ATOMOS III, MiTS, COMFORT & MBB II

RESEARCH IN WATERBORNE TRANSPORT AREA 6.3.3/26 (2ND CALL): DEMONSTRATION OF ISC - DISC FINAL REPORT

ID Code: D101.00.01.047.003C
Date: 1998.03.24

CLASSIFICATION AND APPROVAL

Classification: Public after review

DEFINITIONS

Public after Review:

The document may be freely distributed after successful EC review. Publication is governed by the EC Contract and the DISC Consortium Cooperation Agreement

Confidential for the Duration of the Project:

As for 'Confidential', but only for the duration of the Project. After final Project Approval by the EC, status for reports classified 'Confidential for the Duration of the Project' are automatically down-graded to 'Public'.

Confidential:

The document is for use of the Contractors and Sub-Contractors within the DISC Consortium, and shall not be used or disclosed to third parties without the unanimous agreement within the DISC PMC and subsequent EC approval since document classification is part of the EC Contract.

AUTHORS & EDITORS:

The DISC Consortium, represented by

Name:	Signature	Date:
Sven Mathes, ISSUS	_____	
John Koch Nielsen, DMI	_____	
Jørn Engen, KSE	_____	
Eigil Haaland, KNCA	_____	

APPROVAL:

Approved for release by:

Erik Styhr Petersen, SCL
DISC Project Manager _____

DOCUMENT HISTORY AND VERSION CONTROL

ISSUE	INITIALS DATE	PAGE	FILE NAME SHORT DESCRIPTION OF CHANGES
001	ESP 1996.12.04	-	053_001.doc First Issue
001A	ESP 1996.12.05	All	047_001.doc Second Issue
001B	ESP 1997.01.09	All	047_001a.doc Final Issue for Internal Review
001C	ESP 1997.01.20 1997.01.21	All	047_001b.doc Final Issue, Revised in Trondheim Workshop
002	ESP 1997.01.23	New Section 4.1 Added	047_002.doc First Issue, 2nd Workshop Report
002A	ESP 1997-01-27	All	047_002A.doc Second issue, 2nd Workshop report
002B	ØJR 1997-02-10	All	047_002B.doc Third issue, 2nd Workshop report, all comments to date included.
002C	ØJR 1997-02-20	All	047_002C.doc. Fourth issue of 2nd workshop report. Change system/component to architecture, add abbreviation, included comments
002D	ESP 1997.02.24	All	047_002D.doc Final Issue of 2nd Workshop Report. Minor changes of layout etc.
003A	ESP 1997.03.19	All	047_003A First consolida.docted issue of Final Report (3rd Workshop Report).
003B	ESP 1997.04.10	Sec. 4.11.4.3 revised	047_003B.doc Corrected in accordance with revised ISSUS input
003C	ESP 1997.05.18	All	047_003C.doc Final version corrected in accordance with review comments from SINTEF, MARINTEK, MAN/B&W, Lloyd's Register of Shipping

DISCLAIMER

Neither the DISC Consortium nor any of its officers, employees or agents shall be responsible or liable in negligence or otherwise howsoever in respect of any inaccuracy or omission herein.

Without derogating from the generality of the foregoing neither the DISC Consortium nor any of its officers, employees or agents shall be liable for any direct or indirect or consequential loss or damage, personal injury or death, caused by or arising from any information, advice or inaccuracy or omission herein.

FOREWORD

Entirely in line with the rest of the world of advanced electronics, marine automation has undergone a dramatic development since the 2nd World War, and especially since the middle nineteen-sixties. The invention and subsequent application of the microprocessor as a general control device started to take hold in the early part of the nineteen-eighties, and added tremendous momentum to the development of very sophisticated electronic systems for process control. The incredible power and versatility of the PC, combined with an almost incomprehensible low price, seen in the perspective of half a century, or indeed just a decade, has further fueled a hereto unseen development pace.

The current state-of-the-art truly reflects this, with an abundance of systems for all imaginable applications to be found on the world market for marine electronics. Due to the extremely powerful development tools also available for low-cost computers, the market itself has however changed significantly over the last few years - where it until recently took the economic power of large companies to create and implement novel systems, even very small companies, as compared to world-scale, can undertake such work, given sufficient dedication and a 'handful' of skilled persons.

This very diversified market does however suffer from one major problem, namely the lack of a standard for integration of information, and hence for practical integration of systems. The obvious ill effect of this lack is so far that a vendor in almost all cases have to supply hardware as well as software, which of course raises development cost, but also in some cases result in a duplication of already existing hardware, which however can not be utilized due to the missing standardization.

Perhaps less obvious, but perhaps more grave, is the fact that applications that utilizes information across different systems, or applications that could be enhanced if they did have access to cross-system sensor or process values, in general are very difficult and very costly to implement, again due to lack of standardization. The fact that such applications in many cases would be within the safety-related or decision-support area just adds to the magnitude of the issue.

Two further issues underlines the need for standardization of integration of information. The first of these issues is the very visible and easily comprehensible topic of Human Machine Interfaces, or HMI for short. Within the current constraints, every manufacturer is forced to provide his own terminal for system operations, and, indeed, his own more or less proprietary HMI. While acknowledging that many of these are well-functioning, it is at the same time almost certain that they are operated in a manner different from every other system on board the vessel in question. This leads to daily confusion, and might lead to something much more serious in the face of an emergency. Again, standardization would solve this problem, and would further add the great benefit that the HMI of the complete system could be assessed and adjusted to the greatest possible suitability, taking the most likely, and the most important, scenarios of operation into account.

The second of these issues is the question of system coherency and dependability. One of the probable problems with the current state-of-the-art is that it is next to impossible to predict all aspects of performance and behavior of ISC systems building on non-standardized integration, or in some of the worst cases, on integration of a more or less prototype nature. The more and more insistent requirements for formal safety assessment simply cannot be supported in any meaningful way with present practice, or at least not in any economical fashion.

DISC was conceived on this background; an entirely sound recognition of the need for standardization in the field of marine electronics, with Integrated Ship Control as the focal point, but taking all of the above issues into account.

But even though the background justified the creation of DISC, the implementation of the project, however, looked slightly dicey: For DISC to have an impact, it would need to have most of the major European players, and as such, competitors, in the field as partners, and it would need to create consensus among these trend-setting corporations.

Everybody recognized this sensitive issue from the beginning, but in spite of the slight apprehension, a consortium totaling almost 30 important companies was formed early in 1996, as a joint venture between the German-led COMFORT Consortium, the Norwegian-led MiTS Consortium, the French-led MBB II Consortium and the Danish-led ATOMOS III Consortium.

DISC would not have happened, had DG VII (Transport) of the European Commission not decided to take a chance and fund the proposed project, which was untypical in almost every way: A joint venture between four consortia, a very high number of partners irrespectively of the management setup, a remarkably short realization period of just half a year, and a somewhat unusual work method - all the work to be done during three, one-week long workshops.

DISC would not have worked, had the partners not cooperated above and beyond what can normally be expected. Everybody showed up for the first DISC workshop with open minds, and were able to set the basic, commercial concerns aside, at least for the duration of the project. The generic level of DISC was of course an aid to this, but without the goodwill and dedication of the partners, things could rapidly have turned for the worse.

DISC could not have succeeded, had the representatives from SINTEF, ISSUS and Dassault Electronique, Coordinators of MiTS, COMFORT and MBB II, respectively, not carried their part of the burden on the management front, and had they not done it competently.

But it did happen, it did work, and with hindsight, it was a success.

An enormous bulk of work still remains to be done before DISC will have significant impact, in terms of continued development, in terms of practical application, in terms of demonstration, and last but absolutely not least, in terms of dissemination. In spite of all of this, what needs to be noted is the fact that the foundation for a single, open European Standard for Integrated Ship Control has been established, and building on the DISC experience, and the attitudes of those involved, I’m sure that the desired impact will come, sooner rather than later.

1997.03.19
Erik Styhr Petersen
DISC Project Manager

CONTENTS

1. INTRODUCTION 12

1.1 About This Document 12
1.2 About DISC and the DISC Consortium 12
1.2.1 ATOMOS II/ATOMOS III 13
1.2.2 MiTS 14
1.2.3 COMFORT 14
1.2.4 MBB/MBB II 15

2. RESEARCH OBJECTIVES 16

3. DEFINITIONS 17

3.1 Abbreviations 17
3.2 General Terms 19
3.3 Terms Related to Computer Based Systems. 22
3.4 Terms Related To Verification and Validation 24
3.5 Terms Related to Human Machine Interface 25

4. THE DISC 4-LAYER STANDARD REFERENCE MODEL 26

4.1 Guidance on the Anticipated Application of the DISC Standard 26
4.1.1 Initial Steps 26
4.1.2 Implementation 27
4.1.3 Maintenance 28
4.2 The DISC Information Flow Model 29
4.3 The DISC 4-Layer ISC Model 30
4.4 Definition of the Validation & Verification (V&V) Layer of the DISC Standard 31
4.4.1 Introduction 31
4.4.2 Safety Integrity Levels 32
4.5 Requirements to the Verification & Validation Layer of the DISC Standard 36
4.6 Definition of the User/HMI Layer of the DISC Standard 37
4.6.1 Two Levels of the HMI 37
4.6.2 Standards and Guidelines 37
4.7 Requirements to the User/HMI Layer of the DISC Standard 38
4.7.1 General Requirements to User Interface 38
4.7.2 Requirements to Applications 38
4.7.3 Requirements to the Integration of Applications 38
4.8 Definition of the Application Layer of the DISC Standard 40
4.8.1 Application in context 40
4.8.2 Types of Applications 41
4.8.3 Generic Application 42
4.8.4 Specification mechanisms 43
4.8.5 Abstraction Pyramid of the DISC ISC-system 43
4.8.6 Functional Description of Applications (the Function Block concept) 44
4.8.7 General Information Models 46
4.8.8 Examples of Function Block representations 47
4.9 Requirements to Application Layer of the DISC standard. 49
4.10 Definition of the Architecture Layer of the DISC Standard 50
4.10.1 Scope 50
4.10.2 Different views of the system 50
4.10.3 The Application Layer Interface 53
4.11 Requirements to the System Architecture 54
4.11.1 Requirements to the Framework 54

4.11.2 Requirements to the Functional Description 54
4.11.3 Requirements to General Information Models 56
4.11.4 Requirements to Services 56
4.11.5 Requirements to the Technical Implementation 59
4.11.6 Safety enhancement through the use of function blocks 59
4.11.7 Use of SIL in function block systems 59
4.11.8 Object orientation in function blocks 60

5. DEVELOPMENT OF SKELETON STANDARD 61

5.1 Verification & Validation 61
5.1.1 Identification of areas to be described by the Skeleton Standard 61
5.1.2 Applicable Standards 62
5.1.3 Standards to be Adapted 62
5.1.4 Missing Standards 62
5.2 User/HMI Layer 64
5.2.1 Standards and guidelines. 64
5.2.2 Identification of areas to be described by the Skeleton Standard 64
5.2.3 General Description 66
5.2.4 Applicable Standards 67
5.2.5 Standards to be Adapted 68
5.2.6 Missing Standards 68
5.3 Applications Layer 69
5.3.1 Identification of areas to be described by the Skeleton Standard 69
5.3.2 General Description 69
5.3.3 Applicable Standards 69
5.3.4 Standards to be Adapted 69
5.3.5 Missing Standards 69
5.4 Architecture Layer 70
5.4.1 Identification of areas to be described by the Skeleton Standard 70
5.4.2 Applicable Standards 71
5.4.3 Standards to be Adapted 71
5.4.4 Missing Standards 74

6. TECHNOLOGIES 75

6.1 The Verification & Validation Layer 75
6.1.1 The approval process 75
6.1.2 General example of an approval process 76
6.1.3 Approval of DISC layers 77
6.1.4 V&V Techniques 77
6.1.5 Static V&V Techniques 78
6.1.6 Dynamic V&V Techniques 82
6.1.7 General 83
6.1.8 Summary of V&V Techniques 85
6.2 Architecture Layer 86
6.2.1 New technologies introduced 86
6.2.2 Required technologies 86
6.2.3 Emerging technologies with a possible impact on the system architecture 86

7. SPECIFIC FUTURE FUNCTIONS 88

7.1 The Verification & Validation Layer 88
7.2 User/HMI Layer 89
7.2.1 Introduction 89
7.2.2 Shore based support 89
7.2.3 Advanced cargo operation: 90
7.2.4 Emergency response decision support systems 90
7.2.5 Voyage planning 91

7.2.6 Navigation 91
7.2.7 Advanced docking systems 91
7.2.8 Platform monitoring 92
7.2.9 Advanced simulation and training 93
7.2.10 Tele-operations 94
7.3 Applications Layer 95
7.3.1 Scope 95
7.3.2 Navigation and voyage planning 97
7.3.3 Emergency management 99
7.3.4 Cargo Management 101
7.3.5 Administrative applications. 103
7.3.6 Maintenance management system 105
7.3.7 Engine Room , Alarms and Control 106
7.3.8 Information gathering manager 107
7.3.9 Maritime Black Box (MBB) 108
7.3.10 System Manager 109
7.3.11 Human Interface Manager (HMI) 110
7.3.12 Common Database 111
7.4 Architecture Layer 112

8. TOOLS & TECHNIQUES 113

8.1 The Verification & Validation Layer 113
8.2 User/HMI Layer 113
8.2.1 Requirements to techniques 113
8.2.2 Human engineering analysis techniques 113
8.3 Applications Layer 118
8.4 Architecture Layer 118
8.4.1 System Integration Tools 118
8.4.2 Debugging, Commissioning and Test Tools 119
8.4.3 Run-time Tools 119

9. VALIDATION & VERIFICATION: MINIMUM DEMONSTRATION REQUIREMENTS 121

9.1 The Verification & Validation Layer 121
9.1.1 ISC development lifecycle 121
9.1.2 Relationship between development lifecycle and V&V principles 123
9.1.3 Validation planning 125
9.1.4 Minimum V&V techniques for the demonstrator 126
9.1.5 Minimum requirements for external V&V 126
9.2 User/HMI Layer 128
9.2.1 Evaluation activities in the analysis phase 129
9.2.2 Evaluation activities in the design phase. 130
9.2.3 Integration evaluation activities 130
9.2.4 Minimum requirements for demonstration 131
9.3 Applications Layer 133
9.3.1 Minimum Requirements to Demonstrator Applications 133
9.3.2 Minimum Requirements to Demonstration Scenarios 133
9.3.3 Requirements to Demonstrator Application Development 133
9.4 Architecture Layer 135
9.4.1 Demonstration Requirements for the Requirements and Specification Phase 135
9.4.2 Demonstration Requirements for the Design and Implementation Phase 135
9.4.3 Demonstration Requirements for the Integration Phase 136
9.4.4 Demonstration Requirements for the System in Operation 137
9.4.5 Minimum demonstration requirements for validation and verification 137

10. WORK REQUIRED PRIOR TO DEMONSTRATION 138

10.1 The Verification & Validation Layer 138

10.1.1 Safety lifecycle and risk assessment 138
10.1.2 Usability trials 138
10.2 User/HMI Layer 140
10.3 Applications Layer 141
10.4 Architecture Layer 142
10.4.1 General Means for Information Transport 142
10.4.2 Implementation of the Application Layer Interface 142
10.4.3 Establishment of the Functional Specification Framework 143
10.4.4 Implementation of Applications 143
10.4.5 Tools to Support the Process of Verification and Validation 144
11. STANDARDS REFERENCES 145

LIST OF ILLUSTRATIONS

Figure 4-1 - The Anticipated Application of the DISC Standard 27
Figure 4-2 - The DISC Information Flow Model 29
Figure 4-3 - The 4-Layer DISC ISC Model 30
Figure 4-4 - Safety Integrity Levels 33
Figure 4-5 - Allowed Data-streams in a 3 SIL Integrated System, Sharing a Database 34
Figure 4-6 - Application in Context 40
Figure 4-7 - Generic Application 42
Figure 4-8 - Abstraction Pyramid of the DISC ISC-System 43
Figure 4-9 - Function Block 45
Figure 4-10 - Structure of Function Blocks, Groups of Function Blocks (also a Function Block) and Physical Nodes. .. 45
Figure 4-11 - Application with Low Level and High Level Interface 46
Figure 4-12 - Example of Function Block Representation of a High Level Application, Damage Control Decision Support component of the Emergency Management application in context 47
Figure 4-13 - Example of Function Block Representation of a Low Level Application, Control of a simple Valve. 48
Figure 4-14 - Three views of a system 50
Figure 4-15 - Implemented on One Network 51
Figure 4-16 - Implementation on one CPU 52
Figure 4-17 - The Application Layer Interface 53
Figure 4-18 - Applications, Function Blocks, Services and Physical Resources 57
Figure 7-1 - Application Specific Future Functions - Information Flow 95
Figure 8-1 - The System Integration Process 119
Figure 9-1 - ISC Development Phases 122
Figure 9-2 - Overall System Lifecycle 123
Figure 9-3 - The Design Cycle Aiming at Optimal Human Involvement in HMI Systems (Neerincx, 1997) 128

1. INTRODUCTION

1.1 About This Document

The current document is the final version of the DISC 3rd Workshop Report, which at the same time constitutes the final report of the DISC Project. Hence, this report is a consolidated version of the results of all the three DISC workshops:

- 1st DISC Workshop, held in Barcelona, Spain, week 49/96
- 2nd DISC Workshop, held in Trondheim, Norway, week 04/97
- 3rd DISC Workshop, held in Bruxelles, Belgium, week 09/97

The contents of this deliverable can be considered as final, however to be applied with some caution. What is described in this document has been approved by all partners in each of the four participating consortia, ATOMOS III, MiTS, COMFORT and MBB II, but the time-wise and financial limitations of the DISC project should be taken properly into account. While this does not affect the overall consensus relating to the project results, these do at the same time represent a framework that is not necessarily complete, and that has not undergone the modifications and adjustments normally associated with the initial application of a novel concept such as DISC.

DISC is indeed a concept, conceived to create the basic foundation for a future, open European Standard for Ship Control. With reference to 4.1, Guidance on the Anticipated Application of the DISC Standard, however, it needs to be noted that DISC is not just an IT-standard, but spans much wider, and is considered to have operational impact in parallel to the obvious impact on the world of manufacturers of marine electronics and automation.

1.2 About DISC and the DISC Consortium

The direct aim of the DISC Consortium is to demonstrate in a tangible way the results of the 4th FP relating to Integrated Ship Control (ISC), but over and beyond that, it is the aim of the Consortium to support the establishment of one European/International Standard in this field.

Ultimately, the objectives of the DISC Consortium supports the Common Transport Policy of the EU, since the establishment of an ISC standard has great potential for a positive impact on the shipping industry as a whole. This will in turn increase market awareness and subsequent use of shipping as a valid alternative to land and air-based transport, easing congestion in the air, on roads and on railways.

Relating to the economy and competitiveness of shipping, ISC is one of the more important contributors, being an enabling technology that allows a reduced manning level on modern vessels, due to increased operational efficiency. By standardizing ISC, the risk of a compromised transport safety is reduced, since the integration of systems will be better defined, easier to accomplish and hence less error prone than is considered to be the case today.

In order to support the goal of a standard, the DISC Consortium has been formed under the coordination of Scandlines (the marketing name for DSB REDERI A/S) as a joint venture between four major contributors to the European ISC research:

- The ATOMOS III Consortium
- The MiTS Consortium
- The COMFORT Consortium
- The MBB II Consortium

These consortia collectively represent:

- The two most important European R&D consortia on Integrated Ship Control (ATOMOS III and MiTS).
- The most important European R&D consortium on VTMIS (COMFORT).
- The only European R&D consortium on maritime black box technology (MBB II).
- Most of the major European manufacturers of shipboard automation and navigation equipment, research institutes and Classification Societies, in combination with market importance and influence sufficient to suggest and implement European Standards in the field.
- 29 European partners from 13 countries in Europe (all EU member states except Austria, Luxembourg and Belgium, including Norway), who are aggressively pursuing the common goals of DISC, are represented in the consortium. Of these, the four Consortium Coordinators (Scandlines (DK), SINTEF (N), ISSUS (D) & DASSAULT ELETRONIQUE (F)) would serve as Contractors, and the rest of the consortia members as Associated Contractors.

The DISC Consortium is the only European Consortium that attempts to integrate all of the above technologies into a seamless, coherent system. Further, to our knowledge, a consortium of such composition and scope is unprecedented in European Waterborne Research and Development.

1.2.1 ATOMOS II/ATOMOS III

ATOMOS III is coordinated by Scandlines (SCL, which is the marketing name for DSB REDERI A/S (DK)), the largest ferry operator world wide, and comprises Lloyd's Register of Shipping (UK), STN Atlas Elektronik GmbH (D), Cetemar S.A (E), Danish Maritime Institute (DK), Lyngsø Marine A/S (DK), D'Appolonia S.p.A (I), Logimatic A/S (DK), Aalborg University Center (DK), the National Technical University of Athens (GR), MAN B&W (D) and TNO (NL).

The ATOMOS II Consortium is currently undertaking 4th FP research in areas 6.3.3/24 & 25, which forms the EU-defined base for task 26. The research carried out by the ATOMOS III Consortium has two objectives, the first being the development of a conceptual standard for the ship control centre working environment, including the corresponding human machine interface, aimed at the enhancement of safety and efficiency through improved operator comfort, workload and awareness, screen presentation and other relevant factors.

The design of the ATOMOS III ship control center is intended to take full account of human capabilities, limitations and performance limits as its constraints. Advanced information processing will be developed to enhance the capabilities of the user to achieve greater safety and efficiency.

The second objective of ATOMOS III R&D is to enhance maritime operational safety and efficiency through an improvement of ship-borne command, control, alarm and information systems as much as practically possible, taking cost-benefit issues into account.

This objective is to be achieved through design, implementation and subsequent validation of a conceptual standard for a safe, efficient and open ISC system which allows cost-effective interoperability and interconnection between system modules from different suppliers. In order to facilitate interoperability ATOMOS II shall define a harmonized user interface (in accordance with the first objective) and provide a standardized process network.

1.2.2 MiTS

The MiTS Consortium consists of equipment manufacturers and research institutions that are users of MiTS (Maritime Information Technology Standard): SINTEF Electronics and Cybernetics, MARINTEK, Autronica A/S, Kongsberg Norcontrol Automation A/S, Scana Moland A/S, Marinor Shipping and Offshore Systems A/S and Det Norske Veritas.

MiTS is a control system integration protocol designed for high level integration of different manufacturers' sub-systems. MiTS Consortium is an off-spring of MiTS Forum, a non profit interest organization for MiTS users.

The companies' interest in MiTS has its origin in a recognized need for standards in integrated ship control (ISC) systems. MiTS Forum and, hence, MiTS consortium has mainly focused on system architectures, communication protocols and standards for Human-Machine Interfaces (HMI).

MiTS Forum and all companies supporting MiTS are committed to and actively work for the establishment of open standards for integrated ship control systems. This includes participation and chairmanship in IEC work groups. MiTS members are also participating in US standardization work through ANSI/ASTM.

1.2.3 COMFORT

COMFORT is a consortium created from the former TAIE and RTIS consortia working on VTS/VTMIS issues under the EURET 1.3 framework. Within the 4th Framework Programme subgroups of the COMFORT consortium are responsible for Task 15 (INCARNATION), a task to produce a traffic image for inland waterway vessels by both, radar and transponder information, for Task 27 (COMFORTABLE), a task to improve VTS/VTMIS by ECDIS and transponder functionality's and Task 46 (MASSTER), a task providing solutions for effective use of maritime simulation. Even if the participants of said projects and of DISC mostly are not identical, close co-operation within the consortium is ensured under the umbrella of COMFORT.

The COMFORT Consortium is coordinated by ISSUS (Institute of Ship operation, Maritime transport and Simulation) and consists of the following 32 equipment manufactures, operator of simulators or research institutes: Alenia (IT), ANAST (BE), AVV (NL), DASA (DE), DMI (DK), ENMB (EE), ERAAM (FR), ERAAM Consultants (FR), HITT (NL), HWSFW (DE), IFN (FR), INGENIA (FR), INRETS (FR), ISSUS (DE), IST (PT), MaRan (NL), MSCN (NL), MSR (NL), Opeform (FR), PTMM (IT), RPA (NL), S.I. (UK), STN ATLAS (DE), TUC (GR), TUD (NL), VTT (FI), WMC (UK), ELP (UK), NEC (IE), TNO-FEL (NL), CETEMAR (ES) and CNR-IAN (IT).

1.2.4 MBB/MBB II

The MBB consortium is currently engaged in the development of a Maritime Black Box system under task 6.3.2/22 of the 4th FP Waterborne Transport.

The aim of this task is to analyze the need for, the application requirements and the benefits of the introduction of a voyage data recorder - the Maritime Black Box - and to provide a demonstrator. This recoverable equipment will help to enhance safety of the maritime traffic by determining and eliminating accident causes and improve awareness. The data recorded by the Maritime Black Box is provided by number of subsystems on a vessel, and shall be consistent and reliable.

The participation of the MBB consortium main partners in the MBB II consortium for the Demonstrator for Integrated Ship Control development will bring into the project the experience acquired during the Maritime Black Box research program.

The MBB II consortium has been created on the bases of the MBB consortium which consists of industrial companies (Dassault Electronique (France), Norcontrol (Norway), Sirehna (France), Kvaerner (Norway).

The objective of the MBB II is to demonstrate the capacity for the MBB system to be integrated in the ISC system and to participate to the development of the conceptual standard, for the ship control centre working environment.

2. RESEARCH OBJECTIVES

Defined from the outset of the DISC joint venture, the Consortium has three common objectives, as follows:

- Long term objective: To strive actively for one, open European/International Standard for ISC, and thereby establish a comparative and competitive edge for the European Maritime Industry on world-wide terms.
- Medium term objective: To demonstrate feasibility of the suggested European/International ISC standard by way of a demonstrator integrating at least relevant results from ATOMOS II, MiTS, COMFORT and MBB.
- Short term: To define the scope of the European/International ISC standard, including possible demonstration means. To define the scope, key steps and possible time-table for the feasibility demonstration, including the physical and the logical layout, the applicable standards, technologies, validation schemes, necessary research to be undertaken etc.

The current project reflects on the Consortium short term objective, and has as such the following specific objectives and scope:

- Establishment of the 'skeleton' for a future European/International ISC Standard, including the identification of the availability of suitable technologies and the identification of future operational-, safety- and efficiency-improving functions to be adopted by the standard. The work described will build on applicable ATOMOS II, MiTS, COMFORT & MBB research and results, but will not be limited to those in terms of width.
- Establishment of the minimum requirements for the feasibility demonstration and validation of the core technologies involved in the suggested standard, and their integration into one coherent system. The demonstrator is foreseen to include representative real-time systems, representative non real-time systems and representative, advanced applications utilizing system data, all operating in a fully integrated manner. The demonstrator will as a minimum comprise the means necessary to fully integrate ATOMOS II, MiTS, COMFORT & MBB, but will not be limited to those in terms of scope.

It is stressed that DISC is created with enhanced effectiveness, competitiveness, usability and safety as the ruling objectives. A user-centred approach has been applied throughout the development.

3. DEFINITIONS

3.1 Abbreviations

ACL	Agent Communication Language
ALI	Application Layer Interface
API	Application Program Interface
ARCS	Admiralty Raster Chart Service (raster type electronic map)
ASN.1	Abstract Syntax Notation No. 1 (ISO 8824)
ATM	(Networking Technology)
CAL	CAN Applications Layer
CAN	Car Area Network
CD	Compact Disc (CD-ROM; Mass Storage for Computer Data)
CEN	The European Standards Organization
CM	Condition Monitoring
CMS	CAN-based Message Specification
COTS	Commercial Off The Shelf (equipment or software)
CPU	Central Processing Unit
CRT	Cathode Ray Tube
DBT	Identifier Distributor
DGPS	Differential GPS (Global Positioning System)
DIN	Deutche Industrie Norm
ECDIS	Electronic Chart Display and Information System (IMO regulated)
ETA	Estimated Time of Arrival
EXPRESS	An information modeling language (ISO 10303-11)
FMECA	Failure Mode and Effect Causal Analysis
FTA	Fault Tree Analysis
GMDSS	Global Maritime Distress and Safety System
GPS	Global Positioning System
GUI	Graphical User Interface
HART	Highway Addressable Remote Transmitters
HAZOP	Hazard and Operability Analysis
HMD	Head Mounted Displays
HMI	Human Machine Interface
HTML	Hyper-Text Mark-up Language
IEC	International Electrotechnical Commission
IETM	Interactive Electronic Technical Manual
IMDG	International Maritime Code for the Carriage of Dangerous Goods
IHO	International Hydrographic Office (UN organization)
IMO	International Maritime Organisation (UN organization)
ISA	Instrument Society of America
ISC	Integrated Ship Control (system)
ISO	International Standards Organization
IT	Information Technology
KIF	Knowledge Interchange Format
KQML	Knowledge Query and Manipulation Language
LAU	Level of Automation

LCD	Liquid Crystal Display
LED	Light Emitting Diode
LIB	Library for user interface components
LMT	Layer Management
MBB	Maritime Black Box
MMS	Manufacturing Messaging Service (ISO 9506)
NATO	North Atlantic Treaty Organization
NEN	Standards Organization
NMT	Network Management
ODBC	Open Data-Base Connectivity (Microsoft)
ODVA	Open DeviceNet Vendors Association
OLE	Object Linking and Embedding
OPC	OLE for Process Control
PADT	Programming And Debugging Tools
PC	Personal Computer
PHA	Preliminary Hazard Analysis
RAM	Random Access Memory
RAS	Requirements Allocation Sheets
RDD	Requirements Driven Development
RSG	NATO Research Group
SAR	Search and Rescue
SIL	Safety Integrity Level
SDS	System Design Specification
SQL	Standard Query Language (for data-bases)
SRS	System Requirement Specification
STEP	Standard for Exchange of product model data (ISO 10303)
PID	Proportional, Integrating and Derivative (regulator)
QBE	Query By Example
UID	User Input Device
UIDT	User Interface Design Tools
UIMS	User Interface Management Systems
URS	User Requirements Specification
UPS	Uninterruptable Power Supply
V&V	Verification and Validation
VDU	Visual Display Unit
VR/VE	Virtual Reality/Virtual Environment
VTMIS	Vessel Traffic and Management Information System
WIMP	Windows, Icons, Mouse and Pull-down/pop-up menus
WWW	World Wide Web

3.2 General Terms

Alarm is for warning of abnormal condition and is a combined visual and audible signal, where the audible part calls the attention of personnel, and the visual part serves to identify the abnormal condition.

An **application** is a subsystem solving or assisting in solving a defined task in the ISC system by:

- Process information
- Exchange of information with other applications
- Controlling/monitoring hardware devices (sensors, actuators, VDUs, etc)

Every unit connected to the system architecture layer is an application

An **application layer interface (ALI)** is a standardized means for:

- Information exchange between applications
- Interoperability and configuration of the applications

through the Architecture layer and communication/supervision with the Architecture layer.

Field instrumentation comprises all instrumentation that forms an integral part of a process segment to maintain a function. The field instrumentation includes:

- Sensors, actuators, local control loops and related local processing as required to maintain local control and monitoring of the process segment.
- User interface for manual operation (when required)

Other equipment items do not, whether they are implemented locally or remotely, belong to the field instrumentation. This applies to data communication and facilities for data acquisition and pre-processing of information utilized by remote systems.

A **function** is a specified purpose to be accomplished (derived from ISO 2382-10). It is normally implemented by software and/or hardware. Functions normally use input information provided by other functions and provides output information to other functions. A function may be divided into a set of (sub)functions.

Notes:

1 The purpose may be accomplished involving user(s) or mechanical equipment or electrical equipment (including computer equipment) or a combination, being fully manual to fully automatic.

2 A function consists of one or more of the following elements:

- indication
- monitoring
- control
- alarm
- safety
- decision support

- reporting

A **function bloc***k* is defined as an abstract entity describing some relation between information inputs, information outputs, function block state and physical effects which are the consequences of changes in function block state.

Indications are the visual presentation of process equipment values or system status to a user.

Interface (ISO 2382-9) is a shared boundary between two functional units, defined by functional characteristics, common physical interconnection characteristics, signal characteristics, or other characteristics, as appropriate. Note - The concept involves the specification of the connection of two devices having different functions.

Ma*x***. unavailable time** is the maximum duration of time the function is allowed to be unavailable, i.e. the maximum permissible time lag involved in restoring lost function upon failure.

A **pre-warning** indicates a process equipment or system state that needs attention.

Processes are components of the environment of technical systems. They are controlled or monitored by the process equipment. A process can be described by a set of state parameters.

Process e*q***uipment** is the equipment (mechanical (machinery, pumps, valves, etc.) or environmental (smoke, fire, waves, etc.)) monitored and controlled by an instrumentation and automation system.

A **process segment** is a collection of mechanical equipment with its related field instrumentation, e.g. a machinery or a piping system. Process segments belonging to essential systems are referred to as essential.

Safety shut-down is a safety action that will be initiated upon failure and is to result in shut-down of the process equipment or part of the process equipment in question.

A **service** is output information provided by an application. Services are provided either on request from an other application (based on input information) or through subscription to information which is event driven (or triggered by rules from other applications.).

A **standards reference model** is the model that describes the interrelationships between Verification and Validation, HMI, Applications and architecture in a system of ISC standards.

A **system** consists of several system components. It has a defined border to the environment outside the system. System components normally interact with each other and with the system enviroment. A component of a system is again a system in itself. The following sub-types of system are defined:

1. A **comple***x* **system** is a system for which all functional and failure response properties for the completed system cannot be tested with reasonable efforts. Units and systems handling application software belonging to several functions, and software that includes simulation, calculation and decision support modules are normally considered as complex.
2. An **essential system** is a system supporting equipment which needs to be in continuous operation, e.g. for maintaining the vessel's propulsion and steering functions
3. An **instrumentation and automation** system includes all components of a technical system necessary for monitoring, control and safety of monitoring of a defined set of processes. This includes all resources required to support one specific function, including:

- The field instrumentation (senors and actuators) of one or more process segments.
- All necessary resources needed to maintain the function including system monitoring and adequate self-check.
- All user interfaces.

4. An **important system** is a system supporting equipment which need not necessarily be in continuous operation, but which is necessary to maintain the vessel's main functions related to safety (Power generation, Propulsion, Steering, Fire protection, detection and extinction, Drainage and Bilge pumping, Ballasting, Cargo handling and Anchoring and Mooring).
5. An **integrated system** is a combination of computer based systems which are interconnected in order to allow common access to sensor information and/or command/control.
6. An **non-important system** is a system supporting functions not covered by the other two (essential and important) categories.
7. A **technical system** is a made system composed of components which are made to control and/or monitor other components. Components controlled or monitored by other components are called **processes**. Components controlling and/or monitoring processes are called **applications**

A **tag** is defined as a named information entity.

User is any human being that will use a system or device, e.g. captain, navigator, engineer, radio operator, stock-keeper, etc. The user may be onboard or ashore (company staff ashore).

Uninterruptable Power Supply (UPS) is a device supplying output power in some limited time period after loss of input power with no interruption of the output power.

Wor*k*station is a position at which one or several functions constituting a particular activity are carried out.

3.3 Terms Related to Computer Based Systems.

An **application** is computer software performing defined tasks in an ISC system. An application provides services to other applications (e.g. task solving, communication, database, VDU applications etc.). The purpose of an application is called **functionality**

Basic software is the software necessary for the hardware to support the application software. Basic software normally includes the operating system and additional general software necessary to support the general application software and project application software.

Computer includes any programmable electronic system, including main-frame, mini-computer or micro-computer.

A **computer tas*k*** is, in a multiprocessing environment, one or more sequences of instructions treated by a control program as an element of work to be accomplished by a computer.

Data are defined as a bit-pattern out of context.

A **data object** is considered to be one of the information elements associated with a function block.

Data communication lin*k*s includes point to point links, instrument net and local area networks, normally used for inter computer communication on board vessels. A data communication link includes all software and hardware necessary to support the data communication. For local area networks, this includes network controllers, network transducers, the cables and the network software on all nodes.

General application software is computer software performing general tasks related to a process equipment being controlled or monitored, rather than to the functioning of the computer itself.

Information is defined to be data in context.

An **instrument net** is used for data communication within the field instrumentation connecting instruments in a network.

A **local area networ*k*** is used for data communication between the field instrumentation and the other parts of a system, and between different systems.

A **node** is as process segment or a part of the system connected as part of the data communication link.

A **point to point** link is used for data communication between two dedicated nodes.

Project application software is computer software performing tasks related to the actual process equipment for a specific project.

A **software module** is an assembly of code and data with a defined set of input and output, intended to accomplish a function and where verification of intended operation is possible through documentation and tests.

A **unit** is an entity of hardware, software, or both.

3.4 Terms Related To Verification and Validation

Availability refers to the ratio of actual service time to expected service time at sea.

Maintainability (ISO 2382-14) refers to the ease with which maintenance of a functional unit can be performed in accordance with prescribed requirements.

Mean Rate Accuracy (ISO 2382-21) refers to the error margin excluding errors caused by noise as input, which should not be exceeded when a device is used under normal operating conditions. Note: ISO 2382-21 does not define accuracy alone.

SIL - Safety Integrity Level (ISO 1508). One of 4 possible discrete levels for specifying the **safety integrity** requirements of the safety functions to be allocated to the safety related systems. Safety Integrity level 4 has the highest level of safety integrity, Safety Integrity level 1 has the lowest.

Validation means confirmation by examination and provision of objective evidence that the particular requirements for a specific intended use are fulfilled.

Verification means confirmation by examination and provision of objective evidence that the specified requirements have been fulfilled.

3.5 Terms Related to Human Machine Interface

HMI-Application interface is the shared boundary between the information presentation and user input, and the application layer being the service provider.

Human centred design is, when designing a user interface, the process of assessing the tasks to be performed, taking into account the human capability, limitations, etc., and also taking into account the different operational modes (normal (stable), transient (e.g. start/sop), degraded operation (failure of equipment outside or inside the computer based system) and emergency operation). A major attribute to the human centred design is to bring the end users into the design loop.

Human Machine Interface (HMI) is the shared boundary between the user and the display unit(s) and user input device(s) of the computer based systems.

Technology centred design is, in design of a user interface, the process of assessing the tasks to be performed, taking into account the technical capability, limitations, etc.

User interface is the shared boundary between the user and environment including the computer based systems.

Visual Display Unit (VDU) is any area where information is displayed including indicator lamps or panels, instruments, mimic diagrams, Light Emitting Diode (LED) display, Cathode Ray Tube (CRT), and Liquid Crystal Display (LCD).

User Input Device (UID) is any device from which a user may issue an input including handles, buttons, switches, keyboard, joystick, pointing device, voice sensor and other control actuators.

4. THE DISC 4-LAYER STANDARD REFERENCE MODEL

4.1 Guidance on the Anticipated Application of the DISC Standard

The main purpose of the DISC standard is to ensure full integration of shipboard systems in a standardized fashion, with the aim of enhanced maritime efficiency, competitiveness and safety.

The means to obtain the desired enhancements are laid out in the DISC standard, but for clarity and ease of comprehension, the following section outlines the scope and time-wise duration of applying the standard.

It is of major importance that everybody relating to the use and implementation of a DISC-compliant system realizes that DISC is more than an IT standard: DISC is also a framework for the application of the User-Centered Design Approach to command and control of ships, and includes comprehensive requirements to Verification and Validation of the Ship Control Center layout and functionality's, that forms an integral part of the standard.

The main implication of this is that the application of DISC cannot be removed from the specification, design, construction and operation of the ship itself; DISC needs to be interactively customized for each specific type and size of vessel, and it's owner & operator.

4.1.1 Initial Steps

Bearing the above in mind, the application of DISC starts very early in the design process for any new ship, and should as the initial step consider the information flow to and from the vessel in question. Section 4.2 contains a rough example of such deliberations, developed in the beginning of the DISC project as a development tool, but much more in-depth studies of the information flows to, from and around the anticipated ship should be carried out, in order to ensure that the overall system requirements to the ISC system of this ship will meet the needs of the intended operation in technical terms.

On a parallel basis, assessment of typical user tasks to be carried out on board needs to be addressed, in way of a function and task analysis or similar, to be used in the distribution of functions across the ISC system and the users of the ISC system. Part of this analysis should be dedicated to objective analysis of the average capabilities of the users, and their particular needs, in order to derive the desired ISC layout and subsequently to prepare first the User Requirements Specification (URS), and then the System Requirements Specification (SRS) for the ISC system, in terms of ergonomics and usability, and derived technical issues, if any.

Exchange of high-level information between the ship Owner and the Verification & Validation Institute.

The importance of these initial steps cannot be underestimated, and should be subject to extensive validation throughout the specification process. Section 8 contains description of generic tools in support of this process, and relates to the issues of risk assessment, safety assessment, assessment of correct ergonomics etc.

4.1.2 Implementation

Given a well prepared System Requirements Specification (SRS), the process continues with the preparation of the Systems Design Specification (SDS), based on the SRS, and involving continuous dialogue with the ship Owner and the Verification and Validation Institute(s) involved in the actual instance.

Following the SDS, the key functions in the continuation of the development process of the DISC ISC system resides with the Systems Integrator, who is responsible for the technical integration of the combined system, and hence for the communication with external sources as per the figure below (Figure 4-1).

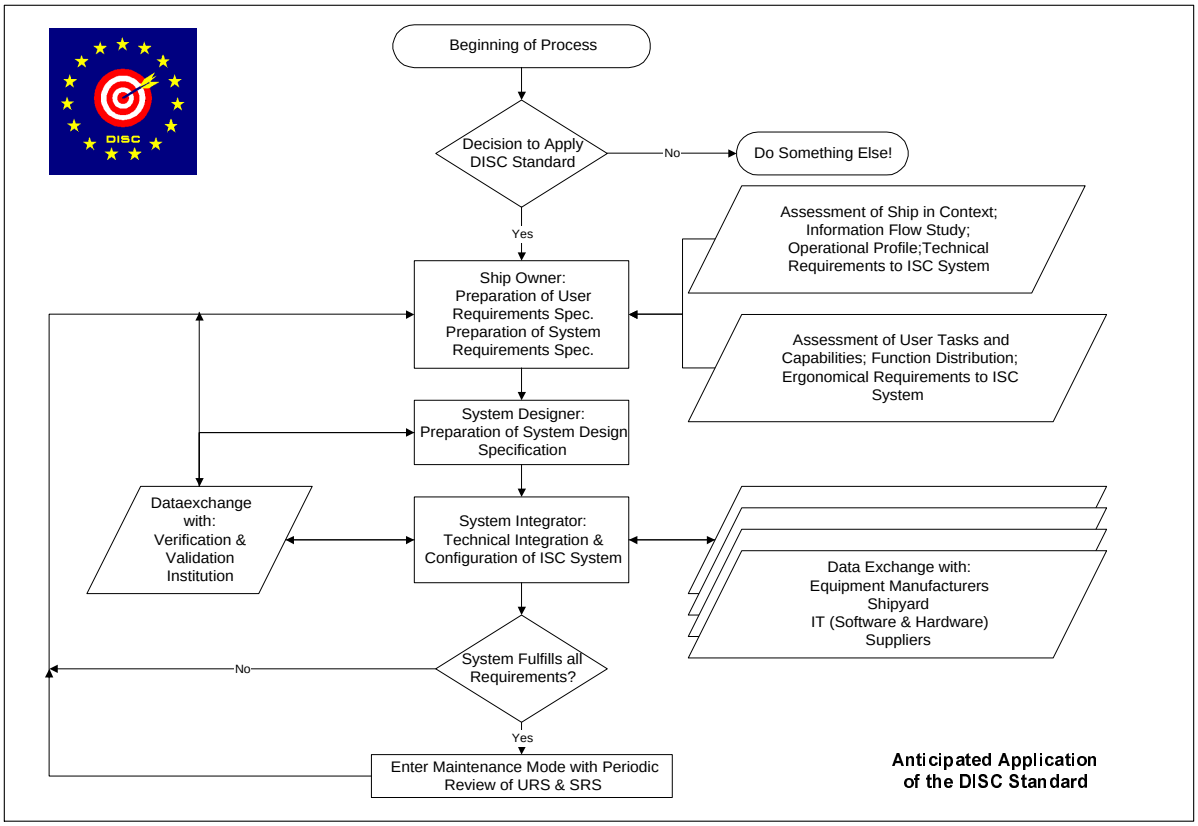

Figure 4-1 - The Anticipated Application of the DISC Standard

The actual considerations in terms of the DISC standard and the supporting, international standards are described in depth throughout this report, with section 4 dedicated to the definition of, and requirements to, the 4 different layers of the DISC standard:

- Section 4.4 relates to the Definition of the Validation & Verification (V&V) Layer of the DISC Standard
- Section 4.5 relates to the Requirements to the Verification & Validation Layer of the DISC Standard
- Section 4.6 relates to the Definition of the User/HMI Layer of the DISC Standard
- Section 4.7 relates to the Requirements to the User/HMI Layer of the DISC Standard

- Section 4.8 relates to the Definition of the Application Layer of the DISC Standard
- Section 4.9 relates to the Requirements to Application Layer of the DISC standard.
- Section 4.10 relates to the Definition of the Architecture Layer of the DISC Standard
- Section 4.11 relates to the Requirements to the System Architecture.

Additionally, section 5, titled 'Development of Skeleton Standard', puts DISC into the context of international standards, draft standards and de-facto standards.

Referring to the actual process of creating an instance of DISC, intensive liaison between the Systems Integrator and the ship Owner, the shipyard and the various equipment suppliers (software and hardware) involved in the actual instance of ISC system shall be foreseen, and needs formalized support in terms of tools and technologies for the interchange of information. Likewise, very close cooperation and interchange of Verification and Validation information between the Systems Integrator and the body undertaking the approval of the system shall be foreseen and needs support.

Detailed references to this kind of tools and the technologies they support can be found in sections 6 and 8, respectively. The same main chapters contains information and guidance on creating instances of DISC in terms of application development and design of the HMI. Finally, these two main sections contains requirements and guidance on the Verification and Validation procedures that relates to the creation of a DISC instance.

4.1.3 Maintenance

A major benefit of DISC resides in the flexibility of the standard, and hence in the systems that complies to the standard. It the ever changing world of shipping it must be foreseen that the initial constraints and requirements that originally were the foundation for the User Requirements Specification and hence the System Requirements Specification will change over time, as will the capabilities of the crew of vessel, and very likely, the operational profile of the vessel as well.

It is therefore considered desirable to take the original specifications up for re-assessment at regular periodic intervals, or in cases where major re-fit or conversion of the vessel takes place. Using the same tools and technologies as originally applied, it is thus possible to specify the necessary changes to the actual instance of a DISC-compliant ISC system, which then in turn can be implemented with relatively ease.

4.2 The DISC Information Flow Model

As the initial activity of the project, a context model of the ship within the industrial/transport environment was developed (see Figure 4-2), to serve as a tool for the determination of requirements to the ship, and hence the Integrated Ship Control System.

It should be pointed out that Figure 4-2 **does not** try to reflect the total information flow in the shipping industry, but only those that directly seems to influence the ISC system. Hence, a lot of relations are ignored, among these for instance the interaction of VTMIS with a lot of other shore-based services. In the current context, all this information is assumed represented by the arrows indicated.

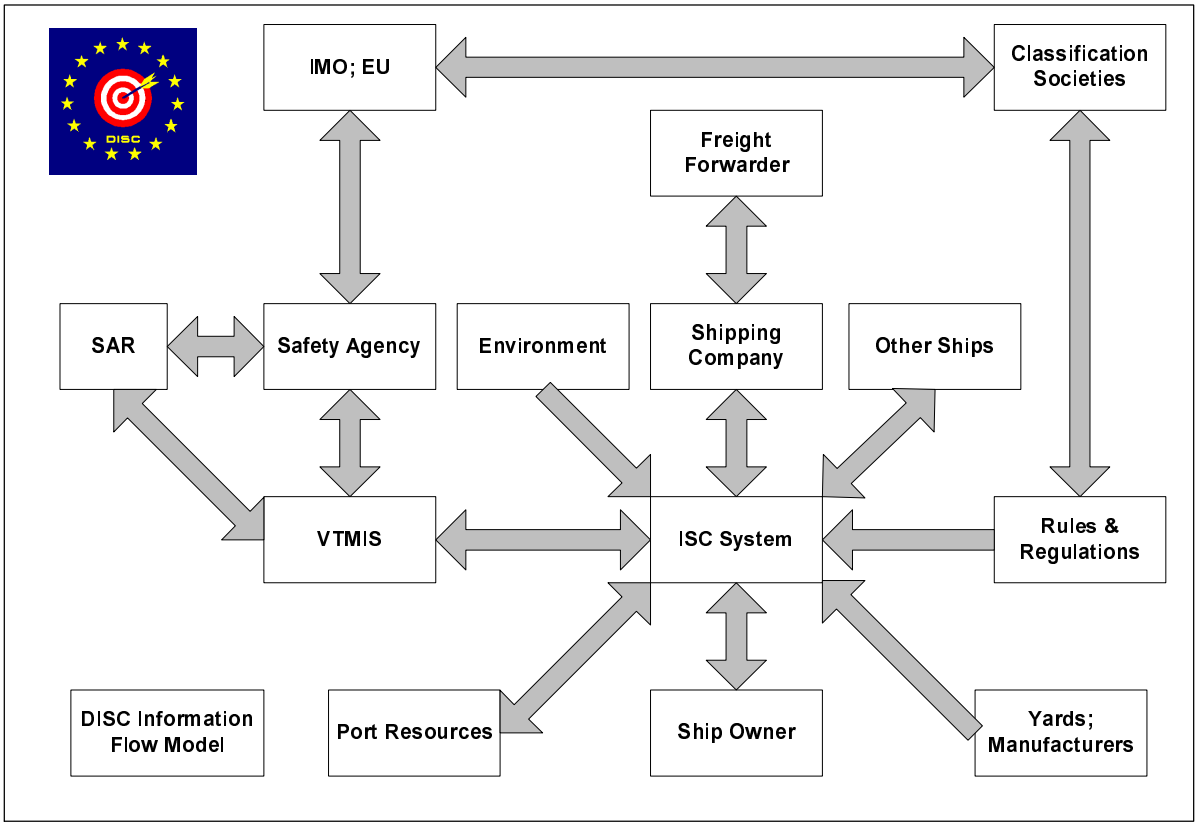

Figure 4-2 - The DISC Information Flow Model

Consortium-wide development and parallel discussion of the 'Information Flow Model' resulted in a number of general requirements:

- System should exhibit maximum flexibility and modularity to support unknown future functions
- System should support technologies needed to enhance the integration of ships into the transport chain
- System should support a vast increase in communications needs as compared to present-day requirements

4.3 The DISC 4-Layer ISC Model

In turn, the initial discussions on the DISC Information Flow model led to the confirmation of the 4-layered DISC ISC model, depicted in the following (see Figure 4-3).

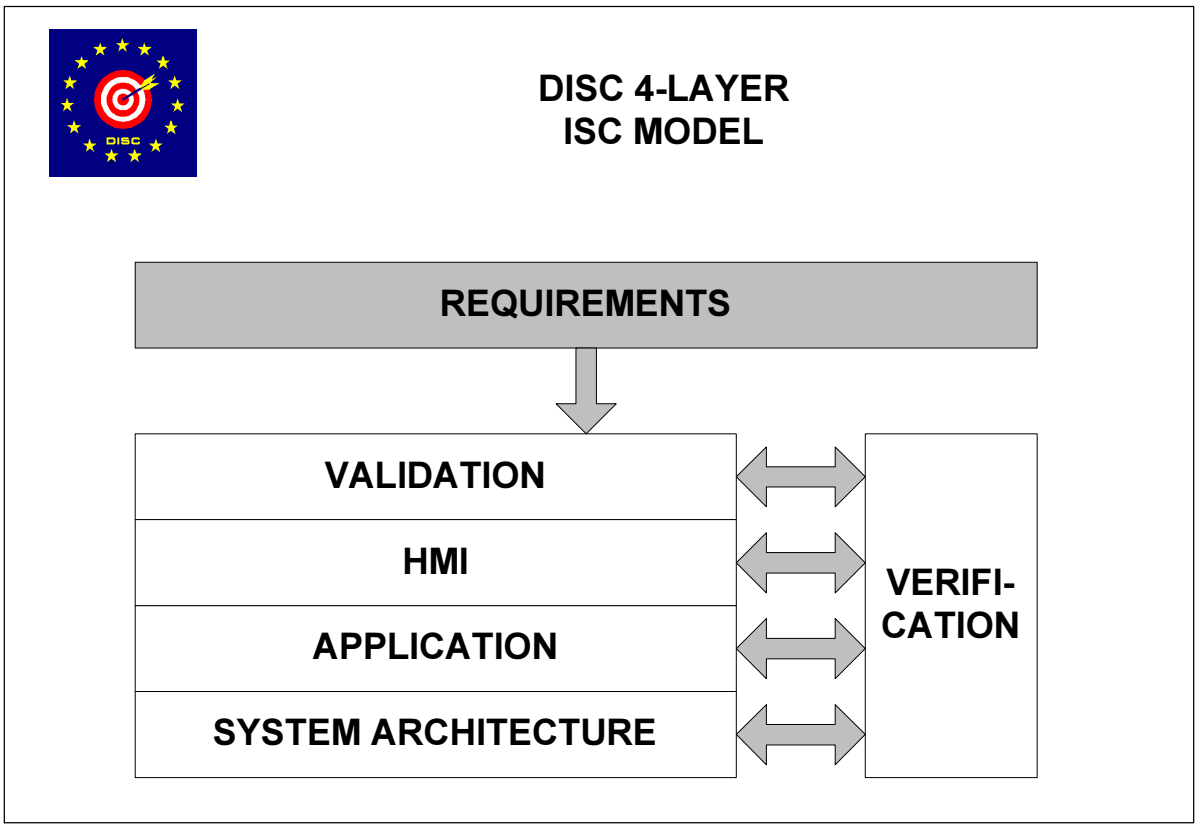

Figure 4-3 - The 4-Layer DISC ISC Model

The above model can be taken as a sectional enlargement of the ISC system contained in DISC Information Flow model; where in the above 'Requirements' encompasses all the various requirements derived from the greater model.

These **Re*q*uirements** are passed through each of the 4 layers of the ISC system, and are added to and deducted from on the way towards the System Architecture Layer, upon which the requirements are 'bounced' back, now in shape of **Limitations** and **Possibilities**, thus illustrating the couplings between the layers of the ISC system.

All through the process, the actual state is measured against the verification requirements.

4.4 Definition of the Validation & Verification (V&V) Layer of the DISC Standard

4.4.1 Introduction

The widespread use of standardized integrated ship control will only be realized if all concerned parties (including system suppliers, owners, operators, IMO, flag states, classification societies and the general public) have confidence in the new technology. Furthermore, the use of ISC can potentially increase shipping safety by enabling new applications, and by encouraging the transfer of new technology from high value ships to lower value ships. In detail the application of the validation and verification guidelines in this document is intended to have the following advantages:

- Enable cost effective development of a dependable integrated system
- Assist demonstration that ISC satisfies IMO, flag state and classification requirements.
- Support modularity.
- Using verified components as far as possible.
- Enable a consistent approach to be applied at all system layers (e.g. HMI, Application and System Architectures layers)
- Encourage the adoption of new technology by demonstrating that potential hazards associated with new applications will not adversely affect safety, and that claimed safety and operability improvements are indeed realized.
- Ease maintainability of the ISC

Several standards has been validated in order to find definitions of validation and verification which is suitable for this purpose. Among them, the definitions found in IEC1508: Functional safety[1]: safety related systems (draft) are found as the best basis:

Validation : Confirmation by examination and provision of objective evidence that the particular requirements for a specific intended use are fulfilled

Verification: Confirmation by examination and provision of objective evidence that the specified requirements have been fulfilled

According to these definitions,

Validation is concerned with **getting the right system** and includes such activities as:

- Customer review of a supplier requirements document
- Experimental use of a prototype
- User trials of a completed system
- Demonstrating that the system is suitable for a particular use

Verification is concerned with **getting the system right** and includes such activities as:

- Checking anything that is produced during the design against the requirements for that phase
- Testing of the end result according to specifications
- Unit-testing of software components against its design specification
- Structured walk-through to verify that the design implements the stated requirements
- Verifying that the coding standard has been complied with

4.4.2 Safety Integrity Levels

Present classification of systems is based on functionality, for instance as described in IEC 1508 Ch.3.1 "Essential (Instrumentation and Automation) System". This is unlikely to remain adequate in the future for the following reasons:

1. Increasing integration may create new hazards or mitigate existing hazards
2. Completely new functionality's may not fit into a predetermined scheme.

A classification of the system should be based on a method that:

1. Defines the safety targets for the total system
2. Identifies and analyzing potential hazards
3. Analyzes the risk associated with the hazards
4. Determines how to reduce the risks to an acceptable level by

    - Avoiding the hazard (for example by re-positioning machinery or equipment)
    - Defining suitable operating procedures
    - Ensuring that equipment that supports safety functions is sufficiently dependable

The end result of these four activities will be a measure of the reliability required for the safety related system. The concept of safety integrity level, as introduced by the IEC 1508, provides the reliability measure classification needed. SILs are defined in IEC 1508 on a scale running from 1 to 4 representing a notional reliability with which a safety function is performed. This section outlines the general concept of SIL. Detailed definitions of SIL levels may be found in IEC 1508 Pt.1 Ch.7.5.

It is recognized that a great deal of accepted wisdom is encoded in existing standards, rules and regulations, and that it is unlikely that the overall hazards associated with each main ship system will change. It is also recognized that a full safety analysis for each ship, and for each ship system, is unlikely to be required for each new building because of the commonality between ships of a given type. However, if existing practices or results are substituted for a full safety analysis this should be documented and justified.

The DISC approach to verification and validation is not expected to present significant difficulties for suppliers using current good engineering practice. However, it does represent a change of direction which is intended to shape the practical implementation of verification and validation for the truly integrated ship control systems of the future.

Ship IT-systems incorporate systems which may not be considered to have any relation to the safety of the vessel. However, when such functions are integrated into an ISC, the potential hazard of such an integration should be analyzed. In those cases where the effect of integration is such that the SIL of this function can be

accepted to be lower than SIL1, the application of the methods and principles recommended for V&V by this standard can be left out.

Safety targets may be determined in several ways. It is envisaged that this will ultimately be a matter for legislative bodies. However, for the purposes of DISC the following principle is to be adopted:

The use of an application will create a ris*k* to the safe operation of ships that is no greater than the ris*k*s currently presented by e*x*isting technology and practices (as determined, for e*x*ample, from incident databases).

The output from the hazard identification will include the SIL for each ISC application and component, including components from each DISC layer. The safety verification for that application will be a structured demonstration that the SIL has been achieved. This will be facilitated by the following "composition principle" for SILs.

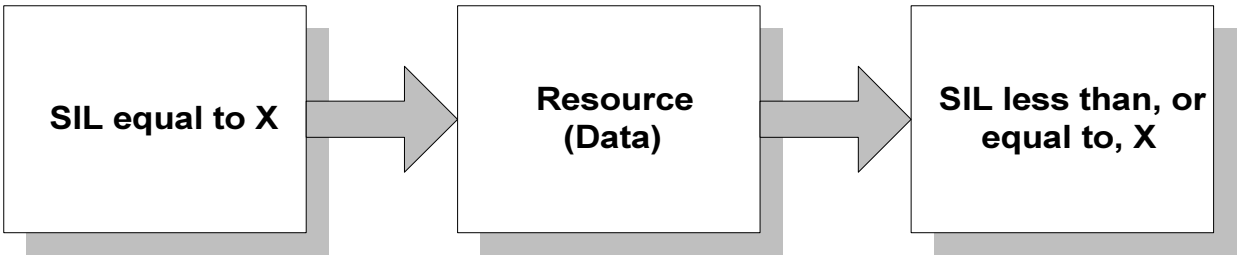

Figure 4-4 - Safety Integrity Levels

The arrows in Figure 4-4 represent "flow of possible failures". For example if a SIL 2 application generates data, then any application relying on that data can be **at most** SIL 2. The principle is further elaborated below. In this figure datastreams toward a higher SIL is not allowed.

The same principle applies to all shared resources. For example:

- Processors
- Physical memory
- Networks

If the developer wishes to claim that the use of a shared resource by a lower SIL application or system entity does not reduce the SIL of other applications, suitable isolation of the lower SIL entity must be demonstrated, ie. that failures of the lower SIL entity cannot cause failures in the higher SIL entity.

It is also possible to increase SIL level by combining lower SIL applications or system entities in a suitable way. For example, three diverse SIL1 data sources could be combined by a SIL2 voter to produce a SIL2 process value for use by SIL2 or lower applications.

On the other hand, it must be noted that the arrangement of several same SIL entities do not necessarily produce a same SIL application. The way they are combined has to be proved as relevant with regard to the SIL of primary components.

A key advantage of the SIL concept is that applications can be built from components (such as the ISC architecture or other applications) using components certified to a given SIL level. This will give a

disciplined approach to safety verification whilst encouraging reuse and modularity, stream-lining the approvals process, and minimizing re-assessment of existing components.

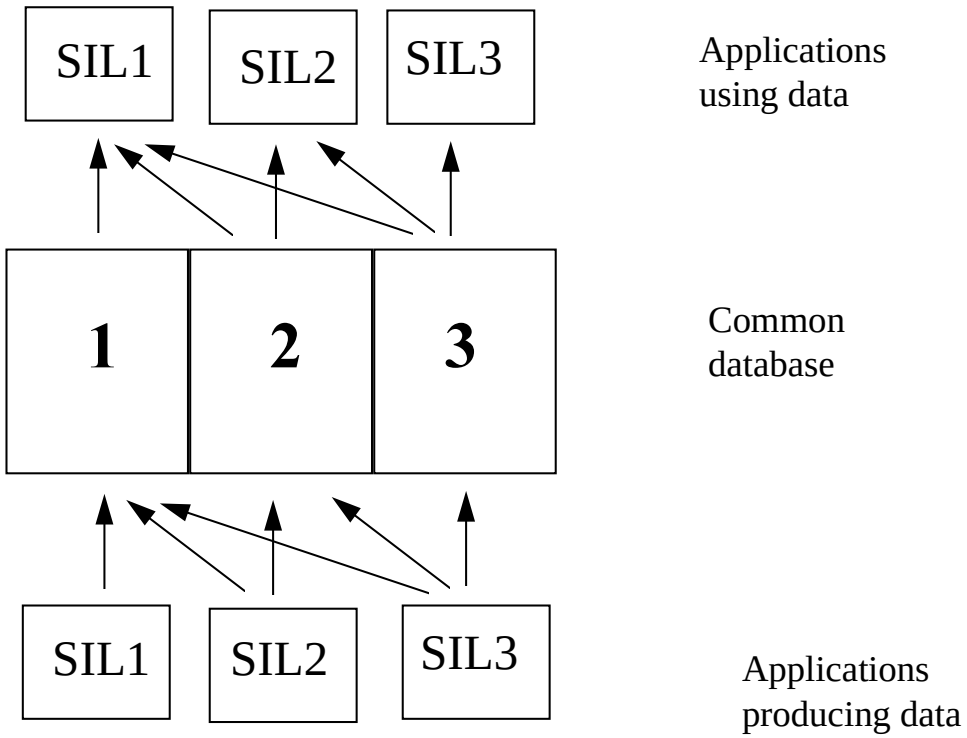

Figure 4-5 - Allowed Data-streams in a 3 SIL Integrated System, Sharing a Database

It is envisaged that the same general concept can be applied to the verification of other dependability attributes, such as functionality (Functionality Integrity Level - FIL).

As can be derived from this description , V&V places no specific technical requirements on the system e.g.:

- no requirement on physical redundancy of networks
- no requirement on physical isolation of processes on different processors
- no requirements on the synchronisation e.g. latency of timing

The technical re*q*uirements are defined in the specifications and in the functionality descriptions of the components and the whole system. **The V&V** requires the justification that this specification is always satisfied to a given reliability. For instance if latency is required to SIL 2 no message must fail to reach its destination more than once pr. 100 years of continuous operation.

IEC1508 make distinction between system with continuous mode of operation and demand mode of operation. It does define that SIL X means probability of failure to perform designed function either in demand or continuous mode of operation as follows:

- **Demand mode of operation**: between 10^{-x} and $10^{-(x+1)}$ failures per demand
- **Continuous mode of operation**: between 10^{-x} and $10^{-(x+1)}$ dangerous failures per year

Example: SIL 2 is component or system which has between 10^{-2} and 10^{-3} failures per demand or between 10^{-2} and 10^{-3} dangerous failures per year.

Whichever mode can be used as a basis on definition as far as its use is justified so that the overall safety target is achieved. A component is in a continuous mode of operation, if it functions continuously even if it is not demanded e.g. a network should be considered a continuous mode system. Emergency shutdown system is a demand mode system, but it normally has a continuous mode subsystem for monitoring the overall status of the process.

4.5 Re*q*uirements to the Verification & Validation Layer of the DISC Standard

The following general requirements are to be set to the V&V layer :

- To define the acceptance- and or certification procedures for each identified entity of each layer
- To define corresponding guidance to developers
- Build on successful projects of the same nature from other relevant EU-projects in the transport sector.
- To be built on existing international standards as far as applicable to the maritime field.
- Ensure that requirements from international bodies are fulfilled.
- Ensure that entities with different safety integrity levels can be accommodated in the ISC architecture without compromising the safety
- Integrity levels are used mainly concerning safety, but can also be used in the context of other dependability attributes.
- Encompass or specify or refer to procedures to establish integrity levels for entities.

As integrity levels helps thinking about safety and other dependability's, e.g. backup/redundancy systems from the point of view of operability levels, prioritizing of applications and messages between them. SILs are used in this standard to achieve expected future demands for ship and ship operation safety.

4.6 Definition of the User/HMI Layer of the DISC Standard

4.6.1 Two Levels of the HMI

With an increased level of automation the operator's task will switch from manual control of a single function to the supervision of several functions. The general task of an operator in such task environments is to make a comparison between observations and his or her expectations and prior knowledge, in order to minimize deviations between the actual state of the system and a goal or reference state. In general, the quality of information transfer through the HMI can be considered from the perspective of two levels of an adaptive information processing system.

At the first level, the HMI presents the relevant state information in relation to reference values. Besides actual state information relevant information may consist of related information like signal trends and histories. At the control side the operator is provided with the means to adjust set-points of the lower-level automatic feedback loops. Thus, at this level there is a direct relation between input signal and response.

At the second level of information transfer, higher order objectives, determined by the operational goals and criteria for safety and efficiency, are translated into pre-programmed rules for the first level. Of vital importance to the quality of the decision-making process is the ability to adjust this process on the basis of actual developments in and around the ship. This modeling process is of a more long-term nature than the direct compensation at the feedback level, and provides the HMI system with the ability to anticipate future developments, eventually resulting in an overall increase of response speed. At this level the HMI is defined in terms of user guidance in the selection and integration of lower-level information for efficient monitoring on the basis of prior information and knowledge and in determining the necessary interventions in the ISC for a timely correction of disturbances that cannot be compensated by the automatic controllers. According to the definitions above the requirements at this level refer to the HMI-Application interface.

4.6.2 Standards and Guidelines

With regard to the first level of the HMI, much research has been conducted providing standards for the way in which information should be presented. At the second level, however, there are only few guidelines that can assist in designing the HMI. The main reason for this lack of standardized knowledge is that support tools are highly dependent upon the interaction between failure probability and uncertainty in the information from the applications layer on the one hand, and knowledge and information task goals of the operator on the other. For the specification of the information requirements and support tools a user-centred design should therefore be applied. According to such a design process, human-centred issues are considered early and continuously in the product life-cycle.

The principle activities are:

- Understand and specify task goals and context of use;
- Generate the user and organizational requirements;
- Design potential solutions and produce prototypes;
- Evaluate designs against user criteria and task performance

4.7 Re*q*uirements to the User/HMI Layer of the DISC Standard

4.7.1 General Re*q*uirements to User Interface

Compatibility:	Minimize the amount of information re-coding that will be necessary
Consistency:	Minimize the difference in dialogue both within and across various user interfaces
Memory:	Minimize the amount of information that the user must maintain in short term memory
Structure:	Assist the user in developing a conceptual representation of the structure of the system so that they can navigate through the interface
Feedback:	Provide the user with feedback and error-correction capabilities
Workload:	Keep user mental workload within acceptable limits
Individualization:	Accommodate individual differences among users through automatic adaptation or user tailoring of the interface

These general requirements relate to user interface for any application and include:

- Workspace design and arrangement
- User input device and display unit design
- Requirements related to screen based systems
- Design of workplace
- Work Environment

4.7.2 Re*q*uirements to Applications

A set of requirements will also be given to applications, when found necessary. These requirements will only have to be complied with when applications covered by these requirements are being installed on board a ship. There is no requirement that the applications should be installed on a particular vessel.

4.7.3 Re*q*uirements to the Integration of Applications

The integration of the separate applications will be an important part of the standard. These requirements should focus on the interface between the different applications and the task of structuring the information/control functionality for the combined process. The modules integrating separate applications will also be regarded as applications.

General design objectives for such an Integration of Applications are to:

- Reduce mental workload of the operator,
- Assist the operator in maintaining an accurate internal representation of the process,
- Involve the operator in an active way in the problem-solving process,
- Adapt the available information to the cognitive processes of the operator,
- Foster operator acceptance,

- Reduce cognitive tunnel vision, that is, the tendency of humans to focus on one single part of the process, ignoring the rest (including alarms).

In achieving these general objectives, the human-machine interface should have an appropriate degree of user friendliness and transparency.

4.8 Definition of the Application Layer of the DISC Standard

4.8.1 Application in context

The figure below defines applications and application layer interface in relation to the architecture layer, c.f. the figure below (Figure 4-6 - Application in Context). Every application is linked to the system architecture via an «Application Layer Interface» (ALI). The basic services for e. g. resource management, information exchange or configuration monitoring are provided via the ALI. Because all these services can also be seen as applications, the system architecture itself is a «virtual entity» which serves as a means to connect all applications.

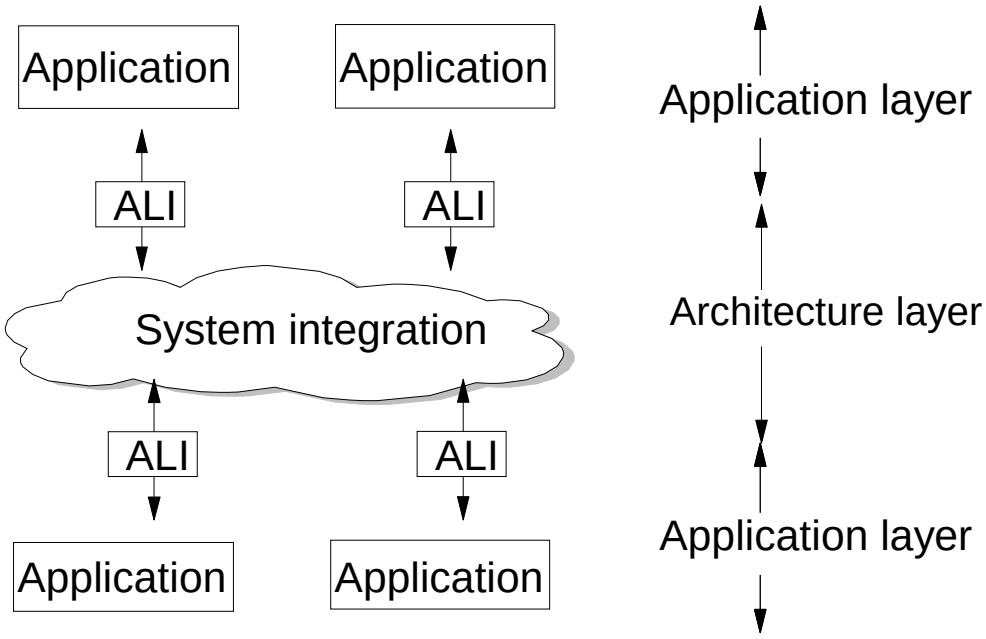

Figure 4-6 - Application in Context

All applications are service providers and services range from providing a single figure to responding to queries of high abstraction level like e.g. “Provide fuel consumption figures from previous west bound voyages going from A to B during winter months”. Services may also comprise physical effects.

Applications may be shipboard or shore-based (connected through the ISC Network).

An application may either subscribe to a service (event driven) or request a service.

A register will be maintained (at system level) listing:

- Applications present in the ISC system including context based relative priority (reflecting the safety criticality of the application)
- All services presently available in the ISC system
- Safety criticality of application (safety-critical, safety-related or non-safety-related)

Redundancy of safety critical applications is ensured by dynamic distribution of applications on available hardware and displays.

4.8.2 Types of Applications

A number of applications have been identified in order to illustrate the potential scope of applications which may belong to the ISC system. Below are given examples of applications:

Applications Solving User-related Tas*k*s

- Navigation
- Cargo management and loading
- Emergency Management
- Propulsion and steering
- Anchoring and mooring
- Communication / information gathering
- Hull integrity
- Maintenance and diagnosis
- Black Box
- Administrative functions

System-oriented Applications

- System Manager
- Common database
- HMI manager

The functionality's of the applications are described in section 7.3. A few comments are given to the second group below. All three types of system oriented applications have to be present in the ISC system.

System Manager

This application solves conflicts in the ISC system in case of resource allocation problems (manpower and/or hardware/equipment resources).

System Manager tasks include:

- prioritising tasks requiring user interaction (based on the context based priorities)
- filtering of information to be presented to the user in case of emergencies
- reallocation of VDU´s if one VDU is down.

Each ISC application must register with the System Manager to signal its presence and readiness

Common Database

The common database application subscribes to data from applications, store them, and provide them (on request or on subscription basis) to other applications. (The use of a common database does not exclude local databases holding application specific data).

HMI manager

The HMI manager controls the use of the HMI devices. This application subscribes/request data from the applications solving user related functions and exchanges information with the attached HMI devices accordingly.

4.8.3 Generic Application

A generic application has been constructed comprising the following components:

1. Planning / Decision Support
2. Monitoring
3. Control
4. Recording/ Replaying / Documentation
5. Communication
6. Help / Training / Simulation

Applications are represented by function blocks and may be built from one or more function blocks, function blocks are reused to the widest possible extent throughout the ISC system. The function block concept is described in the sections below.

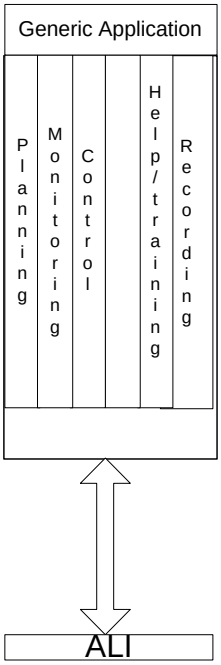

Figure 4-7 - Generic Application

An application can be considered as a function block consisting of a number components each represented by a function block which in turn comprises an arbitrary number of function blocks.

4.8.4 Specification mechanisms

One goal with the next generation ISC standards must be that the specification mechanisms used on the application layer (user requirements specification to functional specification) is useful also in the architecture layer when a system is committed to hardware and software. This specification mechanism shall be fully object oriented. One can say that the meeting point between the application and the architecture layer is this functional specification format.

Scope of specification mechanisms:

- Specification of information models
- Specification of data flows
- Specification of state changes
- Specification of dynamics

The objective within the R&D project DISC is, to provide a framework for the integration of the physical components of the system which guarantees maximum flexibility and modularity. To do this it needs to consider (at least) three different description forms:

1. A framework for the organization of functional relationships (modules) in the ISC.
2. A framework for the organization of information elements in the ISC.
3. Bridge the gap between one information or function specification format and several possible hardware and software realizations. This is covered in the architecture sections.

4.8.5 Abstraction Pyramid of the DISC ISC-system

Information elements and functional relationships in an ISC can be organised in an abstraction pyramid as illustrated in Figure 4-8.

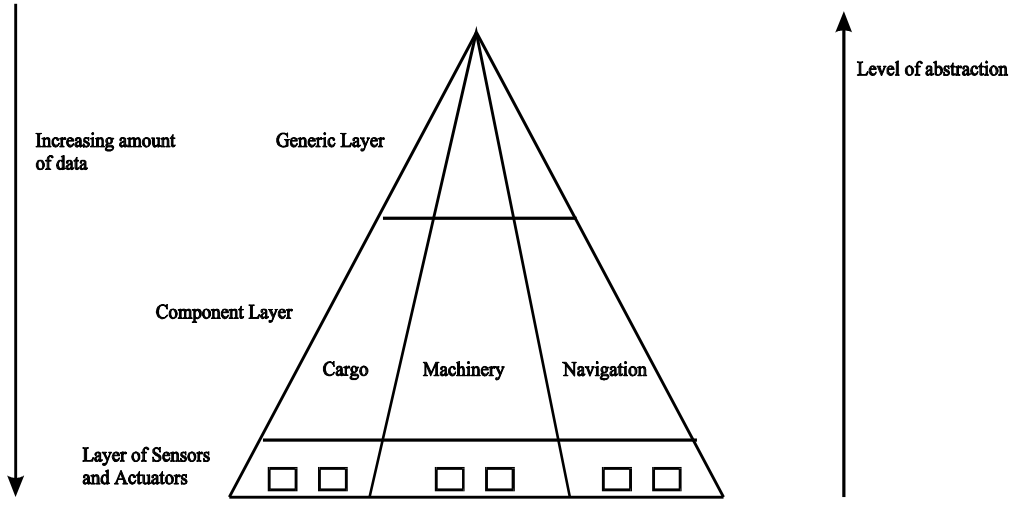

Figure 4-8 - Abstraction Pyramid of the DISC ISC-System

The basic idea is that low level information and functions, represented by sensors and actuators are gradually being refined into fewer and more qualified information elements. On the top of the pyramid one can envisage the few information elements that are of interest to the ultimate user of the ship, the owner of the cargo. These can be, e.g., estimated time of arrival (ETA) and the port one leaves (A) and the one, one expects to arrive in (B).

The pyramid can be characterised by the following properties:

- The number of information elements and function blocks increases as one goes down. On the bottom level one can have tens of thousands of individual entities.
- The information contents and abstraction level increases as one goes upwards. Several position sensors on low levels (GPS, DGPS, Loran-C, dead reckoning, motion reference units) will be integrated into one reference position for the ship. Similarly, a specialised machinery control system will be generalised to one generic main propulsion function block.
- At the top of the pyramid there is the so-called "Generic Level". This level contains generic ship information, e.g., position, speed, heading, destination, name of ship, general machinery condition etc. and also generic functions as navigation, propulsion control, cargo control etc. These entities can be made available in the same manner for any type of ship. Therefore, this level can be modelled in a standardised way which is valid for every ship.
- The "Component Level" contains ship specific details that are not possible to generalise. One must resort to, e.g., general entity-relationship models to describe the relationship between the various information elements. The level contains the specific configuration and implementation of the ship concerned.
- At the lowest level, the "Level of Sensors and Actuators", one will see a tendency to the decrease in the number of meta-models, i.e., sensors and actuators can probably be described in a common framework with relatively few variants, e.g., valves can be generalised into 5 to 10 groups. Therefore, similarly to what was the case on the generic level, it is possible to model this level in a standardised form.

It must be a goal to standardise a set of entities belonging to the upper part of the pyramid. One should aim at having a set of generic units available with highest possible degree of quality. Examples are position, speed, heading, draught, destination port and, similarly for functions, the main functional blocks for a ship.

4.8.6 Functional Description of Applications (the Function Bloc*k* concept)

To capture functional relationships between information elements and to facilitate a modular and flexible structure it is convenient to use the concept of function blocks. This concept has been used in MMS, MiTS, Fieldbus and other contexts. A general function block is shown in Figure 4-9.

The function block consists of the following parts:

- Inputs: These are information elements read from other function blocks.
- Outputs: These are information elements generated as a result of operations within the function block.
- Status: These are readable information elements describing the status of the function block.
- Physical effects: These are descriptions of effects of operations in the function block on the physical entities outside the control system, e.g., on actuators. It can also be descriptions of effects of physical events on the outside of the ISC on the state or outputs of the function block.

- Parameters: any values affecting the function of the function block, e. g. set points, mode control.....
- Function: This is a description of the operation performed by the function block.

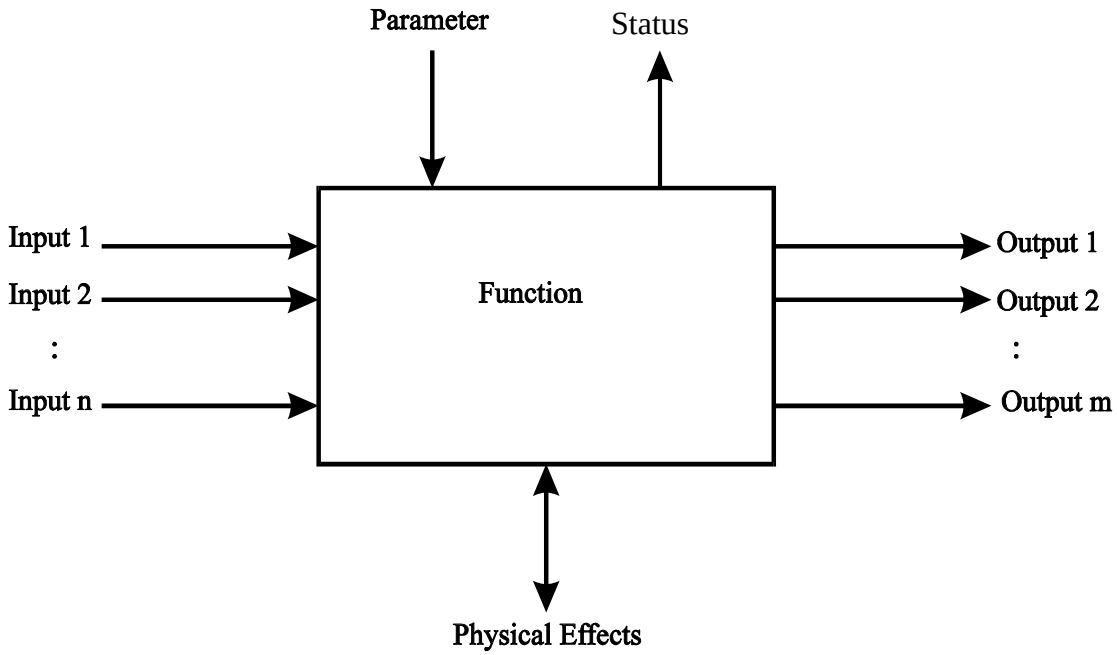

Figure 4-9 - Function Block

The basic idea is that functions in the system shall be made up by one or any higher number of function blocks. One physical node in the system can be implementing one or more functions.

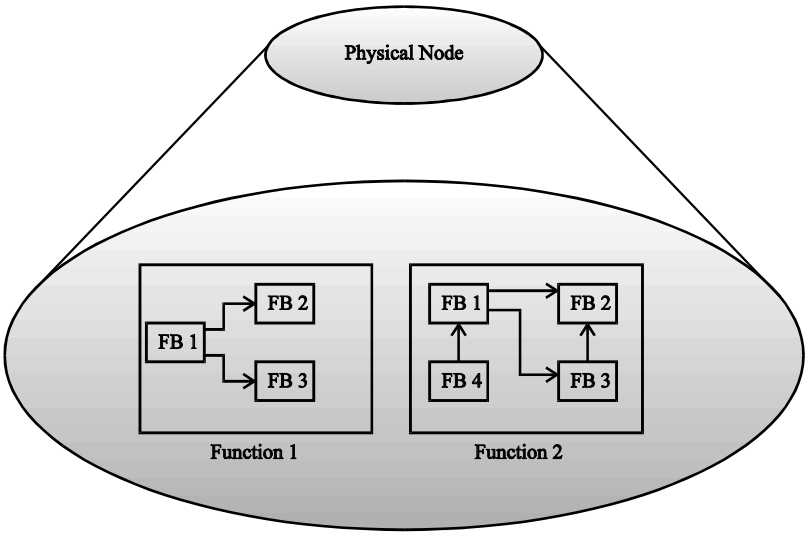

Figure 4-10 - Structure of Function Blocks, Groups of Function Blocks (also a Function Block) and Physical Nodes.

It has to be pointed out that the concept of the function block facilitates the de-coupling of internal information flows (given by the input/output-channels) and information exchange between the ISC and its environment (Parameters/Status and Physical Effects). This results in clearly defined system boundaries and interfaces.

Applications can be described by function blocks both on the general level for connection to the main ISC network and function blocks intended for connection to more field-bus type networks. This is illustrated in Figure 4-11.

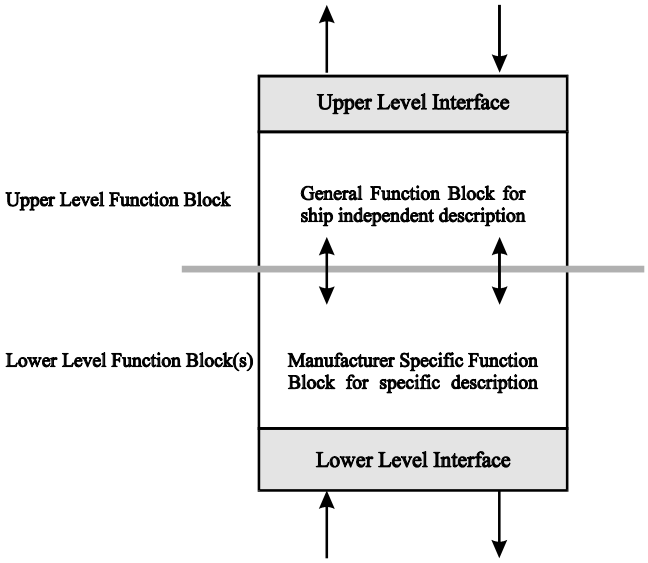

Figure 4-11 - Application with Low Level and High Level Interface

The high level function block could be part of the general and ship independent description of the ISC. One or more lower function blocks could in principle be an application specific or manufacturer specific specification.

4.8.7 General Information Models

In principle all applications can be represented by function blocks, however it is envisaged that some applications (e.g. applications with an undefined SIL) may not make use of the function block representation.

There is also a need to capture how high level information entities has been arrived at, i.e., what lower level elements were used and how were they processed.

To care for these two requirements it is proposed to use an entity-relationship model, e.g., implemented in EXPRESS. This model should capture all or the most important relationships between the information elements (data objects) in the complete ISC system or in the part of the system that is modelled. It is assumed that a set of ship generic function blocks provide a set of basic information elements from which the model can be extended. Other function blocks must provide mechanisms by which information elements in the model can be retrieved from the system.

One must assume that information elements can be made available both through normal function blocks and through an information model gateway. One must further assume that the latter will have a lower quality of **information transfer** service, e.g., no subscribe, only request/Acknowledge, than the normal function block.

4.8.8 E*x*amples of Function Bloc*k* representations

Two examples illustrating the broadness of the usability of the function block concept have been selected and presented in the following. As an example of a function block representation of a high level application the damage control decision support component of the Emergency Management application has been selected, the function block representation is illustrated in Figure 4-12.

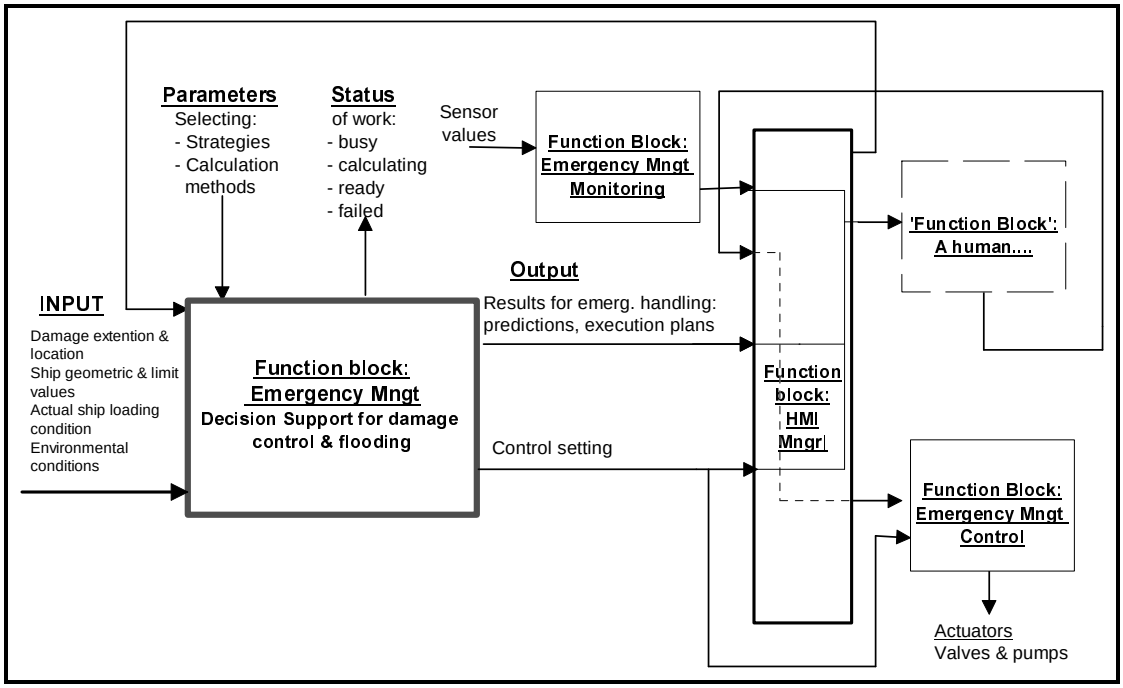

Figure 4-12 - Example of Function Block Representation of a High Level Application, Damage Control Decision Support component of the Emergency Management application in context

Short descriptions of the data and control flows for Damage Control function block are given in the following:

- The **input** data comprising damage description (damage extent), static and dynamic ship data, and environmental conditions can be provided either by the user, by sensors or by the common database.
- The **parameters** controlling the behavior of this function block determines the selection of strategies and calculation methods for optimized decision support.
- The **status** of the calculation will be indicated as e.g. “busy” or “ready”.

The **output** (decision support in the form of predictions, execution plans for remedial actions etc.) will be presented to user through the HMI Manager function block and upon acceptance by the user the remedial actions (= control settings = physical effects) will be executed by the control component of the emergency

management system. The user will always have a possibility to overrule the proposals provided by the decision support system and request execution of his own suggestions for remedial actions.

As an example of a function block representation of a low level application a valve controller has been selected, the function block representation is illustrated in Figure 4-13.

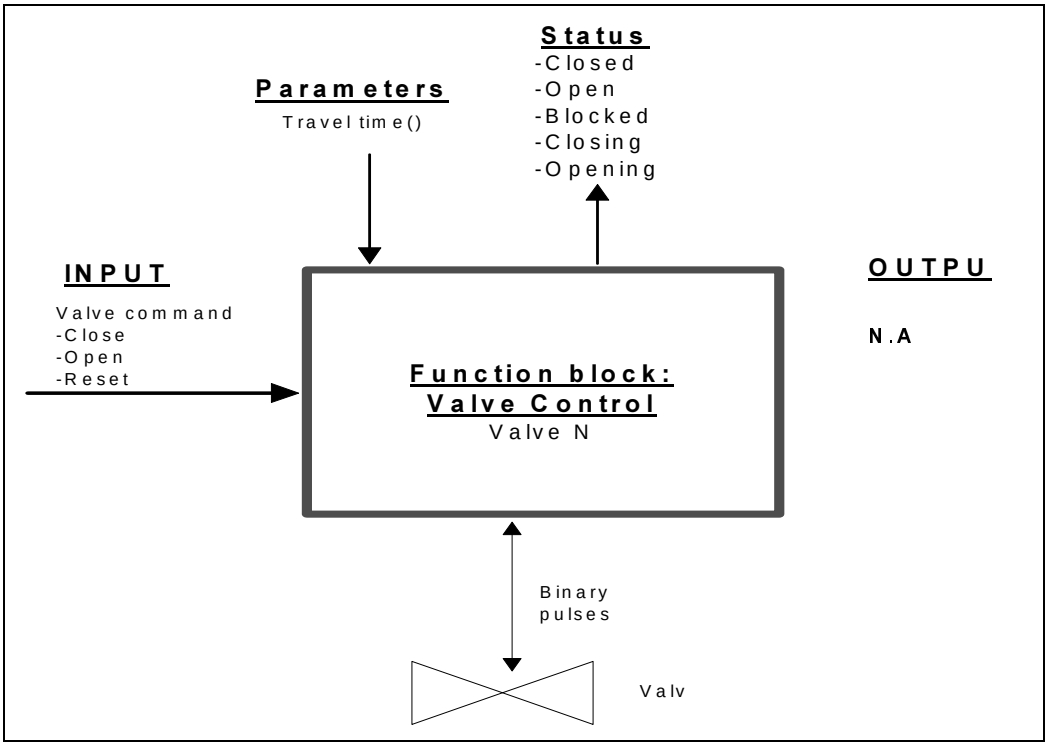

Figure 4-13 - Example of Function Block Representation of a Low Level Application, Control of a simple Valve.

Short descriptions of the data and control flows for valve control function block are given in the following:

- The **input** data comprises the valve commands (close, open or reset)
- The only **parameter** input is the requested travel time for the valve.
- The valve **status** information is either closed, open, blocked, closing or opening
- No **output** is generated by the function block
- The **physical** effects is the actual valve position

4.9 Re*q*uirements to Application Layer of the DISC standard.

In addition to the requirements imposed by the Validation and Verification and the HMI layers of the DISC standard the applications shall conform with the following requirements:

- The structure of the application shall support the concept of function blocks and comply with the requirements listed in section 4.11.2.
- The application shall make its services available to other applications. All information flow to or from an application shall be exchanged through the ISC Network. Applications should use a high level, formal language for interchange of information and knowledge through an ISC network which bridges disparate computer programs on different hardware and software.
- The application shall be modularized such that other applications can provide services (input) to components of the application.
- The application shall (to the widest possible extent) be distributable on several machines.
- The application shall comprise imbedded help, training and simulation facilities.
- The application shall support diagnostic and fault-detection procedures in accordance with the actual SIL-level.

4.10 Definition of the Architecture Layer of the DISC Standard

4.10.1 Scope

The System Architecture layer represents the implementation of an ISC system and also the framework for building applications. The objective within the R&D project DISC is to provide a framework for the integration of the physical components of the system which guarantees maximum flexibility and modularity. To do this it needs to consider (at least) three different description forms:

1. A framework for the organisation of information elements in the ISC (see 4.8.7).
2. A framework for the organisation of functional relationships (modules) in the ISC (see 4.8.6).
3. Bridge the gap between one information or function specification format and several possible hardware and software realisations (see 4.10.2).

4.10.2 Different views of the system

The drawing in Figure 4-14 illustrates three different views of the control system.

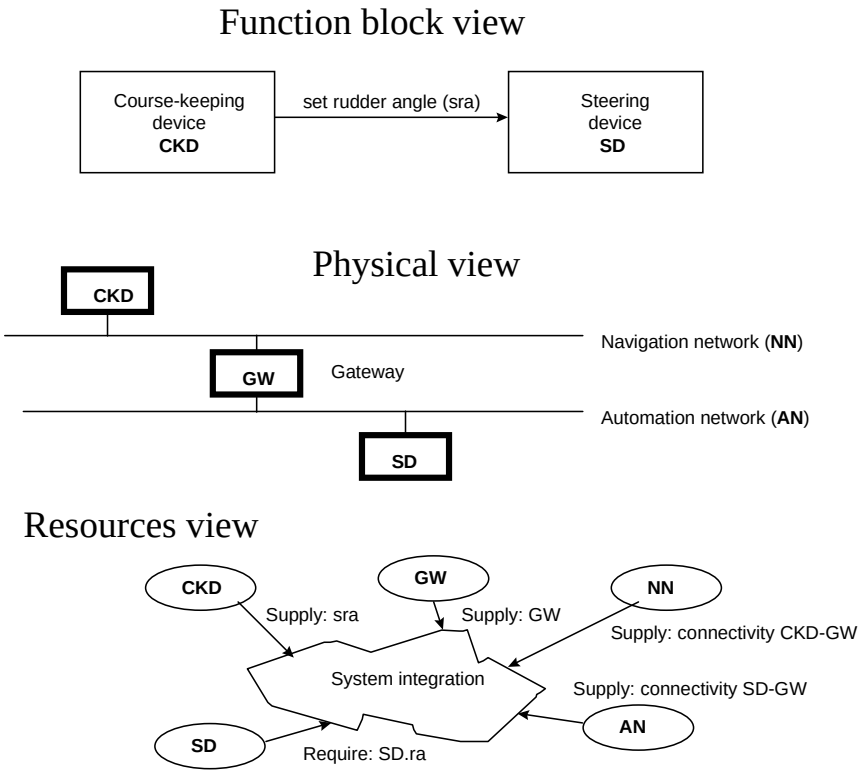

Figure 4-14 - Three views of a system

These views are:

1. **The function bloc*k* view**. This view captures a relatively high level and abstract description of how the different applications require and supply services to the integrated control system. Services that can be captured here are information exchange, CPU and HMI services.
2. **The physical view**. Any system described by function blocks can be implemented in any number of ways. This particular structure illustrates a system implemented with two distinct networks (bridge and automation) with some kind of gateway between them.
3. **The resource view**. This view captures the bridge between the function block specification and the physical implementation. The idea is that each component of the system describes what resources they require and supply in the physical system. The resource requirements from the function blocks are indicated as ovals as is the resources supplied by the physical devices and function blocks.

Note: We expect that data communication requirements on the resource view level will be represented in a manner compatible to the general resource requirements specifications that have to be defined on this level.

The purpose of this division into several views are as follows:

1. Abstract representation mechanisms (function blocks) shall be convertible to specifications of resource requirements for a practical implementation. The function block representation shall be independent of physical realisation. Several realisations shall be possible for the same function block structure (see Figure 4-15 and Figure 4-16 - Implementation on one CPU).
2. It shall be possible to construct tools that can generate configuration information for the various physical devices that make up a completed ISC system.
3. It shall be possible to analyse the physical realisation in the resources view for implementation dependent restrictions, e.g., real-time, before the design is committed to hardware.

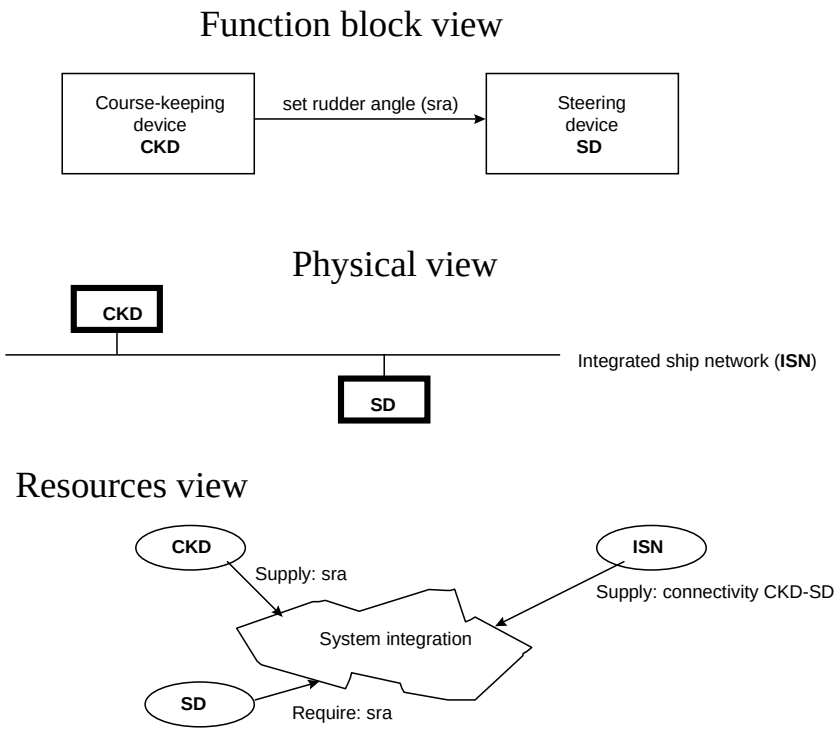

Figure 4-15 - Implemented on One Network

Figure 4-15 and Figure 4-16 - Implementation on one CPU show other realisations of the same function block and try to indicate the changes we see in the resources view. Figure 4-14 showed two networks connected together with a gateway. This has a consequence on the resource level by introducing the gateway resource and requiring some kind of integration of requirements and services from different sides of the gateway.

Figure 4-15 shows the same system implemented on one network. This means that the resources provided by the two network ovals and the gateway in Figure 4-14 is represented as one network resource in Figure 4-15.

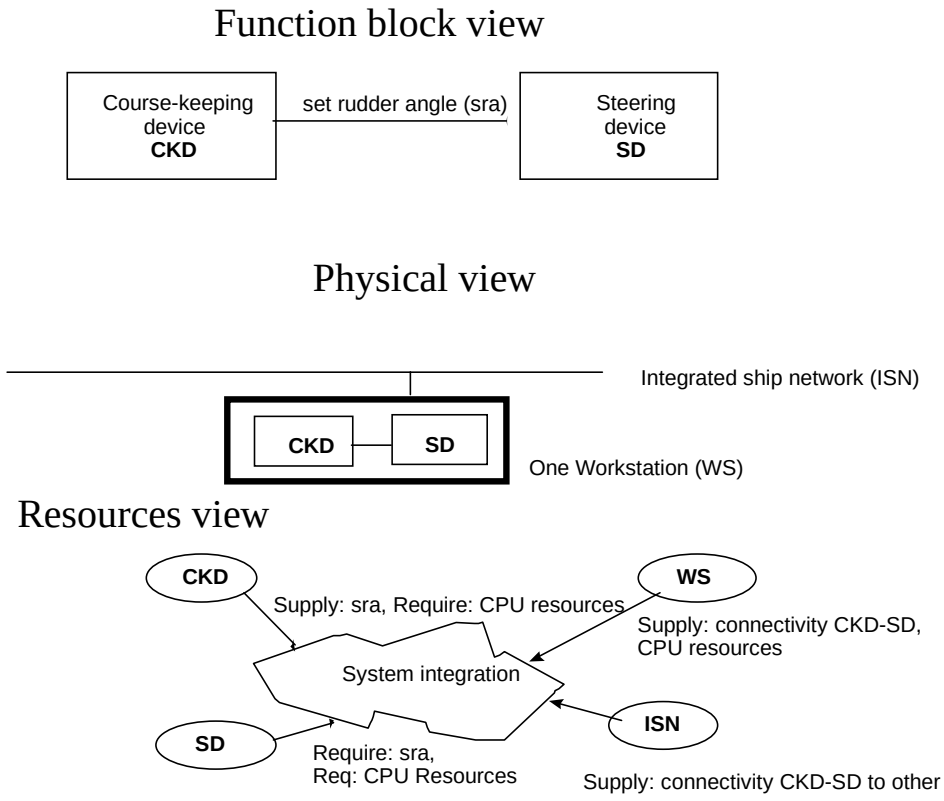

Figure 4-16 - Implementation on one CPU

Figure 4-16 shows the implementation in one CPU. In this case the resource requirements (and services) must include also CPU (or rather program execution facilities). The CPU will also take care of the arbitration of data requests and services. Note that the ISN service is still included to give other devices access to the information exchanged between the two function blocks.

4.10.3 The Application Layer Interface

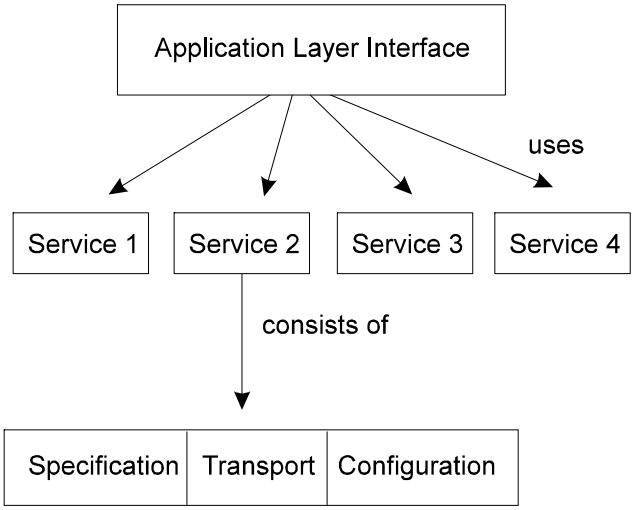

Figure 4-17 - The Application Layer Interface

The Application Layer Interface (ALI, see section 4.8.1) is the interface between each individual application and the rest of the system. The ALI provides services to the application. Each service provided has three different aspects (Figure 4-17):

1. The «content» of the service must be supplied by some other application (or physical entity: Network, CPU, memory) in the system. Some applications or entities will supply these services, other will use them. Thus, the service have to be «**transport**ed» between these entities.
2. It is necessary to have means to give and retrieve information about what services are available. This is the **specification** part of the service.
3. It is necessary to have means to establish and maintain links between the user and supplier of the service. This is the **configuration** part.

Note that the transport of services will be divided into two groups dependent on whether it is physical resources or information that is transported.

4.11 Re*q*uirements to the System Architecture

4.11.1 Re*q*uirements to the Framewor*k*

The System Architecture layer, as part of the reference model, shall facilitate the description of integrated ship control systems in a way that allows:

- The use of information already contained in the system in new functions introduced into the system at a later time (flexibility and information availability).
- The exchange of different manufacturers' equipment implementing the same function (modularity and interoperability).
- The specification of requirements to the exchange of information in the ISC (quality of service, restrictions to enhance cost-effectiveness of system).
- The verification that requirements are met in a physical realisation of the system (safety, security, reliability, maintainability, availability).
- Flexibility in implementation, e.g., different technical trade-off's can be made in different instantiations of systems. Different architectures and different communication protocols should be usable.
- Facilitate exchange of data between applications transparent of location. Also with regard to ship and shore.
- Consistency in implementation, i.e., that an implementation that has been verified will also work as expected.
- Ability to handle today's systems as well as tomorrow's.
- The use of new approaches for the implementation (e.g. the Internet-implementation language 'Java' or Software Agents).

4.11.2 Re*q*uirements to the Functional Description

The function block (see 4.8.6) shall capture the requirements to the function implemented or to the information sent or received by the function block. For the function block itself the following requirements shall be stated:

- Criticality of function (e.g. by SIL - Safety Integrity Level)
- Distributable on several machines

For the information that is sent or received, the following requirements shall be stated:

- Timeliness (response times)
- Time consistent data sets
- Required or maximum sample rate
- Maximum number of clients
- Criticality
- Security related attributes (restricted access)

These properties shall complement the actual value of the information object. The list of properties shall be as extensive as possible, i. e. no unnecessary restrictions shall be put on the description structure. One can use ASN.1, EXPRESS, MiTS Companion Standard, C++ or any other suitable language to describe the elements. The following requirements to the description language apply:

- Machine readable
- Unambiguous
- Human readable
- Tools available

The description language has to facilitate the representation of the following contents:

- Information entities
- Functionality
- Attributes
- Resource use (e. g. CPU load, maximum sample rate) for supplying and requiring information transfer services

All information elements shall be individually distinguishable (have unique names) and they should be complemented by a set of attributes specifying the quality of the information:

- Raw or processed
- Origin of values used in processing (see also information model requirements)
- Accuracy
- Time stamp
- Status of information generator (OK, suspicious, bad, off)

One will also need to specify the type of information exchange one requires:

- Request/acknowledge (transaction)
- Subscribe (reliable, unreliable)

One should also be able to capture object oriented principles in the function block paradigm:

- Meta-models and instantiations covering a number of different types of functions that can be individually instantiated.
- Specialisation of one meta-model into several sub-models. This can be used to enable version variants and generalisation of some functions, e.g., valve control.
- Aggregation where some function blocks can be grouped together into one unit.

Apart from the normal application-oriented function blocks one will also need some meta-functions representing aspects of the data communication mechanisms, e.g., alarms from the transport level itself or possibly general database services, e.g., for trending of certain information elements.

4.11.3 Re*q*uirements to General Information Models

The general information model shall provide a structured description of all information elements available from the integrated system. The model can be used for information retrieval from the system or for, e.g., diagnostics and supervision purposes at the system management level.

The information model shall be:

- Machine readable and unambiguous
- Human readable (as one incarnation)
- Supported by tools
- Based on standards

The model will be organised according to the abstraction pyramid (cf. Section 4.8.5) so that generality and ship independent increases as one moves upward in the model. It is an aim that it shall be possible to trace the origin of all processed information to its source.

The data models shall be useful for on-line information as well as for the organisation of information collected by, e.g., a trending or recording data base.

It shall be possible to do retrieval of information described by the model through one or more commonly available data-base mechanisms, e.g., ODBC. This type of retrieval may or may not support real-time quality of service.

One assumes that the model will contain at least two views:

- A functional view linking function block inputs and outputs to each other and, in a sense, describes the control and supervision function of the system.
- A data refinement view linking status outputs to each other which describes how high level information entities are derived from lower level entities.

The model may be static (generated off line from the system configuration information) or dynamic (generated on-line by submissions from the active function blocks in the system).

The information model should be easy to link to other models describing, e.g., the structural or the dynamic properties of the ship (e.g., EXPRESS/STEP models).

The model shall also be organised in a way that allows an object oriented description of the system and the information entities.

4.11.4 Re*q*uirements to Services

To identify tools and technologies used by the system architecture, the services to be provided by the system architecture have to be defined first. To do this, the concept of function blocks is used. An application will be modelled as one or more function blocks. The basic idea is that there is a mapping between these function blocks and the services required by the function blocks. From the function blocks the services used can be

identified. This approach gives a hierarchical structure which establishes a link between low level and high level use of services.

Services can be classified in the following way:

- Services for physical resources
- Generic services
- Special services/extended services

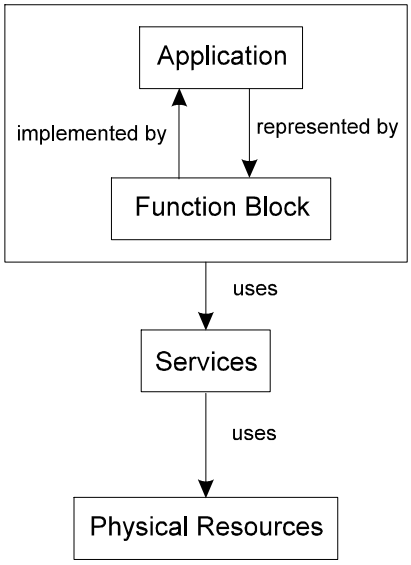

Figure 4-18 - Applications, Function Blocks, Services and Physical Resources

4.11.4.1 Physical Resources

The following services are required to cope with the physical resources of the system:

- Data transport facilities: give information about transport delay (dependent from network bandwidth, load etc.), redundancy and quality of service.
- Program execution facilities: give information about CPU load, existing/required memory (slow memory, e. g. hard disc, fast memory, e. g. RAM), OS-services
- Services to manage HMI resources (e. g. VDUs, printers, keyboards, pointing devices)
- File services, storage handling (CD, discs, tape etc.)
- Support of restrictions due to Safety Integrity Levels (SIL).

4.11.4.2 Generic Services

- Facilitate exchange of **objects** between applications transparent of location, hardware and operative systems. By object is meant a set of data elements and a associated set of presentation attributes and possibly code.
- General information exchange based on the function block approach:

  - Information model, i. e. information type definition
  - «Information about information», i. e. identity and properties of information source, information quality.
- Time distribution and control
- Have means to ensure, or at least, detect and report that some application always is ‘alive’.
- Network load supervision and control in order to prioritise applications in cases of overload.

4.11.4.3 Special and E*x*tended Services

- Provide access to non-real-time, historical data and static data (e.g. ship particulars). This should be based on a combination of subscription and on standard data access means like SQL and ODBC.
- Redundancy of safety critical applications should be ensured by context based dynamic distribution of applications and resources (e.g. CPU and VDU).
- Master arbitration protocol, management of several clients controlling one device or application.
- Queries in the information model, e. g. the ability to request information entities with certain properties
- Implement generic services to support applications with specific requirements such as:
  - «System Manager»
  - Black Box
  - Alarm System
- Support or implement services to detect failures of functionality at the system architecture level, e. g. faults of the data transport mechanisms.

4.11.4.4 Configuration Management

The configuration management which is encapsulated in the Application Layer Interface handles the actual configuration and gives access to the actual and the intended configuration of the system. It should provide the following functions:

- Configuration version control.
- Monitoring of the actual configuration; sending of alarms on network failures, lost of clients etc.
- Query functions to access information about other clients
- Control functions to change configurations.
- Identification of clients.
- Debug functions and backtracking of information propagation.
- Access to an internal representation of the actual and the intended configuration.
- priority of requests
- monitoring and control of network load, i. e. identification of the number of clients, protection mechanisms (enable/disable access to the network for certain clients)
- Security.
- Name look-up
- Maintain and make available a list, on request, with the following information:
  - Applications present in the ISC system
  - All services presently available in the ISC system

- Attributes of present applications and services such as safety criticality, priority, originator etc.
- System/application status.

4.11.5 Re*q*uirements to the Technical Implementation

All requirements to the system architecture shall be implementable with reasonable efforts by modern technology.

Existing standards shall be taken into consideration as far as useful. If possible, existing standards for protocols and bus systems should be used.

The number of different information transfer mechanisms (protocols, etc.) needed to fulfil the requirements should be as few as possible. One should consider if the use of only one overall protocol might be applicable.

4.11.6 Safety enhancement through the use of function bloc*k*s

The function block approach can contribute significantly to the reduction of complexity in integrated systems. Reduction of complexity is an important factor for efficient assessment of the system safety level (SIL) through the application of validation and verification techniques as described in chapter 6.

The reduction of complexity comes through the modularization principle used in function blocks. One will get this reduction because:

- The interface to the function block (via the ALI) guarantees that the function block cannot influence on the system in any other way than as described in that interface.
- The behaviour of the function block as is evident on its interface, will normally be a sub-set of all the behaviours implemented in the function block. This is particularly visible in hierarchies of function blocks.
- The functionality of the ALI needs to have limitations that disallow common internal errors inside the function block to propagate to the surrounding system.
- The functionality of the ALI should, if possible, be limited so that the system of function blocks lends itself to analysis.

To achieve this reduction of complexity one needs to design the ALI with great care. Not only must the ALI be designed as described above, but it must also guarantee that it has no side effects. One will require that the ALI is built to a SIL at least equal to the highest SIL of any of the function blocks it connects. This does not only apply to the ALI, but also to any component going into the realisation of the ALI, e.g., networks, protocols and gateways.

4.11.7 Use of SIL in function bloc*k* systems

As described in the previous section, one need to design the ALI and its corresponding components to a high enough SIL. This is necessary to allow function blocks of lower or equal SIL to be interconnected.

When one looks at the function blocks themselves, one part of the design process will be to assign SIL requirements to the function blocks to achieve a high enough SIL on the functions that the function blocks shall implement. One should keep in mind that a too high SIL may require very costly and, in principle, unnecessary V&V procedures. A well tried method to reduce system SIL is to use a hierarchical approach where the high SIL function blocks are kept small and simple and at the bottom of the system, close to the controlled or monitored process. Increasingly lower, but more complex, function blocks can be built on top of this.

4.11.8 Object orientation in function blocks

Encapsulating large parts of an application's behaviour inside the function block and only give access to those behaviours that are of interest to connected function blocks, is similar to the data abstraction principle used in object orientation, except that it is employed on behaviour. Not only in this respect can function blocks be seen as a variant of object orientation:

- Specialisation needs to be employed to enable the implementation of special behaviours or interfaces in general function block templates.
- Aggregation is necessary to allow one function block to be composed of smaller and lower level function blocks.
- Data abstraction and encapsulation is directly employed by the function block.

5. DEVELOPMENT OF SKELETON STANDARD

5.1 Verification & Validation

5.1.1 Identification of areas to be described by the S*k*eleton Standard

Below is a list of areas for V&V assessment. The principles described are general and thus relevant also for ISC-systems.

5.1.1.1 Generalized DISC V&V Assessment Method Rules.

The assessment method is to address following 5 abstract issues:

1. Process (product development)
2. Product (an ISC system or any of its components)
3. Environment (ship external factors such as sea-state, traffic, geographical position and ship internal factors such as lighting, electromagnetic compatibility, temperature, permitted cargo types)
4. Human Factors
5. Conformance

For each of these issues the following key principles is to be applied, as far as relevant.

5.1.1.2 Key Principles

1. The deployment of new systems and the integration of existing systems may introduce new hazards and therefore require a risk-based approach, starting with a suitable hazard analysis. The justification for the level of risk shall be based on safety targets set by IMO or best current safety levels commonly accepted in shipping.
2. Human operator must be ”in the loop”. Evaluation of system safety should include the **total system**, that is both the technical equipment and the operator using it in a realistic range of environmental conditions. The requirements for training operators should be taken into account.
3. Verification and validation are needed throughout the lifecycle of the system, including the entire development process and any modifications. Functional testing alone is not sufficient proof of safety.
4. In order to verify or validate a system or component, traceable documentation is required for all phases of the development process. (i.e. system requirement specification, safety requirement specification, software safety requirements, test methods and results etc.). This requirement implies that sufficient document control and software configuration management are applied.
5. System integration needs to be validated and verified. The verification and validation of the components of a system are in themselves not sufficient evidence of safety. The process of system integration is itself subject to all of the requirements of this standard.
6. It is expected that ISC will result in different systems sharing resources (such as networks and processing hardware). When this happens it is necessary to demonstrate adequate isolation of safety critical components, to ensure that failures in other components cannot contribute to failures in the safety critical components.

7. It is expected that COTS components (Commercial Off The Shelf) may be used in developing integrated systems. When this is done a sound demonstration of component reliability should be provided, based on in-service reliability data.

5.1.2 Applicable Standards

1. IEC 1209: Integrated Bridge Systems (IBS)[2], Operational and Performance Requirements, Methods of testing and required test results (draft)

2. relevant IACS rules

3. relevant IMO rules

4. ISO9001: Model for quality assurance in design, development, production, installation and servicing, 1994[3].

5. ISO9000-3: Quality Management and Quality Assurance Standards: Guidelines for the Application of ISO 9001 to the development, supply and maintenance of software[4].

5.1.3 Standards to be Adapted

1. IEC1508: Functional safety: safety related systems (draft)

2. PRO96 Chapter C Assessment Procedure in Generalized Assessment Method (GAM) Rules CASCADE Deliverable CAS/LR/WP2.T3/SM/D2.3.1/0.C, Sept. 1996[5].

3. PRI96 Chapter D Assessment Principles in Generalized Assessment Method (GAM) Rules CASCADE Deliverable CAS/LR/WP2.T3/SM/D2.3.1/0.C, Sept. 1996[6].

5.1.4 Missing Standards

5.1.4.1 Verification and validation of COTS

Verification and validation of the integration of COTS is not fully addressed in any published standard or guideline documents. Some of the items that need to be addressed are:

- use of past operational history as proof
- Record of reliable use for many years - can it be used as proof and how? (documentation, but what about e.g. WINDOWS NT)
- number of installations performed.
- extent of use (operational history of all features)
- possibility of regression failures on new installations.
- range of past operating environment has to be applicable of the intended future use
- relation to the safety target

Because any given operational profile can not be guaranteed to excite failure modes which may be important in the future, the use of operational history for a SIL 3 or SIL 4 system should be complemented by the use of an analytical technique such as fault tree analysis.

5.2 User/HMI Layer

5.2.1 Standards and guidelines.

Standards

In Europe, the European Directive 90/270/EEC on display screen equipment addresses minimum requirements necessary to meet objectives for the usability of operator-computer interfaces. These requirements refer to general issues like 'suitability for the task', 'easy to use' and 'applying principles of software ergonomics'. Although these issues seem rather vague, standards are under development by international organizations like the International Organization for Standardization (ISO) which may serve as a reasonable alternative for demonstrating compliance with good ergonomics practice. The general consensus is that these ISO standards will become the most likely candidates for future international standards. ISO comprises 107 countries all over the world, including the USA. Wherever appropriate, ISO standards are adopted by the European Standardization Organization (CEN) as part of the creation of the Single Market. CEN standards will also replace national standards in the EC and EFTA member states (e.g. DIN, BS and NEN). Furthermore, ISO or its European equivalents are often referenced to by EC member states to implement the obligations placed on them by the European Directive 90/270/EEC on ergonomic requirements for display screen workstations. In doing so the member states transpose these obligations into appropriate national laws and regulations. So, the message is, that the best strategy is to focus on the ISO standards.

Guidelines

In the 80's, the Graphical User Interface (GUI) became familiar to more and more users who did not have specific computer expertise. These interfaces were based on the 'WIMP'-concept (Windows, Icons, Mouse and Pull-down/pop-up menus). In particular the direct manipulation, provided by WIMP-interfaces, was thought to improve the usability for non-expert users. In the 90's, GUI's are becoming widespread and a standard "look-and-feel" has been developing for user interfaces of different platforms (Mac, Motif, PM, Windows). Usability guidelines and style guides have been developed for these interfaces. Many of these guidelines are of a more or less overlapping nature (see, for instance, Williges et al., 1987 for a classification scheme or Smith & Mosier, 1986 for an extensive review of guidelines for software interface design).

Currently, a new development has been started. In CERN, Switzerland, a first version of a graphical user interface for hypermedia, the contraction of hypertext and multi-media, was developed for Internet, called Mosaic. Web browsers, such as Mosaic, can be used to search for multi-media information in databases via a network. New web browsers are being developed and used by more and more people (Netscape Navigator, MS Explorer). These browsers involve a new "look-and-feel" and standards are being set currently. As this technology is developing fast, empirical-founded guidelines and style guides are hardly available.

5.2.2 Identification of areas to be described by the S*k*eleton Standard

a) The following list is general and related to user interface for any application:

Wor*k*station design and arrangement

- Location of visual display units and user input devices

- Allocation of functions to screen-based systems

User input device and display unit design

- General
- Alarm handling
- Remote control
- User input devices
- Visual display units
- Colors
- Illumination
- Requirements for preservation of night vision

Additional re*q*uirements to screen based systems

- Computer Dialogue
- Application screen views
- System Access
- Decision support
- Planning

Design of Wor*k*place

- Control Suite: all compartments, backup, security, relaxation areas, machine and equipment rooms, access etc.
- Control Centre: location of people and equipment, task match, manning issues, team task allocation, communications etc.

Wor*k* Environment

- Vibration
- Noise
- Lighting
- Temperature
- Ventilation
- Surfaces
- Colors
- Safety of personnel

b) A set of requirements will also be given for specific applications, when found necessary. These requirements will only have be complied with when applications covered by these requirements are being installed on board a ship. There is no requirement that the applications should be installed on a particular vessel.

5.2.3 General Description

List of standards and studies:

The following is a list of standards and work identified as being related to the HMI task. The list is not exhaustive, but should be used as a basis for future work. Selection for the list is done in 5.2.3 and 5.2.4.

ISO 13407, Human centered design processes for interactive systems[7]

ISO 11064, Ergonomic design of control centers[8]

ISO 9241, Ergonomical requirements for office work with visual display terminals (VDTs)[9]

ISO 8468, Ship's bridge layout and associated equipment - Requirements and guidelines[10]

IEC 447, Standard directions of movements for actuators which control the operation of electrical apparatus[11]

EN 894, Safety of machinery - Ergonomic requirements for the design of display and control actuators[12]

IEC 1209, Integrated bridge systems (IBS), operational and performance requirements, methods for testing and required test results (draft).

Standards/wor*k* related to the various functions/tas*k*s:

Common to all tasks:

- ISO 11064
- ISO 9241
- ISO 8468
- IEC 447
- Rules for classification of ships
- ATOMOS reports on information requirements and support
- MiTS Style Guide v. 3.0

Specific functions/tasks:

a) Emergency control workstation (fire detection/extinction, contingency handling, muster handling, etc.):

b) Navigation:

- IMO performance standards
- IEC 1209 (work on integrated bridge control)
- Framework for an ergonomically assessment of the human-machine interface of ECDIS equipment (TNO, December 1995)[13]
- Framework for an technical assessment of the human-machine interface of ECDIS equipment (TNO, December 1995)[14]

c) Cargo handling:

- Rules for classification of Ships (e.g. DNV Additional Class notation CCO)

d) Propulsion/steering/power generation/ballast/bilge (engine room operation):

- No specific standards identified

e) Hull integrity (load stability, stress, load calculation):

- No specific standards identified

f) Anchoring/mooring:

- No specific standards identified

g) Maintenance:

- No specific standards identified

h) Ship-shore / ship-ship communication:

- No specific standards identified.

I) Administrative functions (crew, purchase, finance):

- ISO 9241

5.2.4 Applicable Standards

Several of the standards referred to (e.g. ISO 13407) are related to the design process, and do not give requirements to the product itself. The general idea behind the DISC initiative is to make a set of requirements for the integrated ship control systems and to make the requirements in a way that it is possible to verify compliance with the requirements.

On the other hand, it will be useful for the manufacturer designing a system, to be in compliance with the ISC standard to have references to standards giving guidance to the design process. In other words, the standards will be referred to, but the text will not be included as part of the ISC standard.

The set of requirements for HMI (except the IMO performance standards) will be extracted from the various applicable standards/requirements and included in the ISC standard.

a) Standards/requirements referred to only:

- ISO 13407, Human centred design processes for interactive systems
- ISO 11064, Ergonomic design of control centers
- ISO 9241, Ergonomic requirements for office work with visual display terminals (VDTs)

b) Standards/requirements to be part of the ISC standard, but only referred to:

- IMO performance standards

c) Standards/requirements to be fully or partly included in the ISC standard:

- ISO 8468, Ship's bridge layout and associated equipment - Requirements and guidelines
- IEC 447, Standard directions of movements for actuators which control the operation of electrical apparatus
- Rules for classification of Ships
- ISO 1209 (work on integrated bridge control)
- Framework for an ergonomically assessment of the human-machine interface of ECDIS equipment (TNO, December 1995)

5.2.5 Standards to be Adapted

As the different related standards are to be used as a basis for a new ISC standard, there will be no need for adaptation of existing standards.

5.2.6 Missing Standards

- Standards for multipurpose user input devices/visual display units
- Standards for HMI API
- Standards for cargo management
- Standards for usability of hypermedia systems
- Standards for emergency management

5.3 Applications Layer

5.3.1 Identification of areas to be described by the S*k*eleton Standard

See chapter 4.8.2 , Types of Applications.

5.3.2 General Description

See chapter 4.8.3, Generic Application.

5.3.3 Applicable Standards

Many of the standards already mentioned in the Validation and Verification and the HMI layers are relevant for the Application layer as well. However in the following are a few additional standards listed:

- ISO 10303 Standard for the Exchange of Product model data, notably parts on EXPRESS (data definition (and validation) language) and Data Access Interface[15].
- ODBC: database queries.
- IMO performance standards.
- IEC standards.

5.3.4 Standards to be Adapted

- ACL : Agent Communication Language which includes a knowledge query language (KQML - Knowledge Query and Manipulation Language) and facilitates high level knowledge interchange (KIF - Knowledge Interchange Format)

5.3.5 Missing Standards

- ISO STEP product model standards for various areas of ship operation.
- Curriculum and user interface for computer based training modules
- Standard for on-line, interactive documentation for manuals etc., for example in the form of SGML document type definitions

5.4 Architecture Layer

5.4.1 Identification of areas to be described by the S*k*eleton Standard

The main objectives of the skeleton standard for the system architecture are:

- To give a clear definition of the framework as explained in chapter 4.7. This topic contains the framework for the system architecture layer itself as well as the functional description of the system architecture by function blocks (see 4.7.3) and general information models (4.7.4).
- To establish requirements. Basing on the standardized framework user requirements have to be defined covering the different aspects of the system architecture. Especially, the following fields must be taken into account (cf. IEC 12178): Integration issues; differentiation of information; requirements regarding transfer-/transaction times; resilience; topology aspects; system management.

5.4.1.1 Framewor*k* for the System Architecture

The skeleton standard shall contain a description of the framework as given in section 4.7, including an unambiguous definition of the three layers (generic layer, component layer, layer of sensors and actuators, cf. 4.7.2) forming the basis of the framework. Also the interfaces between these layers have to be defined clearly.
The skeleton standard shall contain a determination of the different components of the system architecture level. This is mainly necessary to distinguish this level from the application level and the human-machine interface level of the model.

The standard shall mainly take into account the following topics:

1. Identification of the "functional entities" included in each layer of the framework.
2. Determination of requirements regarding the communication protocols used in the different layers. Establishing of standard formats for information exchange within the layers and between them.
3. Identification of standardized information entities especially in the generalized "generic layer" on top of the "abstraction pyramid" (see Figure 4-8).

5.4.1.2 Functional Description

To provide a common standard for information exchange between functional entities, a generalized functional description of the system architecture is needed. For this purpose, in chapter 4.8 the concept of function blocks has been introduced. The skeleton standard of the system architecture shall contain a clear definition of this concept. In particular, the following topics shall be included:

1. Determination, which functional entities can be described with function blocks.
2. Definition of the types of "flows" via the boundaries of the function blocks (see Figure 4-9 - Function Block).
3. Ordering of the function blocks within a hierarchical description of the system architecture (see Figure 4-10 - Structure of Function Blocks, Groups of Function Blocks (also a Function Block) and Physical Nodes.).

4. Development of a set of standard function blocks which can be used to describe generic functional relationships.
5. Supply standardized description languages to present function blocks in graphical or textual way.

Beside function blocks, also general information models are needed, which should be created as entity-relationship models. The skeleton standard shall provide description languages for this model.

5.4.1.3 User Re*q*uirements

The skeleton for the system architecture of an ISC shall cover the following fields with respect to user requirements:

- **Integration**: The main issue is to define the skeleton standard independent of the system architecture implementation. The standard shall be open for different physical network configurations, protocols and operating systems. To reach this goal, the skeleton standard should base on common industrial standards or should be at least compatible to them.
- **Differentiation of information**: The standard should contain requirements for unambiguous data type definitions. This definitions should take account not only of the type of information transmitted but also of additional properties like time criticality or accuracy.
- **Re*q*uirements regarding transfer-/transaction times**: Transfer times should be reliable and predictable.

5.4.2 Applicable Standards

All identified standards containing requirements for ISC-systems or components of an ISC shall be taken into account. Especially the following standards are relevant for the skeleton standard of the system architecture:

- **IEC 1209 (draft): IBS — Integrated Bridge Systems**. This standard establishes a set of general requirements regarding the system architecture of integrated bridge systems.
- **IEC 1162: Maritime navigation and radiocommunication e*q*uipment and systems - Digital interfaces**[16]. Low-level and high-level data transfer protocols are defined within this standard. The protocols are customized for the use for maritime purposes.
- **IMO performance standards**. These standards established by the International Maritime Organization are dealing mainly with user requirements on specialized navigational and radiocommunications equipment.

5.4.3 Standards to be Adapted

Regarding system architecture, functional description and user requirements a lot of standards exist in the field of process control and discrete parts manufacturing. Although these sectors are in general quite similar to integrated ship control, standards established for this fields can only be applied partly or have to be adapted. This section gives for the areas mentioned above an overview of standards which can be adapted for the skeleton standard.

5.4.3.1 System Architecture Framework

The following standards have been found to be applicable in this area:

- ISO 11065 Glossary for Automation[17]
- ISO 12178 Real Time Communication Architecture[18]
- IEC 92-504 Electrical Installation Control and Instrumentation on Ships[19]

Additionally, there are several industrial specifications covering this area which should be taken into account:

- EN 50170 European Fieldbus: Profibus, FIP, P-Net[20]
- SDS Smart Distributed System
- ODVA Open DeviceNet Vendors Ass.
- CAN/CAL CAN Application Layer
- OPC OLE for Process Control

ISO 12178: Industrial automation – Time-critical communications architectures – User requirements

The focus of this standard is to identify the communication needs of tightly coupled control systems. It summarizes user requirements for time-critical communication systems and provides a short explanation of these requirements. It takes into account that these systems have to allow time-critical as well as non-time-critical communications. ISO 12178 has been mainly established for the field of discrete parts manufacturing, but it is explicitly mentioned that the requirements defined within this standard are also applicable in other fields. The main emphasis lies on requirements for event driven communication scenarios, state driven scenarios are considered as a special case of them. The standard focuses primarily on a network which is optimized for dynamic, unpredictable communications sequencing.

The standard also discusses "user requirements for metrics and benchmarks which can be used to compare and manage the performance of networks".

CAN: Controller Area Network / CAL: CAN Application Layer

CAN is a broadcast-oriented communication protocol. It covers layer 2 (data link) of the ISO 7-layer model. The CAN-protocol has been standardized in ISO 11898. CAL is a higher layer protocol based on CAN. It is under the control of the CAN-in-Automation users group (CiA). CAL provides the four application layer service elements:

- CAN based Message Specification (CMS)
- Network Management (NMT)
- Identifier Distributor (DBT)
- Layer Management (LMT)

ODVA: Open DeviceNet Vendors Association

DeviceNet is a low-cost communication link based on the CAN communication protocol. ODVA publishes and distributes the DeviceNet specification which describes the DeviceNet protocol and hardware, software and communication requirements for DeviceNet products. The DeviceNet communication protocol covers layer 7 (application) of the ISO 7-layer model. It bases on Peer-to-Peer data exchange and supports a client/server system architecture. A DeviceNet node is modeled as a collection of Objects with attributes, services and behaviors. The DeviceNet specification provides a set of standard objects describing the main components in the network.

OPC: OLE for Process Control

OPC is a communication standard developed by Microsoft based on OLE (Object Linking and Embedding). The purpose of OPC is to improve the interoperability between automation/control applications, field systems/servers and business/office applications. OPC defines standard objects, methods and properties built on OLE component technologies for servers of real-time information to transfer data to standard OLE-enabled clients.

5.4.3.2 Functional Description

Within this area, the following standards are of interest. Most of these standards contain specifications of the function block approach:

- ISO 9506 Part 1+2 MMS (Companion Standard)[21]
- IEC 870-5 various telecommunication equipment[22]
- IMO Performance Standards
- IEC 1162 Part 1-4
- IEC 1131 Programmable Controllers
- ISA SP 50 Instrument Society of America
- HART Highway Addressable Remote Transmitters, DDL

IEC 1131, Part 1-4: Programmable controllers

This standard deals with programmable (logic) controllers (PLCs) as components of a control system and its peripherals like programming and debugging tools (PADTs), test equipment (Tes) and man-machine-interfaces (MMIs). The aim of this standard is

- to give definitions and define the main properties of PLC’s
- to establish minimum requirements for PLC’s
- to give a detailed specification of the most important programming languages for PLC’s
- to supply general advice and guidance for the user
- to specify the communication between PLC’s and other systems using the MMS-specification.

IEC 1131 is divided into 4 parts. The focus of these parts lies in the following fields:

- **Part 1: General information**: This part establishes definitions and contains a glossary of terms used within this standard. Furthermore it explains basic functional properties of PC's.
- **Part 2: Equipment requirements and test**: Part 2 specifies electrical, mechanical and functional requirements for PC's and peripherals. It lists, which information has to be provided by the supplier of PC's. Furthermore it contains test procedures and test equipment to be used.
- **Part 3: Programming languages**: This part contains common programming languages used for PC's. It includes textual programming languages as well as graphical languages. Among others, the function block language is specified within this part.
- **Part 4: User guidelines**: This part gives general guidelines for the application of this standard.

5.4.3.3 User requirements

Most of the standards listed in the sections above contain also user requirements. Especially the user requirements contained in ISO 12178 and IEC 1131 should be adapted to the skeleton standard of the system architecture of ISC.

5.4.4 Missing Standards

Similar to ISO 12178 there should be a standard which identifies user requirements for the architecture of ISC-systems. This standard should also establish the framework of the system architecture as explained in section 4.7. This framework should be the basis to define the following topics:

- General structure of the system architecture
- Components of the system architecture of the ISC
- Establishing of requirements to the framework in accordance with section 4.8.1
- Requirements for network protocols depending on the different layers of the framework (cf. section 4.7.2)
- Definition of data formats and standard data types to be transmitted via the network
- Customizing of standardized methods for the functional description (function blocks, entity-relationship models) to meet the requirements established in section 4.8.2 and 4.8.3

6. TECHNOLOGIES

6.1 The Verification & Validation Layer

6.1.1 The approval process

The approval process is an instrument for the manufacturers (suppliers), classification societies, national MSAs (Marine Safety Authorities) and IMO to ascertain that ISC systems are sufficiently safe.

For the manufacturers there should be instruments to enable them to carry out the approval process as smoothly as possible.

The modularity expected in the future makes it possible to integrate the total ISC system from hardware, software and architectural components coming from a variety of manufacturers. It is therefore expected that the ISC systems of tomorrow will be so different both in scope and technology from what they are today, that there is a need for new approach to the approval process.

The approval can be carried out as **type approval** or **individual approval** of the system or its components. The technical assessment **method** as such does not differ significantly in these two cases, but type approval is intended for a product to be manufactured in quantities whereas individual approval is done for one-off system. The scope of the assessment is however different as the **type approval** assessment must address a described scope of use for the equipment, whereas individual approval is restricted to one particular application only, i.e. one delivery. Therefore type approval is more extensive than individual approval.

The solution proposed as part of the DISC approach is basically a concept of **modular approval** (”object oriented approval process”), which meets both the needs arising from the increased complexity of the technical systems and also cost effectiveness of the approval process.

The approval process shall address the following issues

- what sort of assessment should be done
- what is needed to have an approval
- how verification is done for components and systems built out of components

The approval process is to be done more in an **audit mode** rather than as actual testing of the system, and it is to be carried out by a notified body e.g. classification society. The difference from today’s practise is a move from complete system testing towards verifying whether the target safety level has been achieved throughout the system.

V&V shall be carried out on two levels, which are component and ISC levels. The type approval is not expected to be applicable to all ISCs installed onboard, because tomorrow as today these differ from ship to ship. Type approval is expected to be done for most components. The component level approval should address the following issues:

- the component itself can consist of approved sub-components (note: **type** approval of a component requires **type** approval of its components)
- the validity / scope of approval of sub-components

- contents of approval certificate
- SIL of the component
- context of the approval (for certain purpose of use)
- description of V&V methods used
- test records
- functional, performance and interface specifications
- what the system must do - i.e. functions
- what it should not do and how it should behave in error conditions
- how reliably it does those things (related to SIL)

The ISC level of approval should address the following issues:

- system approval is basically same as required for component approval
- the validity / scope of approval of components
- contents of approval certificate
- SILs of the ISC components
- context of the approval (for certain purpose of use)
- description of V&V methods used
- test records
- functional, performance and interface specifications
- what the system must do - i.e. functions
- what it should not do and how it should behave in error conditions
- how reliably it does those things (related to SIL)
- maintenance

For the approval process it is important to have documentary evidence that V&V is implemented on the whole product i.e. ISC. The assessor is not interested to repeat the tests himself (although assessors will normally need to witness some tests), but is interested how the tests are done and there should be records about how the tests have been done. Justification is needed on what techniques have been used. These records should be sufficient to reproduce the tests if necessary.

6.1.2 General e*x*ample of an approval process

The approval process of ISC is carried out today in four phases, which are 1) request by the manufacturer for approval of the system including provision of appropriate documents, 2) audits of documentation, 3) tests of the system and 4) issue of approval certificate.

The modular approval of the future ISCs will differ from today in the following ways:

- The responsibility for demonstrating a safe system rests on the developer. Assessors will verify that this demonstration is adequate.

- Approval certificates for the components (such as function blocks) will contain sufficient information to allow these components to be used in a variety of approval contexts.

Some specific items should be taken into consideration such as:

- record of the approval of the components

- record of development from specification to functional description to complete components including SIL assessment
- rigorous implementation of components with different SIL classification
- records of component, subsystem and system testing
- training records

6.1.3 Approval of DISC layers

6.1.3.1 Approval of function blocks

The function block approach (ref. chapter 4.11.3) gives possibility to build validated systems out of validated function blocks. This kind of approach is expected to require validating the functional and non-functional attributes (including interfaces) e.g.

- SIL of the function block (in its actual physical implementation)
- resource requirements such as processor time, memory, network use
- things it must not do (e.g. violation of memory allocation)
- implementation issues

6.1.3.2 Approval of HMI

Although training and experience can be validated and verified, there is no known way by which any reliability figure can be accurately assigned to a human operator. Additional methods should be used to ascertain that the human operator will be in the design loop during development and validation, as well as providing input for the training program design as early as possible.

The usability trials for a new system or subsystem should/can be used for comparison of the performance with existing systems in order to ascertain that the total safety using an ISC with new functions does not reduce the safety level from current levels.

As described in chapter 6.2 usability trials are recommended as tools both for validation and verification of HMI.

In order to have the HMI approved records of such trials must be added to the approval documents containing references to applied standards and relevant methods.

6.1.4 V&V Techniques

This section describes validation and verification techniques which can be used during system development. Each technique is accompanied by a description of why it should be used, an indication of the SIL levels at which it should be used, and when it may be used in the lifecycle.

For convenience, V&V techniques are divided into static, dynamic and general. Static techniques are those which can be performed without a working system (for example on a specification or model of the system). Dynamic techniques are those requiring a working system. General techniques are those required throughout the lifecycle to provide an “infrastructure” for validation (for example, configuration management).

This section only discusses V&V techniques. However developers must also used development techniques appropriate to the required SIL.

For a bibliography of techniques and measures the reader should refer to IEC 1508 Part 7, NASA, DOD, MOD etc.

6.1.5 Static V&V Techni*q*ues

6.1.5.1 Safety Analysis

Safety analysis entails identifying and documenting the hazards associated with a system throughout its lifecycle, and evaluating the risks associated with those hazards. Risk is defined to be a combination of the consequence of a hazard and its probability.

For well known systems safety analysis may be as simple as relying upon existing understanding of risks as encoded in IMO regulations, classification society rules, international standards and so forth. If existing regulations are used, reliance on them should be justified and documented.

However this approach must not be relied on when new systems (or new combinations of existing systems) are developed. This is because new systems, and combinations of existing systems, may create new hazards, as is already becoming evident from experience with integrated bridge systems. In these cases a formal safety analysis shall be performed and documented.

Note that in the sections below some techniques are described as “bottom up” and others are described as “top down”. Both bottom up and top down techniques should be used to ensure completeness of the safety analysis.

Safety analysis should include the human operator “in the loop” either by analysis and calculation or by appropriate usability trials using a simulator.

6.1.5.1.1 PHA Preliminary Hazard Analysis

Analysis performed at the system level to identify safety-critical areas, to provide an initial assessment of hazards, and to identify requisite hazard controls and follow-on actions. The results from this analysis are to be used as input to the system requirements specification.

Life cycle phase: feasibility (may also be used in later stages, e.g. before a design change)

SIL level: all (it is required to know the risks before determining the SIL)

6.1.5.1.2 HAZOP Hazard and Operability Analysis

A structured analysis performed by a team of experience personnel from representative disciplines aimed at identifying failure modes which may lead to hazards. HAZOP is performed on a suitable model of the process using agreed checklists. Note that special HAZOP variants have been developed for software systems, and these should be used where applicable.

Life cycle phase: specification onwards

SIL level: all (it is required to know the risks before determining the SIL). If HAZOP is not used an equally effective alternative shall be used and documented.

6.1.5.1.3 Fault tree analysis (FTA)

A top-down analysis method aimed at identifying all causes (including combinations of causes) of each hazard.

Life cycle phase: any

SIL level: 3 and 4, but also recommended for 1 and 2

Note: FTA is the only known method to work out SILs for the whole system, even though it need not be used in detail for lower SIL modules later in the project.

6.1.5.1.4 Software FTA

This is similar to normal FTA but applied to detailed software design of code.

Life cycle phase: any

SIL level: 3 and 4, but also recommended for 1 and 2

6.1.5.1.5 ETA Event tree analysis

A bottom-up method to determine the sequence of events that can develop in a system after an initiating event.

Life cycle phase: any

SIL level: 3 and 4, but also recommended for 1 and 2

6.1.5.1.6 Failure mode and criticality analysis (FMECA)

A bottom up analysis to determine the consequences of component failures.

Life cycle phase: any

SIL level: 3 and 4, but also recommended for 1 and 2

6.1.5.2 Design Verification

6.1.5.2.1 Specification review

All specifications (for example system functional specification, software design specification, module specification) shall be formally reviewed to verify that they implement the requirements set by specifications from previous life-cycle phases (e.g. the software design must implement the system specification). Particular attention should be given to the traceability of safety requirements. There exist several helpful approaches to formal walkthrough e.g. Fagan inspection).

Life cycle phase: any

SIL level: all

6.1.5.2.2 Design simulation

An animated model of a proposed design solution which can be used to explore the consequences of design decisions will be satisfied. Examples include: control-room simulation, animated Petri nets etc.

Life cycle phase: design

SIL level: not mandatory

6.1.5.2.3 Formal specification / proof techniques

A range of mathematically based techniques which allow a precise description of a system at any stage in its development, and which enable mathematical proof that implementations satisfy their specifications.

Life cycle phase: any

SIL level: 3 and 4

6.1.5.2.4 Mathematical modeling

Any mathematical technique used to calculate non functional attributes of a design, such as queuing theory, worst-case timing analysis etc.

Life cycle phase: design

SIL level: 3 and 4, but may also be useful for 1 and 2

6.1.5.3 Code Verification

6.1.5.3.1 Coding standards and guidelines

Coding standards shall be available and enforced to ensure:

- unsafe language features are not used
- only standard features are used (to ensure portability)
- a consistent and readable style is achieved
- undue module size and complexity is avoided
- correct use of language features to limit complexity (e.g. information hiding principles)

Coding standards may be enforced by walkthrough, compile time checks and static analysis.

Life cycle phase: implementation

SIL level: all

6.1.5.3.2 Walkthrough

Formal review of code against a documented design specification aimed at verifying the code implements the design. Particular attention should be given to the traceability of safety requirements. There exist several helpful approaches to formal walkthrough (e.g. Fagan inspection).

Life cycle phase: implementation

SIL level: all

6.1.5.3.3 Compile time check

Tool-supported checks of language syntax and static semantics (e.g. static array bound checks). Many compilers incorporate such features. For some languages (such as C) separate analysis tools are available.

Life cycle phase: implementation

SIL level: all

6.1.5.3.4 Static code analysis

Analysis performed on a suitable model of the implementation (often derived directly by automatic translation from the source code) to verify:

- control flow structure
- correct data use (eg absence of read-before-first-write anomalies)

- information flow (eg that input-output dependance relations are as expected)

Life cycle phase: implementation

SIL level: 3 and 4

6.1.5.3.5 Formal proof (of correct implementation)

Mathematical proof that a model of the implementation (usually equivalent to, or automatically derived from, the source code) implements the design, or has other desired properties.

Life cycle phase: implementation

SIL level: recommended at SIL 4

6.1.6 Dynamic V&V Techni*q*ues

6.1.6.1 Structured testing

Testing is to be performed at every identified level in the system hierarchy (for example module/unit test, integration/sub-system test, software system test). Testing at each level should be a demonstration that the system properties specified at that level are present, assuming the lower level entities function correctly.

Life cycle phase: all development phases

SIL level: all

6.1.6.2 Functional testing

Testing to verify that the specified characteristics of the system have been achieved. Input data which adequately characterize normal operation are given to the system. The output from the system are observed and is compared with the specification.

Life cycle phase: implementation

SIL level: all

6.1.6.3 Stress testing

Testing designed to operate the test subject at the limit of or in excess of design loading in order to show that it will stand normal workloads.

Life cycle phase: implementation

SIL level: all

6.1.6.4 Boundary values

The input domain is divided into a number of classes. The tests cover the boundaries and extremes of the classes. The use of null values such as arithmetic zero is often error prone and demands special attention e.g.:

- zero divisor
- blank ASCII
- empty stack or list element
- full matrix
- zero table entry

Life cycle phase: implementation

SIL level: all

6.1.6.5 Random testing

Random tests aimed at directly measuring software reliability. Note that this technique is limited by the difficulty of generating sufficient independent test scenarios, and ensuring that they are representative of the target environment. This technique can be useful for establishing non functional requirements (where there is no need to calculate the expected values of outputs).

Life cycle phase: implementation

SIL level: not required, but may be useful at all SIL levels. Most useful, and also most difficult at higher SILs.

6.1.6.6 Usability trials

Tests performed with end users in a realistic operating environment in order to verify that specific operability requirements have been achieved.

Life cycle phase: all

SIL level: all

6.1.7 General

6.1.7.1 QA-system

Suitable quality assurance system shall be used. This shall include software change management and version control. As a minimum this must ensure traceability and control of all of the documentation required to demonstrate conformance to all sections of this standard.

Life cycle phase: all

SIL level: all

6.1.7.2 Proven in use

This is a method by which field experience in different applications is used to prove that the equipment is working according to its spec. The requirements are: Unchanged specification, 10 systems in different applications, 10^5 operating hours, or at the least 1 year of service life

The documentation should include:

- exact designation of the system and its components including version control for hardware
- users and time of application
- operating hours
- procedures for the selection of the systems and applications procured to the proof
- procedures for the fault detection and fault registration as well as fault removal.

Life cycle phase : specification onward

SIL level 1,2 and 3

6.1.8 Summary of V&V Techni*q*ues

Method	**Life Cycle Phase**	**SIL** 1	2	3	4	**Type of Techni*q*ue**
Preliminary Hazard Analysis	feasibility	X	X	X	X	Safety Analysis
HAZOP Hazard and operability analysis	requirements to system design	X	X	X	X	Safety Analysis
Fault tree analysis (FTA)	late requirements to system design	x	x	X	X	Safety Analysis
Software FTA	late requirements to system design	x	x	X	X	Safety Analysis
ETA Event tree analysis	early system design	x	x	X	X	Safety Analysis
Failure mode and criticality analysis - FMECA	system to detailed design	x	x	X	X	Safety Analysis
Specification review	any	X	X	X	X	Design Verification
Design simulation	design					Design Verification
Formal specification / proof techniques	any			X	X	Design Verification
Mathematical modelling	design	x	x	X	X	Design Verification
Coding standards and guidelines	implementation	X	X	X	X	Code Verification
Walkthrough	implementation	X	X	X	X	Code Verification
Compile time check	implementation	X	X	X	X	Code Verification
Static code analysis	implementation			X	X	Code Verification
Formal proof (of correct implementation)	implementation				X	Code Verification
Structured testing	all development phases	X	X	X	X	Dynamic v&v technique
Functional testing	implementation	X	X	X	X	Dynamic v&v technique
Stress testing	implementation	X	X	X	X	Dynamic v&v technique
Boundary values	implementation	X	X	X	X	Dynamic v&v technique
Random testing	implementation					Dynamic v&v technique
Usability trials	all	X	X	X	X	Dynamic v&v technique
QA System	all	X	X	X	X	Gen. recommended
Proven in Use	specification onwards	X	X	X		Gen. recommended

Legend: x recommended, X highly recommended
Note: some of the detailed techniques are interchangeable eg. testing

6.2 Architecture Layer

Technologies regarding the system architecture can be grouped into three categories:

- **New technologies introduced**: Technologies which are introduced within this document to develop and operate an ISC as specified by the DISC standard.
- **Re*q*uired technologies**: Technologies to be developed to fulfill the requirements established by this document
- **Emerging Technologies**: Overview of emerging technologies which could have an impact on the system architecture of future ISC-systems.

6.2.1 New technologies introduced

The concept of function blocks is not new. It has been applied in many types of standards and proprietary description formats. Examples of the former are HART and Fieldbus function blocks (ISA SP50).

However, there have not to our knowledge been any initiative to capture the resource aspects in the function block paradigm. In addition to ensuring compatibility and interoperability (by specifying data exchanges), we do also propose to introduce specifications of resource use in the model. This will enable the analysis of performance and an assessment of resource use in a complete system. To enable the resource view of the function block paradigm, it is necessary to go via an intermediate level which specifies the resource attributes. Function blocks are mapped to a resource requirement specification which is further enhanced by adding implementation derived resource attributes (network interface type, CPU resources) to it. Again, we do not view this as a new technology - it is perhaps new use of established technology.

6.2.2 Re*q*uired technologies

There are no particular required technology apart from what has been discussed earlier in the report. Some of the more important are the function block idea as partly developed in other contexts and various analysis and compiler technologies as necessary to implement function block tools.

6.2.3 Emerging technologies with a possible impact on the system architecture

Applicable technology will rapidly change and properly lead to new possibilities and functions in the future. The approach described here is technology-independent and allows new technical solutions to be adapted on basis of the same functional description. It is therefore not reasonable to restrict oneself to a defined set of today (or tomorrow) available technology.

There are several developments that can have an impact on how future ISC systems will be implemented on the system architecture level. Some of these are mentioned below:

- **Low orbit satellites for telecommunication**. These satellites may offer low latency and low cost transport links between ship and shore. This will obviously have an impact on what kind of functional distribution one will see between ship and various shore based entities.

- **Transponder technology and VTMIS**. This may have a similar impact as satellites, but on the procedural level. New types of land-based infrastructure will impact the way one think about ship operation and, hence, how ship-shore communication links are used.
- **New operational procedures**. There are some developments of how ships are operated. Integrated transport chains, cargo owner in focus and high speed crafts exemplify this. These changes will define new requirements for ship operation and, hence, ship control. This may also impact the way ship control systems are constructed.
- **Internet and HTML**. The WWW technology may have impact on the way information is transported, but we expect that it will be more important in the area of HMI.
- **Java**. Java is not new technology as such, but it promises to be a safe and standardized way to transport procedural information from node to node. This may be useful for HMI, but it may also offer new opportunities for the construction of control systems.
- **Agent technology** is an extension of the server-client technology where the client can download executable code to servers. The server can further transport agents to other servers. By using, e.g., Java as the procedural language one sees some possibilities in the extension of today ISC systems to new types of functionality.
- **Bandwidth cost**. New technology in networking, e.g., ATM and fiber optics promise to decrease the cost of bandwidth. This can impact ISC systems in that more information can be made available without increasing the cost of communication.
- **Vision technology and multimedia**. Lower bandwidth cost will also increase the usefulness of multimedia use of integrated ship control networks. This can be used to implement new functions, which again may impact the way an ISC is constructed.
- **Formal verification**. There is a slow, but steady development in the area of formal verification of systems. The problem is, as it has been for some time, the complexity of the systems, We expect that methods for formal verification may be easier to apply on highly modular systems and that they may be more appropriate for future systems.

One major problem with many of the new technologies are that they contain increased complexity within themselves and that they also tend to increase the complexity of systems they are applied in. This may be a major problem with respect to system reliability and safety.

7. SPECIFIC FUTURE FUNCTIONS

7.1 The Verification & Validation Layer

The recommended method for V&V as described in this report is a safety based approach rather than a prescriptive approach. In this way the V&V methods to be applied are independent of this nature of future functions and the tools and technologies applied in these new functions.

Therefore no specific future functions are discussed in this chapter.

7.2 User/HMI Layer

7.2.1 Introduction

Ship operation can be described as a hierarchically ordered process of activities; a planning, monitoring and executing level can be distinguished. At the highest level, operations are planned. This is an infrequent activity, which mainly consists of decision making, based upon information of varying reliability from different sources. At the intermediate level, deviations between the plan and the actual progress are monitored in order to anticipate future states. Expected deviations are minimized at the lowest level by making adjustments to setpoints and actuator settings. Because of the present day technological developments, future operator tasks in this process are expected to shift from manual control to decision making and supervision of automatons. As an example, for the navigation task, at the planning level the officer can be assisted by computers for optimizing the route with regard to criteria as time and economy (minimum fuel usage). At the intermediate level, the computer can support monitoring functions with regard to target detection and tracking. At the lowest level, the adaptive autopilot can more or less replace the helmsman. In this section, a number of these present day developments in the different areas of ship operation which are foreseen to have implications for future human tasks are addressed, together with new developments in user-interface technology to possibly support this future role.

7.2.2 Shore based support

The need for expert assistance from a shore based location can arise from several reasons. On the one hand, the qualification and experience of the crew may be at a low level. On the other hand, by introducing complex equipment and systems on board the vessel, additional assistance may be necessary for efficient problem solving.

Shore based support may be divided in two parts:

(i) **shore based support for vessel e*x*ternal matters (as part of the traffic)**

This support comprises shore based services regarding the behaviour of a vessel as a part of the traffic flow, when entering a port. The present situation is described by the presence of VTSs and pilotage services. The VTS is managing the navigable space for all vessels, whilst piloting provides services to an individual vessel, to support navigation in constrained waters such as approaches and in ports.

The future situation in a medium time interval will develop to the provision of information by the VTS to the ship being enhanced by electronic means, such as the provision of accurate traffic images on board, enriched with relevant data on the vessels. Vessels are providing data (such as identity, ETAs, draughts and Dangerous Goods' information) to the VTS using some kind of transponder technology. This information may also be used by the resource managers within a port to optimize their safety operations such as tugs and mooring gangs as well as the intended loading and unloading activities when the vessel is alongside its berth.

The future situation on a long time interval will develop to the provision of information by a traffic management organization restricted to a strategic level, for example by providing each vessel in the area of coverage with a track to be followed and the indication of desired speed profiles. These tracks are determined by simulation before being provided to the vessels. These coordinated tracks and speed profiles

will be used to hook up the track pilot and the engine speed controller. In this situation the VTS functions and the pilotage functions are merged into one traffic management organization.

(ii) **shore based support for vessel internal matters**

Shore based support for vessel internal matters includes diagnosis and repair of failures and support in loading, ballasting and stress calculations when an emergency has arisen.
Manning requirements on ships based on the availability of personnel will determine the quantity of shore based support. For shore-based support related to difficult ship internal operations, such as diagnosis and repair of machinery and equipment, decision support or expert systems may be used, as well as video conferencing techniques for supervising repairs. In this way the required expertise on board is provided through the shipping company, the classification society or even through the supplier companies.

7.2.3 Advanced cargo operation:

The main areas of concern are**:**

Cargo & Cargo Operation Planning: This would involve collection of information (charter requirements, cargo type & specifics, segregation requirements, compatibility with ship's systems and previous cargo history of tanks and systems, cargo terminal restrictions), preparation and simulation of a loading/discharging plan (pump and valve control, strength and stability control, optimization and time consumption). Some owners will do this work onboard, some owners ashore.

Procedures and Training: Owners are to a continuously larger degree expected to provide operation manuals, and to document that they follow such manuals during cargo operations. There may be a potential in providing the manual on the computer, with links to the above planning and the below operation tools. Training of crew will become more and more important, and a simulation tool may provide efficient training.

Loading/Discharging Control & Monitoring: Centralized control of prime movers (electrical, hydraulic or steam), valves and possibly auxiliary systems for cargo and ballast operations. Monitoring of technical system status, ship's strength, segregation, trim, list, air draft etc. If connections to loading/ discharging plan and operation manual, the control may be automatic/ semiautomatic and safety interlocks may be provided. Watch-call may be provided.

7.2.4 Emergency response decision support systems

Emergency response decision support systems are systems that provide a structure in the emergency response to the master of a ship. Such systems will improve the decisions taken by a master in emergencies since many of these decisions are taken in conditions of great uncertainty, in the aftermath of a calamity that has occurred. Information presentation should be adapted to the way in which masters will assess the situation of the ship and to the variables on which decisions are based. The HMI can be developed using user-centred design techniques.

A number of mathematical and prediction models are available to assist in determining the present and future state of the vessel, the progress of fire, etc. These models can help the captain in determining the best options open to him. In order to understand the results of the models in one instance, 3D visualisation techniques may be used.

The navigation of the vessel after an incident (if she is still floating and under command) should also be considered to be a part of the emergency response display in determining the best options to abate the results of a catastrophe.

Taking decisions is also obscured by the non-availability of information regarding the positions of passengers and crew (a personnel tagging system that is armed when an accident has occurred), and the status of the life saving appliances if the vessel needs to be abandoned.

7.2.5 Voyage planning

Voyage planning for future trips between two ports should take into account the recordings of past trips in order to improve the planning of these trips. The information of these recordings should be selected on the basis of well defined criteria and presented in a convincing way as to improve the performance of the vessel. A HMI should be defined that provides the trends and the contribution of the last trip to the trend.

7.2.6 Navigation

A point that is important from the HMI point of view is the integration of traffic and chart information, for instance by overlaying the traffic display on an ECDIS chart. Many observers are adamant that grounding avoidance and collision avoidance, as being the main monitoring tasks of the navigator should be on one and one display alone. This point of view, however, is not generally accepted and a final international opinion on this point is not yet given by IMO.

Collision avoidance using ECDIS type of displays are in development but the HMI has not yet been fully tested, nor that agreement has been reached on the best way to calculate any areas to be avoided when modifying the track of a vessel when the risk of collision exists. Collision avoidance may be facilitated if transponder technology would provide the vessel concerned with ground course and ground speed of the vessels involved.

When navigators continue to use visual inputs during the process of monitoring the track of a vessel, an overlay technique as augmented reality may be of great value in combination with prediction models. Objects and predicted tracks are being projected on the front windows on the basis of the position of the vessel, the position of the navigator in relation to the front window and the direction of his head and a database of the ship and the environment. Some applications of augmented reality are now being investigated.

The HMI of a voyage optimization system, such as determining optimal rev's, optimal settings of a track(auto) pilot, optimal draughts to obtain minimal fuel consumption at a given displacement is complex. Such a HMI should deal with a large amount of inter-linked variables, which are essentially of a stochastic nature. Previous settings should be displayed with their performance indicators, in order to find the best solution.

7.2.7 Advanced doc*k*ing systems

One example is to use a track pilot to bring the ship all the way into the docking position.

The main areas of concern are:

Doc*k*ing Operation Planning: This would involve collection of information for the specific harbour (harbour chart, expected position accuracy, shore based position sensors, tracks, shore base supported tracks, previous experiences, harbour restrictions etc.)

Procedures and Training: The degraded operation mode of an advanced docking system will be a fully manual operation. It is therefore extremely important that the crew are trained to do that (as in aviation systems). A simulation tool will provide efficient training.

Control & Monitoring: An advanced docking system must be initialized and monitored by the crew. The crew must therefore be supplied with information to make the right decision if the operation shall be started, stopped or continued.

In general, the navigation displays must show information such as:

- "Zoomed" ship in ECDIS
- Speed (graphic presentation)
- Rudder and propulsion
- Machinery Information
- Water Jets and thrusters
- Weather/Current/Harbour Information
- Traffic Monitoring
- Night Vision Display
- Video Display
- 3D model of harbour/Augmented reality

For the manual mode (degenerated mode) the following have to be considered :

- Choice of input media (joystick control, advanced joystick control etc.)
- Assistance from other crew members (using video, VHF, etc.)
- Change of operation position

7.2.8 Platform monitoring

- Alarm handling

  Alarms are to be routed to the user being responsible for the different functions. This may be accomplished by alarm grouping, filtering and/or routing and making alarms of different criticality distinguishable.

- Attention alarms/safety actions

  Issues on attention alarms and automatic safety actions are still important topics for human factors research. Regarding attention alarms, for instance, experimental research has shown that under critical conditions people may have the tendency to completely ignore alarms which should have been paid attention to, even if they were provided as clearly discernible auditory signals. Applying automatic safety actions, which has a direct relation to the more general problem of optimal (possibly dynamic) function allocation between humans and machines, should be accompanied by a careful analysis of how to provide the human operator with the required ‘mode awareness’ of the partly automated system, to facilitate human-computer cooperation instead of human-computer competition.

- Condition monitoring (CM)

  CM is normally related to monitoring of equipment, e.g. in the engine room. The concept of CM may also be related to the total condition of the ship and the ship operation, including maneuvring aspects.

- Diagnosis and decision support

  To support the diagnosis process of malfunctioning platform equipment by means of the human-machine interface, experimental human factors research has shown that in the problem-solving process, people tend to solve problems in isolation, ignoring relations between systems (characteristic of the 'cognitive lockup' phenomenon). Additional research on decision support has shown that, in order to overcome this tendency, besides 'off-line' methods like integrated training, 'on-line' decision support according to the 'information aid' concept may further improve performance in the problem solving process. According to this concept, by providing context-sensitive (or 'state dependent') information in a structured way, the operator is assisted in developing a more integrated representation of the problem to be solved, containing elements referring to both the problem structure as well as the evolution of the actual problem in time (problem dynamics).

For efficient use of electronic documentation in general, all types of hyper-links are to be considered, including hypertext, hypermedia, etc., now being commonly used on the Internet. One particularly interesting feature is the Interactive Electronic Technical Manual (IETM). This may include on-line help, maintenance manuals, spare parts catalogue, etc. being context/object sensitive (e.g. getting specific information for a marked object on the VDU). Possible usability bottlenecks to be addressed in designing these systems are (Neerincx et al, 1996):

- The information can be rather disseminated in hypermedia systems. Consequently, adequate navigation around this information can be difficult and users may lose their way or may get disoriented.
- It can be difficult for the user to assess whether relevant information is available.
- The provision of information is often fragmentary, so that users can hardly acquire an overview of the information.
- Hypermedia systems are dynamic with a number of information providers and a number of information seekers. In particular for such systems, maintenance and updating procedures are required. However, such procedures are often lacking, so that the information provision can become more and more unbalanced and chaotic. Subsequently, among other things, users will be uncertain about the actuality and correctness of the information.

7.2.9 Advanced simulation and training

New tools and techniques will provide means to make more realistic and more flexible simulators. With the powerful hardware available at low cost today, a lot of these simulation facilities may be installed on board a vessel. Examples of such equipment may be 3D radar's, 'ECDIS++', giving an equivalent view as outside.

Simulation is already used a lot for general training purposes. By introducing 3D modeling one may be able to simulate close to real situations, e.g. by the use of virtual reality/virtual environment techniques.

This technique includes three elements:

- Autonomy. This is seen as the degree to which different agents in VE (aside from the user) act and respond independently to changes in the environment
- Interaction. The user should be able to have an effect on the objects or conditions in the virtual environment and use sensory feedback to determine the next course of action.
- Presence. This is the sensation of the user being immersed in the virtual world. Presence is induced by different input and output channels in the sensory trackers and displays, and their ability to realistically display the virtual environment.

VR technologies is much more driven by computer graphics applications than the requirements associated with VR.

- Head Mounted Displays (HMD) are often associated with VR/VE. This display provides the user with a view of the VE that responds if the user is turning his head and looks around.
- Sensoring technology includes magnetic, acoustic, optical or mechanical systems. Magnetic and mechanic systems are widely used. Development pace in sensor technology has been moderate.
- Speech recognition and interaction is seen as an important element of VR systems. Systems are being on the market that provide a reasonable degree of speaker independence. The technology development is not very fast and is mainly driven by the telephone market for automated customer services. The interaction technology for speech response and general sound simulation is rather advanced.
- Haptic feedback is the least developed area of VR. Some systems are being developed in universities and a limited number of commercial systems is available. A well known system is a device mounted with sensors on the fingers within a position sensing glove. If activated, they provide the sensation of touching a surface.

Alternatives are being considered, such as the location of physical devices, which are also available in the HMD. This would constitute the best solution for the intermediate term, especially for marine VE/VR applications.

7.2.10 Tele-operations

Tele-operations will play a role when a vessel is abandoned by way of a precautionary action, but the ship still floats. If possible, remote control can be exercised from a control life boat. A second application may be the use of one manned command ship in the centre of a squadron of ships which are remotely controlled by the command ship.

7.3 Applications Layer

7.3.1 Scope

In this section the following application types are described.

- System Manager
- HMI Manager
- Navigation and Voyage Planning
- Cargo management
- Maintenance system
- Engine room, Alarms and Control
- Emergency Management
- Administrative systems
- Information gathering manager
- Black Box system

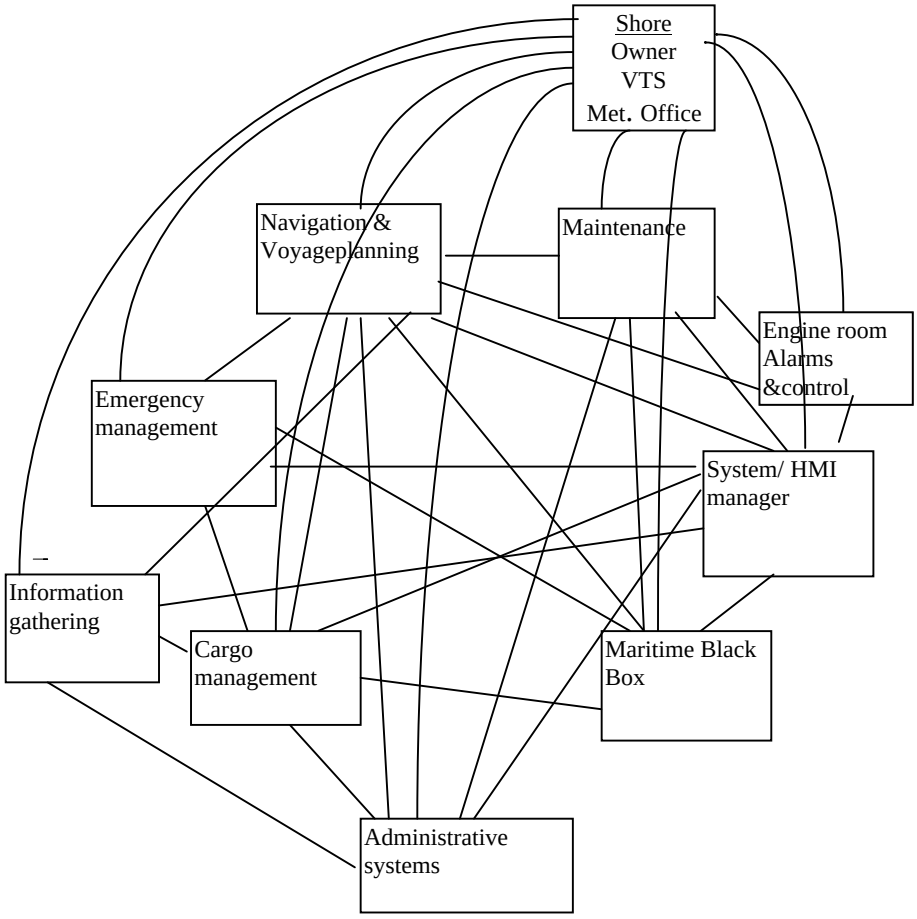

Figure 7-1 - Application Specific Future Functions - Information Flow

Each line in Figure 7-1indicates the need for information exchange between applications in an ISC system. The Information Gathering application will draw on information pertinent to all the other applications (though only a few lines have been drawn in the figure in order to avoid overloading the figure).

The descriptions given in the following illustrate the functionality's which would be expected for State-of - the-art and future integrated applications. As the main innovative aspects of the DISC standard are related to the real integration of applications, special emphasis has been put on integration aspects. Each application type has been decomposed into groups of functions which again comprise a number of function blocks. The function represented by a function block is considered to be the smallest unit which can be provided by an independent supplier and integrated to the open ISC system. One function block may be used by several applications.

For each function block the anticipated **Level of Automation** (**LAU**) (~ level of requested user interaction) has been indicated. Two columns are presented: the first column: **LAU-Today** indicates the most probable level of automation for State-of-the-art applications under due consideration of today's rules and regulations while the second column **LAU-future** indicates the expected level for automation in the future (a prerequisite for the **LAU-future** will in most cases be that the required information exchange is made possible). The levels of automation have been classified according to the following scheme:

4 Fully automatic
3 Automatic, but supervised by human
2 Manual, but support provided by application
1 Fully manual

LAU's applies to groups of function blocks (which in itself can be considered as a high level function block) as well as the low level function block level.

7.3.2 Navigation and voyage planning

7.3.2.1 Description

Navigation and voyage planning application includes

- Long term voyage planning (including voyage optimization)
- Short term voyage planning (including manoeuvre planning)
- Navigation monitoring (sensor information, radar information, transponder feedback)
- Navigation control (auto-pilot, track-pilot)
- Navigation recording.

7.3.2.2 Function blocks

Function block groups	Function Blocks	LAU Today	LAU Fut.	Information flow
Nav. Voyage planning Long-term	• Way Point Planning • Route and Speed Optimization • Hard weather avoidance /Seakeeping • Propulsion / Fuel consumption	2 3 2 3	4 4 4 4	Voyage instructions (charter contract) from ship owner. Routing information from Met. Office Hydro and meteorological info from Met. Office Geographical data from chart company Voyage plan to Short term planning function, Cargo Management System, Maintenance System and Ship Owner
Nav. Voyage Planning Short Term	• Manoeuvre Planner (inclusive Own-ship Manoeuvring characteristics.) • Collision Avoidance • Traffic analyzer • Automatic berthing and mooring. • Simulation and prediction	3 2 3 2	4 4 3 2	Voyage plan from long term planning part Traffic information from VTS centre Transponder input, radar, infrared detection of traffic from Nav. monitoring component. Manoeuvring plan to Nav. Control function. Short term voyage plan to VTS centre
Nav. Monitoring	Processed sensor data like: • GPS • Compass • Under Keel Clearance • Radar	3	4	Input from sensors Traffic information from VTS Radar images from own ship or VTS Output to Black Box Alarms: • Off course alarm

	Obstacle detection: Infrared obstacle detection, transponder ,trafic info	3	4	• WP arrival alarm • Passing distance alarm • Time to Closest Point of Approach alarm
Nav. Control	Trackpilot (Manoeuvring plan executer) incl. e.g. Autopilot and Speed pilot	3	4	Manoeuvring plan from Short Term Voyage Plannning function Control parameters (setpoints/limits) to rudders, engines and thrusters.
Nav. Recording /Replay	Storage Retrieval	4	4	Voyage data from planned and actually travelled route, steering and engine commands, to and from common database, shore based office and Black Boks

7.3.3 Emergency management

7.3.3.1 Description

A future emergency management system providing decision support in case of emergencies related to fire, stability and/or structural integrity is anticipated to comprise the following groups of function blocks: monitoring, decision support, control and recording.

The integrated emergency management system will provide trend analysis, diagnosis and decision support (suggestions to the operator for remedial actions) in case of fire, stability problems (due to e.g. hull damages, cargo shifts, grounding or similar) and structural integrity problems or combinations of these scenarios. The system shall provide decision support in a coordinated and prioritized way (e.g. before recommending use of sprinklers on a large open deck the system shall evaluate the impact on the ships stability, and in case of a fire combined with stability problems the system should ideally be able to prioritize the use of limited resources like e.g. crew and pumps) .

The remedial actions proposed by the decision support system will after acknowledgment by the operator be executed through control systems connected to actuators.

7.3.3.2 Function block*s*.

Function block groups	**Function block*s***	**LAU Today**	**LAU Fut.**	**Information flows**
Em.Man. Monitoring	Fire Monitoring Stability Monitoring Hull integrity Monitoring	3	4	• Fire sensor input: heat, smoke, wind speed and direction, state of fire doors, dampers and sprinklers • Stability sensor input: tank level gauges, draught, heel, trim, state of pumps and valves, leakage/ water ingress. • Hull integrity sensor input: stress, pressure and accelerometers, sea state. • Information from Cargo Management system
Em.Man. Decision Support	• fire and smoke propagation function • stability (incl. damage stability) function • structural function • fatigue function • corrosion function • geometric ship model • evacuation plan	2	3	Sensor input from Emergency Management Monitoring function. Decision support (proposals for remedial actions) to the user

	• lifesaving appliances • maintenance of sufficient power supply and propulsion capability • simulation function • help function • GMDSS			Emergency calls
Em.Man. Control	• Fire Control • Stability Control • Damage Control • Hull integrity Control • Main engine and aux. engine control.	2	4	Control commands to: • Pumps • Valves • Extinguishing systems (sprinklers, gas, powder, water fog systems) • Alert systems • Navigation command systems (auto-and track pilots) • Fire doors and dampers • Ventilation systems • Evacuation guidance systems
Em. Man. Recording		4	4	All sensor signals, decision support provided and control commands executed will be recorded and made available to the Black Box system.

7.3.4 Cargo Management

7.3.4.1 Description

The cargo management system provides decision support in the areas of

- cargo handling
- stability assessment
- cargo surveillance

and has access to a cargo database containing all pertinent information. The cargo management system maintains ship cargo information including history (in order to be able to calculate the current loading condition)

7.3.4.2 Function blocks

Function block groups	Function blocks	LAU Today	LAU Fut.	Information flows
Carg. Man. Planning and Decision Support	Cargo distribution Cargo stowage recommendations on storage conditions and separation rules (IMDG) Special precautions with respect to restrictions on stability, draughts,trim, bending moments, freeboard etc. Cargo hold geometry	2	4	Charter info from owner Voyage plan from Nav. Syst. Cargo information (e.g. container type, weight, bulk type,density etc.) from charterer/owner Loading/unloading plans (pumps, valves) to Carg. Man. Control function Cargo distribution to owner Cargo handling details to port authorities Administrative info to customs Stability info from Emergency Management system
Carg. Man. Monitoring	Loading/unloading operation: • tank level monitoring • ballast operation • inert gas generation • gas detection Cargo surveillance: • temperature monitoring • gas detection • fire detection	3	4	Cargo information from Carg. Man. Planning function Sensor input (tank levels, gas detectors, etc.) Stress levels from emergency management system Alarms to HMI Feedback on loading operations to operator and control syst. Stability information from Emergency Management system
Cargo Man. Control	Semi automatic loading/ unloading function	3	4	Loading/unloading plan from Cargo Man. planning function Feedback on loading operations

				from Cargo Man. monitoring function. Control signals to actuators (pumps, valves,....)

7.3.5 Administrative applications.

7.3.5.1 Description

Administrative functions is functions related to organization, management, commercial, logistics and official reporting.

7.3.5.2 Function blocks.

Function block groups	Function blocks	LAU Today	LAU Fut.	Information flows
Crew management	Manning Travel arrangement Payroll/Allotment Licenses Certificates Crew list Declaration lists Scheduling Vaccination Competence Passport/visa Cabin mngt.	2	3	• Crew change & particulars between manning agent, ship and ship management. • Payroll payments to bank. • Crew/declaration list to port authorities.
Provision	Inventory Cost control	2	3	• Purchase requirements to PO modul
Slopchest/ Bonded store	Inventory Invoicing	3	4	• Purchase requirements to PO modul • Invoice info. to the payroll system. • Crew list from Crew mngt. syst.
Bunker quality adm.	Bunker analyse mngt.	3	4	• Bunker oil analyses results from analysing company. • Tuning parameters to engine control system.
Reporting	Port/VTS reporting Cargo reporting Ship owner reporting Claim reporting Non-conformity reports	2	4	• Temperatures, pressures, vibrations. levels, speed etc from sensors. • Values from cargo mngt. system. • EDI messages to authorities and principals.
Masters Account	Cash flow	2	3	• Provision and slopchest(See above)
Purchase	Purchase Requisition Receival	2	3	• Provision inventory • Maintenance: stock control. • Slopchest inventory

	Prices			• PO and delivery info to/from supplier • Payment transaction to Masters account. • Item receival info. to Provision, slopchest and maintenance system.
Total ship QA mngt. (ISM)	QA manuals Checklists Reporting	2	2	• QA manuals updates ship - shore. • Reports to shore.

7.3.6 Maintenance management system

7.3.6.1 Description

Management of maintenance related activities including

- Planning
- Resource allocation
- Reporting
- Condition monitoring

7.3.6.2 Function blocks.

Function blocks groups	Function blocks	LAU Today	LAU Fut.	Information flows
Maintenance management	Stock control Equipment mngt. Scheduling Job packing. Preventive maint. Condition based maintenance. Opportunity based maintenance. Corrective maint. Personnel allocation Spare requirements Job description Drawing/manual mngt. Supplier info.	2	3	• Maintenance requirements/advice from supplier • Equipment particular from supplier. • Crew lists/competence from the Crew management system. • Routing plan • Docking info. from technical mngt. onshore. • Fault diagnostic from condition monitoring. • PO and delivery info. to/from supplier • Purchase requirements to PO module • Payment transaction to Masters account. • Lead-times/prices from supplier. • Criticality assessment from FMECA.
Condition Monitoring	Trending Diagnostic Advise Severity	2	4	• Temperatures, pressures, vibrations. levels, speed etc. from sensors. • Fault diagnostic and severity to maintenance system. • Alarms
Remote diagnostic, advice and repair.		3	3	

7.3.7 Engine Room , Alarms and Control

7.3.7.1 Description

This application performs the monitoring and control in the engine room(s)

7.3.7.2 Function bloc*k*

Below are listed some of the main functions of a state-of-the-art alarm and control system for engine room operations.

Function groups	**Function bloc*k*s**	**LAU Today**	**LAU Fut.**	**Information flows**
Auxiliary engine	• Start & stop • Preheating • Fuel-limiter • Prelubrication • Heavy fuel/Diesel change over • Start Blocking	3	4	Sensor and control signals
Main engine	• Start & stop • Preheating • Fuel-limiter • Prelubrication • Start Blocking	3	4	Sensor and control signals
Black out start	• Start	4	4	Sensor and control signals
Power Management system	• Diesel Generator Control • Shaft Generator Control • Breaker Control • Heavy Consumer Indicator • •	4	4	Sensor and control signals
Steering gear	• Steering machine	4	4	Sensor and control signals

7.3.8 Information gathering manager

7.3.8.1 Description

One of the key issues in the future will be to provide a tool able of combining the following features:

- Browsing information from World Wide Web servers both internally (Intranet) and public (Internet). E.g. Electronic documentation, Port authorities, classification society, governmental rules and regulation, supplier information.
- Selection , gathering and combining relevant information from the various sources.E.g query tools.
- Personal communication. E.g Email, fax, telex, voice mail, video mail, paging, satellite (Std. A and Std. C)

7.3.8.2 Function bloc*k*

Function bloc*k* groups	**Function bloc*k*s**	**LAU Today**	**LAU Fut.**	**Information flows**
Hyper media	• WWW Intranet/Internet	2	2	To/from all applications. linked to the ISC system and world wide information sources.
Personal communication	• Fax • Email • Telex • Voice mail • Video mail	2	3	
Query tools	• Associative searches for information • Report generation/presentation	2	3	To/from all applications. linked to the ISC system and world wide information sources.

The LAU concept does not seem very applicable for this kind of applications which merely act as HMI for information and communication systems and the indicated levels are of limited use.

7.3.9 Maritime Blac*k* Bo*x* (MBB)

7.3.9.1 Description

The Maritime Black Box records information related to the status of the systems on the ship and information used as a basis for the decision making onboard. The primary concern for Black Box application is the reliability and security of the recorded information, therefore dataattributes such as origin, quality, identification of class are extremely important. The security (or inviolability) of the data has to be ensured using access restriction services provided by the system architecture layer.

7.3.9.2 Function bloc*k*s

Function bloc*k* groups	**Function bloc*k*s**	**LAU Today**	**LAU Fut.**	**Information flows**
Maritime Black Box	• Setup (MBB status, data configuration) • Recording • Emergency handling (release, emergency beacon, communication with shore)	3 *)	4	Input: • System manager (situation assessment) • Systems Status (engine, steering, power) • Cargo status (stability) • Safety Status (fire, pumps, valves, dampers, doors) • Alarm Status (navigation, fire) • Navigation Status(position, speed, heading) • Hull Condition (stress, accelerations, pressures) • Ship Motion (pitch, roll, heel, trim) Output: • MBB status to shore and system manager

*) Automatic release is only performed today in case of sinking

7.3.10 System Manager

7.3.10.1 Description

The system manager solves conflicts in the ISC system in case of resource allocation problems (manpower and/or hardware/equipment resources).

It is (among other things) capable of

- prioritizing tasks requiring user interaction (based on the context based priorities)
- filtering of information to be presented to the user in case of emergencies

Each ISC application must register with the System Manager to signal its presence and readiness.

7.3.10.2 Function blocks

Function block groups	**Function blocks**	**LAU Today**	**LAU Fut.**	**Information flows**
Descriptive/ configuration management	Configuration management List of services/objects Access rights Application priority definition Data definition Resource info.	3	4	Status feedback from all applications
Control	Load control Resource allocation. Application firing, killing and delaying	3	4	Control commands to applications
Monitoring	Integrity monitoring. Situation assessment	3	4	Status input from applications

7.3.11 Human Interface Manager (HMI)

7.3.11.1 Description

HMI Manager: controls and monitors all visual, audible, and tactile devices (VAT-devices).

7.3.11.2 Function bloc*k*

Function Bloc*k* Group	**Function bloc*k***	**LAU Today**	**LAU Fut.**	**Information flows**
HMI Manager	• dynamic configuration of VAT devices • information encoding and decoding • input validation • output device selection and prioritizing • decide priority of information and ranking/sequencing of alarms and other info	4	4	• information packages, tags etc. from other applications • encoding of information by tags or other means • ranked information and alarms to selected VAT output device

7.3.12 Common Database

7.3.12.1 Description

The Common Database is a storage and retrieval medium used for storage of data which are shared by different applications.

7.3.12.2 Function block

Function Block Group	Function blocks	LAU Today	LAU Fut.	Information flows
Common Database	• Database tools	4	4	• Information to and from applications

7.4 Architecture Layer

The contributions from the systems and components represent «enabling technology», i.e., the system architecture will not in itself supply new functions, but it will make new functions possible. This means that most new functions will appear as applications. It may, however, be useful to mention some of the enabling technology represented by the system architecture:

- **Increased integration within the ship** will obviously enable new types of applications that use information from many sources to better determine and control the overall ship and cargo operation.
- **Open integration** will make it easier for third party specialists to develop these new applications.
- **Open ownership of information**. Open integration will also allow the ship operator or owner access to more information in and about the ship. This will, e.g., allow applications that reuse information between ships in a fleet or between several generations of ships.
- **Ship-shore integration**. Increased integration between ship and shore will make it easier to control groups of ships, either as a fleet or regarding the optimization of common resource use.
- **Increased observability** of control systems should make it possible to build safer and more maintainable systems.

The main idea presented in the architecture parts of this report is to develop a new strategy for the description of ISC systems. This strategy, based on function blocks, should make it possible to attain the functional benefits outlined above.

8. TOOLS & TECHNIQUES

8.1 The Verification & Validation Layer

Tools and Techniques are discussed in Chapter 6.1 of this report.

8.2 User/HMI Layer

8.2.1 Re*q*uirements to techni*q*ues

Technological advances like the ones addressed in the area of integrated ship control will interpose new information handling and display devices between the operator and the rest of the system. Those developments may lead to an increase of both the amount of information presented to the operator and the information "density" per area of the human-machine interface. At the same time, progress in "automation" has driven the functions performed by humans increasingly towards monitoring, supervising and decision making. Therefore, it is generally assumed that new advanced systems may place high demands on the cognitive aspects of operator behavior and reduce demands on other human capabilities. Thus, regarding the human-machine interface of such systems, it is essential to maintain an operator-centred design philosophy to overcome limitations, enhance abilities and foster acceptance. In this section, available analysis and design techniques according to this operator-centred approach will be further elaborated. This survey of techniques is largely based on the work of NATO RSG 14 on analysis and design techniques for man-machine systems design (Beevis, 1992).

8.2.2 Human engineering analysis techni*q*ues

In parallel to the essential steps as commonly encountered in systems engineering, human engineering analysis techniques may be ordered according to the following design steps in manned systems design:

- Mission and Scenario Analysis
- Functional Analysis
- Function Allocation Analysis
- Task Analysis
- Performance Prediction
- Interface and Workspace Design

Following this approach, operational needs for the human-machine system are transformed into the specification of the human-machine interface and workspace, following a series of steps involving analysis, synthesis, trade-off studies and simulation and test. First, by analyzing the mission, system functions are determined. The analysis of system functions leads to functional requirements which are the basis for allocating the functions to humans and machines. The detailed function analysis identifies the task performance required from the operator and the required machine processes. Finally, the analysis of the operator tasks and machine processes provide the data for the design of the operator workstation and work

environment. Further, this design process should be iterative, meaning that mission and function analysis, allocation of functions and determination of tasks and interface requirements may be repeated several times.

8.2.2.1 Mission and Scenario Analysis

Techniques for mission and scenario analysis describe the overall requirements of the system under development, in terms which provide information for subsequent human engineering analyses. They are used to describe what the system must do (the operational requirement) and the circumstances and environment in which it must be done.

For high complexity systems, two basic types of analyses were identified by RSG 14:

- Narrative mission descriptions, providing a written or point form of a mission,
- Graphic mission profiles, which provide the mission information in graphic form.

8.2.2.2 Function Analysis

Function analysis is a necessary step in systems engineering, leading to systems synthesis, trade-off studies and a system description. It consists of analyzing the system in terms of the functions which much be performed, rather than in terms of specific sub-systems. Function analysis is hierarchical in nature, and proceeds in a top-down fashion. Each phase in the analysis is the basis for the analysis in the subsequent phases.

The main types of function analyses used in human engineering for the analysis of complex systems are:

Function Flow Diagrams - FFDs

Function flow diagrams identify the sequential relationships of the functions required to perform the mission and operations analysed in the mission and scenario analysis. They are developed at an increasing level of detail, down to the level where specific tasks can be identified for performance by hardware, software or human operators.

Behavior Graphs

Behavior graphs are combined control and information flow graphs for describing system behavior within the Requirements Driven Development (RDD) systems engineering methodology. The graphs show system behavior explicitly as a function of time. The data flow is shown on the horizontal axis and time on the vertical axis. The graphs are used for function analysis at the system level and for scenario modeling.

8.2.2.3 Function Allocation Analysis

Function allocation analysis, assigning functions to people (‘liveware’), hardware or software, provide the basis for subsequent efforts relating to crew or operator task analysis and description, operator performance analysis, display and control selection or design and crew-station design, development and evaluation.

For the design of complex systems, three techniques have been identified having “medium” applicability:

Review of potential operator capabilities

The review of potential operator capabilities documents those abilities of expected system or equipment users which are relevant to the operation of the system. The technique requires information on the expected operator population and potential roles, duties and functions. Further, detailed in formation is required on operator capabilities to perform those functions.

Function allocation evaluation matri*x*

The technique sums weighted scores of human and machine capabilities to make function allocation decisions. The form used to record these comparisons is called a **function allocation screening wor*k*sheet**.

Re*q*uirements Allocation Sheets

Requirements allocation sheets (RAS) are used to translate functions into performance and design requirements. For each function identified, the corresponding RAS describes the purpose of the function, parameters of the design, design constraints and requirements for (human) performance.

8.2.2.4 Tas*k* Analysis

In the context of human engineering, a task is defined as a system function that has been allocated to a human operator. Task analysis, the analysis of these tasks, is one of the most common activities of the human engineering specialist. There are two major goals of task analysis: one is to define what an operator will be required to do, to permit the application of knowledge on human performance; the other goal is to define what an operator will do in order to determine how he or she will interact with the rest of the system. A completed task analysis specifies the activities of the operator.

For the analysis of tasks in high-complexity systems, the following techniques have been identified:

Operational Se*q*uence Diagrams

Operational Sequence Diagrams provide a graphic presentation of the flow of information, decisions and activities in a system, using a set of five basic graphic symbols and an associated grammar. The technique is tailored for the representation of the flow of information, with symbols for the transmission, receipt, processing and use of previously stored information. The diagrams show the sequence of tasks or actions in a vertical sequence: they can be annotated with time information or a time line.

Critical Tas*k* Analysis

This technique analyses “critical” operator tasks in detail according to a specified standard. The standards require these tasks to be decomposed to the sub-task level and subjected to a detailed analysis. The analysis is performed in terms of the information required, perceptual load, decision(s) taken, action taken to implement the decision, feedback provided as a result of the action, communication with others, and any constraints of the interface, workspace and environment.

8.2.2.5 Performance Prediction

Techniques for performance prediction are used to predict or analyze how well operators will perform their assigned tasks once these have been defined by the techniques surveyed in the previous sections. Performance prediction is related to interface and workspace design, since estimates of human performance are dependent on the features of the human-machine interface.

In general, approaches to describing human performance may include **real world observations, field studies, man-in-the-loop-simulator studies, rapid prototyping, laboratory e*x*periments** and pure **computer-simulation studies** including simplified representations of human behavior.

The first two approaches, which are relevant to both new and old systems, have the common drawback that they fail to consider and control an unknown number of variable factors which affect behavior. Obviously, the conditions in which real world observations and field studies are made are very close to actual operations. This is true to a lesser extent for simulator studies. Generally, these approaches (real world observations and field trials) are suited to description and analysis of the mission, incidents and accidents and the operator's activities.

Simulator and laboratory e*x*periments are aimed at prediction of performance for routine and emergency conditions, for instance to test different interface concepts. However, these techniques may have the drawback of doubtful generalization from the artificial test conditions to reality. Therefore, from a methodological point of view, there would be an optimum in simulator experiments, which have sufficient representation of the real world to generalize results for practical conditions and are sufficiently controlled to allow the interpretation of results. This type of experiment offers the opportunity to judge human variance in performance relative to the variance due to the use of alternative pieces of equipment, procedures, etc.

Rapid prototyping involves the use of representations of human-machine interfaces in quasi-realistic scenarios. This technique will be further elaborated in the next section.

8.2.2.6 Interface and Wor*k*space Design

The final goal of the human engineering analyses as described here is to identify design requirements and to facilitate the application of human factors knowledge to the design of systems and equipment.

Two techniques have been reported by RSG 14, applicable to the interface design of complex systems:

Critical design re*q*uirements

The technique identifies design requirements which are critical to the operation of the system, to provide a basis for interface and workspace design. The technique identifies the following information:

- The functions performed by the system
- The operator tasks
- The outputs of each task
- The critical operating variables for each task
- The critical design requirements that affect these variables

The critical operating variables, or the design requirements identified by the analysis, may be weighted to facilitate evaluation of competing design concepts.

Lin*k* Analysis

Link analysis is a technique for evaluating and improving the layout of equipment and the operator-machine interface by minimizing the "costs" associated with transitions between different items of equipment or different components of the interface. It is concerned with the relative positions, frequencies and importance of use of the different components, and how their use can be arranged most effectively. It can be applied to the layout of a specific human-machine interface, or to the layout of a crew compartment for several operators and items of equipment. Using the technique, the links are charted as a to-from matrix, noting the frequency and/or strength or importance of the links.

Rapid Prototyping and User Interface Management Systems (UIMS).

Besides the two previous analytical techniques to derive interface design requirements, in the computer science and human-computer interaction communities there is a growing emphasis on the use of **Rapid Prototyping** and **User Interface Management Systems** (UIMS). These tools permit the rapid creation and modification of the human-machine interface without the need to realize the complete underlying application software or hardware. However, most rapid prototyping tools require support facilities and application specific software to represent mission scenarios and operation dependent aspects of the human-machine interface, such as maps and mission event generators. The prototype interfaces primarily serve to enhance communications and feedback between designers and users. The rationale for rapid prototyping is that system interactions and user requirements cannot be predicted completely. It is argued that it is more effective to produce the equivalent of a **dynamic moc*k*-up** and study how prospective users interact with the system, then modify it and thus develop it iteratively, rather than to analyze all requirements exhaustively. However, in this process care should be taken that evaluations are reduced to little more than judgments of appearance, rather than as an evaluation of functionality. Overall, what appears to be required is a task analytical approach which represents what the operator will be doing with a new system, coupled with a "usability analysis" which indicates how the user expects the system to behave, followed by prototyping and rigorous evaluation. A straight forward example in this field is the (partial) replacement of traditional wooden mock-ups by a virtual environment using 3D modeling techniques. In this way, design and user-based evaluation of, for instance, a particular bridge layout may be performed in a more iterative way, taking maximum advantage of the flexibility offered by these techniques.

HMI Builders

For the actual development of user interfaces (prototypes), several software packages are available. Some of these COTS development tools are platform independent, others are platform dependent. A general classification scheme for these user interface development tools or components is the following:

- UIDT: User Interface Design Tools; Editors for designing user interfaces and user interface components
- UIMS: User Interface Management System; More advanced. Functionality of UIDT, plus interactive testing of code
- VID: Virtual Instrument Designers; Specialized UIDT with special widgets for virtual instruments
- LIB: Library for user interface components; Collection of specialized software components that can be used by programmers.

8.3 Applications Layer

The tools needed with respect to the application layer include

- CASE-tools (Computer Aided Software Engineering)
- fourth-generation languages for declarative queries and data manipulation (e.g. SQL, QBE)
- Java-tools for developing applets on world-wide-web
- PIT for design and realization of discrete event simulation systems
- Database management systems (probably Object-oriented)
- High level programming language compilers

8.4 Architecture Layer

The tools needed with respect to system architecture can be divided into the following groups:

1. System integration tools
2. Debugging, commissioning and test tools
3. Run-time tools

Beside this, a set of general tools is needed which provide a way to set up different models. These models are required during the development phase of specific components and the system integration phase. The tools should enable especially the creation of the following models in a standardized format:

- Functional descriptions (function blocks).
- Resource descriptions.
- Information models.

Two general aspects of systems compiled by components of different manufacturers are the system maintenance and the responsibility in case of system faults. System integrators will properly become responsible for the maintenance of the entire system in the future. Faulty system behavior is often caused by un-predicted interactions between several subsystems. Tools to support maintenance of the overall system and to enable the assessment of failures and faulty subsystems can help to support these tasks.

8.4.1 System Integration Tools

System integration tools support the system designer to set up a integrated system from components developed by different manufacturers (fig. 8.4.1). The specification of the ISC puts requirements on the following design phase. The result of the design phase is a functional description of the entire system given by a set of function blocks delivered by the manufacturers of the ISC components. This function block model will be used as a basis to implement the system. The implementation results in an individual configuration. The result of every phase of this process has to be checked on the input within a closed loop. The entire system integration process should be accompanied by a continuos approval.

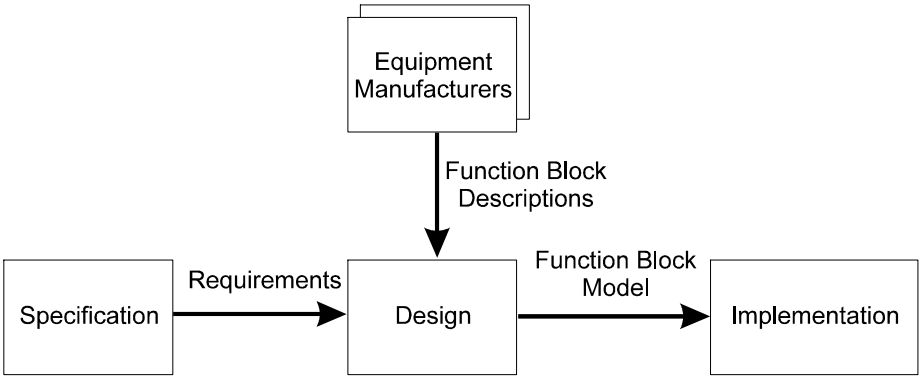

Figure 8-1 - The System Integration Process

Using this approach to describe the process of system integration, several tools necessary within every phase of this process can be identified:

- **Specification tools**: Support the specification of requirements to the ISC design. The resulting requirements should be in machine-readable format.
- **Design tools**: Means to support the design phase of the integration process, including:
  - Composition of different components described by function blocks
  - Checking and tracing of matched and unmatched services
  - Assessment of the use of resources
  - Methods to compare the resulting function block model with the requirements.
- **Implementation tools**: These tools support the compiling of the ISC from the components according to the function block description. Means to check the implementation against the functional description of the ISC, e. g.:
  - match between required and provided resources
  - existence of all services required.
- **Documentation tools**: Tools should be available to support the documentation of the entire integration process. This documentation should be usable for verification and validation

8.4.2 Debugging, Commissioning and Test Tools

During the installation phase special tools are needed for the configuration, debugging and monitoring of the system. Assessment of the proper system under operational conditions should be provided. Means for documentation of individual changes and configuration during the installation process should be available.

8.4.3 Run-time Tools

These tools are used during the life-cycle of the system under operational conditions. The following can be distinguished:

- Maintenance of the system configuration: Tools to maintain, configure and assess the system during operation by the ship officer and service personal.
- Monitoring of the available services and system activities: Assessment of the used system services and resources.

- Documentation of configuration and changes within the system: Continuous documentation during the life-cycle of the system to keep trace on changes and events (faults, shutdowns, errors, etc.) during operation.
- Version control: Tools to control versions of the system components.
- Diagnostic Tools: Integrated fault diagnostic means to enable the ship officer and service personal to detect faulty subsystems. A remote access to the system to allow service and diagnosis from ashore should be available.

9. VALIDATION & VERIFICATION: MINIMUM DEMONSTRATION REQUIREMENTS

9.1 The Verification & Validation Layer

The goal of the chapter is to identify the characteristic aspects of the DISC V&V that must be exercised to demonstrate (minimum) DISC feasibility and compliance.

In chapter 5 the key principles of the DISC V&V were laid down. The approval process and the V&V techniques were described in chapter 6. This chapter will also link these principles and techniques into guidelines for their application in the demonstrator.

There are three possible levels of V&V for a demonstration of DISC. The development of a demonstrator should select the level of V&V appropriate to the project:

1. Minimum Feasibility - All layers of the DISC model act independently to assure the satisfaction of test requirements for a demonstration of the layer. External V&V is not used. In this case: The HMI process will not operate and HMI of applications will not be assessed. The V&V process will not operate and achievement of DISC V&V principles will not be assessed.

2. Feasibility - Each layer of the DISC model defines a set of requirements which are required to demonstrate that it is feasible to achieve a DISC system development. These requirements will include HMI and V&V requirements for applications and architecture and V&V requirements for HMI. These requirements will be derived from the DISC standard. Each team and/or partner will define a development and V&V process to a/met their requirements and b/to demonstrate that these requirements have been worked towards throughout the project. External V&V is used to assess that the methods followed and evidence produced by the teams/projects are sufficient to demonstrate that the feasibility requirements have been met. This V&V need not be complete - appropriate sampling can be used.

3. Conformance - Each layer of the DISC model defines a set of requirements and attributes which have to be proved to be present in the development process and final ISC in order for it to be certified as DISC compliant. An independent, external V&V function assesses whether the system meets these requirements.

9.1.1 ISC development lifecycle

As the V&V process is continuous throughout the entire lifecycle i.e. development and integration of the ISC, it is divided to internal and external parts. The internal V&V process is the producer development and quality assurance. The external assessment on the other hand is carried out for the formal approval by classification societies or authorities and consists of the review and audit of internal V&V documents.

Figure 9-1 - ISC Development Phases - illustrates each phase of an ISC development lifecycle.

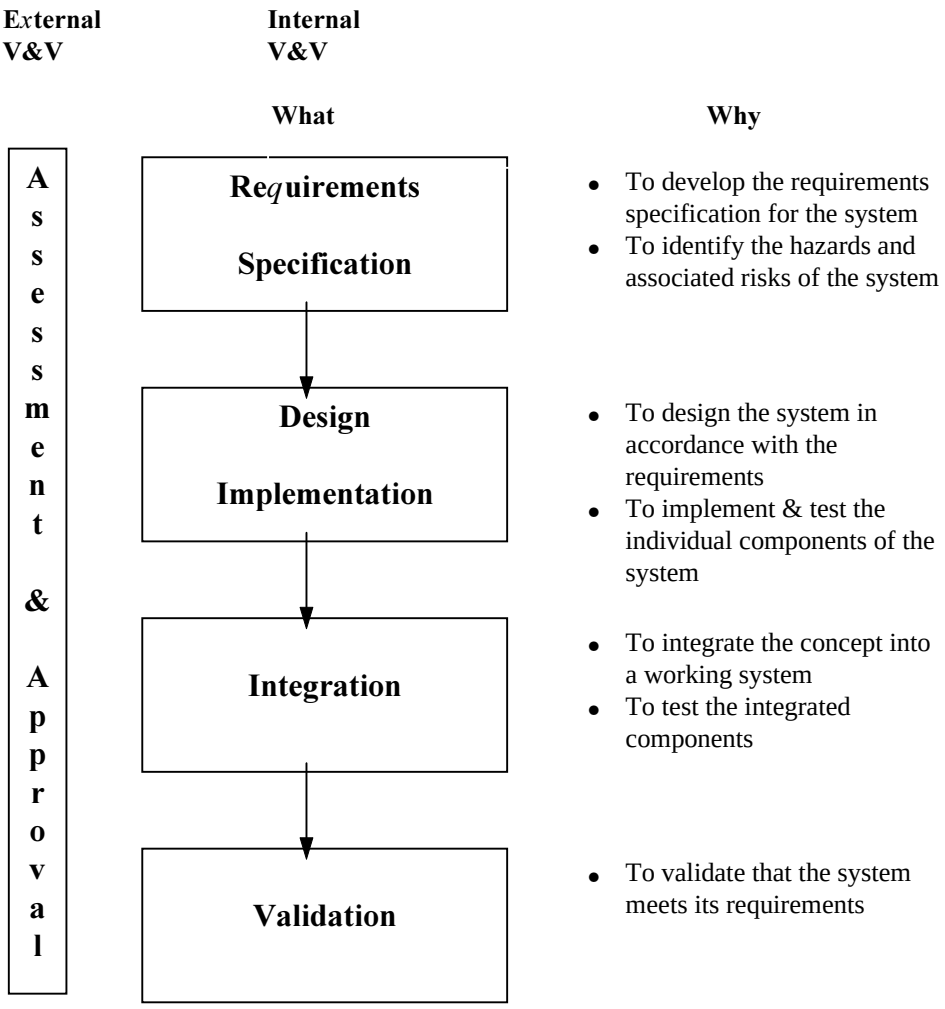

Figure 9-1 - ISC Development Phases

Each of the boxes above includes V&V implementation and there should be feedback and interaction between each of them. These phases can be implemented in variety of lifecycles depending on particular project goals.

Figure 9-2 below gives a more detailed example of what an overall system lifecycle should be. This lifecycle demonstrates the early consideration of overall safety issues. A hazard identification process feeds a quantitative hazard analysis, which defines the safety integrity levels for each component of the total system.

The implementation at this stage is not defined. During steps 5 (Requirements allocation) and 9 (Realization) the functions of the system and their safety requirements are allocated between hardware, software and human operator. The complete system is integrated and validated against the overall safety requirements. The detail of this process and the methods used are particular to the goals and purpose of the system and have to be consciously selected by the project staff. The whole point is to make the project staff to consider safety issues from the start and not just follow a standard process.

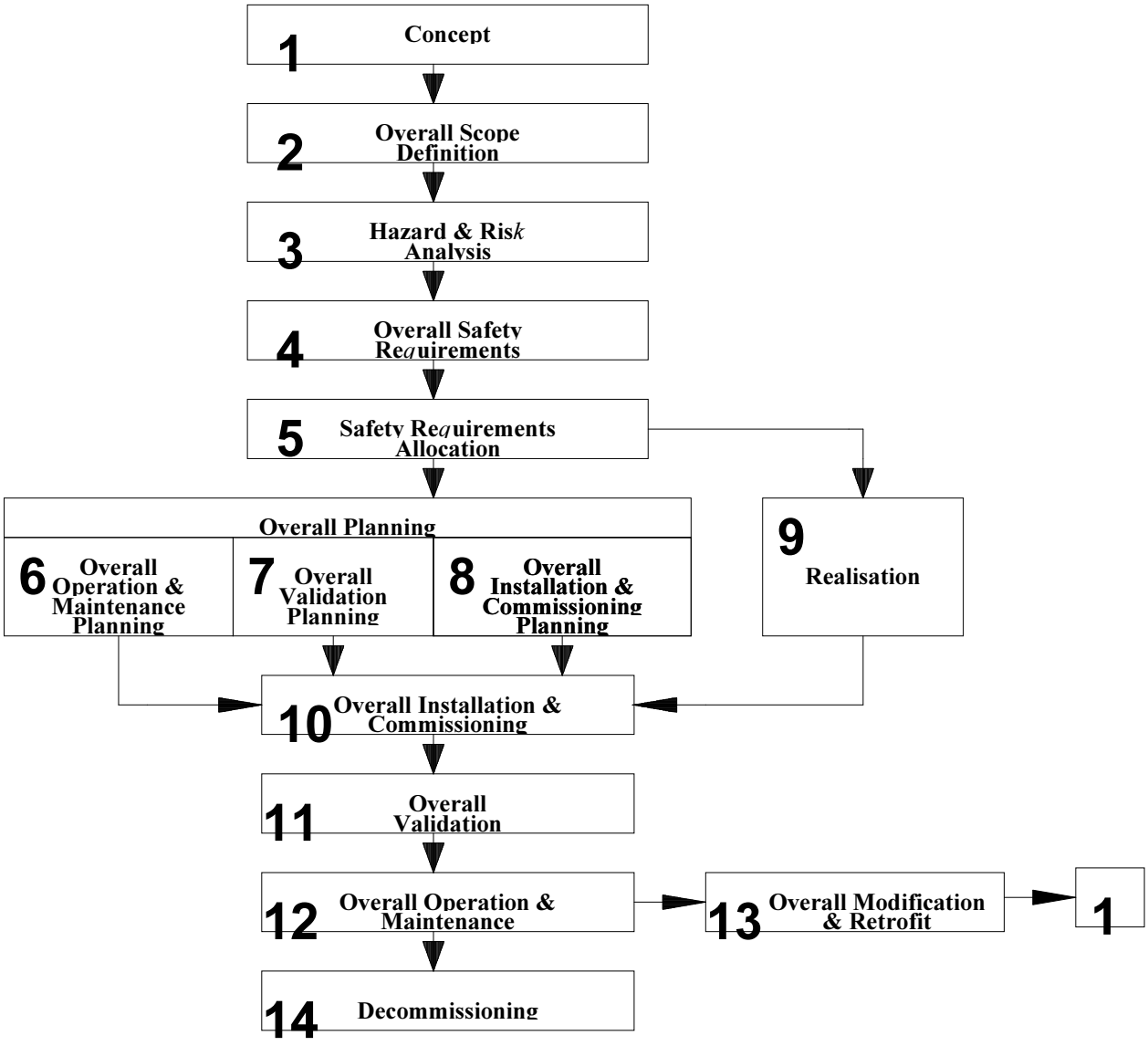

Figure 9-2 - Overall System Lifecycle

9.1.2 Relationship between development lifecycle and V&V principles

As a minimum requirement all V&V principles listed in chapter 5 must be met and objective evidence of this must be provided. The principles are relevant throughout lifecycle and some principles are particularly relevant to certain development activities. These principles are repeated below:

I RISK BASED The deployment of new systems and the integration of existing systems may introduce new hazards and therefore require a risk-based approach, starting with a suitable hazard analysis. The justification for the level of risk shall be based on safety targets set by IMO or best current safety levels commonly accepted in shipping.

II HUMAN CENTERED Human operator must be ”in the loop ”. Evaluation of system safety should include the **total system**, that is both the technical equipment and the operator using it in a realistic range of environmental conditions. The requirements for training operators should be taken into account. **The system shall be designed to give appropriate indications of abnormal operating conditions, and operating procedures for such situations shall be in place.**

III LIFECYCLE Verification and validation are needed throughout the lifecycle of the system, including the entire development process and any modifications. Functional testing alone is not sufficient proof of safety.

IV DOCUMENTATION In order to verify or validate a system or component, traceable documentation is required for all phases of the development process. (i.e. system requirement specification, safety requirement specification, software safety requirements, test methods and results etc.). This requirement implies that sufficient document control and software configuration management are applied.

V INTEGRATION System integration needs to be validated and verified. The verification and validation of the components of a system are in themselves not sufficient evidence of safety. The process of system integration is itself subject to all of the requirements of this standard.

VI SEPARATION It is expected that ISC will result in different systems sharing resources (such as networks and processing hardware). When this happens it is necessary to demonstrate adequate isolation of safety critical components, to ensure that failures in other components cannot contribute to failures in the safety critical components.

VII SERVICE HISTORY It is expected that COTS components (Commercial Off The Shelf) may be used in developing integrated systems. When this is done a sound demonstration of component reliability should be provided, based on in-service reliability data.

The following table shows which principles are applicable to each lifecycle phase. More detailed descriptions of these are presented in chapter 5.

The designers or the project should select the V&V techniques appropriate to the target SIL of each component or sub-system developed. The guidance to determine the SIL of a module is given in section 4.4.2 and section 6.1.1 recommends the techniques appropriate to each particular SIL. The important thing for the external approval is that justification is provided by the developer as to why a particular technique has been selected and how it has been implemented.

Project activity	**Applicable V&V principle**						
	I	II	III	IV	V	VI	VII
Requirement specification	x	x	x	x			
Design - system	x	x	x	x	x	x	
Design - detail		x*	x	x	x	x	x
Implementation			x	x	x	x	x
Integration			x	x	x	x	x
Validation	x	x	x	x	x	x	x

Table 9-1 - Relationship between Development Lifecycle and the V& V Principles
Legend: * = Where applicable

Method	**Life Cycle Phase**	**SIL** 1	2	3	4	**Type of Techni*q*ue**
Preliminary Hazard Analysis	feasibility	X	X	X	X	Safety Analysis
HAZOP Hazard and operability analysis	requirements to system design	X	X	X	X	Safety Analysis
Fault tree analysis (FTA)	late requirements to system design	x	x	X	X	Safety Analysis
Software FTA	late requirements to system design	x	x	X	X	Safety Analysis
ETA Event tree analysis	early system design	x	x	X	X	Safety Analysis
Failure mode and criticality analysis - FMECA	system to detailed design	x	x	X	X	Safety Analysis
Specification review	any	X	X	X	X	Design Verification
Design simulation	design					Design Verification
Formal specification / proof techniques	any			X	X	Design Verification
Mathematical modelling	design	x	x	X	X	Design Verification
Coding standards and guidelines	implementation	X	X	X	X	Code Verification
Walkthrough	implementation	X	X	X	X	Code Verification
Compile time check	implementation	X	X	X	X	Code Verification
Static code analysis	implementation			X	X	Code Verification
Formal proof (of correct implementation)	implementation				X	Code Verification
Structured testing	all development phases	X	X	X	X	Dynamic v&v technique
Functional testing	implementation	X	X	X	X	Dynamic v&v technique
Stress testing	implementation	X	X	X	X	Dynamic v&v technique
Boundary values	implementation	X	X	X	X	Dynamic v&v technique
Random testing	implementation					Dynamic v&v technique
Usability trials	all	X	X	X	X	Dynamic v&v technique
QA System	all	X	X	X	X	Gen. recommended
Proven in Use	specification onwards	X	X	X		Gen. recommended

Table 9-2 - Summary of V&V Techniques
Legend: x recommended, X highly recommended; Note: Some of the detailed techniques are interchangeable eg. testing

9.1.3 Validation planning

One key element to implement successful V&V is that suitable quality assurance system shall be used. This shall include software change management and version control. As a minimum this must ensure traceability and control of all of the documentation required to demonstrate conformance to all sections of this standard. Each development group of the project should define an appropriate V&V plan themselves, because this is the only way to ascertain that they really understand the V&V process. This increases the likelihood of achieving the required level of quality and safety.

The key requirements of a V&V plan are:

1) Schedule for the validation

2) Persons performing the validation
3) Identify the system to be validated
4) Identify the relevant operational modes of the system
5) Technical strategy for validation
6) Specific measures & techniques to confirm that system requirements are met
7) Reference to the applicable requirements specification
8) Environment & equipment for the validation activities
9) Criteria for pass/fail of validation
10) Policy & procedures for evaluating validation results (esp. failures)
11) Cover validation of random & systematic failures
12) Choice of manual/automated techniques, static/dynamic techniques, analytical/statistical

9.1.4 Minimum V&V techni*q*ues for the demonstrator

In a previous section the key principles that must be checked for the demonstrator Validation & Verification are recalled. Each of these principles were connected to a set of techniques. In order to allow the practical V&V of the demonstrator, the following techniques shall be used where appropriate. As a set the techniques in this list covers all the V&V principles, they are well understood and well established and finally they are applicable to all SILs. Each technique should be assessed for its applicability during the development of an ISC. Justification of the selection of the techniques is required.

	Principle:	Technique to be deployed:
•	I. Risk based :	HAZOP Hazard and operability analysis FTA Fault tree analysis
•	II. Human centred :	Usability trials
•	III. Lifecycle :	Specification review Walkthrough
•	IV. Documentation :	Specification review
•	V. Integration :	Specification review Structured testing
•	VI. Separation :	FMECA Failure mode and criticality analysis
•	VII. Service history :	Functional testing Proven in use

9.1.5 Minimum re*q*uirements for e*x*ternal V&V

The aspect of external assessment for V&V purposes must be demonstrated. This is represented by the Approval activity shown in the generic lifecycle in Figure 9-1 - ISC Development Phases. This will both check that the implemented ISC complies with the DISC requirements and will confirm that the V&V approach itself is feasible i.e. that it can be successfully applied to a DISC development.

Approval will directly check compliance with the DISC V&V requirements as it is primarily built on an assessment of the V&V evidence that has been produced. However, because of this, it will also check compliance with the requirements of the other DISC layers and will be, in effect, an overall validation of the DISC approach.

The overall strategy for external assessment is to aim to provide a high level of confidence that each of the V&V principles has been met. This is primarily achieved through well established manual methods that are geared to assessing the evidence produced rather than methods geared to producing the detailed V&V evidence itself. The key issue is of completeness with respect to the principles but a sampling approach can be used with respect to evidence collection and assessment.

The techniques to be deployed will be technical review of documents and records (product checks) and project audit against the development lifecycles in use (process checks). These will be supported by checklists of DISC requirements and procedures for performing structured interviews and test witnessing during audit activities.

Tool support for assessment is not well established. However two tools which should be developed to directly support the assessment activity are:

- a data base of DISC requirements cross referenced to applicable layer and lifecycle phase
- a tool to configuration manage checklists

Although the V&V evidence must be produced for the whole development, it is not a minimum requirement that the whole ISC development should be subject to external assessment. A representative sample of the individual ISC sub-systems or layers will be selected as targets for external assessment with the goal of proving the feasibility of the Approval approach in contributing to the V&V goals.

9.2 User/HMI Layer

In the present section we will indicate how the HMI design process is related to validation and verification activities. The main topic is to specify how evaluation processes are conducted in the various design phases in order to ensure an optimal tuning of system operations to human needs and capabilities. In subsection 9.2.4. the minimal requirements are given for the demonstration of the HMI. As noted in paragraph 8.2.2, human engineering analysis techniques can be ordered according to the following steps in the design process for manned systems:

- Mission and scenario analysis
- Functional analysis
- Function allocation analysis
- Task analysis
- Performance prediction
- Interface and workspace design

The first three sub-processes are an integral part of the general design process, and are to be conducted at the very start of a design process. The function allocation analysis, for example, is closely related to possibilities of automation as specified by the application layer. Specifically, the input for this process is both knowledge from the application level (automation level but also elements of the scenario analysis such as likely system failure) and knowledge of strengths and weaknesses in human thinking. In a narrow sense, the design and evaluation of the HMI refer to the last three sub-processes: task analysis, performance prediction and interface and workspace design. The HMI design process is an iterative process consisting of successive analyzing and specification processes. At the beginning of the design process both the human and machine tasks should be specified (task analysis). Performance predictions are needed to define how the task will be evaluated in specific terms and how the interface could contribute to performance improvement. After these specifications the design of the actual interface and workspace starts. Figure 9-3 shows the design process as a cycle that can be run several times to specify the model, to refine it, to test the design and to adjust or extend it. This process stops when the design proves to be attuned to the knowledge and processing capabilities of the human task performers in the performance analysis.

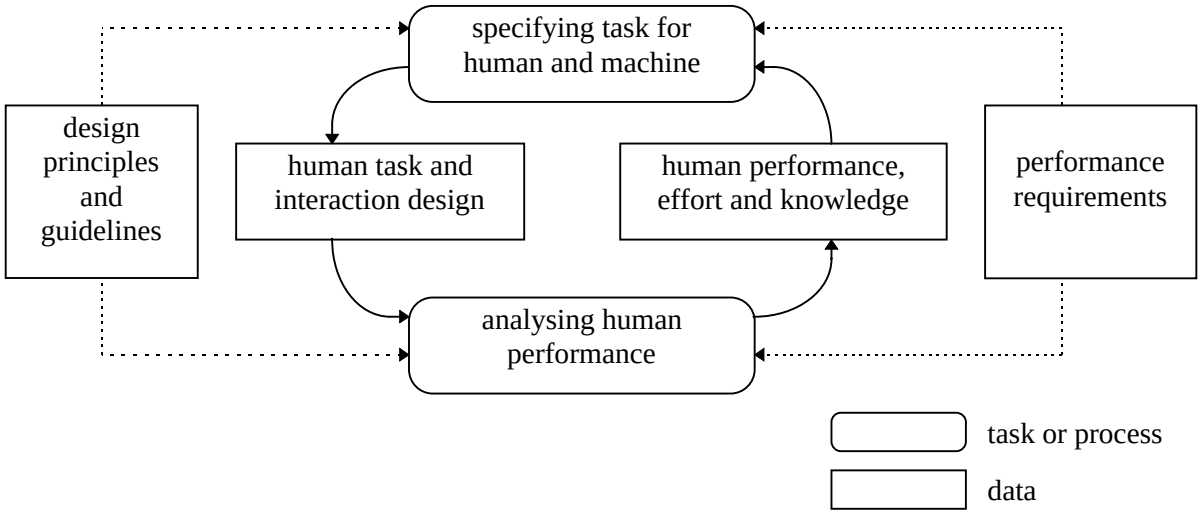

Figure 9-3 - The Design Cycle Aiming at Optimal Human Involvement in HMI Systems (Neerincx, 1997)

Going into more detail on the verification and validation processes for the HMI, it can be stated that evaluation processes occur at various stages of the design process. The present description of these evaluation processes is largely based on the work of NATO RSG 19 in which a framework was provided for cognitive analysis, design and evaluation (Essens, et al., 1994). RSG 19 has identified three stages in the design process: the analysis phase, the design phase and the evaluation phase. During the analysis phase, evaluation activities are required to verify the outcome of various task and cognitive analyses: to what extent is the task analysis sufficient in describing the necessary steps to attain the task goals. In general, this step provides the design principles: at a rather abstract level it is specified which steps are required to perform the task. During design, evaluation is required to verify the design concept and implementation. The question to be answered is whether the chosen design solution is effective and efficient in performing the task. During the final overall evaluation phase, an assessment of the efficacy and usability of the HMI is carried out, along with an indication of the impact of the HMI on the task and organization. The end product of this final stage provides the design guidelines. Based on the assessment of users' behavior in dealing with the actual prototype, design guidelines can be given in order to improve the design. As implied by this statement, evaluations should not only be geared towards performance, but should have some diagnostic value such that inefficient task performance can be related to underlying principles.

9.2.1 Evaluation activities in the analysis phase

Formally, the evaluation activities in the analysis phase comprise a behavioral model, a cognitive model, cognitive requirements and performance measures. These evaluations early in the design process are of crucial importance for directing the precise lay-out of the HMI. Particularly, it provides the background for understanding possible problems that may arise in the operational mode and possibilities for support concepts.

Behavioral model

The behavioral model concerns a description of how the task is to be completed in the light of the task goals. As such, this analysis explicates the analysts' (implicit) assumptions on how the task is accomplished and may give directions to support possibilities. For example, it may be identified that quite a large amount of information needs to be traded-off, suggesting that some kind of support may be needed that reduces computational load and, consequently, probability of error. A behavioral model may be evaluated by expert judgments, by computer simulations or by using experimental techniques. Computer simulations are a rather formal way of assessing whether the behavioral model leads up to the attained goal and may be rather costly. On the other hand, it provides a good insight into potentially neglected steps early in the design process as the whole chain of actions needs to be stated explicitly. Experimental techniques are typically used to assess the robustness of a model. For example, it may be assessed whether the model changes under various environmental (weather, traffic) and task conditions (time pressure, complexity).

Cognitive model

The cognitive model is closely related to the behavioral model but emphasizes the cognitive processes underlying conducting the task. Aspects that are relevant here include workload (amount of information to be processed in some time span), complexity (number of tasks to be monitored at the same time, amount of information to be integrated) and decision making (effort/accuracy trade-offs, satisfying). Obviously, verifying the cognitive model is a rather difficult task, but even informal assessments may reveal potential problems that may arise in the design phase.

Cognitive re*q*uirements

The next step is to verify the accuracy and completeness of the cognitive requirements associated with the task. The criteria for such an evaluation are highly task specific. Criteria that may be relevant are: enhancing the overview of the whole system, performing a sub-task at the right time, decreasing workload, extending memory capacity, predicting or forecasting future states accurately or ensuring a proper understanding. One of the main advantages of doing this evaluation process, is that it guides the way towards efficient decision support. If the cognitive requirements are specified it will be more easy to predict the impact of decision support on actual performance. For example, if it is assessed for a specific task that it is important that the operator maintains an overview of the complete system at any time, it will give a direction to the design of the interface.

Performance criteria

The last set of evaluation activities concern the performance criteria. Of course, these criteria are highly related to the cognitive requirements, but emphasize task outcomes (measurable aspects of task performance). Of importance here is that a list of criteria are given that indicate how the HMI will help the operator in dealing with the task (how can it, for example, be assessed that the HMI provides the operator the required overview of the system) and to determine the efficacy of the HMI. In general, the outcome of this phase will provide the design principles (see left side of Figure 9-3): general principles that provide the ingredients for conducting the task at a rather abstract level.

9.2.2 Evaluation activities in the design phase.

In the design phase evaluation activities are geared towards efficacy and implementation of the design. Evaluation activities involve an assessment of whether the cognitive requirements are indeed met in the proposed design and adequacy. Early evaluations of the design can be done by analytical assessment of costs, effectiveness and expert opinion. Later on, when the design has proven to be reasonably adequate, mock-ups or early prototypes may be tested (see 8.2.2.6).

9.2.3 Integration evaluation activities

In the more formal evaluation phase, when a design has been developed that has a fair chance of meeting the requirements, it is tested whether the HMI provides the possibilities to attain the task goals in terms of operational performance under more or less realistic conditions. From a methodological point of view, there is an optimum in simulator experiments, which have sufficient representation of the real world to generalize results for practical conditions and are sufficiently controlled to allow the interpretation of results. In this phase, the impact of the HMI design on task and possibly organizational performance is assessed, together with the usability of the system. Rather than just assessing the HMI effectiveness, the evaluation should preferably also be diagnostic, that is, if performance is not sufficient, the evaluation should give an indication of its causes. What is also of importance here is to assess any side-effects. The HMI can, for example, have an impact on various roles that people have in a team, or on training requirements.

Scenarios

In the more formal evaluation phase, the evaluations should be as empirical as possible, that is by controlled experimentation. For this experimentation, the development of scenario's forms a critical part. In addition to efficiency, it is essential to assess risk factors that may increase the probability of accidents. The criticality of a scenario used for evaluating the HMI, is defined by both risks due to system failure, the environment

and human thinking. Generally, accidents occur because of a complex interaction between human, machines and environment/context. The general goal of interface design is to optimize this interaction, taking into account the strengths and weaknesses of both humans and machines.

Usability

Issues related to usability and user acceptance all come down to the question of whether users can employ the system in a reasonable manner and relate to human-computer interaction and the ease with which users can be trained. Here it may be observed that the operations that can be performed by the computer are not compatible with the goals the user wants to achieve, or that the information provided does not corresponds with the user's information needs.

Usability can be distinguished at two levels: the task level and the communication level. The task level refers to the goals the user wants to attain, or to the HMI functionality. Can the user carry out the task satisfactorily in interaction with the HMI? At this level, a mapping is made between the user's goal sequences, his or her information needs and the operations and information that can be required from the system.

At the communication level, the dialogue language is described. For the communication level of graphical user interfaces, detailed guidelines and style guides have been established. These guidelines apply to the design phases later in the design process, that is, after the specification of the task level.

User acceptance

In addition to usability, it is essential to assess whether the user population will accept the HMI, which will have impact on their motivation and willingness to use it. Specific criteria include: confidence, ease of use, acceptability, extent of use, ease of training, and documentation. These aspects can be assessed by questionnaires, interviews and observations. In addition, unobtrusive methods to assess the user's ability to work with the system may provide useful data (for example by recording button presses and reaction times). In addition, the relation between the HMI and the user (preference, level of expertise, background) must be assessed.

Documentation

An iterative evaluation strategy may decrease the probability that serious problems are discovered relatively late in the design process. However, even if it would occur, the results of an evaluation can provide crucial data for future endeavors, so that other developers can avoid similar pitfalls. For this reason, it is recommended to document results, lessons learned, techniques and recommendations so that future analysis, design and development efforts can be enhanced.

9.2.4 Minimum re*q*uirements for demonstration

In the text above we have given the requirements for designing the HMI in relation to verification and validation. By using a user-centred design approach, the users are involved at an early stage, increasing possibilities to optimize the HMI to user goals and information needs. An adequate specification of scenarios allows for a critical test of performance improvements related to both efficiency and safety.

For demonstration purposes this approach needs to ensure the following minimal requirements:

- The operator should be assisted in maintaining an accurate mental representation of the process;

- Available information should be adapted to the cognitive processes of the user;
- The operator should be involved in an active way in the problem-solving process;
- Mental workload of the operator should be minimized;
- The operator is provided with feedback and error-correction capabilities.

At the dialogue level, usability and acceptance should be maximized:

- 'Look and feel' should comply with standards and guidelines mentioned in section 5.2.3.;
- Differences in dialogue both within and across various user interfaces should be minimized;
- Computational demands, e.g. amount of information re-coding and memory load, should be minimized.

9.3 Applications Layer

9.3.1 Minimum Re*q*uirements to Demonstrator Applications

The minimum requirements to the applications to be demonstrated are listed in the following:

- The selected applications shall have user interaction.
- At least one application shall be a safety critical (**important**) system.
- The selected applications shall allow demonstration of shifting priorities (due to shifts in the scenario e.g. shift from a planning situation to an emergency situation).
- The selected applications shall demonstrate the improved functionality's achieved by integrating applications through an ISC system.
- The demonstrator shall include real-time as well as non-real-time applications.
- At least one application shall utilize data generated by another application made available through a common database.
- At least two applications shall exchange data information directly through the ALI (not through the common database) transparent of location.
- At least one application shall illustrate the possibility of transferring control (of the application) to a shore based HMI.
- At least one application shall be operable from ashore as well as onboard.
- At least one application shall make use of historical data for e.g. trend analysis.
- All components of a generic application:
  Planning / Decision Support
  Monitoring
  Control
  Training / Simulation / Help
  Recording
  shall be present in the demonstrator (meaning that each of these components shall be present in at least one application in the demonstrator).

9.3.2 Minimum Re*q*uirements to Demonstration Scenarios

The minimum requirements to the scenarios used for demonstrations are listed below:

- The scenarios shall demonstrate the benefits of the DISC-concept
- All major functions in the demonstrator shall be demonstrated
- The scenarios shall demonstrate the benefits of the conflict resolution, prioritizing and resource allocation provided by the System Manager (including HMI manager)
- There shall be an integrated scenario management system (simulator) enabling simulation of sensor signals for all applications present in the demonstrator.

9.3.3 Re*q*uirements to Demonstrator Application Development

The following general requirements to the development have been identified:

- The demonstrators shall to the widest possible extent be based on existing applications.
- All selected applications shall adopt and comply with the HMI standard.
- All applications shall comply with the ALI concept and ALI shall be developed for all applications.
- The applications shall be described using Function Block terminology.

9.4 Architecture Layer

Figure 9-1 outlines the development phases of an ISC. With reference to that figure one can also look at the verification and validation of the system architecture part of DISC as follows:

- **Re*q*uirements and specification**: Verify that the principle of function blocks can be used in the specification of an ISC system and that sufficient resource requirements (regarding safety, timing, resource use etc.) can be captured in the specification.

- **Design and implementation**: Verify that a function block based specification can be realized with a set of particular manufacturer's equipment. This includes a verification that resource requirements can be analyzed in a particular realization.

- **Integration**: Verify that the design process, using function blocks, leads to a physical realization that works as expected. This includes verification that the physical components of the system architecture (networks, gateways, CPUs, VDUs, etc.) satisfy their requirements.

In addition there is also the need to consider verification and validation of the DISC standard during the systems normal (or abnormal) operation:

- **Operation**: Verify that the DISC system architecture layer supplies the specified services during the system's normal operation and also that it continues to function during some common abnormal situations (overload, reconfiguration, equipment down etc.).

9.4.1 Demonstration Re*q*uirements for the Re*q*uirements and Specification Phase

The function block approach is an important tool during the requirements and specification phase. Based on the user's specification of the ISC, the necessary functionality will be realized by selecting and interconnecting a set of appropriate function blocks. Each function block specifies the information types provided and/or requested and the requirements on the information flows (i. e. the requirements for connectivity between the applications). During this phase the function block diagram will also be enhanced with requirements on, e.g., transport delays, processing delays and safety integrity levels.

The following minimum demonstration requirements are identified for this phase:

1. Demonstrate that the individual applications can be described as function blocks.
2. Demonstrate that the system and its requirements can be described in a set of function blocks.

9.4.2 Demonstration Re*q*uirements for the Design and Implementation Phase

This phase consists of two partly independent parts which are of interest to the DISC system architecture verification and validation. The first part is that the individual applications shall be implemented according to their function block specification and, by that, also establish what resources they **supply** to the system (e.g., transport capacity, processing times, spare CPU power etc.). The second part of this phase is that the system is designed by selecting a set of manufacturer specific implementations of the system's function

blocks and verify that the selected implementation satisfies the resource **re*q*uirements** specified in the previous phase. This second part will also include the selection of appropriate network and gateway implementations with corresponding resource views. In the future one will also consider other common resources as, e.g., VDUs and CPUs. It is the opinion of this work group that this aspect shall not be demonstrated since this area is not yet sufficiently mature.

Note that the function block approach as it is described here assumes that one verifies that a set of (off the shelf) applications satisfy specified requirements. Another approach would be to **design** applications to the same requirements. The approach used here is more appropriate for system integration in a setting where the same application components are available from several different manufacturers. The second approach is more appropriate in scenarios where systems and components are custom made.

In the system verification step one has to check whether the requirements on the system (connectivity, safety and resource use) have been fulfilled by the selected set of application components. For a demonstration one must foresee the possibility that the selection and verification steps have to be executed iteratively, e.g., because one has found “loose ends” in the topology (i. e. one function block does not get its required input information) or that there are other inconsistencies in the function blocks selected. This illustrates the need to have approved function block libraries with corresponding conformance classes that can create a manufacturer independent basis for implementation of applications. This is, however, a significant task that cannot be demonstrated at this stage.

The following minimum demonstration requirements are identified for this phase:

1. Demonstrate that the individual application’s resource contributions can be described in an appropriate resource description format (resource view as described in section 4.10.2).
2. Demonstrate that the system requirements regarding connectivity can be verified by resolving the actual connections given the selected network(s). One must in particular verify that connectivity across gateways can be resolved by having at least some connections going across one or more gateway.
3. Demonstrate that the system’s resource requirements can be verified based on the applications’ and common resources’ resource views.

9.4.3 Demonstration Re*q*uirements for the Integration Phase

The integration phase will use the actual physical application and general components to create a working system. This may also require some configuration of individual pieces of equipment to make them fit into the selected physical realization. The DISC system architecture should make it possible to do this individual configuration so that the following integration proceeds with minimal problems.

The following minimum demonstration requirements are identified for this phase:

1. Demonstrate that the individual applications can be completely configured before actual integration is performed.
2. Demonstrate that the functionality of the system is as specified by performing appropriate functionality tests. This shall verify the existence of all functionality specified, availability of all information required and that there are sufficient physical resources.
3. Demonstrate interoperability, i. e. the use of two different applications in one ISC which have an identical function block description. It should be demonstrated that these applications can be replaced arbitrarily without unnecessary affecting the overall performance of the ISC system.

4. Demonstrate that the resource use specified by the system's resource view is as predicted. This can be done for network resources by measuring the actual load on one or more of the system's networks.

Note that the system architecture layer do not specify verification tests for parts of the system that are considered applications. This applies to, e.g., system manager and common data-bases. Currently there are neither any tests for system level services based on the basic ALI services. This applies to, e.g., the mater arbitration protocol. Tests of these functionality's should, if required, be included in the testing of the system's overall functionality.

9.4.4 Demonstration Re*q*uirements for the System in Operation

The verification and validation of the DISC system architecture during the operations phase consists of showing that all system layer services work as expected in normal and some abnormal situations. This is to large degree a verification of the required functionality of the application layer interface (ALI). The following minimum demonstration requirements are identified for this phase:

1. Verify that the ALI can supply any type of information with any specified additional attributes, such as quality or generation time. The ALI shall also be able to supply information about the current configuration of the system, both with regard to function blocks and information elements that are available.
2. Verify automatic configuration by adding a new application to the system. The addition of an application with only inputs from other applications should be possible without any reconfiguration of the system.
3. Verify detection of lost application by removing one existing application. All affected applications should be notified. On the application layer one should also verify that the applications take appropriate corrective measures.
4. Verify that the reintroduction of the same (or compatible) application makes the system commence normal operation.
5. Verify the resource use restriction mechanisms of the ALI by increasing system or component load above the threshold of the ALI. One should observe a discriminating reduction in supplied resources. This is most easily done by examining information transport resources.

9.4.5 Minimum demonstration re*q*uirements for validation and verification

The overall strategy is to validate the system architecture with a mixture of manual and automated techniques with the goal of confirming the validity of an architecture implemented by a function block specification.
The following V&V techniques are expected to be used:

- FMECA should be applied on the function block description of the system to verify its applicability on this type of system description. Note that a hazard identification of some form is needed as a context for this technique.
- Structured testing shall be applied on all implementations of function blocks to verify the functionality and the resource view specification of the module.
- Specification review to verify function block descriptions.
- Mathematical modeling to analyze and verify resource usage during design.
- Functional testing to verify the overall functionality actually delivered by a function block.
- Stress testing to confirm network behavior is maintained under high load or saturation conditions.

10. WORK REQUIRED PRIOR TO DEMONSTRATION

10.1 The Verification & Validation Layer

The DISC V&V principle contain certain aspects that are novel, or are not well established, in the marine context. Those aspects that require further development are identified and described below. Specific tasks to achieve this are defined and these tasks will need to be implemented before demonstration is achievable.

It should be noted that, unlike some novel aspects of the other DISC layers, the underlying principles of the V&V layer are reasonably well established in other application sectors. Hence the main requirements in terms of V&V work required to support demonstration are built around explaining or extending principles and techniques. This is particularly focused at quantifying issues where possible and integrating the V&V approach with the development processes adopted to implement the other DISC layers. A suitable maritime V&V scheme, based on best practice in other sectors of industry, which operationalizes the DISC principles will be selected.

10.1.1 Safety lifecycle and ris*k* assessment

Relevant principle: I Risk based, III Lifecycle, VI Separation

Issue: Formal risk assessment within a defined safety lifecycle is not typically part of marine systems development. The approach & techniques for hazard identification & analysis and for risk classification may not be well understood by vendors. In addition the concept of SIL's is relatively new and there may be little practical experience of their application.

Work required:

- Identify relevant safety standards & publications that are particularly suitable as training material, especially on the topics of HAZOP and FTA.
- Perform top level PHA for a model ISC.
- Define some safety targets based on marine experience.
- Develop training materials specifically on the concept of a Safety Lifecycle and its implications for the V&V responsibilities for affected organizations.
- Develop practical guidelines for both identifying required SILs for systems/components and for demonstrating the SIL actually achieved.
- Prepare checklists for V&V use addressing the key DISC requirements relating to both the ISC product characteristics and the development process.

10.1.2 Usability trials

Relevant principle: I Risk based, II Human centered, VII Service history

Issue: In the human centered design of an ISC system there is a need for "keeping the human operator in the loop" throughout the whole design lifecycle. The usability of the total ISC or some of its applications should

be tested for the HMI aspect on the basis of how the planned mission can be fulfilled using them. There are several families or classes of agreed/well documented methods for collecting and validating user requirements. Usability is one of the measurable attributes of a product/system, which validates the operation of the total system.

Work required:

- Investigate mapping between human centered techniques and the level of assurance of meeting user needs from each technique in the marine environment.
- Define a mapping between SIL and degree of usability.
- Define a framework for verification that an appropriate method has been selected and performed to give the required degree of assurance of the required level of usability.

10.1.3 Product Service History

Relevant principle: VII Service history

Issue: The benefits of re-usable components in developing a robust ISC have been emphasized. Where such COTS products are used there is typically insufficient existing lifecycle evidence to demonstrate conformance with the V&V principles. Although an argument of integrity achieved based on in service experience is directly admitted by V&V principle VII the implications of this, in terms of the range and quality of service history evidence that is needed may not be fully appreciated.

Work required:

- Elaborate the V&V technique ‘Proven in use’ into more detailed requirements for the in-service evidence that is needed (taking account of different SIL’s).
- Establish target levels for the types of service history evidence e.g. for the number of identical components deployed; for the total number of operational hours.
- Identify tools that are suitable for overall configuration management of the ISC.
- Establish a database of proven components that are proposed for use and ensure it can be populated with references to the service history evidence.
- Establish the role and extent of use of conformance class certification of standard function blocks.

10.2 User/HMI Layer

The HMI work prior to demonstration will be a process as described in section 9 in the steps as presented below.

All user interfaces operated by the same user have to comply with the process outlined in section 9.2.4. The task analysis will be based on input from existing applications, input from users of these applications, input from classification societies and input from other competent bodies. The redesign process of the existing applications, using a user centred-approach, is subsequently done by a HMI committee in co-operation with the application suppliers. An iterative process starts with modification of the applications, and production of new applications as required, according to the specifications laid out in the first phase of design process. Assessment of performance measures is done, which may result in further modifications and validation of the applications. The first part of the user design process will focus on the task level, then guidelines for the 'look and feel' level will be laid out.

The task analysis and initial prototyping will indicate whether and to what extent the HMI of different applications will need to be redesigned. Similarly, the style guide for the 'look and feel' will only be available, after a detailed analysis of the functionality of the applications has been carried out.

The list below summarizes tasks to be done prior to demonstration:

1. Selection of a set of applications that are relevant for demonstrating the functionality of the HMI. The functionality of each application should be specified.
2. One or several tasks need to be selected for which an analysis needs to be made, specifying goals and information needs. Existing function and function allocation analyses can be used to select the most critical tasks. In addition, information may be gathered from existing applications, from users of these applications, from classification societies and inputs from other competent bodies. In selecting tasks, the focus should be on the integration aspect.
3. Guidelines needs to be developed for the communication level (look and feel) which should comply with relevant international standards (see 5.2.4). These guidelines are related to:

    - Workspace design and arrangement
    - User input device and display unit design
    - Style guide
    - Design of workplace
    - Work environment

4. Scenarios need to be specified both for testing (critical conditions) and demonstration.
5. Development of an experimental environment allowing for testing HMI on pre-specified performance criteria.

10.3 Applications Layer

The following activities are considered mandatory in order to prepare for a successful demonstration:

- Development of DISC interface specification standards
- Provision of tools for implementation of ALI
- Provision of tools for testing of DISC ALI interface compliance
- Adoption of DISC HMI and ALI standards for development of each application.
- Specification and validation of demonstration scenarios (including specification of scenario management (simulator) requirements)
- Provision of demonstration environment (preparation of Mock-Up), including simulator Hook-Up.
- Integration testing in demonstration environment.
- A System Manager shall be developed
- Documentation of demonstrator (reports, papers, brochures etc.) and DISC skeleton standard.

10.4 Architecture Layer

The following categories can be distinguished with respect to the work required prior to demonstration:

- General means for information transport.
  This group contains the physical equipment facilitating the communication between different applications including network protocols, the network itself and gateways.
- Specification, design and implementation of the Application Layer Interface (ALI).
  The ALI is the core component connecting each application to the system in a standardised way. The functionality of this component has been explained in section 4.8.1. Based on this specification, the ALI has to be implemented and connected to the DISC demonstrator applications.
- Establishment of the functional specification framework.
  A standardised language for the functional specification outlined in section 4.8.6 has to be generated. Another important task to be fulfilled is to develop guidelines for the use of function blocks, resource views and information models.
- Implementation of applications using the framework.
  Development and implementation of applications compliant to the DISC standard.
- Specification and development of means to support the verification and validation process with respect to the system architecture.

In the following sections, the work required to put into practice the subjects outlined above will be explained in detail.

10.4.1 General Means for Information Transport

One or more network protocols which comply with the DISC standard have to be specified. In particular, these protocols have to fulfil the requirements given by the Application Layer Interface (see 10.4.2).

At least two different protocols are recommended - one for the real-time network and another one for the administrative net. In addition, necessary gateways have to be specified and implemented.

10.4.2 Implementation of the Application Layer Interface

It has to be ensured that the ALI gives access to the services specified in the DISC standard (see 4.11.4). To verify that the specification of the ALI according to the standard is technology independent, these component should be implemented on different platforms. Special attention should be given to the control functions of the ALI, e. g. limiting the access to given resources or prioritising of applications. It should be pointed out that the Application Layer Interface and the Functional Specification Framework represent the key concepts of the system architecture of DISC. Both have a high potential for standardisation. It should therefore be put careful attention to the realisation of this concepts.

10.4.3 Establishment of the Functional Specification Framework

A standardised language for functional specifications according to the methodology outlined in section 4.8.6 has to be generated or adapted. This language has to comply to the requirements given in section 4.11.2.

Guidelines explaining the use of the framework should be developed. These guidelines affect the essential components of the framework - function blocks, information models and resource view - and should clarify the following points:

- Definition
- Field of application
- Working methodology
- Verification and Validation procedures
- Examples

Every component integrated into the demonstrator has to be equipped with a resource view which can be used to determine the resources allocated by the component concerned. The following resources shall be captured by the resource view:

- network resources
- CPU resources
- HMI resources
- Safety integrity levels (SIL).

It should be noted, that compared to other types of resources, network resources can be verified in the easiest way.

10.4.4 Implementation of Applications

The demonstrator applications have to be specified, designed and implemented compliant with the framework given in the DISC standard (see sections 4.8 and 4.10). Particularly, every application has to have a function block specification and a resource view.

To facilitate the demonstration of the process of system development and integration, a library of function blocks containing the functional description of all applications which shall be integrated has to be established.

The exchange of information between applications should be based on one or more information models. These models have to be prepared before implementing the demonstrator.

The function block library and the information model has to be implemented by the vendor of the specific application.

10.4.5 Tools to Support the Process of Verification and Validation

The following tools have to be provided to support the process of verification and validation of the system architecture:

- network load measurement and manipulation
- tool to monitor the behaviour of applications, the communication via the network and the actual physical resources.

11. STANDARDS REFERENCES

[1] IEC 1508: Functional Safety
[2] IEC 1209: Integrated Bridge Systems
[3] ISO 9001: Model for Quality Assurance in Design, Development, Production, Installation and Servicing.
[4] ISO 9000-3: Quality Management and Quality Assurance Standards: Guidelines for the Application of ISO 9001 to the Development, Supply and Maintenance of Software
[5] PRO96 Chapter C Assessment Procedure in Generalized Assessment Method (GAM) Rules CASCADE Deliverable CAS/LR/WP2.T3/SM/D2.3.1/0.C, Sept. 1996
[6] PRI96 Chapter D Assessment Principles in Generalized Assessment Method (GAM) Rules CASCADE Deliverable CAS/LR/WP2.T3/SM/D2.3.1/0.C, Sept. 1996
[7] ISO 13407, Human centered design processes for interactive systems
[8] ISO 11064, Ergonomic design of control centers
[9] ISO 9241, Ergonomical requirements for office work with visual display terminals (VDTs)
[10] ISO 8468, Ship's bridge layout and associated equipment - Requirements and guidelines
[11] IEC 447, Standard directions of movements for actuators which control the operation of electrical apparatus
[12] EN 894, Safety of machinery - Ergonomic requirements for the design of display and control actuators
[13] Framework for an ergonomically assessment of the human-machine interface of ECDIS equipment (TNO, December 1995)
[14] Framework for an technical assessment of the human-machine interface of ECDIS equipment (TNO, December 1995)
[15] ISO 10303: Standard for the Exchange of Product model data, notably parts on EXPRESS (data definition (and validation) language) and Data Access Interface
[16] IEC 1162: Maritime navigation and radiocommunication equipment and systems - Digital interfaces
[17] ISO 11065: Glossary for Automation
[18] ISO 12178: Real Time Communication Architecture
[19] IEC 92-504: Electrical Installation Control and Instrumentation on Ships
[20] EN 50170: European Fieldbus: Profibus, FIP, P-Net
[21] ISO 9506 Part 1+2: MMS (Companion Standard)
[22] IEC 870-5: various telecommunication equipment