations to the evacuation distance. Some other factors can be added to cost function to improve the study.

## **7. CONCLUSIONS AND THE FUTURE WORK**

### **7.1 Conclusions**

Ship design is a complex problem requiring multiple criteria decision making, creativity, consisting of uncertainties and iterative steps to achieve the main goal or sub-goals. A number of approaches have been utilised to solve this problem. An Expert System approach was applied to the ship design problem in the preliminary design stage in the current work. The problem domain for the application was defined as general arrangement and the accommodation layout design.

Two problem solution approaches which are Expert Systems and plant layout were examined to develop a ship layout design assistant in the current work. Adoption of the experience, gained in both Expert Systems and plant layout, into ship design were conducted during this dissertation.

Considering the complex nature of ship design, at the beginning of the research, it was not known whether it was possible to develop a ship design assistant by use of knowledge based model. The study showed that under the certain assumptions and constrains, a computer model can be developed by CLIPS Expert System shell.

Some major and several minor contributions were observed in the current work:

- a knowledge based ship design method was developed for feeder type container ships,
- a container ship design knowledge base was extracted from domain experts. The rules and regulations from the source books were used to complete the knowledge base,
- the heuristic knowledge of experts was represented as production rules,
- historical data from existing ships were used in statistical analysis form. Induction reasoning method was adopted to improve some mathematical formulas from cases,

- human flow was utilised to develop a relational chart which shows the relations between compartments. The relational chart was used to locate the rooms in a deck. Hence topological relations were represented implicitly,
- a concept model was developed including ship design, knowledge based design and Expert System methods,
- ALDES system was developed to be used as a design assistant. Design requires achieving from the case to an end product so forward chaining search strategy was applied to the system. Deduction was used to achieve goals from cases,
- decomposition was applied to divide the ship into manageable parts to be coded on the design point of view,
- the first module was developed for general arrangement design including main dimensions, arrangement of bulkheads, location and calculation of accommodation deck size,
- crew number calculation is a simple method with requiring the usage of a large number of regulations. This process became automatic with an independent module of ALDES,
- the evacuation of ships is one of the most important safety concern in ships. This topic was chosen as cost function of the system because its importance and popularity. The ALDES system was applied to different cases to demonstrate the possibility of shortened the way to a safety point,
- deck layout topological relations have been represented by facts created from relational charts and heuristics, and distribution of required rooms over the deck were made with topological relations,
- ALDES is the first Expert System application at ITU on ship design. So there are some lacking and weak points of the system. On the other hand large amount of knowledge about container ship design was coded in the system and became modifiable easily. The large amount of knowledge was made available for others, through this work. This helps to use this knowledge freely at the communication age,
- the structure of CLIPS provides an explanation facility during its runs with presenting fired rules and activated facts. It helps the user the follow the

procedure. ALDES runtime is within a few seconds for the requirements i.e. speed, capacity,

- the application demonstrates that the container ship design problem is suitable for ES because of its size which is neither too small not worth applying nor too big to handle. The quality of expertise is high even if there are some lack of knowledge,
- DELPHI provides a user friendly interface which can hide all CLIPS environment,
- although the knowledge base can be developed further, the very large knowledge base was gathered during this work. A number of experts from the industry contributed to this work, hence this work helps to promote use of this approach in the industry,

ALDES system is a useful tool as a design assistant in the preliminary design stage, but there are still some shortcomings of the system:

- ship design is very complex team work with a large knowledge base. The relation between experts from different systems such as HVAC, electricity etc. during ship design has not been included in the current approach,
- improving an ES is another teamwork requiring one or more domain and one computer expert. So it is difficult to design a system by acting as a domain expert at the same time by coding the domain knowledge as a computer expert,
- during this work a number of domain experts have been consulted. But it has not been easy to acquire the domain knowledge even from good experts. The experts have been wishing to help, but some of them did not believe in Expert Systems. Even if they believed and willing to help their knowledge did not cover all heuristic knowledge about the container ship design,
- human brain go through the rules in an order without even thinking on them, but to make computer do the same is very difficult, in some cases impossible. Using Expert System Shells make it easier with a variety of pattern search methods which should be investigated,
- the subprograms for calculations such as stability, seakeeping, power were not available to integrate the system or some available ones were difficult or impossible to integrate,

- the heuristic knowledge was represented using production rules with a success but the volume of the knowledge made it difficult to include all the possible different methods to achieve the goal,

## **7.2 The Future Work**

1. The knowledge base was developed for the purpose of general arrangement and accommodation layout. This knowledge base should be extended to cover engine room, tanks, deck arrangement, such as mooring, anchoring, and for systems design to able to include design integration. This can be achieved with independent works on systems design Expert System development as a separate work.
2. The application of Expert Systems into ship design is not thoroughly investigated, because of its complexity. Additionally it requires a teamwork and a lot of time for the development. This can be achieved with continuation of the current work.
3. One of the main sources of heuristic information is past cases. A database of principle characteristics of past cases have been used in the current work. But no attempt to acquire general arrangement and accommodation layout have been made in the database. A case based design reasoning research would be a beneficial contribution to the state of the art.
4. The system should be tested by some other users especially from the industry to see the difficulties which users can encounter.
5. The subject has been studied in various research establishments, but it is the first application in ITU, and should be improved.
6. The cost function can be improved by adding some environmental factors such as noise and vibration, HVAC etc. to be used in the layout problem.
7. The naval ship design and passenger ship accommodation design are more complex than cargo ship accommodation design as heuristics for layout is more comprehensive. Expert Systems should be applied to these ship types to carry out the difficulties with their designs.
8. Expert Systems are useful to develop multiple solutions, several methods such as fuzzy logic should be adopted into these systems to solve the selection or optimisation problems.

9. Shape generation and location is another problem which could be integrated to improve ALDES system.
10. Visualisation tools such as VRML (Virtual Reality Modelling Language) could be integrated to the system to represent the end product.
11. In addition to design problem, design approving can be dealt with Expert Systems. Such Expert System requires the full design to be represented in computer environment in detail to check IMO regulations and classification society rules. This task should also be repeated for every ship. Expert systems can be utilised to recognise ship arrangement from the drawings.
12. This work contains only feeder type container ship. This range can be enlarged to include continental size of container ships.



## REFERENCES

- Aguilar, G.D., Yamamoto, H. and Koyama, T., 1996. Development of a hull form definition tool with a related knowledge-based advisory system, *Journal of Marine Science and Technology*, SNAJ Publication. 1, pp. 138-148.
- Aka, S., 1999. Private communication, Istanbul.
- Akagi, S. and Fujita, K., 1987. Building an expert system for the preliminary design of ships, *AI EDAM*, 1(3), pp.191-205.
- Akıntürk, A., Atlar, M. and Çalışal, S.M., 1995. Preliminary design of multi-hull fishing vessels using an expert system environment, *The Sixth International Symposium on Practical Design of Ships and Mobile Units*, September, South Korea, pp.2.1288-2.1299.
- Alasehir, E., 2000. Ship database, *Final Year Project*, Faculty of Naval Architecture & Ocean Engineering, Istanbul Technical University (in Turkish).
- Alonson, F. and Garcia, L., 1997. Virtual reality and ship design, *9<sup>th</sup> International Conference on Computer Applications in Ship Building, ICCAS'97*, October, Japan, Ed. K. Johnson and T. Koyama, pp. 121-134.
- Andrews, D., 1981. Creative ship design, *Transaction of The Royal Institution of Naval Architects*, RINA, 123, pp. 447-458.
- Andrews, D., 1985. An integrated approach to ship synthesis, *Transaction of The Royal Institution of Naval Architects*, RINA, 127, pp. 73-102.
- Appel, J. M., 1977. Plant Layout and Material Handling, pp. 1-100, Third Edition, John Wiley and Sons, New York.
- Beier, K.-P., 1978. Computer aided ship design system, *WEGEMT Summer School on Advanced Ship Design Techniques*, Held at The University of Newcastle Upon Tyne.
- Bertram, V., 1998. Knowledge-based system for marine applications. *26<sup>th</sup> WEGEMT School on Expert Systems for Marine Applications*, Hamburg.
- Börü, S., 1999. Private communication, İstanbul.
- Brown, D. K., 1986. Defining a warship, *Naval Engineers Journal*, 98-2, 31-40.

- Buxton, I.L.**, 1972. Engineering economics applied to ship Design, *Transaction of The Royal Institution of Naval Architects*, RINA, **114**, 409-428.
- Carlson, C.M. and Fireman, H.**, 1987. General arrangement design computer system, and methodology, *Naval Engineers Journal*, **99-3**, 261-273.
- Cort, A., and Hills, W.**, 1987. Space layout design using computer assisted methods, *Naval Engineers Journal*, **99-3**, 249-260.
- Coyne, R. D., Rosenman, M. A., Radford, A. D., Balachandran, M. and Gero, J. S.**, 1990. Knowledge-Based Design Systems, pp.3-456, Addison-Wesley Publishing Company, Sydney.
- Dayı, O.**, 1998. Private communication, Istanbul.
- Demirsoylu, A.** 1999. Private communication, Istanbul.
- Duffy, A.H.B.**, 1986. Computer modelling of early stage numerical ship design knowledge and expertise, *PhD Thesis*, Department of Ship and Marine Technology, University of Strathclyde.
- Duffy, A.H.B.**, 1993. Lectures on expert systems in engineering. course notes, Strathclyde University, Second edition, October.
- Dym, C.L. and Levitt, R.E.**, 1991. Knowledge-Based Systems in Engineering, pp. 211-328, McGRAW-HILL International Editions, New York.
- Ericstad, S.O.**, 1994. Improving concept exploration in the early stages of the ship design process, *5<sup>th</sup> International Marine Design Conference*, IMDC'94, pp. 492-503, Delft.
- Evans, J.H.** 1959. Basic design concepts, *Naval engineers Journal*. November, **71-4**, 671-678.
- Firebaugh, M.W.**, 1989. AI, A Knowledge-Based Approach, pp. 73-100. PWS-KENT Publishing Company, Boston.
- Francis, R.L. and White, J.A.**, 1974. Facility Layout and Location: An Analytical Approach, pp. 1-100, Prentice-Hall Inc, New Jersey.
- Gemiadamları Mevzuatı**, 1997. T.C. Başbakanlık, Denizcilik Müsteşarlığı. Deniz Ulaştırması Gn. Md., Başbakanlık Basımevi. Ankara.
- Giarratano, J. and Riley, G.**, 1994. Expert Systems: Principles and Programming, pp. 1-594 PWS Publishing Company, Boston.
- Göke, G.** 1998. Private communication, İstanbul.
- Gürsel, T.**, 1999. Private communication, Tuzla.

- Han, S.-H. and Lee, D.**, 1996. Integration of ship design software modules utilizing the superapplication concept, *Journal of Marine Science and Technology*, SNAJ, **1**, 268-273.
- Helvacioğlu, S.**, 1995. Artificial intelligence, expert systems and expert systems applications in ship design, *Naval Architecture and Ocean Engineering Technical Congress Proceedings*, pp. 24-34 (in Turkish).
- Hills, W., Phil, M., and Buxton, I.L.**, 1989. Integrating ship design and production considerations during the pre-contract phase, *Transaction of The Royal Institution of Naval Architects*, RINA, **131**, 1-14.
- ILO**, 1998, American Bureau of Shipping, ILO Conventions Surveys, Check Sheets.
- Jackson, P.**, 1999. Introduction to Expert Systems, pp. 1-96, Addison-Wesley Publications, New York.
- Keane, A.J.**, 1987. A computer based method for hull form concept design: application to stability analyses, *Transaction of The Royal Institution of Naval Architects*, RINA, **129**, 1-12.
- Kenneth, K., Brower, K.S. and Walker, K.W.**, 1986. Ship design computer programs an interpolative technique, *Naval Engineers Journal*, **99-3**, 74-87.
- Kim, Y.D., Yum, J.S., and Lee, S.S.**, 1997. An initial ship design with step database, *9<sup>th</sup> International Conference on Computer Applications in Ship Building*, ICCAS'97, October, Japan, Ed. K. Johnson and T. Koyama, pp. 599-610.
- Koh, H.S.** 1991. A method for computer-aided general arrangement design of ships, *Ship Technology Research*, **38**, pp. 140-156.
- Kupras, L.K. and de Zwaan, A.P.**, 1978. Preliminary ship design with interactive graphical aids, *WEGEMT Summer School on Advanced Ship Design Techniques*, Held at The University of Newcastle Upon Tyne.
- Lamb, T.**, 1969. A ship design procedure, *Marine Technology*, October, **6-4**, 362-405.
- Lee, K.H. and Lee, D.**, 1997. Generation of design candidates and design assistance by using case-based reasoning at preliminary design stage, *9<sup>th</sup> International Conference on Computer Applications in Ship Building*, ICCAS'97, October, Japan, Ed. K. Johnson and T. Koyama pp. 483-522.
- Lee, K.Y. and Lee, K.H.**, 1996. Knowledge-based optimum ship design, *Ship Technology Research*, **43**, pp. 106-114.
- Lee, K.Y., Suh, S.W., Shin, D.W., Lee, D.K., Kang, W.S. and Kim, Y.D.**, 1995. Development of a computerised ship design system, *The Sixth*

- Lyon, D., and Mistree, F., 1985.** A computer based method for the preliminary design of ships, *Journal of Ship Research*, **29**, Dec., pp. 251-269.
- MacCallum, K.J., 1978.** Computers graphics in ship design, *WEGEMT Summer School on Advanced Ship Design Techniques*, Held at The University of Newcastle Upon Tyne.
- MacCallum, K.J., 1982.** Creative ship design by computer, *9<sup>th</sup> International Conference Computer Applications in the Automation of Shipyard Operation and design (ICCAS)*, pp.55-62.
- MacGregor-Conver, 1996.** Container Securing Manual.
- Mandel, P. and Leopold, R., 1966.** Optimisation methods applied to ship design, *The Society of Naval Architects and Marine Engineers*, SNAME. **74**, 477-505.
- Martin, J., 1997.** Virtual reality techniques for ship and submarine design. *9<sup>th</sup> International Conference on Computer Applications in Ship Building, ICCAS'97*, October, Japan, Ed. K. Johnson and T. Koyama, pp. 149-162.
- Mistree, F., Smith, W.F., Bras, B.A., Allen, J.K. and Muster, D., 1990.** Decision – based design: a contemporary paradigm for ship design, *The Society of Naval Architects and Marine Engineers*, SNAME, **98**, pp.565-597.
- Munro-Smith, R., 1975.** Elements of Ship Design, pp. 15-40, Marine Media Management Limited, London.
- Murphy, R.D., Sabat, D.J. and Taylor, R.J., 1965.** Least cost ship characteristics by computer techniques, *Marine Technology*, **2**, April.
- Muther, R., 1961.** Systematic Layout Planning, pp. 15-120. Industrial Education Institute Mass, Boston.
- Nowacki, H., Brusis, F. and Swift, P.M., 1970.** Tanker preliminary design-an optimization problem with constraints, *The Society of Naval Architects and Marine Engineers*, SNAME. **78**, 357-372.
- Odabasi, A.Y., 1998.** Private communication, Istanbul.
- Palekar, U.S., 1999.** Plant Layout, IE261 Facilities Planning and Design Lecture Notes, University of Illinois.
- Parsaye, K. and Chignell, M., 1988.** Expert Systems for Experts, pp.1-30, John Wiley and Sons. Inc., New York.
- Pattison, D.R., Spencer, R.E. and van Griethuysen, W.J., 1982.** The computer aided ship design system goddess and its application to the structural

design of royal navy warships, *Computer Applications in the Automation of Shipyard Operation and Design IV*, pp. 341-353.

**Ray, T. and Sha, O.P.**, 1994. Multicriteria optimization model for a containership design, *Marine Technology*, **31(4)**, October, pp. 258-268.

**Reich, Y.**, 1998. Lectures on knowledge-based systems and CLIPS, *26<sup>th</sup> WEGEMT School on Expert Systems for Marine Applications*, Hamburg.

**SBG**, 1982, Safety code for shipping enterprises, See-Berufsgenossenschaft, Edition 1981, edited February 1982 by Germanischer Lloyd.

**Schalkoff, R.J.**, 1990. AI An Engineering Approach, pp. 1-50, McGraw-Hill Int. Editions, New York.

**Sekimoto, T., Shimuzi, K. and Koyama, T.**, 1988. An object oriented toll for the preliminary design of ships, *J.S.N.A. Japan*, **164**, Dec., pp. 167-177.

**Sharp, R.**, 1973. Optimum space allocation within buildings, *Building Science*, **8**, pp. 201-205.

**Sismanyazici, H.**, 1999. Private communication, Istanbul.

**Smith, W.F. and Mistree, F.**, 1994. The development of top level ship specifications: a decision-based approach, *5<sup>th</sup> International Marine Design Conference*, IMDC'94, pp. 59-77.

**Smith, W.F., Kamal, S. and Mistree, F.**, 1987. The influence of hierarchical decisions on ship design, *Marine Technology*, **2**, April, pp. 131-142.

**SOLAS**, 1997, International Maritime Organisation, IMO Publications, London.

**Taggart, R. (Editor)**, 1980. Ship Design and Construction, pp. 1-443, SNAME Publication, New York.

**Taylor, W.A.**, 1989. What Every Engineer Should Know about Artificial Intelligence. pp. 143-144, The MIT Press, Cambridge.

**Turkish Lloyd**, 1996. Rules for Classification of Steel Ships, Part A, Chapter –Hull, Istanbul.

**Turkish Lloyd**, 1999. Rules for Classification of Steel Ships Part B, Chapter 4-Machinery, Istanbul.

**Van Hees, M.Th.**, 1992. QUAESTOR: a knowledge-based system for computations in preliminary ship design, *International Symposium on Practical Design of Ships and Mobile Units*, PRADS, pp. 2.1284-2.1297.

**Van Hees, M.Th.**, 1995. Towards practical knowledge-based design modelling, *The Sixth International Symposium on Practical Design of Ships and Mobile Units*, PRADS, September, pp. 2.1300-2.1311.

- Van Hees, M.Th.**, 1998. Expert governed parametric model assembling, 26<sup>th</sup> *WEGEMT School on Expert Systems for Marine Applications*, Hamburg.
- Warszawski, A.**, 1972. Layout and location problems-the quantitative approach, *Building Science*, 7, pp. 61-67.
- Watson, D.G.M. and Gilfillan, M.**, 1976. Some ship design methods, *Transaction of The Royal Institution of Naval Architects*, RINA, 118, 279-303.
- Watson, D.G.M.**, 1998. *Practical Ship Design*, pp. 1-464, Elsevier Publication, Amsterdam, Lausanne, New York, Oxford, Shannon, Singapore, Tokyo.
- Watson, D.G.M.**, 2000. Private communication, Glasgow.
- Welsh, M.**, 1989. A computer aided conceptual ship design system incorporating expert knowledge, *PhD Thesis*, Department of Marine Technology, University of Newcastle-upon-Tyne, September.
- Welsh, M.**, 1994. Private communication, Newcastle.
- Williamson, M.**, 1986. *AI for Microcomputers: The Guide for Business Decision-makers*, pp. 3-11, Brady Communications, New York.
- Winstanley, G.** (Editor), 1991. *AI in Engineering*, pp. 1-100, John Wiley and Sons, Chichester.
- Yazdani, M.** (Editor), 1986. *AI: Principles and Applications*, pp. 1-150, Chapman and Hall, London.
- Ye, Q.**, 1991. A ship design method involving uncertainties and multiple criteria, *PhD Thesis*, Department of Ship and Marine Technology, University of Strathclyde.

## APPENDIX A CLIPS EXPERT SYSTEM SHELL

### A.1 The History of CLIPS

The origins of the C Language Integrated Production System (CLIPS) date back to 1984 at NASA's Johnson Space Center. The prototype version of CLIPS was developed in the spring of 1985 in a little over two months. Particular attention was given to making the tool compatible with expert systems. Thus, the syntax of CLIPS was made to very closely resemble the syntax of a subset of the ART expert system tool developed by Inference Corporation. Although originally modeled from ART, CLIPS was developed entirely without assistance from Inference or access to the ART source code.

The first release of CLIPS to groups outside of NASA, version 3.0, occurred in the summer of 1986. Originally, the primary representation methodology in CLIPS was a forward chaining rule language based on the Rete algorithm.

Version 5.0 of CLIPS, released in the spring of 1991, introduced two new programming paradigms: procedural programming (as found in languages such as C and Ada) and object-oriented programming (as found in languages such as the Common Lisp Object System and Smalltalk). The object-oriented programming language provided within CLIPS is called the CLIPS Object-Oriented Language (COOL). Because of its portability, extensibility, capabilities, and low-cost, CLIPS has received widespread acceptance throughout the government, industry, and academia. The development of CLIPS has helped to improve the ability to deliver expert system technology throughout the public and private sectors for a wide range of applications and diverse computing environments. CLIPS is being used by over 4,000 users throughout the public and private community including: all NASA sites and branches of the military, numerous federal bureaus, government contractors, universities, and many private companies. CLIPS is now maintained as public domain software by the main program authors who no longer work for NASA.

### A.2 Interacting With Clips

CLIPS expert systems may be executed in three ways: interactively using a simple, text-oriented, command prompt interface; interactively using a window/menu/mouse interface on certain machines; or as embedded expert systems in which the user provides a main program and controls execution of the expert system. In addition, a series of commands can be automatically read directly from a file when CLIPS is first started or as the result of the **batch** command.

The generic CLIPS interface is a simple, interactive, text-oriented, command prompt interface for high portability. The standard usage is to create or edit a knowledge base using any standard text editor, save the knowledge base as one or more text

files, exit the editor and execute CLIPS, then load the knowledge base into CLIPS. The interface provides commands for viewing the current state of the system, tracing execution, adding or removing information, and clearing CLIPS.

### A.2.1 Integration with Other Languages

When using an expert system, two kinds of integration are important: embedding CLIPS in other systems, and calling external functions from CLIPS. CLIPS was designed to allow both kinds of integration. Using CLIPS as an embedded application allows the easy integration of CLIPS with existing systems. This is useful in cases where the expert system is a small part of a larger task or needs to share data with other functions. In these situations, CLIPS can be called as a subroutine and information may be passed to and from CLIPS. It also may be useful to call external functions while executing a CLIPS construct or from the top-level of the interactive interface. CLIPS variables or literal values may be passed to an external function, and functions may return values to CLIPS. The easy addition of external functions allows CLIPS to be extended or customized in almost any way.

CLIPS provides three basic elements for writing programs: primitive data types, functions for manipulating data, and constructs for adding to a knowledge base.

## A.3 Basic Programming Elements

CLIPS provides three basic elements for writing programs: primitive data types, functions for manipulating data, and constructs for adding to a knowledge base.

### A.3.1 Data Types

CLIPS provides eight primitive data types for representing information. These types are **float**, **integer**, **symbol**, **string**, **external-address**, **fact-address**, **instance-name** and **instance-address**. Numeric information can be represented using floats and integers. Symbolic information can be represented using symbols and strings.

A **number** consists *only* of digits (0-9), a decimal point (.), a sign (+ or -), and, optionally, an (e) for exponential notation with its corresponding sign. A number is either stored as a float or an integer. Any number consisting of an optional sign followed by only digits is stored as an **integer** (represented internally by CLIPS as a C long integer). All other numbers are stored as **floats** (represented internally by CLIPS as a C double-precision float).

A **symbol** in CLIPS is any sequence of characters that starts with any printable ASCII character and is followed by zero or more printable ASCII characters.

A **string** is a set of characters that starts with a double quote (") and is followed by zero or more printable characters. A string ends with double quotes. Double quotes may be embedded within a string by placing a backslash (\) in front of the character. A backslash may be embedded by placing two consecutive backslash characters in the string.

An **external-address** is the address of an external data structure returned by a function (written in a language such as C or Ada) that has been integrated with CLIPS. This data type can only be created by calling a function (i.e. it is not possible to specify an external-address by typing the value).

A **fact** is a list of atomic values that are either referenced positionally (ordered facts) or by name (non-ordered or template facts). Facts are referred to by index or address.

An **instance** is an **object** that is an instantiation or specific example of a **class**. Objects in CLIPS are defined to be floats, integers, symbols, strings, multifield values, external-addresses, fact-addresses or instances of a user-defined class. A user-defined class is created using the **defclass** construct. An instance of a user-defined class is created with the **make-instance** function, and such an instance can be referred to uniquely by address. Within the scope of a module, an instance can also be uniquely referred to by name. An **instance-name** is formed by enclosing a symbol within left and right brackets.

An **instance-address** can only be obtained by binding the return value of a function called **instance-address** or by binding a variable to an instance matching an object pattern on the LHS of a rule. A reference to an instance of a user-defined class can either be by name or address; instance-addresses should only be used when speed is critical.

In CLIPS, a placeholder that has a value (one of the primitive data types) is referred to as a **field**. The primitive data types are referred to as **single-field values**. A **constant** is a non-varying single field value directly expressed as a series of characters. A **multifield value** is a sequence of zero or more single field values. Collectively, single and multifield values are referred to as **values**.

In CLIPS, a **variable** is a symbolic location that is used to store values. Variables are used by many of the CLIPS constructs (such as **defrule**, **deffunction**, **defmethod**, and **defmessage-handler**).

### A.3.2 Functions

A **function** in CLIPS is a piece of executable code identified by a specific name which returns a useful value or performs a useful side effect (such as displaying information). There are several types of functions. **User defined functions** and **system defined functions** are pieces of code that have been written in an external language (such as C, FORTRAN, or Ada) and linked with the CLIPS environment. System defined functions are those functions that have been defined internally by the CLIPS environment. User defined functions are functions that have been defined externally of the CLIPS environment.

The **deffunction** construct allows users to define new functions directly in the CLIPS environment using CLIPS syntax. Functions defined in this manner appear and act like other functions, however, instead of being directly executed (as code written in an external language would be) they are interpreted by the CLIPS environment.

Function calls in CLIPS use a prefix notation – the arguments to a function always appear after the function name. Function calls begin with a left parenthesis, followed by the name of the function, then the arguments to the function follow (each argument separated by one or more spaces). Arguments to a function can be primitive data types, variables, or another function call. The function call is then closed with a right parenthesis.

### A.3.3 Constructs

Several defining **constructs** appear in CLIPS: **defmodule**, **defrule**, **deffacts**, **deftemplate**, **defglobal**, **deffunction**, **defclass**, **definstances**, **defmessage-handler**, **defgeneric**, and **defmethod**. All constructs in CLIPS are surrounded by parentheses. The construct opens with a left parenthesis and closes with a right parenthesis. Defining a construct differs from calling a function primarily in effect. Typically a function call leaves the CLIPS environment unchanged. Defining a construct, however, is explicitly intended to alter the CLIPS environment by adding to the CLIPS knowledge base. Unlike function calls, constructs never have a return value.

## A.4 Data Abstraction

There are three primary formats for representing information in CLIPS: facts, objects and global variables.

### A.4.1 Facts

Facts are one of the basic high-level forms for representing information in a CLIPS system. Each **fact** represents a piece of information which has been placed in the current list of facts, called the **fact-list**. Facts are the fundamental unit of data used by rules. Facts may be added to the fact-list (using the **assert** command), removed from the fact-list (using the **retract** command), modified (using the **modify** command), or duplicated (using the **duplicate** command) through explicit user interaction or as a CLIPS program executes. The number of facts in the fact-list and the amount of information that can be stored in a fact is limited only by the amount of memory in the computer. If a fact is asserted into the fact-list that exactly matches an already existing fact, the new assertion will be ignored. A fact can be specified either by **fact-index** or **fact-address**. Whenever a **reset** or **clear** command is given, the fact-indices restart at zero. A fact may also be specified through the use of a fact-address. A fact-address can be obtained by capturing the return value of commands which return fact addresses (such as **assert**, **modify**, and **duplicate**) or by binding a variable to the fact address of a fact which matches a pattern on the LHS of a rule. A **fact identifier** is a shorthand notation for displaying a fact.

#### A.4.1.1 Ordered Facts

**Ordered facts** consist of a symbol followed by a sequence of zero or more fields separated by spaces and delimited by an opening parenthesis on the left and a closing parenthesis on the right. The first field of an ordered fact specifies a “relation” that

applied to the remaining fields in the ordered fact. For example, (father-of jack bill) states that bill is the father of jack.

Fields in a non-ordered fact may be of any of the primitive data types (with the exception of the first field which must be a symbol), and no restriction is placed on the ordering of fields. The following symbols are reserved and should not be used as the *first* field in any fact (ordered or non-ordered): *test*, *and*, *or*, *not*, *declare*, *logical*, *object*, *exists*, and *forall*. These words are reserved only when used as a deftemplate name.

#### A.4.1.2 Non-ordered Facts

Ordered facts encode information positionally. To access that information, a user must know not only what data is stored in a fact but which field contains the data. **Non-ordered (or deftemplate) facts** provide the user with the ability to abstract the structure of a fact by assigning names to each field in the fact. The **deftemplate** construct is used to create a template which can then be used to access fields by name. The deftemplate construct is analogous to a record or structure definition in programming languages such as Pascal and C.

The deftemplate construct allows the name of a template to be defined along with zero or more definitions of **named fields** or **slots**. Unlike ordered facts, the slots of a deftemplate fact may be constrained by type, value, and numeric range. In addition, default values can be specified for a slot. A slot consists of an opening parenthesis followed by the name of the slot, zero or more fields, and a closing parenthesis.

The immediate advantages of clarity and slot order independence for deftemplate facts should be readily apparent. In addition to being asserted and retracted, deftemplate facts can also be modified and duplicated (using the **modify** and **duplicate** commands). Modifying a fact changes a set of specified slots

#### A.4.1.3 Initial Facts

The **deffacts** construct allows a set of *a priori* or initial knowledge to be specified as a collection of facts. When the CLIPS environment is reset (using the **reset** command) every fact specified within a deffacts construct in the CLIPS knowledge base is added to the fact-list.

### A.4.2 Objects

An **object** in CLIPS is defined to be a symbol, a string, a floating-point or integer number, a multifield value, an external-address or an instance of a user-defined class. Objects are described in two basic parts: properties and behavior. A **class** is a template for common properties and behavior of objects which are **instances** of that class.

Objects in CLIPS are split into two important categories: primitive types and instances of *user-defined* classes. These two types of objects differ in the way they are referenced, created and deleted as well as how their properties are specified.

Primitive type objects are referenced simply by giving their value, and they are created and deleted implicitly by CLIPS as they are needed. Primitive type objects have no names or slots, and their classes are predefined by CLIPS. The behavior of primitive type objects is like that of instances of user-defined classes, however, in that you can define message-handlers and attach them to the primitive type classes. It is anticipated that primitive types will not be used often in an object-oriented programming (OOP) context; the main reason classes are provided for them is for use in generic functions.

The primary difference between object slots and template (or non-ordered) facts is the notion of inheritance. Inheritance allows the properties and behavior of a class to be described in terms of other classes. COOL supports multiple inheritance: a class may directly inherit slots and message-handlers from more than one class. Since inheritance is only useful for slots and message-handlers, it is often not meaningful to inherit from one of the primitive type classes, such as MULTIFIELD or NUMBER.

#### A.4.2.1 Initial Objects

The **definstances** construct allows a set of *a priori* or initial knowledge to be specified as a collection of instances of user-defined classes. When the CLIPS environment is reset (using the **reset** command) every instance specified within a **definstances** construct in the CLIPS knowledge base is added to the instance-list.

#### A.4.2.2 Global Variables

The **defglobal** construct allows variables to be defined which are global in scope throughout the CLIPS environment. That is, a global variable can be accessed anywhere in the CLIPS environment and retains its value independent of other constructs. In contrast, some constructs (such as **defrule** and **deffunction**) allow local variables to be defined within the definition of the construct. These local variables can be referred to within the construct, but have no meaning outside the construct.

### A.5 Knowledge Representation

CLIPS provides heuristic and procedural paradigms for representing knowledge.

#### A.5.1 Heuristic Knowledge – Rules

One of the primary methods of representing knowledge in CLIPS is a rule. Rules are used to represent heuristics, or “rules of thumb”, which specify a set of actions to be performed for a given situation. The developer of an expert system defines a set of rules which collectively work together to solve a problem. A **rule** is composed of an **antecedent** and a **consequent**. The antecedent of a rule is also referred to as the **if portion** or the **left-hand side** (LHS) of the rule. The consequent of a rule is also referred to as the **then portion** or the **right-hand side** (RHS) of the rule.

The antecedent of a rule is a set of **conditions** (or **conditional elements**) which must be satisfied for the rule to be applicable. In CLIPS, the conditions of a rule are satisfied based on the existence or non-existence of specified facts in the fact-list or specified instances of user-defined classes in the instance-list. One type of condition which can be specified is a **pattern**. Patterns consist of a set of restrictions which are used to determine which facts or objects satisfy the condition specified by the pattern. The process of matching facts and objects to patterns is called **pattern-matching**. CLIPS provides a mechanism, called the **inference engine**, which automatically matches patterns against the current state of the fact-list and instance-list and determines which rules are applicable.

The consequent of a rule is the set of actions to be executed when the rule is applicable. The actions of applicable rules are executed when the CLIPS inference engine is instructed to begin execution of applicable rules. If more than one rule is applicable, the inference engine uses a **conflict resolution strategy** to select which rule should have its actions executed. The actions of the selected rule are executed (which may affect the list of applicable rules) and then the inference engine selects another rule and executes its actions. This process continues until no applicable rules remain.

In many ways, rules can be thought of as IF-THEN statements found in procedural programming languages such as C and Ada. However, the conditions of an IF-THEN statement in a procedural language are only evaluated when the program flow of control is directly at the IF-THEN statement. In contrast, rules act like WHENEVER-THEN statements. The inference engine always keeps track of rules which have their conditions satisfied and thus rules can immediately be executed when they are applicable.

### A.5.2 Procedural Knowledge

CLIPS also supports a procedural paradigm for representing knowledge like that of more conventional languages, such as Pascal and C. Deffunctions and generic functions allow the user to define new executable elements to CLIPS that perform a useful side-effect or return a useful value. These new functions can be called just like the built-in functions of CLIPS.

Message-handlers allow the user to define the behavior of objects by specifying their response to messages. Deffunctions, generic functions and message-handlers are all procedural pieces of code specified by the user that CLIPS executes interpretively at the appropriate times. Defmodules allow a knowledge base to be partitioned.

### A.6 Basic Cycle Of Rule Execution

Once a knowledge base (in the form of rules) is built and the fact-list and instance-list is prepared, CLIPS is ready to execute rules. In a conventional language, the starting point, the stopping point, and the sequence of operations are defined explicitly by the programmer. With CLIPS, the program flow does not need to be defined quite so explicitly. The knowledge (rules) and the data (facts and instances)

are separated, and the inference engine provided by CLIPS is used to apply the knowledge to the data. The basic execution cycle is as follows:

- a) If the rule firing limit has been reached or there is no current focus, then execution is halted. Otherwise, the top rule on the agenda of the module which is the current focus is selected for execution. If there are no rules on that agenda, then the current focus is removed from the focus stack and the current focus becomes the next module on the focus stack. If the focus stack is empty, then execution is halted, otherwise step *a* is executed again.
- b) The right-hand side (RHS) actions of the selected rule are executed. The use of the **return** function on the RHS of a rule may remove the current focus from the focus stack. The number of rules fired is incremented for use with the rule firing limit.
- c) As a result of step b, rules may be **activated** or **deactivated**. Activated rules (those rules whose conditions are currently satisfied) are placed on the **agenda** of the module in which they are defined. The placement on the agenda is determined by the **salience** of the rule and the current **conflict resolution strategy**. Deactivated rules are removed from the agenda. If the activations item is being watched then an informational message will be displayed each time a rule is activated or deactivated.
- d) If **dynamic salience** is being used, the salience values for all rules on the agenda are reevaluated. Repeat the cycle beginning with step a.

## A.7 Conflict Resolution Strategies

The **agenda** is the list of all rules which have their conditions satisfied (and have not yet been executed). Each module has its own agenda. The agenda acts similar to a stack (the top rule on the agenda is the first one to be executed). When a rule is newly activated, its placement on the agenda is based (in order) on the following factors:

- a) Newly activated rules are placed above all rules of lower salience and below all rules of higher salience.
- b) Among rules of equal salience, the current conflict resolution strategy is used to determine the placement among the other rules of equal salience.
- c) If a rule is activated (along with several other rules) by the same assertion or retraction of a fact, and steps a and b are unable to specify an ordering, then the rule is arbitrarily (*not randomly*) ordered in relation to the other rules with which it was activated. Note, in this respect, the order in which rules are defined has an arbitrary effect on conflict resolution (which is highly dependent upon the current underlying implementation of rules). *Do not* depend upon this arbitrary ordering for the proper execution of your rules.

CLIPS provides seven conflict resolution strategies: depth, breadth, simplicity, complexity, lex, mea, and random. The default strategy is depth. The current strategy can be set by using the **set-strategy** command (which will reorder the agenda based upon the new strategy).

### A.7.1 Depth Strategy

Newly activated rules are placed above all rules of the same salience. For example, given that fact-a activates rule-1 and rule-2 and fact-b activates rule-3 and rule-4, then if fact-a is asserted before fact-b, rule-3 and rule-4 will be above rule-1 and rule-2 on the agenda. However, the position of rule-1 relative to rule-2 and rule-3 relative to rule-4 will be arbitrary.

### A.7.2 Breadth Strategy

Newly activated rules are placed below all rules of the same salience. For example, given that fact-a activates rule-1 and rule-2 and fact-b activates rule-3 and rule-4, then if fact-a is asserted before fact-b, rule-1 and rule-2 will be above rule-3 and rule-4 on the agenda. However, the position of rule-1 relative to rule-2 and rule-3 relative to rule-4 will be arbitrary.

### A.7.3 Simplicity Strategy

Among rules of the same salience, newly activated rules are placed above all activations of rules with equal or higher specificity. The **agenda** is the list of all rules which have their conditions satisfied (and have not yet been executed). Each module has its own agenda. The agenda acts similar to a stack (the top rule on the agenda is the first one to be executed). When a rule is newly activated, its placement on the agenda is based (in order) on the following factors:

- a) Newly activated rules are placed above all rules of lower salience and below all rules of higher salience.
- b) Among rules of equal salience, the current conflict resolution strategy is used to determine the placement among the other rules of equal salience.
- c) If a rule is activated (along with several other rules) by the same assertion or retraction of a fact, and steps a and b are unable to specify an ordering, then the rule is arbitrarily (*not randomly*) ordered in relation to the other rules with which it was activated. Note, in this respect, the order in which rules are defined has an arbitrary effect on conflict resolution (which is highly dependent upon the current underlying implementation of rules). *Do not* depend upon this arbitrary ordering for the proper execution of your rules.

CLIPS provides seven conflict resolution strategies: depth, breadth, simplicity, complexity, lex, mea, and random. The default strategy is depth. The current strategy can be set by using the **set-strategy** command (which will reorder the agenda based upon the new strategy).

## A.8 LHS Syntax

This section describes the syntax used on the LHS of a rule. The LHS of a CLIPS rule is made up of a series of conditional elements (CEs) that must be satisfied for

the rule to be placed on the agenda. There are eight types of conditional elements: **pattern** CEs, **test** CEs, **and** CEs, **or** CEs, **not** CEs, **exists** CEs, **forall** CEs, and **logical** CEs. The **pattern** CE is the most basic and commonly used conditional element. **Pattern** CEs contain constraints which are used to determine if any pattern entities (facts or instances) satisfy the pattern. The **test** CE is used to evaluate expressions as part of the pattern-matching process. The **and** CE is used to specify that an entire group of CEs must all be satisfied. The **or** CE is used to specify that only one of a group of CEs must be satisfied. The **not** CE is used to specify that a CE must not be satisfied. The **exists** CE is used to test for the occurrence of at least one partial match for a set of CEs. The **forall** CE is used to test that a set of CEs is satisfied for every partial match of a specified CE. Finally, the **logical** CE allows assertions of facts and the creation of instances on the RHS of a rule to be logically dependent upon pattern entities matching patterns on the LHS of a rule (truth maintenance).

### A.8.1 Pattern Conditional Element

**Pattern conditional elements** consist of a collection of field constraints, wildcards, and variables which are used to constrain the set of facts or instances which match the pattern CE. A pattern CE is satisfied by each and every pattern entity that satisfies its constraints. **Field constraints** are a set of constraints that are used to test a single field or slot of a pattern entity. A field constraint may consist of only a single literal constraint, however, it may also consist of several constraints connected together. In addition to literal constraints, CLIPS provides three other types of constraints: connective constraints, predicate constraints, and return value constraints. Wildcards are used within pattern CEs to indicate that a single field or group of fields can be matched by anything. Variables are used to store the value of a field so that it can be used later on the LHS of a rule in other conditional elements or on the RHS of a rule as an argument to an action.

The first field of any pattern *must* be a symbol and can not use any other constraints. This first field is used by CLIPS to determine if the pattern applies to an ordered fact, a template fact, or an instance. The symbol *object* is reserved to indicate an object pattern. Any other symbol used must correspond to a deftemplate name (or an implied deftemplate will be created). Slot names must also be symbols and cannot contain any other constraints. For object and deftemplate patterns, a single field slot can only contain one field constraint and that field constraint must only be able to match a single field (no multifield wildcards or variables). A multifield slot can contain any number of field constraints.

## **APPENDIX B. REGULATIONS**

### **B.1 Rules and Regulations For Ship Design**

Following knowledge base is prepared from German Port Standards, SOLAS requirements and ILO considerations which is taken from ABS worksheets for accommodation design.:

#### **B.1.1 Stairways**

SBG-SW1. [SBG,V.HULL-EQUIPMENT,#91/2]

If there are more than 3 steps then must be provided by handrails or rails at least on one side if the stairs are 0.80 m. wide or wider then rails must be on either side. If wider than 1.80 m. then there must be an additional rail in the middle.

SBG-SW2. [SBG-V.HULL-EQUIPMENT,#91/3]

In principle, stairs should be arranged in longitudinal ship direction.

SBG-SW3. [SBG,V.HULL-EQUIPMENT,#91/4]

Clear width is not less than 0.60 m.

SBG-SW4. [SBG-V.HULL-EQUIPMENT,#91/5]

Clear vertical height above the stairs must be at least 2.00 m.

SBG-SW5. [SBG,V.HULL-EQUIPMENT,#91/6]

The foothold area before stairs and landings must not be less than 0.6 m. x 0.6 m.

SBG-SW6. [SBG,V.HULL-EQUIPMENT,#91/8]

The inclination of the stairways towards the horizontal shall be  $45^{\circ}$ , inclination shall not exceed  $55^{\circ}$ . In main and auxiliary machinery spaces and other narrow spaces: max. inclination is  $60^{\circ}$ .

SBG-SW7. [SBG,V.HULL-EQUIPMENT,#91/9]

The height between two landings is not more than 3.70 m.

SBG-SW8. [SBG,V.HULL-EQUIPMENT,#91/11]

Deck openings from stairs to accommodation or service spaces must be at least B/5 apart from shell side. Where this is not possible, a second stairway shall be arranged but not at the same side.

SBG-SW9. [SBG,V.HULL-EQUIPMENT,#91/12 D (1)]

The height of the steps is to be 0.18 m. to maximum 0.23 m. The steps of a stairway must have equal distances one from the other.

SBG-SW10. [SBG,V.HULL-EQUIPMENT,#91/12 D (4)]

The doors belonging to the stairs are to be designed in a corresponding breadth.

SOLAS-SW2. [SOLAS'97,CHAPTER-II-2,PART-C,REGULATION-46/1]

Stairways which penetrate only a single deck shall be protected at least at one level by at least B-0 class division and self closing doors. Stairways which penetrate more than a single deck shall be surrounded by at least A-0 class divisions and be protected by self-closing doors at all levels.

### **B.1.2 Companion Ways**

SBG-CW1. [SBG,V.HULL-EQUIPMENT,#92/1]

The clear height of companion ways on and below deck has to be min. 2.0 m. above deck.

SBG-CW2. [SBG,V.HULL-EQUIPMENT,#92/2]

Companion ways on the weather deck between accommodation and operational installations are to be provided on each side of the hatches.

SBG-CW3. [SBG,V.HULL-EQUIPMENT,#92/3]

Companion passages below deck of the same minimum measurements are admissible as a substitute. They must be sufficiently lighted and ventilated.

SBG-CW4. [SBG,V.HULL-EQUIPMENT,#92/4]

Companion passages between accommodation for crew members of the machinery department and the machinery spaces must be arranged completed wind and weather protected.

SOLAS-PCW1. [SOLAS'97,CHAPTER-II-2,PART-B,REGULATION-28/1.5.3]

Doorways and corridors and intermediate landings included in means of escape shall be sized in the same manner stairways.

### B.1.3 Emergency Escapes

SBG-EE1. [SBG,V.HULL-EQUIPMENT,#93/3]

Emergency escapes and emergency exits must lead to the open air or into a safe area by the shortest possible way. They must be sufficiently lighted, even if the main electrical source fails.

SBG-EE2. [SBG,V.HULL-EQUIPMENT,#93/4]

Emergency escapes and emergency exits must not be narrowed and are always to be kept clear.

SBG-EE3. [SBG,V.HULL-EQUIPMENT,#93/5 (D) 1]

Each compartment is to have two ways of exit in different directions.

SOLAS-EE10. [SOLAS'97,CHAPTER-II-2,PART-B.REGULATION-28/2.2]

One of the escape routes from the machinery spaces where the crew is normally employed shall avoid direct access to any special category space.

SOLAS-EE11. [SOLAS'97,CHAPTER-II-2,PART-B.REGULATION-28/3.1]

Two means of escape shall be provided from each machinery space

Below the bulkhead deck two means of escape shall consist of

i- two sets of steel ladders as widely separated as possible leading to doors in the upper part of the space similarly separated and from which access is provided to the appropriate lifeboat and liferaft embarkation decks. One of them shall provide continuous fire shelter from the lower part of the space to a safe position outside the space or

ii- One steel ladder like above mentioned and in a position well separated from the ladder referred to, a steel door capable of being operated from each side and which provides access to a safe escape route to the embarkation deck.

SOLAS-EE12. [SOLAS'97,CHAPTER-II-2,PART-B.REGULATION-28/4]

Lifts shall not be considered as forming one of the required means of escape.

SOLAS-EE13. [SOLAS'97,CHAPTER-II-2,PART-C,REGULATION-45/1.1,1.2.1,1.2.2]

Means of escape

i- at all levels of accommodation there shall be provided at least two widely separated means of escape from each restricted space or group of spaces.

ii.i) Below the lowest open deck the main means of escape shall be a stairway and the second may be a trunk or a stairway.

ii.ii) Above the lowest escape shall be stairways or doors to a open deck or a combination.

SOLAS-EE14. [SOLAS'97, CHAPTER-II-2, PART-C, REGULATION-45/1.4]

i-No dead end corridors having a length of more than 7 m. shall be accepted.

SOLAS-EE15. [SOLAS'97, CHAPTER-II-2, PART-C, REGULATION-45/3.3.1]

Two means of escape shall be provided from each machinery space of category A. In particular one of the following provisions shall be complied with:

a) two sets of steel leaders as widely separated as possible leading to doors in the upper part of the space similarly separated and from which access is provided to the open deck. In general one of these ladders shall provide continuous fire shelter from the lower part to a safe place.

b) Instead of two ladders, there shall be one ladder and one door which is well separated from the ladder and operated from each side and provides access to a safe escape.

v) Lift shall not be considered as a means of escape

#### **B.1.4 Sick Room**

SBG-SR1. [SBG, REGULATION<sup>1</sup>, SECTION-2, 8/2]

Each bed must have at least 12 m<sup>3</sup> of air space, or at least 10 m<sup>3</sup> of air space where there are more than two beds. The entrance must be wide enough to enable a sick person to be carried in on a stretcher.

SBG-SR2. [SBG, REGULATION, SECTION-2, 8/3]

At least 1 bed up to 30 person. The height of the sick-room in at least 210 cm. If there are more than two beds the others may be fitted as upper berths. The lower berth shall be at least 50 cm. above the floor.

ILO-SR1. [ILO.No.92-ABS-CHECK-SHEET-A, REVISION.1, ARTICLE-14/1]

If this ship carries a crew of fifteen or more and is engaged in voyages of more than 3 days duration, there should be a separate hospital accommodation provided.

#### **B.1.5 Habitability**

SBG-CH1. [SBG, APPENDIX<sup>2</sup>, 1.151]

---

<sup>1</sup> Regulation for the care of the sick on merchant vessels, 25<sup>th</sup> April 1972

The height from floor to ceiling of the accommodation spaces must be at least 2 m.

SBG-CH2. [SBG,APPENDIX, 1.156]

Furnishing must not have any sharp edges. Occupancy of a living compartment must not be exceed the following maximum numbers.

SBG-CH3. [SBG,APPENDIX,1.25 (1) 1]

Ship's officers, other and senior ratings: 1 person per compartment.

SBG-CH4. [SBG,APPENDIX,1.25 (1) 2]

Junior ratings: 2 person per compartment.

ILO-CH1. [ILO.No.92-ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-10/7]

The clear headroom in crew sleeping rooms should not be less than 190 cm.

The maximum number of persons should be allowed to occupy sleeping rooms not in excess of the following:

ILO-CH2. [ILO.No.92-ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-10/9.a]

Officers in charge of a department, navigating and engineer officers in charge of a watch and senior radio officers or operators one person per room.

ILO-CH3. [ILO.No.92-ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-10/9.b]

Other officers one person per room, and in no case more than two.

ILO-CH4. [ILO.No.92-ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-10/9.c]

Petty officers: one or two persons per room and in no case more than two.

ILO-CH5. [ILO.No.92-ABS-CHECK-SHEET-A, REVISION.1,ARTICLE-10/9.d]

Other ratings two or three persons per room.

ILO-CH6. [ILO.No.92-ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-10/12]

There should be on individual berth for each member of the crew.

ILO-CH7. [ILO.No.133-ABS-CHECK-SHEET-B,REVISION.1,ARTICLE-10/12]

---

<sup>2</sup> Appendix to the order regarding crew accommodation on board merchant ships

The minimum headroom in all crew accommodation where full and free movement necessary of at least 198 cm.

SOLAS-CH1. [SOLAS'97,CHAPTER-III,PART-B,REGULATION-11/3]

Muster and embarkation station shall be readily accessible from accommodation and work areas.

### **B.1.6 Lighting**

SBG-L1. [SBG,APPENDIX,1.161]

With the exception of pantries, storerooms and rooms containing sanitary equipment, all accommodation must receive adequate daylight.

### **B.1.7 Living Compartments**

SBG-LC1. [SBG,APPENDIX,1.21 (3)]

The living compartments must not be connected directly through openings to the storerooms, holds engine rooms, or boilers rooms, galleys drying rooms, communal washrooms or toilets. There must be general working passages or air locks between the living compartments and the other rooms.

SBG-LC2. [SBG,APPENDIX,1.23 (2)]

Each living compartment must be fitted with a fixed fold down or pull-out table or desk, a mirror, an electric shaver socket, a connection to a community aerial, a small cupboard for toilet requisites for each member of crew, a book shelf, a sufficient number of clothes hooks and comfortable seating. In addition there must be a drawer with a capacity of 0.1 m<sup>3</sup> for each crew. The living compartment windows must be fitted with curtains.

SBG-LC3 [SBG,APPENDIX,1.25 (2)]

Those without watch duty do not share a living compartment with crew members who do watch.

SBG-LC4. [SBG,APPENDIX,1.26 (2)]

A room must be provided on the ship for storing trunks and similar bulky articles belonging to the crew.

SBG-LC5. [SBG,APPENDIX,1.53]

On ships of 1600 GRT or above changing rooms must be provided for the engine room staff. They must be outside the engine casing but easily accessible from there and must be equipped with individual lockers as well as wash-basins and baths or showers.

SBG-LC6. [SBG,APPENDIX,1.7 (1)]

On ships up to 3000 GRT, a room must be provided to serve as an office for the deck and machinery personnel.

SBG-LC7. [SBG,APPENDIX,1.21 (3)]

On ships of 3000 GRT or above the deck service and the machinery service will each be provided with a room to serve as on office

ILO-LC1. [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-6/1]

The location of crew accommodation and means of access, structure and arrangement in relation to other spaces is to insure adequate security, protection against weather and sea, and insulation from heat or cold, undue noise or effluvia from other spaces.

ILO-LC2. [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-6/2]

There shouldn't be any direct openings between sleeping rooms and spaces for cargo and machinery or galleys, lamp and paint rooms or engine deck and other bulk storerooms, drying rooms, communal wash places or water closets.

ILO-LC3. [ILO.No.92,ABS-CHECK-SHEET-A,REV.1,ARTICLE-6/3.a,3.b.3.c]

External bulkheads of sleeping rooms and messrooms should be adequately insulated.

b) Machinery casings and all boundary bulkheads of galleys and other spaces in which heat is produced should be adequately insulated where there is a possibility of resulting heat effects in adequately insulated where there is a possibility of resulting heat effects in adjoining accommodations or passageways.

c) Adequate protection should be provided from heat effects of steam and hot water service pipes.

ILO-LC4. [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-6/5]

Sleeping rooms, recreation rooms, mess rooms and alleyways in the crew accommodation space should be adequately insulated to prevent condensation or overheating.

ILO-LC5. [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-6/6.a]

Main steam and exhaust pipes for winches and similar gear installed should not pass through crew accommodations.

ILO-LC6 [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-6/6.b]

If there is such piping installation through alleyways leading to accommodations, then they shall be adequately insulated and encased.

ILO-LC7 [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-7/1]

Sleeping rooms and mess rooms should be adequately ventilated.

ILO-LC8. [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-9/2.a]

All crew spaces should be adequately lighted.

ILO-LC9. [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-10/1]

Sleeping rooms should be situated above the load line amidships or aft.

ILO-LC10. [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-10/23]

Each sleeping room shall be provided with a table or desk of the fixed drop-leaf or slide out type, and with comfortable seating accommodations as necessary.

ILO-LC11. [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-10/27]

Sleeping rooms shall be fitted with a mirror, small cabinets for toilet requisites, a book rack and a sufficient number of coat hooks.

ILO-LC12. [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-10/28]

As far as practicable, the berthing of crew members should be arranged that watches are separated so that no dayman share a room with a watch keeper.

ILO-LC13. [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-12/1]

a- A space or spaces to which the crew can have access when off duty should be provided on a open deck. b- The space of adequate area should be having regard to the size of the ship and the crew.

ILO-LC14. [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-13/8]

All water closets should have ventilation to the open air independently of any other part of the accommodation.

ILO-LC15. [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-13/14]

Facilities provided or drying clothes is a compartment separate from sleeping rooms and mess rooms should be adequately ventilated and heated, and equipped with lines or other fittings for hanging clothes.

ILO-LC16. [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-15/1]

Sufficiently and adequately ventilated accommodation for the hanging of oil skins should be provided outside but convenient to the sleeping rooms.

ILO-LC17. [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-15/2]

In ships of over 3000 tons one room provided and equipped for use as an office for the deck department and one for the engine department.

ILO-LC18. [ILO.No.92,ABS-CHECK-SHEET-B,REVISION.1,ARTICLE-5/8]

Where practicable in ships of 3000 tons or over the chief engineering and chief navigating officers have in addition to their sleeping rooms, an adjoining sitting room or day room.

ILO-LC19. [ILO.No.92,ABS-CHECK-SHEET-B,REVISION.1,ARTICLE-8/2]

In ships of 5.000 tons or over but less than 15000 tons individual sleeping rooms for at least fire officers, have attached to them a separate private bathroom fitted with a water closet, as well as a tub and/or shower bath and a wash basin having hot and cold running fresh water.

ILO-LC20. [ILO.No.92,ABS-CHECK-SHEET-B,REVISION.1,ARTICLE-8/2.a]

In ships of 10000 tons or over but less than 15000 tons, the sleeping rooms of all other officers should have private or inter communicating bathrooms similarly fitted.

ILO-LC21. [ILO.No.92,ABS-CHECK-SHEET-B,REVISION.1,ARTICLE-8/3]

In ships of 15000 tons or over individual sleeping rooms for officers should have attached to them a separate private bathroom fitted with a water closed as well as a tub and/or shaver bath and a wash basin having hot and cold running fresh water.

ILO-LC22. [ILO.No.92,ABS-CHECK-SHEET-B,REVISION.1,ARTICLE-8/4]

In ships of 25.000 tons or over, other than passenger ships, a bathroom for every two ratings should be provided either is an intercommunicating compartment between and joining sleeping rooms or opposite the entrance of such rooms, which has been fitted with a water closet as well as tub and/or shower bath and a wash basin having hot and cold running fresh water.

ILO-LC23. [ILO.No.92,ABS-CHECK-SHEET-B,REVISION.1,ARTICLE-8/7]

The facilities provided should include: washing machines, drying machines or adequately heated and ventilated drying room, irons and ironing boards or their equivalent.

ILO-LC24. [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-9/2]

In ships of 1600 tons or over, other than ships which have private sleeping rooms and private or semiprivate bathrooms for all engine department personnel, facilities shall be provided for changing clothes which shall be:

Located outside the machinery space but with easy access to it.

Fitted with individual clothes lockers as well as with tubs and/or shower baths and wash basins having hot and cold running fresh water.

### **B.1.8 Crew Bunks/Beds**

SBG-CB1. [SBG,APPENDIX,1.22/(1)]

The bunk must be at least 2.00 x 0.8 m.

SBG-CB2. [SBG,APPENDIX,1.22/(2)]

Bunks must not be placed side by side.

SBG-CB3. [SBG,APPENDIX,1.22/(3)]

The lower bunk of two must be at least 0.30 m. above the floor, and the top bunk must be at least 0.80 m. below the lowest part of the ceiling above the bunk.

ILO-CB1. [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-10/13]

If the berths are installed side by side, the access to each berth should be gained without going over the other party.

ILO-CB2. [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-10/15]

If double tier berths are fitted, the lower berth should be not less than 30 cm. above the floor.

b) The upper berth should be placed approximately midway between the bottom of the lower berth and the lower side of the deck head beam.

ILO-CB3. [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-10/16]

The minimum inside dimensions of each dimensions of each berth should be not less than 190 cm. x 68 cm.

ILO-CB4. [ILO.No.133,ABS-CHECK-SHEET-B,REVISION.1,ARTICLE-5/10]

The minimum inside dimensions of a berth should be 9.8 cm. by 30 cm.

### **B.1.9 Crew Lockers**

SBG-CL1. [SBG,APPENDIX,1.23/(1)]

A lockable clothes locker must be available for each crew member. Locker must have a clear height inside of at least 1.70 m. and an internal cross-section area of at least 1.25 m<sup>2</sup>.

SBG-CL2. [SBG,APPENDIX,EDITION-PAGE.10-In ref.to.no-1.23-para.2]

Clothes lockers must have an internal measurement of at least 55 cm. in one direction. The height of the clothes hanger bar above the inside floor of the locker must be not less than 140 cm.

SBG-CL3. [SBG,APPENDIX,EDITION-PAGE.9-In ref.to.no-1.23-para.1]

The cupboard for toilet requisites belonging to the crew member must be above the wash-basin or its immediate vicinity.

ILO-CL1. [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-10/22]

a- The room furnishing include a clothes locker for each occupant. b- The minimum dimensions should not be less than 152 cm. in height with a cross section area of 0.193 m<sup>2</sup> and fitted with a shelf and hasp for a padlock.

ILO-CL2. [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-10/25]

The drawer of equivalent space for each occupant not less than 0.0056 m<sup>3</sup>.

#### **B.1.10 Floor Area**

In living compartments the floor area per crew member must not be less than

SBG-FA1. [SBG,APPENDIX,1.24/(1).1]

3.75 m<sup>2</sup> on ships of less than 3000 GRT

SBG-FA2. [SBG,APPENDIX,1.24/(1).2]

4.25 m<sup>2</sup> on ships of 3000<GRT<10000

SBG-FA3. [SBG,APPENDIX,1.24/(1).3]

4.75 m<sup>2</sup> on ships of 10000 GRT or more

In living compartments, however in which two crew members are accommodated, the floor area per crew member must not be less than

SBG-FA4. [SBG,APPENDIX,1.24/(2).1]

2.75 m<sup>2</sup> on ships of less than 3000 GRT

SBG-FA5. [SBG,APPENDIX,1.24/(2).2]

3.25 m<sup>2</sup> on ships of 3000 GRT or more but less than 10000 GRT

SBG-FA6. [SBG,APPENDIX,1.24/(2).3]

3.75 m<sup>2</sup> on ships of 10000 GRT or more.

In officers living compartments the floor area must be not less than

SBG-FA7. [SBG,APPENDIX,1.24/(4).1]

6.5 m<sup>2</sup> on ships of less than 3000 GRT

SBG-FA8. [SBG,APPENDIX,1.24/(4).2]

7.5 m<sup>2</sup> on ships of 3000 GRT and above if no separate living room or workroom is available in addition to the bedroom.

SBG-FA9. [SBG,APPENDIX,1.24/(5)]

The area occupied by the bunks, lockers and seating can be included in the calculation of total floor area. Small or irregular-shaped areas, which do not effectively increase the space for movement and cannot be used as storage space will be excluded.

The minimum floor area per person of sleeping rooms should be intended for ratings not less than.

ILO-FA1. [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-10/4.a]

1.85 m<sup>2</sup> in vessels under 800 tons.

ILO-FA2. [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-10/4.b]

2.35 m<sup>2</sup> vessels 800 tons or over but under 3000 tons.

ILO-FA3. [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-10/4.c]

2.78 m<sup>2</sup> in vessels of 3000 tons or over.

If the number of crew is more than enough than the minimum floor area could be reduced. Minimum floor area should not be less than

ILO-FA4. [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-10/5.b(i)]

1.67 m<sup>2</sup> Per person in ships under 3000 ton.

ILO-FA5. [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-10/5.b(ii)]

1.85 m<sup>2</sup> Per person in ships under 3000 ton or over.

ILO-FA6. [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-10/6]

The measurement of the floor are includes the space occupied by berths and lockers, chests of drawers and seats and excludes small or irregularly shaped spaces which do not add effectively to the space available for free movement and the installation of furniture.

The floor area, Per person of sleeping rooms for ratings should not be less than:

ILO-FA7. [ILO.No.133,ABS-CHECK-SHEET-B,REVISION.1,ARTICLE-5/1.a]

In ships of 100 tons or over but less than 3000 tons 3.75 m<sup>2</sup>

ILO-FA8. [ILO.No.133,ABS-CHECK-SHEET-B,REVISION.1,ARTICLE-5/1.b]

In ships of 3000 tons or over but less than 10000 tons 4.25 m<sup>2</sup>

ILO-FA9. [ILO.No.133,ABS-CHECK-SHEET-B,REVISION.1,ARTICLE-5/1.c]

In ships of 10000 tons or over 4.75 m<sup>2</sup>

The floor area, Per person of sleeping in rooms intended for two ratings not less than:

ILO-FA10. [ILO.No.133,ABS-CHECK-SHEET-B,REVISION.1,ARTICLE-5/2.a]

In ships of 1000 tons or over but less than 3000 tons 2.75 m<sup>2</sup>.

ILO-FA11. [ILO.No.133,ABS-CHECK-SHEET-B,REVISION.1,ARTICLE-5/2.b]

In ships of 3000 tons or over but less than 10000 tons 3.25 m<sup>2</sup>

ILO-FA12. [ILO.No.133,ABS-CHECK-SHEET-B,REVISION.1,ARTICLE-5/2.c]

In ships of 10000 tons or over 3.75 m<sup>2</sup>

In sleeping rooms for officers where no private sitting room or day room is provided, the floor area should not be less than.

ILO-FA13. [ILO.No.133,ABS-CHECK-SHEET-B,REVISION.1,ARTICLE-5/6.a]

6.5 m<sup>2</sup> in ships of less than 3000 tons

ILO-FA14. [ILO.No.133,ABS-CHECK-SHEET-B,REVISION.1,ARTICLE-5/6.b]

7.5 m<sup>2</sup> in ships of 3000 tons or over.

ILO-FA15. [ILO.No.133,ABS-CHECK-SHEET-B,REVISION.1,ARTICLE-6/1]

The floor area of mess rooms for officers and for ratings should be not less than 1 m<sup>2</sup> per person of the planned seating capacity.

#### **B.1.11 Mess Rooms.**

SBG-MR1. [SBG,APPENDIX,1.31/(1)]

The floor area for each person must not be less than 1 m<sup>2</sup> A seat must be provided for each person.

SBG-MR2. [SBG,APPENDIX,1.31/(1).1]

The messes must be sufficient for the number of crew members who normally use them at the same time.

SBG-MR3. [SBG,APPENDIX,1.31/(1).2]

The messes must be situated near the galley and be separated from the living quarters.

SBG-MR4. [SBG,APPENDIX,1.32/(1)]

Separate equipment for washing and storing the dishes used in the messen must be available.

SBG-MR5. [SBG,APPENDIX,EDITION-PAGE.11-In ref.to.no-1.31-para.2]

A min width of 60 cm. will be provided for each dining place.

ILO-MR1. [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-11/1]

Sufficient mess room accommodation should be provided in the vessel.

ILO-MR2. [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-11/2]

If the vessel is less than 1000 tons, there should be separate mess room accommodations for: Master and officers, petty officers and other ratings.

ILO-MR3. [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-11/3]

If the vessel is in 1000 tons or over, there should be separate mess room accommodations. for:

Master and officers

Deck department petty officers and other ratings.

Engine department petty officers and other ratings.

ILO-MR4. [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-11/5]

The dimensions and equipment of each mess room should be sufficient for the number of persons likely to use them at any one time.

ILO-MR5. [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-11/8]

Mess rooms shall be located apart from the sleeping rooms and as close as practicable to the galley.

#### **B.1.12 Pantry**

SBG-PN1. [SBG,APPENDIX,1.32/(2)]

If the pantries are provided, these must be accessible direct from the messes or must be easily reached from the mess.

SBG-PN2. [SBG,APPENDIX,1.32/(2)]

The pantries must be specially equipped with a double sink, a fitment for dispensing disposable hand towels, a cold and warm drinking water supply, a spent air extraction system, a refrigerator large enough for the crew or and equipment for preparing hot drinks at any time.

SBG-PN3. [SBG,APPENDIX,1.32/(3)]

If there is no pantry, equipment in para 2. must be provided in mess room or somewhere else near by.

### **B.1.13 Recreation Rooms.**

SBG-RR1. [SBG,APPENDIX,1.33/(1)]

Suitably located and appropriately equipped recreation rooms must be provided for the crew. Messes may be used. There must be at least one bookshelf in the recreation rooms and facilities for reading and writing.

SBG-RR2. [SBG,APPENDIX,1.33/(2)]

On ships of 8000 GRT and above, a smokers room or reading room in which films or TV can be watched, and a hobby and games room must be provided.

ILO-RR1. [ILO.No.92.ABS-CHECK-SHEET-A,REVISION.1.ARTICLE-12/2]

Recreation accommodation, conveniently situated and appropriately furnished should be provided for officers and for ratings.

ILO-RR2. [ILO.No.133,ABS-CHECK-SHEET-B,REVISION.1.ARTICLE-7/3]

In respect of ships of 8000 tons or over should have a smoking room or library room in which films or TV may be shown and a hobby and game room should be provided.

### **B.1.14 Galleys**

SBG-GR1. [SBG,APPENDIX,1.41]

Galleys must be equipped:

cooking utensils, a double sink, a wash basin and a dispenser for disposable hand towels a tap for cold and warm drinking water, the necessary cupboards, shelves and crockery racks of a suitable rust-proof material a spent air extraction system, two floor drains.

### **B.1.15 Provision Rooms**

SBG-PR1. [SBG,APPENDIX,1.42/(1)]

The provision rooms, apart from the cold stores, must be kept dry and well ventilated. The goods must be stored separately according to their different temperature requirements.

### **B.1.16 Sanitary Equipment**

Washing facilities, lavatories.

SBG-SR1. [SBG,APPENDIX,1.511/(1).1]

a wash basin with cold and warm drinking water for every 4 male and every 4 female.

SBG-SR2. [SBG,APPENDIX,1.511/(1).2]

One bath or one shower with cold and warm drinking water for every 6 male and every 6 female

SBG-SR3. [SBG,APPENDIX,1.511/(1).3]

a toilet compartment for every 6 male and for every 6 female, but a total of at least.

3 on ships of less than 1.000 GRT.

4 on ships of 1000 GRT and above, but less than 5000 GRT

6 on ships of 5000 GRT and above

SBG-SR4. [SBG,APPENDIX,1.511/(3).1]

There must be separate toilets for serving and catering staff on the scale laid down in sentence 1.

On ships of 5000 GRT or above.

SBG-SR5. [SBG,APPENDIX,1.511/(2).1]

Every living compartment must be equipped with a wash basin is built into the shower room or bathroom.

SBG-SR6. [SBG,APPENDIX,1.511/(2).2]

A room with bath or shower, a wash basin with cold and warm drinking water and a lavatory will be provided for each officer next to his living compartment; the wash basin may be installed in the living compartment.

SBG-SR7. [SBG,APPENDIX,1.511/(3)]

On ships of 10000 GRT and above a room with bath or shower, a wash-basin and a lavatory will be provided near their living compartments for every two members of crew who are not officers: the wash basin can be installed in the living compartments.

SBG-SR8. [SBG,APPENDIX,1.511/(4)]

On ships of 1600 GRT or above, a separate room with a toilet and a wash-basin must be provided in a position near to and accessible from the bridge and the engine room. The same applies to the radio station on ships on which the wireless officers or operators are accommodated in separate living rooms.

SBG-SR9. [SBG,APPENDIX,1.513/(2)]

The drain and outlet pipes must be not run along the ceilings of messes living and provision rooms or galleys and pantries. They must not reach the open air in the vicinity of the suction inlets for the drinking water treatment plant.

SBG-SR10. [SBG,APPENDIX,1.514/(1)]

Lavatory accommodation must have an exhaust ventilator, a washhandl basin and. hygenic arrangement for drying hands.

SBG-SR11. [SBG,APPENDIX,1.514/(2)]

Lavatories must be located separately in the vicinity of living compartments and washrooms. They must be accessible only from passages for general use or from the washrooms.

SBG-SR12. [SBG,APPENDIX,1.52]

The following equipment must be available on ships, depending on the size of the crew: washing machines, appropriate ventilating and heating and lines or other devices for hanging up clothes, ironing equipment.

SBG-SR13. [SBG,APPENDIX,1.8]

Compartments or containers must be provided in which liquid and solid paste can be collected during the time when ship is in harbour.

ILO-SR1. [ILO.No.92.ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-6/13]

Sufficient drainage should be provided .

ILO-SR2. [ILO.No.92.ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-13/1]

Sufficient sanitary accommodations including wash basins and tub and/or shower baths should be provided.

ILO-SR3. [ILO.No.92.ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-13/2]

The following minimum number of separate water closets should be provided.

In ships under 800 tons: 3

In ships under 800 tons or over but under 3000 tons: 4

In ships 3000 tons or over: 6

In ships where the radio officers or operators are accommodated in an isolated position, sanitary facilities should be provided nearby adjacent there to.

ILO-SR4. [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-13/4]

Sanitary facilities for all members of the crew who do not occupy rooms to which private facilities are attached shall be provided for each group of the crew on the following scale.

One tub and/or shower bath for every 8 persons or less.

One water closet for every eight persons or less.

One wash basin for every six persons or less.

ILO-SR5. [ILO.No.92,ABS-CHECK-SHEET-A,REVISION.1,ARTICLE-13/12]

Facilities for washing and drying clothes should be provided on a scale appropriate to the size of the crew and the normal duration of the voyage.

ILO-SR6. [ILO.No.133,ABS-CHECK-SHEET-B,REVISION.1,ARTICLE-9/1]

In ships of 1600 tons or over the following should be provided.

A separate compartment containing a water closet and a wash basin hot and cold running fresh water with an easy access of the navigating bridge deck primarily for those on duty in the area.

A water closet and a wash basin having hot and cold running fresh water within easy access of the machinery space if not fitted near the engine room control centre.

### **B.1.17 Structure**

SOLAS-ST1. [SOLAS'97,CHAPTER-II-1,PART-B,REGULATION-11/2]

Collision bulkhead should be:

i- watertight,

ii- from the forward perpendicular not less than 5% of the length or 10 m whichever the less.

SOLAS-ST2. [SOLAS'97,CHAPTER-II-1,PART-B,REGULATION-11/3]

If there is any part of the ship below the waterline extends forward of the forward perpendicular the distance shall be measured from a point:

i- at the mid length of the extension or

ii- at a distance 15% of the length of the ship forward of the forward perpendicular or

iii- at a distance 3 m. forward of the forward perpendicular which ever gives the smallest value.

SOLAS-ST3. [SOLAS'97,CHAPTER-II-1,PART-B,REGULATION-11/4]

The bulkhead may have steps or recesses provided they are within the limits prescribed in step 1 or 2

SOLAS-ST4. [SOLAS'97,CHAPTER-II-1,PART-B,REGULATION-11/4]

Pipe piercing the collision bulkhead shall be fitted with suitable

i- Valve , operable from above the free board deck

ii- Valve chest shall be secured at the bulkhead inside the forepeak.

SOLAS-ST5. [SOLAS'97,CHAPTER-II-1,PART-B,REGULATION-11/5]

If there is a long forward superstructure is fitted the collision bulkhead shall be extended weathertight to the deck next above the freeboard deck.

SOLAS-ST6. [SOLAS'97,CHAPTER-II-1,PART-B,REGULATION-11/7]

Openings in the extension of the collision bulkhead above the freeboard deck i- shall be restricted to the minimum ii- shall be capable or being closed weathertight.

SOLAS-ST7. [SOLAS'97,CHAPTER-II-1,PART-B,REGULATION-11/8]

Machinery bulkheads shall be watertight up the free board deck.

SOLAS-ST8. [SOLAS'97,CHAPTER-II-1,PART-B,REGULATION-11/9]

Stern tubes shall be enclosed in a watertight space.

SOLAS-ST9. [SOLAS'97,CHAPTER-II-1,PART-B,REGULATION-12-1/1,2,3,4]

If the ship is not a tanker there shall be a DB fitted from the collision bulkhead to the after bulkhead. This shall be practical and compatible with:

i- the design, ii- proper working, iii- the inner bottom shall be continued out to the ship side in such a manner as to protect the bottom to the turn of the bilge, iv- small wells in the DB shall not extend in dept. more than necessary, v- a well extended to the outer bottom may be permitted at the after end of the shaft tunnel of the ship, vi- DB need not be fitted way of watertight compartments..

SOLAS-ST10. [SOLAS'97,CHAPTER-II-1,PART-B,REGULATION-14/1]

If a bulkhead is watertight each watertight subdivision bulkhead shall be constructed to be capable of supporting with a proper margin of resistance.

i- the pressure due to the max. head of water (in damage as well)

ii- at least the pressure due to a head of water up to the margin line

SOLAS-ST11. [SOLAS'97,CHAPTER-II-1,PART-B,REGULATION-14/2.1]

Steps and recesses in bulkheads shall be watertight.

SOLAS-ST12. [SOLAS'97,CHAPTER-II-1,PART-B,REGULATION-14/2.2]

If a frame or beam passes through a watertight deck or bulkhead such a deck

or bulkhead shall be made structurally watertight without wood or cement

SOLAS-ST13. [SOLAS'97,CHAPTER-II-2,PART-A,REGULATION-3/3.1,3.3,3.4]

If the division is "A" class:

i- Shall be steel or equipment.

ii- Shall prevent passage of smoke and flame to the end of 1 hour of the standard fire test,

iii- Shall be insulated with non-combustible materials.

SOLAS-ST14. [SOLAS'97,CHAPTER-II-2,PART-A,REGULATION-3/4.1,4.3]

If the division is "B" class:

i- shall prevent the passage of flame to the end of ½ hour of the standard fire test.

ii- shall be non-combustible

SOLAS-ST15. [SOLAS'97,CHAPTER-II-2,PART-A,REGULATION-3/5]

If the division is "C" class:

i- shall be non-combustible

#### **B.1.18 Emergency**

SOLAS-E1. [SOLAS'97,CHAPTER-II-1,PART-D,REGULATION-43/1.2]

There shall be the emergency source of electrical power

i) Shall be located above the uppermost continuous deck and

- ii) Shall be accessible from the open deck
- iii) Shall not be located forward of the collision bulkhead.

#### **B.1.19 Life Saving**

SOLAS-LS1. [SOLAS'97,CHAPTER-III,PART-B,REGULATION-7/1.2]

At least one lifebuoy on each side of the ship shall be fitted with a buoy and lifeline complying with the requirements of reg. 31.4 equal in length at which it is stowed above the waterline in the lightest seagoing condition or 30 m. which ever is the greater.

SOLAS-LS2. [SOLAS'97,CHAPTER-III,PART-B,REGULATION-7/2.1]

A lifejacket shall be provided for every person on board the ship

- i) a number of lifejackets suitable for children equal at least %10 of the num. of the num. of passengers.

SOLAS-LS3. [SOLAS'97,CHAPTER-III,PART-B,REGULATION-7/3.1]

An immersion suit shall be provided for every person assigned to crew the rescue boat.

SOLAS-LS4. [SOLAS'97,CHAPTER-III,PART-B,REGULATION-11/1]

Lifeboats and liferafts for which approved launching appliances are required shall be stowed as close to accommodation and service spaces as possible.

SOLAS-LS5. [SOLAS'97,CHAPTER-III,PART-B,REGULATION-11/2]

Muster station shall be provided close to the embarkation stations. Each muster station shall have sufficient space to accommodate all persons assigned to muster at that station.

SOLAS-LS6. [SOLAS'97,CHAPTER-III,PART-B,REGULATION-12]

Launching station shall be placed with a clearance from the propeller and as far as possible survival craft, except survival craft specially designed for free fall launching can be launched down the straight side of the ship if positioned forward they shall be located abaft the collision bulkhead in a sheltered position.

SOLAS-LS7. [SOLAS'97,CHAPTER-III,PART-B,REGULATION-13/1]

Such survival craft shall be stowed

- i- So that neither the survival craft nor its stowage arrangements will interface with the operation any other survival craft or rescue boat at any other launching station.

- ii- As near the water surface as is safe and practicable and, in the case of a survival craft other than a liferaft indented for throw overboard launching, in such a position,

that the survival craft. In the embarkation. Position in not less than 2 m. above the waterline with the ship full loaded and trimmed and listed up to  $20^0$  or to the angle which the watertight edge become submerged, whichever is less.

iii- 2 crew can carry out preparations for embarkation and launching in less than 5 min.

SOLAS-LS8. [SOLAS'97,CHAPTER-III,PART-B,REGULATION-13/2]

On cargo ships  $30 < L < 120$  m. each lifeboat shall be stowed that the after end of life boat is not less than the length of the life boat forward of the propeller. Ships  $L > 120$  m. in length each lifeboat shall be stowed that the after end of lifeboat is not less than 1,5 times the length of the Lb. Forward of the propeller.

SOLAS-LS9. [SOLAS'97,CHAPTER-III,PART-B,REGULATION-13/6]

Liferafts intended for throw-overboard launching shall be so towed as to be readily transferable for launching on other side of the ship unless liferafts to be capable of being launched on either side are stowed on each side of the ship.

SOLAS-LS10. [SOLAS'97,CHAPTER-III,PART-B,REGULATION-14]

Rescue boats shall be stowed

i) in a state of continuous readiness for launching is not more than 5 min.

ii) In a position suitable for launching and recovery

iii) So that neither the rescue boat nor its stowage arrangements will interface with the operation of any survival craft at any other launching station.

SOLAS-LS11. [SOLAS'97,CHAPTER-III,PART-B,REGULATION-15/2]

Each lifeboat shall be provided with an appliance which is capable of launching and recovering the lifeboat.

SOLAS-LS12. [SOLAS'97,CHAPTER-III,PART-B,REGULATION-17]

A line-throwing appliance complying shall be provided.

SOLAS-LS13. [SOLAS'97,CHAPTER-III,PART-B,REGULATION-26/1.1.1.1]

Cargo ships shall carry

i- one or more lifeboats such aggregate capacity on each side of the ship as will accommodate the total numbers of persons on board.

ii- In addition a liferaft or liferafts, capable of being launched on either side of the ship and of such aggregate capacity as will accommodate the total number of persons onboard.

SOLAS-LS14. [SOLAS'97,CHAPTER-III,PART-B,REGULATION-26/1.4]

If the survival craft are stowed far. from the stem or stern for more than 100 m. the ship shall carry an extra liferaft on the other side.

SOLAS-LS15. [SOLAS'97,CHAPTER-III,PART-B,REGULATION-26/2]

Cargo ships shall carry at least one rescue boat. A lifeboat may be accepted as a rescue boat, provided that is also complies with the requirements for a rescue boat.

SOLAS-LS16. [SOLAS'97,CHAPTER-III,PART-B,REGULATION-27/1.1]

Cargo ships shall carry not less than the number of lifebuoys prescribed in the following table.

|           | min |
|-----------|-----|
| L<100     | 8   |
| 100<L<150 | 10  |
| 150<L<200 | 12  |
| L>200     | 14  |

SOLAS-LS17. [SOLAS'97,CHAPTER-III,PART-B,REGULATION-27/3.3]

If ship has a

i- davit-launched liferafts,

ii- has liferafts served by equivalent approved appliances capable of being used on both sides of the ship and which do not require entry into the water to board the liferaft or.

iii- Is constantly engaged on voyages in warm climates there shall not be immersion suits.

SOLAS-LS18. [SOLAS'97,CHAPTER-III,PART-C,REGULATION-31/1]

Every lifebuoy shall:

i) have an outer diameter of not less than 800 mm. and an inner diameter of not less than 400 mm.;

ii) have a mass of not less than 2.5 kg.

SOLAS-LS19. [SOLAS'97,CHAPTER-III,PART-C,REGULATION-3.2]

The liferaft shall be fitted with an efficient painter of length equal to not less than twice the distance from the stowed position to the waterline in the lightest seagoing condition or 15 m. whichever is the greater.



## **APPENDIX C. KNOWLEDGE ELICITATION (INTERVIEW)**

1. How do you determine the main dimensions?
2. How do you decide the number of container bays, rows and tiers under deck?
3. How do you calculate the length of a cargo hold?
4. How do you calculate the distance between two cargo holds? How do you use this void?
5. How do you calculate the minimum required engine power? Usually which theory do you use for the calculation?
6. How do you determine the location, length and height of an engine room?
7. How do you calculate minimum required distance for a stern tube? How do you decide the location of the stern tube?
8. How do you decide the breadth of side tanks?
9. How do you decide the location of tanks?
10. How do you decide the distance between a container and a side tank?
11. How do you determine the distance between a container tier and a hatch cover?
12. How do you calculate the depth of hatch covers?
13. How do you determine the maximum length of a hatch cover?
14. How do you determine the width of hatch cover?
15. How do you decide the hatch coaming height?
16. How do you decide the weather deck general arrangement?
17. How do you decide the location of lifts? If there is any.
18. How do you decide the number of container bays, rows and tiers above deck?
19. If lifts are located between cargo holds, how do you calculate the minimum required distance for lift foot?
20. How do you decide container locations on cargo hold covers?
21. How do determine the bridge height and the number of floor according to crew number?

22. How do you start accommodation area design?
23. How do you decide the shape of passageways?
24. How do you locate staircases?
25. How do you decide the type of staircase?
26. Which living compartment can be located first?



## **Appendix D. DELPHI**

User interface, graphic visualisation of layout and output functions of ALDES have been programmed in Delphi programming environment. Main tasks performed in Delphi are given as follows:

- Input of initial case (TEU, Speed, Range),
- Assignment of defaults, input of user preferences with a windows based user interface,
- Running of expert system developed in CLIPS by use of CLIPS DLL,
- Displaying the results of expert system modules both in tabular and graphic environment,
- User interaction of graphic attributes (size, position and rotation) of the objects,
- Visualisation and interactive display of layout, and
- Export of layout into a CAD program (e.g. AUTOCAD).

### **D.1 Delphi**

DELPHI is a visual programming environment on MS-Windows 95/98/2000/NT family operating system. It is an object oriented PASCAL language with graphic functions. Object Oriented Programming (OOP) offers following capabilities for knowledge based design:

- Objects: A data type that includes both attributes (properties) and methods associated with these attributes. An example of an object can be given as a room, which has length, width, and depth attributes and has a method to calculate volume attribute of the room.

- **Classes:** A Class is the abstract description of a group of objects, each with a specific state but all capable of performing some operations. Objects are instances of classes. Classes have attribute fields represented by frames and methods both generative and interpretative knowledge
- **Inheritance:** Inheritance is of the key issues of OOP and indicates that a new class, defined as a subclass of an existing parent class, inherits all attribute fields and methods of its parent class. Inheritance can be used to express generalisation and represents “is a” relation. Hence, if a generic model is created as a class, the subclass can inherit generic model and can be failed for specific purpose.
- **Polymorphism:** Polymorphism indicates the ability of a class, belong to a parent class, to change its properties and its methods inherited from the parent. Polymorphism and inheritance can form very powerful mechanism for design from prototypes.
- **Encapsulation:** This feature indicates that an object can hide some of attributes and methods from the rest of the program.

## **D.2 Data Representation in ALDES**

A class hierarchy has been established as shown in Figure D.1. Root Class named ship object has been defined as a subclass of graphic-central of DELPHI. This ensures to inherit basic object properties. Ship object has properties length, width, depth, position about x, y, z axes, rotation about x, y, z axes, weight, size etc. and methods to plan itself on profile, plan and cross section view, move and resize.

All objects in the ship, e.g. decks, rooms, bulkheads, doors, furniture etc., were instances of classes derived from ship object. All of these subclasses have a parameter called owner, which indicates which object owns the current class. Figure D.2 shows ownership relation. This feature enables ALDES to move all furniture windows, doors and walls with the room.

- **Classes:** A Class is the abstract description of a group of objects, each with a specific state but all capable of performing some operations. Objects are instances of classes. Classes have attribute fields represented by frames and methods both generative and interpretative knowledge
- **Inheritance:** Inheritance is of the key issues of OOP and indicates that a new class, defined as a subclass of an existing parent class, inherits all attribute fields and methods of its parent class. Inheritance can be used to express generalisation and represents “is a” relation. Hence, if a generic model is created as a class, the subclass can inherit generic model and can be failed for specific purpose.
- **Polymorphism:** Polymorphism indicates the ability of a class, belong to a parent class, to change its properties and its methods inherited from the parent. Polymorphism and inheritance can form very powerful mechanism for design from prototypes.
- **Encapsulation:** This feature indicates that an object can hide some of attributes and methods from the rest of the program.

## **D.2 Data Representation in ALDES**

A class hierarchy has been established as shown in Figure D.1. Root Class named ship object has been defined as a subclass of graphic-central of DELPHI. This ensures to inherit basic object properties. Ship object has properties length, width, depth, position about x, y, z axes, rotation about x, y, z axes, weight, size etc. and methods to plan itself on profile, plan and cross section view, move and resize.

All objects in the ship, e.g. decks, rooms, bulkheads, doors, furniture etc., were instances of classes derived from ship object. All of these subclasses have a parameter called owner, which indicates which object owns the current class. Figure D.2 shows ownership relation. This feature enables ALDES to move all furniture windows, doors and walls with the room.

- **Classes:** A Class is the abstract description of a group of objects, each with a specific state but all capable of performing some operations. Objects are instances of classes. Classes have attribute fields represented by frames and methods both generative and interpretative knowledge
- **Inheritance:** Inheritance is of the key issues of OOP and indicates that a new class, defined as a subclass of an existing parent class, inherits all attribute fields and methods of its parent class. Inheritance can be used to express generalisation and represents “is a” relation. Hence, if a generic model is created as a class, the subclass can inherit generic model and can be failed for specific purpose.
- **Polymorphism:** Polymorphism indicates the ability of a class, belong to a parent class, to change its properties and its methods inherited from the parent. Polymorphism and inheritance can form very powerful mechanism for design from prototypes.
- **Encapsulation:** This feature indicates that an object can hide some of attributes and methods from the rest of the program.

## **D.2 Data Representation in ALDES**

A class hierarchy has been established as shown in Figure D.1. Root Class named ship object has been defined as a subclass of graphic-central of DELPHI. This ensures to inherit basic object properties. Ship object has properties length, width, depth, position about x, y, z axes, rotation about x, y, z axes, weight, size etc. and methods to plan itself on profile, plan and cross section view, move and resize.

All objects in the ship, e.g. decks, rooms, bulkheads, doors, furniture etc., were instances of classes derived from ship object. All of these subclasses have a parameter called owner, which indicates which object owns the current class. Figure D.2 shows ownership relation. This feature enables ALDES to move all furniture windows, doors and walls with the room.