The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author’s permission.

L’auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L’auteur conserve la propriété du droit d’auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.
APPLICATION OF ECOLOGICAL INTERFACE DESIGN TO AVIATION

Master of Applied Science, 1997

Nick Dinadis

Department of Mechanical and Industrial Engineering

University of Toronto

ABSTRACT

The objective of this thesis is to determine the extent to which previous research on Ecological Interface Design (EID), conducted in the context of process control, can be adapted to design more effective status displays for engineering systems in aviation cockpits. The primary outcome of this research is a prototype EID interface for the engineering systems of Hercules C-130 Model E-H aircraft. This thesis includes the currently available literature on aircraft status display design, principles of EID, previous EID research conducted by the Cognitive Engineering Laboratory at the University of Toronto, two field studies observing Canadian Armed Forces (CAF) Hercules crews, analysis and design of the prototype EID interface, an evaluation of the prototype conducted with CAF Hercules C-130 Flight Engineers, and a comparison of the operating philosophies inherent to EID and those currently existing in the aviation industry. The results of this study provide a proof of concept showing that: a) the principles of EID can be meaningfully applied to the aircraft engineering systems of a Hercules C-130; b) EID needs to be supplemented by more specific design principles; c) it is possible to effectively integrate EID with these other design principles. In conclusion, EID has been shown to be a viable candidate for the design of engineering systems status displays within the aviation industry.
ACKNOWLEDGMENTS

This research was sponsored by a grant from the Defence and Civil Institute of Environmental Medicine. The Natural Sciences and Engineering Research Council of Canada also contributed through research equipment grants. Much gratitude is owed to Professor Kim J. Vicente, whose wisdom and guidance improved the quality of both this document and the prototype EID interface described herein. Contributors whom I would like to acknowledge include the contract monitor David Beevis, the Hercules C-130 Flight Engineers of the Canadian Armed Forces (particularly Sgt. Gary Wall), Edward Panyan, Kit-Man Cheung, Cathy Burns and the other members of the Cognitive Engineering Laboratory, Professor Paul Milgram, Professor Julius Grodski and Virtual Prototypes Inc. Finally, I would like to thank Kelly, Jim and Chris, and my parents, whose unwavering patience and support contributed in so many ways. Thank you all.
# TABLE OF CONTENTS

Abstract .............................................................................................................................. ii

Acknowledgments ........................................................................................................... iii

Table of Contents ........................................................................................................... iv

List of Tables .................................................................................................................... v

List of Figures .................................................................................................................. vi

List of Appendices .......................................................................................................... x

List of Abbreviations ...................................................................................................... xi

Chapter 1: Introduction .................................................................................................... 1

Chapter 2: Status Displays for Engineering Subsystems in Aviation Cockpits - A
Literature Review ........................................................................................................... 5

Chapter 3: Observations of Flight Engineers During Refueling Missions ..................... 35

Chapter 4: Protocol Analysis of a Hercules C-130 Crew During a Simulator Session .... 55

Chapter 5: Analysis of C-130 Engineering Systems ....................................................... 68

Chapter 6: Interface Description ..................................................................................... 103

Chapter 7: Interface Evaluation ....................................................................................... 152

Chapter 8: Impact of the Operating Philosophy on the Design of Interfaces ............... 159

Chapter 9: Conclusions .................................................................................................. 171

Chapter 10: References .................................................................................................. 176

Appendices ...................................................................................................................... 183
LIST OF TABLES

Table 1: Layout of the status gauge panel .............................................................. 42
Table 2: List of all the faults which were to occur during the LP3 session ........... 54
Table 3: List of events including approximate time of initiation ....................... 57
LIST OF FIGURES

Figure 1: Photograph of a Hercules C-130 Aircraft, Canadian Armed Forces Number 332 ......................................................... 35

Figure 2: View of a Typical Hercules C-130 Cockpit .............................................. 41

Figure 3: Close-up of the Engine Status Gauges .................................................. 42

Figure 4: View Of The Overhead Instrumentation Panel ........................................ 45

Figure 5: Diagram of the Overhead Instrumentation Panel ........................................ 46

Figure 6: Copy of a Hercules Flight Engineer Log .................................................... 49

Figure 7: External Tank Pump Has Been Marked As Being Unreliable .................. 51

Figure 8: Representation of the AES According to Part-whole and Means-ends Hierarchies .................................................. 69

Figure 9: Example of the Component Level in the Part-Whole Hierarchy .............. 71

Figure 10: Example of the Subsystem Level Within the Part-Whole Hierarchy ......... 73

Figure 11: Views E1 and F1 According to Part-Whole and Means Ends Hierarchies .... 74

Figure 12: Abstract Function - System Level in the Part-whole Means-ends Hierarchy for the Engine System ................................................................. 76

Figure 13: Abstract Function - Subsystem Level in the Part-whole Means-ends Hierarchy for the Engine System ................................................................. 77

Figure 14: Abstract Function - System Level in the Part-whole Means-ends Hierarchy for the Fuel System ................................................................. 78

Figure 15: Abstract Function - Subsystem Level in the Part-whole Means-ends Hierarchy
Figure 16: Abstract Function - Unit Level in the Part-whole Means-ends Hierarchy for the Fuel System ................................................................. 80

Figure 17: Generalized Function - Subsystem Level in the Part-whole Means-ends Hierarchy for the Engine System ........................................ 83

Figure 18: Generalized Function - Unit Level in the Part-whole Means-ends Hierarchy for the Engine System .................................................. 84

Figure 19: Generalized Function - Sub-unit Level in the Part-whole Means-ends Hierarchy for the Engine System ........................................ 85

Figure 20: Generalized Function - Component Level in the Part-whole Means-ends Hierarchy for the Engine Oil System ................................. 87

Figure 21: Generalized Function - Component Level in the Part-whole Means-ends Hierarchy for the Engine Fuel System ........................................... 88

Figure 22: Generalized Function - Unit Level in the Part-whole Means-ends Hierarchy for the Fuel System ................................................................. 89

Figure 23: Generalized Function - Sub-unit Level in the Part-whole Means-ends Hierarchy for the Fuel System .................................................. 90

Figure 24: Generalized Function - Component Level in the Part-whole Means-ends Hierarchy for the Fuel System ............................................ 92

Figure 25: Link Between Generalized Function and Physical Function Levels in the Part-whole Means-ends Hierarchy for the Engine System ........... 93

Figure 26: Link Between Generalized Function and Physical Function Levels in the Part-
Figure 45: Example of Low Fuel Flow to Engine #2 ......................................................... 139
Figure 46: Propeller De-Couples From the Engine .......................................................... 141
Figure 47a: Fuel System Completely Balanced ............................................................... 143
Figure 47b: Running All Engines Off of Tank One Only ................................................. 144
Figure 47c: Mistakenly Compensating with Tank 2 Booster Pumps Only (After Period of
Time) ................................................................................................................................. 145
Figure 47d: Situation Left to Worsen - Fuel System Completely Out of Balance .......... 146
LIST OF APPENDICES

Appendix A: FE Protocol ................................................................. A-1
Appendix B: Hercules Flight Engineer Log ................................. B-1
Appendix C: Aircraft Specifications ............................................. C-1
Appendix D: Instructions on How to Run Prototype .................... D-1
Appendix E: List of Files Used in Creation of Prototype .............. E-1
Appendix F: VAPS® ATN for Interface Application ....................... F-1
Appendix G: List of VAPS® Integration Editor Connections for Each File Used in Interface ......................................................... G-1


**LIST OF ABBREVIATIONS**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A/C</td>
<td>Aircraft</td>
</tr>
<tr>
<td>AC</td>
<td>Aircraft Commander (Pilot)</td>
</tr>
<tr>
<td>ACT</td>
<td>Aircrew Coordination Training</td>
</tr>
<tr>
<td>AGL</td>
<td>Above Ground Level</td>
</tr>
<tr>
<td>AH</td>
<td>Abstraction Hierarchy</td>
</tr>
<tr>
<td>ASI</td>
<td>Attitude Situation Indicator</td>
</tr>
<tr>
<td>ASRS</td>
<td>Aviation Safety and Reporting Service</td>
</tr>
<tr>
<td>ATN</td>
<td>Action Transition Network</td>
</tr>
<tr>
<td>CAF</td>
<td>Canadian Armed Forces</td>
</tr>
<tr>
<td>CB</td>
<td>Circuit Breaker</td>
</tr>
<tr>
<td>CEL</td>
<td>Cognitive Engineering Laboratory</td>
</tr>
<tr>
<td>CFB</td>
<td>Canadian Forces Base</td>
</tr>
<tr>
<td>CRT</td>
<td>Cathode Ray Tube</td>
</tr>
<tr>
<td>DCIEM</td>
<td>Defense and Civil Institute of Environmental Medicine</td>
</tr>
<tr>
<td>EICAS</td>
<td>Engine Indication and Crew Alerting System</td>
</tr>
<tr>
<td>EID</td>
<td>Ecological Interface Design</td>
</tr>
<tr>
<td>E-M</td>
<td>Electro-mechanical</td>
</tr>
<tr>
<td>E-MACS</td>
<td>Engine Monitoring and Control Systems</td>
</tr>
<tr>
<td>EOP</td>
<td>Emergency Operating Procedures</td>
</tr>
<tr>
<td>EPR</td>
<td>Engine Pressure Ratio</td>
</tr>
<tr>
<td>FCU</td>
<td>Fuel Control Unit</td>
</tr>
<tr>
<td>FE</td>
<td>Flight Engineer</td>
</tr>
<tr>
<td>FO</td>
<td>First Officer (Co-Pilot)</td>
</tr>
<tr>
<td>gal.</td>
<td>Gallons</td>
</tr>
<tr>
<td>HSCT</td>
<td>High Speed Civil Transport</td>
</tr>
<tr>
<td>IMC</td>
<td>International Meteorological Conditions</td>
</tr>
<tr>
<td>IFF</td>
<td>Indication Friend or Foe</td>
</tr>
<tr>
<td>IFR</td>
<td>Instrument Flight Rules</td>
</tr>
<tr>
<td>KIAS</td>
<td>Knots Indicated Airspeed</td>
</tr>
<tr>
<td>LP</td>
<td>Lesson Plan</td>
</tr>
<tr>
<td>NTSB</td>
<td>National Transportation Safety Board</td>
</tr>
<tr>
<td>P&amp;ID</td>
<td>Piping and Instrumentation Diagram</td>
</tr>
<tr>
<td>psi</td>
<td>Pounds per Square Inch</td>
</tr>
<tr>
<td>RPM</td>
<td>Revolutions Per Minute</td>
</tr>
<tr>
<td>SOP</td>
<td>Standard Operating Procedures</td>
</tr>
<tr>
<td>SRK</td>
<td>Skills, Rules, Knowledge</td>
</tr>
<tr>
<td>SSSI</td>
<td>Single-Sensor Single-Indicator</td>
</tr>
<tr>
<td>TD</td>
<td>Temperature Datum</td>
</tr>
<tr>
<td>TIT</td>
<td>Turbine Inlet Temperature</td>
</tr>
<tr>
<td>U.S.</td>
<td>United States</td>
</tr>
<tr>
<td>U/S</td>
<td>Unserviceable</td>
</tr>
<tr>
<td>V1</td>
<td>Minimum Take-off Velocity</td>
</tr>
</tbody>
</table>
CHAPTER 1: INTRODUCTION

1.1. INTRODUCTION

The objective of this thesis is to determine the extent to which previous research on Ecological Interface Design (EID), conducted in the context of process control (see Vicente, in press), can be adapted to design more effective status displays for engineering systems in aviation cockpits. The engineering systems of a Lockheed Hercules C-130 Model E-H aircraft were used as a testbed.

1.1.1 Problem Description

Aviation status displays have traditionally been designed to present crews with physical information describing the status of individual components and subsystems, primarily via electro-mechanical (E-M) gauges (Stokes & Wickens, 1988). More recently, research conducted on semantic display design issues have concentrated on the development of pictorial versions of traditional displays (Baty, 1976; Reising and Emerson, 1982). The limitations of such designs are well known and documented (Goodstein, 1981; Woods, 1991). EID was proposed by Vicente and Rasmussen (1992) to address some of these limitations, by integrating different kinds of representations in a single interface. EID suggests that the interface should contain higher order functional information (e.g., system purposes, and first principles) as well as the traditional lower order physical information (e.g. component settings). It is based on two seminal concepts from cognitive engineering research: The Abstraction Hierarchy (AH); and the Skills, Rules, Knowledge (SRK) framework (Rasmussen, 1986). Incorporating the principles of
EID can help operators in determining whether or not the system is achieving its intended purposes and functions, and in selecting the appropriate actions required to achieve these purposes.

EID has been empirically evaluated in a number of experiments at the Cognitive Engineering Laboratory (CEL) at the University of Toronto, in the context of a thermal-hydraulic process simulation (Vicente, Christoffersen, & Pereklita, 1995; Pawlak & Vicente, 1996; Christoffersen, Hunter, & Vicente, 1994). These systems, although smaller in scope, are similar to the fuel, hydraulic, and engine systems that must be monitored by crews in the aviation domain. Thus, the feasibility of the EID approach in aviation shall concentrate on the development of displays for such systems. The goal of the project was to determine whether previous research on EID could be leveraged to develop new display design concepts for aviation cockpits that go beyond the limitations of traditional E-M, and more recent pictorial displays.

1.1.2 Approach

A number of design methodologies, aside from EID, have been proposed to address the aviation related issues described above, such as: emergent feature displays (Bennett & Flach, 1992; Berringer & Chrisman, 1991); integrated displays (Calhoun & Herron, 1981); pictorial displays (Reising & Emerson, 1982; Way et al., 1990); and task-oriented displays (Abbott, 1989, 1990; Abbott & Palmer, 1989). These proposals will be reviewed in this thesis. They have been chosen because they are a representative sample of the publicly available work on aviation status display design that utilize graphical tools to present the status of engineering systems in cockpits. The EID approach and its
principles are presented within this thesis, and the reasons why EID is relevant to the aviation domain are also outlined.

In addition to reviewing the literature, two field studies were conducted with Canadian Armed Forces Hercules crews. Emphasis was placed on observing the Flight Engineers (FEs) to obtain a better understanding of the intricacies involved in their daily duties. The first field study took place between Alert, Canada and Thule, Greenland. The follow-up study involved observing FEs during simulator sessions, at Canadian Forces Base (CFB) Trenton. Both of these field studies have been included in this thesis, and examine issues such as: the role of the FE; the information available to the FEs and how they have adapted to, and modified, this information. Display design recommendations which could be used to benefit FEs of traditional Hercules C-130 aircraft are also provided.

EID was used to analyze the engine and fuel systems of Lockheed Hercules C-130 models E-H aircraft, which are currently in service in the Canadian Forces. An EID interface was designed based on the knowledge elicited from the literature review and the field studies, as well as the AH analysis of the systems, using "VAPS" on a SGI Indigo machine.

1.1.3 Outline

The thesis is organized as follows: first, the available literature directly relevant to the design of engineering status displays is reviewed; a chapter is then devoted to each of the field studies performed; the next chapter focuses on the AH representations that were developed to describe the various engineering systems, namely the Fuel and Engine
Systems (containing oil, hydraulics etc.). These representations were subsequently used to identify the information content and structure of the interface, which were incorporated into the design of the interface; next, the design itself, along with the rationale behind it, as well as an informal evaluation will be presented; the thesis then concludes with a summary of the lessons learned from this feasibility study, with implications for the design of glass cockpit displays for modern aircraft.
CHAPTER 2: STATUS DISPLAYS FOR ENGINEERING SUBSYSTEMS IN AVIATION COCKPITS - A LITERATURE REVIEW

2.1. PREFACE

This chapter describes the results of a review of the literature on status displays for engineering systems in aviation cockpits in order to determine if EID has a unique contribution to make to this area. The remainder of this chapter is organized as follows. First, the scope of the review and the sources examined will be defined. Second, literature relevant to the design of status displays for engineering subsystems will be reviewed. Third, previous research on EID will be summarized.

2.2. SCOPE OF REVIEW

2.2.1 Focus

There is a great deal of literature on aviation cockpit display design. This review focuses specifically on cognitive and perceptual factors that impact human performance in using visual displays for determining the status of engineering subsystems in commercial and military transport cockpits. There are many other related areas of the literature that were specifically excluded from consideration because they are less relevant to the aims of this thesis. For example, there is a body of research specifically on the use of colour in displays to match human physiological and perceptual capabilities and limitations (e.g., Stokes, Wickens, & Kite, 1990, chapter 7; Umbers & Collier, 1990). Although colour is an option for coding displays developed in the context of this contract, this thesis will not
specifically investigate the effects of various types of colour coding on human performance. Similarly, the research on display legibility and readability, investigating the interaction between glare, high intensity illumination, night conditions, and various characteristics of display hardware (e.g., Tannas, 1985; Wisler, Doyle, & Giuglianotti, 1994) will not be considered for analogous reasons. There is also a growing literature on the design of head-up displays (HUDs) (e.g., Stokes, et al., 1990, chapter 8; Adam, 1994). However, HUDs are usually used to display navigational and primary flight control information, not information about the status of engineering subsystems. Thus, the HUD literature was not reviewed either.

2.2.2 Sources Consulted

The following sources were consulted as reference material for the review: textbooks, conference proceedings, journal articles, aviation magazines, and technical reports. These written documents were complemented by conversations held with several experts in the field. Most of these conversations took place at the Aviation Psychology Symposium held in Columbus, Ohio in April, 1995. The following people were consulted: Barbara Barnett (McDonnell Douglas), Dennis Berringer (FAA), Steve Casner (NASA Ames), Sherry Chappell (NASA Ames), Asaf Degani (NASA Ames), Bill Jones (Delta Airlines), Gary Klein (Klein Associates), Kathy Mosier (NASA Ames), Ev Palmer (NASA Ames), Mike Palmer (NASA Langley), John Reising (Wright Patterson AFB), Vic Riley (Honeywell), Bill Rogers (BBN Laboratories), Marianne Rudisill (NASA Langley), Nadine Sarter (Ohio State University), Paul Schutte (NASA Langley), John Wise (Embry Riddle University), and David Woods (Ohio State University). These conversations
helped us in conceptualizing the various factors that are important to this thesis, and in developing leads to specific references. Finally, we also spoke with two simulator trainers, a pilot, and one simulator technician at Air Canada (Jack Smith, J. C. Bergeron, and Lyle Savel, respectively), and observed a number of simulator sessions on a Boeing 767. These sessions provided us with rudimentary insight into the design of existing status displays for several cockpits (B-767, B-747, A-320, B-737), and a better understanding of the operating philosophy (e.g., role of the human in the system, approach to training, reliance on procedures, etc.) of one commercial airline.

2.3. Status Display for Engineering Systems

The bulk of visual display design research within the aviation domain has focused on navigation and attitude displays (e.g., Baty, 1976; Clay, 1993). Thus, there is very little research that we could find explicitly on status displays for engineering subsystems. (There is probably more research than we found, but much of it has probably been conducted by airframe manufacturers and is therefore not easily accessible in the open literature or is proprietary in nature). This section reviews the relevant research we found on emergent features displays, pictorial displays, and task-oriented displays.

2.3.1 Emergent Feature Displays

Emergent features displays comprise a growing body of research (see Bennett & Flach, 1992) that attempts to address one of the major limitations of traditional display design, namely that only low-level data are usually presented (Goodstein, 1981). As a
result, operators are forced to integrate these data to extract the information required to answer higher-level questions that are closer to system goals. Emergent features displays unburden the operator of this cognitively demanding task. This is accomplished by showing the status of high-level system properties directly, in addition to displaying low-level data. This principle is best illustrated by an example, such as the well-known octagon or polar star display (Coekin, 1969). This display represents the status of eight lower level system variables describing the status of a particular system or subsystem. The value of each variable is represented as the length of a spoke in the octagon. The variables are normalized such that when they are all at their expected values, their lengths are the same. The ends of the spokes are connected by thin lines. When all eight variables are in their expected state, a symmetrical octagon is formed. Thus, an emergent feature (symmetry) is mapped onto a higher-order system property (healthy system). When the system is not in its normal state, the emergent feature is violated, indicating a problem in a salient fashion. Moreover, the way in which the octagon deforms is intended to provide information as to the nature of the problem. With such displays, operators do not have to mentally integrate individual data in order to determine the status of a higher-order function; instead, they can see the status of the function directly.

Emergent feature displays have been applied to a wide variety of applications, including nuclear power plants, thermal-hydraulic processes, and primary flight displays in aviation (Bennett & Flach, 1992). They would seem to be of obvious relevance to the design of status displays for engineering subsystems. However, only two papers that empirically investigate this concept in an aircraft system context were found. Berringer
and Chrisman (1991) integrated an octagon display with a flight-attitude display so that the flight display was in the center of the octagon display. The rationale was that pilots could use their foveal vision to acquire information from the flight-attitude display, while simultaneously monitoring the octagon display for abnormal deviations using their peripheral vision. Unfortunately, the data presented in the octagon display did not represent a meaningful set of parameters. Instead, this display was used to represent a context-free (i.e. generic) set of variables that subjects were required to monitor. Four different versions of the octagon display were compared over two experiments. The subjects, who were not pilots, were required to perform a tracking (flight) task, detect failures in the peripheral octagon display, and perform a digit canceling task that was used as a measure of effort expended on the other two tasks. Abnormalities were introduced every four seconds, and each trial lasted two minutes. Berringer and Chrisman (1991) state that "the results suggest that order of merit of displays may be altered by small format changes and that the performance benefits of lower-graphic iconic displays stem from factors beyond the simple physical proximity of the component indicators" (p. 133). In short, they were unable to pinpoint the factors which caused one polar start display to outperform another.

In any case, the results obtained in these experiments are of questionable external validity since the conditions under which the data were collected are not at all representative of the situations in operational settings. First, the variables displayed in the octagon display were arbitrary. The way in which these variables would be monitored in an actual cockpit would surely be influenced by the significance and relative priority of the
displayed variables. Such influences were not captured in this study. Second, abnormalities were introduced at the rate of one every four seconds, which is far more frequent than one would expect to confront in an actual flight. Third, the trials were of a very brief duration. These last two factors would tend to cause subjects to monitor the peripheral display much more closely than they would in the representative case where nothing goes wrong during the vast majority of the time. Finally, the subjects were not pilots and were not given extensive experience at the tasks that they were required to perform. Thus, it is not known to what extent the results obtained can be generalized to experienced professionals. Thus, there are no definitive, generalizable design implications that one can extract from this research for the purposes of the present work.

Calhoun and Herron (1981) also conducted a study of emergent feature displays (although they used the term “integrated display”). They compared three different displays for engine and hydraulic subsystems in an A-7D fixed-base, single-seat cockpit simulator. The first display consisted of a set of traditional electromechanical circular gauges. Alarm limits were indicated on each individual gauge with coloured tape. The second display integrated the relevant information into a set of deviation bar graphs on a cathode ray tube (CRT) display. The bar graphs were normalized with respect to the expected values for each parameter. In addition, alarm limits were explicitly shown on the same scale and were coded with shading and flashing. The third display was similar to the second except that it used colour to indicate system state, including alarms. The flight displays were identical for the three display conditions. Eighteen A-7D pilots served as subjects in the experiment in a within-subjects design. They were required to maintain
flight control and report engine parameter failures when they occurred. The results indicate that the integrated displays allowed pilots to identify engine parameter abnormalities faster and more accurately than the conventional display set. The use of colour produced no significant effects, although pilots preferred the colour version of the integrated display. These findings are of particular interest, when one considers that they were obtained under very representative conditions.

In summary, the study conducted by Calhoun and Herron (1981) and research conducted in domains outside of aviation suggest that emergent features can be of benefit compared to more traditional display design practices when designing status displays for engineering subsystems.

2.3.2 Pictorial Displays

Pictorial displays, which depict variables in a realistic manner, have been used extensively in aviation (Stokes & Wickens, 1988), more so than in other domains, such as process control. This probably stems from the fact that in aviation, many of the quantities of interest have a meaningful physical referent (e.g., the view out of the cockpit, highways in the sky), whereas those in other domains frequently do not (e.g., energy, radioactivity, pH). Thus, it is not surprising to find that there has been a sustained effort to develop status displays for engineering subsystems based on pictorial formats.

A multi-year research program conducted by Wright Patterson AFB, in cooperation with Boeing and McDonnell Douglas, has examined the use of pictorial displays in a military context. One of the earlier efforts in this program was conducted by Reising and Emerson (1982) who describe pictorial status displays for the fuel, hydraulics, engines,
and electrical engineering subsystems. These displays are intended to replace traditional electromechanical gauges and earlier text-based CRT displays. Way et al. (1990) describe some later research conducted under the same program. (Note that there were several intervening reports describing this line of research that we were unable to obtain. These are cited in Way et al., 1990). This later research investigated the impact of 3-D imagery in cockpit display development. Interestingly, the results of this study indicated that adding stereopsis to a pictorial systems status display did not result in a significant change in either performance or workload. However, the report states that pilots generally found that the pictorial formats satisfied their information requirements. Way et al. also point to previous research that had shown that pilots find pictorial displays to be quite acceptable. This is not surprising, given the widespread use of pictorial formats in cockpits. These displays provide pilots with familiar icons that they can recognize easily. They also show the locations of different components or subsystems, as well as the topological connections between them. As such, they provide a physical representation that probably maps well onto pilots' mental models. However, as we will see in a subsequent section, there are limitations associated with the type of information that can be effectively displayed in a pictorial format.

2.3.3 Task-Oriented Displays

Task-oriented displays are similar to emergent feature displays, in that they are intended to support users' needs at a higher level that is more relevant to the users' task compared to more traditional display designs that provide users only with low-level data. Researchers at NASA Langley have conducted several experiments on task-oriented
displays pertinent to the present thesis. The most directly relevant and informative of these is the research of Abbott (1989, 1990) on aircraft engine displays. Abbott developed a novel engine display based on the task-oriented display design framework, whose goals are to present information tailored to the demands of a task, in a form that is easy to perceive. By presenting higher-order information in a manner that is easy to perceive, users are not required to derive information from low-level data (thereby reducing cognitive load) and can instead rely on their powerful pattern recognition capabilities. Using this theoretical framework, Abbott designed an innovative status display for engine subsystems, known as E-MACS (Engine Monitoring and Control Systems). Rather than displaying raw engine data in a traditional circular gauge format, the E-MACS display normalizes these data with respect to the expected values for each parameter, resulting in a set of deviation column graphs. These graphs directly display deviations from goals in a salient manner. Digital representations are provided for each parameter as well. Alarms are built into the display by changing the colour of the deviation column graphs to yellow when they enter a caution range, and to red when they enter a warning range. Finally, the E-MACS display also contains a predictor display to help regulate the thrust of the engines.

Abbott conducted an experiment comparing his E-MACS display to the EICAS (Engine Indication and Crew Alerting System) display that currently exists in many Boeing glass cockpits. The EICAS format consists primarily of circular gauges with caution and warning areas labeled along the circumference of the gauge. In addition, vertical "bow tie" bar graph displays are used to represent oil quantity, temperature, and pressure. The
EICAS display had already been shown to be superior to the conventional electromechanical instruments that it replaced (Abbott, 1989). The primary differences between the EICAS and the E-MACS displays are: a) the E-MACS display provides a predictor for the engine thrust, whereas the EICAS display provides a predictor for the engine pressure ratio (EPR); and b) the E-MACS display directly represents the deviation between the parameter values and their expected values, whereas the EICAS display represents the current value and the caution and warning limits on a common scale.

Sixteen pilots participated in the experimental evaluation of the two display formats. Subjects were seated in an aircraft simulator with both primary and navigation displays. The engine displays were shown only during set times for a brief duration (2 - 4 s) so that subjects would not (artificially) spend too much time looking at the displays. During the rest of the time, the screen was blank. Note that subjects were not required to perform any other pilot tasks (e.g., flying) during the tests. Moreover, unlike actual cockpit situations, no global warning/alarm was provided beyond the information given in the two engine displays. The subjects’ task was to monitor the engine displays, when they appeared for a brief moment, and detect faults in the engine subsystem. The results indicate that subjects preferred the E-MACS display designed by Abbott. In addition, fault detection performance was much more accurate with this display than with the EICAS display. These findings provide empirical support for the idea of displaying information that is directly tailored to task demands in a salient fashion.

Although these results are encouraging, it is important to note that fault diagnosis and fault compensation performance were not evaluated in this experiment. Just because a
display effectively supports detection does not necessarily mean that it can effectively support diagnosis and compensation as well. In fact, there are indications that the E-MACS display may not effectively support fault diagnosis. This speculation arises from our observation that the display does not show the relationships between the different engine parameters. Rather, it merely shows how far each individual parameter is from its expected value. In Rasmussen’s (1983) terms, the E-MACS display seems to provide support for rule-based behaviour, not knowledge-based behaviour, because there is no externalized mental model of the engine subsystem available to the pilot. This point will be elaborated upon in a later section discussing EID.

Another example of work in this area conducted at NASA Langley is the research of Palmer (1989) on system status displays for engine subsystems. A first experiment was conducted comparing two types of system status displays for a generic turbofan engine. The pictograph display contains a physical, iconic schematic, showing the topological connections between major components. It is similar to the piping and instrumentation diagram (P&ID) displays routinely used in process control systems (see next section). The block diagram display does not contain an iconic schematic, but instead shows the functional relations between the major components. Each of these displays only represented the qualitative status of the components: good, partially failed, and failed. Subjects were given one and five second exposures to the various display formats and were then given a comprehension test evaluating their knowledge of the status of the various engine components. The results revealed that subjects preferred the pictograph
display. However, subjects performed best with the block diagram display, although this difference was not statistically significant. Thus, no firm conclusions can be made.

In a second experiment, Palmer (1989) compared the performance of the pictograph and block diagram displays with a text display that contained the same information but in an alphanumeric format. In this experiment, these displays were used to represent fault history information. After a brief exposure to the display, subjects were required to recall the sequence of events representing the propagation of the fault. The results indicated that the pictograph led to the best performance, followed by the block diagram and then the text display. Only the latter two were not statistically significantly different from each other. This suggests that pictorial displays, which contain the topological connection between components, can lead to improved retention of information. However, fault diagnosis and fault compensation performance were not evaluated.

Trujillo (1994), also at NASA Langley, conducted an experiment investigating the effect of predictive and history information in helping pilots determine when displays will reach an unacceptable region. While the displays in the experiment were not specifically designed for engineering subsystems, the issue investigated is potentially relevant to the present contract. One of our earlier field studies indicated that rate information plays a critical role in fault detection and diagnosis for the flight engineer (see Chapter 4). Three different displays were compared in Trujillo's study: a standard round dial gauge, a standard gauge with history information, and a standard gauge with predictive information. The displays were dynamic but generic in the sense that they did not represent any particular system or meaningful set of parameters. There were three levels
of trial difficulty and two levels of exposure time (5 s and 10 s). Active airline pilots served as subjects, and their task was to estimate, using a rating scale, at what time in the future the value would reach an alert range. Surprisingly, the results showed that the displays with predictive and history information were both less accurate than the standard round dial gauge. Also, there was no difference in the time taken to make a decision. However, subjective ratings indicated that the predictive display required the least effort. This display was also the one that the pilots were most confident in using.

There are several factors that limit the external validity of the experiment. First, because the displays represented arbitrary information, it is unlikely that the scan patterns pilots used were representative of those used when observing meaningful data. Second, the task that subjects were required to do was not a very meaningful one. There is no reason to believe that pilots will develop the ability to accurately estimate, in a verbal, but quantitative fashion, how many seconds into the future a particular display will enter into an alert region. A more meaningful measure would have been to detect an out of range parameter, or the ability to anticipate an alert by acting appropriately. Thus, the only conclusion that one can derive from this experiment is that history and predictive information, as implemented in this study, make rating judgments of time to alert less accurate than having a standard round dial gauge.

Palmer and Abbott (1994) conducted two experiments comparing the effects of different types of display formats on fault detection and state identification performance. Both experiments were conducted with static displays. There were two types of displays in Experiment 1: circular gauge vs. “bow tie” bar graph. For each display type, there were
two presentation conditions: one in which a context-sensitive expected value range was superimposed onto the display; and one without any expected value information. The subjects were Boeing 737 pilots. Their task was to view a static display and then report which parameters were abnormal. Subjects were allowed to control the exposure times. It is important to note that all display conditions also had colour coded alarms on them. The results indicated that the circular gauges led to better performance than the "bow tie" displays, probably because the former were more familiar to the subjects. There were no significant effects involving the addition of expected value information. Palmer and Abbott speculate that the reason for this may have been the colour coded alarms in all displays, which allowed subjects to identify the failed components, regardless of whether they had expected value information or not. Thus, no strong conclusions can be made from these findings.

Experiment 2 compared the performance of three types of displays: the circular gauge with expected value information, the bow-tie bargraph with expected value information, and a column deviation bargraph based on the E-MACS display designed by Abbott (1989), described above. The task and the subjects were the same as in Experiment 1. The results indicated that the column deviation format led to significantly more accurate fault detection performance than either of the other two display formats. In addition, the column deviation formats were the most preferred by pilots. There was no main effect of display type for response time. These results suggest that providing information in a manner that is salient and defined with respect to system goals, as the column deviation format does, can lead to more accurate fault detection performance.
However, fault diagnosis and fault compensation performance were not investigated in this study, as was the case with Abbott (1989). Thus, it is not known whether the advantages of the column deviation format display generalize to these other important cognitive activities.

2.3.4 Critical Evaluation

The majority of computer display design guidelines and principles in the aviation and aerospace domains have focused on issues associated with visual form (e.g., NASA, 1980). These syntactic approaches, which are concerned almost exclusively with the surface features of the display, are necessary but not sufficient for effective performance. As Woods, O'Brien, & Hanes (1987) have indicated, such approaches will reduce the probability of slips, but do very little to address mistakes. To address the latter, one needs to focus on the information content of the displays, as well as the mapping between content and form. Only by adopting such a semantic approach can one create designs that lead to truly effective human performance.

Most of the work on aviation displays has been limited to the development of pictorial displays, which present operators with physical representations of the system (e.g., Baty, 1976; Reising & Emerson, 1982). The focus on pictorial realism is understandable in the aviation domain, since unlike other domains such as nuclear power plants, some of the goal-relevant properties in aviation are directly perceivable by human perceptual systems (under some conditions, at least). For example, the out-of-the-cockpit view can provide useful information to a pilot, whereas in a power plant, much of the information that operators are concerned with (e.g., pressure, energy, radiation), does not
have a natural pictorial representation. While the use of pictorial displays in aviation is understandable, they do have their limitations. Pictorial displays are only one kind of representation. They facilitate the representation of physical information, but do not easily capture higher order functional information, which operators require to perform some domain tasks. Pictorial displays focus on physical realism, rather than functional realism.

There is some research within the aviation domain that has been conducted on semantic display design issues that go beyond those associated with pictorial realism. Two approaches that we have uncovered in the limited work conducted are emergent feature displays and task-oriented displays. There is evidence in the literature that emergent feature displays can lead to improved performance in monitoring the status of engineering subsystems (Calhoun & Herron, 1981). Also, task-oriented displays have been shown to lead to better fault detection performance (Abbott, 1989, 1990). However, displays that have been shown to improve fault diagnosis and compensation performance seem to be rare. Thus, it seems fair to conclude that significant issues pertaining to the design of aviation status displays remain to be addressed. More specifically, comparatively little work has been done on comprehensively building domain semantics into status displays in order to support pilots during challenging fault management activities.

The following section reviews a body of research on interface design that may be helpful in addressing this gap in the literature. This work has been conducted in the domain of process control, not aviation. However, given the similarity between process
control systems and engineering systems on aircraft, there is reason to believe that this research may generalize to the design challenges found in aviation.

2.4. PREVIOUS RESEARCH ON ECOLOGICAL INTERFACE DESIGN

Ecological interface design (EID) is a theoretical framework for interface design for complex human-machine systems (Vicente & Rasmussen, 1990, 1992). The goal of EID is to allow operators to take advantage of their powerful perception and action capabilities while simultaneously providing the support required for problem solving activities that are required to effectively deal with unanticipated events. This section describes previous work on EID in order to assess its applicability to aviation.

2.4.1 Ecological Interface Design

EID is based on two seminal concepts from cognitive engineering research, the abstraction hierarchy (AH) and the skills, rules, knowledge framework (Rasmussen, 1986). The AH is a multilevel knowledge representation framework that can be used to develop physical and functional plant models, as well as the mappings between them. It is used in EID to identify the information content and structure of the interface. In addition to specifying interface content, the principles of EID also suggest how information should be displayed in an interface, thereby eliciting operators' powerful pattern recognition and psychomotor abilities. Thus, EID recommends that information be presented in such a way as to promote skill- and rule-based behaviour, allowing operators to deal with task demands in a relatively efficient and reliable manner. Knowledge-based behaviour is also
supported by embedding an AH representation of the work domain in the interface. This provides operators with an external visualization of plant structure and dynamics, which offers support during abnormal situations requiring problem solving.

The three principles of EID, each corresponding to a given level of cognitive control, are as follows:

1. Skill-based behaviour (SBB) - to support interaction via time-space signals, the operator should be able to act directly on the display. The structure of the displayed information should be isomorphic to the part-whole structure of movements.
2. Rule-based behaviour (RBB) - provide a consistent one-to-one mapping between the work domain constraints and the cues or signs provided by the interface.
3. Knowledge-based behaviour (KBB) - represent the work domain in the form of an AH to serve as an externalized mental model that will support knowledge-based problem solving.

For a detailed description and justification of these principles, see Vicente and Rasmussen (1990, 1992). For a detailed example showing how these principles can be applied to interface design, see Vicente and Rasmussen (1990) and Dinadis and Vicente (1996).

Because the AH plays such an important role in EID, it is important to describe it in more detail. The AH is a multilevel representation format that describes the various layers of constraint in the work domain. Each level represents a different model of the system. For many complex human-machine systems, five levels of constraint have been found to be of use:

1. The purposes for which the system was designed (Functional Purpose);
2. The intended causal structure of the process in terms of mass, energy, information, or value flows (Abstract Function);

3. The basic functions that the plant is designed to achieve (Generalized Function);

4. The characteristics of the components and the connections between them (Physical Function);

5. The appearance and spatial location of those components (Physical Form).

Higher levels represent functional information about system purposes, whereas lower levels represent physical information about how those purposes are realized by plant components.

There are two advantages to adopting the AH as a basis for interface design (Vicente & Rasmussen, 1992; Bisantz & Vicente, 1994). First, this approach allows one to identify, a priori, the information needed to cope with events which are unfamiliar to operators and which have not been anticipated by designers. This is a very important property since such unanticipated events pose the greatest threat to system safety in complex industrial systems. Whereas traditional approaches to interface design rarely make an attempt to deal with this problem, the AH was explicitly designed to deal with unanticipated events. Secondly, the AH is also a psychologically relevant problem representation. There is a significant body of empirical research from a number of diverse domains showing that problem solving protocols can be mapped onto an AH representation (see Rasmussen, 1986 and Vicente & Rasmussen, 1992 for reviews). Thus, in addition to satisfying the engineering requirement of containing the information needed
to cope with unanticipated events, the AH also satisfies the psychological requirement of providing a representation that is consistent with operators' problem solving processes.

As a result of its unique characteristics, EID is an obvious candidate for designing graphic, computer interfaces for complex human-machine systems where unanticipated events can occur. In the following subsection, some of the previous research conducted on EID will be reviewed.

2.4.2 Previous Research

Starting point. Initial work on EID was conducted in the process control domain. A review of the research and development efforts on multilevel interface design for power plant control rooms revealed that several researchers had developed interface designs that were based on an AH representation of the plant. Thus, these previous efforts were of direct relevance to EID. Unfortunately, up until the time that the review was conducted (1990), no experiment had been conducted comparing a multilevel interface based on an AH representation to any other type of interface (Vicente, 1992b). As a result, the direct empirical evidence for the EID framework was very weak, not because the results obtained were equivocal, but rather because appropriate studies simply had not been conducted. This provided a starting point for a research program on EID.

The remainder of this subsection will describe some of the recent research that has been conducted on EID. To reduce the scope to a manageable level, the review will be focused on the research that has been conducted by the Cognitive Engineering Laboratory (CEL) at the University of Toronto.
Research vehicle. Most of the CEL research on EID has been conducted with two related systems, DURESS and DURESS II. DURESS II is an updated, interactive version of DURESS, a thermal-hydraulic process control microworld simulation program. The original DURESS system was a non-interactive system that presented subjects with real-time, "canned" scenarios. Subjects were able to view the behaviour of the system but not to control it. DURESS II, in addition to some minor structural changes, differs from the original primarily in that it allows for real-time interaction, so that subjects may actively control the system.

DURESS II consists of two redundant feedwater streams (FWSs) that can be configured to supply water to either, both, or neither of two reservoirs. Each reservoir has associated with it an externally determined demand for water that can change over time. The system's purposes are twofold: to keep each of the reservoirs at a prescribed temperature (40°C and 20°C), and to satisfy the current mass (water) output demand rates. To accomplish these goals, the subject has control over eight valves, two pumps, and two heaters. All of these components are governed by first order lag dynamics, with a time constant of 15s for the heaters and 5s for the remaining components. A diagram of the system can be found in Vicente and Rasmussen (1990).

Two different interfaces for DURESS II have been designed to date. The first one is based on a Piping and Instrumentation Diagram (P&ID) representation of the system. This interface provides a physical view of the system, focusing on displaying the current state of all of the components and the goal variables. This display format is similar to the mimic diagrams that have been traditionally used in aviation to display engineering
systems. The P&ID format was chosen for research on EID because it is typical of how existing computer interfaces for process control systems have been designed. Thus, it serves as a meaningful control condition.

The second interface for DURESS II was based on the principles of EID. A detailed explanation of how this interface was designed can be found in Vicente and Rasmussen (1990), so only a brief overview will be provided here. The EID interface contains all the information contained in the P&ID interface, but it also contains higher-order functional information that was identified through an AH analysis of DURESS (see Vicente & Rasmussen, 1990). Examples of functional information include valve flow rates, heat transfer rates, and in particular, mass and energy topologies. This added information provides a basis for supporting effective reasoning during unanticipated events, one of the important goals of the EID framework. In addition to the inclusion of this high-order relational information, another distinctive feature of the EID interface is the form in which information is provided. The interface was designed so as to preserve a one-to-one mapping between domain constraints and the salient, perceptual properties of the interface, thereby providing an external visualization of the system that is intended to enhance information extraction. This is in keeping with the EID goal of supporting the power of perception and action. A more detailed description of the rationale behind the design of this interface can be found in Vicente and Rasmussen (1990).

Laboratory findings. The first experimental evaluation of EID was conducted in the context of the non-interactive version of DURESS (Vicente, 1992a). The goal of that study was to compare the earlier versions of the P&ID and EID interfaces in terms of how
well they support problem solving behaviour, or KBB. The experimental evidence indicated that the EID interface provided better support for KBB than the P&ID interface. While this finding was encouraging, the study was limited in several ways. First, the subjects were either theoretical experts or novices at generic thermal-hydraulic principles, but neither group had any substantial experience with DURESS itself. Second, the experiment evaluated the subjects' ability to diagnose and remember the values of dynamic, “canned” scenarios. Therefore, subjects did not interactively control the system. While there were reasons for conducting the experiment under this restricted set of conditions (cf. Vicente, 1992a), it remained to be seen whether the advantage of the EID interface would still hold with subjects with extensive experience controlling DURESS.

The second evaluation of EID was also conducted in the context of DURESS (Vicente, Christoffersen, & Pereklita, 1995). In that study, subjects again diagnosed real-time canned scenarios, but this time, only using the EID interface. In addition, verbal protocols were collected as the subjects tried to diagnose the nature of the events presented to them. A process tracing analysis was conducted by mapping subjects' verbalizations onto a two dimensional problem space representation of DURESS, defined by an AH and a part-whole hierarchy (Rasmussen, 1986). Process measures of performance were correlated with product measures of performance to determine whether there were any statistically significant relationships between subjects' cognitive strategies and their diagnosis accuracy. The results indicate that the greater the extent to which subjects adopted the top-down "zooming-in" strategy, that the AH is intended to support (Rasmussen, 1986), the more accurate their diagnosis performance was. This experiment
was the first to empirically demonstrate the problem solving advantages associated with reasoning in an AH problem space. Nevertheless, the results are limited by the fact that the subjects did not interactively control the system.

A third study was designed to overcome the limitations of the previous two (Pawlak & Vicente, 1996). The experiment was conducted with the interactive DURESS II simulation so that subjects could control the system components. Slightly revised versions of the P&ID and EID interfaces were compared. Subjects were given extensive practice at controlling the system (one hour per weekday for four weeks) with one of the two interfaces before their performance on normal events and unfamiliar faults was evaluated. The results revealed that, under normal conditions, there was no performance difference between the EID and P&ID interfaces. However, dual task performance results indicate that the P&ID interface relies more on verbal resources, whereas the EID interface requires more spatial resources. Furthermore, a process tracing analysis of the fault trials showed that the EID interface led to faster fault detection and more accurate fault diagnosis performance. In addition, two deficiencies of the EID interface were identified, one suggesting a need for integrating trend information with emergent feature displays, and the other suggesting that displays tailored for enhanced perception of the system state may not always be well suited for fault compensation. The primary contribution of this study, then, was to compare the P&ID and EID interfaces under a more representative (Brunswik, 1956) set of experimental conditions.

The next experimental evaluation of EID was intended to investigate the long-term influences of EID on operator performance and knowledge (Christoffersen, Hunter, &
Vicente, 1994). The EID and P&ID interfaces for DURESS II were again compared. More specifically, a longitudinal experiment lasting six months was conducted to compare these interfaces under a variety of conditions, including normal trials, routine faults, and non-routine faults. Just as in the previous study, subjects controlled the system every weekday (not including holidays) for about one hour per day. Product measures (e.g., time, actions) and process measures (e.g., verbal protocols) of performance were collected. In addition, several knowledge elicitation measures were occasionally administered to determine how the subjects’ knowledge organization evolved over time. At the end of the experiment, subjects switched interfaces and had to control DURESS II under normal and routine fault conditions using the new interface. The primary findings of the study were threefold. First, on normal trials, there was very little difference in the average performance of the interface groups. The group using the P&ID interface, however, consistently showed more variability in their performance, occasionally taking much longer than usual to complete the required tasks. This effect was found to hold even after five months of practice, and was specific to the interface, not to the subjects. Second, for both routine and non-routine faults, the EID interface was found to lead to better performance, especially with respect to diagnosis accuracy. These effects seemed to stem from strategy differences between interface groups, which were in turn a result of the interaction between the subjects’ knowledge and the information provided in the interfaces. Third, subjects using the EID interface who actively explored the system and reflected on the feedback provided were able to achieve levels of adaptation and performance not observed with subjects using the P&ID interface. When adopting a
surface approach to learning, however, subjects using the EID interface were likely to exhibit a very shallow knowledge base and poor performance (although no worse than that attained by subjects using the P&ID interface with a comparable level of motivation). Thus, it appears that there are certain preconditions which have to be satisfied if the benefits of the EID interface are to be fully elicited.

**Industrial prototype.** Although the results obtained from the laboratory studies with DURESS and DURESS II had been very encouraging, it was important to determine if the EID framework can be scaled up to complex industrial systems. As a first step in this direction, a project was undertaken with ABB Corporate Research - Heidelberg to design a prototype EID interface for the feedwater subsystem of an ABB conventional power plant (Dinadis & Vicente, 1996). This proved to be a very valuable experience.

The study showed that the EID framework can be meaningfully applied to industrial systems that are much larger in scale than the DURESS system that had been used to evaluate EID. An AH analysis of the feedwater subsystem was conducted and served to identify the information content of the interface. This information was then mapped onto salient perceptual cues in the interface that are intended to make information extraction accurate and efficient.

Another significant outcome of the feasibility study was a proof of principle that EID can be integrated with other interface design concepts. EID was only intended to address some basic issues in interface design (Vicente & Rasmussen, 1992), and so there are several important design problems for which it does not provide guidance. Perhaps the most salient example is the issue of navigation. There are many different ways to display a
multilevel interface based on an AH across display pages or windows (Vicente, 1992c), but EID does not prescribe which is the most effective. In fact, there is very little empirical evidence on which to base such a decision (Vicente, 1992b). However, the concept of visual momentum (Woods, 1984), which was specifically developed to address this issue, was used in the development of the ABB prototype.

Similarly, EID is also silent on specific issues about coding of visual form. Because it is intended to be a general framework (Vicente & Rasmussen, 1992), EID motivates designers to pursue certain objectives (e.g., mapping domain constraints onto perceptual signs in the interface), but it does not offer a set of procedural steps for how to achieve those objectives. Again, however, it was possible to rely on other design principles (i.e., perceptual organization techniques) to address this interface design issue. These examples reinforce the point made by Vicente and Rasmussen (1992) that following the principles of EID alone does not allow one to design an effective interface for large scale systems. However, these results also show that EID provides a solid basis for design that addresses fundamental issues, and moreover, that it can be integrated with other important design principles. This makes EID a viable candidate for designing interfaces for complex industrial systems.

Technology Transfer to Industry

If an interface design framework is to be more than an academic curiosity, it must be perceived by industry to be relevant to significant design problems. How well has EID fared in this regard?
Despite the relative novelty of the framework, there has already been a limited success in transferring EID to industry. This technology transfer has occurred primarily in the nuclear and process control industries and to varying degrees. A very modest form of technology transfer has occurred with AECL (Atomic Energy Canada Limited) Research and Honeywell. Both of these companies have adapted the EID interface for DURESS into prototypes that are intended to illustrate state-of-the-art interface design concepts that may find themselves in advanced control rooms of the future.

Researchers at the Nuclear Engineering Laboratory of Toshiba in Japan (Monta et al., 1991) have gone much further. They have adopted EID as the basis for designing their advanced control room for a next generation boiling water reactor plant. In addition to adopting the framework, they have also incorporated and adapted specific features of the EID interface for DURESS (e.g., the mass balance graphics) into some of their displays. This application is notable since it has been conducted at the scale of a full-scope nuclear power plant simulator. It is important to note, however, that this application of EID has yet to be evaluated.

More recently, Mitsubishi Atomic Power Industries in Japan has demonstrated a very strong interest in EID. They have contracted Battelle to initiate a five year research program, solely on EID. The program is recent, and results are not yet available.

In summary, EID has already attracted a fair amount of attention, at least from the nuclear industry. This is notable for several reasons. First, most of the research on EID has been conducted on a simplified yet representative process control simulation. This demonstrates that it is possible to conduct laboratory research that has meaningful
implications for complex, applied problems. Second, technology transfer from basic research to industry has been very hard to come by in human factors. The limited success of EID therefore stands out from the norm.

Nevertheless, it is important to point out that the true value of EID to industry has yet to be firmly established. It would certainly be very premature to suggest that all control rooms for nuclear power plants, for example, should be based on EID. Many research issues still remain to be addressed. More importantly, EID has yet to be evaluated on a large scale with professional operators. Thus, while there are many reasons to think that EID is a promising avenue to pursue, it is much too early to recommend it as a proven design framework.

Relevance to Aviation

For the purposes of this thesis, the most important unanswered question regarding the EID framework is its generalizability beyond the domain of process control. The research conducted to date has shown that an interface based on EID can lead to substantial improvements in performance compared to more traditional P&ID formats. Interestingly, the P&ID format seems to be the traditional approach to displaying the status of engineering systems in aviation, just as it is the traditional approach in process control. This suggests that EID might lead to an improvement in performance in aviation as well. It is important to note, however, that before this thesis research, this was merely a conjecture, since EID had never been applied to aviation. Thus, the first step is to determine whether the principles of EID can be applied to aviation. This proof of concept is the primary objective of this thesis.
Before the design process can be initiated, it is necessary to first understand as much about the domain in which the interface will potentially be used. In order to obtain this prerequisite knowledge, two field studies were conducted. They will be discussed next.
CHAPTER 3: OBSERVATIONS OF FLIGHT ENGINEERS DURING REFUELING MISSIONS

3.1. MOTIVATION

Between April 14 and April 21, 1995, a number of Canadian Air Forces (CAF) Hercules crews were observed in flight. The crews' mission was to replenish the fuel supplies in Alert, a strategic CAF base. Due to Alert's extremely northern latitude, approximately 86° N, shipment by any form of transportation other than aircraft is not feasible. Military personnel in Alert rely on weekly aircraft arrivals to restock food supplies and other necessary staples. Because so many personnel depend upon them,
crews will often endure extremely hazardous weather conditions in order to complete their missions. In doing so however, they constantly place themselves in abnormal flight conditions which often require unorthodox flying methods. Because of its very nature, military flying is very different from commercial flight; there is more leniency extended to crews to attempt more difficult landings when required. During wartime situations, crews can be forced to make hazardous landings and fly dangerous flight paths without choice.

The purpose of this study was to observe the crews, with an emphasis on the duties of the Flight Engineer (FE). In this chapter, the role of the FE is examined; the information available to the FEs and how they have adapted to, and modified this information is also discussed. This is followed by recommendations which could be used to benefit the FEs of traditional Hercules C-130 aircraft.

3.2. Method

Information for this study was gathered during an eight day period in the month of April, with approximately 15 hours of flight per day for six days (90 hours). Poor weather conditions in Alert prevented flights during the first two days. Data were collected via interviews with FEs during and after flights, and on-line observations.

3.3. The Role of the FE in the Aircraft

Military terminology describes any abnormality or fault as a “snag.” Due to the age of Hercules aircraft, most snags are mechanical problems that require in-flight troubleshooting by the FE. Thus, the FE is intimately familiar with the aircraft; and will
often arrive at the aircraft a full one and a half hours prior to take-off in order to perform a meticulous inspection, both inside and outside the vehicle. Upon completion of pre take-off checks, the FE essentially acts as a third pilot. However, unlike the FE in the Aurora aircraft, who controls the thrust of the engines during take-off, the FE in a Hercules monitors the engine temperatures and the thrust settings, advising the Pilot Flying to reduce or increase thrust as necessary. Aborting a mission during take-off is seldom done, but it is nevertheless one of the duties of the FE to make such a recommendation if necessary. Although the final decision to abort a take-off belongs to the Aircraft Commander (AC), once the FE calls an aborted take-off, the AC will normally comply.

There were three aborted take-offs during an observed simulator session before the aircraft was finally in the air. Rare as they are though, the probability of a severe problem occurring due to a fault increases while the aircraft is just about to take-off or land. As a result, simulator sessions stress situational awareness and fault management during this period of flight. Upon achieving cruising altitude, the FE’s job becomes a vigilance task, accompanied by his perpetual responsibilities as the aircraft’s mechanic. However, if a snag were to occur - as they frequently do in the simulator - the FE once again accepts the role of the primary problem solver, and does his best to maintain the integrity of the aircraft.

According to the Hercules C-130 Crew Guide, page 8-2 (Department of National Defence, 1987), the official CAF role of the FE is as follows:

a. Computes take-off, climb, cruise, and landing data; adjusts engine controls
in co-ordination with the pilot to maintain required power during climb and cruise flight conditions. Maintains power plant cruise control data.

b. Operates system controls; regulates electrical system.

c. Controls cabin air to provide proper cabin ventilation, pressurization, and temperature.

d. Operates all anti-icing systems.

e. Starts gas turbine compressor and air turbine motor to provide auxiliary power as required.

f. Operates external light panel.

g. May operate aft cargo door and ramp in flight.

h. Observes engine instruments, systems indicators, and control devices.

i. Continuously monitors turbine inlet temperature, tachometers and torquemeters, and reports unusual conditions to the pilot.

j. Monitors circuit breakers, fuel flow, temperatures and pressure indicators, electrical voltage and loads, and cabin pressure control and altitude indicators.

k. Observes warning lights and fire detection indicators.

l. Reports abnormal conditions to the pilot and recommends corrective action.

m. Operates Navigator's equipment when the Navigator is not on board.

n. Performs pre-flight and post-flight inspections.
o. Inspects turboprop engine for general condition of turbine blades and for absence of fuel leaks.

p. Troubleshoots malfunctioning airplane systems in flight.

q. May supervise the removal and replacement of all airplane system components if qualified maintenance personnel are not available.

r. Performs fuel management.

s. Takes emergency procedure actions as required by the flight manual and/or the pilot.

What is interesting to note about the previous list is that the FE's regular duties basically permit him to do everything but "steer" the airplane. It should therefore not be surprising that the FE is the most knowledgeable crew member with respect to the aircraft itself. In order to achieve the status of a FE for Hercules C-130 Aircraft, one must first advance through the various ranks within the military and obtain a minimum level of Sergeant within the CAF's Flight Engineering Division. Becoming a Junior Flight Engineer requires a minimum of five years within the Airframe, Aeroengine, or Instrument/Electrical trades of the CAF, as well as a trade qualification level of five, followed by a one-year training course. After four years as a helicopter or twin engine FE, and one more year of courses, the engineer becomes a potential FE for the Hercules. At this time, he has gained extensive knowledge pertaining to every aspect of the aircraft. Many of the trades personnel can provide the exact location of any part of the aircraft, what it affects, and how the system can be circumvented, even before they obtain FE status. Following this, a
potential FE must go through numerous simulator sessions before being permitted to occupy the FE's seat in a C-130 as a trainee with another qualified FE on board. The only major disadvantage to the current training system is thatFEs only occupy their role for about 15 years, as the average age of new FEsin the Canadian Armed Forces is approximately 35, and retirement occurs at the age of 55. In order to preserve the experience level in the forces, the older FEstend to depart from their in-flight duties and assume the role of trainer and assessor before they retire.

3.4. C-130 INSTRUMENTATION

The most effective method of conveying the degree of difficulty associated with the FE's duties is to first present the layout of the cockpit, and to subsequently discuss the controls and instrumentation available to the FE. Figure 2 is a photograph of an existing Hercules C-130 cockpit. This cockpit was conceived in the 1960's, and the instrumentation depicted is commonly referred to as first generation (Wickens, 1992). All gauges are electromechanical, and no CRTs are available. In the last decade, Hercules aircraft have been retro-fitted with CRTs for the Navigator, but this addition does not benefit the FE. Portions of the cockpit not visible in Figure 2 include the panels of circuit breakers located along the sides of the cockpit just behind the pilots' seats, and hydraulic gauges and switches, located in front of the First Officer (FO).
3.4.1 Engine Status Gauges

Figure 3 presents a close-up of the main instrumentation panel, which is located in the main field of vision for the FE, and is known as the Engine Status Gauge panel. The FE monitors this panel, and uses its information to troubleshoot problems as required. The panel is set up as an 8x4 array of circular dials, with each column corresponding to a particular engine, and each row corresponding to a specific engine measurement. The most important information pertaining to the performance of the aircraft is located at the top, with the relatively less important information at the bottom. What is essential to
Figure 3: Close-up of the Engine Status Gauges

notice about the dials is not where they are placed, as Lockheed (the manufacturers of the Hercules) did well in organizing the information topographically, but rather the means by which information is conveyed. Good topography is essential in this type of single sensor single indicator (SSSI) layout (SSSI means that each instrument is dedicated to giving the status of one component, (Goodstein, 1981)). The disadvantages of such a design include relying on the operator to mentally combine and compile sensor data into useful information himself, and having to incorporate a large amount of instrumentation into minimal cockpit real estate. Recent advances in CRT technology have led to glass cockpits that can provide an equivalent amount of information that is automatically compiled for the pilot, while conserving cockpit real-estate (Sexton, 1988). However,
such designs are still in their infancy and are not without problems (Hughes, 1995).

Another important aspect which the SSSI instruments do not offer is context sensitivity.

In the case of the Hercules, it appears that the manufacturers did not concern themselves with obtaining the most appropriate dials for the task. The second and sixth rows of dials are an example of this. The second row, as indicated in Table 1, contains the tachometers.

In a Hercules, if the propellers are not spinning above 95%, there is a problem (unless the engine has been shut off), as is indicated by the red and green pieces of tape (see Figure 2). There is no need to have a gauge which ranges from zero to 100%. What is required is a gauge that ranges from 90% to 100%. A similar problem exists with the sixth row of dials which contains the oil pressure of both the engines and the gears (gears refer to the main gearing which drives many of the subsystems, such as the hydraulic pumps and the propellers). The tape segments to the left of this dial indicate the gear oil pressure limits, while the tape at the bottom indicates engine oil pressure limits. The upper bound of the dial need not be above 2.5, as opposed to 5. Half of the dial is comprised of unnecessary space that could have been much better utilized had a smaller range been used.

<table>
<thead>
<tr>
<th></th>
<th>E1</th>
<th>E2</th>
<th>E3</th>
<th>E4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Torquemeters in lb. x1000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tachometers rpm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIT N2 °C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fuel Flow rpm x1000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oil Temp °C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oil Press. Eng. &amp; Gear psi x100</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oil Quantity US Gal.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oil Cooler Flap %</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Table 1: Layout of the Status Gauge Panel*
Compounding the context sensitivity problem is the limited precision inherent in the electromechanical devices. With the engines off, as is shown in Figure 3, the smaller dials on the RPM gauge (second row) are displaying different values, even though they should all be zero. This manifests itself as an issue when synchronizing the propellers. The FE must maintain the propellers spinning at the same rate in order to prevent aircraft fatigue and sickness to the crew and passengers caused by the "wowing" that occurs when the propellers are out of phase. Similarly, in the sixth row (Oil Pressure), the dials range from -10 to 10 psi. Although a small range is used, a problem still exists because the acceptable operating range for the engines is also 10 psi. Similar problems can be observed in the other rows of dials as well, with the exception of the seventh, containing the oil quantity. Before discussing how the FEs cope with the minimal information which they are given, it is necessary to introduce the rest of the cockpit components, namely those located in the Overhead Instrumentation Panel.

3.4.2 Overhead Instrumentation Panel

The Overhead Instrumentation Panel (shown in Figure 4) is located above the FE's field of view. Some of these instruments are placed over top of the FE's head, requiring him to lean back to read a gauge or adjust a setting (this has lead to complaints of back problems). Since the panel is somewhat convoluted, Figure 5 has been included to provide extra insight into the panel layout. Although somewhat poorly positioned in the cockpit, the designers at Lockheed realized that the FEs would be manipulating the fuel and the electrical systems frequently. They are therefore situated as close to foveal vision
along the upper panel as possible. However, this positioning also places these dials at a distance from the FE. This does not create much difficulty during normal flight, but in an abnormal situation, the FE is incessantly leaning over the pilot's shoulder to adjust knobs, and is constantly having his work interrupted by the AC who is performing his own duties.

The presentation of information on the Overhead Instrumentation panel (Figure 4) is similar to P&IDs; lines are drawn to depict physical connections, and labeled dials replace labeled pictorial representations of components. The major difference compared to P&ID however, is that the panel is not static and can be manipulated by turning the dials. For example, in the case of the fuel tank instruments, the red lines indicate which components are topographically linked to one another, so that the FE does not have to completely
memorize the information. More importantly, the dials also have lines on them to indicate status. If a dial is turned so as to break the collinearity, the valve is off, and fuel is not supposed to flow. If the line on the dial lines up with the piping, then the valve is fully open, and fuel is supposed to flow at its maximum rate. Once again however, the

Figure 5: Diagram of the Overhead Instrumentation Panel

SSSI design lacks functional information. For example, there is no indication as to how much fuel is flowing through each valve, to alert for blockages. Thus, in order to effectively control the system, FEs are forced to enhance the minimal information which they are given, and have done so through simple modifications and ingenuity, which shall be discussed next.
3.5. **Information Enhancement**

A Hercules aircraft has thousands of parts and hundreds of dials to maintain. There is plenty that can mechanically go wrong without having to consider inaccurate instrumentation. There are certain aids that FEs learn to use, and have developed over the long course of the C-130's lifetime, to help cope with the minimal information which with they are provided. Two such examples are the use of sound, and the use of ledgers.

In an aircraft where there is no trend information nor the ability to directly observe rates of change on instrumentation, the FEs have resorted to the use of log entries to aid them during their missions. A Hercules Flight Engineer Log (HFEL) is shown in Figure 6.

This tool has undergone numerous alterations, reflecting the needs of the FEs during a flight. As indicated in Subsections 3.1 and 3.2, the FE must monitor the status gauges very closely, and the log reflects this. For any flight longer than two hours in duration, the FE must record the values of the instruments every hour. If a problem arises, or the FE suspects that a reading is deviating very slowly, then he would record the values of that gauge every 15 minutes, or whatever interval he considered appropriate.

Aside from its use in calculating rates of change in instruments, the HFEL is also used for computing aircraft fuel amounts. Sections dedicated to fuel consumption allow the FE to record how long the engines have been running and to correct recorded values for ambient conditions. Complicated calculations are simplified by the addition of various empirically developed constants, such as those listed in the “Fuel Used to Top of Climb” portion of the HFEL, located on page 1 of Figure 6, as well as the constants indicating
"Fuel Density Variations", on the last page. Further explanation of the HFEL is provided in Appendix B.

In a noisy aircraft such as the Hercules, sound can play an important role in aiding the FE. With experience, FEs learn to associate various sounds with particular faults. One example is the noise created by the propellers. If one of the propellers rotates at a certain phase out of the rest, there could be medical consequences to the crew and passengers. The FEs have learned to capitalize on the "wowing" sounds created during such situations to determine the accuracy of the gauge readings. There is an automated device known as the syncrophaser which is designed to rotate all the propellers at the same rate. If the device is not functioning correctly, wowing occurs and the FE can respond to the problem before it is felt by the passengers. Although noise can be used as an aid to an experienced FE, it can also cause serious difficulties to an inexperienced one, especially when flying with an inexperienced crew. An example of noise offering misleading information is a phenomena which occurs during take-off, known as the Threshold Bump. In cold-weather situations, as the aircraft picks up speed, and the Temperature Datum (TD) system crosses over from a temperature controlling to a temperature limiting mode (to prevent an over-temperature situation), a distinct "whir" is heard accompanied by a brief loss of thrust of about 1000 lb. to each engine. While observing a flight in a simulator session, a fault was placed during take-off which occurred at the approximate time that the threshold bump would occur. The fault was a
massive bleed air leak between the third and fourth engines (so that it could not be visually observed from the cockpit). This resulted in a loud "bang" accompanied by a loss of 2000-3000 lb. thrust to each engine (intended to mimic the threshold bump). If left unnoticed, the 600°C air being expelled from the ruptured bleed air pipe would sever all wiring leading to the fourth engine. In the extreme case, the hot air could also potentially cause the external tank to explode and destroy the wing if left unnoticed. This fault is introduced to prevent the FEs from becoming accustomed to the "whir" and power loss, and to teach them to verify all the awkward (if not abnormal) sounds they hear. The most interesting aspect of this simulator session was not the fact that the fault occurred, but the actions that the FE took during take-off. Both the FE and the pilots were experienced, so
they immediately set the throttle to temperature limiting at take-off, thereby preventing the Threshold bump from occurring. Thus, once the FE heard an odd sound, the take-off was immediately aborted, and the problem was rectified before it became exacerbated. Presenting a situation in which a FE cannot trust his ears during a routine flight demonstrates how the FEs have used their extensive knowledge of the aircraft to proactively compensate for one of its many deficiencies.

3.6. Compensating For Deficiencies

This section is devoted to anecdotes observed during the “BoxTop” mission. In such missions, where aircraft are in the air 24 hours per day under different crews, it is essential that outgoing FEs relate snag information to boarding FEs. In one situation, an FE noticed that the external tank gauge was acting peculiarly, and decided to watch it closely. After determining that the gauge was unreliable, he placed a piece of tape on the dial in the form of an “X” (See Figure 7). This clearly indicated to all oncoming FEs that the gauge was somehow faulty and should not be used, while at the same time, reminding him not to look at, nor mistake it for another.

A similar situation occurred when an extra crew member noticed that there was a crack in one of the windshield panels. This type of situation can be dangerous as the window might break, causing the cabin to depressurize. In order to keep a close eye on the windshield while still monitoring his instruments, the FE used a magic marker to trace the crack. Thus, every 15 minutes when he observed the pane, he could tell how far the crack had progressed without interrupting his duties. These are very efficient methods of
time and resource management, whereby the FEs effectively "create" information (tape on the gauge and marker on the windshield) to help them deal with abnormalities.

Interestingly, Vicente and Burns (1995) have observed and documented similar behaviour in nuclear power plant operators, who also work with SSSI interfaces. In their study, operators were observed opening chart recorder doors to make them stand out from others, and they also pulled the tray of the most important chart recorder out in order to distinguish it from the rest. Although the previous examples might not seem interesting, it must be kept in mind that the FE is problem solving on the fly, which compounds the effects of every fault.

A final anecdote involves one of the flights into Alert. A crew carrying a full
payload of fuel encountered a major snag. As they approached the airstrip and sent the signal to lower their landing gears, the system did not respond. When the FE checked the circuit breaker panel, he noticed that the breaker had popped for the landing gear. It is well documented that the FE is permitted to reset a circuit breaker only once. If it fails, he has to resort to other means of accomplishing his goal (e.g., manually lowering the gear). Instead of manually operating the system though, the FE decided to convince the aircraft that everything was normal. He asked the FO to lower the gear and as the switch was set the FE pushed the breaker back in to complete the circuit. The FE explained that it did not matter how much electricity was sent to the solenoid valve which held the gear up, as long as it switched and lowered the gear. Because the landing gear lowered, the FE was able to determine that the solenoid valve was stuck, and that it was not a problem with the gearing itself. Had the gear not gone down after this attempt, the FE would have had to resort to manually lowering the gear. This is a much more time consuming and stressful process, especially for someone who had logged 17 hours of flight that day. Once again, the FE was able to use his detailed knowledge of the aircraft to enhance information and produce a viable solution.

3.7. Implications

The previous observations have resulted in several short term and long term implications that could help to better serve the FEs of Hercules aircraft.

Several inexpensive modifications that could be made include:
- Replace the dials on the Status Gauge Displays to better indicate the situation at hand (more context sensitive).
- Place light bulbs on the top of the status display to indicate non-functioning instruments in the Overhead Panel. Bulbs should be colour coded to match the coding on the panel (i.e., red for fuel, yellow for electrical systems, orange for de-icing)
- Use simulator sessions to concentrate on ‘awkward’ (e.g., propagating) faults and the ‘tricks’ (e.g., using the circuit breaker to lower the landing gear) used to circumvent them. Current simulator sessions incorporate awkward faults, but supervisors don’t always share their knowledge with less experienced FEs. This extra transfer of information could prove useful during a real fault situation.

Aside from the above simple modifications, more complicated measures could be adopted to aid the FE.

- CRT displays should be used to replace gauges. This would improve the visibility and sensitivity of measurements if done properly.
- The Status Display Gauges should be physically moved forward to decrease the amount of straining required by the FE to read a dial.
- The Overhead displays could be moved to the flanks of the FE so that they would be constantly visible and within reach.
Of all the recommendations made, the FEs of Hercules aircraft would most likely object to digitizing the Status Display Gauges. Because they are very mechanically inclined, they appreciate being able to interchange one dial for another with little difficulty when something goes wrong. One major hurdle for FEs to accept is the likelihood that a main battery pack could fail, rendering all electronic CRT based systems (and dials) useless.

3.8. SUMMARY

The duties of an FE in a Hercules C-130 aircraft require constant vigilance during monotonous tasks. This is accompanied by the continuous possibility of a serious aircraft malfunction requiring creative, and sometimes unorthodox, problem solving. The above discussion has provided some insight on how FEs accomplish their tasks, which involve extremely high levels of discipline and efficient resource management. The next chapter provides additional insight into the responsibilities of the FE, particularly under abnormal situations.
CHAPTER 4: PROTOCOL ANALYSIS OF A HERCULES C-130 CREW DURING A SIMULATOR SESSION

4.1. PREFACE

This chapter is dedicated to better understanding the task demands inherent in the duties of a FE. In the previous chapter, the role of the FE was analyzed through direct observation and through personal interviews, both during and after missions. This chapter describes the results of a protocol analysis study observing CAF crews during two in-depth simulator sessions. With the aid of a FE instructor, a predetermined set of faults (unknown to the crew) was utilized in an attempt to extract as many of the nuances of the FE’s job as possible. Only the results from the first half of a four hour long “LP 3” simulator session will be presented. A list of faults which occurred during the segment analyzed are listed in Table 2. As indicated, some of the faults were either not given or were not on the video tape that was analyzed.

<table>
<thead>
<tr>
<th>Malfunctions - Emergency A</th>
<th>On Tape?</th>
<th>Given?</th>
<th>Fault Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>A/C will not accept power</td>
<td>N</td>
<td>Y</td>
<td>1</td>
</tr>
<tr>
<td>#4 Engine stalled start</td>
<td>N</td>
<td>Y</td>
<td>2</td>
</tr>
<tr>
<td>#2 Engine hung start</td>
<td>Y</td>
<td>Y</td>
<td>3</td>
</tr>
<tr>
<td>#1 Hydraulic shutoff valve circuit breaker trips</td>
<td>Y</td>
<td>Y</td>
<td>4</td>
</tr>
<tr>
<td>#1 Fuel gauge hard over (high)</td>
<td>Y</td>
<td>Y</td>
<td>5</td>
</tr>
<tr>
<td>Co-Pilot’s ASI fails</td>
<td>Y</td>
<td>Y</td>
<td>6</td>
</tr>
<tr>
<td>Multiple bird strikes #1 and #2 engines</td>
<td>Y</td>
<td>Y</td>
<td>7</td>
</tr>
<tr>
<td>Nose gear fails to extend</td>
<td>N</td>
<td>N</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 2: List of all the faults which were to occur during the LP3 session
4.2. **METHOD**

Data for analysis were obtained at the CAF full motion Hercules C-130 simulator at CFB Trenton. There are typically two to three faults introduced before the aircraft becomes airborne, and FEs arrive at the simulator prepared for faults. Thus, vigilance does not play as significant a role as it does during a mission. However, this is not always the case, as is demonstrated by the following example. One of the instructors decided to simplify an inexperienced crew's flight by choosing not to introduce any faults during their simulator session. However, the crew's expectancy for a fault was such that they mistook normal instrument readings as abnormal, and guided the aircraft into the ground! In this rare simulator incident, the normal instrument fluctuations combined with an over-anxious crew caused the accident to occur. Had the crew been flying a regular mission, their expectancies would have been sufficiently altered to negate the instrument noise as a potential fault. Aside from such rare occurrences, the former statements remain true.

Data during the session observed were recorded on video tape and obtained during interviews with the FEs. The raw data provided for this analysis were a video tape of the session and a copy of the C-130 Flight Crew Emergency Check List (Department of National, 1993) that is used by FEs to troubleshoot. The events and verbalizations captured on tape during the first two hour portion of the "LP 3" session were then transcribed (See Appendix A). Included in the transcriptions are the remarks of the FE, the Pilot (AC), the Co-pilot (FO). The comments made by the Load master (LM), the Air Traffic Controller, and the instructor are all made by the instructor himself. One major deviation from reality was the instructor's interjections, which often cued the FE to
answer a particular problem. Finally, quotes taken from the sessions are presented in this chapter. Square brackets indicate comments inserted by the author in order to clarify certain crew comments. The scenario run by the crew began with the aircraft not accepting power, followed by the #4 engine stalling on its start. Unfortunately, neither of these faults was captured on tape, and as a result, they are not considered in the analysis.

4.2.1 Flight Simulator Session Description

Below is a brief summary of the events occurring during the first leg of the LP3 Long session. The actual time required was approximately 2 hours, with five snags recorded on video tape. The description of the scenarios given to the instructors was as follows:

**LOCAL TRAINER**
The purpose of this session is to expose crews to critical flight handling in IMC conditions with high density altitude conditions. The scenarios contained herein will challenge the crews by requiring them to apply good judgment and effective use of ACT as well as flight handling. In conjunction, various snags and emergencies will be revisited.

This session will take place utilizing Calgary International Airport visual scene on the Western disk, and will serve as a logical follow on from the previous LP 2 Long Session terminating in Calgary [and continuing on to Edmonton]...

**OVERVIEW**
This session will include the following sequences: [abridged]
1. Heavy weight (150,000 lb.) / minimum weather take-off below cat 1.
2. Aborted take-offs
3. Engine malfunctions greater than V1 [minimum take-off speed]
4. 2 engine operation / 2 engine IFR approach and overshoot
5. Windmill taxi start.
6. Max effort landing
In the session analyzed, the order of events, including the approximate time which they occurred and whether they were used or not, are indicated in Table 3. Detailed descriptions of each event are presented in the following section.

<table>
<thead>
<tr>
<th>Time</th>
<th>Description of Event</th>
<th>Removed?</th>
<th>Fault #</th>
</tr>
</thead>
<tbody>
<tr>
<td>0:00</td>
<td>#2 Engine Hung Start</td>
<td>Y</td>
<td>3</td>
</tr>
<tr>
<td>3:50</td>
<td>#1 Hydraulic Shutoff Valve Circuit Breaker (CB) Trips</td>
<td>Y</td>
<td>4</td>
</tr>
<tr>
<td>10:10</td>
<td>#1 Fuel Gauge Hard Over</td>
<td>N</td>
<td>5</td>
</tr>
<tr>
<td>15:25</td>
<td>Co-Pilot’s Attitude Situation Indicator (ASI) Fails During Takeoff</td>
<td>Y</td>
<td>6</td>
</tr>
<tr>
<td>17:15</td>
<td>Multiple Bird Strikes on Left Side - Both Engines Fail</td>
<td>N</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 3: List of Events Including Approximate Time of Initiation

4.3. RESULTS AND DISCUSSION

The results and discussions below serve to provide the reader with insight into the knowledge and creativity required by a FE in a Hercules aircraft. In each scenario, the fault is described, accompanied by the information provided to the FE, his actions, the correct response, and a brief analysis of the observations made.

4.3.1 Engine Hung Start - Fault #3

The first recorded trial is a #2 engine stalled start scenario. When an engine “hangs” or stalls, it is usually due to: a) poor fuel scheduling by the Fuel Control Unit (FCU); b) clogged Fuel Filter; or c) improper functioning of the Temperature Datum (TD) System. This fault is indicated by the Turbine Inlet Temperature (TIT) and the engine
59

RPM running at reduced levels (approximately 35% to 50% of normal). Because the FE constantly monitors these gauges during engine start-up, he immediately identified the fault. This is evidenced by the AC asking “You got a clock running on this one?” meaning that he had noticed the engine starting slower than expected. When the instructor asked what the probable fault causes were, the FE correctly cited improper fuel scheduling by the FCU as the most probable source. The instructor then proceeded to advance time to permit the repair of the unit, and the crew successfully restarted the engine. Repairing such a fault in “real” time might be as simple as changing the fuel filter (20 min), or as involved as testing the FCU and possibly ordering parts, which could take weeks. If the fuel filter was not clogged, the FE also has the alternative of attempting an enriched start (higher fuel to air ratio) to evade repairing the FCU. This is a simple and common fault, requiring little troubleshooting effort by the FE. It was most likely inserted as an exercise to ease the FE into a “problem solving mode.”

4.3.2 Hydraulic Shutoff Valve Circuit Breaker (CB) Trips - Fault #4

In this fault, the hydraulic pressure would not bleed off when set to do so after the #1 engine had started and reached 72% (of full) RPM. This is indicative of the bleed-off pumps operating incorrectly. It is a fault which is not immediately noticed, as it occurs after the engine has successfully started, and can potentially create subsequent difficulties if not diagnosed. The most probable cause for this event is a faulty circuit breaker, which can be quickly remedied. However, if the circuit breaker is not the problem, then the prognosis becomes very involved: the valve cannot be closed without first shutting off the engine. On a real aircraft, the FE might try banging on the valve to see if it seized, but he
would most likely have to order parts and then perform a couple of hours of work to get the system back on line. This fault is slightly more difficult than the previous one, because an inexperienced FO might become too involved in his taxi check to be aware of the gauge readings located directly in front of him. Fortunately, the FO was experienced and immediately reported the snag to the rest of the crew. What is interesting however, is the conversation that followed involving the circuit breaker which had popped for engine #1. The FO mistakenly called #2 engine's circuit breaker (CB) popped and a good amount of confusion ensued. He realized that there was an error in his call, and remarked, “What the hell?” as he announced the snag. After a bit of deliberation, he realized the CB popped for engine #1 and not for #2. It is possible that the FO would have not made the same mistake had the CBs been individually numbered instead of simply being grouped together.

The crew subsequently recognized that the problem was simple and continued with their pre-taxi check. In particular, the FE was most anxious to continue. This fault was also simple to diagnose, and was most likely inserted to ensure that the FO was alert.

4.3.3 Fuel Gauge Hard Over - Fault #5

As the session picks up momentum, the faults become more intricate and difficult to detect. Subsequent to the FE calling the pre-takeoff check during taxiing, the instructor inserted the Fuel Gauge Hard-Over [High] fault, which is usually caused by faulty wiring. The timing of this fault was essential, as the instructor intended to determine how well the FE divided his attention between the standard operating procedures (SOP) and the instrument panels. The FE immediately noticed a problem with the fuel gauge for tank #1, and directly confronted the onboard instructor:
FE: You guys [instructors] probably have a snag in there, but it’s almost normal...when you start taxiing, the levels change.

Instructor: [unintelligible]

FE: It’s not hard over no...The only time they [dials] flip this hard over one way or another is when they’re erratic. So unless you guys [instructors] can make ‘em go hard over, I’m not going to pull. [the circuit breaker]

The FE explains to the instructor why he believed the snag was due to taxiing. He recognized that the most common source for such an error was faulty wiring, and he indicated a correct course of action for this cause. In this particular scenario however, the FE made an incorrect diagnosis, and did not consider a less common source for the fault; water in the tank (but this could be partially blamed on the quality of the simulator itself, which the FE was quick to indicate). The FE should have recognized the second possibility, but instead shrugged off a question involving this by replying, “sometimes they [fuel gauges] do stuff like that and they come back in a half hour or so. Saves a lot of maintenance.”

If there was water in the tank, there might have been a problem taking off with a hypothetical large weight imbalance. Also, in sub-zero temperatures, water in the tank would have posed a greater threat, as there is a possibility for ice crystal formation. It should be noted however, that this was not the FE’s “check ride” in the simulator and he was most likely a little more “relaxed” than he normally would be. Finally, although the FE did not pinpoint the exact nature of this fault as sought by the instructor, it would not have been a problem in a real mission, since determining whether or not there is water in the tank involves a simple process of checking the amount of condensation in each tank, which he would have done prior to entering the aircraft.
4.3.4 Co-pilot’s Attitude Situation Indicator (ASI) Fails During Takeoff - Fault #6

A co-pilot attitude situation indicator fault is caused by either the failure of the instrument, or by a plugged static pitot tube. In this test, the instructor selected a failed instrument, indicating that it was broken by sticking at 40 KIAS (knots indicated air speed). The only option available to the crew is to abort the flight, allowing the snag to be repaired but delaying the flight schedule significantly. This fault was also inserted to determine how well the FO divides his attention amongst his instruments.

During the takeoff run, the FO’s ASI increased and remained at approximately 40 KIAS. The FO immediately called an abort of the takeoff, and then briefly discussed the possible failures. In this case, the fault was not mission-threatening, as the pilot’s ASI was still fully functional. The FO however made a wise decision to abort the take-off, as there is little reason to continue when one can repair a snag so early in the run. In the debriefing, it was determined that another factor was considered by the FO. The chances of the pilot’s ASI failing would have dramatically increased (by instructor intervention) had the FO decided not to abort the flight. In order to be able to cope with the rest of the flight, the FO reduced his potential workload by choosing to repair the snag on the ground.

4.3.5 Multiple Bird Strikes on Left Side - Both Engines Fail - Fault #7

Birds are a pilot’s nemesis; the pilot can do nothing to avoid them, but the damage they can impart to an aircraft can be quite severe. This fault scenario involved multiple bird strikes to the left side of the aircraft (both engines) at a pre-set height of 50’ above
ground level (AGL). As birds are ingested into an engine, they can damage the compressor and the turbine, often resulting in a fire or turbine overheating. Bird strikes tend to occur at very demanding times for the crew (during takeoff or landing), but are usually easily detectable. Symptoms of a bird strike involve loss of torque, indicated by instruments and a distinct audible noise. Accompanying the torque loss is an increase of the TIT and large amounts of vibration. In these situations, the crew has little time to respond to the problem before the engines go down, and during this time, they have to prepare for an emergency landing while addressing both torque and potential fuel imbalances. The fuel imbalance is caused by the left engines using up considerably less fuel due to their failures. The crew is only permitted to land with a maximum imbalance of 1500 lb., which can occur quite quickly during take-off when the thrusters are fully extended. A main consideration for the crew is the landing gear; if it has to be lowered, they have to do so prior to shutting down both left engines, which serve the utility hydraulic system that lowers the gear. However, lowering the gear also generates extra drag which is a great burden for the remaining two engines while carrying a full load of fuel. To counteract these problems, the crew has the option of jettisoning fuel.

During take-off, ATC informed the crew of excessive bird activity. They did not seem surprised when the #2 engine began to fail. The FE mentioned that the engine was in mechanical (i.e., mechanically governed), and that the torque was wandering, but still running. When asked if engine #1 was running well, the FE continued on his #2 engine observations - “Still got overheat in #2 here. It's going to blow itself off the wing in a minute.” As the AC was trying to gather his information, he believed it would be easier
to shut the #2 engine down, and concentrate on flying. His justification for this decision was that he was still getting power out of #1, so he could afford to shut down #2. Realizing that #1 was going down as well, he also ordered that the fuel be dumped down to 10,000 lb. to prevent an imbalance and lighten the load on the two remaining engines. Once the crew noticed that smoke was pouring out the back of engine #1, they decided to lower their altitude and the FE suggested pulling the circuit breaker to turn on the high boosters until the engine gave out. After the boosters were ignited, smoke continued to bellow out of the engine, and the AC ordered that the #1 engine be turned off. The FE alerted the AC that he had forgotten about the landing gear, and the AC checked himself and asked for the post-takeoff check list. Had they continued on, the crew would have not been able to automatically lower the landing gear, and would have been forced to do so manually. In this situation, however, the choice of automatically lowering the gear hurt rather than aided the crew. The extra drag caused by the lowered gear created more of a problem than lowering the gear manually; essentially, the physical effort saved by using the automatic systems to lower the gear did not justify the added effort on the aircraft’s two remaining engines. Later in the fault, as the crew awaited landing, the FE noticed that the fuel was not dumping as quickly as expected. This was a result of fault #4 which occurred earlier in the session, and was forgotten. It was fortunate that the crew was flying in simulated warm (+20°C) weather. Had they been flying in sub-zero temperatures, the water in the tank may have formed ice crystals, which would have compounded the problems encountered by the crew.
4.3.6 Summary

As the instructor indicated at the end of the session, referring to the bird strike fault:

"Nicely done, what we’ve done here is give you a really complex scenario. A lot of things went wrong. You have to make some choices about runway length, and it’s a bit time compressed, so it is important to know what to do and what not to do..." The crew performed well together and were able to detect, diagnose, and remedy faults quickly and accurately. Although the session was not the most challenging of the base’s arsenal, the combination of the relatively simple faults with the extremely short duration of the flight (two hours) made it a challenging exercise.

4.4. IMPLICATIONS OF THE STUDY’S FINDINGS

The implications of the results of the study can be separated into two categories: a) Choice of Scenarios; and b) Interface Design. Each of these will be considered.

4.4.1 Implications of the Choice of Scenarios

The scenarios selected directly tested the FE’s abilities to effectively monitor instruments and diagnose faults simultaneously. This session was focused on ground faults, accompanied by one difficult fault in the air. One example of a fault which would have given the FE additional task and resource allocation duties is a nose gear extension failure. Although this fault was originally in the session itinerary, the instructor chose not to insert it during flight, as the FE was already quite busy. More subtle faults in the air
such as slow fuel or oil leaks, or oil bubbling would also challenge the FEs without overloading them. The implications which can be drawn from the protocol analysis regarding choice of scenarios are as follows. The scenarios must be chosen so as to follow a "naturally destructive path." There should be one fault which propagates into other, more serious faults. In the events observed, there was very little connection between faults, and there was little for the FE to do in terms of deciding on viable solutions which were dependent upon previous solutions. The only decision made by the FE which propagated to a second fault was the fuel gauge hard over fault affecting aircraft control during landing. This pair of faults does not fall into the category described above, as the fuel gauge did not directly produce the bird strike fault.

4.4.2 Implications for Interface Design

Recommendations for the redesign of the physical cockpit layout have been proposed in Chapter 3. This portion of the paper concentrates on recommendations for the redesign of instruments, and the incorporation of instruments not currently available. The recommendations made below did not solely arise out of the session discussed, but are the result of an accumulation of observations made over three entire simulator sessions.

- The most important implication for design that can be determined from in flight observation of a crew involves incorporating trend and instantaneous rate information. This information would provide the FE with an easy gauge for how fast levels in the aircraft are changing. Currently, the FE can utilize the HFEL
(Appendix B) to record information once an hour - or can cleverly use available space to record information once every quarter hour. This technique, however, still involves a bit of number crunching which is not infallible during stressful situations. An example of providing a graphical display which provides trend information and other higher order information can be seen in Vicente and Rasmussen (1990). In the bird strike fault, for example, trend information would have been very helpful in determining how fast (if at all) the tanks were dumping fuel.

- Replace the dials in the status gauges to better indicate the situation at hand (more context sensitive). Specifically, if the range of the component is 0-50, a dial capable of measuring 0-500 is much less useful than one which measures 0-60. This implication is the result of previous sessions.

Therefore, what is learned from the analysis of the protocol, is that FEs lack trend and rate information. When the first Hercules C-130 aircraft rolled out onto the tarmac, designers had very little space with which to work, and were forced to compensate for trend information with the HFEL. With the advent of the glass cockpit, reasons for omitting trend information no longer exist. The remaining question becomes how does one best present this information.
CHAPTER 5: ANALYSIS OF C-130 ENGINEERING SYSTEMS

5.1. Analysis

This chapter describes the system analysis that preceded the design of the EID interface for the engineering systems of a Hercules C-130 E-H model aircraft. The main focus of this portion of the thesis will be on describing the AH representations that were developed to describe the various engineering systems: namely the Fuel and Engine Systems (containing oil, hydraulics etc.). Although both systems can, in principle, be displayed in the same AH, they were separated to facilitate both discussion of their contents, and ease of display. These representations were subsequently used to identify the information content and structure of the interface, to be described in the following section.

5.1.1 Work Domain Representation

The first step in developing an EID interface is to create a multilevel representation of the Aircraft Engineering Systems (AES), structured according to two dimensions: an AH progressing from functional to physical system models; and a part-whole hierarchy progressing from the whole AES to its individual components. This two-dimensional problem space is graphically represented in Figure 8. The views that were adopted for the Engine System are prefaced with the label “E”, and those for the Fuel System are labeled with an “F”. Cells labeled with an “X” or with “N/A” were not found to be very useful for
Figure 8. Representation of the AES According to Part-whole/Means-ends Hierarchies
this particular system. For example, the top right cell (Functional Purpose - Component) marked with an “X” would be useful for an engineer initially designing a component, but it is not very useful from a systems perspective. Cells marked with an N/A indicate views within the problem space that could have been incorporated via video feeds, but were omitted due to the specific nature of the system being designed for. FEs in a C-130 are able to personally inspect many of the aircraft components because they are exposed - a luxury not available to commercial crews. Thus, the FE can request that the Load Master or another crew member check the appearance of a component, thereby negating the necessity of expensive video equipment.

5.1.1.1 Part-Whole Hierarchy

Beginning with the part-whole dimension, five levels of resolution were adopted: component; sub-unit; unit; subsystem; and system. Throughout this section, graphical representations of the cells being referenced in the text have been incorporated in the left margin to help the reader follow the discussion. Objects which reside in the component level include valves, fuel nozzles, oil coolers, tanks, etc. As is exemplified in Figure 9, this level of resolution is highly detailed, but not more so than the physical components that the crew would have to work with. A simple example to illustrate this point is the “Fuel Control Unit,” which is displayed as a single component. This component is composed of many valves, springs, governing systems, and so on, but these were considered too fine a level of resolution for our purposes, and were thus not represented. The next level in the part-whole hierarchy is the sub-unit level, which was created through a meaningful aggregation of components. For example,
the various components which together serve to control the amount of oil stored in an
engine's oil tank, are grouped together to form the oil tank sub-unit. When the sub-units
are aggregated in turn, the unit level is obtained. To illustrate, all the sub-units
which work together to provide an engine with fuel are grouped together.

Moving one level to the left again focuses attention to the subsystem level, which for an
engine is simply comprised of the oil, fuel, gearbox, compressor, combustion,
and turbine subsystems. This aggregation is displayed in Figure 10. Finally, the
highest level in the part-whole hierarchy is the system level, shown in Figure 8. This level
represents the entire Fuel System as well as each engine as single objects. As
will be shown below, defining the various levels of the part-whole hierarchy
provides a mechanism for operators to alter their focus of attention; moving from less
detailed views of the plant (e.g., subsystem), to more detailed views (e.g., component),
and vice-versa.

5.1.1.2 Abstraction Hierarchy

The AH, which is conceptually orthogonal to the part-whole dimension, consists of
five levels of description, as shown in Figure 8 and specified below.

Functional purpose. Objects at this level of abstraction correspond to system goals, and
therefore are appropriately described only at the "system" level of the part-whole
hierarchy. As shown in Figure 11 (Views E1 and F1), there is one main goal for
each of the Fuel and Engine Systems. The Fuel System’s goal is to provide enough fuel to

---

1 The cellophane overlays are the result of a secondary analysis of the information currently provided to the Flight Engineers and will be
described in detail in section 5.1.2
Figure 10: Example of the Subsystem Level in the Part-whole Hierarchy for the Engine System
Functional Purpose - Engine System

View EI

Provide Required Thrust

View EI

Provide Required Thrust

Provide Required Thrust

Provide Required Thrust

Functional Purpose - Fuel System

View FI

Provide Enough Fuel to Complete Mission

Figure 11: Views EI and FI According to Part-whole/Means-ends Hierarchy
fly to a certain predetermined destination; and the Engine System’s goal is to provide sufficient torque with a corresponding Turbine Inlet Temperature (TIT).

At this level of abstraction, the entire AES can be described in terms of the status of these two goals.

*Abstract function.* This level describes the entire system in terms of first principles (i.e., mass and energy conservation laws). As is shown in Figure 8, this level of abstraction can be meaningfully applied at two levels of decomposition for the Engine System (system and subsystem) and at three levels of decomposition for the Fuel System (system, subsystem, and unit).

With respect to the engines, at the “system” level of aggregation (View E2, Figure 12), each entire engine can be described as three mass sources connected to a single mass sink (the atmosphere) through a mass transport. Similarly, each engine can also be described as three energy sources connected to a single energy sink through an energy transport. At the “subsystem” level of aggregation (View E3, Figure 13), both the mass and energy transports are divided up into their separate streams, and the figure begins to resemble a typical engine. The polygonal shapes were purposefully chosen to resemble the standard compressor and the turbine symbols for identification purposes only.

The Fuel System at the “system” level of aggregation (View F2, Figure 14) can also be described in a like manner to that of the engine system, with a single source connected to a sink for both the mass and energy balances. The “subsystem” level
Figure 12: Abstract Function - System Level in the Part-whole/Means-ends Hierarchy for the Engine System
Figure 14: Abstract Function - System Level in the Part-whole/Means-ends Hierarchy for the Fuel System
for the Fuel System (View F3, Figure 15) sees the mass source being separated out into left and right wing sources, as well as having the transports parsed out into their different paths. At this level, the mass and energy sinks are also described in terms of the left or right side of the aircraft. The next applicable level which can be discussed in the abstract function terminology is the fuel “unit” level. As can be seen in Figure 16 (View F4), the left and right wing sources of fuel have been decomposed into the individual tanks. The transports of fuel have also been extracted to indicate both the direct tank-to-engine feeds, as well as the manifold to engine feeds. More importantly, the transports of fuel have been marked as “outer” and “inner” mass transports. The “outer” engine transports include the tanks and manifolds that direct fuel to an engine, while the “inner” engine transports contain fuel that has actually been sent to the engine (i.e., past the fire shut-off valves labeled with the prefix G or H in Figure 9). This is the clearest indication of how the Engine and Fuel System hierarchies mesh to effectively form one overall system. In the figures presented, the Fuel System’s sinks are the sources of fuel to the Engine System. This fact becomes important as one attempts to prevent a potential problem from propagating any further than necessary between the two systems.

The most important benefit of the abstract function level of abstraction is that it allows the FEs to reason about the AES in terms of first principles. This is particularly important under abnormal situations, since under such situations, reasoning based on the normal functioning of physical components is unreliable.
Figure 16: Abstract Function - Unit Level in the Part-whole/Means-ends Hierarchy for the Fuel System
Generalized Function. Flows and storage of different liquids, gasses, and heat are described at the next level of abstraction. As shown in Figure 8, four levels of decomposition/aggregation were defined for the Engine System: subsystem; unit; sub-unit; and component. Within the Fuel System, three levels were defined: unit; sub-unit; and component. The generalized function - engine subsystems level is presented in Figure 17 (View E4), and shows how the mass sources are split up into distinct fluid components. The general flows of air, oil, and fuel can be followed using the information provided in Figure 17, which can also be used to isolate losses/gains of mass and energy for a particular unit. Note that the diagram only displays one engine, as the other three are exact duplicates of the remaining portion of the Engine AH. As the subsystem is decomposed further (View E5, Figure 18), the various engine subsystems identified are parsed into units. For example, the Reduction Gear contains independent units responsible for both stopping and providing the proper shaft speed for the propeller. The separate subsystems presented in view E5, such as the Reduction Gear, are individually labeled and each of their units are enclosed within a larger rectangle. This method of presenting information simply facilitates the reader's understanding of how the subsystems and units are connected. Further decomposition along the generalized function layer results in the sub-unit level of aggregation. Figure 19 (View E6) demonstrates how the flow of oil is affected by the various sub-units which belong within the oil units identified above. In this level, one can see that the oil tank supplies oil flow, a reservoir for the oil, and a method
Figure 17: Generalized Function - Subsystem Level in the Part-whole/Means-ends Hierarchy for the Engine System
Figure 18: Generalized Function - Unit Level in the Part-whole/Means-ends Hierarchy for the Engine System
of pressurization. A final decomposition within the generalized function layer reveals the component level, which for ease of description, has been divided into two major sections: the oil and the fuel components (still within the Engine System). The components, which combine to provide the functions inherent to the various sub-units, are described in terms of flows of mass and heat in this cell of the problem space (Views E7o and E7f, Figures 20 and 21). The actual topology of the components is shown here, and the functional role of each component is explicitly represented. The component level within the Engine indicates how the fuel and oil are respectively modified in terms of flows of mass and heat, as each traverses its course within the engine.

The Fuel System can be described along three levels of aggregation/decomposition for the generalized function level of abstraction: unit; sub-unit; and component. For the unit level (View F5, Figure 22), the information provided is similar to the abstract function - fuel unit level previously discussed. This is the case because the only mass being manipulated within the Fuel System is the fuel. The difference between the two layers of abstraction is that the abstract function level describes the system in terms of mass and energy balances, and the generalized function level specifies the fluid as being fuel. When the unit level is decomposed into its sub-units, the resultant diagram begins to resemble the P&ID for the system (View F6, Figure 23). It is somewhat difficult to ascertain how the sub-units aggregate to form the unit level, but upon closer examination, the different groupings should become evident. For example, all of the boxes on the left of the diagram labeled with a “1”, except for the “Engine 1 Flow” box, are all part of the “outer engine 1” flow. The “Engine 1 Flow” box is a place holder
Figure 20: Generalized Function - Component Level in the Part-whole/Means-ends Hierarchy for the Engine Oil System
Figure 21: Generalized Function - Component Level in the Part-whole/Means-ends Hierarchy for the Engine Fuel System
Figure 22: Generalized Function - Unit Level in the Part-whole/Means-ends Hierarchy for the Fuel System
for the inner engine 1 flow previously discussed. A decomposition of the sub-units reveals the component level for the Fuel System (View F7, Figure 24). This level provides flows to and from groups of components, which are useful when trying to narrow the possible cause(s) of an abnormal flow measurement. Although somewhat messy and confusing to look at, the information provided is complete, and is in fact only as complicated as the actual system that it is intended to describe. Some features were incorporated into this diagram to aid in the understanding of how the components fit together to form the sub-units. All the objects within a dotted rectangle combine to create the respective tank sub-units. Note that the manifolds have also been expanded to include all of the valves and other equipment which affect the flow of fuel through the respective manifold.

Although not required by the AH analysis, two additional representations were incorporated to aid the reader in understanding the mapping between the generalized function and physical function levels for the engine (described below). The diagrams (Figures 25 and 26) are redundant, in that they repeat information presented in the two levels that they link. Nonetheless, they may help to provide an insight into the complexity of each of the components in terms of the multiple roles that they serve. There are two rows in each of the diagrams. The lower row represents the components, whereas the upper row represents the functional role that each respective component plays within the system. Due to the one-to-many mapping between a component and the multiple roles that it might serve, the relationship between the generalized function and physical function levels can be better understood if presented in a
Figure 25: Link Between Generalized and Physical Function Levels in the Part-whole/Means-ends Hierarchy for the Engine System
Figure 26: Link Between Generalized Function and Physical Function Levels in the Part-whole/Mean-ends Hierarchy for the Engine System
non-topological manner. For example, the Fuel Control Unit serves five separate functions. If simply presented in the customary manner, it might appear that there are five components and not just the one.

**Physical Function.** It is only at this level of abstraction that information regarding the state of the physical components themselves is first encountered (as opposed to the state of the functions that these components are intended to achieve). This level of abstraction is best viewed at the component level in the part-whole hierarchy. The resulting system views, shown in Figures 27 (Engine) and 28 (Fuel), are familiar P&ID representations of the two systems. At this level, the status of the various tanks, valves and pumps are represented.

**Physical Form.** Information can be obtained by the FE via a video connection to the aircraft components, or by personally inspecting them. Currently, the situation within a C-130 is such that the FE is able to personally inspect many of the aircraft components during flight. Although not included in Figure 8, this information can be very important, particularly when faults propagate between components that are functionally dissimilar, but physically proximate (cf. Bisantz & Vicente, 1994).

### 5.1.2 Information Representation

A parallel analysis was performed, whereby the information currently presented to the FEs was compared to the information content prescribed by the part-whole/means-end representation shown in Figure 8 (see Reising & Sanderson, 1996). The cellophane overlay on each of the figures (11-28) serve as an indication of how an FE would be able to obtain the information for the objects within a particular cell of the AH framework.
Figure 27: Physical Function - Component Level in the Parts-whole/Means-ends Hierarchy for the Engine System
Seven distinct modes of information presentation are available to the FEs. These are listed and described below in Figure 29, and their respective icons are also illustrated.

The Green Circle icon represents instrumentation already available to FEs which provide the information required, as prescribed by the AH analysis. In both the Engine and Fuel System analyses, it becomes apparent that the majority of the information currently directly presented to the FEs can be mapped onto the Physical Function - Component level of the hierarchy framework (Views E8 and F8) Figure 27 and 28. This is consistent with the technology, and with SSSI approach to interface design that was prevalent at the time of the aircraft’s construction. What is also interesting, however, is that the functional purpose - system level information is also directly presented to the FEs. This seems to indicate that the designers knew what information the FEs required to satisfy their goals, but may have been forced by the available technology to forego presenting all but the minimum information required, in order to conserve valuable cockpit real-estate. Included under the category of components marked with a green circle, is a sub-set of a special class of components that can be observed directly by the crew, but not through the use of instrumentation. A specific example is the refuel drain valve. When the aircraft is airborne, the dump valves are automatically set to open, and the drain valve is set to close. If the refuel drain valve is stuck in the open position, then fuel will be permitted to flow overboard, which will be observed by the crew. Although another information set could have been created for this type of component, it was decided that it be a part of the “green circle” group even though instrumentation is not provided.
<table>
<thead>
<tr>
<th>Colour</th>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Green</td>
<td><img src="image" alt="Green" /></td>
<td>Instrumentation already provided</td>
</tr>
<tr>
<td>Red</td>
<td><img src="image" alt="Red" /></td>
<td>Can be measured with additional instrumentation</td>
</tr>
<tr>
<td>Purple</td>
<td><img src="image" alt="Purple" /></td>
<td>Can be measured with current instruments, but requires manipulation of system configuration</td>
</tr>
<tr>
<td>Yellow</td>
<td><img src="image" alt="Yellow" /></td>
<td>Measureable on the ground only (not in flight)</td>
</tr>
<tr>
<td>Blue</td>
<td><img src="image" alt="Blue" /></td>
<td>Measurement attainable from current instrumentation, but requires calculation</td>
</tr>
<tr>
<td>Black</td>
<td><img src="image" alt="Black" /></td>
<td>Not measurable</td>
</tr>
<tr>
<td>Grey</td>
<td><img src="image" alt="Grey" /></td>
<td>Automatic Controller</td>
</tr>
</tbody>
</table>

Figure 29: Information Retrieval Methods and their Corresponding Symbols
Another information category consists of items that can be measured and manipulated only while on the ground. These are marked with a yellow diamond. An example of this type of component is the tank drain valve, which requires personnel to physically turn the valve located on the underwing. Although a flowmeter could be attached to each drain valve, the effort required would not be worth the gain in information, as a pre-take-off requirement is that these types of components are checked to ensure that they are in their proper position.

Blue pentagons represent information that is attainable from current instrumentation, but requires calculation to derive the desired measurement. Purple triangles are similar to pentagons, in that the desired information is attainable from current instrumentation, but requires manipulation of the system configuration to achieve the desired measurement. There are also many blue pentagons displayed in the upper levels of the analysis, which indicate that FEs are required to combine and calculate data based on the instrumentation provided. As one traverses the hierarchy analysis from bottom right to top left (i.e. from physical function - component to functional purpose - system), as a general rule, the amount of effort required by the FEs to obtain higher level information increases. This is indicated by the increasing number of pentagons and triangles. For example, there is currently only one method available to the FE to ascertain the amount of “outer engine” flow (see section 3.4), and that is by diverting the fuel to the pressure meter located in the crossfeed manifold. This also requires the FE to shut off all other crossfeed flows, as any other fuel in the manifold would skew the true pressure readings.
sought. This is an example of where a purple triangle would be placed on the cellophane overlay.

In order to circumvent this issue, the designers provided the FEs with another tool to aid them in obtaining the information required. This tool is known as the FEs logbook, and is described in depth in Appendix B. The FEs are required to maintain the logbook if they fly in excess of two hours (a minimum of one reading per hour thereafter). When an FE notices a potential problem, then he would tend to take more frequent measurements of a set of components to determine, for example, if there is an abnormally high rate of fuel loss in a particular tank. This method requires the FE to manually calculate actual rates of change, and compare them to ideal rates, which he must also compute. This process can become quite inefficient, time consuming, and cognitively taxing when the FE is faced with diagnosing an abnormality.

Yet another shape which appears frequently in many of the cell views, is the red octagon. It symbolizes a measurement which could be taken, but currently is not. It is the superset of the pentagon and the triangle representations. For example, the flow through a valve could be measured with a flowmeter by adding one near the valve. Of course, this would also require the addition of many measurement devices, which would provide the necessary data, but might prove to be uneconomical in terms of both cost and available power.

The remaining two icons represent components which operate automatically (black "star" symbol), and a subset of this group, which contains items that operate automatically and cannot be measured without extreme implementation cost (grey "plus" symbol). An
example of the former is the automatic fuel density correction feature of the Temperature Datum Valve. An example of the latter is the status of the mesh filter within a fuel filter.

In summary, these analyses have served two important functions. The work domain representation (Section 5.1.1) analysis identified eight distinct and meaningful AES views for each of the Engine and Fuel Systems. These depict the different views within a part-whole, means-end problem space (see Figure 8). The other cells in Figure 8 were not found to be very useful for this particular system. This is consistent with the findings of previous research that has consistently shown that the two problem space dimensions are coupled in practice - abstract views are more meaningful at higher levels of resolution, whereas concrete physical views are more meaningful at detailed levels of resolution (Vicente, 1992b). The primary purpose of this analysis was to identify the content and structure of the information that is to be included in an EID interface. The second analysis provided insight into how well the current C-130 interface design provides the necessary information to the FEs, as is dictated by this hierarchy analysis. It was shown that the information requirements are satisfied at the extreme ends of the part-whole means-end problem space, and are only moderately covered at every level in between. The lack of information at these intermediate levels represent "blind spots" for the crew. This means that if the system is not manipulated, or calculations are not made, to elicit information, a fault might be permitted to propagate and grow unnoticed until it is too late for the crew to recover. In the following section, the methods by which the views obtained were mapped onto the visual form of the interface will be shown.
CHAPTER 6: INTERFACE DESCRIPTION

Once the AH analysis of the engineering systems was completed, the information content and structure of the interface were defined. The next step was to map the representations onto specific visual forms. This purpose of this chapter is to describe the resulting EID interface developed for the Hercules C-130, including how it relates to the AH hierarchy analysis, and how it can potentially be used in three potential fault situations.

6.1. GENERAL DESCRIPTION OF PROTOTYPE EID INTERFACE

Figure 30 illustrates the viewport configuration utilized in the prototype EID interface that was developed for the Aircraft Engineering Systems (AES) of a Lockheed C-130 Hercules, model E-H. The workspace is divided into six distinct display regions, each serving a unique role in communicating information to the FE. Iconic versions of Figure 30 appear in the left-hand column throughout the text, with one highlit region which corresponds to the viewport being discussed.

Figure 30: Viewport Layout in EID Aviation Display
Each of the viewports introduced above are linked to the AH analysis by the information which they contain. Following the principles of EID, the five displays (Viewports 1-4b) were designed to include all eight engine and fuel system representations identified in the previous chapter (see Figure 8). The fact that there are less displays than system representations results from the ability to integrate two or more different views of a system into a single visual form (Vicente, 1992b). Figure 31 describes the links to the Engine System AH according to viewport number, and Figure 32 illustrates the links to the Fuel System AH by viewport number. The colour of each view in Figures 31 and 32 corresponds to the viewport colour displayed in Figure 30. For example, in Figure 32, parts of views F1-F6 in the part-whole means-end hierarchy have been integrated to create the FUEL SYSTEM OVERVIEW display. This is indicated by the colour green, which corresponds to viewport 2, overlaid on cells 1-6 in Figure 32. In cells where there are two or more colours, the information identified in the AH is located in more than one viewport. For example, in Figure 32, cell 7, the colours yellow and pink indicate that the physical purpose - component information (i.e., component flows) can be found in both Viewports 4 and 4b.

The six display regions indicated in Figure 30 are:

Viewport 1 - AES PERFORMANCE: Figures 32 and 33 illustrate the connection between the information provided in the interface, and where this information requirement exists in the AH analysis. This display is shown in Figures 33 and 34.
<table>
<thead>
<tr>
<th></th>
<th>SYSTEM</th>
<th>SUBSYSTEM</th>
<th>UNIT</th>
<th>SUBUNIT</th>
<th>COMPONENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>FUNCTIONAL PURPOSE</td>
<td>View E1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ABSTRACT FUNCTION</td>
<td>View E2</td>
<td>View E3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GENERALIZED FUNCTION</td>
<td></td>
<td></td>
<td>View E4</td>
<td>View E5</td>
<td>View E6</td>
</tr>
<tr>
<td>PHYSICAL FUNCTION</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PHYSICAL FORM</td>
<td></td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Figure 31: AH Links to Viewports for the Engine System
Figure 32: AH Links to Viewports for the Fuel System
Viewport 2 - **ENGINE SYSTEM OVERVIEW**: Various examples of this display are shown in Figures 33, 35, and 36. It contains information relevant to each engine’s goals, that are associated with the higher levels of the AH (Figure 31).

Viewport 3 - **FUEL SYSTEM OVERVIEW**: Examples of this display can be found in Figures 33 and 36. Similar to Viewport 2 in purpose, the FUEL SYSTEM OVERVIEW contains information at the higher order levels of the AH, relevant to the fuel system's goals (see Figure 32).

Viewport 4 - **SYSTEMS DISPLAY**: This display contains information located in views E4-E8 of the AH, that are relevant to the AES components' flows and settings. Figures 34, 41, and 47 contain examples of this viewport.

Viewport 4b - **SYSTEM INFORMATION**: This display provides information required in views E2-E7 of Figure 31, and view F7 in Figure 32. It was created to facilitate movement between the overview displays (Viewports 1-3) and the systems display (Viewport 4). Examples of this viewport are Figures 42-47.

Viewport 5 - **THRUST DISPLAY**: This display was incorporated in the interface for demonstration purposes only. Four “thrust levers” set the thrust for each respective engine to either reverse (“REV”); idle (“IDLE”); cruise (“CRUISE”); or take-off (“FULL”). This display would not exist in a ‘real’ interface, and thus will no longer be discussed in this thesis.
Figure 33 is an illustration of the interface designed for the aircraft engineering systems of the Hercules C-130 aircraft, using VAPS™ prototyping software on a SGI Indigo workstation. The configuration displayed in this Figure shall henceforth be referred to as the 'default view'. Now that the viewports have been introduced in relation to the AH, and the interface has been presented, each of the viewports will be described in detail.

6.2. AES PERFORMANCE DISPLAY

The AES Performance displayed in the upper left corner of the interface (Figure 33) indicates how well each of the AES are cooperating to propel the aircraft to its desired destination. A close-up of this portion of the interface is provided in Figure 34. A satellite image with overlaid relevant cities and military installations describes the territory being flown in. Gray concentric circles, combined with a 'zoom' rotor, provide a goal-relevant frame of reference for object distances. The most important features of the AES performance display are the red circles which shrink proportionally to the distance that the aircraft is able to fly, incorporating factors such as fuel levels, engine efficiencies, airspeeds, outside air temperatures, and altitudes, etc. This distance is calculated based on current flight conditions, but can be upgraded to predict maximum flight distances using a pre-programmed flight path and meteorological conditions which the aircraft encounters during flight. A solid red circle alerts the crew as to how far they can safely attempt to fly, while a dashed red circle indicates the absolute maximum distance that can be flown before running out of fuel. This display precludes the necessity to perform detailed calculations and guess-work regarding the available distance that can be flown. Therefore,
it is postulated that the display would be invaluable in situations where the crew is forced
to divert to another airport or military facility.

The red circles (indicating total remaining flight distance) can also be consulted
when refueling. As tanks are replenished, the circles expand to reflect the additional fuel
taken on. Although this does not directly benefit FEs of the Hercules during normal
operations (i.e., in flight), the functionality of the display allows for such an occurrence.

Although the AES Performance display is not intended to serve as an alarm
display, the crew should be able to obtain a ‘feel’ for the rate of decrease of available flight
distance, according to flight levels, thrust settings, and other factors. With experience,
they could potentially notice an abnormal gradient in the available flight distance and
commence a fault search before being notified by an alarm.

![Figure 34: Close-up of the Aircraft Engineering Systems Overview Display](image_url)
6.3. ENGINE OVERVIEW

The ENGINE OVERVIEW, located in the upper-middle of Figure 33, is the most important display for the engine systems of the Hercules. It contains the 12 most important parameters for each engine, which when combined, provide a quick and accurate description of the performance of each engine. Within the AH, the Engine Overview display contains views relevant to the higher order engine functions. The parameters are organized within a polar star containing 12 spokes (one for each parameter). Figure 35 provides an example of this arrangement.

Figure 35: View of an Engine Specific Polar Star Diagram
The specific clockwise ordering is consistent with instrumentation currently employed in Hercules C-130 cockpits. The parameters involved are:

a) Engine torque (Torque):

b) Propeller rate of revolution as a percentage of normal (%RPM)

c) Turbine inlet temperature in °C (TIT)

d) Fuel flow to the engine nozzles lbs/hr (FF)

e) Average of engine and gear box oil temperature (°Cmill)

f) Engine oil pressure (P_{eng})

g) Gear box oil pressure (P_{gear})

h) Amount of oil in the oil tank (L_{oil})

i) Percentage of maximum which the coolant flaps are open (%_{flap})

j) Bleed air pressure (P_{air})

k) Engine hydraulics indication (Hyd)

l) Engine electronics indication (Elec)

Numerous benefits can be realized by organizing critical engine parameters in this manner. The most evident is the emergent feature inherent in any polar star diagram (see Section 2.3.1). Specifically, any deviations from a "normally" operating engine (represented by the green dashed dodecahedron) can be easily perceived by the FE. Actual engine performance is presented by a solid dodecahedron superimposed on the polar star diagram. Moreover, the way in which the polygon deviates from the ‘normal’ (i.e. ‘goal’) state is intended to provide an indication of the fault source. For example, faults which
involve low fuel flow to an engine should contain common deviations from the normal\(^1\) state, thereby expediting and increasing the accuracy of diagnosis. To aid in identifying performance boundaries, alarm levels have been incorporated into the polar star display in the form of tick marks along each spoke axis: yellow for cautions; red for warnings. The colour of the entire polygon which indicates actual engine performance also changes to the alarm level corresponding to the worst engine parameter.

Another emergent feature derived from the adjacent placement of the four engine overviews is the relative size of each engines' polar star. The four thrust states that each engine can obtain are: reverse; engine idle; cruise; and full power. In the prototype, the overall area that the polar star encloses is directly proportional to the aircraft thrust setting.

When all of the above features are combined, a FE should be able to simultaneously diagnose the state of all four engines as well as their relative performance. Examples of how the ENGINE SYSTEM OVERVIEW display can be used in conjunction with the other viewports are discussed section 6.7.

6.3.1 User Interaction

The FE can interact with the ENGINE SYSTEM OVERVIEW display by selecting any engine spoke, or by selecting the center of each engine's polar star. When the user clicks on a spoke, the SYSTEMS DISPLAY (Viewport #4 in Figure 30, the ‘normal’ state is identified as the expected performance of an Allison T56 A15 engine, not an ideal turboprop engine.

\(^1\) The ‘normal’ state is identified as the expected performance of an Allison T56 A15 engine, not an ideal turboprop engine.
described in section 6.5) automatically changes to “zoom in” on the components responsible for producing the information. The SYSTEM INFORMATION display (Viewport #4b in Figure 1, described in section 6.6) also changes to provide information corresponding to the spoke selected. If the FE selects the center of the polar star display, the system responds by presenting only engine wide parameters within the SYSTEM INFORMATION display, and removes all presented information in the SYSTEMS DISPLAY. To return to the ‘default’ view, the FE must click on the ‘Overall Fuel System and Flow Indication Button’ which is described in the following section.

6.4. FUEL SYSTEM OVERVIEW

The FUEL SYSTEM OVERVIEW, Viewport #3 located in the middle left of Figure 33, contains functional information relevant to the higher order fuel system requirements in the AH (see section 5.1). A close-up of this viewport is provided in Figure 36. It has been organized in the interface to logically progress from the state of the aircraft’s entire fuel supply, to the relative wing-side supplies, and then to each independent tank. This format places the engines at the bottom of the display. Although this is contradictory to the fuel indications currently presented in a Hercules cockpit, it was decided that the hierarchical progression from the individual tanks to the eventual sum of the fuel system’s parts (and the corresponding fuel system’s goals), was better depicted in this format.
Figure 36: Close-up of the Fuel System Overview Display

The four levels of information (in increasing importance) provided in this display are: a) Fuel Flow (FF) to each engine, located at the bottom of Figure 36; b) tank fuel levels, and indications of flow to and from tanks (located above the FF indicators in Figure 36); c) wing side fuel amounts and fuel flows, including dumped or lost fuel (located above the tanks in Figure 36); d) overall fuel system amounts, flows, losses (intentionally or otherwise), and predicted amounts (this graphic is located at the top of Figure 36).

Each of the levels shall now be described in greater detail.
6.4.1 Engine Fuel Flow Indication

As previously discussed, the engine fuel flow indications are based upon standard CAF Hercules iconic engine diagrams (Air Transport Group, 1993, Vol. 2), and are located at the bottom of the FUEL SYSTEM OVERVIEW display (Figure 36). Each icon contains a bar which fills proportionally to the actual fuel flow being sent to the corresponding engine system. This is in contrast to the Fuel Flow spoke of the engine's polar star display which indicates the fuel flow being received by the engine nozzles.

Alongside the iconic engine diagram is a triangle which indicates the fuel flow requested by the engine (according to the thrust level). If the triangle is at the same level as the bar, then the expected amount of fuel is being sent to the engine by the fuel system. If the triangle is above or below the bar, then there is respectively insufficient or supplementary flow to the engine. Thus, by comparing the relative position of the triangle to the bar, an FE can quickly ascertain whether or not cross-feeding should be engaged to supplement an insufficient fuel supply to an engine, whether the engine and fuel system are operating as expected, or whether supplementary fuel to an engine can be diverted elsewhere. All of the main engine icons are equivalent and are physically located at the same level. The emergent feature of fuel flow bar height is yielded in this portion of the display; if all the thrust levers are located at a common setting, and the fuel systems are operating normally, the fuel flow bar heights of all the engine icons should be equal.

6.4.2 Tank Fuel Amounts and Limits

All eight tanks are represented along a common axis and have equal dimensions within the FUEL SYSTEM OVERVIEW display (above the fuel flow indicators in Figure
36). The area of the tank which is filled in blue corresponds to the amount of usable fuel in the tank. Even though the tanks have different initial amounts of fuel, the equivalence in their representation is essential for diagnosis purposes. At this level of abstraction, the most important parameter is the relative amount of fuel within the tank itself that can be used to feed an engine. As a secondary benefit, the task constraint of balancing tanks is also facilitated with these tank representations, through the emergent feature of the line created by connecting the levels of fuel in each tank.

Along the bottom of each tank icon is a bar indicating actual fuel flow from the tank, as well as a triangle, which indicates expected fuel flow from the tank according to the overall fuel system configuration. If the fuel flow increases beyond the green triangle (or goal state), a supplementary red bar is displayed which extends beyond the goal state to the actual flow rate (Figure 37, left tank). The triangle itself also changes colour to yellow or red, to indicate the discrepancy. The red bar extending from the goal flow rate was used to maintain a common description for fuel being dumped - in this case, via a leak in the tank itself. Any flow out of a tank that is less than that required by the fuel system, is indicated by a fuel flow bar shorter than the level indicated by the triangle (Figure 37, right tank). In such circumstances, only the triangle changes colour. This situation is indicative of a partially blocked tank outlet valve.

Connecting the tanks are enhanced piping and instrumentation diagrams that FEs are accustomed to consulting. The purpose of the piping is to indicate whether there is flow (of any rate) to or from a tank or manifold. Piping intended to direct fuel away from engines (i.e. refueling or dumping of fuel) is represented by a dark orange line for no flow;
and a bright orange thicker line, to indicate flow within the pipe (see Figure 38). The valves provided in the display are identical in appearance and relative location to those currently installed in Hercules C-130 Model E-H aircraft cockpits. The valves are intended to facilitate the transition between the prototypical display, and that which FEs are accustomed to. Although the valves along with the piping provide overview, rather than detailed, exact fuel system configuration (only the important valves are presented), the general system configuration displayed is sufficient for determining whether or not there is fuel within a particular manifold. At each extreme end of the tank representations are bar graphs which indicate fuel being dumped or unintentionally lost. The scale on these bars is artificially enhanced to clearly indicate small fuel losses. Any fuel flow deviations discussed above are also mimicked in these "dump flow" bars.

In the example shown in Figure 38, fuel is being cross-fed from Tank 1 to all the engines (indicated by the brighter green piping), and fuel is also being dumped from Tank 3 (indicated by the bright orange piping, as well as the filled red ‘dump flow’ bar.)
6.4.3 Wing Tank Amounts and Flow Levels

The second row of graphical icons in the FUEL SYSTEM OVERVIEW display (Figure 36) contains fuel relevant information corresponding to each side (or wing) of the aircraft. The “Left Wing Tank” is the sum of the fuel amounts of Tanks 1, 2, Left Hand Auxiliary (LH Aux), and LH External. The “Right Wing Tank” amount is similarly derived. The flow bar at the bottom of each wing tank icon is obtained by summing the
corresponding tank flows, and operates in the same manner as the tank flow bars discussed previously.

The most innovative feature of the wing tank level of graphical icons is the method used to connect the relative "Wing Tank" heights. A task constraint imposed on FEs is that the relative difference between the left and right wing tanks never exceeds 1500 lbs. To enhance situational awareness of this issue, a graphical representation was utilized to provide the state of the task constraint at a glance (see Figure 39). A triangle, which has its focal point at its corresponding wing tank level, identifies a region equivalent to the allowable level of fuel (meeting the task constraint) in the opposite tank. In this case, the region swept corresponds to 3000 lbs of fuel. When both triangles are combined, the triangles form a unique shape corresponding to the relative levels of fuel in each wing. If there is no space between the two triangles, the task constraint is being met. Moreover, the amount of overlap is proportional to the relative equality of the two wing levels. This concept can be better understood by examining Figure 39.

6.4.4 Overall Fuel System Amount and Flow Indication

This graphic, located at the top of the FUEL SYSTEM OVERVIEW display (Figure 36), serves multiple purposes. It is the most direct link to the AIRCRAFT ENGINEERING SYSTEMS display (Viewport #1) for the fuel system, and also provides the status of the highest order information, Figure 32 Views F1-F6, related to the fuel system. This information is displayed in the familiar form of a tank graphic. The sum of
Figure 39: Example of Possible Tank Balance Indicator States
the usable fuel in all of the aircraft's tanks is indicated by a filled blue bar, and the fuel flow out of the entire fuel system (both actual and expected) is displayed in the same manner to those previously discussed. There are two major enhancements to this graphic compared to that of a regular tank icon. The first (located to the right of the tank) is the addition of a label containing the actual amount of fuel (in lbs.) as well as a predictor for the amount of fuel in the tank in one hour's time. This predictor is calculated using ground speed, wind speed, thrust positions, current total fuel flow, and other factors. The addition of this simple and effective information provides the FE with another method of gauging the consumption efficiency of the aircraft. The second enhancement appears on the right side of the same tank graphic, providing the translation of the amount of fuel in the tank into a distance that can be flown by the aircraft using a predictor arrow, and units of km for tank gradations. This enhancement is also the simplest means of linking the AES display to the FUEL SYSTEM OVERVIEW. In the prototype, this value is simply based on instantaneous conditions, which are extrapolated to the proposed end of the flight. As GPS and computerized systems become more prevalent on Hercules aircraft, it would be possible to enter a proposed flight plan and destination in order to provide more accurate extrapolated flyable distances.

6.4.5 User Interaction

The FE can interact with the FUEL SYSTEM OVERVIEW in one of two ways: by clicking on a valve representation; or by selecting a tank or any set of piping. Clicking on a valve is equivalent to turning the valve to fully open, or fully closed, depending on its current state. When a valve is closed, the valve indicator is perpendicular to the piping,
and the piping is dark blue. When a valve is open, the valve indicator and piping are
collinear, and the piping is bright green. Changes in the status of valves, flows, and tank
amounts propagate throughout the entire interface. Thus, all displays of the same valve,
as well as all the components and systems they affect, are altered whenever any valve is
turned.

When an FE clicks on a tank, or on any piping, he activates a switch which alters
the view in the SYSTEMS DISPLAY (4a), and in the SYSTEMS INFORMATION
display (4b), which shall be discussed next.

6.5. SYSTEMS DISPLAY

The SYSTEMS DISPLAY (Viewport 4), located in the lower right portion of the
interface (Figure 33), contains information relevant to the lowest
levels of the AH for all of the engineering subsystems. The
information displayed here is context sensitive because it is
dependent upon the user's interaction with the other screens.

Once a particular portion of the SYSTEMS DISPLAY is selected, panning capabilities are
facilitated through the use of sliders, located along the left and top perimeter of the
viewport. Several examples of the information that can be displayed here will be provided
next.
6.5.1 *Engine System Zoomed*

If the FE clicks on any one of the spokes in any of the ENGINE OVERVIEWS, the SYSTEMS DISPLAY will automatically zoom in to the appropriate section of the engine system that corresponds to the chosen spoke. The available switches were discussed in Section 6.3.1. Once zoomed to a given section of an engine, the FE is given detailed physical information pertaining to fuel pressures and flows, and oil pressures and flows. The option of requesting service records of any component is also available. Service records contain the last snag (abnormal incident) reported on the component, the last date it was serviced, as well as an option for recording a recent snag (see 6.5.4.1).

6.5.2 *Fuel System Zoomed*

If an FE chooses to select any of the available switches in the FUEL SYSTEM OVERVIEW display (with the exception of the Overall Fuel System Amount and Flow Indication or Wing Tank and Flow Levels graphics) the SYSTEMS DISPLAY responds by zooming in on the relevant component. An example of this view is presented in Figure 40. In this example, the FE has chosen to obtain detailed information about tank 4. All information required, including any combination of pressure, temperature, flow rate, and tank amount are automatically provided. Service records of any component are also available as in the Engine System Display. If the FE chooses either of the fuel system overview graphics, the SYSTEMS DISPLAY alters to depict the entire fuel information similar to, but more detailed than, the FUEL SYSTEMS OVERVIEW. This view is necessary to properly re-configure and understand how the Fuel System operates. Once again, any manipulation of this display is reflected in the FUEL SYSTEM overview
display. If the FE is more interested in information about the general operation of the Fuel System within each wing, then he would click on one of the wing tank graphics in order to pull up a similar view to the general fuel system view above, with the exception being that it includes general manifold flow information. This view is useful when narrowing the search of a faulty component which is causing the flow of fuel within the fuel system to act abnormally. The last option available to an FE for changing the fuel system view is the “Fuel Balance” switch (triangles extending from the wing tank amounts), as shown in Figure 41. This switch was incorporated to facilitate satisfying another fuel system constraint which includes maintaining similar tanks (e.g., 1 and 4, or 2 and 3) within 700 lbs. of each other and maintaining adjacent tanks (e.g., 1 and 2, or 2 and 3) to within 1000 lbs. of each other. The view corresponding to selection of this switch is similar to the wing tank graphics, with the main difference being the range of the scale of the tanks, which is limited to 10,000 lbs. FEs can use the familiar and powerful “fuel balances” to deduce whether one of the tanks is losing fuel too rapidly, before it becomes an issue.

With the information provided in the SYSTEMS DISPLAY, FEs should be able to confirm or refute the hypotheses they generated by observing the ENGINE SYSTEM OVERVIEW and FUEL SYSTEM OVERVIEW displays.

6.5.3 Tank Balance Process View

The Tank Balance indications (Figure 39) demonstrate that task constraints can be effectively incorporated into an EID display. The task constraint chosen was the balancing of fuel in each tank, as previously mentioned. This view is obtained whenever the FE selects the ‘balance’ switch located in the FUEL SYSTEMS OVERVIEW viewport.
(Section 6.4). The resultant process view, shown in Figure 41, provides all the
information required by an FE to determine whether system task constraints have been
violated. The switch in the FUEL SYSTEM OVERVIEW also acts as an alarm, always
indicating the status of the most deviant parameter. Thus, it is possible that the wings may
be perfectly balanced, while some of the tanks are out of balance. In this situation, the
switch's shape would appear to be balanced, but its red colour would clearly signify that
the task constraint has been violated somewhere within the system.

In summary, the primary purposes of the SYSTEMS DISPLAY are: a) to support
the localization of faults that have been detected (and perhaps diagnosed at a coarse level)
using the three overview displays; and b) to support planning and compensatory activities.

6.5.4 Component Descriptions

The following describes each group of components, as well as the measurement
and input devices utilized in the SYSTEMS DISPLAY portion of the interface (Viewport
4). The measurement indicators are described first, followed by the components which
FEs can manipulate. A concerted effort was put forth to follow the guidelines discussed in
**Tank:** Tanks are represented by a large rectangular container with a blue bar on right side that increases in height, giving the appearance of water filling a tank. Adjacent to the top of the bar is a green triangle, as well as a digital readout of the level. The green triangle indicates the amount of fuel that the tank should contain, and the height of the bar indicates the actual tank level. If the two are not aligned then there is a problem which is indicated by the triangle changing colour from green to yellow or red, depending upon the severity of the fault. A hollow rectangle is also placed behind the bar for the purpose of determining rate of change of the data as well as high and low measurement points. Within the tank are the various components physically located inside the tank, such as pumps and filler valves, each of which can be manipulated separately.

**Flow Meters:** There are two types of flow meters used in the interface: Fuel Flow Meters (green bar indicating flow, shown to the right) and Oil Flow Meters (brown bar indicating flow). The flow meters have the familiar shape of a physical flow meter, with a coloured bar that increases in height as the flow increases. Both a green triangle and a digital value of the flow travel alongside the bar indicating flow level. A hollow rectangle behind
the flow bar also serves to indicate the relative displacement from minimum and maximum measurement points.

If the flow exceeds the maximum or the minimum of the meter, a new bar will appear above the meter to alert the FE. Without this information, FEs would not be able to discriminate between maximum or minimum scale readings, and off scale readings (cf. Mumaw et al., 1991).

**Thermometers:** Thermometers used in this interface resemble ordinary mercury thermometers, with a red ‘fluid’ rising and falling inside a long tube with a bulb at the base (shown to the right). Thermometers behave in the exact same manner as the flow meters above, with the exception of the ‘off scale low’ feature. When the thermometer reading falls below its minimal measurement level, the liquid within the bulb disappears.

**Pressure Meters:** The oil pressure meters are blue in colour, and operate in the same manner as do the flow meters.

**Torque Meter:** The torque meter also operates in a similar manner to the flow meters. However, there are three major distinctions. The first is the purple colour for the bar; the second is the non-linear scale of the torque meter, which highlights the zone of interest (between 19200 and 19800 lbf.); and the third is the addition of the red alarm level indication. This functionality could have been placed on all meters, but it was used only on this gauge as an example of how alarms might be incorporated into the
interface. In an operational interface, alarm level indications would be placed on all meters.

**Piping:** Piping used in this interface provides two pieces of functional information: the presence of flow in a segment of piping; and the purpose of the piping (whether it feeds an engine, refuels, dumps fuel, or feeds a manifold).

Piping which has no flow carries only topological information, and this is reflected in the dull colour used in its representation. Piping which carries a flow is both thicker and brighter in colour than its dry counterpart. The classifications of flows are as follows:

- **Direct flow to an engine** - Solid, thin, and dark blue with no flow.
  - Solid, thick, and bright green with flow.

- **Engine flow via a manifold** - Dashed, thin, and dark blue with no flow.
  - Dashed, thick, and bright green with flow.

- **Direct dump flow** - Solid, thin, and dark orange with no flow.
  - Solid, thick, and bright orange with flow.

- **Dump or refuel flow via a manifold** - Dashed, thin, and dark orange with no flow.
  - Dashed, thick, and bright orange with flow.

The remainder of this subsection describes the controllable components of the interface.
**Valve Settings.** Valves are indicated by iconic versions of the current valve representations in a Hercules. The state of each valve is indicated by its orientation. If the valve is collinear with its surrounding piping, flow is supposed to flow at its maximum rate. If collinearity is broken, flow should be off. The operator selects the state of the valve by 'clicking' on top of the valve graphic. If a valve is faulty, then a red 'X' is superimposed on the valve graphic, and all user interaction on that valve has no effect on the rest of the system.

**Pumps.** For the pumps, a standard icon currently used in Hercules cockpits was utilized (a circle with a '+' inside it). When the valve is off, the representation is hollow, and when on, the pump is filled in the same colour of green used to indicate fuel flow in pipes. Clicking on the pump again will successively turn it on and off. Pump alarm coding is the same as that used for valves.

**Tank Filler Valves.** Tank filler valves are also standard Hercules iconic representations and operate in the same manner as pumps (shown at the bottom right of this page).

**Pan Control.** In both the Fuel and Engine zoom displays, an added feature of panning has been incorporated to allow the FE to gradually move through the zoomed display. The pan controls located on the perimeter of the SYSTEMS DISPLAY are in the form of sliders (See Figure 40) which indicate the
direction in which they can be manipulated. The sliders are ‘smart’ in that they only permit the FE to move within the perimeter of the zoomed display.

6.5.4.1 Requesting Service Information

One additional example of how the FE can interact with the interface involves obtaining information regarding the service records of a specific component. This functionality has only been attached to the engine’s fuel enrichment valve representation, in order to minimize the size of the executable program. When the user clicks on the fuel enrichment valve graphic, a menu appears which contains four options: Report a Snag; Review Snag History; Service Record (Review component servicing); and Quit. The first three options immediately provide the FE with the history of a component’s operation, which can potentially provide insight into the source of a fault. For example, if a component has had a history of intermittent operation, the FE could be made aware of this prior to flight, and be prepared if a snag involving that component occurs.

6.6. SYSTEM INFORMATION

The SYSTEM INFORMATION display is an extension of the SYSTEMS DISPLAY. Information provided within viewport 4b of Figure 33 is directly dependent on the context of the SYSTEMS DISPLAY. There are three types of views that are associated with the SYSTEM INFORMATION Viewport: trend data presentations; engine performance data displays; and a default monitoring screen.
6.6.1 Trend Information

Trend data are presented in the SYSTEMS INFORMATION display unless state diagrams are requested. The incorporation of trend data facilitates diagnosis of slowly evolving events, such as slow fuel leaks from a tank. Any deviations can be effectively monitored over time, instead of only instantaneously, as in the existing Hercules C-130 systems (see Chapter 3).

6.6.2 Engine State Diagrams

By default, the SYSTEMS INFORMATION viewport is set to display state diagrams for each engine (cf. Figure 42). State diagrams are a standard tool used in engineering design to describe the performance characteristics of a thermal-hydraulic system. For a detailed discussion and description of state diagrams related to aircraft, see Oates (1988) and Wood (1991). The vertical axis represents Energy, and the horizontal, Entropy. Although energy is generally described in terms of temperature, units of hp were used to maintain the FE’s familiarity with power production calculations as much as possible. The numbers surrounding the state diagrams represent the different engine stages, and have been extracted from standard engineering engine state numbering methods. Figure 43 shows state diagrams of both an ideal and an actual Allison engine. Zero (0) represents air that has been imported from the environment; 1, the inlet to the compressor; 2, the combustion inlet; stage 2-3 represents the heat acquired through the burning of fuel; 3-4, the work being done by the propeller; 4-5, the work of the turbine; 5-6, the work provided by the exhaust gasses in the form of thrust. Stage 6-0 is an
imaginary connection in aircraft engine cycles, representing the ambient air (modeled as a constant pressure heat sink), recycled from the exhaust to the engine inlet. In a perfectly ideal system, stages 0-2 would be isentropic (constant entropy), represented by vertical lines in the state diagram. Stage 2-3 would follow an isobar (equal pressure), indicating ideal heat addition. Stages 3-6 would also be isentropic, during expansion, and stage 6-0 would follow an isobar indicating constant pressure heat rejection. An example of a state diagram for a perfect engine cycle is provided in Figure 43a.

Since an ideal engine is based on an unobtainable cycle, the state diagrams provided in the interface are based upon a normative Allison T56-A15 engine, (represented in Figure 43b and 43c) by a dotted white open polygon. The solid purple area represents how the actual engine is operating compared to 'normal' operation. Both engine representations are affected by thrust and engine operating parameters. With this
powerful tool, a FE should immediately be able to notice how the engine is straying from its normal, expected performance, and more importantly, be able to identify the stage, or stages responsible for the reduction in performance. Figures 43b and 43c illustrate an engine operating as expected, and in a low fuel flow situation, respectively.

![Image of state diagrams](image)

*Figure 43: State Diagrams of a) a Perfect Engine, b) Normally operating Allison T56-A15 engine, c) Low fuel flow scenario*

The state diagrams have been provided with an additional functionality which permits the FE to choose any segment of the diagram and immediately zoom in on the appropriate engine components which are related to that segment.

### 6.6.3 Monitoring an Engine’s Performance

If a FE requires additional engine performance information beyond that presented in the ENGINE SYSTEMS OVERVIEW display, he would select the center of the engine’s polar star (Figure 35). Upon doing so, the state diagrams of the other three engines would be replaced by the selected engine’s specific performance information, as in Figure 44. The state diagram of the chosen engine would not move. Immediately to the right of the state diagram, is information relating to engine efficiency. The last three graphics on the right are trend plots of Specific thrust (S), Specific Fuel Consumption
(SFC), and Power Specific Fuel Consumption (PSFC) that FEs are accustomed to computing manually. These additional performance indicators will potentially aid the FE in diagnosing the state of an engine.

Figure 44: Engine Performance information for Engine #1

6.7. EXAMPLES OF USAGE

Three hypothetical faults will be used to demonstrate how the information provided in the various viewports can be used together, to guide the FE to an accurate diagnosis of a fault. The first scenario involves low fuel flow to an engine; the second, a propeller that has de-coupled from the engine; and the third, imbalanced fuel tanks.

6.7.1 Low Fuel Flow to Engine #2

The first example consists of a lower than expected rate of fuel being received by the nozzles of engine #2. A partial screen shot of this scenario is shown in Figure 45. In the software prototype, this fault is initiated by pressing the 'F2' key on the keyboard. The abnormal shape of the state diagram and the deviation of Engine #2's polar star are immediately evident. Both shapes are indicative of a low fuel flow rate to the engine
nozzles (polar star: low fuel flow, lower power output, low torque, and a low engine oil pressure; state diagram: shape of diagram, particularly between states 4-5) and when combined, immediately guide the FE towards searching for the root of the problem. The engine icon in the FUEL SYSTEM OVERVIEW display clearly indicates that the flow sent to the engine from the fuel system is sufficient (triangle equal to the bar height). Thus, these three pieces of information allow the FE to deduce that there is a low fuel flow problem, and to also narrow his search domain to within the engine system itself. The FE would then zoom in on the appropriate engine components (e.g., engine nozzles) to pinpoint the source of the error, and decide on his compensatory actions to rectify the fault.

This chain of events is quite different from that currently being followed by FEs. Active problem solving replaces regimented procedure-following. In a similar scenario within a current Hercules aircraft, an FE stated that he would either shut the engine off, or live with the reduced power production, without delving any further into the fault, unless another engine was lost. FEs currently follow this procedure because the information provided to them is insufficient to efficiently search and rectify such a fault. In the software prototype, the interface can be returned to its ‘normal’ operating state by pressing the ‘F1’ key.

6.7.2 Propeller #2 De-couples From the Engine

In the second scenario, the propeller disconnects from the engine drive shaft. In the software prototype, this fault is initiated by pressing ‘F3’ on the keyboard. A screen
shot of this event is shown in Figure 46. For this situation, the only relevant viewports required to diagnose the fault are the ENGINE SYSTEM OVERVIEW and the SYSTEMS INFORMATION displays. As in the previous example, the abnormal shapes of the polar star and state diagram for engine #2 are immediately evident. Indications of propeller decoupling are: Extremely low TIT and FF; Near 0 torque; and hydraulic pressure, generator output, reduction gear section, and oil pressure all near normal. With experience and knowledge of how the state diagram and polar star deviate, the FE should be able to attribute the fault to the propeller disconnecting from the drive shaft. Having arrived at a diagnosis, the FE would proceed as in the former scenario by zooming in on the propeller indications to ensure that his initial diagnosis was correct, and then form a compensatory plan. In the software prototype, the interface can be returned to its ‘normal’ state of operation by pressing ‘F1’

6.7.3 Tanks Become Progressively Imbalanced

The final scenario provided addresses the fuel imbalance task constraint issue mentioned in subsection 6.5.3. Figures 47 a-d indicate the progression of this fault. Although this scenario is unlikely, it demonstrates how the interface would support FEs in dealing with a fuel imbalance incident. In Figure 47a, all tanks are balanced and all task constraints are satisfied. In the software prototype, pressing ‘F6’ on the keyboard simulates the condition in which the FE has fed all engines with tank one only (Figure 47b). Here, tanks one and two are shown to be out of balance, as are tanks one and four. As the fuel in tank one drains, the interface alerts the FE to the fuel imbalance between tanks one and two by the colour change of the tank balance graphic in the FUEL
SYSTEMS OVERVIEW display. If an inexperienced FE does not think ahead, or is too busy, and simply compensates by turning on tank two’s booster pumps, the situation displayed in Figure 47c develops. In the software prototype, this can be simulated by pressing ‘F7’ on the keyboard. In this situation, tanks one and two are again in balance, as intended by the FE. However, the task constraints imposed between tanks one and four have been violated. The wing side constraint is also about to be broken, as it is at its limit, as indicated by the rectangular shape of the balance indicator between the wing side graphics (see Figure 39). If the FE does nothing to remedy the situation, then the situation described by Figure 47d develops (In the software prototype, pressing ‘F8’ key on the keyboard). The relative wing side balance constraint has been broken, as have constraints 1&4 and 2&3 (space between the triangles and red colour, as in Figure 39). The balance indicator between tanks three and four is also balanced, as both tanks have yet to be manipulated. To return to ‘normal’ operation in the software prototype, the user must press ‘F5’ on the keyboard.

This example is powerful in that it clearly teaches the FE to proactively compensate for a fault, and to not perform ‘quick fixes’ (i.e., turning on booster pump 2, as in Figure 47b). These types of fault scenarios have training implications for new FEs, and demonstrate how an EID interface can accommodate both anticipated and non-anticipated faults.
Figure 47c: Mistakenly Compensating With Tank 2 Booster Pumps Only
Figure 47d: Situation is Left to Worsen - Fuel System Completely Out of Balance
6.8. ADDITIONAL DESIGN PRINCIPLES

While the principles of EID played an important role in the design of the interface in Figure 33, it is clear that there were certain design issues that were not intended to be addressed by EID (Vicente & Rasmussen, 1992). For such cases where guidance is not provided by EID, other design principles had to be consulted. The two primary issues that were relevant for this prototype were the issues of visual momentum and perceptual organization.

6.8.1 Visual Momentum

Most of the previous work on EID (Subsection 2.4.2) used an interface that integrated all levels of abstraction into a single, coherent visual form. This was possible, in part, because the system was relatively small in scale. In larger scale systems, such as the Hercules or a power plant, it is not possible to adopt the same approach. Supporting visual momentum (Woods, 1984), allowing operators to effectively extract information as they navigate between and within process views, becomes very important. EID was not meant to directly address this issue, so in designing the EID interface for the Hercules Engineering Systems, the principle of visual momentum was utilized to design display features which are directly intended to support navigation. Examples of such features include the context sensitive spokes in the ENGINE OVERVIEW, the tanks and piping in the FUEL OVERVIEW, the state diagrams in the SYSTEMS INFORMATION displays, as well as the pan functions in the SYSTEMS DISPLAY. Thus, the principle of visual
momentum complemented the EID framework in the design of a more effective interface than could be achieved by using EID alone.

6.8.2 Perceptual Organization

EID also does not address specific issues pertaining to the design of visual form (i.e., what colours to use, the shape of the graphical object). Rather, EID simply states that the constraints of the work domain should be mapped onto perceptual signs in the interface; it does not precisely state how this is to be accomplished (cf. Vicente & Rasmussen, 1992). Therefore, EID must be supplemented by more detailed design principles that can be used to design visual forms that make it easy for operators to effectively extract information from the interface. Design principles developed at Westinghouse, based on existing knowledge of human perceptual organization, were used to accomplish this goal (Mumaw, Woods, & Eastman, 1992; Woods, 1991). These principles can be summarized as follows:

An effective display can be perceptually segmented into several layers where some are more in the background and others are more in the foreground. Background layers are less salient than foreground layers. Therefore, the former should be used to represent less important information (e.g., pipes), whereas the latter should be used to represent the most important information (e.g., alarms, goal areas). This is accomplished by designing visual forms that allow operators to effectively discriminate between these different layers.

Examples of the application of this principle include: the salient AES display flight distance indicators; the much less salient coding used for pipes; and, the distinctive colours
used for the 'balance' indicators in the FUEL SYSTEM OVERVIEW display. Thus, detailed perceptual organization principles complemented the EID framework by further constraining how to achieve the high level objectives specified by EID.

6.9. CONCLUSIONS

The information and guidance provided by EID, supplemented by other design principles, were used in the design of the interface described above. It was intended that the information be mapped in a one to one basis between the domain constraints, and the perceptual signs provided by the interface. To summarize, three principles of EID which were used, are restated along with examples of how they were implemented in the interface:

1. Skill-based behaviour (SBB) - to support interaction via time-space signals, the operator should be able to act directly on the display. The structure of the displayed information should be isomorphic to the part-whole structure of movements. Examples include pan and zoom features, selectable tanks, and valves.

2. Rule-based behaviour (RBB) - provide a consistent one-to-one mapping between the work domain constraints and the cues or signs provided by the interface. The tank balance graphic in the FUEL SYSTEMS OVERVIEW display, and the circular distance indications in the AES PERFORMANCE display.

3. Knowledge-based behaviour (KBB) - represent the work domain in the form of an AH to serve as an externalized mental model that will support knowledge-based problem solving. The AH was used to create the interface, and all cells found to be
relevant for this application have been mapped onto the various viewports (see Figures 31, and 32).

There are numerous benefits that can be gleaned from this approach:

- Operators should be able to determine the state of the system at a glance, and rapidly and effectively extract information from the display. This factor also has implications for training, as the interface should help inexperienced FEs learn how different systems are related to each other.

- The context sensitive approach to the engineering systems, combined with the AH’s accurate system model should reduce the requirement for memorization of both physical and dynamic relationships. At the same time, it should provide a better conceptual understanding of the codependencies within and between systems, as well as the dynamics involved.

- Finally, the design is intended to support planning and compensatory activities by allowing operators to directly act on the display surface.

In order to obtain these benefits, a cost in terms of the level of effort required to produce the prototype must be accounted for. The bulk of the work for this particular thesis was conducted by one person who had minimal initial knowledge of the system or its design basis, but who was very familiar with thermodynamics, thermal-hydraulics, EID, and who had used VAPS\textsuperscript{*} in one other interface design project and had taken a one-week VAPS\textsuperscript{*} course. With this in mind, the levels of effort associated with the various phases of the thesis were:
Analysis of System: 4 person-months
Design of Interface: 1 person-month
Programming: 2 person-months
Total: 7 person-months

This level of effort is very sensitive to the amount of previous knowledge that the designer possesses. For example, someone who was already intimately familiar with the systems could perform the analysis in a considerably shorter time.

The interface described above is only a prototype. It was designed with little input from operators, and it is important that the interface be evaluated rigorously before it can be considered a candidate for any real aircraft. There are several features which could perhaps be improved or incorporated into the design specified in this thesis.

- the control of the pan function could perhaps be improved by including a multi-degree of freedom input device that would allow simultaneous movement in two or even three dimensions.
- the zoom feature itself could be potentially improved upon by incorporating Asea Brown Boveri’s (ABB’s) ideas of additive and alternative information zoom (Elzer, Siebert, & Zinser, 1989).
- more information could be acquired to provide a more realistic and accurate model of engine operation and behaviour.
- all special functions available only on certain components (such as maintenance menu) could be incorporated into components.
CHAPTER 7: INTERFACE EVALUATION

7.1. PRELIMINARY EVALUATION BY FEs

The prototype interface was presented to potential expert users (i.e., FEs) at CFB Trenton (the same base where the simulations were held). Seven FEs, of various ranks (Sergeant (SGT) to Master Warrant Officer (MWO)), participated in the informal evaluation, which was guided by the format proposed in Meister (1985). The FEs (who were labeled FE1 - FE7 in a counterclockwise order based on their seating location) also possessed varying amounts of computer familiarity, which might have also been a factor in their comments. All comments were recorded on cassette tape, and FEs were encouraged to interrupt the presentation to express their opinions. At the end of the demonstration, FEs were asked to comment on the prototype as a whole. A summary of their reactions both during, and after the demonstration, is presented below. Text in square parentheses [ ] represents editorial comments.

7.1.1 Comments in Favor of the Prototype

The prototype was very well received, beyond our expectations. All seven FEs believed the interface to be a "step in the right direction" and expressed a desire to see it implemented as a training tool in the near future. FE3 remarked: "If you are going to incorporate this into the airplane, and make a training aid, I think that it is exactly perfect, because the training aid would do exactly what the airplane does. You couldn't ask for anything better." In general, comments received about the interface were positive, indicating that the systems had been properly modeled, and the AH properly incorporated.
For example, FE6 said: "It is an excellent concept, and it definitely makes things easier to see and de-snag, and analyze"

FE2 agreed: "It is easier to teach people on a system like that..."

FE3 continued: "That section there of the fuel system schematic [Interface Default View, Figure 33] we have on compact disk, where you can manipulate valves etc...but the difference is that you lose that when you get back on the aircraft...You are packing a lot more information in the same space, and doing it so that it is easy to find stuff"

FE6: "In your concept, you have everything tied together so that you can see in this case just the fuel and engines working together, but it is definitely easy."

FE4 continued on to state that "The Fuel System Overview dump meters seemed very useful, and that [he] could have used something like that on a flight [that he was on] last week."

Comments were also received about specific components in the interface, but the FEs were most emphatic about the polar star and state diagram graphics, which they found to be very simple to use. In the prototype version demonstrated to the FEs, the example of a low fuel flow fault situation (Section 6.7.1) was incorrectly presented in the polar star display [Compilation of the prototype caused the polar star polygon to shift to the left]. Within a few seconds of introducing the fault, the FE closest to the display [FE7] was able to correctly identify the fault as being a low power fault, and not a low fuel flow fault, according to the polar star indications. This is a clear indication that the FEs are able to effectively use the graphic. FEs also particularly liked the fact that similar faults caused the polar star to change in a consistent manner. FE3 claimed that "it is giving you a
quicker mental picture of what your problem is, rather than trying to take something like an individual reading off each gauge and then forming it...this is doing it for you.” He continued on to remark that: “We can look at a half a dozen gauges on there [the current cockpit] and say, oh yeah! right away that that is a prop [propeller] de-coupled, but it is a lot easier to see it on this than it would be on those six separate gauges.”

FE7 continued: “Like you said, if you got used to that way [polar star and state diagram together] of presenting it would be easy...”

FE3 replied: “yeah, you’d be able to analyze that like a hundred times faster.”

All FEs agreed that the polar star would be a very useful device in the cockpit, especially during take-off, when time is critical, and they can’t afford the time that it takes with conventional gauges to make a decision. FE3 stated: “You can key into a shape like that, and you’re guaranteed of a quick response when you’re looking. I mean, if you know that a certain shape is going to indicate a certain problem, at a glance, you can see that you are looking at a particular problem, and you go from there. You also have colour coding do you not? Just for an example, if you are running down a runway and you had anything go wrong, it is instantaneous, there is no decision on whether you are going to abort or not, it is right in front of you...it is basically hitting you over the head...”

FE7: “yeah, it is not within the limits that you specified...”

FE3: “Before, you had to say, OK, that gauge is not where it is supposed to be, so then you have got to look at the rest of them...and how much runway have you used up now?”
Aside from the polar star display, there were also other features which the FEs thought that they would find very useful. The following examples serve to reinforce that the additional design principles such as perceptual organization, which provide guidance on visual form can also be effectively incorporated into an EID interface. All FEs liked that they could mark a valve as being unserviceable with a red “X” that was identical to the piece of tape that they currently use. They were also all excited about being able to request service records about all components through the “component service menu”. Visual momentum seems to have been maintained as well, as all FEs found navigation within the interface via selectable icons in the overview displays to be very logical and simple to use.

The above comments indicate that the interface had many good features, but, as in all prototypes, there is always room for improvement. The following subsection discusses some of the concerns voiced by the FEs.

### 7.1.2 Concerns about Interface Features

Concerns voiced during the presentation of the prototype involved minor adjustments to the interface, such as the following comment by FE7, who thought that “there should have been digital indications of the amount of fuel in the Fuel System Overview.” Most such comments had simple solutions which were answered in a later portion of the demonstration. In the above concern, FE7 accepted that he could find more detailed information about the tank in the Systems Display by selecting one of the tanks in the Fuel Systems Overview display. Another similar concern was that of FE7, involving tank instruments; specifically, “what happens to the overview displays when a tank meter
goes hard-over full or hard over empty.” FE6 quickly provided the correct response:

“Yes, presentations showing balances would be affected by the faulty meter... but, the computer would have been measuring [keeping track of the appropriate meters] the whole time [for the AES display], so it should be able to base the amount in the tank on before the meter went faulty and then go on from there.”

Some concerns that were not shared by all FEs do not have readily available responses, and require proper experimental testing before a solution can be found. These types of concerns also sparked debates amongst the FEs. For example, FE1 thought that only the spokes and the individual segments of the polar star that exceeded the specified parameters should change colour, and not the entire polar star indication.

FE4 disagreed “from where I sit, I can barely see the gauges at night, so I would rather have the entire shape change colour.”

FE3 continued on to say that “if for some reason, only one parameter changed enough so that it was at zero, then you might not be able to see the problem if the whole shape did not change colour.”

Another similar debate was sparked by FE7: “There is some information there that I think is probably more than what we need, such as the load efficiencies and the engine performance indicators [in the Systems Information Display]. Maybe somewhere along the line, you might be interested in them, but I don’t imagine that anyone would use them in flight...that is an engineering thing [emphasis added]. That has gone way deeper than anything that you would need in flight...Like that map in the corner [AES], it is an interesting thing, but it is something that I can’t imagine that you would use in flight. You
have got a million other pieces of equipment on the airplane that are going to be doing the same thing."

FE3 disagreed: "Yeah, you have got a lot of different instruments, but you can use this [AES] as a quick reference. All those instruments tie into this display to tell you right away. I can see where we are supposed to go on the map, and according to this, we are not going to make it... just within a millisecond." Again, only testing could provide an adequate solution for such problems.

Finally, concerns were also voiced that involved the layout of the interface, which were valid comments, but would not become a factor until the entire interface was completed, and its location in the cockpit fixed. FE1 and FE7 did not like the placement of the AES overview, and thought that it did not belong in the interface. FE7 continued: "I can probably use that space more effectively instead of having the map there." The demonstrator suggested that it could go somewhere else in the cockpit, and FE3 interrupted to remark that "even if you put that somewhere else, I think it is a great thing, but to have that on the instrument panel right in the center... you could probably gain more by having the information somewhere else. You are getting all that information, so you might as well use it." All FEs then agreed to this.

The concerns raised by the FEs were mostly introduced when confronted with having to offer some negative comments about the interface. Most of their concerns during the demonstration had simple responses, which were offered. However, there were valid issues which could only be properly answered via controlled experimentation, in an actual simulator setting.
7.2. CONCLUSIONS

Some of the literature reviewed (e.g., Veitengruber and Rankin, 1995; Lenorovitz, 1992) stated that the current practice in industry is to use evolutionary rather than revolutionary designs. The rationale is that evolutionary designs are easier to understand, and require less training. The overall response to the interface was therefore much more positive than could have been anticipated, because of the relatively large number of unfamiliar (i.e., revolutionary) features in the interface. Understandability did not seem to be an issue at all, as exemplified by the FE who was able to correctly diagnose a misrepresented fault example. Thus, it seems that the AH, when properly incorporated into the interface, can help designers implement some revolutionary, rather than evolutionary, features that can be well-accepted by users. This statement is reinforced FE7's final comment on the interface as a whole (which all FEs present agreed with): "this [the interface] looks like it would be extremely easy to do trouble shooting...As a project where you are presenting it and you are saying, hey will this work, I'd say yeah, it would work."

Finally, although the initial impressions seem very positive, the entire interface (including unmodeled hydraulic and electronic systems viewports) requires in-depth empirical testing before any definitive conclusions can be drawn.
CHAPTER 8: IMPACT OF THE OPERATING PHILOSOPHY ON THE DESIGN OF INTERFACES

8.1. Motivation

In evaluating whether or not EID can be applied to aviation, the creation of an interface is necessary, but not sufficient. The operating philosophies inherent in EID and those currently existing in the aviation industry must also be compared to determine if benefits can be obtained by using the EID approach. The most fundamental issue that needs to be addressed is the intended role of the human in the system. The resolution of this issue will have a strong impact on how displays should be designed, since different display designs are required to support different types of cognitive activities. This discussion will begin with the operating philosophy that currently seems to be prevalent in many commercial airlines.

8.1.1 Current Airline Practices

The current attitude in the aviation industry seems to be that the pilot's role is not so much to understand in a deep sense what is going on in the system, but rather to diligently execute the steps in a given procedure to satisfy mission goals. This is somewhat of an oversimplification, as the two activities are not mutually exclusive and can be combined to varying degrees. However, there are several pieces of evidence that indicate that this is indeed the current trend in the industry. First, in our discussions with domain experts, the emphasis that is placed in operations to “execute the bold face” became clear. The argument for this seems to be that if something goes wrong, pilots will not have the
combined time and capacity required to diagnose the root cause of the event and then
develop an appropriate compensatory plan. Instead, pilots are supposed to rely on the
designers' analyses and forethought that have been incorporated into the procedures made
available to them. As a result, the focus is more on "doing" rather than on "understanding".

Second, the training programs that pilots undergo are putting less emphasis on
knowledge of how the engineering systems function. Whereas in the past, pilots were
exposed to a great deal of theoretical training and instruction about the functional structure
of the aircraft's systems, today these aspects receive much less attention. For example,
Scott (1995a) quotes one source as saying, “Maybe pilots don’t really need to know the
details of how a system operates anymore” (p. 14). He then adds that this opinion “is
consistent with current airline pilot training philosophies, which tend to de-emphasize in-
depth knowledge of systems” (p. 14; see also Scott, 1995b). Scott (1995b) also quotes
United Airlines' manager of fleet operations for the Airbus A-320 and Boeing 737 as
saying, “we’ve moved away from teaching as much about how the aircraft works” (p. 17).
There seem to be several reasons for this:

1. Aircraft systems are becoming more complex, so the task of learning how they function
   in detail is more difficult than in the past. Scott (1995a) quotes an engineer as saying,
   “Humans only have a finite capacity for absorbing all the details of a [new] system’s
   operation. There’s just too much flexibility and complexity. It’s not only very difficult
to absorb and retain [that information], it’s practically impossible” (p. 14).
2. Engineering systems are being built to be more reliable than in the past. Thus, faults should only occur very infrequently. Moreover, when they do occur, automated systems will frequently be able to detect and diagnose the problem, and inform the pilot as to what actions need to be taken. In the case of the MD-11, the automation even compensates for failures in engineering subsystems by reconfiguring those systems. As a result, the argument is that there is less of a need for understanding on the part of the pilot. For example, one pilot stated that “Today, [airlines] teach new guys only what they need to know to do the job, make them proficient at that, and let the [automated aircraft] take care of the rest of it. Nowadays, they [airlines] don’t want the pilot messing with the system” (Scott, 1995b, p. 18).

3. Airlines want to train their pilots as economically as possible, and the result, according to one pilot, is that they emphasize “button-pushing rather than knowledge of systems” (Scott, 1995a, p. 14).

4. In many cases, a detailed understanding of what is occurring within the aircraft is not required to compensate effectively. Unlike process control systems that have many backup and safety systems that the operators can choose from in fault compensation, aircraft have less structural redundancy. As a result, the attitude is “[The pilot] can’t do anything about some of this stuff, anyway, so why bother explaining it to him?” (Scott, 1995b, p. 18).

5. Finally, there seems to be a general attitude that rare events that fall outside the scope of procedures and the capability of automated systems simply will not occur in practice. For example, “United’s pilots are taught to handle both routine and unusual situations
they will most likely encounter in typical line service, such as adverse weather and wind shear” (Scott, 1995b, pp. 17-18, emphasis added).

The fact that the primary role of the human in the system is one of an executor of procedures, rather than an adaptive problem solving agent, was reinforced by visits to Air Canada simulators. Several observations confirm the trends outlined above. For example, pilots echoed the opinion that problem solving is not required because “when something goes wrong and you don’t know what to do, simply turn it [i.e., the faulty component] off and let the redundant systems kick in.” Similarly, some pilots thought that displays do not need to provide detailed information to support detailed fault diagnosis because they cannot fix the faulty component most of the time anyway. As a result, the pilots tend to focus on compensation activities. Another relevant point is that, during check rides, pilots are told ahead of time which class of failures they will be experiencing in the simulator. As a result, this type of training is primarily geared towards helping pilots cope with familiar expected faults, rather than unfamiliar, unanticipated faults. This is consistent with the pilots’ perceptions of the probability of a rare event occurring. When asked about this possibility, pilots replied that if the event is not in the procedures, then it will not happen. Therefore, our observations at Air Canada are very much in line with the operating philosophies reviewed above.

Interestingly, however, the operating philosophy we observed at Trenton AFB was quite different from what seems to be the norm in commercial airliners. In the military transport setting, we found that FEs are frequently given unfamiliar and very low probability
events during simulator training. A great deal of attention is also devoted to ensuring that
the FEs have a detailed and comprehensive understanding of every component in the
engineering systems. Consequently, in simulator training, FEs are more frequently engaged
in problem solving activities pertinent to fault diagnosis. There are several potential reasons
for this stark difference in philosophy:

1. Military flights can take place in weather conditions that are more severe than what
   would be acceptable for a commercial airliner, and so failures are more likely to occur.
2. In military aircraft, the FE has a greater capability for examining system components and
   trying to repair them in flight.
3. The military aircraft we observed are older and thus the components are likely to be less
   reliable, causing failures to occur more frequently.
4. The military aircraft we observed do not have as much automation as contemporary
   commercial aircraft, so there is a greater burden on FEs to detect, diagnose, and
   compensate for abnormal situations.

Thus, there seem to be strong differences in operating philosophies between commercial

8.1.2 EID Operating Philosophy

The operating philosophy behind EID is vastly different from that found in commercial
aviation. EID was originally developed in the context of process control systems where the
greatest threats to system safety are events that are both unfamiliar to operators and
unanticipated by designers. In these novel situations, operators are required to play the role
of an adaptive problem solver if they are to be able to effectively cope with the rare event. Procedures are not applicable, by definition, because this class of events has not been anticipated by designers. As a result, the operator role that EID tries to support is one of an adaptive, reflective agent. This is accomplished by providing operators with an interface that contains detailed information about the status of system components, functions, and goals, as well as the relations or linkages between all of these. As a result, operators are not told what sequence of actions to take, but instead are given the computer support that is required to develop a safe path of action that is appropriate for the unique contingencies of that event. Clearly, this is very different from designing displays to support procedural activity.

Although these issues have not been researched in detail, it seems that these differences lead to the following implications:

1. An EID interface would have much more information than a traditional cockpit display.
2. Operators (i.e., pilots) would have to have a detailed deep understanding of the system if they are to be able to effectively interpret and use the information provided by an EID display.
3. Operators should be selected according to their capability to be adaptive problem solvers, rather than compliant procedure followers.
4. Procedures would play less of a role during abnormalities.
5. The entire operating philosophy of the airline would have to change.

Clearly, this is tantamount to a revolutionary change in systems design. The question that remains is whether the changes suggested by EID are in fact appropriate for aviation. It is
certainly possible that the EID philosophy may make sense for nuclear power plants, but may simply not be appropriate for aviation.

8.1.3 Reasons Why EID Is Relevant to Aviation

In this subsection, we outline the reasons why we think that the operating philosophy behind EID is in fact worthwhile considering for aviation contexts.

Increase in absolute number of unanticipated events. The amount of air traffic is increasing and will continue to do so in the future. For example, in China alone, the amount of air traffic is expected to double within the next 8 years (Boone, 1995, p. 45). Even if the rate of unanticipated events per aircraft stays the same (note that this is a conservative assumption since unanticipated events are not necessarily linearly related to the amount of air traffic control), then the absolute number of such events per year will increase. It seems that public perception is driven more by the absolute number of accidents rather than relative amount. Therefore, it is likely that more attention will be drawn to these types of events over time. Whereas in the past events like Sioux City (NTSB, 1990) were perceived to be a very rare freak occurrence, it is likely that this will change with the expected increases in air traffic.

Furthermore, as the current new generation of automated planes gets older, it is also possible that failures are more likely to occur. Again, this may draw more attention to "rare" events, thereby increasing the priority of supporting pilots in such challenging situations.
Procedures do not capture all events. Despite the fact that the commercial aviation industry is highly focused on procedural operation, it is clear that procedures simply do not capture all events. There are several sources of evidence that can be used to substantiate this claim. First, there are several authors in the industry who have acknowledged this fact, some who even work on improving the design of procedures (Degani & Wiener, 1994). Second, if one examines the current list of emergency operating procedures available in many aircraft, it is clear that certain events for which there is now a procedure (e.g., flying through volcanic ash) were simply not on the list originally. Third, there are several well-known examples of accidents that were caused by events that were unanticipated and for which there were no procedures (e.g., NTSB, 1990). Thus, there will always be events for which procedures do not exist, and as pointed out above, these can be expected to occur in greater numbers in the future.

More importantly, the fact that there will not always be appropriate procedures has important implications for human performance, as indicated in a study conducted by Rogers, Schutte, and Latorella (in press). In this experiment, pilots had to deal with faults for which there were no checklists (i.e., procedures). “Many of the pilots spent considerable time looking through checklists in an attempt to find one that seemed appropriate for the condition, and in some cases, pilots used an inappropriate checklist to compensate for a fault” (Rogers et al., in press).

Increase in system complexity. The new generation of airplanes (e.g., High Speed Civil Transport (HSCT)) will be more complex and have more redundant backup systems than the current generation (William H. Rogers, personal communication). As a result,
pilots will have more alternatives (i.e., degrees of freedom) that they can choose from in the face of abnormal events. To choose an effective alternative, pilots will need to have a more detailed diagnosis, much like operators in process control systems. This, in turn, means that there will be a greater demand for problem solving and discretionary decision making in order to reconfigure systems in a manner which is appropriate for the given context. In other words, with future generations of aircraft, the pilots’ jobs will become more similar to those of operators in process control systems, where interfaces need to support effective fault diagnosis performance.

**Feedback for proactive control.** To identify problems before they lead to grave consequences requires that one have detailed feedback about the operation of engineering systems. Only with such rich feedback can one proactively deal with problems. Otherwise, compensation can only begin once an alarm is activated, but by this time negative consequences and inefficiencies may have already occurred. For example, Trujillo (1994) cites a significant number of ASRS reports involving slowly developing consequences from failures such as fuel leaks, oil leaks, hydraulics leaks, and engine flameouts. Moreover, she states that in several accidents investigated by the NTSB, fault consequences occurred well before parameters entered into an alarm range. These examples point to the limitations associated with current operating practices that emphasize relying on alarms that only then cue procedures. The examples also show that safety can be improved if the crew is aware of the problem earlier. This is only possible if interfaces provide detailed, salient feedback as to the state of the various systems. The need for proactive monitoring is particularly
great for ETOPS (Extended Transport OPerationS) aircraft which may not have an airport nearby to land at, when a problem arises.

**Feedback for monitoring automation.** Even if the operation of engineering systems is automated, this does not mean that the pilot does not need to be aware of what is occurring in those systems. Since the automation is not perfect, pilots should monitor what the automation is doing. Similarly, if the automation reconfigures a system to deal with a fault, the pilot needs displays with rich feedback to determine what the current configuration of the system is, and how that impacts on the operation of other related systems (see below). Therefore, in order to detect automation failures and to track the response of automated systems effectively, pilots need to have rich feedback. They also need to have a set of expectations about what the system should be doing. An effective interface can help support both of these cognitive activities.

From our observations, it seems that existing displays are not adequately designed to support these functions. This point has repeatedly been made with respect to the flight management systems (opaque operation leading to lack of understanding and failure to detect problems; Wiener, 1988), but the same seems to be true of engineering systems displays. However, the latter have not received the same amount of attention in the literature.

**Revealing interactions between systems.** Current display designs that are used in operation or that are being proposed by researchers are designed to display the state of a single engineering system at a time. We did not find any examples of displays which show interactions between systems, despite the fact that the systems are in fact interconnected,
either physically or functionally. For example, in the C-130, all of the engine displays are collected together in one place but no interactions to other systems (e.g., hydraulics, electrical) are illustrated. Note that the same seems to be true of glass cockpit displays (e.g., the Boeing 767 engine displays). These points of interaction provide paths through which faults can propagate. With current designs, it is up to the pilot (or FE) to retain these interactions in his/her mental model and to reason appropriately in case of a fault. Clearly, this can be a demanding cognitive task, especially under time pressure. It should not be surprising to find, therefore, that "failure to understand all the implications of certain system failures on the capability of other aircraft systems has been cited as a contributing factor in several accident and incident cases" (Palmer & Abbott, 1994, p. 2). To deal with this problem, an interface should clearly reveal the interactions between subsystems.

Supporting multiple fault management strategies. Another important point, which is mentioned by Rogers and Abbott (in press), is that there need not be a conflict between the traditional practice of only providing information to help pilots stabilize for an abnormality, and the practice advocated by EID of supporting deeper diagnosis activities. It is certainly true that the pilot's initial goal in the event of an abnormality is to stabilize the aircraft. Current displays are geared towards supporting this task. However, Rogers & Abbott point out that pilots would also like to subsequently have access to more detailed status information so that they can perform a more detailed diagnosis, after they have stabilized the aircraft. Current displays do not seem to be designed to support this function.
8.2. IMPLICATIONS

In summary, displays that provide rich feedback about engineering system structure and state are required to evaluate system status, to verify actions performed by the automation, and to track interactions caused by the evolution of fault events. In short, better support for KBB is required from advanced engineering status displays (Rogers et al., in press, p. 29). This is precisely the approach taken by EID. Thus, it seems that the principles of EID will become increasingly important in glass cockpit aircraft.

It is unrealistic to expect that the profound changes that would be required to effectively implement EID in aviation would be readily adopted by industry. Fortunately, the HSCT program provides an opportunity to essentially start from scratch. This research program is a rare opportunity for innovation, since many of the constraints that characterize current designs can be relaxed. It seems that the HSCT program thereby provides an important window of opportunity for EID to have an impact on the design of status displays in the aviation industry.
CHAPTER 9: CONCLUSION

This feasibility study has shown that the EID framework can be meaningfully applied to the aviation domain, and that it is not restricted just to process control systems. The AH analysis aided in identifying the content and structure of the information that should be included in the interface. This multilevel representation, which includes both higher order functional information and lower order physical information, provides an informational basis for the supporting of operators during unanticipated events requiring knowledge based behaviour. This information was then mapped onto a set of visual forms that provide a one-to-one mapping between the domain constraints and the perceptual signs provided by the interface. Following are the benefits to the FEs that were predicted (Section 6.9), as well as the observations from the informal evaluation conducted (Section 7.1).

- Operators should be able to easily determine the state of the system at a glance, and to rapidly and effectively extract information from the display. This factor also has implications for training, as the interface should help inexperienced FEs learn how different systems are related to each other. During the evaluation, FEs demonstrated an exceptional ability to use the features of the interface, and all expressed their desire to see a version of the interface used as a training tool.
- The context sensitive approach to the engineering systems, combined with the AH’s accurate system model should reduce the requirement for memorization of both physical and dynamic relationships. At the same time, it should provide a better
conceptual understanding of the co-dependencies within and between systems, as well as the dynamics involved. FEs commented that the interface was complete in its contents of the systems analyzed, and that the relationships between the various systems were apparent and easy to understand.

- Finally, the design is intended to support planning and compensatory activities by allowing operators to directly act on the display surface. FEs were able to easily understand how to operate all components in the interface.

Although some of the literature reviewed (e.g., Veitengruber & Rankin, 1995; Lenorovitz, 1992) stated that the current practice in industry is to use evolutionary rather than revolutionary designs, (because evolutionary designs are easier to understand, and require less training), it seems that the AH, when properly incorporated into the interface, can help designers implement some revolutionary, rather than evolutionary, features that can be well-accepted by users.

The interface described above is only a prototype. Even though initial impressions seem very positive, the entire interface (including unmodeled hydraulic and electronic systems viewports) requires in-depth empirical testing before any definitive conclusions can be drawn. Operator input is necessary, but not sufficient for the interface to be considered a candidate for any real aircraft. There are several features which could perhaps be improved or incorporated into the design specified in this thesis, beyond the comments of the FEs:
- the control of the pan function could perhaps be improved by including a multi-degree of freedom input device that would allow simultaneous movement in two or even three dimensions.

- the zoom feature itself could be potentially improved upon by incorporating Asea Brown Boveri's (ABB's) ideas of additive and alternative information zoom (Elzer, Siebert, and Zinser, 1989).

- more information could be acquired to provide a more realistic and accurate model of engine operation and behaviour.

- all special functions available only on certain components (such as maintenance menu) could be incorporated into all components.

- more than one CRT could be used to reorganize the information displayed.

These and other changes may improve upon the prototype described in this thesis.

Another significant outcome of this study was the reinforcement of the proof of principle that EID can be integrated with other interface concepts. EID was designed to address some basic issues in interface design (Vicente & Rasmussen, 1992). Thus, there are several important design issues for which EID was not intended to provide guidance. Perhaps the most salient example is the issue of navigation. There are many different means to display a multi-level interface the size of the one in this thesis based on an abstraction hierarchy (Vicente, 1992c), but EID does not prescribe which is the most effective, and there is very little empirical evidence on which to base such a decision (Vicente, 1992b). However, the concept of visual momentum (Woods, 1984) was specifically developed to address this issue (Section 6.8.1). In the interface described
above, visual momentum principles were used to determine how different system representations identified in the analysis could be effectively and coherently integrated into a small number of displays.

Similarly, although EID directly addresses information content and structure requirements, it is silent on specific issues related to visual form. EID attempts to guide designers to pursue certain objectives, such as mapping domain constraints onto perceptual cues in the interface, but it does not specify a means of achieving these objectives. Again, however, it was possible to rely on other design principles (i.e., perceptual organization techniques, primarily from the work of Mumaw, Woods, and Eastman, 1992; Woods, 1991) to address this interface design issue (Section 6.8.2).

The above examples reinforce the point made by Vicente and Rasmussen (1992) that following the principles of EID alone does not allow one to design an effective interface for large scale systems. However, they also clearly indicate that EID provides a solid basis for design that addresses fundamental issues, and moreover, that it can be integrated with other important design principles.

In evaluating whether or not EID can be applied to aviation, the creation of an interface is necessary, but not sufficient. The operating philosophies inherent in EID and those currently existing in the aviation industry must also be compared to determine if benefits can be obtained by using the EID approach. The most fundamental issue that needs to be addressed is the intended role of the human in the system. The resolution of this issue will have a strong impact on how displays should be designed, since different display designs are required to support different types of cognitive activities. It was found
that displays which provide rich feedback about engineering system structure and state are required to evaluate system status, to verify actions performed by the automation, and to track interactions caused by the evolution of fault events. In summary, better support for KBB is required from advanced engineering status displays (Rogers et al., in press). This is precisely the approach taken by EID, which indicates that the principles of EID will become increasingly important in glass cockpit aircraft.

In conclusion, although empirical testing is required to evaluate the prototype described in this thesis, this feasibility study has provided a proof of concept showing that the principles of EID can be applied to the aviation domain. It is unrealistic to expect that the profound changes that would be required to effectively implement EID in aviation would be readily adopted by industry. Fortunately, the HSCT program provides an opportunity to essentially start the design process from scratch. This research program is a rare opportunity for innovation, since many of the constraints that characterize current designs can be relaxed. It seems that the HSCT program thereby provides an important window of opportunity for EID to have an impact on the design of status displays in the aviation industry.
CHAPTER 10: REFERENCES


Vicente, K. J. (in press). Improving dynamic decision making in complex systems through ecological interface design: A research overview. *System Dynamics Review*.


Appendices
APPENDIX A: SIMULATOR SESSION PROTOCOL

<table>
<thead>
<tr>
<th>Time (min:sec)</th>
<th>Crew Member</th>
<th>Dialogue</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>FE</td>
<td>Stable</td>
</tr>
<tr>
<td>0:20</td>
<td>AC</td>
<td>Clear 2</td>
</tr>
<tr>
<td></td>
<td>FO</td>
<td>Clear 2</td>
</tr>
<tr>
<td>1:02</td>
<td>AC</td>
<td>You got a clock running on this one?</td>
</tr>
<tr>
<td></td>
<td>FE</td>
<td>Sure do</td>
</tr>
<tr>
<td>1:17</td>
<td>FE</td>
<td>Starter's going</td>
</tr>
<tr>
<td></td>
<td>OB</td>
<td>Incoherent</td>
</tr>
<tr>
<td>1:35</td>
<td>FE</td>
<td>Well, I wanted to. Same reason as a stall, just in case of improper fuel scheduling the other way, the lack of air pressure.</td>
</tr>
<tr>
<td></td>
<td>OB</td>
<td>Incoherent...OK, you've waited 5 min. The snag is removed.</td>
</tr>
<tr>
<td>2:16</td>
<td>AC</td>
<td>Clear 2</td>
</tr>
<tr>
<td></td>
<td>FO</td>
<td>Clear 2</td>
</tr>
<tr>
<td>2:58</td>
<td>FE</td>
<td>Stable</td>
</tr>
<tr>
<td></td>
<td>AC</td>
<td>Clear 1</td>
</tr>
<tr>
<td>3:00</td>
<td>FO</td>
<td>1's Clear</td>
</tr>
<tr>
<td>3:40</td>
<td>FE</td>
<td>Stable</td>
</tr>
<tr>
<td>3:50</td>
<td>FE</td>
<td>Start check's complete</td>
</tr>
<tr>
<td></td>
<td>FE</td>
<td>Taxi check - compass systems</td>
</tr>
<tr>
<td>3:55</td>
<td>FO</td>
<td>It looks like our hydraulic valve isn't closed on #1</td>
</tr>
<tr>
<td></td>
<td>FE</td>
<td>Lovely</td>
</tr>
<tr>
<td>4:10</td>
<td>FE</td>
<td>Both switches off</td>
</tr>
<tr>
<td>4:20</td>
<td>AC</td>
<td>What's happening there?</td>
</tr>
<tr>
<td></td>
<td>FO</td>
<td>Pressure's gainward abnormal so it looks to me like the valves aren't closing</td>
</tr>
<tr>
<td></td>
<td>AC</td>
<td>Off #1 then</td>
</tr>
<tr>
<td></td>
<td>FO</td>
<td>Either that or the switches aren't operating properly</td>
</tr>
<tr>
<td></td>
<td>FO</td>
<td>What have we got here? [looking at the CB] What the hell? Circuit breaker out here on #2 [surprised]</td>
</tr>
<tr>
<td></td>
<td>FE</td>
<td>Hydraulic flare will shut off valves</td>
</tr>
<tr>
<td></td>
<td>AC</td>
<td>What do you want to do with #2?</td>
</tr>
<tr>
<td>4:50</td>
<td>FO</td>
<td>Well, here's what I figured happened. Is that when we started #2, I did the check and everything was the way it should be. When we started #1, the pressure came up. I turned the pump off and the pressure stayed up. I looked over here at the hydraulic fire wall shut off valve and ... is that #2 or #1?</td>
</tr>
<tr>
<td></td>
<td>OB</td>
<td>That's #1.</td>
</tr>
<tr>
<td>5:20</td>
<td>FO</td>
<td>Yeah, #1. Ok, so hydraulic fire wall breaker popped. Reset.</td>
</tr>
<tr>
<td>5:30</td>
<td>FE</td>
<td>Cycle it through again and see what happens</td>
</tr>
<tr>
<td>5:48</td>
<td>FE</td>
<td>Anyway, pre taxi check</td>
</tr>
<tr>
<td></td>
<td>AC</td>
<td>AC</td>
</tr>
<tr>
<td>Time (min:sec)</td>
<td>Crew Member</td>
<td>Dialogue</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>6:25</td>
<td>FO</td>
<td>Co</td>
</tr>
<tr>
<td>6:25</td>
<td>FE</td>
<td>All nav aids and ... IFF</td>
</tr>
<tr>
<td>6:25</td>
<td>AC</td>
<td>AC</td>
</tr>
<tr>
<td>6:25</td>
<td>FO</td>
<td>Co</td>
</tr>
<tr>
<td>7:00</td>
<td>FE</td>
<td>Pre-taxi check complete</td>
</tr>
<tr>
<td>7:05</td>
<td>AC</td>
<td>Taxi Check</td>
</tr>
<tr>
<td>7:20</td>
<td>FE</td>
<td>Taxi Check</td>
</tr>
<tr>
<td>8:50</td>
<td>AC and FO</td>
<td>[Discussing taxiing]</td>
</tr>
<tr>
<td>9:20</td>
<td>FO to the ATC</td>
<td>Canforce 333...</td>
</tr>
<tr>
<td>9:45</td>
<td>FE</td>
<td>Check instruments...</td>
</tr>
<tr>
<td>9:50</td>
<td>AC</td>
<td>AC</td>
</tr>
<tr>
<td>9:50</td>
<td>FO</td>
<td>Co</td>
</tr>
<tr>
<td>10:07</td>
<td>AC</td>
<td>Taxi check's complete</td>
</tr>
<tr>
<td>10:12</td>
<td>FE</td>
<td>Pre take-off check</td>
</tr>
<tr>
<td>10:21</td>
<td>FE</td>
<td>Stand by</td>
</tr>
<tr>
<td>10:21</td>
<td>FE to OB</td>
<td>Fuel gauge. it hasn't gone hard over, it's climbed up</td>
</tr>
<tr>
<td>10:21</td>
<td>FE</td>
<td>You guys probably have a snag in there, but that's almost normal. When you start taxiing, the levels change</td>
</tr>
<tr>
<td>10:30</td>
<td>FE</td>
<td>It's not hard over no. The only time they flip this hard over one way or the other is when they're erratic. So, unless you can make them go hard over, I'm not going to pull [the CB]</td>
</tr>
<tr>
<td>10:47</td>
<td>AC</td>
<td>Secure Pilot</td>
</tr>
<tr>
<td>11:00</td>
<td>OB to FE</td>
<td>Pre takeoff, secure all exits</td>
</tr>
<tr>
<td>11:18</td>
<td>FE</td>
<td>Secure Pilot</td>
</tr>
<tr>
<td>12:00</td>
<td>AC</td>
<td>Normal takeoff. V1 1400 Kts. In emergency, return back here</td>
</tr>
<tr>
<td>13:15</td>
<td>FE</td>
<td>Pre takeoff check's complete</td>
</tr>
<tr>
<td>13:30</td>
<td>ATC</td>
<td>Takeoff cleared on runway 16...</td>
</tr>
<tr>
<td>15:10</td>
<td>ATC</td>
<td>[Takeoff initiated]</td>
</tr>
<tr>
<td>15:36</td>
<td>FO</td>
<td>Abort takeoff, low RSP right side</td>
</tr>
<tr>
<td>15:42</td>
<td>FO</td>
<td>Best I got was it went up to 40 Kts and stuck there.</td>
</tr>
<tr>
<td>16:12</td>
<td>FO</td>
<td>[snag removed]</td>
</tr>
<tr>
<td>16:20</td>
<td>FO</td>
<td>[takeoff re-initiated]</td>
</tr>
<tr>
<td>Time (min:sec)</td>
<td>Crew Member</td>
<td>Dialogue</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>16:46</td>
<td>AC</td>
<td>My wheel.</td>
</tr>
<tr>
<td></td>
<td>FO</td>
<td>Your wheel</td>
</tr>
<tr>
<td>17:05</td>
<td>AC</td>
<td>V1</td>
</tr>
<tr>
<td>17:15</td>
<td>OB</td>
<td>There was significant bird activity on your departure</td>
</tr>
<tr>
<td></td>
<td>FE</td>
<td>I guess so. Turbine overheat on #1</td>
</tr>
<tr>
<td>17:20</td>
<td>AC</td>
<td>Check</td>
</tr>
<tr>
<td>17:30</td>
<td>FE</td>
<td>RPMs #2.</td>
</tr>
<tr>
<td>17:35</td>
<td>FO</td>
<td>Gear's up. #2 is in mechanical.</td>
</tr>
<tr>
<td>17:43</td>
<td>AC</td>
<td>How's #2?</td>
</tr>
<tr>
<td>17:59</td>
<td>AC</td>
<td>Flaps 20</td>
</tr>
<tr>
<td></td>
<td>FO</td>
<td>20's set</td>
</tr>
<tr>
<td>18:10</td>
<td>AC</td>
<td>Got power back on #1. Still got turbine overheat?</td>
</tr>
<tr>
<td>18:15</td>
<td>FE</td>
<td>Still got overheat on #2 here. It's going to blow itself off the wing here in a minute.</td>
</tr>
<tr>
<td>18:20</td>
<td>AC</td>
<td>Go ahead and shut down #2</td>
</tr>
<tr>
<td></td>
<td>FE</td>
<td>#2?!?!?!</td>
</tr>
<tr>
<td>18:24</td>
<td>AC</td>
<td>Still got power out of #1. Shut down #2</td>
</tr>
<tr>
<td></td>
<td>FO</td>
<td>#2</td>
</tr>
<tr>
<td>18:36</td>
<td>FE</td>
<td>Prop's stopped, button's popped, gear's up and flaps are 20</td>
</tr>
<tr>
<td>18:42</td>
<td>AC</td>
<td>Going to be losing #1 soon, let's go ahead and dump fuel to 10000</td>
</tr>
<tr>
<td>18:57</td>
<td>FO</td>
<td>Calgary...suspect bird strike...</td>
</tr>
<tr>
<td>19:17</td>
<td>ATC</td>
<td>Roger...</td>
</tr>
<tr>
<td>19:20</td>
<td>FE</td>
<td>Gotta confirm that it's dumping. The gauge might not be working. Is it still dumping?</td>
</tr>
<tr>
<td>19:37</td>
<td>LM</td>
<td>Still dumping</td>
</tr>
<tr>
<td></td>
<td>AC</td>
<td>What's it look like out there?</td>
</tr>
<tr>
<td></td>
<td>LM</td>
<td>Smoke's pouring out the back.</td>
</tr>
<tr>
<td>19:57</td>
<td>FE</td>
<td>It's giving you very little power.</td>
</tr>
<tr>
<td></td>
<td>AC</td>
<td>Yep. I'm pulling it back here just to get the light out, but it's not going out. Ask if we can get 4700 instead of 7000 [ft]</td>
</tr>
<tr>
<td>20:09</td>
<td>FO</td>
<td>...Unable to climb any higher, requesting 4700 ft.</td>
</tr>
<tr>
<td></td>
<td>ATC</td>
<td>OK...</td>
</tr>
<tr>
<td>20:56</td>
<td>FE</td>
<td>2 engine PMCA is 136 and flaps 20</td>
</tr>
<tr>
<td></td>
<td>AC</td>
<td>Flaps 20. I'm going to leave my flaps 20 and set high boost</td>
</tr>
<tr>
<td>21:13</td>
<td>AC</td>
<td>Is #1 dumping out?</td>
</tr>
<tr>
<td></td>
<td>FE</td>
<td>Yup</td>
</tr>
<tr>
<td>21:30</td>
<td>FE</td>
<td>What I'm thinking is you could get the flaps up and pull the circuit breaker, stick the lever back down so that you can set the high boost</td>
</tr>
<tr>
<td></td>
<td>AC</td>
<td>Sounds good actually</td>
</tr>
<tr>
<td>21:40</td>
<td>AC</td>
<td>Use a little bit more power. Go ahead with flaps up</td>
</tr>
<tr>
<td>22:00</td>
<td>FE to FO</td>
<td>Can you get it Rich? [Reaching for the CB]</td>
</tr>
<tr>
<td>22:13</td>
<td>FO</td>
<td>Circuit breaker is pulled</td>
</tr>
<tr>
<td></td>
<td>FE</td>
<td>Get back on the boost</td>
</tr>
<tr>
<td>Time (min:sec)</td>
<td>Crew Member</td>
<td>Dialogue</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>22:30</td>
<td>FE</td>
<td>Do you want to get the gear down first?</td>
</tr>
<tr>
<td>22:30</td>
<td>AC</td>
<td>Check</td>
</tr>
<tr>
<td>22:30</td>
<td>AC</td>
<td>Shut down #1</td>
</tr>
<tr>
<td>23:00</td>
<td>AC</td>
<td>Let's get the gear down now just in case we have the ??? syndrome</td>
</tr>
<tr>
<td>23:06</td>
<td>FE</td>
<td>Post takeoff complete - all we have to do anyway</td>
</tr>
<tr>
<td>23:19</td>
<td>FE</td>
<td>Still got takeoff power on #3 and #4</td>
</tr>
<tr>
<td>23:26</td>
<td>FE</td>
<td>Losing power on #1</td>
</tr>
<tr>
<td>23:26</td>
<td>AC</td>
<td>Check</td>
</tr>
<tr>
<td>23:26</td>
<td>AC</td>
<td>Shut down #1</td>
</tr>
<tr>
<td>23:32</td>
<td>FO</td>
<td>#1</td>
</tr>
<tr>
<td>23:32</td>
<td>FE</td>
<td>Yeah, looks like it flamed out.</td>
</tr>
<tr>
<td>23:50</td>
<td>FO</td>
<td>Mayday...</td>
</tr>
<tr>
<td>24:20</td>
<td>FO</td>
<td>Left 330</td>
</tr>
<tr>
<td>24:29</td>
<td>AC</td>
<td>We're still dumping fuel. Confirm?</td>
</tr>
<tr>
<td>24:29</td>
<td>FE</td>
<td>We're still dumping fuel. We're down to close to 30000 Just to confirm - you wanted to go to 10?</td>
</tr>
<tr>
<td>25:20</td>
<td>AC</td>
<td>That's affirmative</td>
</tr>
<tr>
<td>25:28</td>
<td>FO</td>
<td>Brakes are emergency Do you want to get the ATM on line?</td>
</tr>
<tr>
<td>25:40</td>
<td>AC</td>
<td>Yeah, it might just give us some help</td>
</tr>
<tr>
<td>26:10</td>
<td>FO to ATC</td>
<td>WE won't be able to clear the runway on landing</td>
</tr>
<tr>
<td>26:10</td>
<td>ATC</td>
<td>Roger, we have confirmed some environmental complaints about your fuel dumping. Several lawyers are contacting DND at this time.</td>
</tr>
<tr>
<td>26:45</td>
<td>AC</td>
<td>Right rudder is getting kind of heavy</td>
</tr>
<tr>
<td>26:53</td>
<td>FE</td>
<td>Just wondering.. any sign of fluid leaks on those engines LM? 1 and 2</td>
</tr>
<tr>
<td>27:00</td>
<td>LM</td>
<td>Don't appear to be.</td>
</tr>
<tr>
<td>27:29</td>
<td>FO</td>
<td>Quite a ways to go. Trying to look for another runway</td>
</tr>
<tr>
<td>28:00</td>
<td>AC</td>
<td>We have no anti-skid, so I'd rather use 16 - it's longer What are we down to on fuel?</td>
</tr>
<tr>
<td>28:14</td>
<td>FE</td>
<td>24000</td>
</tr>
<tr>
<td>28:14</td>
<td>AC</td>
<td>We got 14000 to go. It's dumping kind of slow</td>
</tr>
<tr>
<td>28:25</td>
<td>FO</td>
<td>Yup. What is our dry weight?</td>
</tr>
<tr>
<td>28:50</td>
<td>AC</td>
<td>Jesus, I don't like the way we're descending here. If we get the fuel down, we should be OK. hopefully</td>
</tr>
<tr>
<td>Time (min:sec)</td>
<td>Crew Member</td>
<td>Dialogue</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>29:00</td>
<td>OB</td>
<td>Yeah, the commitment for gear down and heavy wings on 2 engines really hurts you in heat like this. I understand the logic, but it's not that big of an effort to drop the beams and lower the landing gear manually.</td>
</tr>
<tr>
<td>30:00</td>
<td>AC</td>
<td>We're probably going to lose a bit of altitude on this turn.</td>
</tr>
<tr>
<td>30:16</td>
<td>OB to AC</td>
<td>What's the maximum power setting on the gauge used here?</td>
</tr>
<tr>
<td>30:20</td>
<td>AC</td>
<td>1049 I suppose.</td>
</tr>
<tr>
<td>30:43</td>
<td>AC</td>
<td>Let's go for Takeoff power.</td>
</tr>
<tr>
<td></td>
<td>FE</td>
<td>You want Takeoff power now?</td>
</tr>
<tr>
<td></td>
<td>AC</td>
<td>Yeah.</td>
</tr>
<tr>
<td>30:52</td>
<td>FE</td>
<td>set. you got about 800 AGL on the radar altimeter.</td>
</tr>
<tr>
<td>31:06</td>
<td>ATC</td>
<td>Canforce 333...</td>
</tr>
<tr>
<td></td>
<td>FO</td>
<td>Roger 333...</td>
</tr>
<tr>
<td>31:30</td>
<td>ATC</td>
<td>Lights are strength 5</td>
</tr>
<tr>
<td></td>
<td>AC</td>
<td>Airspeed is maintaining and we're climbing a bit.</td>
</tr>
<tr>
<td>32:00</td>
<td>FE</td>
<td>LM, I confirmed earlier that the fuel was dumping, but this Totalizer tells me that it is not. Is it?</td>
</tr>
<tr>
<td>32:06</td>
<td>LM</td>
<td>It's dumping.</td>
</tr>
<tr>
<td>32:28</td>
<td>AC</td>
<td>How far are we from the button of the runway?</td>
</tr>
<tr>
<td></td>
<td>FO</td>
<td>7 NM</td>
</tr>
<tr>
<td></td>
<td>FE</td>
<td>You are about 1 mile from the beacon.</td>
</tr>
<tr>
<td>32:38</td>
<td>AC</td>
<td>I'm starting to pick up some runway lights here.</td>
</tr>
<tr>
<td>33:08</td>
<td>FE</td>
<td>I guess I'm thinking that we might get a possible ...</td>
</tr>
<tr>
<td>33:15</td>
<td>FE</td>
<td>We're basically down to 10000 now. If you want, I can clear the manifold.</td>
</tr>
<tr>
<td>33:20</td>
<td>AC</td>
<td>No, not yet please.</td>
</tr>
<tr>
<td></td>
<td>FE</td>
<td>OK</td>
</tr>
<tr>
<td>33:38</td>
<td>FE</td>
<td>If you want, I can give you the landing check. [perform the landing checklist]</td>
</tr>
<tr>
<td></td>
<td>AC</td>
<td>OK</td>
</tr>
<tr>
<td>34:08</td>
<td>AC</td>
<td>No brakes, no nosewheel steering.</td>
</tr>
<tr>
<td>34:30</td>
<td>FO</td>
<td>Maybe we should stop dumping fuel now. [Aircraft lands - End of Session]</td>
</tr>
<tr>
<td>35:06</td>
<td>AC</td>
<td>Piece of cake.</td>
</tr>
<tr>
<td>35:10</td>
<td>OB</td>
<td>Nicely done. What we've done here is give you a really complex scenario. A lot of things went wrong. You have to make some decisions about runway length, and it's a bit time compressed, so it is important to know what and what not to do. It is unreasonable to you super bad weather, so I gave you something more manageable.</td>
</tr>
</tbody>
</table>

END OF PROTOCOL
APPENDIX B : EXPLANATION OF THE HERCULES FLIGHT ENGINEER LOG (HFEL) ENTRIES

Page 1 of HFEL

Take-Off Data

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airport</td>
<td>Airport flying from</td>
</tr>
<tr>
<td>Fuel Weight Total Gauge</td>
<td>The gauge reading of all tanks containing fuel</td>
</tr>
<tr>
<td>&quot;0&quot; Fuel Weight</td>
<td>Aircraft weight without fuel</td>
</tr>
<tr>
<td>Gross Weight Ramp</td>
<td>Fuel Weight Total Gauge + &quot;0&quot; Fuel Weight</td>
</tr>
<tr>
<td>Altimeter Setting</td>
<td>Airport Information</td>
</tr>
<tr>
<td>Field Elevation/ Press Alt</td>
<td>Airport information</td>
</tr>
<tr>
<td>OAT °C</td>
<td>Outside Air Temperature in °C</td>
</tr>
<tr>
<td>Torque</td>
<td>Maximum torque permitted on Take-off</td>
</tr>
</tbody>
</table>

Running Time

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engines on</td>
<td>The Time (GMT or ZULU) that the engines were started</td>
</tr>
<tr>
<td>Engines off</td>
<td>The Time (GMT or ZULU) that the engines were turned off</td>
</tr>
<tr>
<td>Total</td>
<td>Engines off - Engines on</td>
</tr>
</tbody>
</table>

Landing Data

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airport</td>
<td>Airport flying from</td>
</tr>
<tr>
<td>Altimeter Setting</td>
<td>Airport Information</td>
</tr>
<tr>
<td>Field Elevation/ Press Alt</td>
<td>Airport information</td>
</tr>
<tr>
<td>Landing Weight</td>
<td>Gross Weight at Landing</td>
</tr>
<tr>
<td>Fuel Consumed</td>
<td>Gross Weight at take-off - Gross weight at landing</td>
</tr>
<tr>
<td>Next Fuel Load</td>
<td>To be computed by the FE</td>
</tr>
</tbody>
</table>
Fuel Used to Top Of Climb

Permits the FE to calculate how much fuel was used, and to then estimate whether or not he can afford to climb to another altitude or not. The Multipliers are an approximation of the amount of fuel (Fuel flow) utilized during the portion of the climb. All times are in minutes unless otherwise indicated. Greenwich Mean Time (GMT) and ZULU times are equivalent. The FE can then compare his calculations to the aircraft's instrument readings for comparison purposes. Also, if the fuel instruments were to fail, he would still have a good idea as to how much fuel was still in the tanks.

<table>
<thead>
<tr>
<th>Entry</th>
<th>Multiplier</th>
<th>Consumed</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>T.O Time</td>
<td></td>
<td></td>
<td>Time (GMT or ZULU) that the aircraft took off</td>
</tr>
<tr>
<td>GTC Dur x FF</td>
<td>x5</td>
<td></td>
<td>Time that the Ground Turbine Compressor operated for</td>
</tr>
<tr>
<td>Taxi Dur x FF</td>
<td>x50</td>
<td></td>
<td>Time that the aircraft taxied</td>
</tr>
<tr>
<td>T.O Dur x FF</td>
<td>x150</td>
<td></td>
<td>Time required to take-off</td>
</tr>
<tr>
<td>SOC/TOC</td>
<td>/</td>
<td></td>
<td>Time (ZULU or GMT) that the aircraft began and ended its climb -- Start of</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Climb / Top of Climb</td>
</tr>
<tr>
<td>Climb Dur x FF</td>
<td>x100</td>
<td></td>
<td>Time required to climb</td>
</tr>
<tr>
<td>Cruise Dur x FF</td>
<td>x</td>
<td></td>
<td>Time that the aircraft cruised for -- FE decides, based on speed and</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>altitude, the fuel multiplier.</td>
</tr>
<tr>
<td>SOC/TOC</td>
<td>/</td>
<td></td>
<td>Same as above for another in-flight climb</td>
</tr>
<tr>
<td>Climb Dur x FF</td>
<td>x100</td>
<td></td>
<td>Same as above for another in-flight climb</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td>Total calculated fuel used during the flight</td>
</tr>
</tbody>
</table>

Instructions

A list of instructions inserted as a reminder to the FE's as to how they should complete the HFEL in a standardized manner. Page two is to be completed at the commencement of cruising and hourly thereafter.
**Page 2-3 of HFEL**

**Aircraft Measurements - General**

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time Z</td>
<td>Time of recording (ZULU)</td>
</tr>
<tr>
<td>Pressure Altitude</td>
<td>Outside air pressure at the cruising altitude</td>
</tr>
<tr>
<td>Indicated O.A.T. °C</td>
<td>Indicated outside Air Temperature in °C</td>
</tr>
<tr>
<td>I.A.S. KNOTS</td>
<td>Indicated Air Speed of the Aircraft</td>
</tr>
</tbody>
</table>

**Engine Measurements**

The Numbers next to the entries indicate the engine (1-4) which the individual entry corresponds to.

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Torque</td>
<td>The Torque Value of the Engine</td>
</tr>
<tr>
<td>% RPM</td>
<td>Propeller Rate in Percent RPM</td>
</tr>
<tr>
<td>T.I.T</td>
<td>Turbine Inlet Temperature in °C</td>
</tr>
<tr>
<td>Engine. Oil Temp</td>
<td>Engine Oil Temperature in °C</td>
</tr>
<tr>
<td>Oil Pressure Engine/Gear Box</td>
<td>Oil Pressures of the Engine and the Gear Box</td>
</tr>
<tr>
<td>Eng. Oil Quanttv</td>
<td>Total Amount of Oil in the Engine</td>
</tr>
<tr>
<td>Fuel Flow</td>
<td>Flow of Fuel to Each Engine</td>
</tr>
<tr>
<td>Total</td>
<td>Total Fuel Flow</td>
</tr>
</tbody>
</table>
## Fuel Management Measurements

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tank Contents Gauges</td>
<td>Amount of Fuel Contained in Each of the Tanks According to the Instruments</td>
</tr>
<tr>
<td>Total</td>
<td>Total Fuel According to the Tank Gauges</td>
</tr>
<tr>
<td>Totalizer</td>
<td>Automatic Measurement of Fuel in Tank</td>
</tr>
<tr>
<td>Computed Fuel</td>
<td>Fuel remaining as Calculated by the FE</td>
</tr>
<tr>
<td>Consumed Fuel</td>
<td>Fuel Consumed as Calculated by the FE</td>
</tr>
<tr>
<td>All Up Weight</td>
<td>Total Weight of the Aircraft</td>
</tr>
<tr>
<td>SGR/END</td>
<td>Specific Ground Range Compared to Endurance</td>
</tr>
</tbody>
</table>

1 MAIN
2 MAIN
L.H. AUX
L.H. EXT
R.H. AUX
R.H. EXT
3 MAIN
4 MAIN
## Systems Management Measurements

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prop/Eng. Anti-Icing A/On A/Off</td>
<td></td>
</tr>
<tr>
<td>Wing/ EMP Anti-Icing On Off</td>
<td></td>
</tr>
<tr>
<td>Bleed Air Pressure</td>
<td></td>
</tr>
<tr>
<td>Hydraulics</td>
<td>Amount of Fluid in Each of the Systems Listed</td>
</tr>
<tr>
<td></td>
<td>Utility</td>
</tr>
<tr>
<td></td>
<td>Booster</td>
</tr>
<tr>
<td></td>
<td>Rudder Utility</td>
</tr>
<tr>
<td></td>
<td>Rudder Booster</td>
</tr>
<tr>
<td>Oxygen System Contents</td>
<td>Amount of Oxygen in Tanks</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Pressurization</td>
<td>Cabin Pressure</td>
</tr>
<tr>
<td></td>
<td>Cabin Altitude</td>
</tr>
<tr>
<td></td>
<td>Press Diff</td>
</tr>
<tr>
<td></td>
<td>Difference between inside and outside pressures</td>
</tr>
<tr>
<td>Generators NOR/ABN</td>
<td>Enter whether or not the generators are operating</td>
</tr>
<tr>
<td></td>
<td>normally (NOR) or Abnormally (ABN)</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>T.R.U.'s</td>
<td>Transformer Rectifier Units Measurements</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Unserviceabilities

This space is left for the FE to record all snags encountered during any leg of a mission.

**Aircraft Refueling Report**

The FE is required to investigate any discrepancies in excess of 1000 lb. Following the report is a convenient graph which allows the FE to easily calculate the fuel density variations according to ambient temperature. The density value is then used to calculate the weight of the fuel in the aircraft.

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TANKS</td>
<td></td>
</tr>
<tr>
<td>1 MAIN</td>
<td></td>
</tr>
<tr>
<td>2 MAIN</td>
<td></td>
</tr>
<tr>
<td>L.H. AUX</td>
<td>Amount of Fuel in Each Tank Before and After Being Refilled</td>
</tr>
<tr>
<td>L.H. EXT</td>
<td></td>
</tr>
<tr>
<td>R.H. AUX</td>
<td></td>
</tr>
<tr>
<td>R.H. EXT</td>
<td></td>
</tr>
<tr>
<td>3 MAIN</td>
<td></td>
</tr>
<tr>
<td>4 MAIN</td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>Total Amount of Fuel in all the Tanks</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Uplift By Gauge</th>
<th>Fuel Taken on According to Gauges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uplift By Tender</td>
<td>Fuel Taken on According to Airport</td>
</tr>
<tr>
<td>Discrepancy</td>
<td>Difference Between Gauge and Tender</td>
</tr>
<tr>
<td>Units Added</td>
<td></td>
</tr>
<tr>
<td>Pounds/Unit</td>
<td></td>
</tr>
<tr>
<td>Uplift by Tender</td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX C: FUEL AND ENGINE SPECIFICATION OVERVIEW

The following is a brief introduction to the technical features of the aircraft used as a testbed for this project. This information was obtained from C-130 Flight Engineer Training manuals, Volumes 2 and 3 (Department of National Defence, 1994). The engine information appears on page 1-10 of volume 2, entitled “Aero Engine and Related Systems.”

C.1 ENGINE

Type: Allison T56-A-7B in “E” Model

Allison T56-A-15 in “H” Model

Compressor: Single entry, 14 stage, axial flow

Combustion: Six chambers, through flow type

Turbine: Four stage

Reduction Gear: Two stage reduction 13.54:1

R.P.M. System: Constant speed 100%, Power section R.P.M. 13,820

Reduction Gear: R.P.M. 1,021. Low speed ground idle R.P.M. 69-75.5%

Oil System: Independent dry sump systems. The reduction gear system is made up of one pressure, and two scavenge pumps. The power section is composed of one pressure, and three scavenge pumps. The oil tank and cooler assemblies are nacelle mounted.

Fuel: Hydromechanical, with electric temperature datum control.
C.2 Fuel System

The fuel system is designed to provide flexibility with a minimum of maintenance.

The major difference between the system modeled and that which is described in the FE manual volume 3 entitled “Airframe and Miscellaneous Systems” are the dump valves.

The aircraft dump valves are currently set automatically, but will be retrofit in the future so that their state is controlled by the FEs. The analysis and design presented in this report reflects this upcoming change.

Refueling: Can be accomplished on any combination of tanks simultaneously.

Fuel Feed: Fuel is fed to the engines either directly, or through the crossfeed manifold to the engines. The controllable dump valves allows for fuel to be routed through the refuel manifold as well.

Fuel Dump: When dumping fuel, the dump valves are used. There are certain task constraints associated with dumping fuel and they are as follows:

- Auxiliary and external tanks may be completely emptied.
- Tanks 1 and 4 must retain 323 US gallons
- Tanks 2 and 3 must retain 277 US gallons.

Balancing: A task constraint also exists on the fuel amounts within the tanks.

- Opposite tank pairs must be kept to within 1000 lb. of each other.
- Total weight of the fuel on either side of the aircraft centerline must be kept within 1500 lb. of each other.
APPENDIX D: INSTRUCTIONS ON HOW TO RUN PROTOTYPE

Below are operating instructions for the EID interface for the engineering systems of a Hercules C-130 Model E-H aircraft.

To view the interface, enter the “EID_Hercules_Directory” and type:

EID-Herc

After the system has started up, you may select any of the graphics to obtain a feel for how the interface operates. The Function Keys are special buttons that initiate fault scenarios. Below is a list of Function Keys with the faults they are linked to.

**Keys F1-F3 are used for Engine Faults:**

F1 Resets the system
F2 Causes a Low Fuel Flow Scenario
F3 Causes the Propeller to De-Couple from engine 2

**Keys F5-F8 are used for the Tank Imbalance Scenario:**

F5 Resets the Imbalances
F6 Causes an imbalance between Tank1 and Tank2
F7 and F8 Are left to you to figure out (read the report for the correct answer)

**Keys F9-F10 are used for the faulty valve scenario**

F9 causes valve LL to be faulty
F10 Resets the valve

To quit, press F12
/usr/people/vaps/DCIEM_96/collectors/APU_Flow.COL
/usr/people/vaps/DCIEM_96/collectors/RH_Colour_Green.COL
/usr/people/vaps/DCIEM_96/collectors/Balance_Colour.COL
/usr/people/vaps/DCIEM_96/collectors/RH_Colour_Red.COL
/usr/people/vaps/DCIEM_96/collectors/Balance_Colour_Green.COL
/usr/people/vaps/DCIEM_96/collectors/RH_Colour_Yellow.COL
/usr/people/vaps/DCIEM_96/collectors/Balance_Colour_Red.COL
/usr/people/vaps/DCIEM_96/collectors/RH_Dump_Manifold.COL
/usr/people/vaps/DCIEM_96/collectors/Balance_Colour_State.COL
/usr/people/vaps/DCIEM_96/collectors/RH_Ext_Man_Cap_To_TR.COL
/usr/people/vaps/DCIEM_96/collectors/Balance_Colour_Yello.COL
/usr/people/vaps/DCIEM_96/collectors/RH_Ext_To_Cap_Flow.COL
/usr/people/vaps/DCIEM_96/collectors/Calculation_of_PSFC.COL
/usr/people/vaps/DCIEM_96/collectors/RH_Man_Dump.COL
/usr/people/vaps/DCIEM_96/collectors/Calculation_of_S.COL
/usr/people/vaps/DCIEM_96/collectors/RH_X_Feed.COL
/usr/people/vaps/DCIEM_96/collectors/Calculation_of_SFC.COL
/usr/people/vaps/DCIEM_96/collectors/RH_X_Feed_Pt1.COL
/usr/people/vaps/DCIEM_96/collectors/Dumped.COL
/usr/people/vaps/DCIEM_96/collectors/Dumped_L.COL
/usr/people/vaps/DCIEM_96/collectors/RH_X_Feed_Pt2.COL
/usr/people/vaps/DCIEM_96/collectors/E1_Flow.COL
/usr/people/vaps/DCIEM_96/collectors/Ref_man_Pt1.COL
/usr/people/vaps/DCIEM_96/collectors/E2_Flow.COL
/usr/people/vaps/DCIEM_96/collectors/Ref_man_Pt2.COL
/usr/people/vaps/DCIEM_96/collectors/E3_Flow.COL
/usr/people/vaps/DCIEM_96/collectors/Ref_man_Pt3.COL
/usr/people/vaps/DCIEM_96/collectors/E4_Flow.COL
/usr/people/vaps/DCIEM_96/collectors/Refuel_Manifold_Flow.COL
/usr/people/vaps/DCIEM_96/collectors/FF_Calc.COL
/usr/people/vaps/DCIEM_96/collectors/S.COL
/usr/people/vaps/DCIEM_96/collectors/FF_play_Calc.COL
/usr/people/vaps/DCIEM_96/collectors/SFC.COL
/usr/people/vaps/DCIEM_96/collectors/FF_to_Engine1.COL
/usr/people/vaps/DCIEM_96/collectors/SFC_S_SPFC_Calc.COL
/usr/people/vaps/DCIEM_96/collectors/Fuel_Flow.COL
/usr/people/vaps/DCIEM_96/collectors/Stimulation.COL
/usr/people/vaps/DCIEM_96/collectors/KL_to_SL_Flow.COL
/usr/people/vaps/DCIEM_96/collectors/Sum_of_4_Items.COL
/usr/people/vaps/DCIEM_96/collectors/KR_to_SR_Flow.COL
/usr/people/vaps/DCIEM_96/collectors/Sum_of_Two_Items.COL
/usr/people/vaps/DCIEM_96/collectors/LH_Dump_Manifold.COL
/usr/people/vaps/DCIEM_96/collectors/Tank_amount.COL
/usr/people/vaps/DCIEM_96/collectors/LH_Ext_Man_Cap_To_TL.COL
/usr/people/vaps/DCIEM_96/collectors/Tank_amount_1_4.COL
APPENDIX F: VAPS ATN FOR INTERFACE APPLICATION

UNIT Hercules

DECLARATIONS PRIORITY = 0
{
    # include <math.h>

    int Ent_Plots=0;
    int Fuel_System=1;
    int Fuel_Zoom=0;
    int Engine_Zoom=0;
    int Balance=0;
    int Layer1, Layer2;
    float f=0.0;
    float pot=0.0;
    float zoom=0.0;
    float zoomy=0.75;
    float zoomy=1.2;
    float Tank_Height=1.0;
    float Tank_Height2=1.0;
    float Tank_Height3=1.0;
    float Tank_Height4=1.0;
    int Valve_Pos=1;
}

AUTO MAIN
STATE Start_up
()
{

    *Layer1=Create_Layer(1,0,15,0.0);*
    *Layer2=Create_Layer(2,0,15,0.0);*
    *Set_Current_Layer(Layer1);*
    Get_Frame("Background",0,FOREGROUND_FRAME);
    Get_Frame("map.CRT","map3",0,RASTER_FRAME);
    Get_Frame("polar_star_1",0,FOREGROUND_FRAME);
    Get_Frame("polar_star_2",0,FOREGROUND_FRAME);
    Get_Frame("polar_star_3",0,FOREGROUND_FRAME);
    Get_Frame("polar_star_4",0,FOREGROUND_FRAME);
    Get_Frame("Engine_Thrust",0,FOREGROUND_FRAME);
    Get_Frame("Energy_Env_Penos",0,FOREGROUND_FRAME);
    Drive_Light("E1_Energy_Ref.Energy_Reference.Energy_Ref_Light",1);
    Drive_Light("E2_Energy_Ref.Energy_Reference.Energy_Ref_Light",1);
    Drive_Light("E3_Energy_Ref.Energy_Reference.Energy_Ref_Light",1);
    Drive_Light("E4_Energy_Ref.Energy_Reference.Energy_Ref_Light",1);
    Get_Frame("Energy_Labels",0,FOREGROUND_FRAME);
    Drive_Light("Labels",1);
    Drive_Light("Pos_Lights",2);
    Drive_Light("Pos_Lights",2);
    Get_Frame("Fuel_System_Overview",0,FOREGROUND_FRAME);
    *Set_Current_Layer(Layer2);*
    Get_Frame("map.CRT","Map_Distance",0,FOREGROUND_FRAME);
    Get_Frame("Big_Viewport_CRT","Fuel_system",0,FOREGROUND_FRAME);
    Scale_Frame("Big_Viewport_CRT","Fuel_system",1.1.5);
    Get_Frame("Big_Viewport_CRT","Fuel_Zoomed",0,MAP_FRAME);
    Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
    Get_Frame("Tank_Light_2",0,FOREGROUND_FRAME);
    Hide_Frame("Tank_Light_2");
    Get_Frame("Big_Viewport_CRT",Fuel_Balances",0,FOREGROUND_FRAME);
    Hide_Frame("Big_Viewport_CRT",Fuel_Balances");
    Get_Frame("Big_Viewport_CRT","Engine_Zoomed",0,MAP_FRAME);
    Hide_Object("Big_Viewport_CRT",Engine_Zoomed,Menu");
    Hide_Frame("Big_Viewport_CRT","Engine_Zoomed");
    Get_Frame("kbd_gen_event",0,FOREGROUND_FRAME);
    Get_ATN("Periodic_ATN");
}
(ROTOR map_CRT.Map_Distance.Zoom_Rotor '500')
{
    Scale_Frame("map_CRT","map3",2.2);
} -> Monitor

(ROTOR map_CRT.Map_Distance.Zoom_Rotor '250')
{
    Scale_Frame("map_CRT","map3",3.3);
} -> Monitor

(ROTOR map_CRT.Map_Distance.Zoom_Rotor '2000')
{
    Scale_Frame("map_CRT","map3",1.1);
} -> Monitor

(SWITCH Fuel_Overview_Sw 'Off')
{
    if (Balance==11)
    {
        Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
        Balance=0;
        *Set_Current_Layer(Layer2);*
        if (Fuel_System==1) {
            Hide_Frame("Big_Viewport_CRT","Fuel_system");
            Fuel_System=0;
            Drive_Light("Pot_Lights",2);
        }
        if (Fuel_Zoom==1) {
            Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
            Fuel_Zoom=0;
        }
    }
}

} -> Monitor

(SWITCH Fuel_Overview_Sw 'L_Wing_Tank')
-> Monitor

(SWITCH Fuel_Overview_Sw 'Balance')
{
    Appear_Frame("Energy_Ent_Plots2");
    Drive_Switch("E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch("E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch("E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch("E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Light("E1_Energy_ref.Energy_Reference.Energy_Ref_Light",1);
    Drive_Light("E2_Energy_ref.Energy_Reference.Energy_Ref_Light",1);
    Drive_Light("E3_Energy_ref.Energy_Reference.Energy_Ref_Light",1);
    Drive_Light("E4_Energy_ref.Energy_Reference.Energy_Ref_Light",1);
    *Set_Current_Layer(Layer2);*
    if (Fuel_System==1) {
        Hide_Frame("Big_Viewport_CRT","Fuel_system");
        Fuel_System=0;
    }
    if (Fuel_Zoom==1) {
        Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
        Fuel_Zoom=0;
        Hide_Frame("Tank_Light_2_Test");
    }
    if (Engine_Zoom==1) {
        Hide_Frame("Big_Viewport_CRT","Engine_Zoomed");
        Engine_Zoom=0;
    }
    Drive_Switch("Engine_Selector1",14);
    Drive_Switch("Engine_Selector2",14);
Drive_Switch("".""Engine_Selector3","14");
Drive_Switch("".""Engine_Selector4","14");

if (Balance==0) {
    Appear_Frame("Big_Viewport_CRT","Fuel_Balances");
    Scale_Frame("Big_Viewport_CRT","Fuel_Balances","1.1.1.5");
    Balance=1.
}

Drive_Light("".""Pot_Lights","2");
Drive_Light("".""Pot_Lights","1.2");
} -> Monitor
(SWITCH Fuel_Overview_Sw 'R_Wing_Tank')
} -> Monitor
(SWITCH Fuel_Overview_Sw 'L_Refuel')
{
    if (Balance==1) {
        Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
        Balance=0;
    }
    if (Fuel_System==1) {
        Hide_Frame("Big_Viewport_CRT","Fuel_system");
    }
    if (Engine_Zoom==1) {
        Hide_Frame("Big_Viewport_CRT","Engine_Zoomed");
        Drive_Switch("".""Engine_Selector1","14");
        Drive_Switch("".""Engine_Selector2","14");
        Drive_Switch("".""Engine_Selector3","14");
        Drive_Switch("".""Engine_Selector4","14");
        Drive_Light("".""Pot_Lights","1.2");
    }
    if (Fuel_Zoom==0) {
        Hide_Frame("".""Energy_Ent_Plots2");
        Hide_Frame("".""Energy_Labels");
        "Set_Current_Layer(Layer2):*
        Appear_Frame("Big_Viewport_CRT","Fuel_Zoomed");
        Appear_Frame("".""Tank_Light_2_Tank");
        Fuel_System=0;
        Fuel_Zoom=1;
        Engine_Zoom=0;
    }
    "Set_Current_Layer(Layer1):*
    Drive_Light("".""Pot_Lights","1");
    Drive_Light("".""Pot_Lights","1.2");
potx=1765;
poty=561;

Channel_Member_Put_Buffer."Session_Common","".""Location",MEMBER_DIMEN(1,10.1.CHAN_FLOAT.&potx.1.CHANNEL_WRITE);

Channel_Member_Put_Buffer."Session_Common","".""Location",MEMBER_DIMEN(1,11.1.CHAN_FLOAT.&poty.1.CHANNEL_WRITE);

Drive_Pot("".""vertical",poty);
Drive_Pot("".""horizontal",potx);

"Set_Current_Layer(Layer2):*
Scale_Frame("Big_Viewport_CRT","Fuel_Zoomed",zoomx,zoomy);
Drive_Light("".""Tank_Zoom_Light","1");

Drive_Light("".""Pot_Lights","1.2");
} -> Monitor
(SWITCH Fuel_Overview_Sw 'R_Refuel')
{
    if (Balance==1){
Hide_Frame("Big_Viwrport_CRT","Fuel_Balances");
Balance=0;

} if (Fuel_System==1) {
Hide_Frame("Big_Viwrport_CRT","Fuel_system");
}
if (Engine_Zoom==1) {
Hide_Frame("Big_Viwrport_CRT","Engine_Zoomed");
Drive_Switch("Engine_Selector1",14);
Drive_Switch("Engine_Selector2",14);
Drive_Switch("Engine_Selector3",14);
Drive_Switch("Engine_Selector4",14);
}

if (Fuel_Zoom==0) {
Hide_Frame("Energy_Ent_Plot2");
Hide_Frame("Energy_Labels");

"Set_Current_LayerLayer2");
Appear_Frame("Big_Viwrport_CRT","Fuel_Zoomed");
Appear_Frame("Tank_Light_2_Test");
Fuel_System=0;
Fuel_Zoom=1;
Engine_Zoom=0;

}

"Set_Current_LayerLayer1");
Drive_Light("Pot_Lights",1);
Drive_Light("Pot_Lights_1",2);
pox=578;
poy=561.

Channel_Member_Put_Buff("Session_Common","Location",MEMBER_DIMEN(1),10,1,CHAN_FLOAT,&pox,1,CHANNEL_WRITE);

Channel_Member_Put_Buff("Session_Common","Location",MEMBER_DIMEN(1),11,1,CHAN_FLOAT,&poy,1,CHANNEL_WRITE);

Drive_Pott("vertical",poy);
Drive_Pott("horizontal",pox);

"Set_Current_LayerLayer2");
Scale_Frame("Big_Viwrport_CRT","Fuel_Zoomed",zoomx,zoomy);
Drive_Light("Tank_Zoom_Light",12);

Drive_Light("Pot_Lights_1",2);
}

> Monitor

SWITCH Fuel_Overview_Sw 'Tank_1"
{
if (Balance==1)
Hide_Frame("Big_Viwrport_CRT","Fuel_Balances");
Balance=0;
}
if (Fuel_System==1) {
Hide_Frame("Big_Viwrport_CRT","Fuel_system");
}
if (Engine_Zoom==1) {
Hide_Frame("Big_Viwrport_CRT","Engine_Zoomed");
Drive_Switch("Engine_Selector1",14);
Drive_Switch("Engine_Selector2",14);
Drive_Switch("Engine_Selector3",14);
Drive_Switch("Engine_Selector4",14);
}

if (Fuel_Zoom==0) {
Hide_Frame("Energy_Ent_Plot2");
Hide_Frame("Energy_Labels");
*Set_Current_Layer(Layer2)::*
Appear_Frame("Big_Viewport_CR","Fuel_Zoomed");
Appear_Frame("".""Tank_Light_2","_Test");
Fuel_System=0;
Fuel_Zoom=1;
Engine_Zoom=0;
}

*Set_Current_Layer(Layer1)::*
Drive_Light("".""Pot_Lights",1);
Drive_Light("".""Pot_Lights_1","_2");
potx=2089;
poty=873;

Channel_Put_Buffer("Session_Common","","Location",MEMBER_DIMEN(1),10,1.CHAN_FLOAT,&potx,1.CHANNEL_WRITE);

Channel_Put_Buffer("Session_Common","","Location",MEMBER_DIMEN(1),11,1.CHAN_FLOAT,&poty,1.CHANNEL_WRITE);

Drive_Pot("".""vertical",poty);
Drive_Pot("".""horizontal",potx);

*Set_Current_Layer(Layer2)::*
Scale_Frame("Big_Viewport_CR","Fuel_Zoomed",zoomx,zoomy);
Drive_Light("".""Tank_Zoom_Light",2);

Drive_Light("".""Pot_Lights_1","_2");
} -> Monitor
(SWITCH Fuel_Overview_Sw Tank_2)
{
if (Balance==1){
Hide_Frame("Big_Viewport_CR","Fuel_Balances");
Balance=0;
}
if (Fuel_System==1) {
Hide_Frame("Big_Viewport_CR","Fuel_system");
}
if (Engine_Zoom==1) {
Hide_Frame("Big_Viewport_CR","Engine_Zoomed");
Drive_Switch("".""Engine_Selector1",14);
Drive_Switch("".""Engine_Selector2",14);
Drive_Switch("".""Engine_Selector3",14);
Drive_Switch("".""Engine_Selector4",14);
}

if (Fuel_Zoom==0) {
Hide_Frame("".""Energy_Ent_Plots2");
Hide_Frame("".""Energy_Labels");

*Set_Current_Layer(Layer2)::*
Appear_Frame("Big_Viewport_CR","Fuel_Zoomed");
Appear_Frame("".""Tank_Light_2","_Test");
Fuel_System=0;
Fuel_Zoom=1;
Engine_Zoom=0;
}

*Set_Current_Layer(Layer1)::*
Drive_Light("".""Pot_Lights",1);
Drive_Light("".""Pot_Lights_1","_2");
potx=1726;
poty=873;

Channel_Put_Buffer("Session_Common","","Location",MEMBER_DIMEN(1),10,1.CHAN_FLOAT,&potx,1.CHANNEL_WRITE);
Channel_Member_Put_Buffert"Session_Common"."""."Location".MEMBER_DIMEN(1),11,1.CHAN_FLOAT.&poy.1.CHANNEL_WRITE
Ex;

  Drive_Pot"""".vertical".poy1;
  Drive_Pot"""".horizontal".pots1;

  "Set_Current_Layer(Layer2);"
  Scale_Frame("Big_Viewport_CRT","Fuel_Zoomed",zoomx,zoomy);
  Drive_Light"""".Tank_Zoom_Light",2);

  Drive_Light"""".Pot_Lights",1",2);
} -> Monitor

(SWITCH Fuel_Overview_Sw/"Aux_Tank_L")
{
  if (Balance==1){
    Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
    Balance=0;
  }
  if (Fuel_System==1) {
    Hide_Frame("Big_Viewport_CRT","Fuel_system");
  }
  if (Engine_Zoom==1) {
    Hide_Frame("Big_Viewport_CRT","Engine_Zoomed");
    Drive_Switch("""".Engine_Selector1",14);
    Drive_Switch("""".Engine_Selector2",14);
    Drive_Switch("""".Engine_Selector3",14);
    Drive_Switch("""".Engine_Selector4",14);
  }

  if (Fuel_Zoom==1) {
    Hide_Frame(""""."Energy_Ent_Plots2");
    Hide_Frame(""""."Energy_Labels");

    "Set_Current_Layer(Layer2);"
    Appear_Frame("Big_Viewport_CRT","Fuel_Zoomed");
    Appear_Frame(""""."Tank_Light_2_Text");
    Fuel_System=0;
    Fuel_Zoom=1;
    Engine_Zoom=0;
  }

  "Set_Current_Layer(Layer1);"
  Drive_Light"""".Pot_Lights",1"
  Drive_Light"""".Pot_Lights",1",2"
  poy=1270;
  poy=873;

Channel_Member_Put_Buffert"Session_Common"."""."Location".MEMBER_DIMEN(1).10,1.CHAN_FLOAT.&poy.1.CHANNEL_WRITE
Ex);

Channel_Member_Put_Buffert"Session_Common"."""."Location".MEMBER_DIMEN(1).11,1.CHAN_FLOAT.&poy.1.CHANNEL_WRITE
Ex;

  Drive_Pot"""".vertical".poy1;
  Drive_Pot"""".horizontal".pots1;

  "Set_Current_Layer(Layer2);"
  Scale_Frame("Big_Viewport_CRT","Fuel_Zoomed",zoomx,zoomy);
  Drive_Light"""".Tank_Zoom_Light",4);
} -> Monitor

(SWITCH Fuel_Overview_Sw/"Ext_Tank_L")
{
  if (Balance==1){
    Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
    Balance=0;
  }
  if (Fuel_System==1) {
    Hide_Frame("Big_Viewport_CRT","Fuel_system");
  }
} if (Engine_Zoom==1) {
    Hide_Frame("Big_Viewport_CRT","Engine_Zoomed");
    Drive_Switch("-Engine_Selector","14");
    Drive_Switch("-Engine_Selector2","14");
    Drive_Switch("-Engine_Selector3","14");
    Drive_Switch("-Engine_Selector4","14");
}

} if (Fuel_Zoom==0) {
    Hide_Frame("-Energy_Ent_Plots2");
    Hide_Frame("-Energy_Labels");
    *Set_Current_Layer(Layer2)*;
    Appear_Frame("Big_Viewport_CRT","Fuel_Zoomed");
    Appear_Frame("-Tank_Light_2_Test");
    Fuel_System=0;
    Fuel_Zoom=1;
    Engine_Zoom=0;
}

*Set_Current_Layer(Layer1)*;
Drive_Light("-Pot_Lights","1");
Drive_Light("-Pot_Lights_1","2");
potx=1720;
poty=0;

Channel_Member_Put_BUFFERT "Session_Common".;"",""Location".MEMBER_DIMEN(1),10.1.CHAN_FLOAT.&potx,1.CHANNEL_WRITE
En;

Channel_Member_Put_BUFFERT "Session_Common".;"",""Location".MEMBER_DIMEN(1),11.1.CHAN_FLOAT.&poty,1.CHANNEL_WRITE
En;

    Drive_Pott("","vertical",poty);
    Drive_Pott("","horizontal",potx);

    *Set_Current_Layer(Layer2)*;
    Scale_Frame("Big_Viewport_CRT","Fuel_Zoomed",zoomx,zoomy);
    Drive_Light("-Tank_Zoom_Light","5");
} -> Monitor
(SWITCH Fuel_Overview_Sw 'Ext_Tank_R')
{
    if (Balance==1) {
        Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
        Balance=0;
    }
    if (Fuel_System==1) {
        Hide_Frame("Big_Viewport_CRT","Fuel_system");
    }
    if (Engine_Zoom==1) {
        Hide_Frame("Big_Viewport_CRT","Engine_Zoomed");
        Drive_Switch("-Engine_Selector","14");
        Drive_Switch("-Engine_Selector2","14");
        Drive_Switch("-Engine_Selector3","14");
        Drive_Switch("-Engine_Selector4","14");
    }
}

} if (Fuel_Zoom==0) {
    Hide_Frame("-Energy_Ent_Plots2");
    Hide_Frame("-Energy_Labels");
    *Set_Current_Layer(Layer2)*;
    Appear_Frame("Big_Viewport_CRT","Fuel_Zoomed");
    Appear_Frame("-Tank_Light_2_Test");
    Fuel_System=0;
    Fuel_Zoom=1;
    Engine_Zoom=0;
};

*Set_Current_Layer(Layer1)*/
Drive_Light".""Pot_Lights".1;
  Drive_Light".""Pot_Lights".1","2);
  potx=700;
poty=0;

Channel_Member_Put_Buffert"Session_Common","".""Location",MEMBER_DIMEN(1),10.1.CHAN_FLOAT,&potx.1.CHANNEL_WRIT E);

Channel_Member_Put_Buffert"Session_Common","".""Location",MEMBER_DIMEN(1),11.1.CHAN_FLOAT,&poty.1.CHANNEL_WRIT E);

  Drive_Pott","vertical","poty);
  Drive_Pott","horizontal","potx.1);

*Set_Current_Layer(Layer2)*/
  Scale_Frame("Big_Viewport_CRT","Fuel_Zoomed",zoomx, zoomy);
  Drive_Light".""Tank_Zoom_Light".6);
}) -> Monitor
(SWITCH Fuel_Overview_Sw 'Aus_Tank_R')
{
  if (Balance==1){
    Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
    Balance=0;
  }
  if (Fuel_System==1) {
    Hide_Frame("Big_Viewport_CRT","Fuel_system");
  }
  if (Engine_Zoom==1) {
    Hide_Frame("Big_Viewport_CRT","Engine_Zoomed");
    Drive_Switch","".Engine_Selector1",14);
    Drive_Switch","".Engine_Selector2",14);
    Drive_Switch","".Engine_Selector3",14);
    Drive_Switch","".Engine_Selector4",14);
  }
  if (Fuel_Zoom==0) {
    Hide_Frame".""Energy_Ent_Plot2".);
    Hide_Frame".""Energy_Labels".);

*Set_Current_Layer(Layer2)*/
  Appear_Frame("Big_Viewport_CRT","Fuel_Zoomed");
  Appear_Frame".""Tank_Light_2",Test");
  Fuel_System=0;
  Fuel_Zoom=1;
  Engine_Zoom=0;
}

*Set_Current_Layer(Layer1)*/
  Drive_Light".""Pot_Lights".1;
  Drive_Light".""Pot_Lights".1","2);
  potx=1021;
poty=873;

Channel_Member_Put_Buffert"Session_Common","".""Location",MEMBER_DIMEN(1),10.1.CHAN_FLOAT,&potx.1.CHANNEL_WRIT E;

Channel_Member_Put_Buffert"Session_Common","".""Location",MEMBER_DIMEN(1),11.1.CHAN_FLOAT,&poty.1.CHANNEL_WRIT E);

  Drive_Pott","vertical","poty);
  Drive_Pott","horizontal","potx.1);

*Set_Current_Layer(Layer2)*/
  Scale_Frame("Big_Viewport_CRT","Fuel_Zoomed",zoomx, zoomy);
  Drive_Light".""Tank_Zoom_Light".7);
}) -> Monitor
(SWITCH Fuel_Overview_Sw 'Tank_3')
{
  if (Balance==1) {
    Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
    Balance=0;
  }
  if (Fuel_System==1) {
    Hide_Frame("Big_Viewport_CRT","Fuel_system");
  }
  if (Engine_Zoom==1) {
    Hide_Frame("Big_Viewport_CRT","Engine_Zoomed");
    Drive_Switch("".""Engine_Selector1"."",14);
    Drive_Switch("".""Engine_Selector2"."",14);
    Drive_Switch("".""Engine_Selector3"."",14);
    Drive_Switch("".""Engine_Selector4"."",14);
  }
  if (Fuel_Zoom==0) {
    Hide_Frame("".""Energy_Ent_Plots2"."");
  }
  Hide_Frame("".""Energy_Labels"");
  *Set_Current_Layer(Layer2);*
  Appear_Frame("Big_Viewport_CRT","Fuel_Zoomed");
  Appear_Frame("".""Tank_Light_2_Test"".");
  Fuel_System=0;
  Fuel_Zoom=1;
  Engine_Zoom=0;
}

*Set_Current_Layer(Layer1);*
Drive_Light("".""Put_Lights""."",1i);
Drive_Light("".""Put_Lights_1""."",2i);
poxs=548;
poty=873;

Channel_Member_Put_Buffer("Session_Common","".""Location"".
MEMBER_DIMEN(11,10,1.CHAN_FLOAT,&poxs,1.CHANNEL_WRITE_E);

Channel_Member_Put_Buffer("Session_Common","".""Location"".
MEMBER_DIMEN(11,1.1.CHAN_FLOAT,&poty,1.CHANNEL_WRITE_E);

  Drive_Pott("".""Vertical","".""poty");
  Drive_Pott("".""Horizontal","".""poxs");

  *Set_Current_Layer(Layer2);*
  Scale_Frame("Big_Viewport_CRT","Fuel_Zoomed","zoomx","zoomy");
  Drive_Light("".""Tank_Zoom_Light""."",8i);
}
} -> Monitor

(SWITCH Fuel_Overview_Sw 'Tank_4')
{
  if (Balance==1) {
    Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
    Balance=0;
  }
  if (Fuel_System==1) {
    Hide_Frame("Big_Viewport_CRT","Fuel_system");
  }
  if (Engine_Zoom==1) {
    Hide_Frame("Big_Viewport_CRT","Engine_Zoomed");
    Drive_Switch("".""Engine_Selector1"."",14);
    Drive_Switch("".""Engine_Selector2"."",14);
    Drive_Switch("".""Engine_Selector3"."",14);
    Drive_Switch("".""Engine_Selector4"."",14);
  }
  if (Fuel_Zoom==0) {
    Hide_Frame("".""Energy_Ent_Plots2"".");
  }
Hide_Frame("","Energy_Labels");

/* Set_Current_LayerLayer2 */
  Appear_Frame("Big_Viewport_CRT","Fuel_Zoomed");
  Appear_Frame("","Tank_Light_2_Test");
  Fuel_System=0;
  Fuel_Zoom=1;
  Engine_Zoom=0;
}

/* Set_Current_LayerLayer1 */
  Drive_Light("","Pot_Lights",1);
  Drive_Light("","Pot_Lights_1",2);
  potx=119;
  poty=873;

Channel_Member_Put_Buffer("Session_Common","","Location",MEMBER_DIMEn11,10,1.CHAN.FLOAT,&potx,1.CHANNEL_WRITE);

Channel_Member_Put_Buffer("Session_Common","","Location",MEMBER_DIMEn11,11,1.CHAN.FLOAT,&poty,1.CHANNEL_WRITE);
  Drive_Pot("","vertical",poty);
  Drive_Pot("","horizontal",potx);

/* Set_Current_LayerLayer2 */
  Scale_Frame("Big_Viewport_CRT","Fuel_Zoomed",zoomx,zoomy);
  Drive_Light("","Tank_Zoom_Light",9);
}

} -> Monitor
(SWITCH Fuel_Overview_Sw 'RHDump')
{
  if (Balance==1)
    Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
    Balance=0;
  }
  if (Fuel_System==1) {
    Hide_Frame("Big_Viewport_CRT","Fuel_system");
  }
  if (Engine_Zoom==1) {
    Hide_Frame("Big_Viewport_CRT","Engine_Zoomed");
    Drive_Switch("","Engine_Selector1",14);
    Drive_Switch("","Engine_Selector2",14);
    Drive_Switch("","Engine_Selector3",14);
    Drive_Switch("","Engine_Selector4",14);
  }
  if (Fuel_Zoom==0) {
    Hide_Frame("","Energy_ENT_Plots2");
    Hide_Frame("","Energy_Labels");

/* Set_Current_LayerLayer2 */
  Appear_Frame("Big_Viewport_CRT","Fuel_Zoomed");
  Appear_Frame("","Tank_Light_2_Test");
  Fuel_System=0;
  Fuel_Zoom=1;
  Engine_Zoom=0;
}

/* Set_Current_LayerLayer1 */
  Drive_Light("","Pot_Lights",1);
  Drive_Light("","Pot_Lights_1",2);
  potx=313;
  poty=409;
}

Channel_Member_Put_Buffer("Session_Common","","Location",MEMBER_DIMEn11,10,1.CHAN.FLOAT,&potx,1.CHANNEL_WRITE);
Channel_Member_Put_Buffer "Session_Common", "", "Location", MEMBER_DIMEN(1), 11.1.CHAN_FLOAT, &pox, 1.CHANNEL_WRITE_E;
    Drive_Pot("vertical", poty);  
    Drive_Pot("horizontal", potx);  
    * "Set_Current_LayerLayer2", "*
    Scale_Frame("Big_Viewport_CRT", "Fuel_Zoomed", zoomx, zoomy);  
    Drive_Light("Tank_Zoom_Light", 10);  
} -> Monitor
(SWITCH Fuel_Overview_Sw 'LH_X_Feed')
{
    if (Balance==1) {  
        Hide_Frame("Big_Viewport_CRT", "Fuel_Balances");  
        Balance=0;  
    }
    if (Fuel_System==1) {  
        Hide_Frame("Big_Viewport_CRT", "Fuel_system");  
    }
    if (Engine_Zoom==1) {  
        Hide_Frame("Big_Viewport_CRT", "Engine_Zoomed");  
        Drive_Switch("Engine_Selector1", 14);  
        Drive_Switch("Engine_Selector2", 14);  
        Drive_Switch("Engine_Selector3", 14);  
        Drive_Switch("Engine_Selector4", 14);  
    }
    if (Fuel_Zoom==0) {  
        Hide_Frame("Energy_Ent_Plots2");  
        Hide_Frame("Energy_Labels");  
    }
    * "Set_Current_LayerLayer2", "*
    Appear_Frame("Big_Viewport_CRT", "Fuel_Zoomed");  
    Appear_Frame("Tank_Light_2_Test");  
    Fuel_System=0;  
    Fuel_Zoom=1;  
    Engine_Zoom=0;  
}

* "Set_Current_LayerLayer1", "*
    Drive_Light("Pot_Lights", 1);  
    Drive_Light("Pot_Lights_1", 2);  
    potx=1720;  
    poty=1395;  
}

Channel_Member_Put_Buffer "Session_Common", "", "Location", MEMBER_DIMEN(1), 10.1.CHAN_FLOAT, &pox, 1.CHANNEL_WRITE_E;

Channel_Member_Put_Buffer "Session_Common", "", "Location", MEMBER_DIMEN(1), 11.1.CHAN_FLOAT, &pox, 1.CHANNEL_WRITE_E;
    Drive_Pot("vertical", poty);  
    Drive_Pot("horizontal", potx);  
    * "Set_Current_LayerLayer2", "*
    Scale_Frame("Big_Viewport_CRT", "Fuel_Zoomed", zoomx, zoomy);  
    Drive_Light("Tank_Zoom_Light", 13);  
} -> Monitor
(SWITCH Fuel_Overview_Sw 'RH_X_Feed')
{
    if (Balance==1) {  
        Hide_Frame("Big_Viewport_CRT", "Fuel_Balances");  
        Balance=0;  
    }
    if (Fuel_System==1) {  
        Hide_Frame("Big_Viewport_CRT", "Fuel_system");  
    }
    if (Engine_Zoom==1) {  

}
if (Fuel_Zero==0) {
    Hide_Frame(""."".fuel_balances");
    Balance=0;
}

if (Fuel_System==1) {
    Hide_Frame(""."".fuel_system");
}

if (Engine_Zoom==1) {
    Hide_Frame(""."".engine_zoomed");
    Drive_Switch(""."".engine_selector1",14);
    Drive_Switch(""."".engine_selector2",14);
    Drive_Switch(""."".engine_selector3",14);
    Drive_Switch(""."".engine_selector4",14);
}

if (Fuel_Zoom==0) {
    Hide_Frame(""."".energy_ent_plots2");
    Hide_Frame(""."".energy_labels");

    *Set_Current_LayerLayer2;*
    Appear_Frame(""."".fuel_zoomed");
    Appear_Frame(""."".tank_light_2_test");
    Fuel_System=0;
    Fuel_Zoom=1;
    Engine_Zoom=0;
}

*Set_Current_LayerLayer1;*
Drive_Light(""."".pot_lights",1).
Drive_Light(""."".pot_lights_1",2);
potx=572;
poty=1395;

Channel_Member_Pat_Buffer"Session_Common",""."".Location".MEMBER_DIMEN(11.12.CHAN_FLOAT,&potx,1.CHANNEL_WRITE);

Channel_Member_Pat_Buffer"Session_Common",""."".Location".MEMBER_DIMEN(11.12.CHAN_FLOAT,&poty,1.CHANNEL_WRITE);

Drive_Pot(""."".vertical",poty);
Drive_Pot(""."".horizontal",potx);

*Set_Current_LayerLayer2;*
Scale_Frame(""."".tank_zoom_light",zoomx,zoomy);
Drive_Light(""."".tank_zoom_light",14);

} => Monitor
(SWITCH Fuel_Overview_Sw "El")
{
    if (Balance==1){
        Hide_Frame(""."".fuel_balances");
        Balance=0;
    }

    if (Fuel_System==1) {
        Hide_Frame(""."".fuel_system");
    }

    if (Engine_Zoom==1) {
        Hide_Frame(""."".engine_zoomed");
        Drive_Switch(""."".engine_selector1",14);
        Drive_Switch(""."".engine_selector2",14);
        Drive_Switch(""."".engine_selector3",14);
        Drive_Switch(""."".engine_selector4",14);
    }

    if (Fuel_Zoom==0) {
        Hide_Frame(""."".energy_ent_plots2");
        Hide_Frame(""."".energy_labels");

        *Set_Current_LayerLayer2;*
        Appear_Frame(""."".fuel_zoomed");
        Appear_Frame(""."".tank_light_2_test");
        Fuel_System=0;
        Fuel_Zoom=1;
        Engine_Zoom=0;
    }
*Set_Current_LayerLayer1*;
Drive_Light"1","Pot_Lights","1";
Drive_Light"2","Pot_Lights","1",2);
pots=2095;
poty=1422;

Channel_Member_Put_Buffer"Session_Common","Location",MEMBER_DIMEN(1),10.1.CHAN_FLOAT,&potx,1.CHANNEL_WRITE;

Channel_Member_Put_Buffer"Session_Common","Location",MEMBER_DIMEN(1),11.1.CHAN_FLOAT,&poty,1.CHANNEL_WRITE;

Drive_Pot"","vertical",poty;
Drive_Pot"","horizontal",potx;

*Set_Current_LayerLayer2*;
Scale_Frame("Big_Viewport_CRT","Fuel_Zoomed",zoomx,zoomy);
Drive_Light"9","Tank_Zoom_Light",15;
}

-> Monitor
(SWITCH Fuel_Overview_Sw 'E2')
{
    if (Balance==1) {
        Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
        Balance=0;
    }
    if (Fuel_System==1) {
        Hide_Frame("Big_Viewport_CRT","Fuel_system");
    }
    if (Engine_Zoom==1) {
        Hide_Frame("Big_Viewport_CRT","Engine_Zoomed");
        Drive_Switch"1","Engine_Selector1",14;
        Drive_Switch"2","Engine_Selector2",14;
        Drive_Switch"3","Engine_Selector3",14;
        Drive_Switch"4","Engine_Selector4",14;
    }

    if (Fuel_Zoom==0) {
        Hide_Frame("Energy_Env_Plots2");
        Hide_Frame("Energy_Labels");

        *Set_Current_LayerLayer2*;
        Appearance Frame("Big_Viewport_CRT","Fuel_Zoomed");
        Appearance Frame("Tank_Light_2_Test");
        Fuel_System=0;
        Fuel_Zoom=1;
        Engine_Zoom=0;
    }

    *Set_Current_LayerLayer1*;
    Drive_Light"1","Pot_Lights",1;
    Drive_Light"2","Pot_Lights_1",2);
pots=1613;
poty=1422;

Channel_Member_Put_Buffer"Session_Common","Location",MEMBER_DIMEN(1),10.1.CHAN_FLOAT,&potx,1.CHANNEL_WRITE;

Channel_Member_Put_Buffer"Session_Common","Location",MEMBER_DIMEN(1),11.1.CHAN_FLOAT,&poty,1.CHANNEL_WRITE;

Drive_Pot"","vertical",poty;
Drive_Pot"","horizontal",potx;

*Set_Current_LayerLayer2*;
Scale_Frame("Big_Viewport_CRT","Fuel_Zoomed",zoomx,zoomy);
Drive_Light"9","Tank_Zoom_Light",16;
}

-> Monitor
(SWITCH Fuel_Overview_Sw 'APU')
{
if (Balance==1) {
  Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
  Balance=0;
}
if (Fuel_System==1) {
  Hide_Frame("Big_Viewport_CRT","Fuel_system");
}
if (Engine_Zoom==1) {
  Hide_Frame("Big_Viewport_CRT","Engine_Zoomed");
  Drive_Switch("Engine_Selector1",14);
  Drive_Switch("Engine_Selector2",14);
  Drive_Switch("Engine_Selector3",14);
  Drive_Switch("Engine_Selector4",14);
}
if (Fuel_Zoom==0) {
  Hide_Frame("Energy_Env_Plots2");
  Hide_Frame("Energy_Labels");

  *Set_Current_Layer(Layer2);*
  Appear_Frame("Big_Viewport_CRT","Fuel_Zoomed");
  Appear_Frame("Tank_Light_2_Tes");
  Fuel_System=0;
  Fuel_Zoom=1;
  Engine_Zoom=0;
}

*Set_Current_Layer(Layer1)*;
Drive_Up("Pot_Lights",1);
Drive_Up("Pot_Lights_1",2);
posy=1040;
posx=1422;

Channel_Member_Put_Buffert("Session_Common","Location",MEMBER_DIMEN(11,10,1,CHAN_FLOAT&posy,1,CHAN_CHANNEL_WRITE);

Channel_Member_Put_Buffert("Session_Common","Location",MEMBER_DIMEN(11,11,1,CHAN_FLOAT&posy,1,CHAN_CHANNEL_WRITE);

  Drive_Put("vertical",posy);
  Drive_Put("horizontal",posx);

  *Set_Current_Layer(Layer2);*
  Scale_Frame("Big_Viewport_CRT","Fuel_Zoomed",zoom,zoom);
  Drive_Light("Tank_Zoom_Light",19);
}

} -> Monitor
(SWITCH Fuel_Overview_Sw 'E')
{
  if (Balance==1) {
    Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
    Balance=0;
  }
  if (Fuel_System==1) {
    Hide_Frame("Big_Viewport_CRT","Fuel_system");
  }
  if (Engine_Zoom==1) {
    Hide_Frame("Big_Viewport_CRT","Engine_Zoomed");
    Drive_Switch("Engine_Selector1",14);
    Drive_Switch("Engine_Selector2",14);
    Drive_Switch("Engine_Selector3",14);
    Drive_Switch("Engine_Selector4",14);
  }

  if (Fuel_Zoom==0) {
    Hide_Frame("Energy_Env_Plots2");
    Hide_Frame("Energy_Labels");
}
*Set_Current_Layer(Layer2)*;

  *Set_Current_Layer(Layer1)*;

  fuelsystem=0;
  fuelscale=1;
  enginewidth=0;

  *Set_Current_Layer(Layer2)*;

  Drive_Light("Pot_Lights",1);
  Drive_Light("Pot_Lights",2);
  pots=646;
  poty=1422;

  Channel_Member_Put_Buffet("Session_Common","Location",MEMBER_DIMEN(1),10,1,CHAN_FLOAT,&ots,1,CHANNEL_WRITE);

  Channel_Member_Put_Buffet("Session_Common","Location",MEMBER_DIMEN(1),11,1,CHAN_FLOAT,&ots,1,CHANNEL_WRITE);

  Drive_Pott("vertical",poty);
  Drive_Pott("horizontal",potx);

  *Set_Current_Layer(Layer2)*;

  Scale_Frame("Big_Viewport_CRT","Fuel_Zoomed",zoomx,zoomy);
  Drive_Light("Tank_Zoom_Light",17);

} > Monitor

(SWITCH Fuel_Overview_Sw 'E')

  { if (Balance==1) {
      Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
      Balance=0;
      } if (Fuel_System==1) { Hide_Frame("Big_Viewport_CRT","Fuel_system");
      } if (Engine_Zoom==1) { Hide_Frame("Big_Viewport_CRT","Engine_Zoomed");
      Drive_Switch("EngineSelector1",14);
      Drive_Switch("EngineSelector2",14);
      Drive_Switch("EngineSelector3",14);
      Drive_Switch("EngineSelector4",14);
      } if (Fuel_Zoom==0) { Hide_Frame("Energy_Ent_Plots2");
      Hide_Frame("Energy_Labels");

      *Set_Current_Layer(Layer2)*;

      Drive_Light("Fuel_Zoomed");
      Drive_Light("Tank_Light_2_Text");
      Fuel_System=0;
      Fuel_Zoom=1;
      Engine_Zoom=0;

      *Set_Current_Layer(Layer1)*;

      Drive_Light("Pot_Lights",1);
      Drive_Light("Pot_Lights",2);
      pots=209;
      poty=1422;

      Channel_Member_Put_Buffet("Session_Common","Location",MEMBER_DIMEN(1),10,1,CHAN_FLOAT,&ots,1,CHANNEL_WRITE);

      Channel_Member_Put_Buffet("Session_Common","Location",MEMBER_DIMEN(1),11,1,CHAN_FLOAT,&ots,1,CHANNEL_WRITE);

      Drive_Pott("vertical",poty);
Drive_Pot(".horizontal",potx);

*Set_Current_Layer(Layer2);
Scale_Frame("Big_Viewport_CRT","Fuel_Zoomed",zoomx,zoomy);
Drive_Light(".Tank_Zoom_Light",18);
}
} -> Monitor
(SWITCH Fuel_Overview_Sw 'LH Dump')
{
  if (Balance==1) {
    Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
    Balance=0;
  }
  if (Fuel_System==1) {
    Hide_Frame("Big_Viewport_CRT","Fuel_system");
  }
  if (Engine_Zoom==1) {
    Hide_Frame("Big_Viewport_CRT","Engine_Zoomed");
    Drive_Switch(".Engine_Select1",14);
    Drive_Switch(".Engine_Select2",14);
    Drive_Switch(".Engine_Select3",14);
    Drive_Switch(".Engine_Select4",14);
  }
  if (Fuel_Zoom==0) {
    Hide_Frame(".Energy_Ent_Plots2");
    Hide_Frame(".Energy_Labels");
    *Set_Current_Layer(Layer2);
    Appear_Frame("Big_Viewport_CRT","Fuel_Zoomed");
    Appear_Frame(".Tank_Light_2_Text");
    Fuel_System=0;
    Fuel_Zoom=1;
    Engine_Zoom=0;
  }

*Set_Current_Layer(Layer1);
Drive_Light(".Pot_Lights",1);
Drive_Light(".Pot_Lights_1",2);
potx=2057;
poty=49;

Channel_Member_Put_Buffer".Session_Common..:Session_COMMON..:Location",MEMBER_DIMEN(1),10.1.CHAN_FLOAT&potx,1.CHANNEL_WRITE E1;

Channel_Member_Put_Buffer".Session_Common..:Session_COMMON..:Location",MEMBER_DIMEN(1),11.1.CHAN_FLOAT&poty,1.CHANNEL_WRITE E1;

  Drive_Pot(".vertical",poty);
  Drive_Pot(".horizontal",potx);

*Set_Current_Layer(Layer2);
Scale_Frame("Big_Viewport_CRT","Fuel_Zoomed",zoomx,zoomy);
Drive_Light(".Tank_Zoom_Light",1);
}
} -> Monitor
(SWITCH Fuel_Overview_Sw 'All')
{
  Appear_Frame(".Energy_Ent_Plots2");
  Drive_Switch(".E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
  Drive_Switch(".E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
  Drive_Switch(".E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
  Drive_Switch(".E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);

  if (Balance==1) {
    Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
    Balance=0;
  }
  *Set_Current_Layer(Layer2);
  if (Fuel_System==0) {

Fuel_System=1;
Fuel_Zoom=0;
Engine_Zoom=0;
Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
Hide_Frame("Big_Viewport_CRT","Engine_Zoomed");
Drive_Switch("","Engine_Selector1",14);
Drive_Switch("","Engine_Selector2",14);
Drive_Switch("","Engine_Selector3",14);
Drive_Switch("","Engine_Selector4",14);
Hide_Frame("","Task_Light_1_Test");
Appear_Frame("Big_Viewport_CRT","Fuel_system");
*Set_Current_Layer1*;
Get_Frame("","Energy_Ent_Plots",0,FOREGROUND_FRAME);
* Drive_Light("","E1_Energy_ref.Energy_Reference.Energy_Ref_Light",14);
Drive_Light("","E2_Energy_ref.Energy_Reference.Energy_Ref_Light",14);
Drive_Light("","E3_Energy_ref.Energy_Reference.Energy_Ref_Light",14);
Drive_Light("","E4_Energy_ref.Energy_Reference.Energy_Ref_Light",14);
Drive_Light("","Pos_Lights",2);
Drive_Light("","Pos_Lights",2);
Appear_Frame("","Energy_Labels");
Drive_Light("","Labels",1);
} -> Monitor
(BUTTON_GL.Valve_Ref_Vertical.Valve_State 'Horizontal')
{
Drive_Button("Big_Viewport_CRT","Fuel_system.GL.Valve_Ref_Vertical.Valve_State",2);
Drive_Button("Big_Viewport_CRT","Fuel_Zoomed.GL.Valve_Ref_Vertical.Valve_State",2);
} -> Monitor
(BUTTON_GL.Valve_Ref_Vertical.Valve_State 'Vertical')
{
Drive_Button("Big_Viewport_CRT","Fuel_system.GL.Valve_Ref_Vertical.Valve_State",1);
Drive_Button("Big_Viewport_CRT","Fuel_Zoomed.GL.Valve_Ref_Vertical.Valve_State",1);
} -> Monitor
(BUTTON_GL.Valve_Ref_Vertical.Valve_State 'Horizontal')
{
Drive_Button("Big_Viewport_CRT","Fuel_system.I.Valve_Ref_Vertical.Valve_State",2);
Drive_Button("Big_Viewport_CRT","Fuel_Zoomed.I.Valve_Ref_Vertical.Valve_State",2);
} -> Monitor
(BUTTON_GL.Valve_Ref_Vertical.Valve_State 'Vertical')
{
Drive_Button("Big_Viewport_CRT","Fuel_system.I.Valve_Ref_Vertical.Valve_State",1);
Drive_Button("Big_Viewport_CRT","Fuel_Zoomed.I.Valve_Ref_Vertical.Valve_State",1);
} -> Monitor
(BUTTON_GL.Valve_Ref_Vertical.Valve_State 'Horizontal')
{
Drive_Button("Big_Viewport_CRT","Fuel_system.HL.Valve_Ref_Vertical.Valve_State",2);
Drive_Button("Big_Viewport_CRT","Fuel_Zoomed.HL.Valve_Ref_Vertical.Valve_State",2);
} -> Monitor
(BUTTON_GL.Valve_Ref_Vertical.Valve_State 'Vertical')
{
Drive_Button("Big_Viewport_CRT","Fuel_system.HL.Valve_Ref_Vertical.Valve_State",1);
Drive_Button("Big_Viewport_CRT","Fuel_Zoomed.HL.Valve_Ref_Vertical.Valve_State",1);
} -> Monitor
(BUTTON_GL.Valve_Ref_Vertical.Valve_State 'Horizontal')
{
Drive_Button("Big_Viewport_CRT","Fuel_system.HR.Valve_Ref_Vertical.Valve_State",2);
Drive_Button("Big_Viewport_CRT","Fuel_Zoomed.HR.Valve_Ref_Vertical.Valve_State",2);
} -> Monitor
(BUTTON_GL.Valve_Ref_Vertical.Valve_State 'Vertical')
{
Drive_Button("Big_Viewport_CRT","Fuel_system.HR.Valve_Ref_Vertical.Valve_State",1);
Drive_Button("Big_Viewport_CRT","Fuel_Zoomed.HR.Valve_Ref_Vertical.Valve_State",1);
Monitor

(BUTTON GR.Valve_Ref_Vertical.Valve_State 'Horizontal')
{
  Drive_Button("Big_Viewport_CRT","Fuel_system,GR.Valve_Ref_Vertical.Valve_State",2);
  Drive_Button("Big_Viewport_CRT","Fuel_Zoomed,GR.Valve_Ref_Vertical.Valve_State",2);
}

(BUTTON GR.Valve_Ref_Vertical.Valve_State 'Vertical')
{
  Drive_Button("Big_Viewport_CRT","Fuel_system,GR.Valve_Ref_Vertical.Valve_State",1);
  Drive_Button("Big_Viewport_CRT","Fuel_Zoomed,GR.Valve_Ref_Vertical.Valve_State",1);
}

(BUTTON WR.Valve_Ref_Vertical.Valve_State 'Horizontal')
{
  Drive_Button("Big_Viewport_CRT","Fuel_system,WR.Valve_Ref_Vertical.Valve_State",2);
  Drive_Button("Big_Viewport_CRT","Fuel_Zoomed,WR.Valve_Ref_Vertical.Valve_State",2);
}

(BUTTON WR.Valve_Ref_Vertical.Valve_State 'Vertical')
{
  Drive_Button("Big_Viewport_CRT","Fuel_system,WR.Valve_Ref_Vertical.Valve_State",1);
  Drive_Button("Big_Viewport_CRT","Fuel_Zoomed,WR.Valve_Ref_Vertical.Valve_State",1);
}

(BUTTON WL.Valve_Ref_Vertical.Valve_State 'Horizontal')
{
  Drive_Button("Big_Viewport_CRT","Fuel_system,WL.Valve_Ref_Vertical.Valve_State",2);
  Drive_Button("Big_Viewport_CRT","Fuel_Zoomed,WL.Valve_Ref_Vertical.Valve_State",2);
}

(BUTTON WL.Valve_Ref_Vertical.Valve_State 'Vertical')
{
  Drive_Button("Big_Viewport_CRT","Fuel_system,WL.Valve_Ref_Vertical.Valve_State",1);
  Drive_Button("Big_Viewport_CRT","Fuel_Zoomed,WL.Valve_Ref_Vertical.Valve_State",1);
}

(BUTTON LR.Valve_Ref_Vertical.Valve_State 'Horizontal')
{
  Drive_Button("Big_Viewport_CRT","Fuel_system,LR.Valve_Ref_Vertical.Valve_State",2);
  Drive_Button("Big_Viewport_CRT","Fuel_Zoomed,LR.Valve_Ref_Vertical.Valve_State",2);
}

(BUTTON LR.Valve_Ref_Vertical.Valve_State 'Vertical')
{
  Drive_Button("Big_Viewport_CRT","Fuel_system,LR.Valve_Ref_Vertical.Valve_State",1);
  Drive_Button("Big_Viewport_CRT","Fuel_Zoomed,LR.Valve_Ref_Vertical.Valve_State",1);
}

(BUTTON MR.Valve_Ref_Vertical.Valve_State 'Horizontal')
{
  Drive_Button("Big_Viewport_CRT","Fuel_system,MR.Valve_Ref_Vertical.Valve_State",2);
  Drive_Button("Big_Viewport_CRT","Fuel_Zoomed,MR.Valve_Ref_Vertical.Valve_State",2);
}

(BUTTON MR.Valve_Ref_Vertical.Valve_State 'Vertical')
{
  Drive_Button("Big_Viewport_CRT","Fuel_system,MR.Valve_Ref_Vertical.Valve_State",1);
  Drive_Button("Big_Viewport_CRT","Fuel_Zoomed,MR.Valve_Ref_Vertical.Valve_State",1);
}

(BUTTON TR.Valve_Ref_Vertical.Valve_State 'Horizontal')
{
  Drive_Button("Big_Viewport_CRT","Fuel_system,TR.Valve_Ref_Vertical.Valve_State",2);
  Drive_Button("Big_Viewport_CRT","Fuel_Zoomed,TR.Valve_Ref_Vertical.Valve_State",2);
}

(BUTTON TR.Valve_Ref_Vertical.Valve_State 'Vertical')
{
  Drive_Button("Big_Viewport_CRT","Fuel_system,TR.Valve_Ref_Vertical.Valve_State",1);
  Drive_Button("Big_Viewport_CRT","Fuel_Zoomed,TR.Valve_Ref_Vertical.Valve_State",1);
}

(BUTTON KL.Valve_Ref_Vertical.Valve_State 'Horizontal')
{

Drive_Button("Big_Viewport_CRT","Fuel_system.KL.Valve_Ref_Vertical.Valve_State",1);
Drive_Button("Big_Viewport_CRT","Fuel_Zoomed.KL.Valve_Ref_Vertical.Valve_State",1);

} -> Monitor

(BUTTON KL.Valve_Ref_Vertical.Valve_State 'Vertical')
{
  Drive_Button("Big_Viewport_CRT","Fuel_system.KL.Valve_Ref_Vertical.Valve_State",2);
  Drive_Button("Big_Viewport_CRT","Fuel_Zoomed.KL.Valve_Ref_Vertical.Valve_State",2);
}

} -> Monitor

(BUTTON LL.Valve_Ref_Vertical.Valve_State 'Vertical')
{
  Drive_Button("Big_Viewport_CRT","Fuel_system.LL.Valve_Ref_Vertical.Valve_State",1);
  Drive_Button("Big_Viewport_CRT","Fuel_Zoomed.LL.Valve_Ref_Vertical.Valve_State",1);
}

} -> Monitor

(BUTTON LL.Valve_Ref_Vertical.Valve_State 'Horizontal')
{
  Drive_Button("Big_Viewport_CRT","Fuel_system.LL.Valve_Ref_Vertical.Valve_State",2);
  Drive_Button("Big_Viewport_CRT","Fuel_Zoomed.LL.Valve_Ref_Vertical.Valve_State",2);
}

} -> Monitor

(BUTTON XL.Valve_Ref_Horizontal.Valve_State 'Vertical')
{
  Drive_Button("Big_Viewport_CRT","Fuel_system.XL.Valve_Ref_Vertical.Valve_State",2);
  Drive_Button("Big_Viewport_CRT","Fuel_Zoomed.XL.Valve_Ref_Vertical.Valve_State",2);
}

} -> Monitor

(BUTTON XL.Valve_Ref_Horizontal.Valve_State 'Horizontal')
{
  Drive_Button("Big_Viewport_CRT","Fuel_system.XL.Valve_Ref_Vertical.Valve_State",1);
  Drive_Button("Big_Viewport_CRT","Fuel_Zoomed.XL.Valve_Ref_Vertical.Valve_State",1);
}

} -> Monitor

(BUTTON O.Valve_Ref_Vertical.Valve_State 'Horizontal')
{
  Drive_Button("Big_Viewport_CRT","Fuel_system.O.Valve_Ref_Vertical.Valve_State",1);
  Drive_Button("Big_Viewport_CRT","Fuel_Zoomed.O.Valve_Ref_Vertical.Valve_State",1);
}

} -> Monitor

(BUTTON O.Valve_Ref_Vertical.Valve_State 'Vertical')
{
  Drive_Button("Big_Viewport_CRT","Fuel_system.O.Valve_Ref_Vertical.Valve_State",2);
  Drive_Button("Big_Viewport_CRT","Fuel_Zoomed.O.Valve_Ref_Vertical.Valve_State",2);
}

} -> Monitor

(BUTTON P.Valve_Ref_Vertical.Valve_State 'Horizontal')
{
  Drive_Button("Big_Viewport_CRT","Fuel_system.P.Valve_Ref_Vertical.Valve_State",1);
  Drive_Button("Big_Viewport_CRT","Fuel_Zoomed.P.Valve_Ref_Vertical.Valve_State",1);
}

} -> Monitor

(BUTTON P.Valve_Ref_Vertical.Valve_State 'Vertical')
{
  Drive_Button("Big_Viewport_CRT","Fuel_system.P.Valve_Ref_Vertical.Valve_State",2);
  Drive_Button("Big_Viewport_CRT","Fuel_Zoomed.P.Valve_Ref_Vertical.Valve_State",2);
}

} -> Monitor

(BUTTON XR.Valve_Ref_Horizontal.Valve_State 'Vertical')
{
  Drive_Button("Big_Viewport_CRT","Fuel_system.XR.Valve_Ref_Vertical.Valve_State",2);
  Drive_Button("Big_Viewport_CRT","Fuel_Zoomed.XR.Valve_Ref_Vertical.Valve_State",2);
}

} -> Monitor

(BUTTON XR.Valve_Ref_Horizontal.Valve_State 'Horizontal')
{
  Drive_Button("Big_Viewport_CRT","Fuel_system.XR.Valve_Ref_Vertical.Valve_State",1);
  Drive_Button("Big_Viewport_CRT","Fuel_Zoomed.XR.Valve_Ref_Vertical.Valve_State",1);
}

} -> Monitor

(BUTTON KR.Valve_Ref_Vertical.Valve_State 'Horizontal')
{
  Drive_Button("Big_Viewport_CRT","Fuel_system.KR.Valve_Ref_Vertical.Valve_State",1);
}
Monitor

{BUTTON Big_Viewport_CRT.Fuel_system.HL.Valve_Ref_Vertical.Valve_State 'Horizontal')
  {Drive_Button"Big_Viewport_CRT"."Fuel_Zoomed.HL.Valve_Ref_Vertical.Valve_State".2);
    Drive_Button".".HL.Valve_Ref_Vertical.Valve_State".2);
} -> Monitor

{BUTTON Big_Viewport_CRT.Fuel_Zoomed.HR.Valve_Ref_Vertical.Valve_State 'Horizontal')
  {Drive_Button"Big_Viewport_CRT"."Fuel_system.HR.Valve_Ref_Vertical.Valve_State".2);
    Drive_Button".".HR.Valve_Ref_Vertical.Valve_State".2);
} -> Monitor

{BUTTON Big_Viewport_CRT.Fuel_system.HR.Valve_Ref_Vertical.Valve_State 'Vertical')
  {Drive_Button"Big_Viewport_CRT"."Fuel_Zoomed.HR.Valve_Ref_Vertical.Valve_State".1);
    Drive_Button".".HR.Valve_Ref_Vertical.Valve_State".1);
} -> Monitor

{BUTTON Big_Viewport_CRT.Fuel_Zoomed.HR.Valve_Ref_Vertical.Valve_State 'Vertical')
  {Drive_Button"Big_Viewport_CRT"."Fuel_system.HR.Valve_Ref_Vertical.Valve_State".1);
    Drive_Button".".HR.Valve_Ref_Vertical.Valve_State".1);
} -> Monitor

{BUTTON Big_Viewport_CRT.Fuel_system.HR.Valve_Ref_Vertical.Valve_State 'Horizontal')
  {Drive_Button"Big_Viewport_CRT"."Fuel_Zoomed.HR.Valve_Ref_Vertical.Valve_State".2);
    Drive_Button".".HR.Valve_Ref_Vertical.Valve_State".2);
} -> Monitor

{BUTTON Big_Viewport_CRT.Fuel_Zoomed.GR.Valve_Ref_Vertical.Valve_State 'Horizontal')
  {Drive_Button"Big_Viewport_CRT"."Fuel_system.GR.Valve_Ref_Vertical.Valve_State".2);
    Drive_Button".".GR.Valve_Ref_Vertical.Valve_State".2);
} -> Monitor

{BUTTON Big_Viewport_CRT.Fuel_system.GR.Valve_Ref_Vertical.Valve_State 'Vertical')
  {Drive_Button"Big_Viewport_CRT"."Fuel_Zoomed.GR.Valve_Ref_Vertical.Valve_State".1);
    Drive_Button".".GR.Valve_Ref_Vertical.Valve_State".1);
} -> Monitor

{BUTTON Big_Viewport_CRT.Fuel_Zoomed.GR.Valve_Ref_Vertical.Valve_State 'Vertical')
  {Drive_Button"Big_Viewport_CRT"."Fuel_system.GR.Valve_Ref_Vertical.Valve_State".1);
    Drive_Button".".GR.Valve_Ref_Vertical.Valve_State".1);
} -> Monitor

{BUTTON Big_Viewport_CRT.Fuel_system.GR.Valve_Ref_Vertical.Valve_State 'Horizontal')
  {Drive_Button"Big_Viewport_CRT"."Fuel_Zoomed.GR.Valve_Ref_Vertical.Valve_State".2);
    Drive_Button".".GR.Valve_Ref_Vertical.Valve_State".2);
} -> Monitor

{BUTTON Big_Viewport_CRT.Fuel_Zoomed.WR.Valve_Ref_Vertical.Valve_State 'Horizontal')
  {Drive_Button"Big_Viewport_CRT"."Fuel_system.WR.Valve_Ref_Vertical.Valve_State".2);
    Drive_Button".".WR.Valve_Ref_Vertical.Valve_State".2);
} -> Monitor
(BUTTON Big_Viewport_CRT.Fuel_system.WR.Valve_Ref_Verical.Valve_State 'Vertical')
{
  Drive_Button("Big_Viewport_CRT","Fuel_Zoomed.WR.Valve_Ref_Verical.Valve_State",1);
  Drive_Button("","WR.Valve_Ref_Verical.Valve_State",1);
}

} // Monitor

(BUTTON Big_Viewport_CRT.Fuel_Zoomed.WR.Valve_Ref_Verical.Valve_State 'Vertical')
{
  Drive_Button("Big_Viewport_CRT","Fuel_Zoomed.WR.Valve_Ref_Verical.Valve_State",1);
  Drive_Button("","WR.Valve_Ref_Verical.Valve_State",1);
}

} // Monitor

(BUTTON Big_Viewport_CRT.Fuel_system.WR.Valve_Ref_Verical.Valve_State 'Horizontal')
{
  Drive_Button("Big_Viewport_CRT","Fuel_system.WR.Valve_Ref_Verical.Valve_State",2);
  Drive_Button("","WR.Valve_Ref_Verical.Valve_State",2);
}

} // Monitor

(BUTTON Big_Viewport_CRT.Fuel_Zoomed.LL.Valve_Ref_Verical.Valve_State 'Horizontal')
{
  Drive_Button("Big_Viewport_CRT","Fuel_Zoomed.LL.Valve_Ref_Verical.Valve_State",2);
  Drive_Button("","LL.Valve_Ref_Verical.Valve_State",2);
}

} // Monitor

(BUTTON Big_Viewport_CRT.Fuel_system.LL.Valve_Ref_Verical.Valve_State 'Vertical')
{
  Drive_Button("Big_Viewport_CRT","Fuel_Zoomed.LL.Valve_Ref_Verical.Valve_State",1);
  Drive_Button("","LL.Valve_Ref_Verical.Valve_State",1);
}

} // Monitor

(BUTTON Big_Viewport_CRT.Fuel_Zoomed.LL.Valve_Ref_Verical.Valve_State 'Vertical')
{
  Drive_Button("Big_Viewport_CRT","Fuel_system.LL.Valve_Ref_Verical.Valve_State",1);
  Drive_Button("","LL.Valve_Ref_Verical.Valve_State",1);
}

} // Monitor

(BUTTON Big_Viewport_CRT.Fuel_system.LL.Valve_Ref_Verical.Valve_State 'Horizontal')
{
  Drive_Button("Big_Viewport_CRT","Fuel_Zoomed.LL.Valve_Ref_Verical.Valve_State",2);
  Drive_Button("","LL.Valve_Ref_Verical.Valve_State",2);
}

} // Monitor

(BUTTON Big_Viewport_CRT.Fuel_Zoomed.L.L.Valve_Ref_Verical.Valve_State 'Horizontal')
{
  Drive_Button("Big_Viewport_CRT","Fuel_system.L.Valve_Ref_Verical.Valve_State",2);
  Drive_Button("","L.Valve_Ref_Verical.Valve_State",2);
}

} // Monitor

(BUTTON Big_Viewport_CRT.Fuel_system.L.Valve_Ref_Verical.Valve_State 'Vertical')
{
  Drive_Button("Big_Viewport_CRT","Fuel_Zoomed.L.Valve_Ref_Verical.Valve_State",1);
  Drive_Button("","L.Valve_Ref_Verical.Valve_State",1);
}

} // Monitor

(BUTTON Big_Viewport_CRT.Fuel_Zoomed.L.Valve_Ref_Verical.Valve_State 'Vertical')
{
  Drive_Button("Big_Viewport_CRT","Fuel_system.L.Valve_Ref_Verical.Valve_State",1);
  Drive_Button("","L.Valve_Ref_Verical.Valve_State",1);
}

} // Monitor

(BUTTON Big_Viewport_CRT.Fuel_Zoomed.L.Valve_Ref_Verical.Valve_State 'Vertical')
{
  Drive_Button("Big_Viewport_CRT","Fuel_system.L.Valve_Ref_Verical.Valve_State",1);
  Drive_Button("","L.Valve_Ref_Verical.Valve_State",1);
}

} // Monitor
{ Drive_Button:"Big_Viewport_CRT","Fuel_Zoomed.ML.Valve_Ref_Vertical.Valve_State",1; };
Drive_Button:"ML.Valve_Ref_Vertical.Valve_State",1;

} -> Monitor
{ BUTTON Big_Viewport_CRT,Fuel_Zoomed.ML.Valve_Ref_Vertical.Valve_State 'Vertical'
  
  Drive_Button:"Big_Viewport_CRT","Fuel_system.ML.Valve_Ref_Vertical.Valve_State",1;
  
  Drive_Button:"ML.Valve_Ref_Vertical.Valve_State",1;

} -> Monitor
{ BUTTON Big_Viewport_CRT,Fuel_system.ML.Valve_Ref_Vertical.Valve_State 'Horizontal'
  
  Drive_Button:"Big_Viewport_CRT","Fuel_Zoomed.ML.Valve_Ref_Vertical.Valve_State",2;
  
  Drive_Button:"ML.Valve_Ref_Vertical.Valve_State",2;

} -> Monitor
{ BUTTON Big_Viewport_CRT,Fuel_Zoomed.XL.Valve_Ref_Vertical.Valve_State 'Horizontal'
  
  Drive_Button:"Big_Viewport_CRT","Fuel_system.XL.Valve_Ref_Vertical.Valve_State",1;
  
  Drive_Button:"XL.Valve_Ref_Horizontal.Valve_State",2;

} -> Monitor
{ BUTTON Big_Viewport_CRT,Fuel_system.XL.Valve_Ref_Vertical.Valve_State 'Vertical'
  
  Drive_Button:"Big_Viewport_CRT","Fuel_Zoomed.XL.Valve_Ref_Vertical.Valve_State",2;
  
  Drive_Button:"XL.Valve_Ref_Horizontal.Valve_State",1;

} -> Monitor
{ BUTTON Big_Viewport_CRT,Fuel_Zoomed.XL.Valve_Ref_Vertical.Valve_State 'Vertical'
  
  Drive_Button:"Big_Viewport_CRT","Fuel_system.XL.Valve_Ref_Vertical.Valve_State",2;
  
  Drive_Button:"XL.Valve_Ref_Horizontal.Valve_State",1;

} -> Monitor
{ BUTTON Big_Viewport_CRT,Fuel_system.XL.Valve_Ref_Vertical.Valve_State 'Horizontal'
  
  Drive_Button:"Big_Viewport_CRT","Fuel_Zoomed.XL.Valve_Ref_Vertical.Valve_State",1;
  
  Drive_Button:"XL.Valve_Ref_Horizontal.Valve_State",2;

} -> Monitor
{ BUTTON Big_Viewport_CRT,Fuel_Zoomed.XR.Valve_Ref_Vertical.Valve_State 'Horizontal'
  
  Drive_Button:"Big_Viewport_CRT","Fuel_system.XR.Valve_Ref_Vertical.Valve_State",2;
  
  Drive_Button:"XR.Valve_Ref_Vertical.Valve_State",2;

} -> Monitor
{ BUTTON Big_Viewport_CRT,Fuel_system.XR.Valve_Ref_Vertical.Valve_State 'Vertical'
  
  Drive_Button:"Big_Viewport_CRT","Fuel_Zoomed.XR.Valve_Ref_Vertical.Valve_State",1;
  
  Drive_Button:"XR.Valve_Ref_Vertical.Valve_State",1;

} -> Monitor
{ BUTTON Big_Viewport_CRT,Fuel_Zoomed.XR.Valve_Ref_Vertical.Valve_State 'Vertical'
  
  Drive_Button:"Big_Viewport_CRT","Fuel_system.XR.Valve_Ref_Vertical.Valve_State",1;
  
  Drive_Button:"XR.Valve_Ref_Vertical.Valve_State",1;

} -> Monitor
{ BUTTON Big_Viewport_CRT,Fuel_Zoomed.LR.Valve_Ref_Vertical.Valve_State 'Vertical'
  
  Drive_Button:"Big_Viewport_CRT","Fuel_system.LR.Valve_Ref_Vertical.Valve_State",2;
  
  Drive_Button:"LR.Valve_Ref_Vertical.Valve_State",2;

} -> Monitor
{ BUTTON Big_Viewport_CRT,Fuel_system.LR.Valve_Ref_Vertical.Valve_State 'Horizontal'
  
  Drive_Button:"Big_Viewport_CRT","Fuel_Zoomed.LR.Valve_Ref_Vertical.Valve_State",1;
  
  Drive_Button:"LR.Valve_Ref_Vertical.Valve_State",1;

} -> Monitor
{ BUTTON Big_Viewport_CRT,Fuel_Zoomed.LR.Valve_Ref_Vertical.Valve_State 'Vertical'
  
  Drive_Button:"Big_Viewport_CRT","Fuel_system.LR.Valve_Ref_Vertical.Valve_State",1;
  
  Drive_Button:"LR.Valve_Ref_Vertical.Valve_State",1;

} -> Monitor
{ BUTTON Big_Viewport_CRT,Fuel_system.LR.Valve_Ref_Vertical.Valve_State 'Horizontal'
  
  Drive_Button:"Big_Viewport_CRT","Fuel_Zoomed.LR.Valve_Ref_Vertical.Valve_State",2;

}
Monitor

Drive_Button("LR.Valve_Ref_Veritical.Valve_State",2);

-) -> Monitor
(BUTTON Big_Viewport_CRT,Fuel_Zoomed.XR.Valve_Ref_Veritical.Valve_State 'Vertical')
{
    Drive_Button("Big_Viewport_CRT","Fuel_system.XR.Valve_Ref_Veritical.Valve_State",1);
    Drive_Button("XR.Valve_Ref_Horizontal.Valve_State",1);
}

-) -> Monitor
(BUTTON Big_Viewport_CRT,Fuel_system.XR.Valve_Ref_Veritical.Valve_State 'Horizontal')
{
    Drive_Button("Big_Viewport_CRT","Fuel_Zoomed.XR.Valve_Ref_Veritical.Valve_State",2);
    Drive_Button("XR.Valve_Ref_Horizontal.Valve_State",2);
}

-) -> Monitor
(BUTTON Big_Viewport_CRT,Fuel_Zoomed.XR.Valve_Ref_Veritical.Valve_State 'Horizontal')
{
    Drive_Button("Big_Viewport_CRT","Fuel_system.XR.Valve_Ref_Veritical.Valve_State",2);
    Drive_Button("XR.Valve_Ref_Horizontal.Valve_State",2);
}

-) -> Monitor
(BUTTON Big_Viewport_CRT,Fuel_system.XR.Valve_Ref_Veritical.Valve_State 'Vertical')
{
    Drive_Button("Big_Viewport_CRT","Fuel_Zoomed.XR.Valve_Ref_Veritical.Valve_State",1);
    Drive_Button("XR.Valve_Ref_Horizontal.Valve_State",1);
}

-) -> Monitor
(BUTTON Big_Viewport_CRT,Fuel_Zoomed.TL.Valve_Ref_Veritical.Valve_State 'Horizontal')
{
    Drive_Button("Big_Viewport_CRT","Fuel_system.TL.Valve_Ref_Veritical.Valve_State",1);
    Drive_Button("TL.Valve_Ref_Veritical.Valve_State",1);
}

-) -> Monitor
(BUTTON Big_Viewport_CRT,Fuel_system.TL.Valve_Ref_Veritical.Valve_State 'Vertical')
{
    Drive_Button("Big_Viewport_CRT","Fuel_Zoomed.TL.Valve_Ref_Veritical.Valve_State",2);
    Drive_Button("TL.Valve_Ref_Veritical.Valve_State",2);
}

-) -> Monitor
(BUTTON Big_Viewport_CRT,Fuel_Zoomed.TL.Valve_Ref_Veritical.Valve_State 'Vertical')
{
    Drive_Button("Big_Viewport_CRT","Fuel_system.TL.Valve_Ref_Veritical.Valve_State",2);
    Drive_Button("TL.Valve_Ref_Veritical.Valve_State",2);
}

-) -> Monitor
(BUTTON Big_Viewport_CRT,Fuel_system.TL.Valve_Ref_Veritical.Valve_State 'Horizontal')
{
    Drive_Button("Big_Viewport_CRT","Fuel_Zoomed.TL.Valve_Ref_Veritical.Valve_State",1);
    Drive_Button("TL.Valve_Ref_Veritical.Valve_State",1);
}

-) -> Monitor
(BUTTON Big_Viewport_CRT,Fuel_Zoomed.KL.Valve_Ref_Veritical.Valve_State 'Horizontal')
{
    Drive_Button("Big_Viewport_CRT","Fuel_system.KL.Valve_Ref_Veritical.Valve_State",1);
    Drive_Button("KL.Valve_Ref_Veritical.Valve_State",1);
}

-) -> Monitor
(BUTTON Big_Viewport_CRT,Fuel_system.KL.Valve_Ref_Veritical.Valve_State 'Vertical')
{
    Drive_Button("Big_Viewport_CRT","Fuel_Zoomed.KL.Valve_Ref_Veritical.Valve_State",2);


Drive_Button("";KL.Valve_Ref_Vertical.Valve_State";2);

} -> Monitor
(BUTTON Big_Viewport_CRT.Fuel_Zoomed.KL.Valve_Ref_Vertical.Valve_State 'Vertical')
{ Drive_Button("Big_Viewport_CRT";'Fuel_system.KL.Valve_Ref_Vertical.Valve_State";2);

Drive_Button("";KL.Valve_Ref_Vertical.Valve_State";2);

} -> Monitor
(BUTTON Big_Viewport_CRT.Fuel_system.KL.Valve_Ref_Vertical.Valve_State 'Horizontal')
{ Drive_Button("Big_Viewport_CRT";'Fuel_Zoomed.KL.Valve_Ref_Vertical.Valve_State";1);

Drive_Button("";KL.Valve_Ref_Vertical.Valve_State";1);

} -> Monitor
(BUTTON Big_Viewport_CRT.Fuel_Zoomed.O.Valve_Ref_Vertical.Valve_State 'Horizontal')
{ Drive_Button("Big_Viewport_CRT";'Fuel_system.O.Valve_Ref_Vertical.Valve_State";1);

Drive_Button("";O.Valve_Ref_Vertical.Valve_State";1);

} -> Monitor
(BUTTON Big_Viewport_CRT.Fuel_system.O.Valve_Ref_Vertical.Valve_State 'Vertical')
{ Drive_Button("Big_Viewport_CRT";'Fuel_Zoomed.O.Valve_Ref_Vertical.Valve_State";2);

Drive_Button("";O.Valve_Ref_Vertical.Valve_State";2);

} -> Monitor
(BUTTON Big_Viewport_CRT.Fuel_Zoomed.O.Valve_Ref_Vertical.Valve_State 'Vertical')
{ Drive_Button("Big_Viewport_CRT";'Fuel_system.O.Valve_Ref_Vertical.Valve_State";2);

Drive_Button("";O.Valve_Ref_Vertical.Valve_State";2);

} -> Monitor
(BUTTON Big_Viewport_CRT.Fuel_system.O.Valve_Ref_Vertical.Valve_State 'Horizontal')
{ Drive_Button("Big_Viewport_CRT";'Fuel_Zoomed.O.Valve_Ref_Vertical.Valve_State";1);

Drive_Button("";O.Valve_Ref_Vertical.Valve_State";1);

} -> Monitor
(BUTTON Big_Viewport_CRT.Fuel_Zoomed.P.Valve_Ref_Vertical.Valve_State 'Horizontal')
{ Drive_Button("Big_Viewport_CRT";'Fuel_system.P.Valve_Ref_Vertical.Valve_State";1);

Drive_Button("";P.Valve_Ref_Vertical.Valve_State";1);

} -> Monitor
(BUTTON Big_Viewport_CRT.Fuel_system.P.Valve_Ref_Vertical.Valve_State 'Vertical')
{ Drive_Button("Big_Viewport_CRT";'Fuel_Zoomed.P.Valve_Ref_Vertical.Valve_State";2);

Drive_Button("";P.Valve_Ref_Vertical.Valve_State";2);

} -> Monitor
(BUTTON Big_Viewport_CRT.Fuel_Zoomed.P.Valve_Ref_Vertical.Valve_State 'Vertical')
{ Drive_Button("Big_Viewport_CRT";'Fuel_system.P.Valve_Ref_Vertical.Valve_State";2);

Drive_Button("";P.Valve_Ref_Vertical.Valve_State";2);

} -> Monitor
(BUTTON Big_Viewport_CRT.Fuel_system.P.Valve_Ref_Vertical.Valve_State 'Horizontal')
{ Drive_Button("Big_Viewport_CRT";'Fuel_Zoomed.P.Valve_Ref_Vertical.Valve_State";1);
Drive_Bottom("", "P.Valve_Ref_Vertical.Valve_State", 1);

} -> Monitor
(BUTTON Big_Viewport_CRT.Fuel_Zoomed.Y.Valve_Ref_Vertical.Valve_State 'Horizontal')
{
  Drive_Bottom("Big_Viewport_CRT", "Fuel_system.Y.Valve_Ref_Vertical.Valve_State", 1);
  Drive_Bottom("", "Y.Valve_Ref_Vertical.Valve_State", 1);

} -> Monitor
(BUTTON Big_Viewport_CRT.Fuel_system.Y.Valve_Ref_Vertical.Valve_State 'Vertical')
{
  Drive_Bottom("Big_Viewport_CRT", "Fuel_Zoomed.Y.Valve_Ref_Vertical.Valve_State", 2);
  Drive_Bottom("", "Y.Valve_Ref_Vertical.Valve_State", 2);

} -> Monitor
(BUTTON Big_Viewport_CRT.Fuel_system.Y.Valve_Ref_Vertical.Valve_State 'Horizontal')
{
  Drive_Bottom("Big_Viewport_CRT", "Fuel_Zoomed.Y.Valve_Ref_Vertical.Valve_State", 1);
  Drive_Bottom("", "Y.Valve_Ref_Vertical.Valve_State", 1);

} -> Monitor
(BUTTON Big_Viewport_CRT.Fuel_Zoomed.TR.Valve_Ref_Vertical.Valve_State 'Horizontal')
{
  Drive_Bottom("Big_Viewport_CRT", "Fuel_system.TR.Valve_Ref_Vertical.Valve_State", 1);
  Drive_Bottom("", "TR.Valve_Ref_Vertical.Valve_State", 1);

} -> Monitor
(BUTTON Big_Viewport_CRT.Fuel_system.TR.Valve_Ref_Vertical.Valve_State 'Vertical')
{
  Drive_Bottom("Big_Viewport_CRT", "Fuel_Zoomed.TR.Valve_Ref_Vertical.Valve_State", 1);
  Drive_Bottom("", "TR.Valve_Ref_Vertical.Valve_State", 1);

} -> Monitor
(BUTTON Big_Viewport_CRT.Fuel_Zoomed.TR.Valve_Ref_Vertical.Valve_State 'Vertical')
{
  Drive_Bottom("Big_Viewport_CRT", "Fuel_system.TR.Valve_Ref_Vertical.Valve_State", 1);
  Drive_Bottom("", "TR.Valve_Ref_Vertical.Valve_State", 1);

} -> Monitor
(BUTTON Big_Viewport_CRT.Fuel_system.TR.Valve_Ref_Vertical.Valve_State 'Horizontal')
{
  Drive_Bottom("Big_Viewport_CRT", "Fuel_Zoomed.TR.Valve_Ref_Vertical.Valve_State", 1);
  Drive_Bottom("", "TR.Valve_Ref_Vertical.Valve_State", 1);

} -> Monitor
(BUTTON Big_Viewport_CRT.Fuel_system.KR.Valve_Ref_Vertical.Valve_State 'Horizontal')
{
  Drive_Bottom("Big_Viewport_CRT", "Fuel_system.KR.Valve_Ref_Vertical.Valve_State", 1);
  Drive_Bottom("", "KR.Valve_Ref_Vertical.Valve_State", 1);

} -> Monitor
(BUTTON Big_Viewport_CRT.Fuel_system.KR.Valve_Ref_Vertical.Valve_State 'Vertical')
{
  Drive_Bottom("Big_Viewport_CRT", "Fuel_Zoomed.KR.Valve_Ref_Vertical.Valve_State", 1);
  Drive_Bottom("", "KR.Valve_Ref_Vertical.Valve_State", 1);

} -> Monitor
(BUTTON Big_Viewport_CRT.Fuel_Zoomed.KR. Valve_Ref_Vertical.Valve_State 'Vertical')
{
    Drive_Button("Big_Viewport_CRT"."Fuel_system.KR.Valve_Ref_Vertical.Valve_State",2);
    Drive_Button("KR.Valve_Ref_Vertical.Valve_State",2);
}

} -> Monitor
(BUTTON Big_Viewport_CRT.Fuel_system.KR.Valve_Ref_Vertical.Valve_State 'Horizontal')
{
    Drive_Button("Big_Viewport_CRT"."Fuel_Zoomed.KR.Valve_Ref_Vertical.Valve_State",1);
    Drive_Button("KR.Valve_Ref_Vertical.Valve_State",1);
}

} -> Monitor
(SWITCH Engine_Selector1 'code1')
{
    Appear_Frame(""."Energy_Ent_Plots2"";
    Drive_Switch(""."Fuel_Overview_Sw",1);
    Drive_Switch("".""."Engine_Selector2",14);
    Drive_Switch("".""."Engine_Selector3",14);
    Drive_Switch("".""."Engine_Selector4",14);
    Drive_Light("".""."Pot_Lights",2);
    Drive_Switch("".""."E1_Energy.Ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch("".""."E2_Energy.Ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch("".""."E3_Energy.Ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch("".""."E4_Energy.Ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    if (Balance==1){
        Hide_Frame(""."Big_Viewport_CRT"."Fuel_Balances";
        Balance=0;
    }
    *Set_Current_Layer(Layer1)*;
    Drive_Light("".""."E1_Energy.Ref.Energy_Reference.Energy_Ref.Light",1);
    Drive_Light("".""."E2_Energy.Ref.Energy_Reference.Energy_Ref.Light",1);
    Drive_Light("".""."E3_Energy.Ref.Energy_Reference.Energy_Ref.Light",1);
    Drive_Light("".""."E4_Energy.Ref.Energy_Reference.Energy_Ref.Light",1);
    if (Fuel_Zoom==1);
    {
        Hide_Frame(""."Big_Viewport_CRT"."Fuel_Zoomed";
        Fuel_Zoom=0;
        Hide_Frame("".""."Tank_Light_2.Test";
    }
    if (Fuel_System==1);
    {
        Hide_Frame(""."Big_Viewport_CRT"."Fuel_system";
        Fuel_System=0;
        Hide_Frame("".""."Tank_Light_2.Test";
    }
    if (Engine_Zoom==0);
    {
        Appear_Frame(""."Big_Viewport_CRT"."Engine_Zoomed";
        Scale_Frame(""."Big_Viewport_CRT"."Engine_Zoomed",1.1.5);
        Engine_Zoom=1;
    }
    *Set_Current_Layer(Layer1)*;
    ptx=1043;
    pty=443;
    Drive_Light("".""."Pot_Lights",1);

Channel_Member_Put_Buffert("Session_Common",""."".""."Location",MEMBER_DIMEN(1),12.1.CHAN_FLOAT&pox.1.CHANNEL_WRITE);

Channel_Member_Put_Buffert("Session_Common",""."".""."Location",MEMBER_DIMEN(1),13.1.CHAN_FLOAT&pox.1.CHANNEL_WRITE);
Drive_Plot("vertical",1",poty);
Drive_Plot("horizontal",1",potx);
Drive_Light("Labels",2);

} -> Monitor
(SWITCH Engine_Selector1 'code2')
{
  
  Appear_Frame(""Energy_Ent_Plots2");
  Drive_Switch(""Fuel_Overview_Sw",1);
  Drive_Switch(""Engine_Selector2",14);
  Drive_Switch(""Engine_Selector3",14);
  Drive_Switch(""E1_Energy_ref.Energy_Reference.Hs_Plot.Ref.Hp_s_plot.Energy_Select_sw",1);
  Drive_Switch(""E2_Energy_ref.Energy_Reference.Hs_Plot.Ref.Hp_s_plot.Energy_Select_sw",1);
  Drive_Switch(""E3_Energy_ref.Energy_Reference.Hs_Plot.Ref.Hp_s_plot.Energy_Select_sw",1);
  Drive_Switch(""E4_Energy_ref.Energy_Reference.Hs_Plot.Ref.Hp_s_plot.Energy_Select_sw",1);

  if (Balance==1){
Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
Balance=0;
  }

  *Set_Current_LayerLayer1* :
  Drive_Light(""E1_Energy_ref.Energy_Reference.Energy_Ref_Light",1);
  Drive_Light(""E2_Energy_ref.Energy_Reference.Energy_Ref_Light",1);
  Drive_Light(""E3_Energy_ref.Energy_Reference.Energy_Ref_Light",1);
  Drive_Light(""E4_Energy_ref.Energy_Reference.Energy_Ref_Light",1);

  if (Fuel_Zoom==1);
  
  Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
  Fuel_Zoom=0;
  Hide_Frame(""Tank_Light_1_Test");
  }

  if (Fuel_System==1);
  
  Hide_Frame("Big_Viewport_CRT","Fuel_system");
  Fuel_System=0;
  Hide_Frame(""Tank_Light_1_Test");

  if (Engine_Zoom==0);
  
  Appear_Frame("Big_Viewport_CRT","Engine_Zoomed");
  Scale_Frame("Big_Viewport_CRT",1,1,1,1,5);
  Engine_Zoom=1;
  }

  *Set_Current_LayerLayer1* :
potx=1043;
poty=443;
  Drive_Light(""Pot_Lights_1",1,1);
  Drive_Light(""Pot_Lights",2);

  Channel_Member_Put_Buffet"Session_Common","","Location",MEMBER_DIMEN(1,12,1.CHAN_FLOAT,&potx,1.CHANNEL_WRITE);

  Channel_Member_Put_Buffet"Session_Common","","Location",MEMBER_DIMEN(1,13,1.CHAN_FLOAT,&potx,1.CHANNEL_WRITE);

  Drive_Plot("vertical",1",poty);
  Drive_Plot("horizontal",1",potx);
  Drive_Light(""Labels",2);
} -> Monitor
(SWITCH Engine_Selector1 'code3')
{
  
  Appear_Frame(""Energy_Ent_Plots2");
  Drive_Switch(""Fuel_Overview_Sw",1);
  Drive_Switch(""Engine_Selector2",14);
  Drive_Switch(""Engine_Selector3",14);
if (Balance==1) {
    Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
    Balance=0;
}

*Set_Current_Layer(Layer1)*;
Drive_Light("E1_Energy_ref.Energy_Reference.Energy_Ref_Light",1);
Drive_Light("E2_Energy_ref.Energy_Reference.Energy_Ref_Light",2);
Drive_Light("E3_Energy_ref.Energy_Reference.Energy_Ref_Light",3);
Drive_Light("E4_Energy_ref.Energy_Reference.Energy_Ref_Light",0);

if (Fuel_Zoom==1) {
    Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
    Fuel_Zoom=0;
    Hide_Frame("Tank_Light_2_Test");
}
if (Fuel_System==1) {
    Hide_Frame("Big_Viewport_CRT","Fuel_System");
    Fuel_System=0;
    Hide_Frame("Tank_Light_2_Test");
}
if (Engine_Zoom==0) {
    Appear_Frame("Big_Viewport_CRT","Engine_Zoomed");
    Scale_Frame("Big_Viewport_CRT","Engine_Zoomed",1,1.5);
    Engine_Zoom=1;
}
*Set_Current_Layer(Layer1)*;
px1=156;
px2=443;
Drive_Light("Pot_Lights_1",1);

Channel_Member_Put_Buffer("Session_Common","","Location",MEMBER_DIMEN(1),12,1,CHAN_FLOAT,0x1,CHANNEL_WRITE);

Channel_Member_Put_Buffer("Session_Common","","Location",MEMBER_DIMEN(1),13,1,CHAN_FLOAT,0x1,CHANNEL_WRITE);

Drive_Pot("vertical_1",px1);
Drive_Pot("horizontal_1",px2);

Drive_Light("Labels",2);
} -> Monitor
(SWITCH Engine_Selector1 'coded')

{(Appears Frame("Energy_Entity_Plot2");
  Drive_Switch("Fuel_Overview_Sw",1);
  Drive_Switch("Engine_Selector2",14);
  Drive_Switch("Engine_Selector3",14);
  Drive_Switch("Engine_Selector4",14);
  Drive_Light("Pot_Lights",2);
  Drive_Switch("E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
  Drive_Switch("E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
  Drive_Switch("E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
  Drive_Switch("E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
}
if (Balance==1) {
    Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
    Balance=0;
}
if (Fuel_Zoom==1){
    Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
    Fuel_Zoom=0;
    Hide_Frame("Tank_Light_2_Test");
}
if (Fuel_System==1){
    Hide_Frame("Big_Viewport_CRT","Fuel_system");
    Fuel_System=0;
    Hide_Frame("Tank_Light_2_Test");
}
if (Engine_Zoom==0){
    Appear_Frame("Big_Viewport_CRT","Engine_Zoomed");
    Scale_Frame("Big_Viewport_CRT","Engine_Zoomed",1.15);
    Engine_Zoom=1;
}
*Set_Current_LayerLayer1*;
pots=199;
poty=1070;
Drive_Light",","Pot_Lights_1",",1,

Channel_Member_Put_Buffen"Session_Common","","Location",MEMBER_DIMEN(11,12,1.CHAN_FLOAT,&pots,1.CHANNEL_WRITE);

Channel_Member_Put_Buffen"Session_Common","","Location",MEMBER_DIMEN(11,13,1.CHAN_FLOAT,&poty,1.CHANNEL_WRITE);

    Drive_Pott","vertical_1",poty;
    Drive_Pott","horizontal_1",poty;

Drive_Light","labels",2,);
} -> Monitor
(SWITCH Engine_Selected1 'code5')
{
    Appear_Frame","Energy_Ent_Plots2";
    Drive_Switch","FuelOverview_Sw",1);
    Drive_Switch","Engine_Selector2",14);
    Drive_Switch","Engine_Selector3",14);
    Drive_Switch","Engine_Selector4",14);
    Drive_Light",","Pot_Lights",2,);
    Drive_Switch","E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch","E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch","E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch","E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);

if (Balance==1){
    Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
    Balance=0;
}
*Set_Current_LayerLayer1*;

Driver_Light",","E1_Energy_ref.Energy_Reference.Energy_Ref_Light",1);
Driver_Light",","E2_Energy_ref.Energy_Reference.Energy_Ref_Light",2);
Driver_Light",","E3_Energy_ref.Energy_Reference.Energy_Ref_Light",2);
Driver_Light",","E4_Energy_ref.Energy_Reference.Energy_Ref_Light",0);

if (Fuel_Zoom==1){
    Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
Fuel_Zoom=1;
'Hide_Frame("Tank_Light_2_Test");
}
if (Fuel_System=1);
{
  Hide_Frame("Big_Viewport_CRT","Fuel_system");
  Fuel_System=0;
  Hide_Frame("Tank_Light_2_Test");
}
if (Engine_Zoom==0);
{
  Appear_Frame("Big_Viewport_CRT","Engine_Zoomed");
  Scale_Frame("Big_Viewport_CRT","Engine_Zoomed",1,1.5);
  Engine_Zoom=1;
  "Set_Current_LayerLayer1":
  xpts=665;
  ypts=0;
  Drive_Light("Pot_Lights_1",1);

  Channel_Member_Put_Bufferan"Session_Common","Location","MEMBER_DIMEN(1),12,1.CHAN_FLOAT"&poxy,1.CHANNEL_WRITE);

  Channel_Member_Put_Bufferan"Session_Common","Location","MEMBER_DIMEN(1),13,1.CHAN_FLOAT"&poxy,1.CHANNEL_WRITE);

    Drive_Pot("vertical",1,ypts);
    Drive_Pot("horizontal",1,px);
    Drive_Light("Labels",2);
  } -> Monitor
(SWITCH Engine_Selector1 'code6')
{
  

  Channel_Member_Put_Bufferan"Session_Common","Location","MEMBER_DIMEN(1),12,1.CHAN_FLOAT"&poxy,1.CHANNEL_WRITE);

  Channel_Member_Put_Bufferan"Session_Common","Location","MEMBER_DIMEN(1),13,1.CHAN_FLOAT"&poxy,1.CHANNEL_WRITE);

    Drive_Pot("vertical",1,ypts);
    Drive_Pot("horizontal",1,px);
    Drive_Light("Labels",2);
  } -> Monitor
(SWITCH Engine_Selector1 'code6')
{
  
  Channel_Member_Put_Bufferan"Session_Common","Location","MEMBER_DIMEN(1),12,1.CHAN_FLOAT"&poxy,1.CHANNEL_WRITE);

  Channel_Member_Put_Bufferan"Session_Common","Location","MEMBER_DIMEN(1),13,1.CHAN_FLOAT"&poxy,1.CHANNEL_WRITE);

    Drive_Pot("vertical",1,ypts);
    Drive_Pot("horizontal",1,px);
    Drive_Light("Labels",2);
  } -> Monitor
(SWITCH Engine_Selector1 'code6')
{
{ 
  Appear_Frame("Big_Viewport_CRT","Engine_Zoomed");
  Scale_Frame("Big_Viewport_CRT","Engine_Zoomed",1.1,1.5);
  Engine_Zoom = 1;
}

*Set_Current_LayerLayer1* :
  ptx = 438;
  pty = 441;
  Drive_Light"".""Pot_Lights_1",1;

Channel Mitglied Put Buffen "Session_Common"."".""Location", MEMBER_DIMEN[1], 12, 1.CHAN_FLOAT,&ptx, 1.CHANNEL_WRITE);

Channel Mitglied Put Buffen "Session_Common"."".""Location", MEMBER_DIMEN[1], 13, 1.CHAN_FLOAT,&pty, 1.CHANNEL_WRITE);

  Drive_Pot"".""vertical", pty;
  Drive_Pot"".""horizontal", ptx;
  Drive_Light"".""Labels", 2;
)
} -> Monitor
(SWITCH Engine_Selector1 code?)

{ 
  Appear_Frame("".""Energy_Ent_Plots2");
  Drive_Switch"".""Fuel_Overview_Sw", 1);
  Drive_Switch"".""Engine_Selector2", 14);
  Drive_Switch"".""Engine_Selector3", 14);
  Drive_Switch"".""Engine_Selector4", 14);
  Drive_Light"".""Pot_Lights", 2;
  Drive_Switch"".""E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw", 8);
  Drive_Switch"".""E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw", 8);
  Drive_Switch"".""E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw", 8);
  Drive_Switch"".""E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw", 8);

  if (Balance == 1)
      Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
     Balance = 0;
  }

*Set_Current_LayerLayer1* :
  Drive_Light"".""E1_Energy_ref.Energy_Reference.Energy_Ref_Light", 1;
  Drive_Light"".""E2_Energy_ref.Energy_Reference.Energy_Ref_Light", 2;
  Drive_Light"".""E3_Energy_ref.Energy_Reference.Energy_Ref_Light", 3;
  Drive_Light"".""E4_Energy_ref.Energy_Reference.Energy_Ref_Light", 0;

  if (Fuel_Zoom == 1);
  { 
    Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
    Fuel_Zoom = 0;
    Hide_Frame("".""Tank_Light_2_Test");
  }
  if (Fuel_System == 1);
  { 
    Hide_Frame("Big_Viewport_CRT","Fuel_system");
    Fuel_System = 0;
    Hide_Frame("".""Tank_Light_2_Test");
  }
  if (Engine_Zoom == 0);
  { 
    Appear_Frame("Big_Viewport_CRT","Engine_Zoomed");
    Scale_Frame("Big_Viewport_CRT","Engine_Zoomed",1.1,1.5);
    Engine_Zoom = 1;
  }

*Set_Current_LayerLayer1* :
  ptx = 938;
  pty = 341;
  Drive_Light"".""Pot_Lights_1", 1;
}
Channel_Member_Put_Buffen"Session_Common","","Location",MEMBER_DIMEN(11,12,1,CHAN_FLOAT,&potx,1,CHANNEL_WRITE_E);

Channel_Member_Put_Buffen"Session_Common","","Location",MEMBER_DIMEN(11,13,1,CHAN_FLOAT,&poty,1,CHANNEL_WRITE_E);

    Drive_Pot"","vertical",poty;
    Drive_Pot"","horizontal",potx;

    Drive_Light"","Labels",2);
} -> Monitor
(SWITCH Engine_Selector\c (code8))
{
    Appear_Frame"","Energy_Ent_Plot2";
    Drive_Switch"","Fuel_Overview_Sw",1);
    Drive_Switch"","Engine_Selector2",14);
    Drive_Switch"","Engine_Selector3",14);
    Drive_Switch"","Engine_Selector4",14);
    Drive_Light"","Pot_Lights",2);
    Drive_Switch"","E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch"","E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch"","E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch"","E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);

    if ((Balance==11{
        Hide_Frame("Big_Viewport_CRT","Fuel_Balances")
        Balance=0;
    })

    *Set_Current_LayerLayer1*;
    Drive_Light"","E1_Energy_ref.Energy_Reference.Energy_Ref_Light",1);
    Drive_Light"","E2_Energy_ref.Energy_Reference.Energy_Ref_Light",2);
    Drive_Light"","E3_Energy_ref.Energy_Reference.Energy_Ref_Light",3);
    Drive_Light"","E4_Energy_ref.Energy_Reference.Energy_Ref_Light",0);

    if ((Fuel_Zoom==1):
    {
        Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed")
        Fuel_Zoom=0;
        Hide_Frame("","Tank_Light 2_Test")
    }
    if ((Fuel_System==1):
    {
        Hide_Frame("Big_Viewport_CRT","Fuel_system")
        Fuel_System=0;
        Hide_Frame("","Tank_Light 2_Test")
    }
    if ((Engine_Zoom==0):
    {
        Appear_Frame"Big_Viewport_CRT","Engine_Zoomed")
        Scale_Frame"Big_Viewport_CRT","Engine_Zoomed",1,1,5;
        Engine_Zoom=1;
    }

    *Set_Current_LayerLayer1*;
    potx=19;
    poty=0;
    Drive_Light"","Pot_Lights",1,1);
void Drive_Light("","Labels",2);
    ) -> Monitor
    (SWITCH Engine_Selector1 'code9')
{
    Appear_Frame("","Energy_Ent_Plots2");
    Drive_Switch("","Fuel_Overview_Sw",1);
    Drive_Switch("","Engine_Selector2",14);
    Drive_Switch("","Engine_Selector3",14);
    Drive_Switch("","Engine_Selector4",14);
    Drive_Light("","Pot_Lights",2);
    Drive_Switch("","E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch("","E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch("","E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch("","E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);

    if (Balance==11:
        Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
        Balance=0;
    }
    *Set_Current_Layer(Layer1)*;
    Drive_Light("","E1_Energy_ref.Energy_Reference.Energy_Ref.Light",1);
    Drive_Light("","E2_Energy_ref.Energy_Reference.Energy_Ref.Light",2);
    Drive_Light("","E3_Energy_ref.Energy_Reference.Energy_Ref.Light",3);
    Drive_Light("","E4_Energy_ref.Energy_Reference.Energy_Ref.Light",0);

    if (Fuel_Zoom==1:
    {
        if (Fuel_System==1):
        {
            Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
            Fuel_Zoom=0;
            Hide_Frame("","Tank_Light_2_Test");
        }
        if (Engine_Zoom==0):
        {
            if (Engine_Zoom==0:
            {
                Appear_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed",1.1,1.5);
                Engine_Zoom=1;
            }
            *Set_Current_Layer(Layer1)*;
            potx=0;
            poty=867;
            Drive_Light("","Pot_Lights",1",1);"

    Channel_Member_Put_Buffert("Session_Common","","Location",MEMBER_DIMEN(1),12.1.CHAN_FLOAT,&potx,1.CHANNEL_WRIT_E1;
    Channel_Member_Put_Buffert("Session_Common","","Location",MEMBER_DIMEN(1),13.1.CHAN_FLOAT,&poty,1.CHANNEL_WRIT_E1;

    Drive_Pott("","vertical",poty);
    Drive_Pott("","horizontal",potx);

    ) -> Monitor
    (SWITCH Engine_Selector1 'code10')
{
    Appear_Frame("","Energy_Ent_Plots2");
    Drive_Switch("","Fuel_Overview_Sw",1);
    Drive_Switch("","Engine_Selector2",14);
    Drive_Switch("","Engine_Selector3",14);
    Drive_Switch("","Engine_Selector4",14);
    Drive_Light("","Pot_Lights",2);"}
Drive_Switch","E1_Energy_ref.Energy_Reference.Hs.Plot.Ref.Hp.s.plot.Energy_Select_sw",8);
Drive_Switch","E3_Energy_ref.Energy_Reference.Hs.Plot.Ref.Hp.s.plot.Energy_Select_sw",8);

if (Balance==1){
  Hide_Frame("BigViewport_CRT","Fuel_Balances");
  Balance=0;
}

*Set_Current_LayerLayer1*;

Drive_Light","E1_Energy_ref.Energy_Reference.Energy_Ref.Light",11);
Drive_Light","E2_Energy_ref.Energy_Reference.Energy_Ref.Light",21);
Drive_Light","E3_Energy_ref.Energy_Reference.Energy_Ref.Light",31);
Drive_Light","E4_Energy_ref.Energy_Reference.Energy_Ref.Light",41);

if (Fuel_Zoom==1);
  { Hide_Frame("BigViewport_CRT","Fuel_Zoomed");
    Fuel_Zoom=0;
    Hide_Frame("Tank Light 2 Test");
  }
  if (Fuel_System==1);
    { Hide_Frame("BigViewport_CRT","Fuel System");
      Fuel_System=0;
      Hide_Frame("Tank Light 2 Test");
    }
  if (Engine_Zoom==0);
    { Appear_Frame("BigViewport_CRT","Engine_Zoomed");
      Scale_Frame("BigViewport_CRT","Engine_Zoomed",1,1,5);
      Engine_Zoom=1;
    }
*Set_Current_LayerLayer1*;
pox=300;
poy=500;
Drive_Light","Pot_Lights",11);

Channel_Member_Put_Buffer"Session.Common","","Location",MEMBER_DIM(11,12,1,CHAN_FLOAT,&pox,1,CHANNEL_WRITE);

Channel_Member_Put_Buffer"Session.Common","","Location",MEMBER_DIM(11,13,1,CHAN_FLOAT,&poy,1,CHANNEL_WRITE);

Drive_Pot","vertical",poy);
Drive_Pot","horizontal",pox);

Drive_Light","Labels",2);
} -> Monitor
(SWITCH Engine_Selector1'code 11')
{
  Appear_Frame","Energy_Ent_Plots2");
  Drive_Switch","Fuel_Overview_Sw",11);
  Drive_Switch","Engine_Selector2",14);
  Drive_Switch","Engine_Selector3",14);
  Drive_Switch","Engine_Selector4",14);
  Drive_Light","Pot_Lights",2);
  Drive_Switch","E1_Energy_ref.Energy_Reference.Hs.Plot.Ref.Hp.s.plot.Energy_Select_sw",8);
  Drive_Switch","E3_Energy_ref.Energy_Reference.Hs.Plot.Ref.Hp.s.plot.Energy_Select_sw",8);

if (Balance==1){
  Hide_Frame("BigViewport_CRT","Fuel_Balances");
  Balance=0;
}

*Set_Current_LayerLayer1*;
Drive Light"="E1_Energy_ref.Energy_Reference.Energy_Ref_Light",1;
Drive Light"="E2_Energy_ref.Energy_Reference.Energy_Ref_Light",2;
Drive Light"="E3_Energy_ref.Energy_Reference.Energy_Ref_Light",3;
Drive Light"="E4_Energy_ref.Energy_Reference.Energy_Ref_Light",4;

if (Fuel_Zoom=1):
    { Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
        Fuel_Zoom=0;
        Hide_Frame("Tank_Light_2","Test");
    }
    if (Fuel_System=1):
        { Hide_Frame("Big_Viewport_CRT","Fuel_system");
            Fuel_System=0;
            Hide_Frame("Tank_Light_2","Test");
        }
    if (Engine_Zoom=1):
        { Appear_Frame("Big_Viewport_CRT","Engine_Zoomed");
            Scale_Frame("Big_Viewport_CRT","Engine_Zoomed",1.15);
            Engine_Zoom=1;
        }
    *Set_Current_Layer(Layer1)*:
        px=-300;
        py=-500;
        Drive Light"="Pot_Lights","1",1;

Channel_Member_Put_Buffer"Session_Common","Location",MEMBER_DIMEN(11,12,1,CHAN_FLOAT,&px,1,CHANNEL_WRITE);

Channel_Member_Put_Buffer"Session_Common","Location",MEMBER_DIMEN(11,13,1,CHAN_FLOAT,&px,1,CHANNEL_WRITE);

Drive Pot"="Vertical",py;
Drive Pot"="Horizontal",px;

Drive Light"="Labels",2;
} -> Monitor
(SWITCH Engine_Selector1 'code12')
{
    Appear_Frame"="Energy_Ent_Plots2";
    Drive Switch"="Fuel_Overview_Sw",1;
    Drive Switch"="Engine_Selector2",14;
    Drive Switch"="Engine_Selector3",14;
    Drive Switch"="Engine_Selector4",14;
    Drive Light"="Pot_Lights",2;
    Drive Switch"="E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8;
    Drive Switch"="E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8;
    Drive Switch"="E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8;
    Drive Switch"="E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8;

    if (Balance=1){
        Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
        Balance=0;
    }
    *Set_Current_Layer(Layer1)*:
        Drive Light"="E1_Energy_ref.Energy_Reference.Energy_Ref_Light",1;
        Drive Light"="E2_Energy_ref.Energy_Reference.Energy_Ref_Light",2;
        Drive Light"="E3_Energy_ref.Energy_Reference.Energy_Ref_Light",3;
        Drive Light"="E4_Energy_ref.Energy_Reference.Energy_Ref_Light",4;

    if (Fuel_Zoom=1):
        { Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
            Fuel_Zoom=0;
        }
Hide_Frame("Tank_Light_2_Test");
}
if (Fuel_System==1){
    
    Hide_Frame("Big_Viewport_CRT","Fuel_system");
    Fuel_System=0;
    Hide_Frame("Tank_Light_2_Test");
}
if (Engine_Zoom==0){
    
    Appear_Frame("Big_Viewport_CRT","Engine_Zoomed");
    Scale_Frame("Big_Viewport_CRT","Engine_Zoomed",1.1,5);
    Engine_Zoom=1;
}
*Set_Current_Layer(Layer1)*; 
potx=500;
poty=500;
Drive_Light("Pot_Lights_1",1);

Channel_Member_Put_Buffern(Location, MEMBER_DIM(1),1,CHANCE_FLOAT, 0,1,CHANNEL_WRITE)

Drive_Potn(L,1, poty);
Drive_Potn(L,2, potx);

Drive_Lightn("Labels",2);
} -> Monitor (SWITCH Engine_Selector1 'code13')
{
    
    Appear_Frame("Fuel_Ent_Plots2");
    Drive_Switchn("Engine_Selector1",14);
    Drive_Lightn("Pot_Lights",2);
    Drive_Lightn("Pot_Lights",1,1);
    Drive_Switchn("Fuel_Overview_Sw",1);
    Drive_Switchn("E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switchn("E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switchn("E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switchn("E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);

    if (Engine_Zoom==1){
        
        Hide_Frame("Big_Viewport_CRT","Engine_Zoomed");
        Get_Frame("Energy_Ent_Plots2",0,FOREGROUND_FRAME);
        Engine_Zoom=0;
    }
}
if (Balance==1){
Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
Balance=0;
Hide_Frame("Tank_Light_2_Test");
}
if (Fuel_Zoom==1){
    
    Hide_Frame("Tank_Light_2_Test");
    Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
    Fuel_Zoom=0;
}
if (Fuel_System==1){
    
    Hide_Frame("Big_Viewport_CRT","Fuel_system");
    Fuel_System=0;
}
*Set_Current_Layer(Layer1)*;
Drive_Lightn("E1_Energy_ref.Energy_Reference.Energy_Ref.Light",1);
Drive_Lightn("E2_Energy_ref.Energy_Reference.Energy_Ref.Light",2);
Drive_Lightn("E3_Energy_ref.Energy_Reference.Energy_Ref.Light",3);
Drive Light"."E1_Energy refinery_Energy Reference.Energy_Ref.Light",0);
Drive Light"."Labels",2);
} -> Monitor
{
Appearance"."Energy Ent Plots2";
Drive Switch"."Fuel Overview_Sw",1);
Drive Switch"."Engine Selector1",14);
Drive Switch"."Engine Selector2",14);
Drive Switch"."Engine Selector3",14);
Drive Switch"."Engine Selector4",14);

Drive Switch"."E2_Energy_ref.Energy Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw","8);
Drive Switch"."E3_Energy_ref.Energy Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw","8);
Drive Switch"."E4_Energy_ref.Energy Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw","8);

Drive Light"."Pot_Lights",2);
Drive Light"."Pot_Lights",1",1);

if (Balance==1){
  Hide Frame"."Big Viewport CRT","Fuel Balances"
  Balance=0;
}

if (Fuel Zoom==1){
  Hide Frame"."Big Viewport CRT","Fuel Zoomed"
  Fuel Zoom=0;
  Hide Frame"."Tank Light 2",Test"
}

if (Fuel System==1){
  Hide Frame"."Big Viewport CRT","Fuel system"
  Fuel System=0;
  Hide Frame"."Tank Light 2",Test"
}

if (Engine Zoom==0){
  Appearance"."Big Viewport CRT","Engine Zoomed"
  Scale Frame"."Big Viewport CRT","Engine Zoomed",1.15)
  Engine Zoom=1;
} *Set Current LayerLayer1*;
potx=226;
poty=444;
Drive Light"."Pot_Lights_1",1,1);

Channel Member Put Buffer"Session Common","","Location",MEMBER DIMEN(11.12.1.CHAN FLOAT,potx,1.CHANNEL_WRITE E);
Channel Member Put Buffer"Session Common","","Location",MEMBER DIMEN(11.13.1.CHAN FLOAT,poty,1.CHANNEL_WRITE E);

Drive Pot",".Vertical_{1},poty);
Drive Pot",".Horizontal_{1},potx);

Drive Light"."Labels",2);

Drive Light"."E1_Energy_ref.Energy Reference.Energy_Ref.Light",1);
Drive Light"."E2_Energy_ref.Energy Reference.Energy_Ref.Light",2);
Drive Light"."E3_Energy_ref.Energy Reference.Energy_Ref.Light",3);
Drive Light"."E4_Energy_ref.Energy Reference.Energy_Ref.Light",0);
} -> Monitor
(SWITCH E1_Energy_ref.Energy Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw '1-2')
{
  Drive Switch"."Fuel Overview_Sw",1);
}
Drive_Switch("Engine_Selector1",14);
Drive_Switch("Engine_Selector2",14);
Drive_Switch("Engine_Selector3",14);
Drive_Switch("Engine_Selector4",14);
Drive_Switch("E2_Energy_ref_Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
Drive_Switch("E3_Energy_ref_Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
Drive_Switch("E4_Energy_ref_Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);

Drive_Light("Pot_Lights",2);

if (Balance==1){
    Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
    Balance=0;
}

"Set_Current_LayerLayer1":
Drive_Light("E1_Energy_ref_Energy_Reference.Energy_Ref_Light",1);
Drive_Light("E2_Energy_ref_Energy_Reference.Energy_Ref_Light",2);
Drive_Light("E3_Energy_ref_Energy_Reference.Energy_Ref_Light",3);
Drive_Light("E4_Energy_ref_Energy_Reference.Energy_Ref_Light",0);

if (Fuel_Zoom==1){
    Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
    Fuel_Zoom=0;
    Hide_Frame("Tank_Light_1_Test");
}
if (Fuel_System==1){
    Hide_Frame("Big_Viewport_CRT","Fuel_system");
    Fuel_System=0;
    Hide_Frame("Tank_Light_1_Test");
}
if (Engine_Zoom==1){
    Appear_Frame("Big_Viewport_CRT","Engine_Zoomed");
    Scale_Frame("Big_Viewport_CRT","Engine_Zoomed",1.1,1.5);
    Engine_Zoom=0;
}

"Set_Current_LayerLayer1":
potx=226;
poty=444;
Drive_Light("Pot_Lights",1,1);

Channel_Member_Put_Buffert("Session_Common",""."Location",MEMBER_DIMEN(1),12.1.CHAN_FLOAT,&potx.1.CHANNEL_WRITE);

Channel_Member_Put_Buffert("Session_Common",""."Location",MEMBER_DIMEN(1),13.1.CHAN_FLOAT,&poty.1.CHANNEL_WRITE);

    Drive_Pott("vertical_1",poty);
    Drive_Pott("horizontal_1",potx);
    Drive_Light("Labels",2);
} -> Monitor
(SWITCH E1_Energy_ref_Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw '2 - 3')
{
    Drive_Switch("Fuel_Overview_Sw",1);
    Drive_Switch("Engine_Selector1",14);
    Drive_Switch("Engine_Selector2",14);
    Drive_Switch("Engine_Selector3",14);
    Drive_Switch("Engine_Selector4",14);
    Drive_Switch("E2_Energy_ref_Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch("E3_Energy_ref_Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch("E4_Energy_ref_Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
Drive_Light("0","Pot_Lights",2);

if (Balance==1) {
    Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
    Balance=0;
}
*Set_Current_LayerLayer1*;
Drive_Light("0","E1_Energy_ref.Energy_Reference.Energy_Ref_Light",1);
Drive_Light("0","E2_Energy_ref.Energy_Reference.Energy_Ref_Light",2);
Drive_Light("0","E3_Energy_ref.Energy_Reference.Energy_Ref_Light",3);
Drive_Light("0","E4_Energy_ref.Energy_Reference.Energy_Ref_Light",4);
}
if (Fuel_Zoom==1) {
    if (Fuel_Zoom==0) {
        Hide_Frame("Big_Viewport_CRT","Tank_Light_2_Test");
    }
    if (Fuel_System==1) {
        if (Tank_Light_2_Test) {
            Fuel_System=0;
            Hide_Frame("Big_Viewport_CRT","Fuel_system");
        }
    }
    if (Engine_Zoom==0) {
        if (Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
            Engine_Zoom=1;
        }
        *Set_Current_LayerLayer1*;
        posx=226;
        posy=444;
        Drive_Light("0","Pot_Lights",1);
    }
    Channel_Member_Put_Buffer("Session_Common","Location",MEMBER_DIM(1,12,1,CHAN_FLOAT&posx,1,CHANNEL_WRITE_E1);
    Channel_Member_Put_Buffer("Session_Common","Location",MEMBER_DIM(1,13,1,CHAN_FLOAT&posy,1,CHANNEL_WRITE_E1);
    Drive_Ptr("0","vertical_1",posy);
    Drive_Ptr("0","horizontal_1",posx);
    Drive_Light("0","Labels",2);
} -> Monitor
(SWITCH E1_Energy_ref.Energy_Reference.Hs_Plot_Ref_Hp_s_plot.Energy_Select_sw '3-4') {
    Drive_Switch("0","Fuel_Overview_Sw",1);
    Drive_Switch("0","Engine_Selector1",14);
    Drive_Switch("0","Engine_Selector2",14);
    Drive_Switch("0","Engine_Selector3",14);
    Drive_Switch("0","Engine_Selector4",14);
    Drive_Switch("0","E2_Energy_ref.Energy_Reference_Hs_Plot_Ref_Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch("0","E3_Energy_ref.Energy_Reference_Hs_Plot_Ref_Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch("0","E4_Energy_ref.Energy_Reference_Hs_Plot_Ref_Hp_s_plot.Energy_Select_sw",8);
}
if (Balance==1) {
    Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
    Balance=0;
}
if (Fuel_Zoom=1):
    
    { Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
    Fuel_Zoom=0;
    Hide_Frame("Tank_Light_2_Test");
    }
    if (Fuel_System=1):
    
    { Hide_Frame("Big_Viewport_CRT","Fuel_system");
    Fuel_System=0;
    Hide_Frame("Tank_Light_2_Test");
    }
    if (Engine_Zoom=0):
    
    { Appear_Frame("Big_Viewport_CRT","Engine_Zoomed");
    Scale_Frame("Big_Viewport_CRT","Engine_Zoomed",1.1,1.5);
    Engine_Zoom=1;
    }
    *Set_Current_LayerLayer1*: potx=226;
    poty=444;
    Drive_Light("Pot_Lights_1",1);

Channel_Member_Put_Buffer("Session_Common","","Location",MEMBER_DIMEN1,12,1.CHANNEL_FLOAT&potx,1.CHANNEL_WRITE);

Channel_Member_Put_Buffer("Session_Common","","Location",MEMBER_DIMEN1,13,1.CHANNEL_FLOAT&poty,1.CHANNEL_WRITE);

    Drive_Pott("vertical_1",poty);
    Drive_Pott("horizontal_1",potx);
    Drive_Light("Labels",2);
} -> Monitor

{SWITCH E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw '5-6'}
    
    { Drive_Switch("FuelOverview_Sw",1);
    Drive_Switch("Engine_Selector1",14);
    Drive_Switch("Engine_Selector2",14);
    Drive_Switch("Engine_Selector3",14);
    Drive_Switch("Engine_Selector4",14);
    Drive_Switch("E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch("E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch("E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);

    Drive_Light("Pot_Lights",2);

if (Balance==1){
    Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
    Balance=0;
    *Set_Current_LayerLayer1*:
    Drive_Light("E1_Energy_ref.Energy_Reference.Energy_Ref_Light",1);
    Drive_Light("E2_Energy_ref.Energy_Reference.Energy_Ref_Light",2);
    Drive_Light("E3_Energy_ref.Energy_Reference.Energy_Ref_Light",3);
    Drive_Light("E4_Energy_ref.Energy_Reference.Energy_Ref_Light",0);

if (Fuel_Zoom=1):
    
    { Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
    Fuel_Zoom=0;
    Hide_Frame("Tank_Light_2_Test");
    }
if (Fuel_System==1) {
    if (Engine_Zoom==0) {
        Scale_Framer("Big_Viewport_CRT","Engine_Zoomed");
        Engine_Zoom==1;
    }
    *Set_Current_LayerLayer1*;
    Scale_Framer("Big_Viewport_CRT","Engine_Zoomed");
    Engine_Zoom==1;
}

Channel_Member_Put_Buffer("Session_Common","Location",MEMBER_DIMEN(1.12.1.CHAN_FLOAT.&potx.1.CHANNEL_WRITE));

Channel_Member_Put_Buffer("Session_Common","Location",MEMBER_DIMEN(1.13.1.CHAN_FLOAT.&poty.1.CHANNEL_WRITE));

Drive_Pot("",vertical_1","poty");
Drive_Pot("",horizontal_1","potx");
Drive_Light("","Labels","");

if (Balance==1) {
    Hide_Framer("Big_Viewport_CRT","Fuel_Balances");
}

if (Fuel_Zoom==1) {
    if (Engine_Zoom==0) {
        Scale_Framer("Big_Viewport_CRT",Fuel_Zoomed");
        Fuel_Zoom==1;
    }
    *Set_Current_LayerLayer1*;
    Scale_Framer("Big_Viewport_CRT", Fuel_Zoomed");
    Fuel_Zoom==1;
}

if (Engine_Zoom==0) {
    }
Appear_Frame("Big_Viewport_CRT","Engine_Zoomed");
Scale_Frame("Big_Viewport_CRT","Engine_Zoomed",1.15);
Engine_Zoom=1;
}
*Set_Current_Layer(Layer1)*
 pots=226;
poty=444;
Drive_Light"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""”
pots=1043;
pots=443;
Drive_Light("".""Pot_Lights_1"".""1");

Channel_Member_Put_Buffer("Session_Common".""."".""Location"".""MEMBER_DIMEN(1),12.1.CHAN_FLOAT.&pots,1.CHANNEL_WRITE_Ei);

Channel_Member_Put_Buffer("Session_Common".""."".""Location"".""MEMBER_DIMEN(1),13.1.CHAN_FLOAT.&pots,1.CHANNEL_WRITE_Ei);

Drive_Pot("".""vertical_1"".""pots");
Drive_Pot("".""horizontal_1"".""pots");
Drive_Light("".""Labels"".""3");

}) -> Monitor
(SWITCH Engine_Selector2 'code2')
{
  Appear_Framer("".""Energy_Ent_Plots2"".);
  Drive_Switch("".""Fuel_Overview_Sw""."".1");
  Drive_Switch("".""Engine_Selector1"".""14");
  Drive_Switch("".""Engine_Selector2"".""14");
  Drive_Switch("".""E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw"".""8");
  Drive_Switch("".""E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw"".""8");
  Drive_Switch("".""E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw"".""8");
  Drive_Switch("".""E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw"".""8");

  if (Balance==1)
  {
    Hide_Framer("".""Big_Viewport_CRT"".""Fuel_Balances"".);
    Balance=0;
  }

  *Set_Current_LayerLayer1*:
  Drive_Light("".""E2_Energy_ref.Energy_Reference.Energy_Ref_Light""."".1");
  Drive_Light("".""E1_Energy_ref.Energy_Reference.Energy_Ref_Light"".""2");
  Drive_Light("".""E3_Energy_ref.Energy_Reference.Energy_Ref_Light"".""3");
  Drive_Light("".""E4_Energy_ref.Energy_Reference.Energy_Ref_Light"".""0");

  if (Fuel_Zoom==1)
  {
    Hide_Framer("".""Big_Viewport_CRT"".""Fuel_Zoomed"".);
    Fuel_Zoom=0;
    Hide_Framer("".""Tank_Light_2_Test"".);
  }

  if (Fuel_System==1)
  {
    Hide_Framer("".""Big_Viewport_CRT"".""Fuel_system"".);
    Fuel_System=0;
    Hide_Framer("".""Tank_Light_2_Test"".);
  }

  if (Engine_Zoom==0)
  {
    Appear_Framer("".""Big_Viewport_CRT"".""Engine_Zoomed"".);
    Scale_Framer("".""Big_Viewport_CRT"".""Engine_Zoomed"".""1,1.5");
    Engine_Zoom=1;
  }

  *Set_Current_LayerLayer1*:
  pots=1043;
pots=443;
Drive_Light("".""Pot_Lights_1"".""1");
Drive_Light("".""Pot_Lights_2"".""2");

Channel_Member_Put_Buffer("Session_Common".""."".""Location"".""MEMBER_DIMEN(1),12.1.CHAN_FLOAT.&pots,1.CHANNEL_WRITE_Ei);

Channel_Member_Put_Buffer("Session_Common".""."".""Location"".""MEMBER_DIMEN(1),13.1.CHAN_FLOAT.&pots,1.CHANNEL_WRITE_Ei);
Drive_Pott:"vertical_1",potsy;
Drive_Pott:"horizontal_1",potsy;
Drive_Light:"Labels",3;
} -> Monitor
(SWITCH Engine_Selector? code=?)
{
   Appear_Frame":"Energy_Ent_Plots2";
   Drive_Switch":"Fuel_Overview_Sw",1;
   Drive_Switch":"Engine_Selector1",14;
   Drive_Switch":"Engine_Selector2",14;
   Drive_Switch":"Engine_Selector3",14;
   Drive_Switch":"Engine_Selector4",14;
   Drive_Light":"Pot_Lights",2;
   Drive_Switch":"E2_Energy_ref.Energy_Reference.Hs_Plot_Ref_Hp_s_plot.Energy_Select_sw",8;
   Drive_Switch":"E1_Energy_ref.Energy_Reference.Hs_Plot_Ref_Hp_s_plot.Energy_Select_sw",8;
   Drive_Switch":"E3_Energy_ref.Energy_Reference.Hs_Plot_Ref_Hp_s_plot.Energy_Select_sw",8;
   Drive_Switch":"E4_Energy_ref.Energy_Reference.Hs_Plot_Ref_Hp_s_plot.Energy_Select_sw",8;

   if (Balance=1)
   {
      Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
      Balance=0;
   }
   "Set_Current_Layer(Layer1)";

   if (Fuel_Zoom=1)
   {
      Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
      Fuel_Zoom=0;
      Drive_Frame":"Tank_Light_2_Test";
   }
   "Set_Current_Layer(Layer1)";
   Drive_Light":"Fuel_Zoomed","Fuel_Zoomed";
   Fuel_System=0;
   Fuel_System=1;
   Drive_Frame":"Tank_Light_2_Test";

   if (Engine_Zoom=0)
   {
      "Set_Current_Layer(Layer1)";
      Drive_Frame":"Big_Viewport_CRT","Engine_Zoomed";
      Engine_Zoom=1;
      "Set_Current_Layer(Layer1)";
      ptxt=156;
      ptyt=443;
      Drive_Light":"Pot_Lights","Pot_Lights";
   }

   Channel_Member_Put_Buffer("Session_Common","","Location",MEMBER_DIMEN(11,12,1.CHAN_FLOAT, &ptxt,1.CHANNEL_WRITE);

   Channel_Member_Put_Buffer("Session_Common","","Location",MEMBER_DIMEN(13,13,1.CHAN_FLOAT, &ptxt,1.CHANNEL_WRITE);

   Drive_Pott:"vertical_1",potsy;
   Drive_Pott:"horizontal_1",potsy;
   Drive_Light:"Labels",3;
} -> Monitor
(SWITCH Engine_Selector? code=4)
{
   Appear_Frame":"Energy_Ent_Plots2";
   Drive_Switch":"Fuel_Overview_Sw",1;
   Drive_Switch":"Engine_Selector1",14;
   Drive_Switch":"Engine_Selector2",14;
Drive_Switch("".""Engine_Selector4""."",14);
Drive_Light("".""Pot_Lights""."",2);
Drive_Switch("".""E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw""."",8);
Drive_Switch("".""E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw""."",8);
Drive_Switch("".""E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw""."",8);
Drive_Switch("".""E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw""."",8);

if (Balance==11){
  Hide_Frame(""Big_Viewport_CRT"".""Fuel_Balances"".);
  Balance=0;
}

*Set_Current_LayerLayer1*;
Drive_Light("".""E2_Energy_ref.Energy_Reference.Energy_Ref_Light""."",1);
Drive_Light("".""E1_Energy_ref.Energy_Reference.Energy_Ref_Light""."",2);
Drive_Light("".""E2_Energy_ref.Energy_Reference.Energy_Ref_Light""."",5);
Drive_Light("".""E4_Energy_ref.Energy_Reference.Energy_Ref_Light""."",0);

if (Fuel_Zoom==111){
  Hide_Frame(""Big_Viewport_CRT"".""Fuel_Zoomed"".);
  Fuel_Zoom=0;
  Hide_Frame("".""Tank_Light_2_Test"".);
}
if (Fuel_System==11){
  Hide_Frame(""Big_Viewport_CRT"".""Fuel_system"".);
  Fuel_System=0;
  Hide_Frame("".""Tank_Light_2_Test"".);
}
if (Engine_Zoom==00){
  Appear_Frame(""Big_Viewport_CRT"".""Engine_Zoomed"".);
  Scale_Frame(""Big_Viewport_CRT"".""Engine_Zoomed"".1.1.5);
  Engine_Zoom=11;
}

*Set_Current_LayerLayer1*;
posx=199;
posy=1070;
Drive_Light("".""Pot_Lights""."",1.1);

Channel_Member_Put_Buffert""Session_Common""."".""."".""Location"".MEMBER_DIMEN(1),12,1.CHAN_FLOAT,&posx.1.CHANNEL_WRITE_E;

Channel_Member_Put_Buffert""Session_Common""."".""."".""Location"".MEMBER_DIMEN(1),13,1.CHAN_FLOAT,&posx.1.CHANNEL_WRITE_E;

Drive_Pot("".""Vertical_1""."",posy);
Drive_Pot("".""Horizontal_1""."",posx);
Drive_Light("".""Labels""."",3);
}

}

} -> Monitor

(SWITCH Engine_Selector2 'code5')

{
  Appear_Frame("".""Energy_Ent_Plots2"".);
  Drive_Switch("".""Fuel_Overview_Sw""."",1);
  Drive_Switch("".""Engine_Selector1""."",14);
  Drive_Switch("".""Engine_Selector3""."",14);
  Drive_Switch("".""Engine_Selector4""."",14);
  Drive_Light("".""Pot_Lights""."",2);
  Drive_Switch("".""E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw""."",8);
  Drive_Switch("".""E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw""."",8);
  Drive_Switch("".""E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw""."",8);
  Drive_Switch("".""E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw""."",8);

  if (Balance==11){
    Hide_Frame(""Big_Viewport_CRT"".""Fuel_Balances"".);
    Balance=0;
  }
}
if (Fuel_Zoom=1);
{
    Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
    Fuel_Zoom=0;
    Hide_Frame("Tank_Light_2_Test");
}
if (Fuel_System=1);
{
    Hide_Frame("Big_Viewport_CRT","Fuel_system");
    Fuel_System=0;
    Hide_Frame("Tank_Light_2_Test");
}
if (Engine_Zoom==0);
{
    Appear_Frame("Big_Viewport_CRT","Engine_Zoomed");
    Scale_Frame("Big_Viewport_CRT","Engine_Zoomed",1.1,1.5);
    Engine_Zoom=1;
}
"Set_Current_Layer(Layer1)",
potx=665;
poty=0;
Drive_Light("Pot_Lights","1"");
Channel_Member_Put_Buffer("Session_Common","","Location",MEMBER_DIM(1.12.1.CHAN_FLOAT,\potx,1.CHANNEL_WRITE));
Channel_Member_Put_Buffer("Session_Common","","Location",MEMBER_DIM(1.13.1.CHAN_FLOAT,\poty,1.CHANNEL_WRITE));
Drive_Pot("vertical",\poty);
Drive_Pot("horizontal",\potx);
Drive_Light("Labels","3");
} > Monitor
(SWITCH Engine_Selector2 'code6')
{
    Appear_Frame("Energy_Env_Plots2");
    Drive_Switch("Fuel_Overview_Sw",1);
    Drive_Switch("Engine_Selector1",14);
    Drive_Switch("Engine_Selector2",14);
    Drive_Switch("Engine_Selector4",14);
    Drive_Light("Pot_Lights",2);
    Drive_Switch("E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch("E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch("E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch("E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
}
if (Balance==1)
{
    Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
    Balance=0;
}
"Set_Current_Layer(Layer1)":
Drive_Light("E2_Energy_ref.Energy_Reference.Energy_Ref_Light",1);
Drive_Light("E1_Energy_ref.Energy_Reference.Energy_Ref_Light",2);
Drive_Light("E3_Energy_ref.Energy_Reference.Energy_Ref_Light",3);
Drive_Light("E4_Energy_ref.Energy_Reference.Energy_Ref_Light",0);
if (Fuel_Zoom=1);
{
    Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
}
Fuel_Zoom=0;
Hide_Frame("".;"Tank_Light_2_Test");
}
if (Fuel_System=1):
{
    Hide_Frame("Big_Viewport_CRT","Fuel_system");
    Fuel_System=0;
    Hide_Frame("".;"Tank_Light_2_Test");
}
if (Engine_Zoom=0):
{
    
    Appear_Frame("Big_Viewport_CRT","Engine_Zoomed");
    Scale_Frame("Big_Viewport_CRT","Engine_Zoomed",1,1,5);
    Engine_Zoom=1;
}
*Set_Current_Layer(Layer1)*;
ptx=448;
potx=441;
Drive_Light("".;"Pot_Lights_1",1);
Channel_Member_Put_Buffer("Session_Common","".;"Location",MEMBER_DIMEN(1,12,1.CHAN_FLOAT,&ptx,1.CHANNEL_WRITE));

Channel_Member_Put_Buffer("Session_Common","".;"Location",MEMBER_DIMEN(1,13,1.CHAN_FLOAT,&potx,1.CHANNEL_WRITE));

Drive_Pot("".;"vertical_1",potx);
Drive_Pot("".;"horizontal_1",ptx);
Drive_Light("".;"Labels",3);
} -> Monitor
(SWITCH Engine_Selector2 'code?')
{
    Appear_Frame("".;"Energy_ENT_Plots1");
    Drive_Switch("".;"Fuel_Overview_Sw",1);
    Drive_Switch("".;"Engine_Selector1",14);
    Drive_Switch("".;"Engine_Selector3",14);
    Drive_Switch("".;"Engine_Selector4",14);
    Drive_Light("".;"Pot_Lights",13);
    Drive_Switch("".;"E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_x.plot.Energy_Select_sw",81);
    Drive_Switch("".;"E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_x.plot.Energy_Select_sw",81);
    Drive_Switch("".;"E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_x.plot.Energy_Select_sw",81);
    Drive_Switch("".;"E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_x.plot.Energy_Select_sw",81);
    if (Balance=1){
        Hide_Frame("Big_Viewport_CRT","Fuel_Balance");
        Balance=0;
    }
    *Set_Current_Layer(Layer1)*;
    Drive_Light("".;"E2_Energy_ref.Energy_Reference.Energy_Ref_Light",1);
    Drive_Light("".;"E1_Energy_ref.Energy_Reference.Energy_Ref_Light",2);
    Drive_Light("".;"E3_Energy_ref.Energy_Reference.Energy_Ref_Light",3);
    Drive_Light("".;"E4_Energy_ref.Energy_Reference.Energy_Ref_Light",0);
}
if (Fuel_Zoom=1):
{
    Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
    Fuel_Zoom=0;
    Hide_Frame("".;"Tank_Light_2_Test");
}
if (Fuel_System=1):
{
    Hide_Frame("Big_Viewport_CRT","Fuel_system");
    Fuel_System=0;
    Hide_Frame("".;"Tank_Light_2_Test");
}
if (Engine_Zoom=0):
{
Drive_Pot"vertical_1".pqty 1;
Drive_Pot"horizontal_1".pqty 1;

Drive_Light"Labels".3;
} -> Monitor
(SWITCH Engine_Selector2 "code")
(
Appear_Frame"Energy_Ent_Plots2";
Drive_Switch"Fuel_Overview_Sw".1;
Drive_Switch"Engine_Selector1".14;
Drive_Switch"Engine_Selector3".14;
Drive_Switch"Engine_Selector4".14;
Drive_Light"Pot_Lights".21;

if (Balance==1)
Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
Balance=0;
}

"Set_Current_LayerLayer1":;
Drive_Light"E3_Energy_ref.Energy_Reference.Energy_Ref.Light",31;
Drive_Light"E4_Energy_ref.Energy_Reference.Energy_Ref.Light",01;

if (Fuel_Zoom==1);
{
Hide_Frame("Big_Viewport_CRT","Fuel_Zoommed");
Fuel_Zoom=0;
Hide_Frame("Tank_Light_2 Test");
}
if (Fuel_System==1);
{
Hide_Frame("Big_Viewport_CRT","Fuel_System");
Fuel_System=0;
Hide_Frame("Tank_Light_2 Test");
}
if (Engine_Zoom==0);
{
Appear_Frame("Big_Viewport_CRT","Engine_Zoomed");
Scale_Frame("Big_Viewport_CRT","Engine_Zoomed",1.1.51;
Engine_Zoom=1;
}
"Set_Current_LayerLayer1":
pqty=0;
pqty=0.87;
Drive_Light"Pot_Lights_1",1;

Channel_Member_Put_Buffer"Session_Common","","Location",MEMBER_DIMEN(1),12,1.CHAN_FLOAT, &pqty,1.CHANNEL_WRITE;

Channel_Member_Put_Buffer"Session_Common","","Location",MEMBER_DIMEN(1),13,1.CHAN_FLOAT, &pqty,1.CHANNEL_WRITE;

Drive_Pot"vertical_1",pqty;
Drive_Pot"horizontal_1",pqty;
Drive_Light("","Labels",3);
) -> Monitor
(SWITCH Engine_Selector2 'code10')
{
  Appear_Frame("","Energy_Entity_Plots2");
  Drive_Switch("","Fuel_Overview_Sw",1);
  Drive_Switch("","Engine_Selector1",14);
  Drive_Switch("","Engine_Selector2",14);
  Drive_Switch("","Engine_Selector3",14);
  Drive_Switch("","Engine_Selector4",14);
  Drive_Switch("","Energy_Reference_Hs_Plot_Ref_Hp_s_plots_Energy_Select_sw",8);
  Drive_Switch("","E1_Energy_ref_Energy_Reference_Hs_Plot_Ref_Hp_s_plot_Energy_Select_sw",8);
  Drive_Switch("","E2_Energy_ref_Energy_Reference_Hs_Plot_Ref_Hp_s_plot_Energy_Select_sw",8);
  Drive_Switch("","E4_Energy_ref_Energy_Reference_Hs_Plot_Ref_Hp_s_plot_Energy_Select_sw",8);

  if (Balance>=1){
    Hide_Frame("Big_Viewport_CRT","Fuel_Balance");
    Balance=0;
    "Set_Current_Layer(Layer1)"
  }
  Drive_Light("","E2_Energy_ref_Energy_Reference_Energy_Ref_Light",1);
  Drive_Light("","E1_Energy_ref_Energy_Reference_Energy_Ref_Light",2);
  Drive_Light("","E3_Energy_ref_Energy_Reference_Energy_Ref_Light",3);
  Drive_Light("","E4_Energy_ref_Energy_Reference_Energy_Ref_Light",0);

  if (Fuel_Zoom=1){
    Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
    Fuel_Zoom=1;
    "Tank_Light_2_Test"
  }
  if (Fuel_System=1){
    "Fuel_System"
  }
  if (Engine_Zoom=0){
    "Tank_Light_2_Test"
  }
  if (Engine_Zoom=0){
    Appear_Frame("Big_Viewport_CRT","Engine_Zoomed");
    Scale_Frame("Big_Viewport_CRT","Engine_Zoomed",1.1,1.5);
    Engine_Zoom=1;
    "Set_Current_Layer(Layer1)"
    "potx=300;
    poty=500;
    Drive_Light("","Pot_Lights_1",1);
  }

  Channel_Member_Put_Buffer("Session_Common","","Location",MEMBER_DIMEN(1,12.1.CHAN_FLOAT,&poty.1.CHANNEL_WRITE));
  Channel_Member_Put_Buffer("Session_Common","","Location",MEMBER_DIMEN(1,13.1.CHAN_FLOAT,&poty.1.CHANNEL_WRITE));

  Drive_Pott("","vertical_1",poty);
  Drive_Pott("","horizontal_1",poty);

  Drive_Light("","Labels",3);
) -> Monitor
(SWITCH Engine_Selector2 'code11')
{
  Appear_Frame("","Energy_Entity_Plots2");
  Drive_Switch("","Fuel_Overview_Sw",1);
  Drive_Switch("","Engine_Selector1",14);
  Drive_Switch("","Engine_Selector2",14);
  Drive_Switch("","Engine_Selector3",14);
  Drive_Switch("","Engine_Selector4",14);
  Drive_Light("","Pot_Lights",2);
Drive_Switch="E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hs_plot.Energy_Select_sw".8;
Drive_Switch="E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hs_plot.Energy_Select_sw".8;
Drive_Switch="E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hs_plot.Energy_Select_sw".8;
Drive_Switch="E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hs_plot.Energy_Select_sw".8;

if (Balance==1)
  
  Hide_Frame("Big_Viewport_CRT"."Fuel_Balances");
  Balance=0;
  
  *Set_Current_LayerLayer1*;
  
  Drive_Light="E2_Energy_ref.Energy_Reference.Energy_Ref_Light".1;
  Drive_Light="E1_Energy_ref.Energy_Reference.Energy_Ref_Light".2;
  Drive_Light="E3_Energy_ref.Energy_Reference.Energy_Ref_Light".3;
  Drive_Light="E4_Energy_ref.Energy_Reference.Energy_Ref_Light".0;
  
  if (Fuel_Zoom==1):

    { Hide_Frame("Big_Viewport_CRT"."Fuel_Zoomed");
      Fuel_Zoom=0;
      Hide_Frame(""."Tank_Light.2.Test");
    }
    
    if (Fuel_System==1):

      { Hide_Frame("Big_Viewport_CRT"."Fuel_System");
        Fuel_System=0;
        Hide_Frame(""."Tank_Light.2.Test");
      }
    
    if (Engine_Zoom==0):

      { Appear_Frame("Big_Viewport_CRT"."Engine_Zoomed");
        Scale_Frame("Big_Viewport_CRT"."Engine_Zoomed",1.15);
        Engine_Zoom=1;
      }
  
  *Set_Current_LayerLayer1*;
  ptxt=500;
  ptxt=500;
  Drive_Light="Pot_Lights.1",.1;
Drive_Light"".""E2.Energy_ref.Energy_Reference.Energy_Ref_Light""1;  
Drive_Light"".""E4.Energy_ref.Energy_Reference.Energy_Ref_Light""0;  

if (Fuel_Zoom=1) {  
  
  if (Fuel_System=1) {  
    
    if (Engine_Zoom==0) {  
      
      if (Engine_Zoom==0) {  
        
        Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");  
        Scale_Frame("Big_Viewport_CRT","Engine_Zoomed",1.1,1);  
        
        Set_Current_Layer(1);  
        ptxt=300;  
        pty=500;  
        Drive_Light"".""Pot_Lights_1",1,1;  
        
        Channel_Member.Put_Buffer("Session_Common",""",""Location",MEMBER_DIMEN11,12,1,CHAN_FLOAT.&ptxt,1,CHANNEL_WRITE);  
        
        Channel_Member.Put_Buffer("Session_Common",""",""Location",MEMBER_DIMEN11,13,1,CHAN_FLOAT.&ptxt,1,CHANNEL_WRITE);  
        
        Drive_Pot"".""vertical",1,ptxt;  
        Drive_Pot"".""horizontal",1,ptxt;  
        Drive_Light"".""Labels",2;  
      }  
    }  
  }  
}  

} -> Monitor  
(SWITCH Engine_Selector2'code13')  
  
  if (Switch2"".""Energy_Ent_Plots2""14);  
  Drive_Switch"".""Engine_Selector2",14;  
  Drive_Light"".""Pot_Lights",2;  
  Drive_Light"".""Pot_Lights_1",2;  
  Drive_Switch"".""Fuel_Overview_Sw",1;  

if (Engine_Zoom==1) {  
  
  if (Engine_Zoom==0) {  
    
    if (Balance==1) {  
      Hide_Frame("Big_Viewport_CRT","Fuel_Balances");  
      Balance=0;  
      Hide_Frame("".""Tank_Light_2_Test");  
    }  
  }  
}  

if (Fuel_Zoom==1) {  
  Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");  
}
Fuel_Zoom=0;
}
if(Fuel_System==11)
  { Hide_Frame("Big_Viewport_CRT","Fuel_system"); Fuel_System=0;
  }
  *Set_Current_Layer(Layer1)*;
  Drive_Light("E1_Energy_ref.Energy_Reference.Energy_Ref_Light",1);
  Drive_Light("E2_Energy_ref.Energy_Reference.Energy_Ref_Light",2);
  Drive_Light("E3_Energy_ref.Energy_Reference.Energy_Ref_Light",3);
  Drive_Light("E4_Energy_ref.Energy_Reference.Energy_Ref_Light",0);
  Drive_Light("Labels",3);
} -> Monitor
{
  Appear_Frame("Energy_Ent_Plots2");
  Drive_Switch("Fuel_Overview_Sw",1);
  Drive_Switch("Engine_Selector2",144);
  Drive_Switch("Engine_Selector1",144);
  Drive_Switch("Engine_Selector3",144);
  Drive_Switch("Engine_Selector4",144);
  Drive_Switch("E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
  Drive_Switch("E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
  Drive_Switch("E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
  Drive_Switch("E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);

  Drive_Light("Pot_Lights",2);
  Drive_Light("Pot_Lights",1 1);

  if(Balance==11)
    { Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
      Balance=0;
    }
  if(Fuel_Zoom=1)
    { Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
      Fuel_Zoom=0;
      Hide_Frame("Tank_Light_2_Test");
    }
  if(Fuel_System=1)
    { Hide_Frame("Big_Viewport_CRT","Fuel_system");
      Fuel_System=0;
      Hide_Frame("Tank_Light_2_Test");
    }
  if(Engine_Zoom==0)
    { Appear_Frame("Big_Viewport_CRT","Engine_Zoomed");
      Scale_Frame("Big_Viewport_CRT","Engine_Zoomed",1.1,1.5);
      Engine_Zoom=1;
    }
  *Set_Current_Layer(Layer2)*:
  potx=226;
  poty=444;
  Drive_Light("Pot_Lights",1,1);

  Channel_Member_Put_Buffer("Session_Common","Location",MEMBER_DIMEN(1.2.1.CHAN_FLOAT&potx.1.CHANNEL_WRITE_E);

  Channel_Member_Put_Buffer("Session_Common","Location",MEMBER_DIMEN(1.3.1.CHAN_FLOAT&poty.1.CHANNEL_WRITE_E);

  Drive_Pott("vertical",1,poty);
  Drive_Pott("horizontal",1,potx);
Drive_Light"."Labels".2);

Drive_Light"."E2_Energy_ref.Energy_Reference.Energy_Ref.Light".1);
Drive_Light"."E1_Energy_ref.Energy_Reference.Energy_Ref.Light".2);
Drive_Light"."E3_Energy_ref.Energy_Reference.Energy_Ref.Light".3);
Drive_Light"."E4_Energy_ref.Energy_Reference.Energy_Ref.Light".0);

} -> Monitor
{
    Drive_Switch"."Fuel_Overview_Sw".".1);
    Drive_Switch"."Engine_Selector2".".14);
    Drive_Switch"."Engine_Selector1".".14);
    Drive_Switch"."Engine_Selector".".14);
    Drive_Switch"."Engine_Selector4".".14);
    Drive_Switch"."E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw".".8);
    Drive_Switch"."E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw".".8);
    Drive_Switch"."E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw".".8);

    Drive_Light"."Pot_Lights".2);

    if (Balance==1)
    {
        Hide_Frame"Big_Viewport_CRT"."Fuel_Balances"
        Balance=0;
    }

    *Set_Current_LayerLayer1*;
    Drive_Light"."E2_Energy_ref.Energy_Reference.Energy_Ref.Light".1);
    Drive_Light"."E1_Energy_ref.Energy_Reference.Energy_Ref.Light".2);
    Drive_Light"."E3_Energy_ref.Energy_Reference.Energy_Ref.Light".3);
    Drive_Light"."E4_Energy_ref.Energy_Reference.Energy_Ref.Light".0);

    if (Fuel_Zoom==1)
    {
        Hide_Frame"Big_Viewport_CRT"."Fuel_Zoomed"
        Fuel_Zoom=0;
        Hide_Frame"".Tank_Light_2_Test"
    }

    if (Fuel_System==1)
    {
        Hide_Frame"Big_Viewport_CRT"."Fuel_system"
        Fuel_System=0;
        Hide_Frame"".Tank_Light_2_Test"
    }

    if (Engine_Zoom==0)
    {
        Appear_Frame"Big_Viewport_CRT"."Engine_Zoomed"
        Scale_Frame"Big_Viewport_CRT"."Engine_Zoomed".1.1.5)
        Engine_Zoom=1;
    }

    *Set_Current_LayerLayer1*;
    potx=226;
    potty=444;
    Drive_Light"."Pot_Lights".1",.1;

Channel_Member_Put BUFFERT"Session_Common",".""Location",MEMBER_DIMEN(1),12.1.CHAN_FLOAT,&potx1.CHANNEL_WRITE;

Channel_Member_Put BUFFERT"Session_Common",",",""Location",MEMBER_DIMEN(1),13.1.CHAN_FLOAT,&potty1.CHANNEL_WRITE;

    Drive_Pot"."vertical_1".potx;
    Drive_Pot"."horizontal_1".potx;
    Drive_Light"."Labels".3);

} -> Monitor
Drive Switch:"E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);

Drive Light:"Pot_Lights",2);

if (Balance==1){
  Hide Frame:"Big_Viewport_CRT","Fuel_Balances";
  Balance=0;
}

"Set Current Layer[Layer1]*:
Drive Light:"E2_Energy_ref.Energy_Reference.Energy_Ref_Light",1;
Drive Light:"E1_Energy_ref.Energy_Reference.Energy_Ref_Light",2;
Drive Light:"E3_Energy_ref.Energy_Reference.Energy_Ref_Light",3;
Drive Light:"E4_Energy_ref.Energy_Reference.Energy_Ref_Light",0;

if (Fuel_Zoom==1);
{  
  Hide Frame:"Big_Viewport_CRT","Fuel_Zoomed";
  Fuel_Zoom=0;
  Hide Frame:"Tank_Light_2_Test";
}
if (Fuel_System==1);
{
  Hide Frame:"Big_Viewport_CRT","Fuel_System";
  Fuel_System=0;
  Hide Frame:"Tank_Light_2_Test";
}
if (Engine_Zoom==0);
{
  Appear Frame:"Big_Viewport_CRT","Engine_Zoomed";
  Scale Frame:"Big_Viewport_CRT","Engine_Zoomed",1.1.5;
  Engine_Zoom=1;
}

"Set Current Layer[Layer1]*:
pots=1043;
potsy=443;
Drive Light:"Pot_Lights",1);

Channel Member Put Buffet"Session Common",","Location",MEMBER_DIMEN[12,12,1.CHAN_FLOAT,"pots",1.CHANNEL_WRITE);

Channel Member Put Buffet"Session Common",","Location",MEMBER_DIMEN[13,13,1.CHAN_FLOAT,"pots",1.CHANNEL_WRITE);

Drive Pot","vertical_1","potsy;
Drive Pot","horizontal_1","potsy;

Drive Light","Labels",3);
)
-> Monitor
(SWITCH E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw '4-5')
{
  Drive Switch","Fuel_Overview_Sw",1;
  Drive Switch","Engine_Selec1",14;
  Drive Switch","Engine_Selec1",14;
  Drive Switch","Engine_Selec1",14;
  Drive Switch","Engine_Selec1",14;
  Drive Switch","E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8;
  Drive Switch","E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8;

  Drive Light","Pot_Lights",2);

if (Fuel_Zoom==1):
    {  
      Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");  
      Fuel_Zoom=0;  
      Hide_Frame("Tank_Light","Fuel_Zoomed1");  
    }
  
if (Fuel_System==1):
    {  
      Hide_Frame("Big_Viewport_CRT","Fuel_system");  
      Fuel_System=0;  
      Hide_Frame("Tank_Light","Fuel_Zoomed1");  
    }
  
if (Engine_Zoom==0):
    {  
      Appear_Frame("Big_Viewport_CRT","Engine_Zoomed");  
      Scale_Frame("Big_Viewport_CRT","Engine_Zoomed",1.1.5);  
      Engine_Zoom=1;  
    }
  
  *Set_Current_LayerLayer1*:
    
  pots=226;
  poty=444;
  Drive_Light("Pot_Lights","Fuel_Zoomed1");

Channel_Member_Put_Buffer("Session_Common","Location",MEMBER_DIM(1,12.1.CHAN_FLOAT,pots,1.CHANNEL_WRITE);

Channel_Member_Put_Buffer("Session_Common","Location",MEMBER_DIM(1,13.1.CHAN_FLOAT,poty,1.CHANNEL_WRITE);

  Drive_Putt("Ref.Overview_Sw",vertical_1, pots);  
  Drive_Putt("Ref.Overview_Sw",horizontal_1, pots);  
  Drive_Light("Ref.Overview_Sw",Labels_3);

} -> Monitor

(SWITCH E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw '6-0')

  
  Drive_Switch("Fuel_Overview_Sw",1);
  Drive_Switch("Engine_Selector",0.1,14);
  Drive_Switch("Engine_Selector")
  
  Drive_Switch("Ref.Overview_Sw",Engine_Selector4,14);
  Drive_Switch("Ref.Overview_Sw",Engine_Selector2,14);
  Drive_Switch("Ref.Overview_Sw",Engine_Selector1,14);
  Drive_Switch("Ref.Overview_Sw",Engine_Selector4,14);
  Drive_Switch("Ref.Overview_Sw",Engine_Selector2,14);
  Drive_Switch("Ref.Overview_Sw",Engine_Selector1,14);
  Drive_Switch("Ref.Overview_Sw",Engine_Selector4,14);
  Drive_Switch("Ref.Overview_Sw",Engine_Selector2,14);
  Drive_Switch("Ref.Overview_Sw",Engine_Selector1,14);
  Drive_Switch("Ref.Overview_Sw",Engine_Selector4,14);
  Drive_Switch("Ref.Overview_Sw",Engine_Selector2,14);

Drive_Light("Pot_Lights",2);

if (Balance==1){
    Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
    Balance=0;
  }

  *Set_Current_LayerLayer1*;
  Drive_Light("E2_Energy_ref.Energy_Reference.Energy_Ref_Light",1);
  Drive_Light("E1_Energy_ref.Energy_Reference.Energy_Ref_Light",2);
  Drive_Light("E3_Energy_ref.Energy_Reference.Energy_Ref_Light",3);
  Drive_Light("E4_Energy_ref.Energy_Reference.Energy_Ref_Light",0);

if (Fuel_Zoom==1):
    {  
      Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");  
      Fuel_Zoom=0;  
    }


Fuel_System=0;
Hide_Frame("".""Tank_Light_2_Test""."");
}
if (Engine_Zoom==0);
{
  Appear_Frame("Big_Viewport_CRT"."Engine_Zoomed"."");
  Scale_Frame("Big_Viewport_CRT"."Engine_Zoomed".1.1.5);
  Engine_Zoom=1;
}
*Set_Current_Layer(Layer1)*;
pots=1043;
puty=443;
Drive_Light("".""Pot_Lights_1",1."");

Channel_Member_Put_Buffer("Session_Common","".""Location",MEMBER_DIMEN(1),12,1.CHAN_FLOAT.&potx,1.CHANNEL_WRITE1);

Channel_Member_Put_Buffer("Session_Common","".""Location",MEMBER_DIMEN(1),13,1.CHAN_FLOAT.&poty,1.CHANNEL_WRITE1);

Drive_Pot("".""vertical_1",poty);
Drive_Pot("".""horizontal_1",potx);
Drive_Light("".""Labels",4);
}
}
-> Monitor
(SWITCH Engine_Selector; 'code2')
{
  Appear_Frame("".""Energy_Env_Plots2""."");
  Drive_Switch("".""Fuel_Overview_Sw",1."");
  Drive_Switch("".""Engine_Selector1",14."");
  Drive_Switch("".""Engine_Selector2",14."");
  Drive_Switch("".""Engine_Selector4",14."");
  Drive_Switch("".""E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8."");
  Drive_Switch("".""E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8."");
  Drive_Switch("".""E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8."");
  Drive_Switch("".""E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8."");
  if (Balance==1){
    Hide_Frame("Big_Viewport_CRT"."Fuel_Balances"."");
    Balance=0;
  }
  *Set_Current_Layer(Layer1)*;
  Drive_Light("".""E3_Energy_ref.Energy_Reference.Energy_Ref.Light",1."");
  Drive_Light("".""E1_Energy_ref.Energy_Reference.Energy_Ref.Light",3."");
  Drive_Light("".""E2_Energy_ref.Energy_Reference.Energy_Ref.Light",0."");
  if (Fuel_Zoom==1);
  {
    Hide_Frame("Big_Viewport_CRT"."Fuel_Zoomed"."");
    Fuel_Zoom=0;
  }
  Hide_Frame("".""Tank_Light_2_Test""."");
  if (Fuel_System==1);
  {
    Hide_Frame("Big_Viewport_CRT"."Fuel_system"."");
    Fuel_System=0;
  }
  Hide_Frame("".""Tank_Light_2_Test""."");
  if (Engine_Zoom==0);
  {
    Appear_Frame("Big_Viewport_CRT"."Engine_Zoomed"."");
    Scale_Frame("Big_Viewport_CRT"."Engine_Zoomed".1.1.5);
    Engine_Zoom=1;
  }
  *Set_Current_Layer(Layer1)*;
pots=1043;
poty=443;
    Drive_Light("".""Pot_Lights_1""."",1);
    Drive_Light("".""Pot_Lights""."",2);

Channel_Member_Put_Buffer("Session_Common","".""Location""."",MEMBER_DIMEN(1,1).CHAN_FLOAT.&poty.1.CHANNEL_WRITE);

Channel_Member_Put_Buffer("Session_Common","".""Location""."",MEMBER_DIMEN(1,1,1).CHAN_FLOAT.&poty.1.CHANNEL_WRITE);

    Drive_Pot("".""vertical_1""."",poty);
    Drive_Pot("".""horizontal_1""."",potx);
    Drive_Light("".""Labels""."",4);
}

}> Monitor
    (SWITCH Engine_Selector1 code3')
    {
        Appear_Frame("".""Energy_Ent_Plots2""."";
        Drive_Switch("".""Fuel_Overview_Sw""."",1);
        Drive_Switch("".""Engine_Selector1""."",14);
        Drive_Switch("".""Engine_Selector2""."",14);
        Drive_Switch("".""Engine_Selector4""."",14);
        Drive_Light("".""Pot_Lights""."",1);
        Drive_Switch("".""E1_Energy_unit.Energy_Reference.Hs_Plot_Ref.Hp.s_plot.Energy.Select_sw""."",8);
        Drive_Switch("".""E1_Energy_unit.Energy_Reference.Hs_Plot_Ref.Hp.s_plot.Energy.Select_sw""."",8);
        Drive_Switch("".""E2_Energy_unit.Energy_Reference.Hs_Plot_Ref.Hp.s_plot.Energy.Select_sw""."",8);
        Drive_Switch("".""E4_Energy_unit.Energy_Reference.Hs_Plot_Ref.Hp.s_plot.Energy.Select_sw""."",8);

        if (Balance==1)
            {
                Hide_Frame("".""Big_Viewport_CRT""."".""Fuel_Balances""."";
                Balance=0;
            }

        } Set_Current_LayerLayer1*;
        Drive_Light("".""E3_Energy_unit.Energy_Reference.Energy_Ref.Light""."",1);
        Drive_Light("".""E1_Energy_unit.Energy_Reference.Energy_Ref.Light""."",1);
        Drive_Light("".""E2_Energy_unit.Energy_Reference.Energy_Ref.Light""."",1);
        Drive_Light("".""E4_Energy_unit.Energy_Reference.Energy_Ref.Light""."",2);

        if (Fuel_Zoom=1);
            {
                Hide_Frame("".""Big_Viewport_CRT""."".""Fuel_Zoomed""."";
                Fuel_Zoom=0;
                Hide_Frame("".""Tank_Light_2_Test""."";
            }

        if (Fuel_System=1);
            {
                Hide_Frame("".""Big_Viewport_CRT""."".""Fuel_system""."";
                Fuel_System=0;
                Hide_Frame("".""Tank_Light_2_Test""."";
            }

        if (Engine_Zoom=0);
            {
            Appear_Frame("".""Big_Viewport_CRT""."".""Engine_Zoomed""."";
            Scale_Frame("".""Big_Viewport_CRT""."".""Engine_Zoomed""."",1.1.5);
            Engine_Zoom=1;
            }
        } *Set_Current_LayerLayer1*;
        potx=156;
        poty=443;
        Drive_Light("".""Pot_Lights""."",1);

Channel_Member_Put_Buffer("Session_Common","".""Location""."",MEMBER_DIMEN(1,1,1,1).CHAN_FLOAT.&poty.1.CHANNEL_WRITE);

Channel_Member_Put_Buffer("Session_Common","".""Location""."",MEMBER_DIMEN(1,1,1,1,1).CHAN_FLOAT.&poty.1.CHANNEL_WRITE);

    Drive_Pot("".""vertical_1""."",poty);
Drive_Light","Pot_Lights",2);
Drive_Switch","E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
Drive_Switch","E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
Drive_Switch","E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
Drive_Switch","E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);

if (Balance==1){
  Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
  Balance=0;
  }
  *Set_Current_Layer(Layer1)*
  Drive_Light","E3_Energy_ref.Energy_Reference.Energy_Ref_Light",1);
  Drive_Light","E1_Energy_ref.Energy_Reference.Energy_Ref_Light",3);
  Drive_Light","E2_Energy_ref.Energy_Reference.Energy_Ref_Light",0);
  Drive_Light","E4_Energy_ref.Energy_Reference.Energy_Ref_Light",2);

if (Fuel_Zoom==1){
  Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
  Fuel_Zoom=0;
  Hide_Frame("Tank_Light_2_Test");
}
if (Fuel_System==1){
  Hide_Frame("Big_Viewport_CRT","Fuel_system");
  Fuel_System=0;
  Hide_Frame("Tank_Light_2_Test");
}
if (Engine_Zoom==0){
  Appear_Frame("Big_Viewport_CRT","Engine_Zoomed");
  Scale_Frame("Big_Viewport_CRT","Engine_Zoomed",1.2,1.15);
  Engine_Zoom=1;
  *Set_Current_Layer(Layer1)*
  point=665;
  point=0;
  Drive_Light","Pot_Lights",1,1);

Channel_Member_Put_Buffer("Session_Common",","Location",MEMBER_DIMEN(1,12,1),CHAN_FLOAT,&point,1,CHANNEL_WRITE);

Channel_Member_Put_Buffer("Session_Common",","Location",MEMBER_DIMEN(1,13,1),CHAN_FLOAT,&point,1,CHANNEL_WRITE);

  Drive_Point","vertical_1",point);
  Drive_Point","horizontal_1",point);
  Drive_Light","Labels",4);
  } -> Monitor
(SWITCH Engine_Selector3 `code6`)
  {
  Appear_Frame("E1_Energy_En.Plots2");
  Drive_Switch("Fuel_Overview_Sw",1);
  Drive_Switch("Engine_Selector1",1,4);
  Drive_Switch("Engine_Selector2",1,4);
  Drive_Switch("Engine_Selector4",1,4);
  Drive_Light("Pot_Lights",2);
  Drive_Switch("E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
  Drive_Switch("E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
  Drive_Switch("E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
  Drive_Switch("E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);

  if (Balance==1){
    Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
    Balance=0;
    }
*Set_Current_LayerLayer1*:

```
if (Fuel_Zoom==1):
    if (Fuel_Zoom==1):
        Fuel_Zoom0;
        if (Fuel_Zoom==1):
            Fuel_System0;
            Fuel_System1;
            if (Engine_Zoom==0):
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Engine_Zoom1;
                Engine_Zoom0;
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Engine_Zoom1;
                Engine_Zoom0;
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Engine_Zoom1;
                Engine_Zoom0;
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Engine_Zoom1;
                Engine_Zoom0;
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Engine_Zoom1;
                Engine_Zoom0;
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Engine_Zoom1;
                Engine_Zoom0;
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Engine_Zoom1;
                Engine_Zoom0;
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Engine_Zoom1;
                Engine_Zoom0;
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Engine_Zoom1;
                Engine_Zoom0;
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Engine_Zoom1;
                Engine_Zoom0;
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Engine_Zoom1;
                Engine_Zoom0;
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Engine_Zoom1;
                Engine_Zoom0;
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Engine_Zoom1;
                Engine_Zoom0;
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Engine_Zoom1;
                Engine_Zoom0;
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Engine_Zoom1;
                Engine_Zoom0;
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Engine_Zoom1;
                Engine_Zoom0;
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Engine_Zoom1;
                Engine_Zoom0;
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Engine_Zoom1;
                Engine_Zoom0;
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Engine_Zoom1;
                Engine_Zoom0;
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Engine_Zoom1;
                Engine_Zoom0;
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Engine_Zoom1;
                Engine_Zoom0;
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Engine_Zoom1;
                Engine_Zoom0;
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Engine_Zoom1;
                Engine_Zoom0;
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Engine_Zoom1;
                Engine_Zoom0;
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Engine_Zoom1;
                Engine_Zoom0;
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Engine_Zoom1;
                Engine_Zoom0;
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Scale_Frame("Big_Viewport_CRT","Engine_Zoomed");
                Engine_Zoom1;
                Engine_Zoom0;
```
Appear_Frame("Big_Viewport_CRT","Engine_Zoomed");
Scale_Frame("Big_Viewport_CRT","Engine_Zoomed",1.1,5);
Engine_Zoom=1;
*
*Set_Current_Layer(Layer1)::*
potx=619;
poty=0;
Drive_Light""."Pot_Lights_1",1,1;

Channel_Member_Put_Buffer"Session_Common",""."Location",MEMBER_DIM(1),12,1.CHAN_FLOAT,&poty,1.CHANNEL_WRITE;

Channel_Member_Put_Buffer"Session_Common",",","Location",MEMBER_DIM(1),13,1.CHAN_FLOAT,&potx,1.CHANNEL_WRITE;

Drive_Pot""."vertical",85;
Drive_Pot""."horizontal",potx;

Drive_Light""."Labels",14;
} -> Monitor
(SWITCH Engine_Selector3 code 9)
{
  Appear_Frame""."Energy_Ent_Plots2":
  Drive_Switch""."Fuel_Overview_Sw",14;
  Drive_Switch""."Engine_Selector1",14;
  Drive_Switch""."Engine_Selector2",14;
  Drive_Switch""."Engine_Selector3",14;
  Drive_Light""."Pot_Lights",21;
  Drive_Switch""."E1_Energy_ref_Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8;
  Drive_Switch""."E2_Energy_ref_Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8;
  Drive_Switch""."E3_Energy_ref_Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8;
  Drive_Switch""."E4_Energy_ref_Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8;

  if (Balance==1){
    Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
    Balance=0;
  }

  *Set_Current_Layer(Layer1)::*
  Drive_Light""."E1_Energy_ref_Energy_Reference_Energy_Ref_Light",11;
  Drive_Light""."E2_Energy_ref_Energy_Reference_Energy_Ref_Light",11;
  Drive_Light""."E3_Energy_ref_Energy_Reference_Energy_Ref_Light",11;
  Drive_Light""."E4_Energy_ref_Energy_Reference_Energy_Ref_Light",11;
}

if (Fuel_Zoom==1):
{
  Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
  Fuel_Zoom=0;
  Hide_Frame(""."Tank_Light_2_Test" private");
}
if (Fuel_System==1):
{
  Hide_Frame("Big_Viewport_CRT","Fuel system");
  Fuel_System=0;
  Hide_Frame(""."Tank_Light_2_Test" private");
}
if (Engine_Zoom==0):
{
  Appear_Frame("Big_Viewport_CRT","Engine_Zoomed");
  Scale_Frame("Big_Viewport_CRT","Engine_Zoomed",1.1,5);
  Engine_Zoom=1;
}
*Set_Current_Layer(Layer1)::*
potx=0;
poty=867;
Drive_Light""."Pot_Lights_1",1,1;
Drive_Light("".""Labels""."".4;
} -> Monitor
(SWITCH Engine_Selector3{code 12})
{
  Appear_Frame("".""Energy_Ent_Plots2";
  Drive_Switch("".""Fuel_Overview_Sw""."".1;
  Drive_Switch("".""Engine_Selector1""."".14;
  Drive_Switch("".""Engine_Selector2""."".14;
  Drive_Switch("".""Engine_Selector3""."".14;
  Drive_Light("".""Pot_Lights""."".2;
  Drive_Switch("".""E1_Energy_ref.Energy_Reference.Hs_Plot.Ref.Hp_s_plot.Energy_Select_sw""."".8;
  Drive_Switch("".""E1_Energy_ref.Energy_Reference.Hs_Plot.Ref.Hp_s_plot.Energy_Select_sw""."".8;
  Drive_Switch("".""E4_Energy_ref.Energy_Reference.Hs_Plot.Ref.Hp_s_plot.Energy_Select_sw""."".8;

  if (Balance==1)
  { Hide_Frame(""Big_Viewport_CRT"".""Fuel_Balance"";)
      Balance=0;
  }
  *Set_Current_Layer(Layer1)*;
  Drive_Light("".""E1_Energy_ref.Energy_Reference.Energy_Ref_Light""."".1;
  Drive_Light("".""E2_Energy_ref.Energy_Reference.Energy_Ref_Light""."".3;
  Drive_Light("".""E4_Energy_ref.Energy_Reference.Energy_Ref_Light""."".0;
  Drive_Light("".""E4_Energy_ref.Energy_Reference.Energy_Ref_Light""."".2;

  if (Fuel_Zoom==1);
  { Hide_Frame(""Big_Viewport_CRT"".""Fuel_Zoomed"";
      Fuel_Zoom=0;
      Hide_Frame("".""Tank_Light_2.Test"";
  }
  if (Fuel_System==1);
  { Hide_Frame(""Big_Viewport_CRT"".""Fuel_system"";
      Fuel_System=0;
      Hide_Frame("".""Tank_Light_2.Test"";
  }
  if (Engine_Zoom==0);
  { Appear_Frame(""Big_Viewport_CRT"".""Engine_Zoomed"";
      Scale_Frame(""Big_Viewport_CRT"".""Engine_Zoomed""."".1.1.5;
      Engine_Zoom=1;
  }
  *Set_Current_Layer(Layer1)*;
  potx=300;
  potty=500;
  Drive_Light("".""Pot_Lights"".""."".1.1;

Channel_Member_Put_Buffer(""Session_Common""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".""."".”
if (Balance == 1)
    Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
Balance = 0;

*Set_Current_Layer(Layer1)*;
Drive_Light("E1_Energy_Ref_Energy_Reference:E1_Energy_Ref_Energy_Ref_Light", 1);
Drive_Light("E1_Energy_Ref_Energy_Reference:E1_Energy_Ref_Energy_Ref_Light", 3);
Drive_Light("E2_Energy_Ref_Energy_Reference:E2_Energy_Ref_Energy_Ref_Light", 0);
Drive_Light("E4_Energy_Ref_Energy_Reference:E4_Energy_Ref_Energy_Ref_Light", 2);

if (Fuel_Zoom == 1)
    Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
Fuel_Zoom = 0;
Hide_Frame("Tank_Light_2_Test");

if (Fuel_System == 1)
    Hide_Frame("Tank_Light_2_Test");

if (Engine_Zoom == 0)
    *Set_Current_Layer(Layer1)*;
    $pxx = 500;
    $pxy = 500;
    Drive_Light("Put_Lights_1", 1);

Channel_Member_Put_Buffer "Session_Common", "Location", MEMBER_DIMEN(1, 12, 1, CHAN_FLOAT, $pxx, 1, CHANNEL_WRITE_E);

Channel_Member_Put_Buffer "Session_Common", "Location", MEMBER_DIMEN(1, 13, 1, CHAN_FLOAT, $pxy, 1, CHANNEL_WRITE_E);

Drive_Pot("Vertical_1", $pxy);
Drive_Pot("Horizontal_1", $pxx);
Drive_Light("Labels", 4);
}

-> Monitor
SWITCH Engine_SELECTOR 3 (node 13)

{*
    Drive_Light("Energy_Ent_Plots", 2);
    Drive_Switch("Engine_SELECTOR", 14);
    Drive_Light("Put_Lights", 2);
    Drive_Light("Put_Lights", 2);
    Drive_Switch("Engine_SELECTOR", 11);
    Drive_Switch("E3_Energy_Ref_Energy_Reference:Hs_Plot_Hp_s_plot.Energy_Select_sw", 8);
    Drive_Switch("E4_Energy_Ref_Energy_Reference:Hs_Plot_Hp_s_plot.Energy_Select_sw", 8);
    Drive_Switch("E4_Energy_Ref_Energy_Reference:Hs_Plot_Hp_s_plot.Energy_Select_sw", 8);

    if (Engine_Zoom == 1)
    {
        Hide_Frame("Big_Viewport_CRT","Engine_Zoomed");
        Get_Frame("Energy_Ent_Plots", 0, FOREGROUND_FRAME);
        Engine_Zoom = 0;
    }
if (Balance==1)
{
    Hide_Frame("BigViewport_CRT","Fuel_Balances");
    Balance=0;
    Hide_Frame("Tank_Light_2_Test");
}
if (Fuel_Zoom==1)
{
    Hide_Frame("Tank_Light_2_Test");
    Hide_Frame("BigViewport_CRT","Fuel_Zoomed");
    Fuel_Zoom=0;
}
if (Fuel_System==1)
{
    Hide_Frame("BigViewport_CRT","Fuel_system");
    Fuel_System=0;
}
'Set_Current_Layer(Layer1)''
Drive_Light("",'E1_Energy_ref.Energy_Reference.Energy_Ref_Light",1);
Drive_Light("",'E1_Energy_ref.Energy_Reference.Energy_Ref_Light",2);
Drive_Light("",'E2_Energy_ref.Energy_Reference.Energy_Ref_Light",0);
Drive_Light("",'E4_Energy_ref.Energy_Reference.Energy_Ref_Light",2);
Drive_Light("",'Labels",4);
') -> Monitor
{
    Appear_Frame("",'Energy_Ent_Plots2");
    Drive_Switch("",'Fuel_Overview_Sw",1;
    Drive_Switch("",'Engine_Selector",14;
    Drive_Switch("",'Engine_Selector",14;
    Drive_Switch("",'Engine_Selector",14;
    Drive_Switch("",'Engine_Selector",14;
    Drive_Switch("",'E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",3;
    Drive_Switch("",'E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",3;
    Drive_Light("",'Pot_Lights",2;
    Drive_Light("",'Pot_Lights",1;

if (Balance==1)
{
    Hide_Frame("BigViewport_CRT","Fuel_Balances");
    Balance=0;
}
if (Fuel_Zoom==1)
{
    
    Fuel_Zoom=0;
    Hide_Frame("Tank_Light_2_Test");
}
if (Fuel_System==1)
{
    Hide_Frame("BigViewport_CRT","Fuel_system");
    Fuel_System=0;
    Hide_Frame("Tank_Light_2_Test");
}
if (Engine_Zoom==0)
{
    Appear_Frame("BigViewport_CRT","Engine_Zoomed");
    Scale_Frame("BigViewport_CRT","Engine_Zoomed",1,1,5);
    Engine_Zoom=1;
}
'Set_Current_Layer(Layer1)''
potx=226;
poty=444;
Drive_Light("",'Pot_Lights",1");
Drive_Pot("vertical_1","pots_y");
Drive_Pot("horizontal_1","pots_z");
Drive_Light("Labels",4);
Drive_Light("E3_Energy_ref.Energy_Reference.Energy_Ref_Light",1);
Drive_Light("E1_Energy_ref.Energy_Reference.Energy_Ref_Light",2);
Drive_Light("E2_Energy_ref.Energy_Reference.Energy_Ref_Light",0);
Drive_Light("E4_Energy_ref.Energy_Reference.Energy_Ref_Light",2);
}
}

{
    Drive_Switch("Fuel_Overview_Sw",1);
    Drive_Switch("Engine_Selector",14);
    Drive_Switch("Engine_Selector",14);
    Drive_Switch("Engine_Selector",14);
    Drive_Switch("Engine_Selector",14);
    Drive_Switch("Engine_Selector",14);
    Drive_Switch("E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch("E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch("E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
}

Drive_Light("Pot_Lights",2);
if (Balance==1)
{
    Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
    Balance=0;
}

*Set_Current_Layer(Layer1)*:
Drive_Light("E3_Energy_ref.Energy_Reference.Energy_Ref_Light",1);
Drive_Light("E1_Energy_ref.Energy_Reference.Energy_Ref_Light",3);
Drive_Light("E2_Energy_ref.Energy_Reference.Energy_Ref_Light",0);
Drive_Light("E4_Energy_ref.Energy_Reference.Energy_Ref_Light",2);

if (Fuel_Zoom==1)
{
    Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
    Fuel_Zoom=0;
    Hide_Frame("Tank_Light_2_Test");
}
if (Fuel_System==1)
{
    Hide_Frame("Big_Viewport_CRT","Fuel_system");
    Fuel_System=0;
    Hide_Frame("Tank_Light_2_Test");
}
if (Engine_Zoom==0)
{
    Appear_Frame("Big_Viewport_CRT","Engine_Zoomed");
    Scale_Frame("Big_Viewport_CRT","Engine_Zoomed",1,1.5);
    Engine_Zoom=1;
}

*Set_Current_Layer(Layer1)*:
pots=226;
pots=444;
Drive_Light("Pot_Lights_1",1);
Channel_Member_Put_Buffer("Session_Common","","Location",MEMBER_DIMEN(1),12.1.CHAN_FLOAT,&pox,1.CHANNEL_WRIT E1);

Channel_Member_Put_Buffer("Session_Common","","Location",MEMBER_DIMEN(1),13.1.CHAN_FLOAT,&pox,1.CHANNEL_WRIT E1);

    Drive_Pot(","vertical_1",pox);
    Drive_Put(","horizontal_1",pox);
    Drive_Light(","Labels",4);
}

} -> Monitor

(SWITCH E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_x_plot.Energy_Select_sw 2,3)
{
    Drive_Switch("Fuel_Overview_Sw",1);
    Drive_Switch("Engine_Selector",14);
    Drive_Switch("Engine_Selector1",14);
    Drive_Switch("Engine_Selector2",14);
    Drive_Switch("Engine_Selector4",14);
    Drive_Switch("E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_x_plot.Energy_Select_sw",8);
    Drive_Switch("E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_x_plot.Energy_Select_sw",8);
    Drive_Switch("E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_x_plot.Energy_Select_sw",8);

    Drive_Light("Pot_Lights",2);

    if (Balance==1)
    {
        Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
        Balance=0;
    }
    *Set_Current_Layer(Layer1)*;
    Drive_Light("E3_Energy_ref.Energy_Reference.Energy_Ref_Light",1);
    Drive_Light("E1_Energy_ref.Energy_Reference.Energy_Ref_Light",1);
    Drive_Light("E2_Energy_ref.Energy_Reference.Energy_Ref_Light",0);
    Drive_Light("E4_Energy_ref.Energy_Reference.Energy_Ref_Light",2);

    if (Fuel_Zoom==1);
    {
        Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
        Fuel_Zoom=0;
        Hide_Frame("Tank_Light_2_Test");
    }
    if (Fuel_System==1);
    {
        Hide_Frame("Big_Viewport_CRT","Fuel_system");
        Fuel_System=0;
        Hide_Frame("Tank_Light_2_Test");
    }
    if (Engine_Zoom==0);
    {
        *Appear_Frame("Big_Viewport_CRT","Engine_Zoomed");
        Scale_Frame("Big_Viewport_CRT","Engine_Zoomed",1,1.5);
        Engine_Zoom=1;
    }
    *Set_Current_Layer(Layer1)*;
    *poxx=226;
    *poy=444;
    Drive_Light("Pot_Lights_1",1);

Channel_Member_Put_Buffer("Session_Common","","Location",MEMBER_DIMEN(1),12.1.CHAN_FLOAT,&pox,1.CHANNEL_WRIT E1);

Channel_Member_Put_Buffer("Session_Common","","Location",MEMBER_DIMEN(1),13.1.CHAN_FLOAT,&pox,1.CHANNEL_WRIT E1);

    Drive_Pot(","vertical_1",pox,);
Drive_Light"."Labels".4);  
} -> Monitor
(SWITCH E3".Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw 3-4")
{
  Drive_Switch"."Fuel_Overview_Sw".1;
  Drive_Switch"."Engine_Selector1".14;
  Drive_Switch"."Engine_Selector2".14;
  Drive_Switch"."Engine_Selector3".14;
  Drive_Switch"."E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw".S1;
  Drive_Switch"."E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw".S1;
  Drive_Switch"."E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw".S1;

  Drive_Light"."Pot_Lights".2;

  if (Balance==11){
    Hide_Frame("Big_Viewport_CRT"."Fuel_Balances");
    Balance=0;
  }
  *Set_Current_LayerLayer1*
  Drive_Light"."E3_Energy_ref.Energy_Reference.Energy_Ref_Light".1;
  Drive_Light"."E1_Energy_ref.Energy_Reference.Energy_Ref_Light".3;
  Drive_Light"."E2_Energy_ref.Energy_Reference.Energy_Ref_Light".0;
  Drive_Light"."E4_Energy_ref.Energy_Reference.Energy_Ref_Light".2;

  if (Fuel_Zoom==1);
  {
    Hide_Frame("Big_Viewport_CRT"."Fuel_Zoomed");
    Fuel_Zoom=0;
    Hide_Frame"."Tank_Light_2_Test";
  }
  if (Fuel_System=1);
  {
    Hide_Frame("Big_Viewport_CRT"."Fuel_system");
    Fuel_System=0;
    Hide_Frame"."Tank_Light_2_Test";
  }
  if (Engine_Zoom==0);
  {
    *Appear_Frame"."Engine_Zoomed";
    Scale_Frame("Big_Viewport_CRT"."Engine_Zoomed",1.1,1.5);
  }
  *Set_Current_LayerLayer1*;
  p ox=1043;
  p oy=443;
  Drive_Light"."Pot_Lights".1.11;

Channel_Member_Put_Buffern"Session_Common","member"."Location".MEMBER_DIMEN(11,12,1,CHAN_FLOAT,&pox,1,CHANNEL_WRITE);

Channel_Member_Put_Buffern"Session_Common","member"."Location".MEMBER_DIMEN(11,13,1,CHAN_FLOAT,&poy,1,CHANNEL_WRITE);

Drive_Pot"."vertical_1".p oy;
Drive_Pot"."horizontal_1".p ox;

Drive_Light"."Labels".4);
} -> Monitor
(SWITCH E3".Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw 4-5)
{
  Drive_Switch"."Fuel_Overview_Sw".1;
  Drive_Switch"."Engine_Selector3".14;
Drive_Switch("".""Engine_Selector1"","14");
Drive_Switch("".""Engine_Selector2"",14);
Drive_Switch("".""Engine_Selector3"",14);
Drive_Switch("".""E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw"",8);
Drive_Switch("".""E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw"",8);
Drive_Switch("".""E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw"",8);

Drive_Light("".""Pot_Lights",2);

if (Balance==1){
  Hide_Frame("Big_Vsiewport_CRT","Fuel_Balances");
  Balance=0;
}

*Set_Current_Layer(Layer1)*;

Drive_Light("".""E3_Energy_ref.Energy_Reference.Energy_Ref_Light",1);
Drive_Light("".""E1_Energy_ref.Energy_Reference.Energy_Ref_Light",3);
Drive_Light("".""E2_Energy_ref.Energy_Reference.Energy_Ref_Light",0);
Drive_Light("".""E4_Energy_ref.Energy_Reference.Energy_Ref_Light",2);

if (Fuel_Zoom==1);
{
  Hide_Frame("Big_Vsiewport_CRT","Fuel_Zoomed");
  Fuel_Zoom=0;
  Hide_Frame("".""Tank_Light_2_Test");
}
if (Fuel_System==1);
{
  Hide_Frame("Big_Vsiewport_CRT","Fuel_system");
  Fuel_System=0;
  Hide_Frame("".""Tank_Light_2_Test");
}
if (Engine_Zoom==0);
{
  *Appear_Frame("Big_Vsiewport_CRT","Engine_Zoomed");
  Scale_Frame("Big_Vsiewport_CRT","Engine_Zoomed",1.15);
  Engine_Zoom=1;
}
*Set_Current_Layer(Layer1)*;
potx=226;
poty=444;
Drive_Light("".""Pot_Lights",11);

Channel_Member_Put_Buffert("Session_Common","".""Location",MEMBER_DIMEN(11.12.1.CHAN_FLOAT,potx,1.CHANNEL_WRITE);

Channel_Member_Put_Buffert("Session_Common","".""Location",MEMBER_DIMEN(11.13.1.CHAN_FLOAT,poty,1.CHANNEL_WRITE);

Drive_Pot("".""vertical",1","".poty);
Drive_Pot("".""horizontal",1","".potx);

Drive_Light("".""Labels",4);
} -> Monitor

(SWITCH E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw '5-6')
{
  Drive_Switch("".""Fuel_Overview_Sw",1);
  Drive_Switch("".""Engine_Selector1",14);
  Drive_Switch("".""Engine_Selector2",14);
  Drive_Switch("".""Engine_Selector3",14);
  Drive_Switch("".""E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
  Drive_Switch("".""E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
  Drive_Switch("".""E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
}
Drive_Light","Pot_Lights",2);

if (Balance==1)
{
    Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
    Balance=0;
}

*Set_Current_LayerLayer1*;
Drive_Light","E3_Energy_ref.Energy_Reference.Energy_Ref_Light",1;
Drive_Light","E1_Energy_ref.Energy_Reference.Energy_Ref_Light",3;
Drive_Light","E2_Energy_ref.Energy_Reference.Energy_Ref_Light",0;
Drive_Light","E4_Energy_ref.Energy_Reference.Energy_Ref_Light",2;

if (Fuel_Zoom=1)
{
    Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
    Fuel_Zoom=0;
    Hide_Frame",""Tank_Light_2_Test");
}

if (Fuel_System=1)
{
    Hide_Frame("Big_Viewport_CRT","Fuel_system");
    Fuel_System=0;
    Hide_Frame",""Tank_Light_2_Test");
}

if (Engine_Zoom==0)
{
    *Appear_Frame"Big_Viewport_CRT","Engine_Zoomed");
    Scale_Frame"Big_Viewport_CRT","Engine_Zoomed",1.1,1.5;
    Engine_Zoom=1;
}

*Set_Current_LayerLayer1*;
px=226;
py=441;
Drive_Light","Pot_Lights",1);

Channel_Member_Pat_Buffer"Session_Common",""Location",MEMBER_DIMEN(1),12.1,CHAN_FLOAT,&px,1,CHANNEL_WRITE,E1;

Channel_Member_Pat_Buffer"Session_Common",""Location",MEMBER_DIMEN(1),13.1,CHAN_FLOAT,&py,1,CHANNEL_WRITE,E1;

Drive_Pxt","vertical",1",px);
Drive_Pxt","horizontal",1",py);
Drive_Light","Labels",4);
}

} -> Monitor
(SWITCH E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw '6-0')
{
    Drive_Switch","Fuel_Overview_Sw",1);
    Drive_Switch","Engine_Selector3",1.4);
    Drive_Switch","Engine_Selector1",1.4);
    Drive_Switch","Engine_Selector2",1.4);
    Drive_Switch","Engine_Selector3",1.4);
    Drive_Switch","E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch","E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch","E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);

Drive_Light","Pot_Lights",2);

if (Balance==1)
{
    Hide_Frame"Big_Viewport_CRT","Fuel_Balances");
    Balance=0;
}
if (Fuel_Zoom>1):
    { 
        Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
        Fuel_Zoom=0;
        Hide_Frame("Tank_Light_2_Test");
    }
if (Fuel_System=1):
    { 
        Hide_Frame("Big_Viewport_CRT","Fuel_system");
        Fuel_System=0;
        Hide_Frame("Tank_Light_2_Test");
    }
if (Engine_Zoom=0):
    { 
        Appear_Frame("Big_Viewport_CRT","Engine_Zoomed");
        Scale_Frame("Big_Viewport_CRT","Engine_Zoomed",1.1,5);
        Engine_Zoom=1;
    }
*Set_Current_Layer(1)*;
        ptx=776;
        pty=144;
        Drive_Light(""",""Pot_Lights","1",1);
        Drive_Light(""",""Locat","M","member_dimen","1",1,chan_float,
        &posx,1,channel_write);
        Drive_Light(""",""Labels",4);
} -> Monitor
    { 
        Drive_Switch(""",""Engine_Selector",3",14);
        Drive_Light(""",""Labels",4);
} -> Monitor
{BUTTON Big_Viewport_CRT.Engine_Zoomed.Fuel_Enrich_Valv 'off'
    { 
        Hide_Object("Big_Viewport_CRT","Engine_Zoomed_Menu");
} -> Monitor
{BUTTON Big_Viewport_CRT.Engine_Zoomed.Fuel_Enrich_Valv 'on'
    { 
        Appear_Object("Big_Viewport_CRT","Engine_Zoomed_Menu");
} -> Monitor
{MENU_SELECT Big_Viewport_CRT.Engine_Zoomed_Menu.Test_Menu.Nick.Menu 'quit'
    { 
        Hide_Object("Big_Viewport_CRT","Engine_Zoomed_Menu");
        Drive_Button("Big_Viewport_CRT","Engine_Zoomed_Menu",1);
} -> Monitor
{SWITCH Engine_Selector4 'code1'
    { 
        Appear_Frame(""",""Energy_Eng_Plots2");
        Drive_Switch(""",""Fuel_Overview_Sw",1);
        Drive_Switch(""",""Engine_Selector1",14);
        Drive_Switch(""",""Engine_Selector2",14);
        Drive_Switch(""",""Engine_Selector3",14);
        Drive_Light(""",""Pot_Lights",2);
Drive_Switch("E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
Drive_Switch("E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
Drive_Switch("E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
Drive_Switch("E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);

if (Balance==1){
    Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
    Balance=0;
}

*Set_Current_Layer.Layer1*;
Drive_Light("E4_Energy_ref.Energy_Reference.Energy_Ref_Light",1);
Drive_Light("E1_Energy_ref.Energy_Reference.Energy_Ref_Light",2);
Drive_Light("E2_Energy_ref.Energy_Reference.Energy_Ref_Light",3);
Drive_Light("E3_Energy_ref.Energy_Reference.Energy_Ref_Light",0);

if (Fuel_Zoom==1){
    Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
    Fuel_Zoom=0;
    Hide_Frame("Tank_Light_2",Test");
}
if (Fuel_System==1){
    Hide_Frame("Big_Viewport_CRT","Fuel_system");
    Fuel_System=0;
    Hide_Frame("Tank_Light_2",Test");
}
if (Engine_Zoom==0){
    *Appear_Frame("Big_Viewport_CRT","Engine_Zoomed")*;
    Scale_Frame("Big_Viewport_CRT","Engine_Zoomed",1,1,5);
    Engine_Zoom=1;
}

*Set_Current_Layer.Layer1*;
poxx=1033;
poyy=531;
Drive_Light("Pot_Lights_1",1);

Channel_Member_Put_Buffer("Session_Common","","Location",MEMBER_DIMEN1, Chan_FLOAT,&posx,1, CHANNEL_WRITE_E1);

Channel_Member_Put_Buffer("Session_Common","","Location",MEMBER_DIMEN1, Chan_FLOAT,&posy,1, CHANNEL_WRITE_E1);

Drive_Plot("vertical",poxx);
Drive_Plot("horizontal",poyy);
Drive_Light("Labels",5);

} -> Monitor
(SWITCH Engine_Selector4 code2) {
    *Appear_Frame("Energy_Ent_Plots2")*;
    Drive_Switch("Fuel_Overview_Sw",1);
    Drive_Switch("Engine_Selector1",14);
    Drive_Switch("Engine_Selector2",14);
    Drive_Switch("Engine_Selector3",14);
    Drive_Switch("E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch("E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch("E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch("E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);

    if (Balance==1){
        Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
        Balance=0;
    }

    *Set_Current_Layer.Layer1*;
    Drive_Light("E4_Energy_ref.Energy_Reference.Energy_Ref_Light",1);
Drive_Light"."E1_Energy_ref.Energy_Reference.Energy_Ref_Light",2);
Drive_Light"."E2_Energy_ref.Energy_Reference.Energy_Ref_Light",3);
Drive_Light"."E3_Energy_ref.Energy_Reference.Energy_Ref_Light",0);

if (Fuel_Zoom==1):
    {
    Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
    Fuel_Zoom=0;
    Hide_Frame(""."Tank_Light_2_Test");
    }
if (Fuel_System==1):
    {
    Hide_Frame("Big_Viewport_CRT","Fuel_system");
    Fuel_System=0;
    Hide_Frame(""."Tank_Light_2_Test");
    }
if (Engine_Zoom==0):
    {
    Appear_Frame("Big_Viewport_CRT","Engine_Zoomed");
    Scale_Frame("Big_Viewport_CRT","Engine_Zoomed",1.1,1.5);
    Engine_Zoom=1;
    }
    *Set_Current_LayerLayer1*:
    posx=1043;
    posy=443;
    Drive_Light"."Pot_Lights",1,1,1;
    Drive_Light"."Pot_Lights",2;

Channel_Member_Pot_Buffert"Session_Common",""."Location",MEMBER_DIMEN 1,1,2,1.CHAN_FLOAT,&posy,1.CHANNEL_WRITE EI;

Channel_Member_Pot_Buffert"Session_Common",""."Location",MEMBER_DIMEN 1,1,3,1.CHAN_FLOAT,&posy,1.CHANNEL_WRITE EI;

    Drive_Pot"."Vertical",1,posy;
    Drive_Pot"."horizontal",1,posy;
    Drive_Light"."Labels",5;
} -> Monitor
(sSELECT Engine Selector4 code=3)
{
    Appear_Frame(""."Energy_Ent_Plate2");
    Drive_Switch(""."Fuel_Overview_Sw",1,1);
    Drive_Switch(""."Engine_Selector1",14,1);
    Drive_Switch(""."Engine_Selector2",14,1);
    Drive_Switch(""."Engine_Selector3",14,1);
    Drive_Light"."Pot_Lights",2,1;
    Drive_Switch(""."E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8,1);
    Drive_Switch(""."E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8,1);
    Drive_Switch(""."E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8,1);
    Drive_Switch(""."E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8,1);

    if (Balance==1){
    Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
    Balance=0;
    }
    *Set_Current_LayerLayer1*:
    Drive_Light"."E4_Energy_ref.Energy_Reference.Energy_Ref_Light",1,1;
    Drive_Light"."E1_Energy_ref.Energy_Reference.Energy_Ref_Light",2,1;
    Drive_Light"."E2_Energy_ref.Energy_Reference.Energy_Ref_Light",3,1;
    Drive_Light"."E3_Energy_ref.Energy_Reference.Energy_Ref_Light",0,1;

    if (Fuel_Zoom==1):
    {
    Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
    Fuel_Zoom=0;
    Hide_Frame(""."Tank_Light_2_Test");
}
if (Fuel_System==1):
    
    Hide_Frame("Big_Viewport_CRT","Fuel_system");
    Fuel_System=0;
    Hide_Frame("Tank_Light_2_Test");

    if (Engine_Zoom==0):
        
        Appear_Frame("Big_Viewport_CRT","Engine_Zoomed");
        Scale_Frame("Big_Viewport_CRT","Engine_Zoomed",1.1,1.51);
        Engine_Zoom=1;
    
    "Set_Current_LayerLayer1";
    potx=156;
    poty=44;
    Drive_Light("Pot_Lights_1",1)

Channel_Member_Put_Buffert("Session_Common","Location",MEMBER_DIMEN(1121.CHAN FLOAT,&potx.1.CHANNEL_WRITE);

Channel_Member_Put_Buffert("Session_Common","Location",MEMBER_DIMEN(1131.CHAN FLOAT,&poty.1.CHANNEL_WRITE);

    Drive_Put("vertical_1",poty);
    Drive_Put("horizontal_1",potx);

    Drive_Light("Labels",5);

    } -> Monitor
    (SWITCH Engine_Selector4 'code4')
    
    Appear_Frame("Energy_Eat_Plots2");
    Drive_Switch("Fuel_Overview_Sw",1);
    Drive_Switch("Engine_Selector1",14);
    Drive_Switch("Engine_Selector2",14);
    Drive_Switch("Engine_Selector3",14);
    Drive_Light("Put_Lights_2");
    Drive_Switch("E4_Energy_ref.Energy_Reference.Hs Plot Ref.Hp_s_plot.Energy_Select_sw",11);
    Drive_Switch("E1_Energy_ref.Energy_Reference.Hs Plot Ref.Hp_s_plot.Energy_Select_sw",11);
    Drive_Switch("E2_Energy_ref.Energy_Reference.Hs Plot Ref.Hp_s_plot.Energy_Select_sw",11);
    Drive_Switch("E3_Energy_ref.Energy_Reference.Hs Plot Ref.Hp_s_plot.Energy_Select_sw",11);

    if (Balance==1)
        
        Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
        Balance=0;

        "Set_Current_LayerLayer1";
        Drive_Light("E4_Energy_ref.Energy_Reference.Energy_Ref.Light",11);
        Drive_Light("E1_Energy_ref.Energy_Reference.Energy_Ref.Light",21);
        Drive_Light("E2_Energy_ref.Energy_Reference.Energy_Ref.Light",31);
        Drive_Light("E3_Energy_ref.Energy_Reference.Energy_Ref.Light",0);

    if (Fuel_Zoom==1):
        
        Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
        Fuel_Zoom=0;
        Hide_Frame("Tank_Light_2_Test");

    } -> (Fuel_System==1):

    (Hide_Frame("Big_Viewport_CRT","Fuel_system");
    Fuel_System=0;
    Hide_Frame("Tank_Light_2_Test");

    } -> (Engine_Zoom==0):

    Appear_Frame("Big_Viewport_CRT","Engine_Zoomed");
Scale_Frame("Big_Viewport_CRT","Engine_Zoomed",1,1,1);
Engine_Zoom=1;
}
*Set_Current_Layer(Layer1)*;
pxx=199;
pyy=1070;
Drive_Light("","Pot_Lights_1",1);

Channel<Member_Put_Buffert,"Session_Common","","","Location",MEMBER_DIMEN(1,12,1,CHAN_FLOAT,&pxx,1,CHANNEL_WRITE);

Channel<Member_Put_Buffert,"Session_Common","","","Location",MEMBER_DIMEN(1,13,1,CHAN_FLOAT,&pyy,1,CHANNEL_WRITE);

Drive_Pot("","vertical_1","poty");
Drive_Pot("","horizontal_1","potx");
Drive_Light("","Labels",5);
}

-> Monitor
(SWITCH Engine_Selector4 'code5')
{
Appear_Frame("","Energy_Ent_Plots2");
Drive_Switch("","Fuel_Overview_Sw",1);
Drive_Switch("","Engine_Selector1",1,14);
Drive_Switch("","Engine_Selector2",1,14);
Drive_Switch("","Engine_Selector3",1,14);
Drive_Light("","Pot_Lights",1,2);
Drive_Switch("","E4_Energy_ref.Energy_0 reference.Hs_Plot_Ref.Hp_s.plot.Energy_Select_sw",81);
Drive_Switch("","E1_Energy_ref.Energy_0 reference.Hs_Plot_Ref.Hp_s.plot.Energy_Select_sw",81);
Drive_Switch("","E2_Energy_ref.Energy_0 reference.Hs_Plot_Ref.Hp_s.plot.Energy_Select_sw",81);
Drive_Switch("","E3_Energy_ref.Energy_0 reference.Hs_Plot_Ref.Hp_s.plot.Energy_Select_sw",81);

if (Balance=1){
Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
Balance=0;
}

*Set_Current_Layer(Layer1)*;
Drive_Light("","E4_Energy_ref.Energy_0 reference.Energy_Ref_Light",1);
Drive_Light("","E1_Energy_ref.Energy_0 reference.Energy_Ref_Light",2);
Drive_Light("","E2_Energy_ref.Energy_0 reference.Energy_Ref_Light",3);
Drive_Light("","E3_Energy_ref.Energy_0 reference.Energy_Ref_Light",0);

if (Fuel_Zoom=1);
{
Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
Fuel_Zoom=0;
Hide_Frame("","Tank_Light_2_Text");
}
if (Fuel_System=1);
{
Hide_Frame("Big_Viewport_CRT","Fuel_system");
Fuel_System=0;
Hide_Frame("","Tank_Light_2_Text");
}
if (Engine_Zoom=0);
{
Appear_Frame("Big_Viewport_CRT","Engine_Zoomed");
Scale_Frame("Big_Viewport_CRT","Engine_Zoomed",1,1,1,1,1);
Engine_Zoom=1;
}
*Set_Current_Layer(Layer1)*;
pxx=665;
pyy=0;
Drive_Light("","Pot_Lights_1",1);

Channel<Member_Put_Buffert,"Session_Common","","","Location",MEMBER_DIMEN(1,12,1,CHAN_FLOAT,&pxx,1,CHANNEL_WRITE);
Channel_Member_Put_Buffer "Session_Common","\","Location",MEMBER_DIMEN(1,12,1.CHAN_FLOAT.&poty.1.CHANNEL_WRITE);

Drive_Pot\"."vertical_1","poty\);
Drive_Pot\"."horizontal_1","potx\);

Drive_Light\"."Labels",".5\);
} -> Monitor
(SWITCH Engine_SELECTOR4 'code6')
{
    Appear_Frame\"."Energy_Ref_Plots2\";
    Drive_Switch\"."Fuel_Overview_Sw",1\);
    Drive_Switch\"."Engine_Selector1",14\);
    Drive_Switch\"."Engine_Selector2",14\);
    Drive_Switch\"."Engine_Selector2",14\);
    Drive_Light\"."Pot_Lights",2\);
    Drive_Switch\"."E1_Energy_Ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8\);
    Drive_Switch\"."E1_Energy_Ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8\);
    Drive_Switch\"."E2_Energy_Ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8\);
    Drive_Switch\"."E3_Energy_Ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8\);

    if (Balance==1) {
        Hide_Frame("Big_Viewport_CRT", "Fuel_Balances");
        Balance=0;
    }

    "Set_Current_Layer(Layer1)"
    Drive_Light\"."E4_Energy_Ref.Energy_Reference.Energy_Ref_Light",1;
    Drive_Light\"."E1_Energy_Ref.Energy_Reference.Energy_Ref_Light",2;
    Drive_Light\"."E2_Energy_Ref.Energy_Reference.Energy_Ref_Light",3;
    Drive_Light\"."E3_Energy_Ref.Energy_Reference.Energy_Ref_Light",0;

    if (Fuel_Zoom=1) {
        Hide_Frame("Big_Viewport_CRT", "Fuel_Zoomed");
        Fuel_Zoom=0;
        Hide_Frame\"."Tank_Light_2_Test\";
    }

    if (System=1) {
        Hide_Frame("Big_Viewport_CRT", "Fuel_system");
        Fuel_System=0;
        Hide_Frame\"."Tank_Light_2_Test\";
    }

    if (Engine_Zoom==0) {
        Appear_Frame("Big_Viewport_CRT", "Engine_Zoomed");
        Scale_Frame("Big_Viewport_CRT", "Engine_Zoomed",1,1.5); 
        Engine_Zoom=1;
    }

    "Set_Current_Layer(Layer1)"
    potx=448;
    poty=441;
    Drive_Light\"."Pot_Lights",".1,".1\);
if (Balance == 1) {
    Hide_Frame("Big_Viewport_CRT", "Fuel_Balances");
    Balance = 0;
}

*Set_Current_LayerLayer1*;
Drive_Light("E4_Energy_ref.Energy_Reference.Energy_Ref_Light", 1);
Drive_Light("E1_Energy_ref.Energy_Reference.Energy_Ref_Light", 2);
Drive_Light("E2_Energy_ref.Energy_Reference.Energy_Ref_Light", 3);
Drive_Light("E3_Energy_ref.Energy_Reference.Energy_Ref_Light", 0);

if (Fuel_Zoom == 1) {
    Hide_Frame("Big_Viewport_CRT", "Fuel_Zoomed");
    Fuel_Zoom = 0;
    Hide_Frame("Tank_Light_2_Test");
}
if (Fuel_System == 1) {
    Hide_Frame("Big_Viewport_CRT", "Fuel_system");
    Fuel_System = 0;
    Hide_Frame("Tank_Light_2_Test");
}
if (Engine_Zoom == 0) {
    Appear_Frame("Big_Viewport_CRT", "Engine_Zoomed");
    Scale_Frame("Big_Viewport_CRT", "Engine_Zoomed", 0.0, 0.0, 1.1, 1.1);
    Engine_Zoom = 1;
}
*Set_Current_LayerLayer1*;
potx = 619;
poty = 0;
Drive_Light("Pot_Lights", 1.1);

Channel_Member_Put_Buffer("Session_Common", ",", "Location", MEMBER_DIMEN(1, 1, 2.1, 1.1, CHAN_FLOAT, &potx, 1, CHANNEL_WRITE_E));

Channel_Member_Put_Buffer("Session_Common", ",", "Location", MEMBER_DIMEN(1, 1.3, 1.1, CHAN_FLOAT, &poty, 1, CHANNEL_WRITE_E));

Drive_Pot("vertical", poty);
Drive_Pot("horizontal", potx);

Drive_Light("Labels", 5);
} -> Monitor
(SWITCH Engine_Selector4 "code4")
{
    Appear_Frame("Energy_Ent_Plots2");
    Drive_Switch("Fuel_Overview_Sw", 1);
    Drive_Switch("Engine_Selector1", 1.4);
    Drive_Switch("Engine_Selector2", 1.4);
    Drive_Switch("Engine_Selector3", 1.4);
    Drive_Light("Pot_Lights", 2);
    Drive_Switch("E4_Energy_ref.Energy_Reference.Hs_Plot_Hp_s_plot.Energy_Select_sw", 8);
    Drive_Switch("E1_Energy_ref.Energy_Reference.Hs_Plot_Hp_s_plot.Energy_Select_sw", 8);
    Drive_Switch("E2_Energy_ref.Energy_Reference.Hs_Plot_Hp_s_plot.Energy_Select_sw", 8);
    Drive_Switch("E3_Energy_ref.Energy_Reference.Hs_Plot_Hp_s_plot.Energy_Select_sw", 8);

    if (Balance == 1) {
        Hide_Frame("Big_Viewport_CRT", "Fuel_Balances");
        Balance = 0;
    }
}

*Set_Current_LayerLayer1*;
Drive_Light("E4_Energy_ref.Energy_Reference.Energy_Ref_Light", 1);
Drive_Light("E1_Energy_ref.Energy_Reference.Energy_Ref_Light", 2);
Drive_Light("E2_Energy_ref.Energy_Reference.Energy_Ref_Light", 3);
Drive_Light("E3_Energy_ref.Energy_Reference.Energy_Ref_Light", 0);
if (Fuel_Zoom==1)
{
    Hide_Frame("Big Viewport/CRT","Fuel_Zoomed");
    Fuel_Zoom=0;
    Hide_Frame("Tank_Light_2 Test");
}
if (Fuel_System==1)
{
    Hide_Frame("Big Viewport/CRT","Fuel System");
    Fuel_System=0;
    Hide_Frame("Tank_Light_2 Test");
}
if (Engine_Zoom==1)
{
    Appeare_Frame("Big Viewport/CRT","Engine_Zoomed");
    Scale_Frame("Big Viewport/CRT","Engine_Zoomed",1.1,1.5);
    Engine_Zoom=1;
}
*Set_Current_Layer(Layer1)*:
potx=0;
poty=867;
Drive_Light("Pot_Lights_1",1);
Channel_Member_Put_Buffer("Session_Common","Location",MEMBER_DIMEN[1],12,1.CHAN_FLOAT, &potx, 1.CHANNEL_WRITE);
Channel_Member_Put_Buffer("Session_Common","Location",MEMBER_DIMEN[1],13,1.CHAN_FLOAT, &poty, 1.CHANNEL_WRITE);

Drive_Pot("vertical_1",poty);
Drive_Pot("horizontal_1",potx);

Drive_Light("Labels",5);

} -> Monitor
(SWITCH Engine_Selector 'code'10)
{
    Appeare_Frame("Energy_Entr_Plots2");
    Drive_Switch("Fuel_Overview_Sw",1);
    Drive_Switch("Engine_Selector1",14);
    Drive_Switch("Engine_Selector2",14);
    Drive_Switch("Engine_Selector3",14);
    Drive_Light("Pot_Lights_2");
    Drive_Switch("E1_Energy_ref.Energy_Reference_Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch("E1_Energy_ref.Energy_Reference_Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch("E2_Energy_ref.Energy_Reference_Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch("E3_Energy_ref.Energy_Reference_Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);

    if (Balance==1)
    {
        Hide_Frame("Big Viewport/CRT","Fuel_Balances");
        Balance=0;
    }
*Set_Current_Layer(Layer1)*:
    Drive_Light("E1_Energy_ref.Energy_Reference.Energy_Ref_Light",1);
    Drive_Light("E1_Energy_ref.Energy_Reference.Energy_Ref_Light",2);
    Drive_Light("E2_Energy_ref.Energy_Reference.Energy_Ref_Light",3);
    Drive_Light("E3_Energy_ref.Energy_Reference.Energy_Ref_Light",0);
if (Fuel_Zoom==1)
{
    Hide_Frame("Big Viewport/CRT","Fuel_Zoomed");
    Fuel_Zoom=0;
    Hide_Frame("Tank_Light_2 Test");
}
if (Fuel_System==1)
{
*Set_Current_Layer(Layer1)*;
pots=500;
xy=500;
Drive_Light("Pot_Lights_1".1);1;

Channel_Member_Put_Buffer("Session_Common","Location",MEMBER_DIM(1),12,1.CHAN_FLOAT.&pots,1.CHANNEL_WRITE);

Channel_Member_Put_Buffer("Session_Common","Location",MEMBER_DIM(1),13,1.CHAN_FLOAT.&pots,1.CHANNEL_WRITE);

    Drive_Pott("vertical_1",pots);
    Drive_Pott("horizontal_1",pots);

    Drive_Light("Labels",5);
}

} => Monitor

(SWITCH Engine_Selector4 'code12')
{
    Appear_Frame("Energy_Ent_Plots2");
    Drive_Switch("Fuel_Overview_Sw",1);
    Drive_Switch("Engine_Selector1",14);
    Drive_Switch("Engine_Selector2",14);
    Drive_Switch("Engine_Selector3",14);
    Drive_Light("Pot_Lights",2);
    Drive_Switch("E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch("E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch("E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch("E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);

    if (Balance==1){
        Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
        Balance=0;
    }

    *Set_Current_Layer(Layer1)*;
    Drive_Light("E4_Energy_ref.Energy_Reference.Energy_Ref_Light",1);
    Drive_Light("E1_Energy_ref.Energy_Reference.Energy_Ref_Light",2);
    Drive_Light("E1_Energy_ref.Energy_Reference.Energy_Ref_Light",3);
    Drive_Light("E3_Energy_ref.Energy_Reference.Energy_Ref_Light",0);

    if (Fuel_Zoom==1);
      {
        Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
        Fuel_Zoom=0;
        Hide_Frame("Tank_Light_2.Test");
      }
      if (Fuel_System==1);
      {
        Hide_Frame("Big_Viewport_CRT","Fuel_system");
        Fuel_System=0;
        Hide_Frame("Tank_Light_2.Test");
      }
      if (Engine_Zoom==0);
      {
        Appear_Frame("Big_Viewport_CRT","Engine_Zoomed");
        Scale_Frame("Big_Viewport_CRT","Engine_Zoomed",1.1,5);
        Engine_Zoom=1;
      }

    *Set_Current_Layer(Layer1)*;
    pots=300;
    pots=500;
    Drive_Light("Pot_Lights_1",1);

Channel_Member_Put_Buffer("Session_Common","Location",MEMBER_DIM(1),12,1.CHAN_FLOAT.&pots,1.CHANNEL_WRITE);

Channel_Member_Put_Buffer("Session_Common","Location",MEMBER_DIM(1),13,1.CHAN_FLOAT.&pots,1.CHANNEL_WRITE);
Drive_Rot"="vertical_1".;poty);
Drive_Rot"="horizontal_1".;post);
Drive_Light"="Labels".5i);
} -> Monitor
(SWITCH Engine_Selector4 code 13)
{
  Appear_Frame"="Energy_Ent_Plots2"
  Drive_Switch"="Engine_Selector4".14;
  Drive_Light"="Pot_Lights".2i;
  Drive_Light"="Pot_Lights_1".2i;
  Drive_Switch"="Fuel_Overview_Sw".1i;
  Drive_Switch"="E4_Energy_ref_Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw".8i;
  Drive_Switch"="E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw".8i;
  Drive_Switch"="E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw".8i;
  Drive_Switch"="E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw".8i;

  if (Engine_Zoom==1);
  {
    Hide_Frame("Big_Viewport_CRT","Engine_Zoomed");
    Get_Frame"="Energy_Ent_Plots2".0.FOREGROUND_FRAME
    Engine_Zoom=0;
  }
  
  if (Balance==1){
    Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
    Balance=0;
    Hide_Frame"="Tank_Light_2_Test";
  }
  
  if (Fuel_Zoom==11){
    Hide_Frame"="Tank_Light_3_Test";
    Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
    Fuel_Zoom=0;
  }
  
  if (Fuel_System==11){
    Hide_Frame("Big_Viewport_CRT","Fuel_system");
    Fuel_System=0;
  }
  "Set_Current_LayerLayer1"i;
  Drive_Light"="E4_Energy_ref.Energy_Reference.Energy_Ref_Light".1i;
  Drive_Light"="E1_Energy_ref.Energy_Reference.Energy_Ref_Light".2i;
  Drive_Light"="E2_Energy_ref.Energy_Reference.Energy_Ref_Light".3i;
  Drive_Light"="E3_Energy_ref.Energy_Reference.Energy_Ref_Light".4i;
  Drive_Light"="Labels".5i;
} -> Monitor
{
  Appear_Frame"="Energy_Ent_Plots2"
  Drive_Switch"="Fuel_Overview_Sw".1i;
  Drive_Switch"="Engine_Selector4".14;
  Drive_Switch"="Engine_Selector1".14;
  Drive_Switch"="Engine_Selector2".14;
  Drive_Switch"="Engine_Selector3".14;

  Drive_Switch"="E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw".8i;
  Drive_Switch"="E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw".8i;
  Drive_Switch"="E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw".8i;

  Drive_Light"="Pot_Lights".2i;
  Drive_Light"="Pot_Lights_1".11i;

  if (Balance==1){
    Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
}
Balance=0;

if (Fuel_Zoom=1):
  
  Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
  Fuel_Zoom=0;
  Hide_Frame("Tank_Light_2","Test");

if (Fuel_System=1):
  
  Hide_Frame("Big_Viewport_CRT","Fuel_system");
  Fuel_System=0;
  Hide_Frame("Tank_Light_2","Test");

if (Engine_Zoom=0):
  
  Appear_Frame("Big_Viewport_CRT","Engine_Zoomed");
  Scale_Frame("Big_Viewport_CRT","Engine_Zoomed",1.15);
  Engine_Zoom=1;
  
  "Set_Current_Layer(Layer1)"
  
  Getxy=226;
  Sety=444;
  Drive_Light(""Pots_Lights",1,1);

Channel_Member_Put_Buffer "Session_Common",""Location",MEMBER_DIMEN1,12.1.CHAN_FLOAT,&pos.1.CHANNEL_WRITE E;

Channel_Member_Put_Buffer "Session_Common",""Location",MEMBER_DIMEN1,13.1.CHAN_FLOAT,&pos.1.CHANNEL_WRITE E;

  Drive_Pot("vertical","poty");
  Drive_Pot("horizontal","potx");

  Drive_Light("Labels",15);

  Drive_Light("E1_Energy_ref.Energy_Reference.Energy_Ref_Light",1);
  Drive_Light("E1_Energy_ref.Energy_Reference.Energy_Ref_Light",2);
  Drive_Light("E2_Energy_ref.Energy_Reference.Energy_Ref_Light",3);
  Drive_Light("E3_Energy_ref.Energy_Reference.Energy_Ref_Light",4);

  } -> Monitor

  
  Drive_Switch("Fuel_Overview_Sw",1);
  Drive_Switch("Engine_Selector4",14);
  Drive_Switch("Engine_Selector1",14);
  Drive_Switch("Engine_Selector2",14);
  Drive_Switch("Engine_Selector3",14);
  Drive_Switch("E1_Energy_ref.Energy_Reference.Hs_Plot.Ref.Hp_s_plot.Energy_Select_sw",8);
  Drive_Switch("E2_Energy_ref.Energy_Reference.Hs_Plot.Ref.Hp_s_plot.Energy_Select_sw",8);
  Drive_Switch("E3_Energy_ref.Energy_Reference.Hs_Plot.Ref.Hp_s_plot.Energy_Select_sw",8);

  Drive_Light("Pots_Lights",2);

if (Balance=1){
  Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
  Balance=0;
  
  "Set_Current_Layer(Layer1)"
  
  Getxy=226;
  Sety=444;
  Drive_Light("E4_Energy_ref.Energy_Reference.Energy_Ref_Light",1);
  Drive_Light("E1_Energy_ref.Energy_Reference.Energy_Ref_Light",2);
  Drive_Light("E2_Energy_ref.Energy_Reference.Energy_Ref_Light",3);
  Drive_Light("E3_Energy_ref.Energy_Reference.Energy_Ref_Light",4);
if (Fuel_Zoom=1):
    {
        Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
        Fuel_Zoom=0;
        Hide_Frame("Tank_Light_2_Test");
    }
if (Fuel_System=1):
    {
        Hide_Frame("Big_Viewport_CRT","Fuel_System");
        Fuel_System=0;
        Hide_Frame("Tank_Light_2_Test");
    }
if (Engine_Zoom=0):
    {
        Scale_Frame("Big_Viewport_CRT","Engine_Zoomed".1.1.5);
        Engine_Zoom=1;
        "Set_Current_LayerLayer1";
        potx=226;
        poy=444;
        Drive_Light("Pot_Lights_1",.1);
    }
Channel_Member_Put_Buffer("Session_Common",.1,.1.1,Location",MEMBER_DIM paradigm,1.CHAN_FLOAT&pox,1.CHANNEL_WRITE);
Channel_Member_Put_Buffer("Session_Common",.1,.1,Location",MEMBER_DIM paradigm,1.CHAN_FLOAT&poy,1.CHANNEL_WRITE);
    Drive_Pott("vertical",.1,potx);
    Drive_Pott("horizontal",.1,potx);
    Drive_Light("Labels",.5);
} -> Monitor
(SWITCH E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw 2:3)
    {
        Drive_Switch("Fuel_Overview_Sw",.1);
        Drive_Switch("Engine_Selec".1,14);
        Drive_Switch("Engine_Selec1",.14);
        Drive_Switch("Engine_Selec2",.14);
        Drive_Switch("Engine_Selec3",.14);
        Drive_Switch("Energy_Ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",.8);
        Drive_Switch("E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",.8);
        Drive_Switch("E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",.8);
    }
Drive_Light("Pot_Lights",.2);
if (Balance=1)
    {
        Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
        Balance=0;
    }
"Set_Current_LayerLayer1";
Drive_Light("E1_Energy_ref.Energy_Reference.Energy_Ref_Light",.1);
Drive_Light("E2_Energy_ref.Energy_Reference.Energy_Ref_Light",.2);
Drive_Light("E3_Energy_ref.Energy_Reference.Energy_Ref_Light",.3);
Drive_Light("E4_Energy_ref.Energy_Reference.Energy_Ref_Light",.0);
if (Fuel_Zoom=1):
    {
        Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
        Fuel_Zoom=0;
        Hide_Frame("Tank_Light_2_Test");
    }
if (Fuel_System=1):
    {

F-93

```
hide_frame("Big_Viewport_CRT","Fuel_System");
Fuel_System=0;
hide_frame("Tank_Light_2_Test");
}
if (Engine_Zoom=0):
{
    appear_frame("Big_Viewport_CRT","Engine_Zoomed");
    scale_frame("Big_Viewport_CRT","Engine_Zoomed",1.15);
    Engine_Zoom=1;
}
*set_current_layer(layer1*);
potx=226;
poty=444;
drive_light("Pot_Lights",1.1);
channel_member_put_buf("Session_Common","","Location",member_dimen(1),12.1,chan_float,0);channel_member_put_buf("Session_Common",",","Location",member_dimen(1),13.1,chan_float,0);

```
}

```
exasn Monitoring
(SWITCH E4_energy_ref_energy_reference.Hs_Plot_Ref.Hp_s_plot.energy_select_sw [3-4])
{
        drive_switch("Fuel_Overview_Sw",1);
        drive_switch("EngineSelector1",14);
        drive_switch("EngineSelector1",14);
        drive_switch("EngineSelector2",14);
        drive_switch("EngineSelector3",14);
        drive_switch("E1_energy_ref_energy_reference.Hs_Plot_Ref.Hp_s_plot.energy_select_sw",8);
        drive_switch("E2_energy_ref_energy_reference.Hs_Plot_Ref.Hp_s_plot.energy_select_sw",8);
        drive_switch("E3_energy_ref_energy_reference.Hs_Plot_Ref.Hp_s_plot.energy_select_sw",8);
}

```
}

```
if (balance=1){
    hide_frame("Big_Viewport_CRT","Fuel_Balances");
balance=0;
}
*set_current_layer(layer1*);
drive_light("E4_energy_ref_energy_reference.energy_ref_light",1);
drive_light("E1_energy_ref_energy_reference.energy_ref_light",2);
drive_light("E2_energy_ref_energy_reference.energy_ref_light",3);
drive_light("E3_energy_ref_energy_reference.energy_ref_light",0);
if (Fuel_Zoom=1):
{
    hide_frame("Big_Viewport_CRT","Fuel_Zoomed");
    Fuel_Zoom=0;
    hide_frame("Tank_Light_2_Test");
}
if (Fuel_System=1):
{
    hide_frame("Big_Viewport_CRT","Fuel_system");
    Fuel_System=0;
    hide_frame("Tank_Light_2_Test");
}
if (Engine_Zoom=0):
{
    appear_frame("Big_Viewport_CRT","Engine_Zoomed");
    scale_frame("Big_Viewport_CRT","Engine_Zoomed",1.15);
```

Engine_Zoom=1:
}
*Set_Current_Layer(Layer1):
potx=1043;
poty=443;
Drive_Light("".""Pot_Lights_1",1);

Channel_Member_Put_Buffet("Session_Common",":"".""Location",MEMBER_DIMEN(11,12,1,CHAN_FLOAT,&potx,1,CHANNEL_WRITE));

Channel_Member_Put_Buffet("Session_Common",":"".""Location",MEMBER_DIMEN(11,13,1,CHAN_FLOAT,&poty,1,CHANNEL_WRITE));

Drive_Pot("".""vertical_1",poty);
Drive_Pot("".""horizontal_1",potx);

Drive_Light("".""Labels",5);
}

Monitor

(SWITCH E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw '4-5')
{

Drive_Switch("".""Fuel_Overview_Sw",1);
Drive_Switch("".""Engine_Selector4",14);
Drive_Switch("".""Engine_Selector1",14);
Drive_Switch("".""Engine_Selector2",14);
Drive_Switch("".""Engine_Selector3",14);
Drive_Switch("".""E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
Drive_Switch("".""E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
Drive_Switch("".""E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);

Drive_Light("".""Pot_Lights",2);

if (Balance==1){
Hide_Frame("BigViewport_CRT","Fuel_Balances");
Balance=0;
}
*Set_Current_Layer(Layer1):

Drive_Light("".""E1_Energy_ref.Energy_Reference.Energy_Ref_Light",1);
Drive_Light("".""E1_Energy_ref.Energy_Reference.Energy_Ref_Light",2);
Drive_Light("".""E2_Energy_ref.Energy_Reference.Energy_Ref_Light",5);
Drive_Light("".""E3_Energy_ref.Energy_Reference.Energy_Ref_Light",0);

if (Fuel_Zoom=1);
{
Hide_Frame("BigViewport_CRT","Fuel_Zoomed");
Fuel_Zoom=0;
Hide_Frame("".""Tank_Light_2_Test");
}
if (Fuel_System=1);
{
Hide_Frame("BigViewport_CRT","Fuel_system");
Fuel_System=0;
Hide_Frame("".""Tank_Light_2_Test");
}
if (Engine_Zoom==0);
{
Appear_Frame("BigViewport_CRT","Engine_Zoomed");
Scale_Frame("BigViewport_CRT","Engine_Zoomed",1.1,5);
Engine_Zoom=1;
}

*Set_Current_Layer(Layer1):
potx=226;
poty=444;
Drive_Light("".""Pot_Lights_1",1);
Channel_Member_Put_Buffert("Session_Common", ".", ".", "Location", MEMBER_DIMENU1,12,1.CHAN_FLOAT,potx,1.CHANNEL_WRITE);

Channel_Member_Put_Buffert("Session_Common", ".", ".", "Location", MEMBER_DIMENU1,13,1.CHAN_FLOAT,potx,1.CHANNEL_WRITE);

    Drive_Pot"","vertical_1",".poty");
    Drive_Pot"","horizontal_1",".potx");

    Drive_Light"","Labels",5);
}
} > Monitor

(SWITCH E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw '5-6')
{
    Drive_Switch"","Fuel_Overview_Sw",1);
    Drive_Switch"","Engine_Selector4",1,4);
    Drive_Switch"","Engine_Selector1",1,4);
    Drive_Switch"","Engine_Selector2",1,4);
    Drive_Switch"","Engine_Selector3",1,4);
    Drive_Switch"","E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch"","E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
    Drive_Switch"","E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);

    Drive_Light"","Pot_Lights",2);

    if (Balance=1) {
        Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
        Balance=0;
    }

    "Set_Current_Layer(Layer1)":
    Drive_Light"","E4.Energy_ref.Energy_Reference.Energy_Ref_Light",1);
    Drive_Light"","E1.Energy_ref.Energy_Reference.Energy_Ref_Light",2);
    Drive_Light"","E2.Energy_ref.Energy_Reference.Energy_Ref_Light",3);
    Drive_Light"","E3.Energy_ref.Energy_Reference.Energy_Ref_Light",0);

    if (Fuel_Zoom=1) {
    }

    if (Fuel_System=1) {
    }

    if (Engine_Zoom=1) {
    }

    if (Appear_Frame("Big_Viewport_CRT","Engine_Zoomed"));
    Scale_Frame("Big_Viewport_CRT",1,1,5);
    Engine_Zoom=1;
}

"Set_Current_Layer(Layer1)":
potx=226;
poty=444;
Drive_Light"","Pot_Lights",1,1);
Drive_Pot"."vertical_1","potx1;
Drive_Pot"."horizontal_1","potx1;
Drive_Light"."Labels",5;
} -> Monitor
(SWITCH E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw '6-0')
{
  Drive_Switch"."Fuel_Overview_Sw",1;
  Drive_Switch"."Engine_Selector",14;
  Drive_Switch"."Engine_Selector1",14;
  Drive_Switch"."Engine_Selector2",14;
  Drive_Switch"."Engine_Selector3",14;
  Drive_Switch"."E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8;
  Drive_Switch"."E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8;
  Drive_Switch"."E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8;
}

Drive_Light"."Pot_Lights",2;

if (Balance==1)
{
  Hide_Frame("Big_Viewport_CRT","Fuel_Balances");
  Balance=0;
}

*Set_Current_LayerLayer1* :
Drive_Light"."E4_Energy_ref.Energy_Reference.Energy_Ref_Light",1;
Drive_Light"."E1_Energy_ref.Energy_Reference.Energy_Ref_Light",2;
Drive_Light"."E2_Energy_ref.Energy_Reference.Energy_Ref_Light",3;
Drive_Light"."E3_Energy_ref.Energy_Reference.Energy_Ref_Light",0;

if (Fuel_Zoom==1);
{
  Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
  Fuel_Zoom=0;
  Hide_Frame("Tank_Light_2",Test");
}

if (Fuel_System==1);
{
  Hide_Frame("Big_Viewport_CRT","Fuel_system");
  Fuel_System=0;
  Hide_Frame("Tank_Light_2",Test");
}

if (Engine_Zoom==0);
{
  *Set_Current_LayerLayer1* :
  potx226;
  potx444;
  Drive_Light"."Pot_Lights_1",1;

Channel_Member_Put_Buffer"Session_Common","Location",MEMBER_DIM(1),12,1.CHAN_FLOAT,&potx1,1.CHANNEL_WRITE;

Channel_Member_Put_Buffer"Session_Common","Location",MEMBER_DIM(1),13,1.CHAN_FLOAT,&poty1,1.CHANNEL_WRITE;

Drive_Pot"."vertical_1","potx1;
Drive_Pot"."horizontal_1","potx1;

Drive_Light"."Labels",5;
} -> Monitor
(SWITCH E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw 'off')
{
  Drive_Switch"."Engine_Selector",14;
Drive_Light("Labels",5);
} -> Monitor
{BUTTON Big_Viewport_CRT.Fuel_system.Tank_1_Dump_Pump.Pump_on_off.Pump_Button 'off')
{   Drive_Button("Big_Viewport_CRT.Fuel_Zoomed.Tank_1_Dump_Pump.Pump_on_off.Pump_Button",1);
} -> Monitor
{BUTTON Big_Viewport_CRT.Fuel_system.Tank_1_Dump_Pump.Pump_on_off.Pump_Button 'on')
{   Drive_Button("Big_Viewport_CRT.Fuel_Zoomed.Tank_1_Dump_Pump.Pump_on_off.Pump_Button",2);
} -> Monitor
{BUTTON Big_Viewport_CRT.Fuel_system.Tank_2_Dump_Pump.Pump_on_off.Pump_Button 'off')
{   Drive_Button("Big_Viewport_CRT.Fuel_Zoomed.Tank_2_Dump_Pump.Pump_on_off.Pump_Button",1);
} -> Monitor
{BUTTON Big_Viewport_CRT.Fuel_system.Tank_2_Dump_Pump.Pump_on_off.Pump_Button 'on')
{   Drive_Button("Big_Viewport_CRT.Fuel_Zoomed.Tank_2_Dump_Pump.Pump_on_off.Pump_Button",2);
} -> Monitor
{BUTTON Big_Viewport_CRT.Fuel_system.Tank_3_Dump_Pump.Pump_on_off.Pump_Button 'off')
{   Drive_Button("Big_Viewport_CRT.Fuel_Zoomed.Tank_3_Dump_Pump.Pump_on_off.Pump_Button",1);
} -> Monitor
{BUTTON Big_Viewport_CRT.Fuel_system.Tank_3_Dump_Pump.Pump_on_off.Pump_Button 'on')
{   Drive_Button("Big_Viewport_CRT.Fuel_Zoomed.Tank_3_Dump_Pump.Pump_on_off.Pump_Button",2);
} -> Monitor
{BUTTON Big_Viewport_CRT.Fuel_system.Tank_4_Dump_Pump.Pump_on_off.Pump_Button 'off')
{   Drive_Button("Big_Viewport_CRT.Fuel_Zoomed.Tank_4_Dump_Pump.Pump_on_off.Pump_Button",1);
} -> Monitor
{BUTTON Big_Viewport_CRT.Fuel_system.Tank_4_Dump_Pump.Pump_on_off.Pump_Button 'on')
{   Drive_Button("Big_Viewport_CRT.Fuel_Zoomed.Tank_4_Dump_Pump.Pump_on_off.Pump_Button",2);
} -> Monitor
{BUTTON Big_Viewport_CRT.Fuel_system.Tank_1_Filler.Refuel_Cap.Filler 'off')
{   Drive_Button("Big_Viewport_CRT.Fuel_Zoomed.Tank_1_Filler.Refuel_Cap.Filler",1);
} -> Monitor
{BUTTON Big_Viewport_CRT.Fuel_system.Tank_1_Filler.Refuel_Cap.Filler 'on')
{   Drive_Button("Big_Viewport_CRT.Fuel_Zoomed.Tank_1_Filler.Refuel_Cap.Filler",2);
} -> Monitor
{BUTTON Big_Viewport_CRT.Fuel_system.Tank_2_Filler.Refuel_Cap.Filler 'off')
{   Drive_Button("Big_Viewport_CRT.Fuel_Zoomed.Tank_2_Filler.Refuel_Cap.Filler",1);
} -> Monitor
{BUTTON Big_Viewport_CRT.Fuel_system.Tank_2_Filler.Refuel_Cap.Filler 'on')
{   Drive_Button("Big_Viewport_CRT.Fuel_Zoomed.Tank_2_Filler.Refuel_Cap.Filler",2);
} -> Monitor
{BUTTON Big_Viewport_CRT.Fuel_system.Tank_3_Filler.Refuel_Cap.Filler 'off')
{   Drive_Button("Big_Viewport_CRT.Fuel_Zoomed.Tank_3_Filler.Refuel_Cap.Filler",1);
} -> Monitor
{BUTTON Big_Viewport_CRT.Fuel_system.Tank_3_Filler.Refuel_Cap.Filler 'on')
{   Drive_Button("Big_Viewport_CRT.Fuel_Zoomed.Tank_3_Filler.Refuel_Cap.Filler",2);
} -> Monitor
{BUTTON Big_Viewport_CRT.Fuel_system.Tank_4_Filler.Refuel_Cap.Filler 'off')
{   Drive_Button("Big_Viewport_CRT.Fuel_Zoomed.Tank_4_Filler.Refuel_Cap.Filler",1);
} -> Monitor
{BUTTON Big_Viewport_CRT.Fuel_system.Tank_4_Filler.Refuel_Cap.Filler 'on')
{   Drive_Button("Big_Viewport_CRT.Fuel_Zoomed.Tank_4_Filler.Refuel_Cap.Filler",2);
} -> Monitor
{BUTTON Big_Viewport_CRT.Fuel_system.RH_Aux_Filler.Refuel_Cap.Filler 'off')
{
Drive_Button("Big_Viewport,CRT,Fuel_system,RH_Aux_Filler,Refuel_Cap,Filler",1);
} -> Monitor
(BUTTON Big_Viewport,CRT,Fuel_system,RH_Aux_Filler,Refuel_Cap,Filler On)
{ Drive_Button("Big_Viewport,CRT,Fuel_system,RH_Aux_Filler,Refuel_Cap,Filler",2);
} -> Monitor
(BUTTON Big_Viewport,CRT,Fuel_system,LH_Aux_Filler,Refuel_Cap,Filler Off)
{ Drive_Button("Big_Viewport,CRT,Fuel_system,LH_Aux_Filler,Refuel_Cap,Filler",1);
} -> Monitor
(BUTTON Big_Viewport,CRT,Fuel_system,LH_Aux_Filler,Refuel_Cap,Filler On)
{ Drive_Button("Big_Viewport,CRT,Fuel_system,LH_Aux_Filler,Refuel_Cap,Filler",2);
} -> Monitor
(BUTTON Big_Viewport,CRT,Fuel_system,E_1_Pump,Pump_on_off,Pump_Button Off)
{ Drive_Button("Big_Viewport,CRT,Fuel_system,E_1_Pump,Pump_on_off,Pump_Button",1);
} -> Monitor
(BUTTON Big_Viewport,CRT,Fuel_system,E_1_Pump,Pump_on_off,Pump_Button On)
{ Drive_Button("Big_Viewport,CRT,Fuel_system,E_1_Pump,Pump_on_off,Pump_Button",2);
} -> Monitor
(BUTTON Big_Viewport,CRT,Fuel_system,E_2_Pump,Pump_on_off,Pump_Button Off)
{ Drive_Button("Big_Viewport,CRT,Fuel_system,E_2_Pump,Pump_on_off,Pump_Button",1);
} -> Monitor
(BUTTON Big_Viewport,CRT,Fuel_system,E_2_Pump,Pump_on_off,Pump_Button On)
{ Drive_Button("Big_Viewport,CRT,Fuel_system,E_2_Pump,Pump_on_off,Pump_Button",2);
} -> Monitor
(BUTTON Big_Viewport,CRT,Fuel_system,E_3_Pump,Pump_on_off,Pump_Button Off)
{ Drive_Button("Big_Viewport,CRT,Fuel_system,E_3_Pump,Pump_on_off,Pump_Button",1);
} -> Monitor
(BUTTON Big_Viewport,CRT,Fuel_system,E_3_Pump,Pump_on_off,Pump_Button On)
{ Drive_Button("Big_Viewport,CRT,Fuel_system,E_3_Pump,Pump_on_off,Pump_Button",2);
} -> Monitor
(BUTTON Big_Viewport,CRT,Fuel_system,E_4_Pump,Pump_on_off,Pump_Button Off)
{ Drive_Button("Big_Viewport,CRT,Fuel_system,E_4_Pump,Pump_on_off,Pump_Button",1);
} -> Monitor
(BUTTON Big_Viewport,CRT,Fuel_system,E_4_Pump,Pump_on_off,Pump_Button On)
{ Drive_Button("Big_Viewport,CRT,Fuel_system,E_4_Pump,Pump_on_off,Pump_Button",2);
} -> Monitor
(BUTTON Big_Viewport,CRT,Fuel_system,RH_Ext_Tank_P_2,Pump_on_off,Pump_Button Off)
{ Drive_Button("Big_Viewport,CRT,Fuel_system,RH_Ext_Tank_P_2,Pump_on_off,Pump_Button",1);
} -> Monitor
(BUTTON Big_Viewport,CRT,Fuel_system,RH_Ext_Tank_P_2,Pump_on_off,Pump_Button On)
{ Drive_Button("Big_Viewport,CRT,Fuel_system,RH_Ext_Tank_P_2,Pump_on_off,Pump_Button",2);
} -> Monitor
(BUTTON Big_Viewport,CRT,Fuel_system,LH_Ext_Tank_P,Pump_on_off,Pump_Button Off)
{ Drive_Button("Big_Viewport,CRT,Fuel_system,LH_Ext_Tank_P,Pump_on_off,Pump_Button",1);
} -> Monitor
(BUTTON Big_Viewport,CRT,Fuel_system,LH_Ext_Tank_P,Pump_on_off,Pump_Button On)
{ Drive_Button("Big_Viewport,CRT,Fuel_system,LH_Ext_Tank_P,Pump_on_off,Pump_Button",2);
} -> Monitor
(BUTTON Big_Viewport,CRT,Fuel_system,LH_Ext_Tank_P,Pump_on_off,Pump_Button Off)
{ Drive_Button("Big_Viewport,CRT,Fuel_system,LH_Ext_Tank_P,Pump_on_off,Pump_Button",1);
} -> Monitor
(BUTTON Big_Viewport,CRT,Fuel_system,LH_Ext_Tank_P,Pump_on_off,Pump_Button On)
Drive_Button("Big_Viewport_CRTrawer_system.Tank_4.Filler.Refuel.Cap.Filler",1);

} -> Monitor
{
 Drive_Button("Big_Viewport_CRTrawer_system.Tank_4.Filler.Refuel.Cap.Filler",2);
}

} -> Monitor
(BUTTON Big_Viewport_CRTrawer_system.RH_Aux.Filler.Refuel.Cap.Filler 'off')
{
 Drive_Button("Big_Viewport_CRTrawer_system.RH_Aux.Filler.Refuel.Cap.Filler",1);
}

} -> Monitor
(BUTTON Big_Viewport_CRTrawer_system.RH_Aux.Filler.Refuel.Cap.Filler 'on')
{
 Drive_Button("Big_Viewport_CRTrawer_system.RH_Aux.Filler.Refuel.Cap.Filler",2);
}

} -> Monitor
(BUTTON Big_Viewport_CRTrawer_system.LH_Aux.Filler.Refuel.Cap.Filler 'off')
{
 Drive_Button("Big_Viewport_CRTrawer_system.LH_Aux.Filler.Refuel.Cap.Filler",1);
}

} -> Monitor
(BUTTON Big_Viewport_CRTrawer_system.LH_Aux.Filler.Refuel.Cap.Filler 'on')
{
 Drive_Button("Big_Viewport_CRTrawer_system.LH_Aux.Filler.Refuel.Cap.Filler",2);
}

} -> Monitor
(BUTTON Big_Viewport_CRTrawer_system.E_1_Pump.Pump_on_off.Pump_Button 'off')
{
 Drive_Button("Big_Viewport_CRTrawer_system.E_1_Pump.Pump_on_off.Pump_Button",1);
}

} -> Monitor
(BUTTON Big_Viewport_CRTrawer_system.E_1_Pump.Pump_on_off.Pump_Button 'on')
{
 Drive_Button("Big_Viewport_CRTrawer_system.E_1_Pump.Pump_on_off.Pump_Button",2);
}

} -> Monitor
(BUTTON Big_Viewport_CRTrawer_system.E_2_Pump.Pump_on_off.Pump_Button 'off')
{
 Drive_Button("Big_Viewport_CRTrawer_system.E_2_Pump.Pump_on_off.Pump_Button",1);
}

} -> Monitor
(BUTTON Big_Viewport_CRTrawer_system.E_2_Pump.Pump_on_off.Pump_Button 'on')
{
 Drive_Button("Big_Viewport_CRTrawer_system.E_2_Pump.Pump_on_off.Pump_Button",2);
}

} -> Monitor
(BUTTON Big_Viewport_CRTrawer_system.E_3_Pump.Pump_on_off.Pump_Button 'off')
{
 Drive_Button("Big_Viewport_CRTrawer_system.E_3_Pump.Pump_on_off.Pump_Button",1);
}

} -> Monitor
(BUTTON Big_Viewport_CRTrawer_system.E_3_Pump.Pump_on_off.Pump_Button 'on')
{
 Drive_Button("Big_Viewport_CRTrawer_system.E_3_Pump.Pump_on_off.Pump_Button",2);
}

} -> Monitor
(BUTTON Big_Viewport_CRTrawer_system.E_4_Pump.Pump_on_off.Pump_Button 'off')
{
 Drive_Button("Big_Viewport_CRTrawer_system.E_4_Pump.Pump_on_off.Pump_Button",1);
}

} -> Monitor
(BUTTON Big_Viewport_CRTrawer_system.E_4_Pump.Pump_on_off.Pump_Button 'on')
{
 Drive_Button("Big_Viewport_CRTrawer_system.E_4_Pump.Pump_on_off.Pump_Button",2);
}

} -> Monitor
(BUTTON Big_Viewport_CRTrawer_system.RH_Int_Tank.P.2.Pump_on_off.Pump_Button 'off')
{
 Drive_Button("Big_Viewport_CRTrawer_system.RH_Int_Tank.P.2.Pump_on_off.Pump_Button",1);
}

} -> Monitor
(BUTTON Big_Viewport_CRTrawer_system.RH_Int_Tank.P.2.Pump_on_off.Pump_Button 'on')
{
 Drive_Button("Big_Viewport_CRTrawer_system.RH_Int_Tank.P.2.Pump_on_off.Pump_Button",2);
}

} -> Monitor
(BUTTON Big_Viewport_CRTrawer_system.LH_Int_Tank_P.Pump_on_off.Pump_Button 'off')
{
 Drive_Button("Big_Viewport_CRTrawer_system.LH_Int_Tank_P.Pump_on_off.Pump_Button",1);
}

} -> Monitor
(BUTTON Big_Viewport_CRTrawer_system.LH_Int_Tank_P.Pump_on_off.Pump_Button 'on')
{
} -> Monitor
(BUTTON key_F2 'F2')
{
    if (Balance==1)
    {
        Hide_Frame("Big_Viewport_CRT":"Fuel_Balances");
        Balance=0;
    }
    if (Engine_Zoom==1)
    {
        Hide_Frame("Big_Viewport_CRT":"Engine_Zoomed");
        Engine_Zoom=0;
    }
    if (Fuel_Zoom==1)
    {
        Hide_Frame("Big_Viewport_CRT":"Fuel_Zoomed");
        Hide_Frame("".""Tank_Light_2_Test");
        Fuel_Zoom=0;
    }
    if (Fuel_System==1)
    {
        Hide_Frame("Big_Viewport_CRT":"Fuel_System");
        Hide_Frame("".""Tank_Light_2_Test");
        Fuel_System=0;
    }
}
*Set_Current_Layer(Layer1)+*

    Drive_Light(m"E1_Energy_ref_Energy_Reference_Energy_Ref_Light",2);
    Drive_Light(m"E2_Energy_ref_Energy_Reference_Energy_Ref_Light",1);
    Drive_Light(m"E3_Energy_ref_Energy_Reference_Energy_Ref_Light",2);
    Drive_Light(m"E4_Energy_ref_Energy_Reference_Energy_Ref_Light",0);

    Drive_Light(m"E1_Energy_ref_Energy_Reference_Hs_Plot_Ref_Hp_s_plot_Actual_Engine",1);
    Drive_Light(m"E2_Energy_ref_Energy_Reference_Hs_Plot_Ref_Hp_s_plot_Actual_Engine",2);
    Drive_Light(m"E3_Energy_ref_Energy_Reference_Hs_Plot_Ref_Hp_s_plot_Actual_Engine",1);
    Drive_Light(m"E4_Energy_ref_Energy_Reference_Hs_Plot_Ref_Hp_s_plot_Actual_Engine",1);

    f = 1.0;

Channel_Member_Put_Buffer("Session_Common",m"Location",MEMBER_DIMEN(1),0.CHANNEL_WRITE);

    Drive_Light(m"Pot_Lights",2);
    Drive_Light(m"Pot_Lights_1",2);
    Drive_Light(m"Engine_Light_2",2);
    Drive_Switch(m"Fuel_Overview_Sw",1);
    Drive_Switch(m"Engine_Selector",1);
    Drive_Switch(m"Engine_Selector",1);
    Drive_Switch(m"Engine_Selector",1);
    Drive_Switch(m"Engine_Selector",1);
    Drive_Light(m"".""Labels",3);
}
) -> Monitor
(BUTTON key_F3 'F3')
{
    if (Balance==1)
    {
        Hide_Frame("Big_Viewport_CRT":"Fuel_Balances");
        Balance=0;
    }
    if (Engine_Zoom==1)
    {
        Hide_Frame("Big_Viewport_CRT":"Engine_Zoomed");
        Engine_Zoom=0;
    }
    if (Fuel_Zoom==1)
    {
        Hide_Frame("Big_Viewport_CRT":"Fuel_Zoomed");
        Hide_Frame("".""Tank_Light_2_Test");
        Fuel_Zoom=0;
    }
    if (Fuel_System==1)
    {
        Hide_Frame("Big_Viewport_CRT":"Fuel_System");
        Hide_Frame("".""Tank_Light_2_Test");
    }
Fuel_System=0;

{ Set_Current_LayerLayer1;
}

Channel_Member_Put_Buffer"Session_Common".**."Location".MEMBER_DIMEN(1,6,1.CHAN_FLOAT,&f1.CHANNEL_WRITE);

Drive_Light"Engine_Light_1":1;
Drive_Light"Engine_Light_2":3;
Drive_Light"Engine_Light_3":1;
Drive_Light"Engine_Light_4":1;
Drive_Light"Pot_Lights_1":2;
Drive_Light"Pot_Lights_2":2;
Drive_Switch"Fuel_Overview_Sw":1;
Drive_Switch"Engine_Selector1":13;
Drive_Switch"Engine_Selector2":13;
Drive_Switch"Engine_Selector3":13;
Drive_Switch"Engine_Selector4":13;
Drive_Switch"Engine_Selector":14;

Drive_Light"Labels":3;
-> Monitor
(BUTTON key_F12 'F12')
{
  Remove_Frame("");
  Exit_Ref("bh");
}

} -> EXIT
(BUTTON key_F5 'F5')
{
  if (Balance==0)|
    Appear_Frame("Big_Viewport_CRT","Fuel_Balances");
    Balance=1;
  |
  if (Engine_zoom==1)|
    Hide_Frame("Big_Viewport_CRT","Engine_Zoomed");
    Engine_Zoom=0;
  |
  if (Fuel_Zoom==1) |
    Hide_Frame("Big_Viewport_CRT","Fuel_Zoomed");
    Hide_Frame("Tank_Light_2_Text");
    Fuel_Zoom=0;
  |
  if (Fuel_system==1)|
    Hide_Frame("Big_Viewport_CRT","Fuel_system");
    Hide_Frame("Tank_Light_2_Text");
    Fuel_system=0;
  |
  Drive_Switch("Fuel_Overview_Sw",3);

Channel_Member_Get_Buffer"Fuel_System_Tanks".**."Tank_1_Amount".MEMBER_DIMEN(1,0,1.CHAN_FLOAT,&Tank_Height1.CHANNEL_READ);
  Tank_Height1=8385;

Channel_Member_Put_Buffer"Fuel_System_Tanks".**."Tank_1_Amount".MEMBER_DIMEN(1,0,1.CHAN_FLOAT,&Tank_Height1.CHANNEL_WRITE);
Channel_Member_Get_Buffer("Fuel_Balance", ",", "Tank_1_Amount", MEMBER_DIMEN(1), 0, 1, CHAN_FLOAT, Tank_Height1, 1, CHANNEL_READ);
  Tank_Height1=6985;

Channel_Member_Put_Buffer("Fuel_System_Tanks", ",", "Tank_1_Amount", MEMBER_DIMEN(1), 0, 1, CHAN_FLOAT, Tank_Height1, 1, CHANNEL_WRITE);

Set_Current_Color(2);

Change_Color("Big_Viewport_CRT.Fuel_Balances.Balance_1.2", Balance_Main_1.2, "RH_Balance");
Change_Color("Big_Viewport_CRT.Fuel_Balances.Balance_1.2", Balance_Main_1.2, "LH_Scale");
Change_Color("Big_Viewport_CRT.Fuel_Balances.Balance_1.4", Balance_Main_1.4, "Tank_1_Balance");
Change_Color("Big_Viewport_CRT.Fuel_Balances.Balance_1.4", Balance_Main_1.4, "Tank_4_Balance");
Change_Color("Big_Viewport_CRT.Fuel_Balances.Balance_2.3", Balance_Main_2.3, "Tank_2_Balance");
Change_Color("Big_Viewport_CRT.Fuel_Balances.Balance_2.3", Balance_Main_2.3, "Tank_3_Balance");
Change_Color("Big_Viewport_CRT.Fuel_Balances.Balance_3.4", Balance_Main_3.4, "Tank_4_Balance");
Change_Color("Big_Viewport_CRT.Fuel_Balances.Balance_3.4", Balance_Main_3.4, "Tank_3_Balance");

Drive_Lights(""."", "Fuel_Lights", 1, 2);
Drive_Lights(""."", "Fuel_Lights", 2);

Drive_Switch(4, "Engine_Selec");
Drive_Switch(4, "Engine_Selec");
Drive_Switch(4, "Engine_Selec");
Drive_Switch(4, "Engine_Selec");

Drive_Switch(4, "Engine_Selec");
Drive_Switch(4, "Engine_Selec");
Drive_Switch(4, "Engine_Selec");
Drive_Switch(4, "Engine_Selec");

if (Balance==0)(
  Appearance("Big_Viewport_CRT", "Fuel_Balances");
  Balance=1;
  if (Engine_Zoom==1)(
    Hide_Frame("Big_Viewport_CRT", "Engine_Zoomed");
    Engine_Zoom=0;
  }
  if (Fuel_Zoom==1) {
    Hide_Frame("Big_Viewport_CRT", "Fuel_Zoomed");
    Hide_Frame(""."", "Tank_Light_2_Tes");
    Fuel_Zoom=0;
  }
  if (Fuel_System==1) {
    Hide_Frame("Big_Viewport_CRT", "Fuel_system");
    Hide_Frame(""."", "Tank_Light_2_Tes");
  }
}

Channel_Member_Get_Buffer("Fuel_System_Tanks", ",", "Tank_1_Amount", MEMBER_DIMEN(1), 0, 1, CHAN_FLOAT, Tank_Height1, 1, CHANNEL_READ);
  Tank_Height1=6985;

Channel_Member_Put_Buffer("Fuel_System_Tanks", ",", "Tank_1_Amount", MEMBER_DIMEN(1), 0, 1, CHAN_FLOAT, Tank_Height1, 1, CHANNEL_WRITE);

Channel_Member_Get_Buffer("Fuel_System_Tanks", ",", "Tank_2_Amount", MEMBER_DIMEN(1), 0, 1, CHAN_FLOAT, Tank_Height2, 1, CHANNEL_READ);
  Tank_Height2=7735;

Channel_Member_Put_Buffer("Fuel_System_Tanks", ",", "Tank_2_Amount", MEMBER_DIMEN(1), 0, 1, CHAN_FLOAT, Tank_Height2, 1, CHANNEL_WRITE);
Set_Current_Color(3);

Change_Color("BigViewport_CRT_Fuel_Balances.Balance_1_2.Balance_Main_1_2".,"RH_Balance");
Change_Color("BigViewport_CRT_Fuel_Balances.Balance_1_2.Balance_Main_1_2".,"LH_Scale");
Set_Current_Color(1);

Change_Color("BigViewport_CRT_Fuel_Balances.Balance_1_4.Balance_Main_1_4".,"Tank_1_Balance");
Change_Color("BigViewport_CRT_Fuel_Balances.Balance_1_4.Balance_Main_1_4".,"Tank_4_Balance");
Change_Color("LH_Balance");
Change_Color("RH_Balance");
Set_Current_Color(2);

Change_Color("BigViewport_CRT_Fuel_Balances.Balance_2_3.Balance_Main_2_3".,"Tank_2_Balance");
Change_Color("BigViewport_CRT_Fuel_Balances.Balance_2_3.Balance_Main_2_3".,"Tank_3_Balance");
Change_Color("BigViewport_CRT_Fuel_Balances.Balance_3_4.Balance_Main_3_4".,"Balance_4");
Change_Color("BigViewport_CRT_Fuel_Balances.Balance_3_4.Balance_Main_3_4".,"Balance_4");

Drive_Light("Pot_Lights_1_2");
Drive_Light("Pot_Lights_1_2");

Drive_Switch("Fuel_Overview_Sw",3);
Drive_Switch("Engine_Selector",14);
Drive_Switch("Engine_Selector",14);
Drive_Switch("Engine_Selector",14);

Drive_Switch("E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
Drive_Switch("E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
Drive_Switch("E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);
Drive_Switch("E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp_s_plot.Energy_Select_sw",8);

} -> Monitor
(BUTTON key_F7'F7')
{
  Drive_Switch("Fuel_Overview_Sw",3);

  if (Balance==0){
    Appear_Frame("BigViewport_CRT".,"Fuel_Balances");
    Balance=1.
  }
  if (Engine_Zoom==1){
    Hide_Frame("BigViewport_CRT".,"Engine_Zoomed");
    Engine_Zoom=0;
  }
  if (Fuel_Zoom==1) {
    Hide_Frame("BigViewport_CRT".,"Fuel_Zoomed");
    Hide_Frame("Tank_Light_1_2_Test");
    Fuel_Zoom=0;
  }
  if (Fuel_System==1) {
    Hide_Frame("BigViewport_CRT".,"Fuel_system");
    Hide_Frame("Tank_Light_1_2_Test");
  }

Channel_Member_Get_Buffer("Fuel_System_Tanks","Tank_1_Amount",MEMBER_DIMEN(11,0,1.CHAN_FLOAT, &Tank_Height,1,1.CHANNEL_READ);
  Tank_Height=6385;

Channel_Member_Put_Buffer("Fuel_System_Tanks","Tank_1_Amount",MEMBER_DIMEN(11,0,1.CHAN_FLOAT, &Tank_Height,1,1.CHANNEL_WRITE);

Channel_Member_Get_Buffer("Fuel_System_Tanks","Tank_2_Amount",MEMBER_DIMEN(11,0,1.CHAN_FLOAT, &Tank_Height,2,1.CHANNEL_READ);
  Tank_Height2=6335;

Channel_Member_Put_Buffer("Fuel_System_Tanks","Tank_2_Amount",MEMBER_DIMEN(11,0,1.CHAN_FLOAT, &Tank_Height,2,1.CHANNEL_WRITE);
Set_Current_Color(1);
    Change_Color("Big_Viewport_CRT.Fuel_Balances.Balance.2.3", "Balance.Main.2.3", "Tank.2.Balance");
    Change_Color("Big_Viewport_CRT.Fuel_Balances.Balance.2.3", "Balance.Main.2.3", "Tank.2.Fuel");
    Change_Color("Big_Viewport_CRT.Fuel_Balances.Balance.3.4", "Balance.Main.3.4", "Tank.3.Balance");
    Change_Color("Big_Viewport_CRT.Fuel_Balances.Balance.3.4", "Balance.Main.3.4", "Tank.3.Fuel");
}

Drive_Light("Pot_Lights_1", 2);
Drive_Light("Pot_Lights_2", 2);

Drive_Switch("Fuel_Overview_Sw", 3);
Drive_Switch("FuelSelector2", 14);
Drive_Switch("Engine_Selector2", 14);
Drive_Switch("Engine_Selector4", 14);
Drive_Switch("EngineSelector1", 14);

Drive_Switch("E1_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp.s_plot.Energy.Select_sw", 8);
Drive_Switch("E2_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp.s_plot.Energy.Select_sw", 8);
Drive_Switch("E3_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp.s_plot.Energy.Select_sw", 8);
Drive_Switch("E4_Energy_ref.Energy_Reference.Hs_Plot_Ref.Hp.s_plot.Energy.Select_sw", 8);
    } -> Monitor

{ BUTTON key_F8 'F8'
    Drive_Switch("Fuel_Overview_Sw", 3);

    if (Balance==0){
        Appear_Frame("Big_Viewport_CRT", "Fuel.Balances");
        Balance=1;
    }
    if (Engine_Zoom==1){
        Hide_Frame("Big_Viewport_CRT", "Engine.Zoomed");
        Engine_Zoom=0;
    }

    if (Fuel_Zoom==1){
        Hide_Frame("Big_Viewport_CRT", "Fuel.Zoomed");
        Hide_Frame("Tank.Light.2.Test");
        Fuel_Zoom=0;
    }
    if (Fuel_System==1){
        Hide_Frame("Big_Viewport_CRT", "Fuel.System");
        Hide_Frame("Tank.Light.2.Test");
        Fuel_System=0;
    }

    Drive_Switch("Fuel_Overview_Sw", 1);
    Drive_Switch("FuelSelector2", 13);
    Drive_Switch("Engine_Selector2", 14);
    Drive_Switch("Engine_Selector4", 14);
    Drive_Switch("EngineSelector1", 14);

Channel_Member_Get_Buff("Fuel_System_Tanks","Tank.1.Amount",MEMBER_DIMEN(1),0,1,CHAN_FLOAT,&Tank.Height(1),1,CHANNEL_READ);
    Tank_Height=5485;

Channel_Member_Put_Buff("Fuel_System_Tanks","Tank.1.Amount",MEMBER_DIMEN(1),0,1,CHAN_FLOAT,&Tank.Height(1),1,CHANNEL_WRITE);
Channel Member Get Buff "Fuel System Tanks", "Tank_2_Amount", MEMBER_DIMEN(1), 0, 1, CHAN_FLOAT & Tank_Height 2, 1 CHANNEL_READ;

Tank_Height 2 = 5435;

Channel Member Put Buff "Fuel System Tanks", "Tank_2_Amount", MEMBER_DIMEN(1), 0, 1, CHAN_FLOAT & Tank_Height 2, 1 CHANNEL_WRITE;

Set_Current_Color(1);
Change_Color("Big_Viewport begging CRT Fuel Balances Balance_1 4 Balance Main_1 4", "Tank_1_Balance");
Change_Color("Big_Viewport beggign CRT Fuel Balances Balance_1 4 Balance Main_1 4", "Tank_4_Balance");
Change_Color("Big_Viewport CRT Fuel Balances Balance_2 3 Balance Main_2 3", "Tank_2_Balance");
Change_Color("Big_Viewport CRT Fuel Balances Balance_2 3 Balance Main_2 3", "Tank_3_Balance");

Set_Current_Color(2);
Change_Color("Big_Viewport CRT Fuel Balances Balance_1 2 Balance Main_1 2", "RH Balance");
Change_Color("Big_Viewport CRT Fuel Balances Balance_1 2 Balance Main_1 2", " LH Scale");
Change_Color("Big_Viewport CRT Fuel Balances Balance_3 4 Balance Main_3 4", "Balance_4");
Change_Color("Big_Viewport CRT Fuel Balances Balance_3 4 Balance Main_3 4", "Balance_4");

Drive_Light ("Pot_Lights", 1", 2);
Drive_Lights ("Post_Lights", 2);

Drive_Switch("Fuel_Overview_Sw", 2);
Drive_Switch("Engine_Selection", 14);
Drive_Switch("Engine_Selection", 14);
Drive_Switch("Engine_Selection", 14);
Drive_Switch("Engine_Selection", 14);

Drive_Switch("El_Energy_Energy_Reference_Hs_Plot_Ref_Hp_s_plot.Energy_Select_sw", 8);
Drive_Switch("El_Energy_Energy_Reference_Hs_Plot_Ref_Hp_s_plot.Energy_Select_sw", 8);
Drive_Switch("El_Energy_Energy_Reference_Hs_Plot_Ref_Hp_s_plot.Energy_Select_sw", 8);
Drive_Switch("El_Energy_Energy_Reference_Hs_Plot_Ref_Hp_s_plot.Energy_Select_sw", 8);

}) > Monitor
(BUTTON key_F9 'F9')
{

Drive_Light("Big_Viewport CRT", "Fuel_system_LL Valve_Ref Vertical_Valve_Status", 2);
Drive_Light("Big_Viewport CRT", "Fuel_Zoomed_LL Valve_Ref Vertical_Valve_Status", 2);
Drive_Light("LL Valve_Ref Vertical_Valve_Status", 2);
Disconnect_Plug("Big_Viewport CRT", "Fuel_system_LL Valve_Ref Vertical_Valve_State", BUTTON_INPUT_PLUG);
Disconnect_Plug("Big_Viewport CRT", "Fuel_Zoomed_LL Valve_Ref Vertical_Valve_State", BUTTON_INPUT_PLUG);
Disconnect_Plug("LL Valve_Ref Vertical_Valve_State", BUTTON_INPUT_PLUG);

}) > Monitor
(BUTTON key_F10 'F10')
{

Valve_Pos = 1;

Channel Member Put Buff "Fuel Components", "Valve_LL", MEMBER_DIMEN(1), 0, 1, CHAN_FLOAT & Valve_Pos 1, CHANNEL_WRITE;

Drive_Button("Big_Viewport CRT", "Fuel_system_LL Valve_Ref Vertical_Valve_State", 1);
Drive_Button("Big_Viewport CRT", "Fuel_Zoomed_LL Valve_Ref Vertical_Valve_State", 1);
Drive_Button("LL Valve_Ref Vertical_Valve_State", 1);

Drive_Light("Big_Viewport CRT", "Fuel_system_LL Valve_Ref Vertical_Valve_Status", 1);
Drive_Light("Big_Viewport CRT", "Fuel_Zoomed_LL Valve_Ref Vertical_Valve_Status", 1);
Drive_Light("LL Valve_Ref Vertical_Valve_Status", 1);

Connect_Plug("Big_Viewport CRT", "Fuel_system_LL Valve_Ref Vertical_Valve_State", BUTTON_INPUT_PLUG, "Fuel_Components", "Valve_LL", 0.1);
Connect_Plug("Big_Viewport CRT", "Fuel_Zoomed_LL Valve_Ref Vertical_Valve_State", BUTTON_INPUT_PLUG, "Fuel_Components", "Valve_LL", 0.1);
Connect_Plug("LL Valve_Ref Vertical_Valve_State", BUTTON_INPUT_PLUG, "Fuel_Components", "Valve_LL", 0.1);
APPENDIX G: LIST OF VAPS INTEGRATION EDITOR CONNECTIONS FOR EACH FILE USED IN INTERFACE
--- Channel Connections ---

Channel "Fuel_System_Tanks" (warehouse: DCIEM_Channels)
Tank_1_Amount[0] --> Bar Chart "LH_Fuel_in_tank" Value
Tank_1_Amount[0] --> Scale "LH_Scale" Value
Tank_1_FF[0] --> Bar Chart "LH_FF_bar" Value
Tank_1_FF[0] --> Scale "LH_Tank_FF_Scale" Value
Tank_1_Play[0] --> Bar Chart "Dump_Bar_LH" Value
Tank_1_Play[0] --> Scale "Dump_Scale_LH" Value
Tank_2_Amount[0] --> Bar Chart "Fuel_in_tank_RH" Value
Tank_2_Amount[0] --> Scale "RH_Balance" Value
Tank_2_FF[0] --> Bar Chart "RH_FF_bar" Value
Tank_2_FF[0] --> Scale "FF_Scale_RH" Value
Tank_2_Play[0] --> Bar Chart "Dump_Bar_RH" Value
Tank_2_Play[0] --> Scale "Dump_Scale_RH" Value

*** end of list ***
Channel "Fuel_System_Tanks" (warehouse: DCIEM_Channels)
Tank_1_Amount[0] --> Bar Chart "LH_Fuel_inTank" Value
Tank_1_Amount[0] --> Scale "Tank_1_Balance" Value
Tank_1_FF[0] --> Bar Chart "LH_FF_bar" Value
Tank_1_FF[0] --> Scale "LH_Tank_FF_Scale" Value
Tank_1_Play[0] --> Bar Chart "Dump_Bar_LH" Value
Tank_1_Play[0] --> Scale "Dump_Scale_LH" Value
Tank_4_Amount[0] --> Bar Chart "Fuel_inTank_RH" Value
Tank_4_Amount[0] --> Scale "Tank_4_Balance" Value
Tank_4_FF[0] --> Bar Chart "RH_FF_bar" Value
Tank_4_FF[0] --> Scale "FF_Scale_RH" Value
Tank_4_Play[0] --> Bar Chart "Dump_Bar_RH" Value
Tank_4_Play[0] --> Scale "Dump_Scale_RH" Value

*** end of list ***
-- Channel Connections --

Channel "Fuel_System_Tanks" (warehouse: DCIEM_Channels)
  Tank_2_Amount[0] --> Bar Chart "LH_Fuel_in_tank" Value
  Tank_2_Amount[0] --> Scale "Tank_2_Balance" Value
  Tank_2_FF[0] --> Bar Chart "LH_FF_bar" Value
  Tank_2_FF[0] --> Scale "LH_Tank_FF_Scale" Value
  Tank_2_Clay[0] --> Bar Chart "Dump_Bar_LH" Value
  Tank_2_Clay[0] --> Scale "Dump_Scale_LH" Value
  Tank_3_Amount[0] --> Bar Chart "Fuel_in_tank_RH" Value
  Tank_3_Amount[0] --> Scale "Tank_3_Balance" Value
  Tank_3_FF[0] --> Bar Chart "RH_FF_bar" Value
  Tank_3_FF[0] --> Scale "FF_Scale_RH" Value
  Tank_3_Clay[0] --> Bar Chart "Dump_Bar_RH" Value
  Tank_3_Clay[0] --> Scale "Dump_Scale_RH" Value

*** end of list ***
*** Connections in Prototype <DCIEM_Frames/Balance_Main_3_4> ***

-- Channel Connections --

Channel "Fuel_System_Tanks" (warehouse: DCIEM_Channels)
Tank_3_Amount[0] --> Bar Chart "LH_Fuel_in_tank" Value
Tank_3_Amount[0] --> Scale "Balance_3" Value
Tank_3_FF[0] --> Bar Chart "LH_FF_bar" Value
Tank_3_FF[0] --> Scale "LH_Tank_FF_Scale" Value
Tank_3_Play[0] --> Bar Chart "Dump_Bar_LH" Value
Tank_3_Play[0] --> Scale "Dump_Scale_LH" Value
Tank_4_Amount[0] --> Bar Chart "Fuel_in_tank_RH" Value
Tank_4_Amount[0] --> Scale "Balance_4" Value
Tank_4_FF[0] --> Bar Chart "RH_FF_bar" Value
Tank_4_FF[0] --> Scale "FF_Scale_RH" Value
Tank_4_Play[0] --> Bar Chart "Dump_Bar_RH" Value
Tank_4_Play[0] --> Scale "Dump_Scale_RH" Value

*** end of list ***
**Connections in Prototype <SCIEM_Frames/El_Plots>**

-- Channel Connections --

Channel "Fuel_Components" (warehouse: DCIEM_Channels)

E_L_FLOW[0] <-> Reference "E1_Out" (Local_Common) Location[2]
LH_X_FEED_FLOW[0] <-> Reference "LH_X_FEED" (Local_Common) Location[2]
Valve_GL[0] <-> Reference "GL_Flow" (Local_Common) Location[2]
Valve_LL[0] <-> Reference "LL_Flow" (Local_Common) Location[2]

Channel "Local_Common" (warehouse: SVAPS_DATA)

Location[1] <-> Reference "LH_X_FEED" (Local_Common) Location[0]
Location[2] <-> Reference "LL_Flow" (Local_Common) Location[0]
Location[3] <-> Reference "GL_Flow" (Local_Common) Location[0]

Channel "Stimulation" (warehouse: SVAPS_DATA)

cos_0_01_Hz[0] <-> Reference "GL_Flow" (Local_Common) Location[3]
cos_0_01_Hz[0] <-> Reference "LL_Flow" (Local_Common) Location[3]
ramp_10_0_Hz[0] <-> Reference "GL_Flow" (Local_Common) Location[1]
ramp_10_0_Hz[0] <-> Reference "LL_Flow" (Local_Common) Location[1]
ramp_10_0_Hz[0] <-> Reference "LH_X_FEED" (Local_Common) Location[1]
ramp_10_0_Hz[0] <-> Reference "E1_Out" (Local_Common) Location[1]
sin_0_01_Hz[0] <-> Reference "LH_X_FEED" (Local_Common) Location[3]
square_0_1_Hz[0] <-> Reference "E1_Out" (Local_Common) Location[3]

*** end of list ***
--- Channel Connections ---

Channel "Fuel_Components" (warehouse: DCIEM_Channels)
  E_2_Flow[0] <-> Reference "E2_Out" (Local_Common) Location[2]
  LH_X_Feed_Flow[0] <-> Reference "LH_X_Feed" (Local_Common) Location[2]
  Valve_HL[0] <-> Reference "HL_Flow" (Local_Common) Location[2]
  Valve_ML[0] <-> Reference "ML_Flow" (Local_Common) Location[2]

Channel "Local_Common" (warehouse: SVAPS_DATA)
  Location[1] <-> Reference "LH_X_Feed" (Local_Common) Location[0]
  Location[2] <-> Reference "ML_Flow" (Local_Common) Location[0]
  Location[3] <-> Reference "ML_Flow" (Local_Common) Location[0]

Channel "Stimulation" (warehouse: SVAPS_DATA)
  ramp_10_0_Hz[0] <-> Reference "HL_Flow" (Local_Common) Location[1]
  ramp_10_0_Hz[0] <-> Reference "ML_Flow" (Local_Common) Location[1]
  ramp_10_0_Hz[0] <-> Reference "LH_X_Feed" (Local_Common) Location[1]
  ramp_10_0_Hz[0] <-> Reference "E2_Out" (Local_Common) Location[1]
  sin_0_01_Hz[0] <-> Reference "LH_X_Feed" (Local_Common) Location[3]
  square_0_01_Hz[0] <-> Reference "E2_Out" (Local_Common) Location[3]
  square_0_1_Hz[0] <-> Reference "ML_Flow" (Local_Common) Location[3]
  tri_0_01_Hz[0] <-> Reference "HL_Flow" (Local_Common) Location[3]

*** end of list ***
--- Connections in Prototype <DCIEM_Frames/E3_Plots> ---

-- Channel Connections --

Channel *Fuel_Components* (warehouse: DCIEM_Channels)

- E_3_Flow[0] <-> Reference "E3_Out" (Local_Common) Location[2]
- RH_X_Feed_Man[0] <-> Reference "RH_X_Feed" (Local_Common) Location[2]
- Valve_HR[0] <-> Reference "HR_Flow" (Local_Common) Location[2]
- Valve_MR[0] <-> Reference "MR_Flow" (Local_Common) Location[2]

Channel *Local_Common* (warehouse: SVAPS_DATA)

- Location[1] <-> Reference "RH_X_Feed" (Local_Common) Location[0]
- Location[2] <-> Reference "MR_Flow" (Local_Common) Location[0]
- Location[3] <-> Reference "HR_Flow" (Local_Common) Location[0]

Channel *Stimulation* (warehouse: SVAPS_DATA)

- ramp_10_0_Hz[0] <-> Reference "HR_Flow" (Local_Common) Location[1]
- ramp_10_0_Hz[0] <-> Reference "MR_Flow" (Local_Common) Location[1]
- ramp_10_0_Hz[0] <-> Reference "RH_X_Feed" (Local_Common) Location[1]
- ramp_10_0_Hz[0] <-> Reference "E3_Out" (Local_Common) Location[1]
- sin_0_01_Hz[0] <-> Reference "HR_Flow" (Local_Common) Location[3]
- sin_0_01_Hz[0] <-> Reference "MR_Flow" (Local_Common) Location[3]
- sin_0_01_Hz[0] <-> Reference "E3_Out" (Local_Common) Location[3]
- tri_0_01_Hz[0] <-> Reference "MR_Flow" (Local_Common) Location[3]

*** end of list ***
*** Connections in Prototype <DCIEM_Frames/E4_Plots> ***

-- Channel Connections --

Channel "Fuel_Components" (warehouse: DCIEM_Channels)
  E_4_Flow[0] <-> Reference "E4_Out" (Local_Common) Location[2]
  RH_X_Feed_Man[0] <-> Reference "RH_X_Feed" (Local_Common) Location[2]
  Valve_GR[0] <-> Reference "GR_Flow" (Local_Common) Location[2]
  Valve_LR[0] <-> Reference "LR_Flow" (Local_Common) Location[2]

Channel "Local_Common" (warehouse: SVAPS_DATA)
  Location[1] <-> Reference "PH_X_Feed" (Local_Common) Location[0]
  Location[2] <-> Reference "LR_Flow" (Local_Common) Location[0]
  Location[3] <-> Reference "GR_Flow" (Local_Common) Location[0]

Channel "Stimulation" (warehouse: SVAPS_DATA)
  ramp_10_0_Hz[0] <-> Reference "GR_Flow" (Local_Common) Location[1]
  ramp_10_0_Hz[0] <-> Reference "LR_Flow" (Local_Common) Location[1]
  ramp_10_0_Hz[0] <-> Reference "RH_X_Feed" (Local_Common) Location[1]
  ramp_10_0_Hz[0] <-> Reference "E4_Out" (Local_Common) Location[1]
  sin_0_01_Hz[0] <-> Reference "GR_Flow" (Local_Common) Location[3]
  sin_0_01_Hz[0] <-> Reference "LR_Flow" (Local_Common) Location[3]
  sin_0_01_Hz[0] <-> Reference "RH_X_Feed" (Local_Common) Location[3]
  sin_0_01_Hz[0] <-> Reference "E4_Out" (Local_Common) Location[3]

*** end of list ***
-- Channel Connections --

Channel "Session_Common" (warehouse: $SVAPS_DATA)
Location[1] <-> Reference "E1_Energy_ref" (Local_Common) Location[9]
Location[1] <-> Reference "E1_Energy_ref" (Local_Common) Location[10]
Location[1] <-> Reference "E1_Energy_ref" (Local_Common) Location[2]
Location[1] <-> Reference "E2_Energy_ref" (Local_Common) Location[9]
Location[1] <-> Reference "E2_Energy_ref" (Local_Common) Location[9]
Location[1] <-> Reference "E3_Energy_ref" (Local_Common) Location[2]
Location[1] <-> Reference "E4_Energy_ref" (Local_Common) Location[9]
Location[6] <-> Reference "E3_Energy_ref" (Local_Common) Location[3]
Location[11] <-> Reference "E1_Energy_ref" (Local_Common) Location[0]
Location[12] <-> Reference "E1_Energy_ref" (Local_Common) Location[0]
Location[13] <-> Reference "E3_Energy_ref" (Local_Common) Location[0]
Location[14] <-> Reference "E4_Energy_ref" (Local_Common) Location[0]

Channel "Stimulation" (warehouse: $SVAPS_DATA)
ramp_10_0_Hz[0] <-> Reference "E1_Energy_ref" (Local_Common) Location[1]
ramp_10_0_Hz[0] <-> Reference "E2_Energy_ref" (Local_Common) Location[1]
ramp_10_0_Hz[0] <-> Reference "E3_Energy_ref" (Local_Common) Location[1]
ramp_10_0_Hz[0] <-> Reference "E4_Energy_ref" (Local_Common) Location[1]
sin_0_01_Hz[0] <-> Reference "E1_Energy_ref" (Local_Common) Location[5]
sin_0_01_Hz[0] <-> Reference "E2_Energy_ref" (Local_Common) Location[5]
sin_0_01_Hz[0] <-> Reference "E3_Energy_ref" (Local_Common) Location[5]
sin_0_01_Hz[0] <-> Reference "E4_Energy_ref" (Local_Common) Location[5]

*** end of list ***
– Channel Connections –

Channel "Local_Common" (warehouse: $VAPS_DATA)

Location[1] <-> Reference "Engine_Perf_Ref" (Local_Common) Location[1]
Location[5] <-> Reference "Engine_Perf_Ref" (Local_Common) Location[0]
Location[5] <-> Reference "Engine_Etas_Ref" (Local_Common) Location[0]
Location[9] <-> Reference "Hs_Plot_Ref" (Local_Common) Location[1]
Location[9] <-> Reference "Hs_Plot_Ref" (Local_Common) Location[9]
Location[10] <-> Reference "Hs_Plot_Ref" (Local_Common) Location[10]
Location[12] <-> Reference "Hs_Plot_Ref" (Local_Common) Location[0]

*** end of list ***
Connections in Prototype <DCIEM_Frames/Engine_Etas>

-- Channel Connections --

Channel *Local_Common* (warehouse: $VAPS_DATA)

Location[0] --> Collector *Load_Efficiency*#0* "Stim_In"
Location[0] --> Collector *Thermal_Efficiency*#0* "Stim_in"
Location[1] --> Output Field *Thermal_eta* Value
Location[1] --> Collector *Propulsive_Eta_Calc*#0* "Thermal"
Location[1] <-- Collector *Thermal_Efficiency*#0* Output
Location[2] --> Output Field *Load_eta* Value
Location[2] --> Collector *Propulsive_Eta_Calc*#0* "Load"
Location[2] <-- Collector *Load_Efficiency*#0* Output
Location[3] --> Output Field *Propulsive_eta* Value
Location[3] <-- Collector *Propulsive_Eta_Calc*#0* Output
Location[4] --> Collector *Thermal_Efficiency*#0* "Offset"
Location[5] --> Collector *Load_Efficiency*#0* "Offset"

*** end of list ***
Channel "Session_Common" (warehouse: $VAPS_DATA)
Location[1] <-> Reference "E1_Pol_Star_ref" (Local_Common) Location[0]
Location[2] <-> Reference "E2_Pol_Star_ref" (Local_Common) Location[0]
Location[3] <-> Reference "E3_Pol_Star_ref" (Local_Common) Location[0]
Location[4] <-> Reference "E4_Pol_Star_ref" (Local_Common) Location[0]
Location[12] <-> Reference "E2_Pol_Star_ref" (Local_Common) Location[1]
Location[13] <-> Reference "E3_Pol_Star_ref" (Local_Common) Location[1]
Location[14] <-> Reference "E4_Pol_Star_ref" (Local_Common) Location[1]

*** end of list ***
*** Connections in Prototype <DCIEM_Frames/Engine_Perf> ***

-- Channel Connections --

Channel 'Local_Common' (warehouse: SVAPS_DATA)

Location[0] --> Collector 'S#1'  "In"
Location[0] --> Collector 'SFC#0'  "In"
Location[0] --> Collector 'S#0'  "In"
Location[0] --> Collector 'PSFC#0'  "StimIn"
Location[0] --> Collector 'Calculation_of_S#0'  "In"
Location[0] --> Collector 'Calculation_of_SFC#0'  "In"
Location[0] --> Collector 'Calculation_of_PSFC#0'  "StimIn"
Location[1] --> Plot 'S_Plot'  X Data
Location[1] --> Plot 'SFC_Plot'  X Data
Location[2] --> Collector 'SFC#0'  "thruster"
Location[2] --> Collector 'PSFC#0'  "thrust"
Location[2] --> Collector 'Calculation_of_SFC#0'  "thruster"
Location[2] --> Collector 'Calculation_of_PSFC#0'  "thrust"
Location[3] --> Plot 'S_Plot'  Y Data
Location[3] --> Bar Chart 'Actual_Bar_S'  Value
Location[3] --> Collector 'Calculation_of_SFC#0'  "S"
Location[3] --> Collector 'S#1'  Output
Location[3] --> Collector 'Calculation_of_S#0'  Output
Location[3] --> Plot 'SFC_Plot'  Y Data
Location[4] --> Bar Chart 'Actual_Bar_SFC'  Value
Location[4] --> Collector 'Calculation_of_SFC#0'  Output
Location[5] --> Plot 'PSFC_Plot'  Y Data
Location[5] --> Bar Chart 'Actual_Bar_PSFC'  Value
Location[6] --> Collector 'Calculation_of_PSFC#0'  Output
Location[6] --> Collector 'Calculation_of_S#0'  "Offset"
Location[6] --> Collector 'Calculation_of_PSFC#0'  "Offset"
Location[8] --> Output Field 'S_Field'  Value
Location[8] --> Scale 'S_Scale'  Value
Location[8] --> Collector 'SFC#0'  "S"
Location[8] --> Collector 'S#0'  Output
Location[9] --> Output Field 'Scale_Field_SFC'  Value
Location[9] --> Scale 'SFC_Scale'  Value
Location[9] --> Collector 'SFC#0'  Output
Location[10] --> Output Field 'PSFC_Field'  Value
Location[10] --> Scale 'PSFC_Scale'  Value
Location[10] --> Collector 'PSFC#0'  Output

*** end of list ***
Channel *Local_Common* (warehouse: $VAPS\_DATA$)

*** end of list ***
Channel "Session_Common" (warehouse: $VAPS_DATA)
Location[1] <-> Reference "E1_Thrust" (Local_Common) Location[0]
Location[2] <-> Reference "E2_Thrust" (Local_Common) Location[0]
Location[3] <-> Reference "E3_Thrust" (Local_Common) Location[0]
Location[4] <-> Reference "E4_Thrust" (Local_Common) Location[0]

*** end of list ***
--- Channel Connections ---

Channel "Eng_Zoomed_Channel" (warehouse: DCIEM_Channels)
Oil_Amount[0] --> Bar Chart "Oil_Amount" Value

Channel "Fuel_Components" (warehouse: DCIEM_Channels)
E_1_Flow[0] <-> Reference "Engine_Flow" (Local_Common) Location[0]
E_1_Flow[0] <-> Reference "Heater_to_Pumps" (Local_Common) Location[0]
E_1_Flow[0] <-> Reference "Pump_to_LP_Filt" (Local_Common) Location[0]
E_1_Flow[0] <-> Reference "Filter_to_Pumps" (Local_Common) Location[0]
E_1_Flow[0] <-> Reference "TD_to_Pumps" (Local_Common) Location[0]
E_1_Flow[0] <-> Reference "Pumps_to_FCU" (Local_Common) Location[0]
E_1_Flow[0] <-> Reference "TD_man_Drain" (Local_Common) Location[0]
E_1_Flow[0] <-> Reference "Man_Drain_to_E" (Local_Common) Location[0]
E_1_Flow[0] <-> Reference "man_Drain_eng_2" (Local_Common) Location[0]
E_1_Flow[0] <-> Reference "Tank_Engine" (Local_Common) Location[0]
E_1_Flow[0] <-> Reference "Engine_Tank" (Local_Common) Location[0]
E_1_Flow[0] <-> Reference "Tank_Gear" (Local_Common) Location[0]
E_1_Flow[0] <-> Reference "Gear_Heater" (Local_Common) Location[0]
E_1_Flow[0] <-> Reference "Heater_Cooler" (Local_Common) Location[0]
E_1_Flow[0] <-> Reference "Cooler_Tank" (Local_Common) Location[0]
E_1_Flow[0] <-> Reference "Pumps_South" (Local_Common) Location[0]
E_1_Flow[0] <-> Reference "Pumps_F_Enrich" (Local_Common) Location[0]
E_1_Flow[0] <-> Reference "Enrich_FCU" (Local_Common) Location[0]
E_1_Flow[0] <-> Reference "Overboard_Vent" (Local_Common) Location[0]
E_1_Flow[0] <-> Reference "Gear_Pressure" (Local_Common) Location[0]
E_1_Flow[0] <-> Reference "Tank_Pressure" (Local_Common) Location[0]
E_1_Flow[0] <-> Reference "Engine_Pressure" (Local_Common) Location[0]
E_1_Flow[0] <-> Reference "Nozzle_Pressure" (Local_Common) Location[0]
E_1_Flow[0] <-> Reference "Heater_in" (Local_Common) Location[0]
E_1_Flow[0] <-> Reference "Heater_Out" (Local_Common) Location[0]
E_1_Flow[0] <-> Reference "Cooler_in" (Local_Common) Location[0]
E_1_Flow[0] <-> Reference "Cooler_out" (Local_Common) Location[0]
E_1_Flow[0] <-> Reference "TIT" (Local_Common) Location[0]
E_1_Flow[0] <-> Reference "" (Local_Common) Location[0]
E_1_Flow[0] <-> Reference "RPM" (Local_Common) Location[0]
E_1_Flow[0] <-> Reference "Torque" (Local_Common) Location[0]
E_1_Flow[0] <-> Reference "to_engine" (Local_Common) Location[0]
E_1_Flow[0] <-> Reference "To_Engine" (Local_Common) Location[0]
E_1_Flow[0] <-> Reference "Main_therm" (Local_Common) Location[0]

Channel "Session_Common" (warehouse: SVAPS_DATA)
Location[12] --> Frame "Engine_Zoomed" X Position
Location[13] --> Frame "Engine_Zoomed" Y Position

Channel "Stimulation" (warehouse: SVAPS_DATA)
cos_0_01_Hz[0] <-> Reference "Heater_to_Pumps" (Local_Common) Location[1]
cos_0_01_Hz[0] <-> Reference "Pump_to_LP_Filt" (Local_Common) Location[1]
cos_0_01_Hz[0] <-> Reference "Filter_to_Pumps" (Local_Common) Location[1]
cos_0_01_Hz[0] <-> Reference "TD_man_Drain" (Local_Common) Location[1]
cos_0_01_Hz[0] <-> Reference "Man_Drain_to_E" (Local_Common) Location[1]
cos_0_01_Hz[0] <-> Reference "Heater_Cooler" (Local_Common) Location[1]
cos_0_01_Hz[0] <-> Reference "Cooler_Tank" (Local_Common) Location[1]
cos_0_01_Hz[0] <-> Reference "Nozzle_Pressure" (Local_Common) Location[1]
cos_0_01_Hz[0] <-> Reference "Heater_Out" (Local_Common) Location[1]
cos_0_01_Hz[0] <-> Reference "" (Local_Common) Location[1]
cos_0_01_Hz[0] <-> Reference "Torque" (Local_Common) Location[1]
cos_0_01_Hz[0] <-> Reference "To_Engine" (Local_Common) Location[1]
sin_0_01_Hz[0] <-> Reference "Engine_Flow" (Local_Common) Location[1]
sin_0_01_Hz[0] <-> Reference "Pumps_to_FCU" (Local_Common) Location[1]
sin_0_01_Hz[0] <-> Reference "Pumps_South" (Local_Common) Location[1]
sin_0_01_Hz[0] <-> Reference "" (Local_Common) Location[1]
square_u_1_Hz[0] <-> Reference "Engine_Tank" (Local_Common) Location[1]
square_0_1_Hz[0] <-> Reference "Tank_Temperature" (Local_Common) Location[1]
tri_0_01_Hz[0] <-> Reference "Oil_Tank_Sump_Drain" (Local_Common) Location[0]
tri_0_01_Hz[0] <-> Reference "TD_to_Pumps" (Local_Common) Location[1]
tri_0_01_Hz[0] <-> Reference "man_Drain_eng_2" (Local_Common) Location[1]
tri_0_01_Hz[0] <-> Reference "Tank_Gear" (Local_Common) Location[1]
tri_0_01_Hz[0] <-> Reference "Gear_Heater" (Local_Common) Location[1]
tri_0_01_Hz[0] <-> Reference "Pumps_F_Erich" (Local_Common) Location[1]
tri_0_01_Hz[0] <-> Reference "Enrich_FCU" (Local_Common) Location[1]
tri_0_01_Hz[0] <-> Reference "Gear_Temperature" (Local_Common) Location[1]
tri_0_01_Hz[0] <-> Reference "Engine_Temperature" (Local_Common) Location[1]
tri_0_01_Hz[0] <-> Reference "Heater_in" (Local_Common) Location[1]
tri_0_01_Hz[0] <-> Reference "Cooler_in" (Local_Common) Location[1]
tri_0_01_Hz[0] <-> Reference "RPM" (Local_Common) Location[1]
tri_0_01_Hz[0] <-> Reference "to_engine" (Local_Common) Location[1]
tri_0_01_Hz[0] <-> Reference "Main_therm" (Local_Common) Location[1]
tri_0_1_Hz[0] <-> Reference "Overboard_Vent" (Local_Common) Location[1]

*** end of list ***
-- Channel Connections --

Channel "Local_Common" (warehouse: $VAPS_DATA)
  Location[0] --> Bar Chart "Fuel_in_tank" Value
  Location[1] --> Scale "Dump_Scale" Value
  Location[1] --> Bar Chart "FF_bar" Value
  Location[2] --> Bar Chart "Dump_Bar" Value

*** end of list ***
*** Connections in Prototype <DCIEM_Frames/Flow_Meter> ***

-- Channel Connections --

Channel "Local_Common" (warehouse: $VAPE_DATA)
Location[0] --> Collector "Meter_Collector#0" "Input"
Location[1] --> Collector "Meter_Collector#0" "Stim_in"
Location[2] --> Bar Chart "Flow_Fluid" Value
Location[2] --> Scale "Flow_Scale" Value
Location[2] <-- Collector "Meter_Collector#0" Output

*** end of list ***
Channel "Fuel_Components" (warehouse: DCIEM_Channels)
Boost_Pump_3[0] <-> Reference "Tank_3" (Local_Common) Location[4]
Boost_Pump_3[0] --> Light "P3_E3_fl" State
Boost_Pump_4[0] <-> Reference "Tank_4" (Local_Common) Location[4]
Boost_Pump_4[0] --> Light "P4_E4_fl" State
Cap_LH_Ext[0] --> Light "LH_EXT_ref_fl" State
Cap_RH_Ext[0] --> Light "ref_RH_EXT_fl" State
Dump_Pump_4[0] --> Collector "Dumped_L#0" "DumpFl".
E_1_Flow[0] <-> Reference "Engine_1" (Local_Common) Location[4]
E_1_Flow[0] --> Light "E1_fl" State
E_2_Flow[0] <-> Reference "Engine_2" (Local_Common) Location[4]
E_2_Flow[0] --> Light "E2_fl" State
E_3_Flow[0] <-> Reference "Engine_3" (Local_Common) Location[4]
E_3_Flow[0] --> Light "E3_fl" State
E_4_Flow[0] <-> Reference "Engine_4" (Local_Common) Location[4]
E_4_Flow[0] --> Light "E4_Fl" State
Extra_10[0] <-> Reference "APU" (Local_Common) Location[4]
Extra_10[0] --> Light "APU_fl" State
Extra_2[0] --> Bar Chart 'LH_Dump_Bar' Value
Filler_1[0] --> Light "Refuel_1" State
Filler_2[0] --> Light "Refuel_2" State
Filler_3[0] --> Light "Refuel_3" State
Filler_4[0] --> Light "Refuel_4" State
Filler_LH_Aux[0] --> Light "Refuel_LH_Aux" State
Filler_RH_Aux[0] --> Light "Refuel_RH_Aux" State
LH_Aux_Cap_to_TL[0] --> Light "Ext_Cap TL_fl" State
LH_Dump_Man[0] --> Light "LH_Dump" State
LH_X_Feed_Flow[0] --> Light "LH_XFeed_fl" State
Pump_1_Dump[0] --> Collector "Dumped_L#0" "DumpFl"
Pump_1_E1[0] <-> Reference "Tank_1" (Local_Common) Location[4]
Pump_1_E1[0] --> Light "P1_E1_fl" State
Pump_2_E2[0] <-> Reference "Tank_2" (Local_Common) Location[4]
Pump_2_E2[0] --> Light "P2_E2_fl" State
Pump_LH_Aux[0] <-> Reference "LH_aux" (Local_Common) Location[4]
Pump_LH_Aux[0] --> Reference "KL" (Local_Common) Location[0]
Pump_LH_Aux[0] --> Light "P_AUX_KL_fl" State
Pump_RH_Aux[0] <-> Reference "RH_aux" (Local_Common) Location[4]
Pump_RH_Aux[0] --> Reference "KR" (Local_Common) Location[0]
Pump_RH_Aux[0] --> Light "P_AUX_R_fl" State
RH_Aux_Cap_to_TR[0] --> Light "P_EXT_Cap_TR" State
RH_Dump_Man[0] --> Light "RH_Dump" State
RH_Dump_Man[0] --> Collector "Dumped_L#1" "DumpMan"
RH_X_Feed_Man[0] --> Light "RH_X_Feed_fl" State
Valve_GL[0] --> Reference "GL" (Local_Common) Location[0]
Valve_GR[0] --> Reference "GR" (Local_Common) Location[0]
Valve_HL[0] --> Reference "HL" (Local_Common) Location[0]
Valve_I[0] --> Reference "I_vert" (Local_Common) Location[0]
Valve_LL[0] --> Reference "LL" (Local_Common) Location[0]
Valve_LR[0] --> Reference "LR" (Local_Common) Location[0]
Valve_ML[0] --> Reference "ML" (Local_Common) Location[0]
Valve_MR[0] --> Reference "MR" (Local_Common) Location[0]
Valve_O[0] --> Reference "O" (Local_Common) Location[0]
Valve_P[0] --> Reference "P" (Local_Common) Location[0]
Valve_TL[0] --> Reference "TL" (Local_Common) Location[0]
Valve_TL[0] --> Reference "LH_EXT" (Local_Common) Location[4]
Valve_TR[0] --> Reference "TR" (Local_Common) Location[0]
Valve_TR[0] --> Reference "RH_EXT" (Local_Common) Location[4]
Valve_WL[0] --> Reference "WL" (Local_Common) Location[0]
Valve_WR[0] --> Reference "WR" (Local_Common) Location[0]
Valve_XL[0] --> Collector "Dumped_L#0" "X"
Channel 'Session-Common' (warehouse: $VARPS_DATA)

Location[1] --> Reference "Tank_1" (Local_Common) Location[3]
Location[1] --> Reference "Engine_1" (Local_Common) Location[3]
Location[2] --> Reference "Tank_2" (Local_Common) Location[3]
Location[2] --> Reference "Tank_2" (Local_Common) Location[3]
Location[2] --> Reference "Engine_2" (Local_Common) Location[3]
Location[3] --> Reference "Tank_3" (Local_Common) Location[3]
Location[3] --> Reference "Engine_3" (Local_Common) Location[3]
Location[3] --> Reference "Engine_3" (Local_Common) Location[3]
Location[3] --> Reference "Engine_3" (Local_Common) Location[3]
Location[3] --> Reference "Tank_4" (Local_Common) Location[3]
Location[4] --> Reference "Tank_4" (Local_Common) Location[3]

*** end of list ***
** Channel Connections **

Channel "Local_Common" (warehouse: $VAPS_DATA)

Location[0] --> Collector "Meter_Collector#0" "Input"
Location[1] --> Collector "Meter_Collector#0" "Stim_in"
Location[2] --> Bar Chart "Therm_Fluid" Value
Location[2] --> Scale "Therm_Scale" Value
Location[2] <-- Collector "Meter_Collector#0" Output

*** end of list ***
Channel "Fuel_Components" (warehouse: DCIEM_Channels)

Boost_Pump_3[0] <-> Reference "E_3_Pump" (Local_Common) Location[0]
Boost_Pump_3[0] <-> Reference "T1_Width" (Local_Common) Location[0]
Boost_Pump_3[0] <-> Reference "T_3_Width" (Local_Common) Location[0]
Boost_Pump_3[0] <-> Reference "Light_Tank_3_Width" (Local_Common) Location[0]
Boost_Pump_3[0] --> Collector "E3_Flow#1" "Pump_e3"
Boost_Pump_3[0] --> Collector "RH_X_Feed_Pump#0" "T3"
Boost_Pump_4[0] <-> Reference "E_4_Pump" (Local_Common) Location[0]
Boost_Pump_4[0] <-> Reference "T1_Width" (Local_Common) Location[0]
Boost_Pump_4[0] <-> Reference "T_4_Width" (Local_Common) Location[0]
Boost_Pump_4[0] <-> Reference "Light_Tank_4_Width" (Local_Common) Location[0]
Boost_Pump_4[0] --> Collector "E4_Flow#1" "Pump_e4"
Boost_Pump_4[0] --> Collector "RH_X_Feed_Pump#0" "T4"
Cap_LH_Ext[0] --> Collector "LH_Ext_To_Cap_Flow#0" "Cap"
Cap_LH_Ext[0] --> Collector "LH_Ext_Man_Cap_To_TL#0" "Cap"
Cap_LH_Ext[0] --> Button "LH_Cap_Button" State
Cap_RH_Ext[0] <-> Collector "RH_Ext_To_Cap_Flow#0" "Cap"
Cap_RH_Ext[0] --> Collector "RH_Ext_Man_Cap_To_TR#0" "Cap"
Cap_RH_Ext[0] --> Button "RH_Cap_Button" State
Dump_Pump_3[0] <-> Reference "Tank_3_Dump_Pump" (Local_Common) Location[0]
Dump_Pump_3[0] <-> Reference "T3_Dump" (Local_Common) Location[0]
Dump_Pump_3[0] <-> Reference "Dump_Pump_3_Width" (Local_Common) Location[0]
Dump_Pump_3[0] --> Collector "Tank_3_Dump" State
Dump_Pump_3[0] --> Collector "RH_Dump_Manifold#1" "T3"
Dump_Pump_4[0] <-> Reference "Tank_4_Dump_Pump" (Local_Common) Location[0]
Dump_Pump_4[0] <-> Reference "T4_Dump_A" (Local_Common) Location[0]
Dump_Pump_4[0] <-> Reference "T4_Dump_B" (Local_Common) Location[0]
Dump_Pump_4[0] <-> Reference "Dump_Pump_4_Width" (Local_Common) Location[0]
Dump_Pump_4[0] --> Collector "Tank_4_Dump" State
Dump_Pump_4[0] --> Collector "RH_Dump_Manifold#1" "T4"
E_1_Flow[0] <-> Reference "E_1_Temp" (Local_Common) Location[0]
E_1_Flow[0] --> Reference "E_1_Flow_Scale" (Local_Common) Location[0]
E_1_Flow[0] --> Reference "E_1_Pressure" (Local_Common) Location[0]
E_1_Flow[0] --> Light "E1_F1" State
E_1_Flow[0] --> Collector "E1_Flow#1" Output
E_2_Flow[0] --> Reference "E_2_Temp" (Local_Common) Location[0]
E_2_Flow[0] --> Reference "E_2_Flow_Scale" (Local_Common) Location[0]
E_2_Flow[0] --> Reference "E_2_Pressure" (Local_Common) Location[0]
E_2_Flow[0] --> Light "E2_F1" State
E_2_Flow[0] --> Collector "E2_Flow#1" Output
E_3_Flow[0] --> Reference "E_3_Flow_Scale" (Local_Common) Location[0]
E_3_Flow[0] --> Reference "E_3_Pressure" (Local_Common) Location[0]
E_3_Flow[0] --> Light "E3_F1" State
E_3_Flow[0] --> Collector "E3_Flow#1" Output
E_4_Flow[0] --> Reference "E_4_Temp" (Local_Common) Location[0]
E_4_Flow[0] --> Reference "E_4_Flow_Scale" (Local_Common) Location[0]
E_4_Flow[0] --> Reference "E_4_Pressure" (Local_Common) Location[0]
E_4_Flow[0] --> Light "E4_F1" State
E_4_Flow[0] --> Collector "E4_Flow#1" Output
Extra_1[0] --> Light "APU_Flow" State
Extra_1[0] --> Collector "APU_Flow#0" Output
Extra_10[0] --> Light "RH_Man_Dump" State
Extra_9[0] --> Light "LH_Man_Dump" State
Filler_1[0] <-> Reference "Tank_1_Filler" (Local_Common) Location[0]
Filler_1[0] <-> Reference "T1_Refill" (Local_Common) Location[0]
Filler_2[0] <-> Reference "Tank_2_Filler" (Local_Common) Location[0]
Filler_2[0] <-> Reference "T2_Refill" (Local_Common) Location[0]
Filler_3[0] <-> Reference "Tank_3_Filler" (Local_Common) Location[0]
Filler_3[0] <-> Reference "T3_Refill" (Local_Common) Location[0]
Filler_4[0] <-> Reference "Tank_4_Filler" (Local_Common) Location[0]
Filler_4[0] <-> Reference "T4_Refill" (Local_Common) Location[0]
Filler_LH_Aux[0] <-> Reference "LH_Aux_Filler" (Local_Common) Location[0]
RH_Dump_Man[0] --> Collector "Ref_man_Pt2#0" "RHD"
RH_Dump_Man[0] <-- Collector "RH_Dump_Manifold#1" Output
RH_XFeed_pt_2[0] --> Collector "RH_X_Feed#0" "Part2"
RH_XFeed_pt_2[0] <-- Collector "RH_X_Feed_Pt2#0" Output
RH_X_Feed_Man[0] <-- Reference "I_Right" (Local_Common) Location[0]
RH_X_Feed_Man[0] --> Light "RH_X_Feed" State
RH_X_Feed_Man[0] --> Collector "E3_Flow#1" "Xfeed"
RH_X_Feed_Man[0] --> Collector "E4_Flow#1" "Xfeed"
RH_X_Feed_Man[0] --> Collector "APU_Flow#0" "RH_X_Feed"
RH_X_Feed_Man[0] --> Collector "LH_X_Feed_Pt2#0" "RHX1"
RH_X_Feed_Man[0] --> Collector "Ref_man_Pt2#0" "RHX"
RH_X_Feed_Man[0] --> Collector "KR_to_SR_Flow#0" "RHX"
RH_X_Feed_Man[0] --> Collector "RH_Ext_Man_Cap_To_TR#0" "RHX"
RH_X_Feed_Man[0] <-- Collector "RH_X_Feed#0" Output
RH_Xfeed_pt_1[0] --> Collector "RH_X_Feed#0" "Part1"
RH_Xfeed_pt_1[0] <-- Collector "RH_X_Feed_Pt1#0" Output
Refuel_Man_Flow[0] --> Light "Refuel_Man_Flow" State
Refuel_Man_Flow[0] --> Collector "RH_X_Feed_Pt1#0" "Ref"
Refuel_Man_Flow[0] --> Collector "KL_to_SL_Flow#0" "Ref"
Refuel_Man_Flow[0] --> Collector "KX_to_SR_Flow#0" "Ref"
Refuel_Man_Flow[0] --> Collector "RH_Ext_Man_Cap_To_TR#0" "REF"
Refuel_Man_Flow[0] --> Collector "LH_X_Feed_Pt1#0" "REF"
Refuel_Man_Flow[0] <-- Collector "Refuel_Manifold_Flow#0" Output
Refuel_pt_1[0] --> Collector "Refuel_Manifold_Flow#0" "Part1"
Refuel_pt_1[0] <-- Collector "Ref_man_Pt1#0" Output
Refuel_pt_2[0] <-- Collector "Refuel_Manifold_Flow#0" "Part2"
Refuel_pt_2[0] <-- Collector "Ref_man_Pt2#0" Output
Refuel_pt_3[0] --> Collector "Refuel_Manifold_Flow#0" "Part1"
Refuel_pt_3[0] <-- Collector "Ref_man_Pt3#0" Output
SPR_Man[0] --> Light "SPR_Flow" State
SPR_Man[0] --> Collector "Ref_man_Pt2#0" "SPR"
Stim_in_Press[0] <-- Reference "E_1_Pressure" (Local_Common) Location[1]
Stim_in_Press[0] <-- Reference "E_2_Pressure" (Local_Common) Location[1]
Stim_in_Press[0] <-- Reference "E_3_Pressure" (Local_Common) Location[1]
Stim_in_Press[0] <-- Reference "E_4_Pressure" (Local_Common) Location[1]
Stim_in_Press[0] <-- Reference "X_Feed_Press_M" (Local_Common) Location[1]
Stim_in_Press[0] <-- Reference "Dump_Pump_1_psi" (Local_Common) Location[1]
Stim_in_Press[0] <-- Reference "Boost_Pump_1_psi" (Local_Common) Location[1]
Stim_in_Press[0] <-- Reference "Boost_Pump_2_psi" (Local_Common) Location[1]
Stim_in_Press[0] <-- Reference "Dump_Pump_2_psi" (Local_Common) Location[1]
Stim_in_Press[0] <-- Reference "RH_Aux_Boost_psi" (Local_Common) Location[1]
Stim_in_Press[0] <-- Reference "Boost_Pump_3_psi" (Local_Common) Location[1]
Stim_in_Press[0] <-- Reference "Dump_Pump_3_psi" (Local_Common) Location[1]
Stim_in_Press[0] <-- Reference "Boost_Pump_4_psi" (Local_Common) Location[1]
Stim_in_Press[0] <-- Reference "Dump_Pump_4_psi" (Local_Common) Location[1]
Stim_in_Press[0] <-- Reference "RH_Ext_A_Pres" (Local_Common) Location[1]
Stim_in_Press[0] <-- Reference "RH_Ext_B_Pres" (Local_Common) Location[1]
Stim_in_Press[0] <-- Reference "RH_Ext_A_Pres" (Local_Common) Location[1]
Stim_in_Press[0] <-- Reference "RH_Ext_B_Pres" (Local_Common) Location[1]
Stim_in_Press[0] <-- Collector "Stimulation#0" Output
Stim_in_Therm[0] <-- Reference "TH_1_Temp" (Local_Common) Location[1]
Stim_in_Therm[0] <-- Reference "E_2_Temp" (Local_Common) Location[1]
Stim_in_Therm[0] <-- Reference "E_3_Temp" (Local_Common) Location[1]
Stim_in_Therm[0] <-- Reference "E_4_Temp" (Local_Common) Location[1]
Stim_in_Therm[0] <-- Collector "Stimulation#1" Output
Tank_3_Flow[0] --> Reference "E_3_Temp" (Local_Common) Location[0]
Valve_GL[0] <-- Reference "GL" (Local_Common) Location[0]
Valve_GL[0] --> Collector "E1_Flow#1" "GL"
Valve_GL[0] --> Collector "E1_Flow#1" "LL"
Valve_GR[0] <-- Reference "GR" (Local_Common) Location[0]
Valve_GR[0] --> Collector "E1_Flow#1" "GR"
Valve_HL[0] <-- Reference "HL" (Local_Common) Location[0]
Valve_HL[0] --> Collector "E2_Flow#1" "HL"
Valve_HR[0] <-- Reference "HR" (Local_Common) Location[0]
Valve_HR[0] --> Collector "E3_Flow#1" "HR"
Valve_I[0] <-> Reference 'I' (Local_Common) Location[0]
Valve_I[0] --> Collector 'LH_X_Feed_Pt2#0' 'I'
Valve_I[0] --> Collector 'RH_X_Feed_Pt2#0' 'I'
Valve_J[0] <-> Reference 'J' (Local_Common) Location[0]
Valve_KL[0] <-> Reference 'KL' (Local_Common) Location[0]
Valve_KL[0] <-> Reference 'KL_Flow' (Local_Common) Location[0]
Valve_KL[0] --> Collector 'LH_X_Feed_Pt1#0' 'K'
Valve_KL[0] --> Collector 'KL_to_SL_Flow#0' 'KL'
Valve_KR[0] <-> Reference 'KR' (Local_Common) Location[0]
Valve_KR[0] <-> Reference 'KR_Flow' (Local_Common) Location[0]
Valve_KR[0] --> Collector 'RH_X_Feed_Pt2#0' 'K'
Valve_KR[0] --> Collector 'KR_to_SR_Flow#0' 'KR'
Valve_LL[0] <-> Reference 'LL' (Local_Common) Location[0]
Valve_LL[0] --> Collector 'LH_X_Feed_Pt1#0' 'L'
Valve_LR[0] <-> Reference 'LR' (Local_Common) Location[0]
Valve_LR[0] --> Collector 'LH_Flow#1' 'LR'
Valve_LR[0] --> Collector 'RH_X_Feed_Pt1#0' 'ML'
Valve_ML[0] <-> Reference 'ML' (Local_Common) Location[0]
Valve_ML[0] --> Collector 'ML#0' 'ML'
Valve_ML[0] --> Collector 'LH_X_Feed_Pt1#0' 'M'
Valve_MR[0] <-> Reference 'MR' (Local_Common) Location[0]
Valve_MR[0] --> Collector 'E3_Flow#1' 'MR'
Valve_M[0] --> Collector 'RH_X_Feed_Pt1#0' 'M'
Valve_N[0] <-> Reference 'N' (Local_Common) Location[0]
Valve_O[0] <-> Reference 'O' (Local_Common) Location[0]
Valve_O[0] --> Collector 'APU_Flow#0' 'O'
Valve_P[0] <-> Reference 'P' (Local_Common) Location[0]
Valve_P[0] --> Reference 'P_Flow' (Local_Common) Location[0]
Valve_P[0] --> Collector 'RH_X_Feed_Pt1#0' 'P'
Valve_P[0] --> Collector 'Ref_man_Pt2#0' 'P'
Valve_Q[0] <-> Reference 'QL' (Local_Common) Location[0]
Valve_Q[0] --> Collector 'LH_Dump_Manifold#0' 'QL'
Valve_Q[0] --> Reference 'QR' (Local_Common) Location[0]
Valve_Q[0] --> Collector 'RH_Dump_Manifold#1' 'QR'
Valve_KL[0] --> Reference 'RL' (Local_Common) Location[0]
Valve_KL[0] --> Collector 'LH_Dump_Manifold#0' 'RL'
Valve_KR[0] --> Reference 'RR' (Local_Common) Location[0]
Valve_KR[0] --> Collector 'RH_Dump_Manifold#1' 'RR'
Valve_SL[0] <-> Reference 'SL' (Local_Common) Location[0]
Valve_SL[0] --> Collector 'Ref_man_Pt1#0' 'SL'
Valve_SL[0] --> Collector 'SL_Flow' (Local_Common) Location[0]
Valve_SL[0] --> Collector 'Ref_man_Pt1#0' 'SL'
Valve_SL[0] --> Collector 'KL_to_SL_Flow#0' 'SL'
Valve_SR[0] <-> Reference 'SR' (Local_Common) Location[0]
Valve_SR[0] --> Collector 'Ref_man_Pt3#0' 'SR'
Valve_SR[0] --> Collector 'KR_to_SR_Flow#0' 'SR'
Valve_TL[0] <-> Reference 'TL' (Local_Common) Location[0]
Valve_TL[0] --> Collector 'LH_X_Feed_Pt2#0' 'TL'
Valve_TL[0] --> Collector 'LH_X_Feed_Pt2#0' 'TL'
Valve_TR[0] <-> Reference 'TR' (Local_Common) Location[0]
Valve_TR[0] --> Collector 'TR_Flow' (Local_Common) Location[0]
Valve_TR[0] --> Collector 'RH_X_Feed_Pt2#0' 'T'
Valve_TR[0] --> Collector 'RH_X_Feed_Pt2#0' 'T'
Valve_UL[0] <-> Reference 'UL' (Local_Common) Location[0]
Valve_UL[0] --> Collector 'Ref_man_Pt1#0' 'UL'
Valve_U[0] --> Collector 'UR' (Local_Common) Location[0]
Valve_U[0] --> Collector 'Ref_man_Pt3#0' 'UR'
Valve_WL[0] <-> Reference 'WL' (Local_Common) Location[0]
Valve_WL[0] --> Reference 'WL_Flow' (Local_Common) Location[0]
Valve_WR[0] <-> Reference 'WR' (Local_Common) Location[0]
Valve_WR[0] --> Reference 'WR_Flow' (Local_Common) Location[0]
Valve_XL[0] <-> Reference 'XL' (Local_Common) Location[0]
Valve_XR[0] --> Reference 'XR' (Local_Common) Location[0]
Valve_Y[0] <-> Reference 'Y' (Local_common) Location[0]
Valve_Y[0] --> Collector 'Ref_man_Pt2#0' 'Y'
Channel *Fuel_System_Tanks* (warehouse: DCIEM_Channels)
   Extra_1[0]  <-> Reference "RH_Ext_Amount" (Local_Common) Location[0] G.30
   Tank_1_Amount[0]  <-> Reference "Tank_1_Amount" (Local_Common) Location[0]
   Tank_2_Amount[0]  <-> Reference "Tank_2_Amount" (Local_Common) Location[0]
   Tank_3_Amount[0]  <-> Reference "Tank_3_Amount" (Local_Common) Location[0]
   Tank_4_Amount[0]  <-> Reference "Tank_4_Amount" (Local_Common) Location[0]
   Tank_LH_Aux_Amount[0]  <-> Reference "LH_Aux_Amount" (Local_Common) Location[0]
   Tank_LH_Ext_Amount[0]  <-> Reference "LH_Ext_Amount" (Local_Common) Location[0]
   Tank_RH_Aux_Amount[0]  <-> Reference "RH_Aux_Amount" (Local_Common) Location[0]

Channel *Session_Common* (warehouse: $VAPS_DATA)
   Location[10]  --> Frame "Fuel_Zoomed" X Position

Channel *Stimulation* (warehouse: $VAPS_DATA)
   cos_0_1_Hz[0]  --> Collector "Stimulation#2" "In"
   sin_0_01_Hz[0]  --> Collector "Stimulation#0" "In"
   tri_0_1_Hz[0]  --> Collector "Stimulation#1" "In"

*** end of list ***
+++ Connections in prototype <DCIEM_Frames/Fuel_system> +++

-- Channel Connections --

Channel "Fuel_Components" (warehouse: DCIEM_Channels)

<table>
<thead>
<tr>
<th>Channel</th>
<th>Reference</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boost_Pump_3[0]</td>
<td>Tank_3_Flow</td>
<td>State</td>
</tr>
<tr>
<td>Boost_Pump_3[0]</td>
<td>Collector</td>
<td>E3_Flow1</td>
</tr>
<tr>
<td>Boost_Pump_3[0]</td>
<td>Collector</td>
<td>RH_X_Feed_Pt1#0</td>
</tr>
<tr>
<td>Boost_Pump_4[0]</td>
<td>Tank_4_Flow</td>
<td>State</td>
</tr>
<tr>
<td>Boost_Pump_4[0]</td>
<td>Collector</td>
<td>E4_Flow1</td>
</tr>
<tr>
<td>Cap_LH_Ext[0]</td>
<td>Collector</td>
<td>LH_Ext_To_Cap_Flow0</td>
</tr>
<tr>
<td>Cap_LH_Ext[0]</td>
<td>Collector</td>
<td>LH_Ext_Man_Cap_To_TL#0</td>
</tr>
<tr>
<td>Cap_LH_Ext[0]</td>
<td>Button</td>
<td>LH_Cap_Button</td>
</tr>
<tr>
<td>Cap_RH_Ext[0]</td>
<td>Collector</td>
<td>RH_Ext_To_Cap_Flow0</td>
</tr>
<tr>
<td>Cap_RH_Ext[0]</td>
<td>Collector</td>
<td>RH_Ext_Man_Cap_To_TR#0</td>
</tr>
<tr>
<td>Cap_RH_Ext[0]</td>
<td>Button</td>
<td>RH_Cap_Button</td>
</tr>
<tr>
<td>Dump_Pump_3[0]</td>
<td>Tank_3_Dump</td>
<td>State</td>
</tr>
<tr>
<td>Dump_Pump_3[0]</td>
<td>Collector</td>
<td>RH_Dump_Manifold1</td>
</tr>
<tr>
<td>Dump_Pump_4[0]</td>
<td>Tank_4_Dump</td>
<td>State</td>
</tr>
<tr>
<td>Dump_Pump_4[0]</td>
<td>Collector</td>
<td>RH_Man_Dump#0</td>
</tr>
<tr>
<td>Dump_Pump_4[0]</td>
<td>Collector</td>
<td>RH_Dump_Manifold1</td>
</tr>
<tr>
<td>E_1_Flow[0]</td>
<td>Flow_E1</td>
<td>State</td>
</tr>
<tr>
<td>E_2_Flow[0]</td>
<td>Flow_E2</td>
<td>State</td>
</tr>
<tr>
<td>E_3_Flow[0]</td>
<td>Flow_E3</td>
<td>State</td>
</tr>
<tr>
<td>E_4_Flow[0]</td>
<td>Flow_E4</td>
<td>State</td>
</tr>
<tr>
<td>Extra_1[0]</td>
<td>APU_Flow</td>
<td>State</td>
</tr>
<tr>
<td>Extra_1[0]</td>
<td>Collector</td>
<td>APU_Flow0</td>
</tr>
<tr>
<td>Extra_1[0]</td>
<td>Collector</td>
<td>RH_Man_Dump0</td>
</tr>
<tr>
<td>Extra_9[0]</td>
<td>Collector</td>
<td>LH_Man_Dump</td>
</tr>
<tr>
<td>Filler_1[0]</td>
<td>Tank_1_Filler</td>
<td>State</td>
</tr>
<tr>
<td>Filler_2[0]</td>
<td>Tank_2_Filler</td>
<td>State</td>
</tr>
<tr>
<td>Filler_3[0]</td>
<td>Tank_3_Filler</td>
<td>State</td>
</tr>
<tr>
<td>Filler_4[0]</td>
<td>Tank_4_Filler</td>
<td>State</td>
</tr>
<tr>
<td>Filler_LH_Aux[0]</td>
<td>LH_Aux_Filler</td>
<td>State</td>
</tr>
<tr>
<td>Filler_RH_Aux[0]</td>
<td>RH_Aux_Filler</td>
<td>State</td>
</tr>
<tr>
<td>KL_to_SL[0]</td>
<td>Flow_SL_KL</td>
<td>State</td>
</tr>
<tr>
<td>KL_to_SL[0]</td>
<td>Collector</td>
<td>LH_X_Feed_Pt1#0</td>
</tr>
<tr>
<td>KL_to_SL[0]</td>
<td>Collector</td>
<td>Ref_man_Pt1#0</td>
</tr>
<tr>
<td>KL_to_SL[0]</td>
<td>Collector</td>
<td>KL_to_SL_flow0</td>
</tr>
<tr>
<td>KR_to_SR[0]</td>
<td>Collector</td>
<td>RH_X_Feed_Pt2#0</td>
</tr>
<tr>
<td>KR_to_SR[0]</td>
<td>Collector</td>
<td>Ref_man_Pt3#0</td>
</tr>
<tr>
<td>KR_to_SR[0]</td>
<td>Collector</td>
<td>KR_to_SR_flow0</td>
</tr>
<tr>
<td>LH_Aux_Cap_to_TL[0]</td>
<td>LH_Ext_Man_Flow</td>
<td>State</td>
</tr>
<tr>
<td>LH_Aux_Cap_to_TL[0]</td>
<td>Collector</td>
<td>LH_Ext_To_Cap_Flow0</td>
</tr>
<tr>
<td>LH_Aux_Cap_to_TL[0]</td>
<td>Collector</td>
<td>LH_X_Feed_Pt2#0</td>
</tr>
<tr>
<td>LH_Aux_Cap_to_TL[0]</td>
<td>Collector</td>
<td>Ref_man_Pt1#0</td>
</tr>
<tr>
<td>LH_Aux_Cap_to_TL[0]</td>
<td>Collector</td>
<td>LH_Ext_Man_Cap_To_TL0</td>
</tr>
<tr>
<td>LH_Aux_Flow[0]</td>
<td>LH_Ext_Tank_Flow</td>
<td>State</td>
</tr>
<tr>
<td>LH_Aux_Flow[0]</td>
<td>Collector</td>
<td>LH_Ext_Man_Cap_To_TL0</td>
</tr>
<tr>
<td>LH_Aux_Flow[0]</td>
<td>Collector</td>
<td>LH_Ext_To_Cap_Flow0</td>
</tr>
<tr>
<td>LH_Dump_Man[0]</td>
<td>LH_Dump_man_Flow</td>
<td>State</td>
</tr>
<tr>
<td>LH_Dump_Man[0]</td>
<td>Collector</td>
<td>LH_Man_Dump0</td>
</tr>
<tr>
<td>LH_Dump_Man[0]</td>
<td>Collector</td>
<td>Ref_man_Pt1#0</td>
</tr>
</tbody>
</table>
LH_XFeed_pt_2[0] --> Collector "LH_X_Feed#0"  "Part_2"
LH_XFeed_pt_2[0] --> Collector "LH_X_Feed_Pt2#0"  Output
LH_X_Feed_Flow[0] --> Collector "El_Flow#3"  "Xfeed"
LH_X_Feed_Flow[0] --> Collector "E2_Flow#1"  "Xfeed"
LH_X_Feed_Flow[0] --> Collector "RH_X_Feed_Pt2#0"  "LX"
LH_X_Feed_Flow[0] --> Collector "KL_to_SL_Flow#0"  "LH_X"
LH_X_Feed_Flow[0] --> Collector "LH_Ext_Man_Cap_To_TR#0"  "LHX"
LH_X_Feed_Flow[0] --> Collector "LH_X_Feed#0"  Output
LH_Xfeed_pt_1[0] --> Collector "LH_X_Feed#0"  "Part_1"
LH_Xfeed_pt_1[0] --> Collector "LH_X_Feed_Pt1#0"  Output
N_Flow[0] --> Light "N_Flow"  State
Pump_1_Dump[0] <-> Reference "Tank_1_Dump_Pump"  (Local_Common) Location[0]
Pump_1_Dump[0] --> Light "Tank_1_Dump_Flow"  State
Pump_1_Dump[0] --> Collector "LH_Man_Dump#0"  "P1_Dump"
Pump_1_Dump[0] --> Collector "LH_Dump_Manifold#0"  "T1"
Pump_1_E1[0] <-> Reference "E_1_Pump"  (Local_Common) Location[0]
Pump_1_E1[0] --> Light "Tank_1_Flow"  State
Pump_1_E1[0] --> Collector "El_Flow#3"  "P1"
Pump_1_E1[0] --> Collector "LH_X_Feed_Pt1#0"  "T1"
Pump_2_Dump[0] <-> Reference "Tank_2_Dump_Pump"  (Local_Common) Location[0]
Pump_2_Dump[0] --> Light "Tank_2_Dump_Flow"  State
Pump_2_Dump[0] --> Collector "LH_Dump_Manifold#0"  "T2"
Pump_2_E2[0] <-> Reference "E_2_Pump"  (Local_Common) Location[0]
Pump_2_E2[0] --> Light "Tank_2_Flow"  State
Pump_2_E2[0] --> Collector "E2_Flow#1"  "Pump_e2"
Pump_2_E2[0] --> Collector "LH_X_Feed_Pt1#0"  "T2"
Pump_LH_Aux[0] <-> Reference "LH_Aux_Pump"  (Local_Common) Location[0]
Pump_LH_Aux[0] --> Light "LH_Aux_Flow"  State
Pump_LH_Aux[0] --> Collector "KL_to_SL_Flow#0"  "LAuxP"
Pump_LH_Ext_A[0] <-> Reference "LH_Ext_Tank_Pump"  (Local_Common) Location[0]
Pump_LH_Ext_A[0] --> Collector "LH_Ext_To_Cap_Flow#0"  "Px1"
Pump_LH_Ext_B[0] <-> Reference "LH_Ext_Tank_P"  (Local_Common) Location[0]
Pump_LH_Ext_B[0] --> Collector "LH_Ext_To_Cap_Flow#0"  "Px2"
Pump_RH_Aux[0] <-> Reference "RH_Aux_Pump"  (Local_Common) Location[0]
Pump_RH_Aux[0] --> Light "RH_Aux_Flow"  State
Pump_RH_Aux[0] --> Collector "RR_to_SR_Flow#0"  "RAuxP"
Pump_RH_Ext_A[0] <-> Reference "RH_Ext_Tank_Pump"  (Local_Common) Location[0]
Pump_RH_Ext_A[0] --> Collector "RH_Ext_To_Cap_Flow#0"  "Px1"
Pump_RH_Ext_B[0] <-> Reference "RH_Ext_Tank_P2"  (Local_Common) Location[0]
Pump_RH_Ext_B[0] --> Collector "RH_Ext_To_Cap_Flow#0"  "Px2"
RH_Aux_Cap_to_TR[0] --> Light "RH_Ext_Man_Flow"  State
RH_Aux_Cap_to_TR[0] --> Collector "RH_Ext_To_Cap_Flow#0"  "TRp"
RH_Aux_Cap_to_TR[0] --> Collector "RH_X_Feed_Pt2#0"  "Rex"
RH_Aux_Cap_to_TR[0] --> Collector "Ref_man_Pt3#0"  "RHex"
RH_Aux_Cap_to_TR[0] --> Collector "RH_Ext_Man_Cap_To_TR#0"  Output
RH_Aux_Flow[0] --> Light "RH_Ext_Tank_Flow"  State
RH_Aux_Flow[0] --> Collector "RH_Ext_Man_Cap_To_TR#0"  "RHEX"
RH_Aux_Flow[0] --> Collector "RH_Ext_To_Cap_Flow#0"  Output
RH_Dump_man[0] --> Light "RH_Dump_man_Flow"  State
RH_Dump_man[0] --> Collector "RH_Man_Dump#0"  "RH_Dump_Man"
RH_Dump_man[0] --> Collector "Ref_man_Pt2#0"  "RHD"
RH_Dump_man[0] --> Collector "RH_Dump_Manifold#1"  Output
RH_XFeed_pt_2[0] --> Collector "RH_X_Feed#0"  "Part2"
RH_XFeed_pt_2[0] --> Collector "RH_X_Feed_Pt2#0"  Output
RH_X_Feed_man[0] --> Collector "E4_Flow#1"  "Xfeed"
RH_X_Feed_man[0] --> Collector "E3_Flow#1"  "Xfeed"
RH_X_Feed_man[0] --> Collector "APU_Flow#0"  "RH_X_Feed"
RH_X_Feed_man[0] --> Collector "LH_X_Feed_Pt2#0"  "RHFL"
RH_X_Feed_man[0] --> Collector "Ref_man_Pt2#0"  "RHX"
RH_X_Feed_man[0] --> Collector "KR_to_SR_Flow#0"  "RH_X"
RH_X_Feed_man[0] --> Collector "RH_Ext_Man_Cap_To_TR#0"  "RHX"
RH_X_Feed_man[0] --> Collector "RH_X_Feed#0"  Output
RH_Xfeed_pt_1[0] --> Collector "RH_X_Feed#0"  "Part1"
RH_Xfeed_pt_1[0] --> Collector "RH_X_Feed_Pt1#0"  Output
Valve_TL[0] <-> Reference "TL" (Local_Common) Location[0]
Valve_TL[0] --> Collector "LH_X_Feed_Pt2#0" "TL"
Valve_TL[0] --> Collector "LH_Ext_Man_Cap_To_TL#0" "TL"
Valve_TR[0] <-> Reference "TR" (Local_Common) Location[0]
Valve_TR[0] --> Collector "RH_X_Feed_Pt2#0" "T"
Valve_TR[0] --> Collector "RH_Ext_Man_Cap_To_TR#0" "TR"
Valve_UL[0] <-> Reference "UL" (Local_Common) Location[0]
Valve_UL[0] --> Collector "Ref_man_Pt1#0" "UL"
Valve UR[0] <-> Reference "UR" (Local_Common) Location[0]
Valve UR[0] --> Collector "Ref_man_Pt3#0" "UR"
Valve WL[0] <-> Reference "WL" (Local_Common) Location[0]
Valve WR[0] --> Reference "WR" (Local_Common) Location[0]
Valve XL[0] <-> Reference "XL" (Local_Common) Location[0]
Valve XR[0] --> Collector "LH_Man_Dump#0" "XL"
Valve XR[0] --> Collector "RH_Man_Dump#0" "XR"
Valve Y[0] <-> Reference "Y" (Local_Common) Location[0]
Valve Y[0] --> Collector "Ref_man_Pt2#0" "Y"

*** end of list ***
--- Channel Connections ---

Channel "Local_Common" (warehouse: SVAPS_DATA)
  Location[0] <-- Switch "Energy_Select_sw" State
  Location[10] --> Primitive Object "Broken_Prop" Scale X
  Location[10] --> Primitive Object "Norm_Engine" Scale X
  Location[10] --> Output Field "Compressor" Move X
  Location[10] --> Output Field "Diffuser" Move X
  Location[10] --> Output Field "Diffuser" Move Y
  Location[10] --> Output Field "Ambient" Move X
  Location[10] --> Primitive Object "2-3_Button" Move X
  Location[10] --> Primitive Object "6-0_Button" Move X
  Location[10] --> Primitive Object "5-6_Button" Move X
  Location[10] --> Primitive Object "4-5_Button" Move X
  Location[10] --> Primitive Object "3-4_Button" Move X
  Location[10] --> Primitive Object "0-1_Button" Move X
  Location[10] --> Primitive Object "Ideal_Engine" Scale X
  Location[10] --> Primitive Object "Ideal_Engine" Scale Y
  Location[10] --> Output Field "Turbine" Move X
  Location[10] --> Output Field "Propeller" Move X
  Location[10] --> Output Field "Aircraft_Speed" Move X

*** end of list ***
Channel "Local_Common" (warehouse: $VAPS_DATA)

Location[0] <--- Switch "Energy_Select_sw" State
Location[10] --> Primitive Object "Broken_Prop" Scale X
Location[10] --> Primitive Object "Norm_Engine" Scale X
Location[10] --> Output Field "Compressor" Move X
Location[10] --> Output Field "Diffuser" Move X
Location[10] --> Output Field "Diffuser" Move Y
Location[10] --> Output Field "Ambient" Move X
Location[10] --> Primitive Object "2-3_Button" Move X
Location[10] --> Primitive Object "6-0_Button" Move X
Location[10] --> Primitive Object "5-6_Button" Move X
Location[10] --> Primitive Object "4-5_Button" Move X
Location[10] --> Primitive Object "3-4_Button" Move X
Location[10] --> Primitive Object "0-1_Button" Move X
Location[10] --> Primitive Object "Ideal_Engine" Scale X
Location[10] --> Primitive Object "Ideal_Engine" Scale Y
Location[10] --> Output Field "Turbine" Move X
Location[10] --> Output Field "Propeller" Move X
Location[10] --> Output Field "Aircraft_Speed" Move X

*** end of list ***
-- Channel Connections --

Channel *Local_Common* (warehouse: $VAPS_DATA$)

Location[1] <-> Reference "1_Backup" (Local_Common) Location[1]
Location[1] <-> Reference "Refuel" (Local_Common) Location[1]
Location[1] <-> Reference "QL" (Local_Common) Location[1]
Location[1] <-> Reference "XL" (Local_Common) Location[1]
Location[1] <-> Reference "RL" (Local_Common) Location[1]
Location[2] <-> Reference "1_Backup" (Local_Common) Location[3]
Location[2] <-> Reference "Refuel" (Local_Common) Location[3]
Location[2] <-> Reference "QL" (Local_Common) Location[3]
Location[2] <-> Reference "XL" (Local_Common) Location[3]
Location[5] <-> Reference "QL" (Local_Common) Location[2]
Location[7] <-> Reference "1_Backup" (Local_Common) Location[2]

*** end of list ***
-- Channel Connections --

Channel "Fuel_Components" (warehouse: DCIEM_Channels)
  Pump_1_Dump[0] <-> Reference "1_Backup" (Local_Common) Location[2]
  Refuel_Man_Flow[0] <-> Reference "Refuel" (Local_Common) Location[2]
  Valve_QL[0] <-> Reference "QL" (Local_Common) Location[2]
  Valve_RL[0] <-> Reference "RL" (Local_Common) Location[2]
  Valve_XL[0] <-> Reference "XL" (Local_Common) Location[2]

Channel "Local_Common" (warehouse: SVAPS_DATA)
  Location[1] <-> Reference "RL" (Local_Common) Location[1]

Channel "Stimulation" (warehouse: SVAPS_DATA)
  cos_0_1_Hz[0] <-> Reference "XL" (Local_Common) Location[3]
  ramp_10_0_Hz[0] <-> Reference "1_Backup" (Local_Common) Location[1]
  ramp_10_0_Hz[0] <-> Reference "Refuel" (Local_Common) Location[1]
  ramp_10_0_Hz[0] <-> Reference "QL" (Local_Common) Location[1]
  ramp_10_0_Hz[0] <-> Reference "XL" (Local_Common) Location[1]
  sin_0_01_Hz[0] <-> Reference "RL" (Local_Common) Location[3]
  square_0_1_Hz[0] <-> Reference "Refuel" (Local_Common) Location[3]
  square_0_1_Hz[0] <-> Reference "QL" (Local_Common) Location[3]
  tri_0_01_Hz[0] <-> Reference "1_Backup" (Local_Common) Location[3]

*** end of list ***
Channel "Local_Common" (warehouse: $VAPS_DATA)
Location[1] <-> Reference "Dump_Man" (Local_Common) Location[1]
Location[1] <-> Reference "DL" (Local_Common) Location[1]
Location[1] <-> Reference "EL" (Local_Common) Location[1]
Location[1] <-> Reference "SL" (Local_Common) Location[1]
Location[1] <-> Reference "FL" (Local_Common) Location[1]
Location[1] <-> Reference "Y" (Local_Common) Location[1]
Location[1] <-> Reference "P" (Local_Common) Location[1]
Location[2] <-> Reference "Dump_Man" (Local_Common) Location[3]
Location[2] <-> Reference "EL" (Local_Common) Location[3]
Location[2] <-> Reference "SL" (Local_Common) Location[3]
Location[2] <-> Reference "FL" (Local_Common) Location[3]
Location[2] <-> Reference "Y" (Local_Common) Location[3]
Location[5] <-> Reference "FL" (Local_Common) Location[2]
Location[7] <-> Reference "EL" (Local_COMMON) Location[2]
Location[8] <-> Reference "DL" (Local_COMMON) Location[2]
Location[9] <-> Reference "Dump_Man" (Local_COMMON) Location[2]

*** end of list ***
Channel "Fuel_Components" (warehouse: DCIEM_Channels)
- Filler_1[0] <-> Reference "DL" (Local_Common) Location[2]
- Filler_2[0] <-> Reference "EL" (Local_Common) Location[2]
- Filler_LH Aux[0] <-> Reference "FL" (Local_Common) Location[2]
- KL to SL[0] <-> Reference "SL" (Local_Common) Location[2]
- LH Dump Man[0] <-> Reference "Dump Man" (Local_Common) Location[2]
- Valve P[0] <-> Reference "P" (Local_Common) Location[2]
- Valve Y[0] <-> Reference "Y" (Local_Common) Location[2]

Channel "Stimulation" (warehouse: SVAPS_DATA)
- cos_0_1_Hz[0] <-> Reference "EL" (Local_Common) Location[3]
- ramp_10_0_Hz[0] <-> Reference "Dump Man" (Local_Common) Location[1]
- ramp_10_0_Hz[0] <-> Reference "DL" (Local_Common) Location[1]
- ramp_10_0_Hz[0] <-> Reference "EL" (Local_Common) Location[1]
- ramp_10_0_Hz[0] <-> Reference "FL" (Local_Common) Location[1]
- ramp_10_0_Hz[0] <-> Reference "SL" (Local_Common) Location[1]
- ramp_10_0_Hz[0] <-> Reference "FL" (Local_Common) Location[1]
- ramp_10_0_Hz[0] <-> Reference "Y" (Local_Common) Location[1]
- ramp_10_0_Hz[0] <-> Reference "P" (Local_Common) Location[1]
- sin_0_01_Hz[0] <-> Reference "FL" (Local_Common) Location[3]
- sin_0_01_Hz[0] <-> Reference "Y" (Local_Common) Location[3]
- tri_0_01_Hz[0] <-> Reference "P" (Local_Common) Location[3]
- tri_0_1_Hz[0] <-> Reference "Dump Man" (Local_Common) Location[3]
- tri_0_1_Hz[0] <-> Reference "DL" (Local_Common) Location[3]
- tri_0_1_Hz[0] <-> Reference "SL" (Local_Common) Location[3]

*** end of list ***
Channel *Local_Common* (warehouse: SVA$PS$ DATA)

Location[1] <-> Reference *GL* (Local_Common) Location[1]
Location[1] <-> Reference *LL* (Local_Common) Location[1]
Location[1] <-> Reference *N* (Local_Common) Location[1]
Location[1] <-> Reference *HL* (Local_Common) Location[1]
Location[1] <-> Reference *ML* (Local_Common) Location[1]
Location[1] <-> Reference *KL* (Local_Common) Location[1]
Location[1] <-> Reference *TL* (Local_Common) Location[1]
Location[1] <-> Reference *IL* (Local_Common) Location[1]
Location[1] <-> Reference *GL* (Local_Common) Location[1]
Location[1] <-> Reference *LL* (Local_Common) Location[1]
Location[1] <-> Reference *N* (Local_Common) Location[1]
Location[1] <-> Reference *HL* (Local_Common) Location[1]
Location[1] <-> Reference *ML* (Local_Common) Location[1]
Location[1] <-> Reference *KL* (Local_Common) Location[1]
Location[1] <-> Reference *TL* (Local_Common) Location[1]
*** Connections in Prototype <DCIEM_Frames/LH-XFeed_Flows> ***

-- Channel Connections --

Channel "Local_Common" (warehouse: $VAPS_DATA)

| Location[1] <-> Reference "GL" (Local_Common) Location[1] |
| Location[1] <-> Reference "LL" (Local_Common) Location[1] |
| Location[1] <-> Reference "N" (Local_Common) Location[1] |
| Location[1] <-> Reference "NL" (Local_Common) Location[1] |
| Location[1] <-> Reference "KL" (Local_Common) Location[1] |
| Location[1] <-> Reference "NLL" (Local_Common) Location[1] |
| Location[2] <-> Reference "GL" (Local_Common) Location[3] |
| Location[2] <-> Reference "LL" (Local_Common) Location[3] |
| Location[2] <-> Reference "NLL" (Local_Common) Location[3] |
| Location[2] <-> Reference "I" (Local_Common) Location[3] |
| Location[7] <-> Reference "HL" (Local_Common) Location[2] |
| Location[8] <-> Reference "N" (Local_Common) Location[2] |
| Location[9] <-> Reference "LL" (Local_Common) Location[2] |
| Location[10] <-> Reference "GL" (Local_Common) Location[2] |

*** end of list ***
*** Connections in Prototype <DCIEM_Frames/LH-XFeed_Flows_Test> ***

-- Channel Connections --

Channel "Fuel_Components" (warehouse: DCIEM_Channels)
Valve_GL[0] <-> Reference "GL" (Local_Common) Location[2]
Valve_HL[0] <-> Reference "HL" (Local_Common) Location[2]
Valve_I[0] <-> Reference "I" (Local_Common) Location[2]
Valve_KL[0] <-> Reference "KL" (Local_Common) Location[2]
Valve_LL[0] <-> Reference "LL" (Local_Common) Location[2]
Valve_ML[0] <-> Reference "ML" (Local_Common) Location[2]
Valve_N[0] <-> Reference "N" (Local_Common) Location[2]
Valve_TL[0] <-> Reference "TL" (Local_Common) Location[2]

Channel "Stimulation" (warehouse: SVAPS_DATA)
cos_0.01_Hz[0] <-> Reference "HL" (Local_Common) Location[3]
cos_0.01_Hz[0] <-> Reference "KL" (Local_Common) Location[3]
cos_0.01_Hz[0] <-> Reference "TL" (Local_Common) Location[3]
ramp_10_0_Hz[0] <-> Reference "GL" (Local_Common) Location[1]
ramp_10_0_Hz[0] <-> Reference "LL" (Local_Common) Location[1]
ramp_10_0_Hz[0] <-> Reference "N" (Local_Common) Location[1]
ramp_10_0_Hz[0] <-> Reference "HL" (Local_Common) Location[1]
ramp_10_0_Hz[0] <-> Reference "ML" (Local_Common) Location[1]
ramp_10_0_Hz[0] <-> Reference "KL" (Local_Common) Location[1]
ramp_10_0_Hz[0] <-> Reference "TL" (Local_Common) Location[1]
ramp_10_0_Hz[0] <-> Reference "I" (Local_Common) Location[1]
square_1.0_Hz[0] <-> Reference "N" (Local_Common) Location[3]
tri_0.01_Hz[0] <-> Reference "GL" (Local_Common) Location[3]
tri_0.01_Hz[0] <-> Reference "LL" (Local_Common) Location[3]
tri_0.01_Hz[0] <-> Reference "I" (Local_Common) Location[3]
tri_0.1_Hz[0] <-> Reference "ML" (Local_Common) Location[3]

*** end of list ***
Channel "Local_Common" (warehouse: $VAPS_DATA)
Location[1] <-> Reference "E_Pump_Pres" (Local_Common) Location[1]
Location[1] <-> Reference "Filler_Flow" (Local_Common) Location[1]
Location[1] <-> Reference "LH_Aux_Pump_Fl" (Local_Common) Location[1]
Location[1] <-> Reference "Filler_Flow" (Local_Common) Location[1]
Location[1] <-> Reference "LH_Aux_Pump_Fl" (Local_Common) Location[1]
Location[5] <-> Reference "Amount" (Local_Common) Location[0]

*** end of list ***
Channel "Fuel_Components" (warehouse: DCIEM_Channels)
  Filler_LH_Aux[0] <-> Reference "Filler_Flow" (Local_Common) Location[2]
  Pump_LH_Aux[0] <-> Reference "E_Pump_Pres" (Local_Common) Location[2]
  Pump_LH_Aux[0] <-> Reference "LH_Aux_Pump_F1" (Local_Common) Location[2]

Channel "Fuel_System_Tanks" (warehouse: DCIEM_Channels)
  Tank_LH_Aux_Amount[0] <-> Reference "Amount" (Local_Common) Location[0]

Channel "Local_Common" (warehouse: $VAPS_DATA)

Channel "Stimulation" (warehouse: $VAPS_DATA)
  cos_0_01_Hz[0] <-> Reference "E_Pump_Pres" (Local_Common) Location[3]
  cos_0_01_Hz[0] <-> Reference "Filler_Flow" (Local_Common) Location[3]
  cos_0_01_Hz[0] <-> Reference "LH_Aux_Pump_F1" (Local_Common) Location[3]
  ramp_10_0_Hz[0] <-> Reference "E_Pump_Pres" (Local_Common) Location[1]
  ramp_10_0_Hz[0] <-> Reference "Amount" (Local_Common) Location[1]
  ramp_10_0_Hz[0] <-> Reference "Filler_Flow" (Local_Common) Location[1]
  ramp_10_0_Hz[0] <-> Reference "LH_Aux_Pump_F1" (Local_Common) Location[1]

*** end of list ***
*** Connections in Prototype <DCIEM_Frames/Large_Tank> ***

-- Channel Connections --

Channel "Local_Common" (warehouse: SVAPS_DATA)

Location[0] --> Bar Chart "Fuel_in_tank" Value
Location[0] --> Collector "l_hr_predictor#0" "Tank_Amount"
Location[0] <--> Collector "Sum_of_Two_Items#0" Output
Location[1] --> Collector "Sum_of_Two_Items#0" "Input_1"
Location[2] --> Collector "Sum_of_Two_Items#0" "Input_2"
Location[3] --> Scale "Dump_Scale" Value
Location[3] --> Bar Chart "FF_bar" Value
Location[3] --> Scale "Fuel_Flow_Scale" Value
Location[3] --> Collector "l_hr_predictor#0" "FF_Total"
Location[3] <--> Collector "Sum_of_Two_Items#1" Output
Location[4] --> Collector "Sum_of_Two_Items#1" "Input_1"
Location[5] --> Collector "Sum_of_Two_Items#1" "Input_2"
Location[6] --> Bar Chart "Dump_Bar" Value
Location[6] --> Collector "l_hr_predictor#0" "FF_Play"
Location[6] <--> Collector "Sum_of_Two_Items#2" Output
Location[7] --> Collector "Sum_of_Two_Items#2" "Input_1"
Location[8] --> Collector "Sum_of_Two_Items#2" "Input_2"
Location[9] --> Scale "Km_Scale" Value
Location[9] --> Scale "Mass_Scale" Value
Location[9] <--> Collector "l_hr_predictor#0" Output

*** end of list ***
*** Connections in Prototype <SCIEM_Frames/Main_Tanks_1_4> ***

-- Channel Connections --

Channel "Local_Common" (warehouse: $VAPS_DATA)
Location[0] --> Bar Chart "Fuel_in_tank" Value
Location[0] <-- Collector "Tank_amount_1_4#0" Output
Location[1] --> Scale "Dump_Scale" Value
Location[1] --> Bar Chart "FF_bar" Value
Location[1] --> Scale "Tank_FF_Scale" Value
Location[1] <-- Collector "FF_calc#0" Output
Location[2] <-- Collector "FF_play_calc#0" Output
Location[3] --> Collector "FF_calc#0" "Thrust"
Location[4] --> Bar Chart "Dump_Bar" Value
Location[4] --> Collector "Tank_amount_1_4#0" "Valve"
Location[4] --> Collector "FF_calc#0" "valve"
Location[4] --> Collector "FF_play_calc#0" "valve"
Location[5] --> Collector "Tank_amount_1_4#0" "Stim_in"
Location[5] --> Collector "FF_play_calc#0" "Stim"

*** end of list ***
--- Channel Connections ---

Channel "Local_Common" (warehouse: $VAPS_DATA)

Location[0] --> Bar Chart "Fuel_in_tank" Value
Location[0] <-- Collector "Tank_amount_2_3#0" Output
Location[1] --> Scale "Dump_Scale" Value
Location[1] --> Bar Chart "FF_bar" Value
Location[1] --> Scale "Tank_FF_Scale" Value
Location[1] <-- Collector "FF_Calc#0" Output
Location[2] <-- Collector "FF_play_Calc#0" Output
Location[3] --> Collector "FF_Calc#0" "Thrust"
Location[4] --> Bar Chart "Dump_Bar" Value
Location[4] --> Collector "Tank_amount_2_3#0" "Valve"
Location[4] --> Collector "FF_Calc#0" "valve"
Location[4] --> Collector "FF_play_Calc#0" "valve"
Location[5] --> Collector "Tank_amount_2_3#0" "Stim_in"
Location[5] --> Collector "FF_play_Calc#0" "Stim"

*** end of list ***
-- Channel Connections --

Channel "Local_Common" (warehouse: $VAPS_DATA)
Location[0] --> Bar Chart "Fuel_in_tank" Value
Location[0] <-- Collector "Tank_amount_Ext#1" Output
Location[1] --> Scale "Dump_Scale" Value
Location[1] --> Bar Chart "FF_bar" Value
Location[1] --> Scale "Tank_FF_Scale" Value
Location[1] <-- Collector "FF_Calc#0" Output
Location[2] <-- Collector "FF_play_Calc#0" Output
Location[3] --> Collector "FF_Calc#0" "Thrust"
Location[4] --> Bar Chart "Dump_Bar" Value
Location[4] --> Collector "Tank_amount_Ext#1" "Valve"
Location[4] --> Collector "FF_Calc#0" "Valve"
Location[4] --> Collector "FF_play_Calc#0" "Valve"
Location[5] --> Collector "Tank_amount_Ext#1" "Stim_in"
Location[5] --> Collector "FF_play_Calc#0" "Stim"

*** end of list ***
Channel Connections

Channel "Local_Common" (warehouse: $VAPS_DATA)
Location[0] --> Bar Chart "Fuel_in_tank" Value
Location[0] <-- Collector "Tank_amount_Aux#0" Output
Location[1] --> Scale "Dump_Scale" Value
Location[1] --> Bar Chart "FF_bar" Value
Location[1] --> Scale "Tank_FF_Scale" Value
Location[1] <-- Collector "FF_Calc#0" Output
Location[2] <-- Collector "FF_play_Calc#0" Output
Location[3] --> Collector "FF_Calc#0" "Thrust"
Location[4] --> Bar Chart "Dump_Bar" Value
Location[4] --> Collector "Tank_amount_Aux#0" "Valve"
Location[4] --> Collector "FF_Calc#0" "valve"
Location[4] --> Collector "FF_play_Calc#0" "valve"
Location[5] --> Collector "Tank_amount_Aux#0" "Stim_in"
Location[5] --> Collector "FF_play_Calc#0" "Stim"

*** end of list ***
*** Connections in Prototype <ECIEM_Frames/Map_Distance> ***

-- Channel Connections --

Channel "Local_Common" (warehouse: $VAPS_DATA)
  Location[0] --> Collector "distance_measurement#0"  "scale_in"
  Location[0] <-- Rotor "Zoom_Rotor"  State
  Location[1] --> Bar Chart "distance_bar"  Value
  Location[1] --> Graphical Object "Max_Distance"  Scale X
  Location[1] --> Graphical Object "Max_Distance"  Scale Y
  Location[1] <-- Collector "distance_measurement#0"  Output

Channel "Session_Common" (warehouse: $VAPS_DATA)
  Location[25] --> Output Field "Distance_data"  Value
  Location[25] --> Scale "Distance_Scale"  Value
  Location[25] --> Output Field "Distance_To_Targ"  Value
  Location[25] --> Collector "distance_measurement#0"  "Input"

*** end of list ***
*** Connections in Prototype <DCIEM_Frames/polar_star_2> ***

-- Channel Connections --

Channel "Local_Common" (warehouse: $VAPS_DATA)
Location[1] <-- Switch "Engine_SELECTOR2" State
Location[2] --> Light "Engine_Light_2" State

Channel "Session_Common" (warehouse: $VAPS_DATA)
Location[2] --> Primitive Object "Goal_State" Scale Y
Location[2] --> Graphical Object "Outer_Alarms" Scale X
Location[2] --> Graphical Object "Outer_Alarms" Scale Y
Location[2] --> Graphical Object "Inner_Alarms" Scale X
Location[2] --> Graphical Object "Inner_Alarms" Scale Y
Location[2] --> Primitive Object "Normal_State" Scale Y

*** end of list ***
*** Connections in Prototype <DCIEM_Frames/Oil_Flow_Meter> ***

-- Channel Connections --

Channel "Local_Common" (warehouse: $VAPS_DATA)
Location[0] --> Collector "Meter_Collector#0" "Input"
Location[1] --> Collector "Meter_Collector#0" "Stim_in"
Location[2] --> Bar Chart "Flow_Fluid" Value
Location[2] --> Scale "Flow_Scale" Value
Location[2] <-- Collector "Meter_Collector#0" Output

*** end of list ***
**Connections in Prototype <DCIEM_Frames/Background2>**

-- Channel Connections --

Channel "Session_Common" (warehouse: $VAPS_DATA)
Location[6] <-- Potentiometer "erase_me_lo_Fuel" Value
Location[7] <-- Potentiometer "eras_me_bad_Prop" Value
Location[10] <-- Potentiometer "horizontal" Value
Location[12] <-- Potentiometer "horizontal_1" Value
Location[13] <-- Potentiometer "vertical_1" Value

*** end of list ***