Project contract number: **TIP5-CT-2006-031406**

FLAGSHIP

European Framework for Safe, Efficient and Environmentally-friendly Ship Operations

Instrument type: IP

Specific programme: **Sustainable Surface Transport**

D-D 1.1 TCI and status indicator specification

Start date of project: 2007-01-01
Duration of project: 48 months

Due date: 2008-07-01
Actual delivery date: 2008-09-26

Lead contractor: MARINTEK

Revision: Draft 0.12

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	**Public**
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Document summary information

Authors and contributors

Initials	Author	Organisation	Role
ØJR	Ørnulf Jan Rødseth	MARINTEK	Editor
FD	Francois DREZET	AYFr	Contributor
ESP	Erik Styhr Pedersen	LMA	Contributor
NAB	Nils Albert Jensen	KM	Contributor
KCE	Karl-Christian Ehrke	SAM	Contributor
PNO	Per Norman Oma	AFS	Contributor
RG	Reinert Giere	MW	Contributor

Revision history

Rev.	Who	Date	Comment
0.1	ØJR	2007-04-12	First suggested table of contents
0.2	ØJR	2007-11-20	Suggested new name TSI, added on scope
0.3	All	2007-11-20	Additional data from workshop in Hamburg
0.4	FD/ØJR	2008-02-01	Consolidation with input from AYSA
0.5	ØJR	2008-02-16	New executive summary, new TOC
0.6	ØJR	2008-04-17	Adjustments to better cater for other SP requirements
0.7	FD	2008-05-12	Comments from AYSA
0.8	ØJR	2008-05-14	Final outline after CPH meeting
0.9	All	2008-06-24	Integrated contributions from FD,JKG.
0.10	ØJR	2008-07-11	Additional stuff on TCI and TSI – data models and final draft
0.11	ØJR	2008-09-09	Additional corrections, added system breakdown and Shipdex ref.
0.12	ØJR	2008-09-15	Added conclusion, after CDG meeting

Quality Control

	Who	Date
Checked by lead partner	Ørnulf Jan Rødseth	MAR
Checked by SP	Francois Drezet	AYFr
Checked by internal reviewer	Bernard Dognaux	EMEC

Company internal coding (if any)

Main responsible	Internal reference number

Disclaimer

The content of the publication herein is the sole responsibility of the publishers and it does not necessarily represent the views expressed by the European Commission or its services.

While the information contained in the documents is believed to be accurate, the authors(s) or any other participant in the FLAGSHIP consortium make no warranty of any kind with regard to this material including, but not limited to the implied warranties of merchantability and fitness for a particular purpose.

Neither the FLAGSHIP Consortium nor any of its members, their 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 FLAGSHIP Consortium nor any of its members, their officers, employees or agents shall be liable for any direct or indirect or consequential loss or damage caused by or arising from any information advice or inaccuracy or omission herein.

Executive summary

Based on the concept of Technical Condition Index (TCI), this report develops the definition further and adds the new concept Technical Status Index (TSI). While the TCI says something about the technical condition of a component (100 – as new, 0 – completely destroyed), the TSI says something about the components ability to function as expected (100 – fully operational, 0 – not able to perform any function).

The TSI can be used to show the operator the system's ability to perform its function, e.g., after a series of alerts that may not in themselves give sufficient information about operational capabilities. Thus, the TSI may have applications in alert filtering as well as in general status overview functions.

It is shown in this report that both TCI and TSI may be calculated by using a tree-structured data model where index values on higher level are calculated from them on lower level. Further more, as the tree (Technical Function Block – TFB tree) is mostly the same for both TCI and TSI, there are synergies in integrating these specification structures.

Further more is it shown that the tree to a large degree can be derived from normal safety analysis typically done on critical ship systems. It is also shown how trees from different manufacturers can be integrated to larger more system wide trees.

Examples are given as to how this concept can be realized in a real system. This includes formal specifications of the data structures in UML (Unified Modelling language) and concrete data file formats in XSD (XML Schema description).

The report also contains a description of various data related to the technical condition can be transmitted between ship systems. These data is generally in XML format, but also IEC 61162-1 formats are described.

Annexes give additional background on how to extend the concept to be used in more distributed applications and a brief overview of relevant integration protocols in use on ships today.

List of abbreviations

AIS – Automatic identification system

ARPA – Automatic Radar Plotting Aid ("advanced" radar screen)

CAM – Central Alert management (part of IBS)

CCTV – Closed Circuit Television

CDATA – XML element meaning "character data", encloses "free format" text.

CSI – Control and monitoring Status Indicator

DNV – Det Norke Veritas

DSS – Decision Support System

EIAMUG – European Intelligent Actuation and Measurement User Group

EA – Enterprise Architect (Software design tool from Sparx systems)

ECDIS – Electronic Chart Display and Information System

EMS – Emergency Management System

FMEA – Failure Mode and Effects Analysis

FPGA – Field programmable gate array

GPS – Global positioning system

HVAC – Heat, Ventilation and Air Conditioning

IBS – Integrated Bridge System

ICT – Information and Communication Technology

IPR – Intellectual Property Rights

ISC – Integrated Ship Control

KPI – Key Performance Indicator

LWE – Light Weight Ethernet

MAS – Main Automation System

MRCC – Maritime Rescue Coordination Centre

OPC – Open Process Control, a MicroSoft © developed protocol for process data exchange

PI – Performance Indicator

PMS - Planned Maintenance System.

RCU – Remote Control Unit

S1000D – Equipment specification and documentation standard from aircraft industry (see 3.9).

Shipdex – Shipping related application of S1000D.

SFI – Skips Forsknings Instituttet (as in SFI Group System)

SPI – Ship Performance Index

TCI – Technical Condition Indicator

TFB – Technical Function Block

TOCC – Technical Operations Competence Centre

TSI – Technical Status Indicator (see 2.12).

UML –Unified Modelling Language

VTS – Vessel Traffic Services (“ship traffic monitoring”)

XML – Extensible Markup Language

XSD – XML Schema

Table of contents

1. Introduction

1.1 Vision Statement

The new TCI and technical status indicator (TSI) concept shall ensure that technical information from various ship sensors and subsystems and from different makers (pumps, valves, filters, motors, heaters, etc.) is available for the different technical management systems. The technical information may comprise i.e. fault reports, alarms, and status information. In a typical example a decision support system receiving an alarm from a safety critical pump shall be able to check the status of the back-up systems and group the alarm accordingly into a lower or a higher priority thus releasing the operator from further own investigation of the actual system status.

1.2 Identification of Users

The information provided by the TSI concept will be used primarily by the officer on watch. The concept is especially supporting the one-man-bridge operation because in a typical passage scenario the navigator will not be able to analyse any problems in the technical systems in parallel to his nautical tasks. Further on it will be used by other crewmembers responsible for the engine and related technical systems. It will be used also at shore by the inspectors to compare the performance of technical systems on different vessels in their fleet.

1.3 Mission Statement

The general idea of the TCI and TSI concept is to provide standards so that information can be moved easily between the various technical subsystems and management systems, possibly in both ways. The standards shall cover the structure of the indicators as well as the network infrastructure for the exchange of indicators.

1.4 Scope

The expected outcome of the activities documented in this report is *to make technical information (fault reports, alarms, and status information) available from the technical systems to the technical management systems. This information should for the intended use, possibly be in the form of technical index values (TCI)* [TAv34].

Slightly paraphrased, the purpose of this report is to provide guidance on the ICT infrastructure to ensure simple integration of new Flagship modules into current state of the art integrated ship control (ISC) systems. ISC is here used in the broadest meaning, covering any relatively advanced ship automation/safety/navigation system that is more or less integrated.

This is illustrated in Figure 1 where the most relevant Flagship “modules” are shown at the bottom and the existing ISC sub-systems are shown at the top. The general idea is to provide standards so that information can be moved easily between these systems, possibly both ways. The types of information are also *indicated* in the figure. Information that will be utilized, also in Flagship, is indicated as a solid line. Information that *may* be utilized is indicated as a dashed line. The arrowheads show the most likely flow of information.

The diagram also indicates what information is most likely to be used on the ship for real time operation and planning (top "bus") and what information is most likely to be used off the ship for more tactical and strategic purposes (bottom "bus").

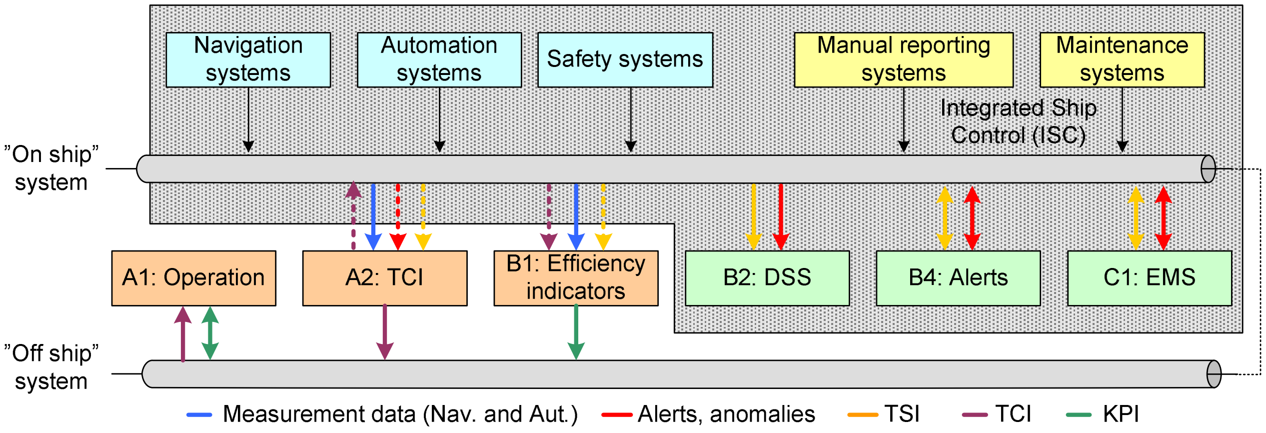


Figure 1 – Scope of D1

The shaded area represents the ISC: It consists of the “traditional” ISC modules (top) as well as the alert management module and the emergency management module (EMS). Other flagship activities may provide modules or software systems like, e.g., the Technical Condition Index module (A2) and the Decision support system (DSS) that may or may not be part of the ISC. If they are not part of the ISC, they will typically be run as off-line modules or non real-time modules.

The figure also indicates two new inputs and outputs from the Flagship modules. One type is a "Key Performance Indicator” (KPI), which may cover technical condition (Technical Condition Index – TCI – can be viewed as a type of KPI), environmental footprint, energy efficiency or other issues. The other is a technical status indicator (TSI - system status data) that is similar to the KPI in that it gives a high level indicator for a system’s or a component’s availability to perform its intended function. However, the status indicator specifically covers operational issues like functionality level and reflects the current ability of the component rather than past or future performance. Later sections will give more accurate definitions of these issues.

The TSI will be associated with a “description” of the system based on Technical Function Blocks (TFB). A TFB shows how a technical function can be decomposed into sub-functions and how these sub-functions relate to each other. In its simplest form, it is only an empty and monolithic block that cannot be decomposed.

The purpose of the TSI and associated technical function blocks is to make it easier for decision support systems or operators to see the functional consequence of a technical malfunction or problem. However, the TSI will be strictly limited to say something about the specific technical system or subsystem. It will not be able to say anything about how this technical system is used to realize various ship management functions – unless this has a one to one relationship to the TFB.

In addition to a description of the TSI and TFB, this document also contains an annex giving an overview of more general data transport mechanisms in current use on ships today and some forecasts as to what will be required in the future to realize the Flagship objectives.

1.5 Contribution to state of the art

The main purpose of sub-project D1 is to support other sub-projects by providing technical interface specifications. However, the concept of the TSI may also have applications outside the purely practical issues within Flagship. In particular, the concept of TSI could be important in conjunction with alert handling on ships. Thus, this document will be part of the input to sub-project B4 that deal with alert management. This document will also be used as background to the IMO work on central alert management, although it is at time of writing not likely that this concept will be included in the new IBS performance standard.

The TSI and its TFB can also provide a link to the work being done in A2 on TCI. The technical hierarch used in A2 can in principle have the same basic structure as the TFB hierarchy although the content of each node differs. This will be investigated in the later phases of the Flagship project. This document points out how this link may be exploited.

The work presented in this deliverable extends the work from DSS_DC [DSS07] in providing a more pure technical status measurement than was attempted in the previous project. DSS_DC attempted to look at on line technical *condition* monitoring as a means of giving the ship crew useful data on the ship's performance ability. However, it became clear that this information, although useful in itself, was less meaningful for on line decision-making than technical *status* type information. This though has been followed up in Flagship D1 and the result is the TSI and how that is linked in to the TCI. This also has an impact on the data structures and standards that are used to encode this information.

1.6 Short overview of document

This chapter (Chapter one) is the introduction to the document. Chapter two contains definitions used in this document. Any abbreviations used in the document are listed at the beginning of the document, after the executive summary.

Chapter three gives an overview of various existing indexing schemes and other information structuring initiatives that are relevant to this sub-project.

Chapter four describes the concept of the Technical Function Block (TFB) and how the TSI is associated with it. This is basically the definition of the concepts with some very basic and not necessarily representative examples.

Chapter five gives some more concrete examples of TFB and TSI and how these can be used in a “real” setting.

Chapter six contains the detailed data modelling for the TSI and the TFB.

Chapter seven contains the conclusions that can be drawn from the work performed so far in sub-project D1.

Chapter eight contains the list of references.

Annexes in chapter 9 contain information that is not central to the overall discussion or is of too much detail to keep in the main text.

2 Definitions

2.1 Alert

Alert is defined as a common term for “Alarms”, “Warnings” and “Cautions” in [NAV 53/WP.2]. It is expected that this terms also will be used in the updated code on alarms and indicators. This document uses this general definition. The different types of alerts imply a prioritization:

- Alarm: This signals a condition that requires immediate action as safety of crew or ship is endangered.
- Warning: This signals a condition that may develop into a dangerous situation over time.
- Caution: This is information of technical and/or operational nature that is brought to the operator’s attention, but which may not require any specific action.

In particular, one should note that many technical alerts are of “warning” or “caution” type. As an example, a failure in one of two redundant technical systems has no immediate operational consequence (see alert types).

Note also that the above reference requires that warnings are upgraded to alarms if they are not acted on within a context dependent time frame.

2.2 Alert types

Alerts may be of conceptually very different types; mainly dependent on the context they occur in. In this document we will distinguish between the following types of alerts:

- *Technical alert*: This is an alert that indicates a certain degree of malfunction in a unit or a system interconnection. An example may be an earth fault in an automation system data bus. This may or may not have a functional consequence.
- *Operational alert*: This is an alert that indicates an operational status, e.g., waypoint reached, tank level exceeded. It normally implies a change in operational procedures, status or attention.
- *Functional alert*: An alert that indicates loss of functionality, i.e., that the propulsion system cannot perform at full power.
- *Safety alert*: An alert that indicates a situation that is a direct hazard to ship, crew or environment. A typical example is a fire alarm or bilge alarm.
- *Emergency alert*: This is a variant of the safety alert that is directed at individual or all personnel on board. Go to muster stations is the typical example of an emergency alert, but alerts related to release of CO_2, closing watertight doors and similar are classified as emergency alerts.

Technical alerts represent technical problems that should be closely linked to a TSI. Also, technical alerts may not have any safety, operational or functional consequence and may not need to be brought to the operators' immediate attention. Thus, the focus in this report is the link between the TSI and technical alerts and how the TSI can be used to reduce the cognitive load on the ship operator.

Note also that this classification is ambiguous and should be used with some care. Deliverables for Subproject B4 will also discuss different types of alerts and the application of similar classification schemes.

2.3 Decision Support System – DSS

A DSS in the context of this document is a module that uses data from one or more of the ISC modules and derives additional operator or system information from that. Thus, with reference to Figure 1, a DSS module is on the lower row of blocks while an ISC module is on the upper.

It is clear that the ambition level of Flagship is to make DSS a part of the ISC, but for the discussions in this document, the above distinction remains.

Note also that modules like C1 and B4 will be more easily integrated in the ISC modules and that the interfaces used by these modules are both of the DSS and ISC type.

2.4 Index

An index is a normalized number (typically 0 to 100 inclusive) that have no dimension. It is used to give a relative measure of a condition or status compared to a reference value.

One may in some cases look at this as a "percentage of full performance", but this is not necessarily true.

See also Indicator (2.5).

2.5 Indicator

An indicator is a dimensioned value that serves as a high level measurement of a component or system's performance in some respect. One example may be tons fuel per kW produced which could be an indicator for energy efficiency.

This document does not always strictly distinguish between indicator and index (see 2.4), but it is normally clear from the context what the implicit meaning is.

2.6 Integrated Ship Control – ISC

Integrated ship control (ISC) is a collection of sub-systems for control and monitoring of a ship. This can include nautical sub-systems (heading control, RADAR, ECDIS, positioning systems etc.), automation sub-systems (engine, power generation, life support, cargo control etc.), safety sub-systems (fire alarm, bilge alarm detectors, stability and strength monitoring, etc.) as well as communication sub-systems (GMDSS, Public Announcement, UHF and other types of communication).

An ISC will be a more or less integrated collection of such sub-systems. This paper does not necessary require any particular degree of integration between these sub-systems. ISC is used as a term to identify the collection of all the sub-systems on a given ship. Note that this definition is not in line with Integrated Bridge System [MSC.64], which requires certain functionalities to be present for a system to be an IBS.

For the purposes of this document, a distinction between DSS modules and ISC modules have been made as described in section 2.3.

2.7 Key Performance Indicator or Index – KPI

See Performance Indicator (section 2.9).

2.8 Notification

An observation by a human of a technical abnormality – normally a defect or degradation of a piece of equipment. The notification may be registered on paper or electronically.

2.9 Performance indicator or index – PI

A Key Performance Indicator or index (KPI) is normally defined as a numerical value that gives a relatively high level description of a critical aspect of one organizations operation. For the purpose of this paper, we will normally use the term “Performance Indicator” (without the “Key”) to denote a coding of a certain sub-system or function’s capabilities. This PI can indicate one of several different aspects of the components capabilities:

- Technical condition index or indicator (TCI) – Measurement of condition state, relative to a system or component’s condition as new (see 2.11).
- Technical status index or indicator (TSI) – Measurement of a component’s or system’s ability to perform its intended function (see 2.12)

These are the indicators relevant in this document, but also other aspects can be encodes, such as:

- Environmental performance indicator – Measurement of how a system or component impacts the environment compared to a reference value, which represents an “optimal” performance.
- Efficiency indicator – Measurement of how efficient a function or component is performing in terms of overall resource consumption compared to output.

In general, one can look at the last group of indicators as "Key performance indicators" (KPI), i.e., indicators that tell something about the overall performance of the observed system.

2.10 Tag

A tag is a text label identifying a specific technical unit, component or system. The text label is usually structured, but in the context of this document, the structuring is of no issue – the tag is looked at as a global identification for the above mentioned entity.

2.11 TCI - Technical condition index

A TCI is a numeric value, between 0 and 100, showing a technical component's, sub-system's or systems technical condition as compared to when it was new. Thus, reduction in the TCI indicates a certain degree of "wear and tear". A more detailed description can be found in section 3.1.

There may not necessarily be a relationship between the TCI and the component's ability to perform its function as measured by the TSI. In a simple example, a pump that has been turned physically off has a zero valued TSI while it may be brand new and have a TCI of 100. Vice versa, a decrease in the TCI of an engine may, e.g., result in bad environmental performance, but have no impact on the actual function as measured by the TSI. However, a TCI that shows zero or near zero should normally mean that the TSI is likewise reduced and this should lead to an immediate technical alert.

Note also that a reduction in the TCI normally must have a physical effect for it to be measurable. The exceptions to this are measurements that are only made to assess the technical condition itself, e.g., thickness or possibly vibration measurements.

2.12 TSI – Technical status index

A TSI is a measurement that shows a component's or system's ability to perform its intended function. Dependent on context, this can be looked at as the availability for its intended function or its functional capability – see also TCI (2.11).

3 Overview of indexing schemes for ships

Indexing of various performance and condition parameters related to ships or ship systems is an aid to keeping track of performance in various areas and also to decide when and how corrective measures are to me made. This section will give an overview of some indexing schemes that are in use by the Flagship partners and in particular look at the synergy between the TCI and the TSI.

Indexing will often require that systems are broken down into sub-systems and components, typically in a tree structure. This chapter will also look at some such hierarchical approaches.

3.1 TCI - Technical condition index

The concept of the TCI is described in [TM99]. It is a numeric value, between 0 and 100, showing a technical component's, sub-system's or systems technical condition as compared to when it was new. Thus, reduction in the TCI indicates a certain degree of "wear and tear". TCI are normally derived from various technical measurements at the lowest level that is aggregated up through the system's physical assembly and is represented as a tree where each node represents a component or sub-assembly or the complete system at the top. The TCI as described in [DSS07] is calculated through an aggregation hierarchy as shown in Figure 2.

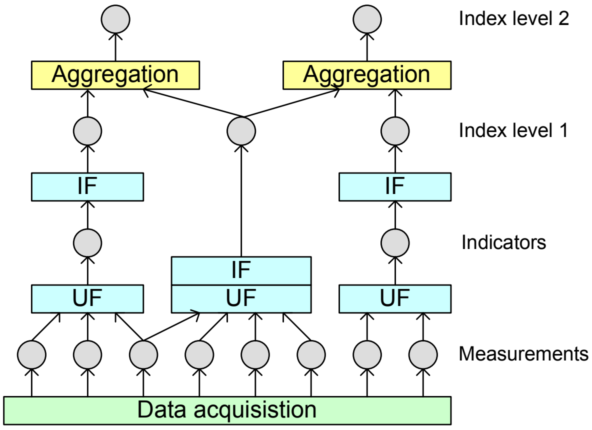


Figure 2 – The TCI calculation hierarchy

All TCI are based on input measurements, either in form of on line data, logged data, deviation reports or other information that is able to say something about a sub-system's technical condition. The measurements are used to do indicator value calculations through a set of User Functions (UF). The indicator values, if available, will give a physical relationship between measurements that is meaningful for technical condition, e.g., relationships between flows, pressures and valve positions.

The indicators are normalized to the 0-100 ranges through Index Functions (IF). This normalization is not necessarily linear or even continuous. Often it is a linearization between a set of threshold values. Figure 3 shows a typical Index Function, again from [DSS07]. The colours indicate acceptable (green), problematic (yellow) and critical (red) technical condition levels. Note that these levels are connected to absolute value ranges on the index scale.

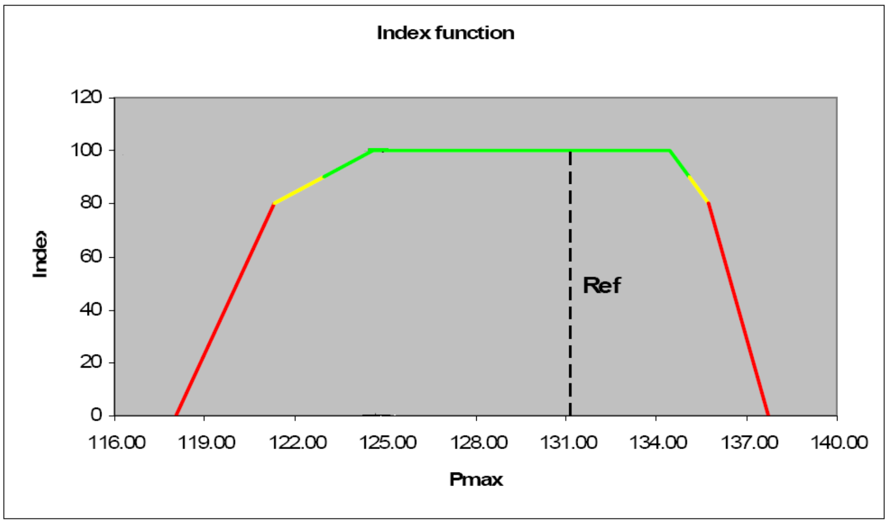


Figure 3 – Typical Index Function

At this stage an index value has been created, at the lowest level of the hierarchy. At higher levels index values are aggregated through aggregation functions to higher-level indexes. An aggregation function is similar to a fuzzy logic [Zadeh65] function in that it performs a pseudo-logical transform of a set of scalar values to one new scalar value where the result can be, e.g., a weighted sum, a logical and arithmetic combination or even an actual fuzzy logic expression over the input values.

The TCI are aggregated in a structure that is based on the technical breakdown of a system or sub-system. This is illustrated in the simplified diagram in Figure 4. The '&' symbol in the circles indicate a cumulative dependency from right hand technical functions to the next left function and a '+' symbol indicates a redundancy in right hand functions. The 'f' indicates a transfer function between input values to indexes.

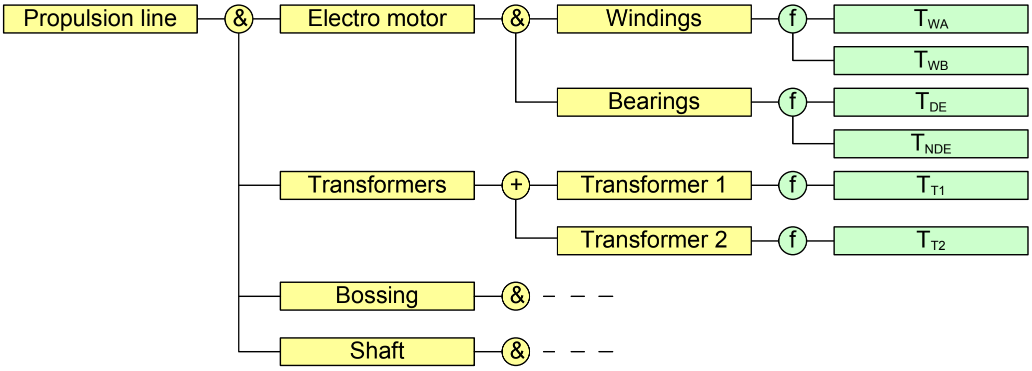


Figure 4 – Simplified TCI structure for propulsion line

The left hand side (yellow) shows the logical part of the structure, or the TCI tree, while the right hand side (green) show the input measurement values – in this example, a number of temperature measurements. Note that a consequence of the discussion above is that input values only occur on the lowest level of the structure. This will also be reflected in the TFB as discussed in Chapter 4.

The TCI is typically a slowly varying index that is often checked with a sampling frequency of weeks or months. While it is obvious that the TCI may have an impact on the equipment's ability to perform its intended function it is also obvious that it will not catch fast developments in capabilities due to break down, faults in connected systems and other causes. This is the reason that the TSI has been introduced as a complement to the TCI.

The TCI is further discussed in deliverables from Subproject A2. This document will only address standards for representation of TCI and associated aggregation trees.

3.2 TSI – Technical status index

The TSI has been developed in Flagship sub-project D1 and is a new concept. Literature searches have not revealed any similar concepts that combine a component based hierarchical breakdown of complex systems with a "diagnostic" capability made available to the user on line.

The TSI is a measurement that shows a component's or system's ability to perform its intended function. The TSI is discussed in detail in Chapter 4 so this section will only point out the similarities to the TCI.

As will be shown later, the structure of the TSI matches that of the TCI and the structure of the TFB (Technical Function Block) hierarchy matches that of the TCI hierarchy. In fact, for most systems one can expect that the general structure of the two systems is closely aligned and even identical. The exception will be the lowest layer of input values. This will be investigated in section 3.3. Thus, it is clear that the data structures used to represent TCI and TSI is aligned. This is further discussed in Chapter 5.

3.3 The TSI compared to the TCI

The Technical Condition Index (TCI) has been introduced in Flagship by subproject A2 and intends to give a high level indication of the technical *condition* of a component, subsystem or system. The TCI says something about the maintenance condition of the unit as compared to new, but this does not necessarily say anything about the availability of the unit for its intended function. Thus, there is a need for an additional indicator that can integrate the information in the alerts and notifications as well as possible also the TCI and say something about how well the unit can be expected to perform at this specific moment. This is the Technical Status Index (TSI).

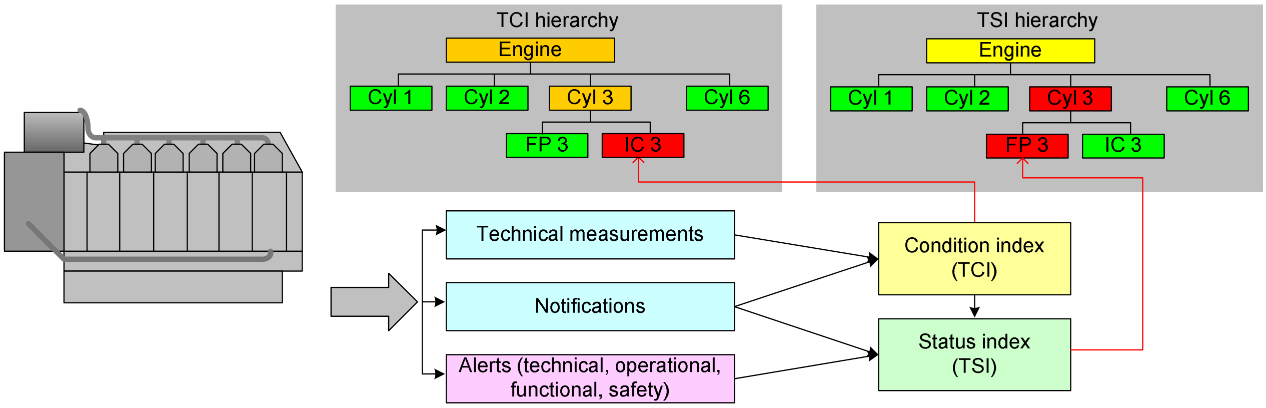


Figure 5 – Example TSI and TCI

This concept is illustrated in Figure 5 where an engine has a reduction both in its condition and status index. However, for the condition, this is caused by analysis showing a reduced technical condition of the fuel injection system (IC 3). This may not directly impact the engine's ability to deliver the expected power. However, it may, e.g., give higher than expected fuel consumption or more emissions than specified.

The status degradation is caused by a complete stop in a fuel pump (FP3). This may not have anything to do with wear and tear, it may e.g., be caused by a problem in the power supply to the pump. Thus, there need not be any relationship between TSI and TCI.

Note also that the consequence on top level may be different for the status and the condition index. This example shows the condition degradation as more severe than the status degradation. This could mean that a 20% loss of available power is less critical than the bad condition as the latter, e.g., may mean unacceptable emissions to air.

3.4 SPI – Ship performance index

A last form of index that is used by Flagship partners is the Ship Performance Index (SPI[1]). This is a set of standardised KPI (Key Performance Indicator) that was developed through a research project involving ship managers from around the world. The purpose was to define a set of KPI that could be used to represent and benchmark various ship performance parameters on a high level. Performance measures covers among other things environmental impact, efficiency, safety and security. The SPI is intended for use at management level and particularly for benchmarking ships in a fleet and possibly also between different operators' ships.

This report will not go into more details on the SPI. However, it may be interesting to see that both TCI and TSI may be input to a SPI. Thus, a more holistic approach to data collection, systemization and use, e.g., through technical hierarchies, also opens up possibilities for information use at higher levels, e.g., in management levels or even by the general public.

3.5 SFI system

For a hierarchical technical indexing scheme to be work, one needs to create a system breakdown that can be used as basis in the hierarchy. This system needs to have two important properties:

1. Its overall structure should be as general as possible so that systems from different manufacturers can be compared and so that one is able to provide a module-based approach to building an actual hierarchy.

2. It must be possible to specialize it into more detailed and specific systems for a real implementation with all its details.

One solution to this is to build a tree structured classification system that is general on the higher levels and can still be detailed enough on lower levels. Creating such a structure is not trivial and some of the problems in the context of the technical function block approach are discussed in section 4.7.

---

[1] www.shipping-kpi.com

For ship systems there is one system that has had some degree of general acceptance that it may be worth mentioning in this context and this is the SFI group system [Xantic01]. It was originally developed by the Ship Research Institute of Norway in 1972, but has been commercialized and is now owned by Xantic. It is a hierarchical decimal system as illustrated in Figure 6.

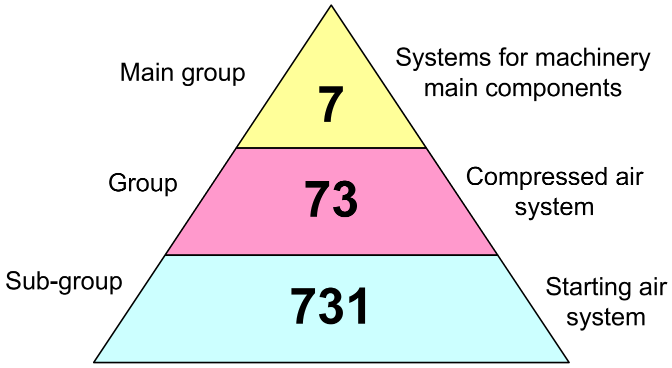


Figure 6 – Structure of SFI system [Xantic01]

The currently defined main groups are:

1 – Ship General
2 – Hull
3 – Equipment for Cargo
4 – Ship Equipment
5 – Equipment for Crew and Passengers
6 – Machinery Main Components
7 – Systems for Machinery Main Components
8 – Ship Common Systems

Groups 0 and 9 are not in use. As the system is proprietary, it is not possible to repeat more levels of the model here. However, the above figure should give an idea of how it is built up.

3.6 DNV system breakdown

DNV has published an alternative way to create a functionally oriented vessel model [Vindøy08]. This proposal has been published and is available from DNV. DNV states that the model will be updated at least annually and it is a part of the classification activities at DNV.

The system is in itself similar to the SFI system in that it is a "Universal Decimal Classification" (UDC) based system, but it has somewhat different classes of functions. The main functional classes are listed below.

000a	General
100a	Main structure
200a	Stability, watertight and weather tight integrity
300a	Hull equipment
400a	Propulsion and steering
500a	Electric power

600a	Machinery and marine piping systems
700a	Navigation, communication and control
800a	Safety
900a	Environment
1000a	Dry cargo
1100a	Liquid and gas cargo
1200a	Drilling and well intervention
1300a	Diving

The classification system has more than 2500 entries in its current version. Some of the top-level entries are listed in Annex D. As the system is open and is fairly comprehensible, it may be a good alternative as a starting point for a more system breakdown.

3.7 Alert management

Alerts can be looked at as a kind of binary index value that warns about a failure in a technical component or system. This is particularly obvious when a low level failure causes a small or large avalanche of secondary alerts. One example is a high temperature alert in a transformer that eventually may cause a full stop in a propulsion line and corresponding alerts throughout the technical hierarchy. Thus, one aspect of alert management is to recognize this hierarchy and distinguish between alerts that are related to eventual loss of functionality and those alerts that are related to fault propagation in the technical system.

Section 5.4 will discuss this issue in more detail and also propose how the TSI concept can be integrated in alert management.

3.8 Fault trees and FMEA

Failure Modes and Effects Analysis (FMEA) and Fault Trees are methods commonly used in ship system design to analyse what happens if components of the system fails. This is mostly used on safety critical systems.

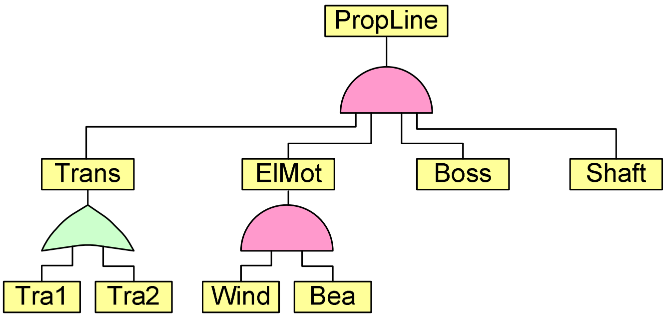


Figure 7 – Fault tree for propulsion line example

Fault Tree [NR81] is a method that enables a quantitative analysis of a system's failure probability from a tree-structured breakdown of the system. This tree structure will be similar or identical to the structure used in the TFB or the TCI hierarchy. As an example, the TCI hierarchy from Figure

4 is represented as a simplified fault tree in Figure 7. The main TCI nodes are found as nodes in the fault tree and the aggregation functions are replaced with *and* (red) and *or* (green) functions.

One should note that the fault tree only overlaps with the TCI tree in the cases where also the TFB hierarchy overlaps with the TCI tree, i.e., when a sub-system function is related to the physical structure of the sub-system rather than to other functional relationships. Thus, a more complete fault tree may also include control and monitoring systems that normally have no place in the TCI hierarchy. This is a problem with the mapping of TCI trees to TFB trees that is discussed in section 4.6.

FMEA (Failure mode and effects analysis) is another analysis type that normally relies on a hierarchical breakdown of the analysed system. The FMEA will normally aim at a more qualitative analysis and is particularly suitable for detecting single point of failures in complex systems. As the FMEA also may be based on a tree structured system description it may provide input for design of the TFB hierarchy.

3.9 Shipdex

The Shipdex[2] project [SX08] has developed a protocol, mainly for describing ship components and their documentation, based on the S1000D system[3]. S1000D is an international standard for the production and reuse of data, stored in a common source database (CSDB). The S1000D project was originally started by AECMA (Association Européene des Constructeurs de Materiel Aerospatial), but is now co-managed, under memoranda of understanding, between ASD (Aerospace and Defence Industries Association of Europa), AIA (Aerospace Industries Association) and ATA (Air Transport Association). The standard is managed by the TPSMG (Technical Publications Specification Maintenance Group), which is co-chaired by the 3 organizations and staffed by specialists from industry and MoDs around the globe[4].

Shipdex uses version 2.3 of the S1000D specification [S1000D23]. The later version of S1000D is 3.0 and version 4.0 is scheduled for the second half of 2008.

Shipdex is currently used to organise documentations for ships and ship systems as well as for more or less complete descriptions of the ship. Thus, Shipdex can also be used to populate maintenance databases. However, Shipdex does not so far define how to exchange maintenance related data. This is defined in S1000D so it should be a fairly simple extension of the protocol.

Shipdex uses the SFI group system (see section 3.5) as main key in tagging components. The specification is in principle open, but one will from a practical perspective need both memberships in the Shipdex organisation and access to the SFI system to use it.

S1000D is a very comprehensive system and seems to contain all types of information and messages one could want to use in a maintenance perspective. However, it is fairly complicated

---

[2] www.shipdex.com

[3] www.s1000d.org

[4] Source: http://www.s1000duser.info/home.htm

and verbose in its implementation and it is not directly applicable as a real-time or near real-time message format.

Thus, to transmit maintenance related information from the technical system to external management systems one should opt for a simpler format.

4 Technical function block and status indicator

This section will give a more detailed description of the concept of TSI and TFB.

4.1 Introduction

Modern ship systems are increasingly modularized and equipped with advanced electronic control and supervision. This has two effects:

- The module may not be as easy to understand as more old-fashioned sub-systems, which consisted of recognizable mechanical components.
- The module may produce a high number of alerts that can be difficult to relate to a functional consequence, as these alerts are associated with anomalies in the electronic control and supervision system rather than functional problems in the unit.

Also more conventional ships are getting more complex and there are a high number of alerts and notifications produced even in these. This is more thoroughly discussed in subproject B4 of Flagship, but for our purposes, it obviously also have consequences:

- The operator is presented with a long list of alerts and notifications and it is again not obvious what consequences these have for overall ship functions.

To make it easier to relate to this complexity, Flagship sub-project D1 proposes to introduce the concept of Technical Function Blocks (TFB) and Technical Status Indicators (TSI).

4.2 The overall concept and graphical representation

The Technical Function Block (TFB) is in general represented as is shown in Figure 8. Vertical lines represent electronic signals (input or output) and thus the observable aspects of the TFB as seen from the ICT system. The horizontal lines show expected effects of the function block and, where applicable, reliance on other TFB effects.

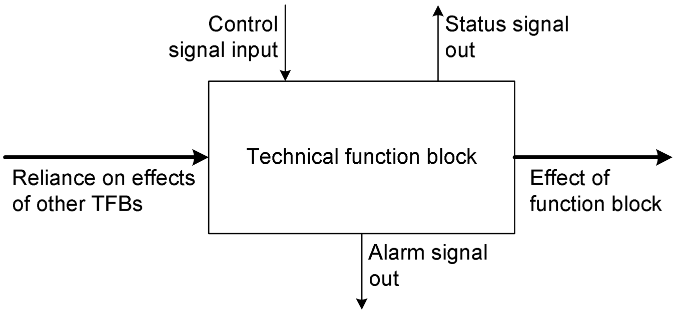


Figure 8 – General representation of a low level TFB

The left side input effect arrow is only applicable when one looks at relationships between TFB, typically inside a higher level TFB.

The Technical Status Indicator (TSI) measures to what degree the TFB is able to provide the desired effect. A value of zero means that the TFB does not provide any useful function at all and a value of one means that it is fully operational without any functional degradation. Any value between one and zero means a partial reduction in performance where the measure may be dependent on the TFB. The TSI is not normally indicated in the outgoing topside arrows. It is usually generated inside the automation system or in a system connected to the automation and monitoring system.

$$TSI = f(I_{1,} I_{2,} \ldots, O_{1,} O_{2,} \ldots, A_{1,} A_{2,} \ldots) \qquad (Eq.\ 1)$$

For an atomic lowest level TFB, the TSI is calculated as a function of inputs, outputs and alert signals as shown in Eq. 1.

The TSI for a given TFB on a higher level can be calculated from the internal networks of lower level TFB and their TSI. This is exemplified in Figure 9.

The TCI represents the technical maintenance condition of the TFB and is further discussed in section 3.3.

4.3 A TFB hierarchy

A TFB on a certain level is typically composed of a number of lower levels TFB. This is exemplified in Figure 9 where the inside of the big rectangle shows three TFB that together make up the one higher level TFB.

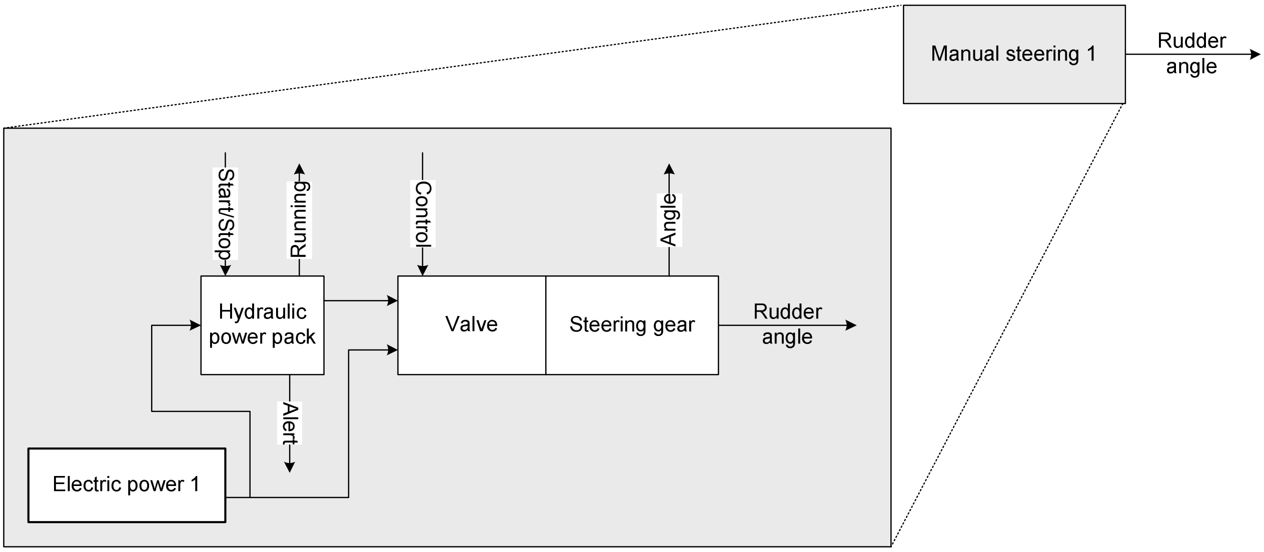


Figure 9 – A TFB hierarchy

The figure shows a manual steering function block that comprises the following lower level TFB:

- The hydraulic power pack is a standard atomic TFB with one input, one status signal and one alert (low pressure). From this, it should be possible to calculate a TSI.
- The valve and steering gear is shown as one atomic TFB with one input and one output signal. As these to physical units are not independently observable, there is reason to divide them into two separate TFB.
- "Electric power 1" is an aggregated TFB with lower level TFB so that it will not have any observable electronic inputs or outputs that are necessary to indicate on this level.

The high level TFB will have a TSI that is a function of the lower level TFB TSI. In the simple case where the TSI is just indicating working (value is 1) or not working (value is 0) this could be represented as in Eq. 2.

$$TSI = TSI_1 \cdot TSI_2 \cdot TSI_3 \qquad (Eq.\ 2)$$

4.4 Components of the TFB system

The technical function block system can be illustrated in UML[5] notation as shown in Figure 10. The label "XSDxxxType" on all classes is an artefact from the designer tool that allows for later export into an XSD[6] file.

This model shows the generalized technical tree (Technical Tree) as being common to both the TCI and the TSI hierarchy. This is an assumption that may not always hold, so in the practical implementation one needs to cater for some nodes being only of TCI or TSI type. However, it should be possible to create one general tree common to both indexing systems.

The tree is a recursive list of nodes (Technical Node) that builds the tree structure. To each leaf node[7] in the tree one can associate a number of tags, through the Tag Name element, that represents the input or output values used to calculate the lowest level indicators and then indexes. The actual signal value is turn associated with the tag. Higher-level nodes in the tree will calculate TCI and TSI only from lower level nodes.

Each node will be specialised into specific data structures for respectively TSI and TCI calculations. This represents the different calculation models and possibly different downward relationships.

The actual TCI and TSI values will be associated with the un-specialised node through a node identifier (NodeID) text string.

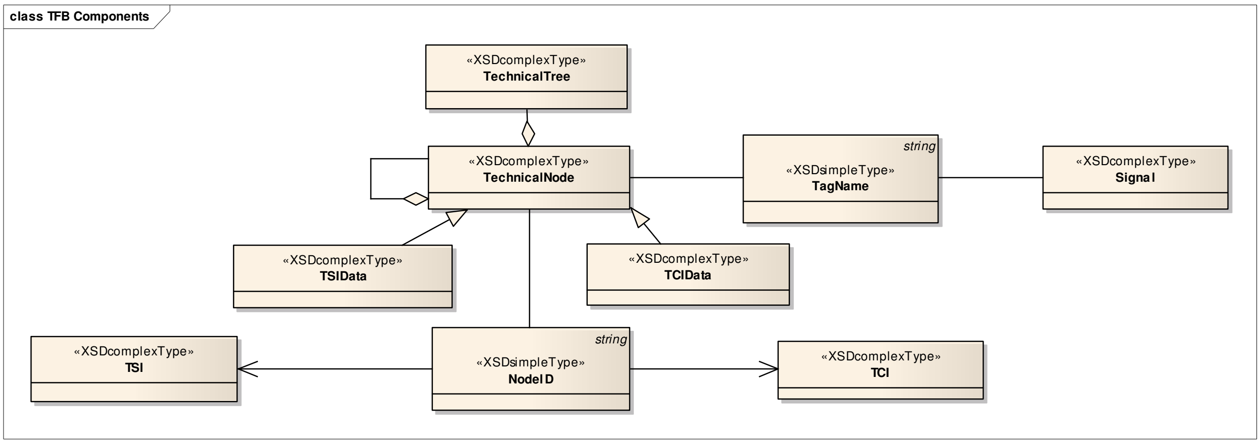


Figure 10 – TFB Components in UML notation

---

[5] Unified Modelling Language, see www.uml.org

[6] XML Schema specification format, see www.w3c.org

[7] Leaf node is a "bottom" node without any children nodes

This is a very schematic representation of the data model and it will be further elaborated in Chapter 6. However, it shows the difference between the specification parts of the model, which is basically the technical tree, and the operative parts of the model, which are the tags, measurement values, TCI and TSI.

It may also be useful to highlight the differences in association to the technical tree from the measurement values with tag names versus the TCI and TSI using the node identifier. This reflects the difference in perspective between the two types of indicators, one being directly associated with the automation and monitoring system and the others being associated with a somewhat more abstract representation of the system.

4.5 Capturing the TFB tree

The main problem with the concept represented by TCI and TSI is the cost and complexity of engineering, i.e., the problem of creating the models that are necessary for the distribution of the technical index values. Thus, the capturing of the technical tree and the associated TSI and TCI nodes is probably the most critical issue in this context.

As was discussed in section 3.8 one possible candidate for information about technical trees is any safety analysis that has been performed. If this is in the form of a fault tree, it should be particularly useful. It may also be possible to use data from a FMEA, but this may be less suitable. There are some electronic formats for fault trees, see e.g. [FSAP07][8] that are able to export information in XML which in turn may be converted to a TFB tree. The fault tree may also be used to create the actual function blocks at least on a high level as it contains certain logic functions and also probabilities that may be used.

One could also be able to use electronic assembly information for some systems to create the TFB. This will, however, require quite a bit of manual manipulation, as one normally will see a much higher degree of detail in such documentation. Neither is it useful to model components that cannot be observed through measurements or manual inspection reports (notifications). Thus, there is a need to aggregate components into nodes in the TFB where each node has a useful function in the tree.

In general, if the TFB concept is adopted by the industry, one will expect that ship components will come with a vendor supplied TFB sub-tree that can be integrated in the overall system. This issue is discussed in the next section.

Also, one may be able to develop general TFB for certain "standard" systems. This standard model may have to be adopted for specific installations, but it would in any case provide a starting point and reduce the required engineering. Examples of standard TFB could be a main engine, auxiliary engines, generators and other components that are routinely controlled and monitored electronically and which have a fairly common physical construction.

---

[8] http://sra.itc.it/tools/FSAP/

4.6 Integration of TFB trees

Another problem that has to be addressed is the integration of different TFB trees into one covering larger systems. This is an issue particularly for power production systems that are relatively complex and may consist of components from different vendors. One example of an integrated system that could be constructed like this is presented in Figure 11.

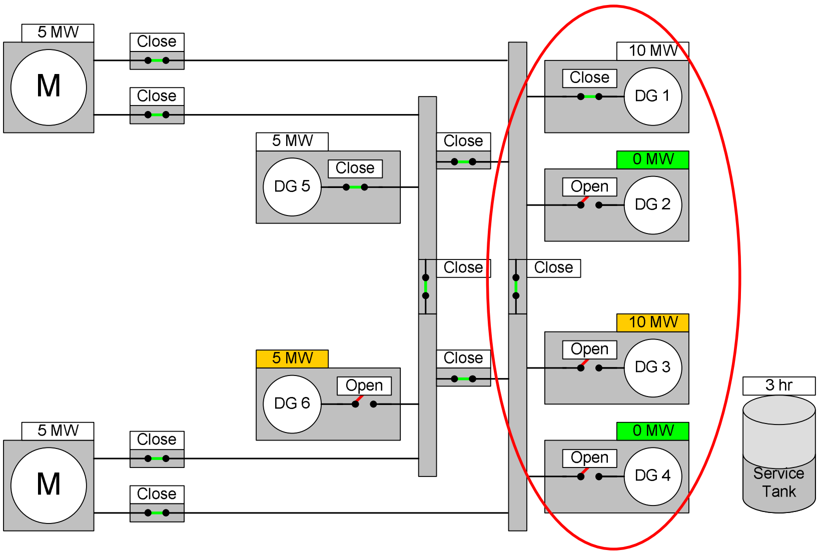


Figure 11 – The DSS_DC example power plant

This example represents part of the power generation function on a diesel-electric ship, taken from [DSS06]. It consists of a number of generator sets connected together through a power distribution system and connected to two electrical motors.

One solution to the integration problem is to let the system integrator also provide the integration of TFB trees. This should normally be a fairly simple procedure as the top level TFB tree will be relatively shallow. It will consist of a collection of the individual TFB trees on the leaf nodes and any higher-level integration nodes. An example is shown in Figure 12. It corresponds to the circled area in Figure 11.

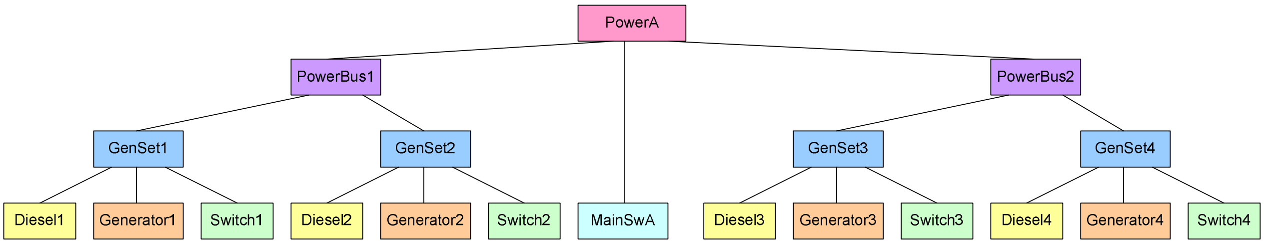


Figure 12 – Example integrated TFB tree

Each of the bottom-most blocks in this TFB corresponds to one physical unit that has its own TFB sub-tree. By taking the TSI and TCI values from the sub-tree's top node, one can propagate this into the next higher-level tree and calculate new TSI and TCI values.

4.7 Limitations of TFB trees

There are limits to what a TFB tree can represent. A slightly contrived part of a heat, ventilation and air conditioning (HVAC) system is shown in Figure 13. This shows the mechanical components as well as the power and control system. The system consists of a duct with one fan and five dampers to provide air to some locations. Each of the controllable units are controlled and monitored by remote control units (RCU) which in turn is connected with two instrument networks to the Main Automation System, module 1 (MAS1). In addition, the connections to the power systems are indicated.

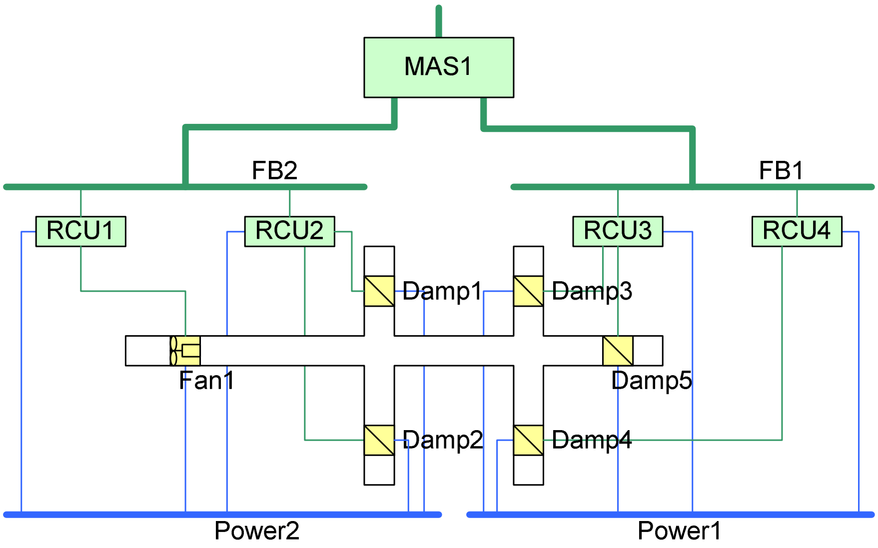


Figure 13 – Example combined technical and control system

The main TFB for this system would model the mechanical system that provides fresh air to various locations on the ship. However, it is obvious that faults or problems in both the power distribution and control system will influence the overall functionality of the HVAC system.

A simple tree structure as the TFB will not be able to accurately show local functional consequences of various failure modes. In the proposed system, the HVAC components would have to be modelled as one TFB tree and the automation and power distribution system as other trees. Thus, it will not directly be possible to find out what a problem in the automation system will have for the HVAC function in a given area of the ship.

However, one may be able to indicate reduced overall functionality in the HVAC system if the three trees are integrated as mentioned in the previous section. Also, by using additional decision support functionality in the form of status assessment modules (see [DSS06]) one can also fairly easily visualize the local consequences of problems.

5 Using the TFB and TSI

This chapter will investigate the uses one can make of the TSI and the TFB. This will cover the capture of the TFB and associated TSI, the use of the TSI as a means to filter out alerts without immediate functional consequence and the TSI as a means to find root causes of system problems. The chapter will also look at the synergies between the TCI concept and the TSI system.

5.1 Integrating ship processes

The EIAMUG project [PRIAM] defined a model for control and monitoring that has been modified slightly into the below figure. This figure shows the operative phase of the model as the three vertical bars. In addition one could add design, commissioning and decommissioning as well as strategic planning functions. However, the focus in Flagship is on operation, as is shown here.

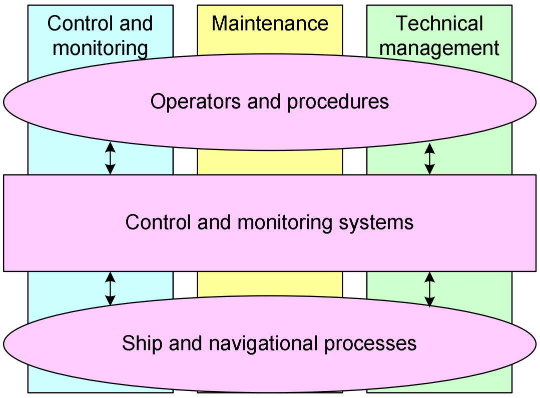


Figure 14 – Control and monitoring model

Operation, according to this model, can be divided into three distinct components. Although the developers of the model were thinking about process control, these components will also be found in ship operation:

- *Control and monitoring*: This is the operation of the ship and its systems, intended to facilitate the voyage as well as the wellbeing and safety of the crew and environment. Roughly, one can say that this corresponds to bridge functions, but it also involves technical systems like HVAC, fresh water, black water and galleys.
- *Maintenance*: These are functions related to the up keeping of the technical systems involved in the two above function groups. This is traditionally an engine room task.
- *Technical management*: These are functions related to finding the best way to use the technical systems. On a ship, this is mainly related to engine room functions, but does also involve technical superintendents and other shore personnel.

A general problem in ships today is that these three components are not very well integrated. Thus, information generated in one sub-system for one specific component will not necessarily be useful by others. This is one of the issues addressed by the TFB concept: Data that is generated from one of the three main process types can be structured and integrated to provide useful information for the other processes.

5.2 Overview of users of TFB information

There are potential users of TSI and TCI on the ship, in the owners' or managers' office and also in the surrounding world. Figure 15 presents a simplified version of this. It has used the EIAMUG (see previous section) model as basis.

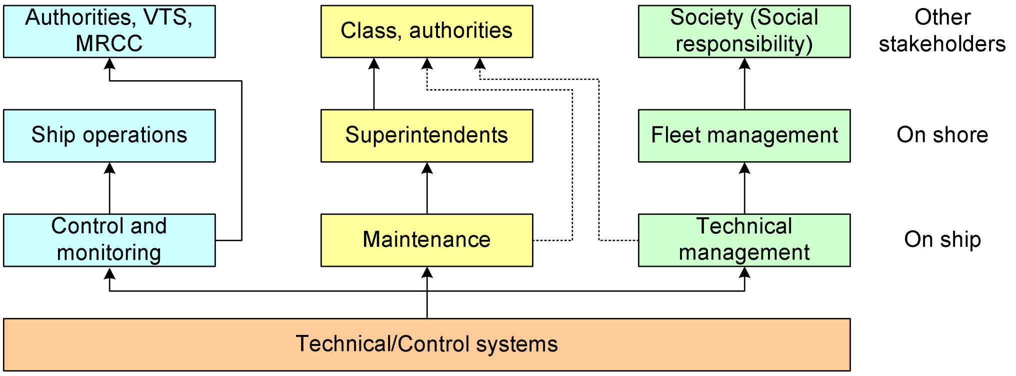


Figure 15 – TSI users

TSI are interesting to those in charge of control and monitoring (bridge) insofar as they indicate an immediate or short-term functional degradation. Thus, they are also of interest by ship operations department and external stakeholders as they may indicate lower than normal performance. External stakeholders may include coastal or port state authorities, vessel traffic services (VTS) or maritime rescue coordination centres (MRCC).

The TCI is obviously also of interest to those that deal with maintenance and technical management. We will not elaborate on this, but only point out that TCI and TSI may in principle also indicate negative environmental impact, which is of interest to the society at large. This is part of the issues discussed in the B1 sub-project of Flagship.

5.3 Using the TCI in maintenance

The TCI has been developed specifically to aid technical management by providing information on slowly varying trends that are not easily observable by manual means. The concept has been proven useful in the TOCC[9] system where more than 50 ships provide information on their main engines. The concept will also be extended in Flagship to also cover auxiliaries, generators and thrusters. Thus, it is clear that the TCI part of the TFB is useful.

However, also for TCI there is a cost problem associated with the specification of the TFB. Experience so far shows that the benefits outweigh this cost, but it would be even more favourable if the information could be reused also to provide TSI type data in the TFB tree.

5.4 TSI and operational alerts

The initial idea behind the TSI was to use it to convert technical alerts into functional consequences. This remains the bearing idea and should be represent a substantial benefit for ship crew as it can reduce the number of alerts that needs to be acted on substantially. It will also

---

[9] Technical Operations Competence Centre – www.tocc.no

enable the deployment of status assessment module displays that can give a better overview of the now situation for the technical systems [DSS05]. Thus, based on the TFB information one can:

1. Suppress alerts that have no immediate operational significance. This will reduce workload and stress level for officers of the watch.
2. Implement a status assessment display that can show functional consequence of alerts that have significance. This display will also be able to show technical status where no alerts are generated.
3. To some degree implement a root cause finding system that may help to identify the underlying reasons for technical or functional alerts.

In many cases one may also implement that a TSI at a value other than one (100% working) should generate an alert in itself (if no other has been generated), either technical or operational. If higher level TFB contains redundancy, the fault in a lower level may only be reason for a technical alert. Otherwise, an operational alert should be expected.

Also, one should consider adding the TSI as an attribute to any generated alerts. This will give a better link between the alert and the technical consequence and reason for the alert.

5.5 TSI as part of operator decision support

The most important use of TSI is as a decision support tool for the operator. This concept was discussed and described in [DSS06], but not tested other than through use of on-line TCI as replacement for a TSI.

The concept allows technical alerts to be presented as functional consequences when the TFB tree is traversed to the higher levels. This is exemplified in Figure 16, where, e.g., a high temperature alert in a generator bearing can be presented as yellow technical status for the relevant generator. Likewise, if the TSI indicates that the component is fully in-operational, the relevant icon should be red.

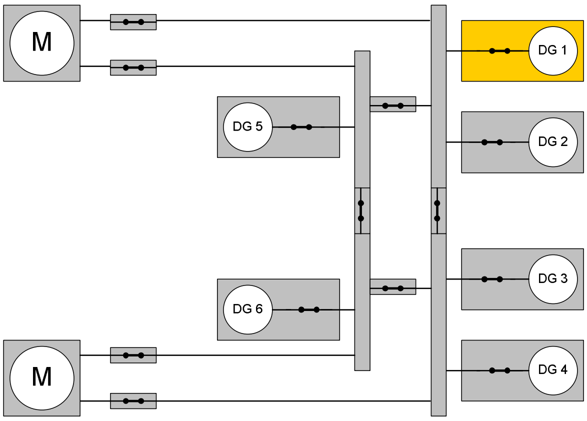


Figure 16 – High-level DSS status display

By using the underlying technical tree one can also visualize where the root cause for the problem exists. This is shown in Figure 17.

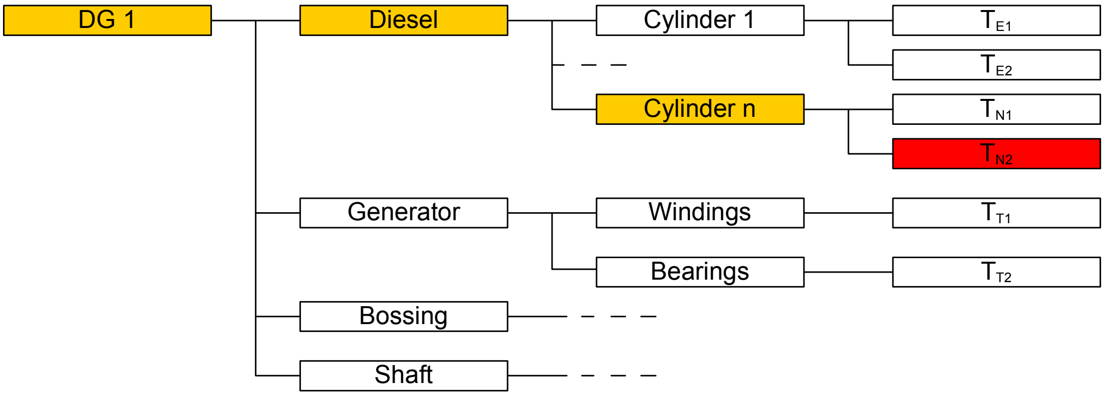


Figure 17 – Alert in a TSI tree

The concept of colour coding is already in use in the TCI system, where each node can have configurable information about when a TCI should be indicated as red, yellow or green/white. For the TSI one can argue that such thresholds should not be configured, but be predefined. In principle, a TSI with value 0 is red, one with value 100 is green/white and all between is yellow. However, for real systems it may be useful to allow other thresholds – in particular where analogue measurements are used to determine status as opposed to binary valued alerts.

5.6 TFB and technical management

Both the TCI and TSI will be useful for management as basis, e.g., for technical oriented KPI. This could provide a more systematic overview of technical system condition and status over time, comparisons between ships and operational regimes etc.

5.7 Practical implementation of TFB

The implementation of a TCI and TSI based system will need both input signals and the tree structure definition file to work properly. This section will look at different ways these information blocks can be exchanged in the system. In particular, it will be investigated how the static TFB tree can be integrated with the dynamic measurements (and TSI/TCI data) in a hierarchical system as the one outlined in **Error! Reference source not found.**.

Three different ways of implementing the system is discussed in the following sections and real implementations may use a combination of one or more of these principles on different layers of the system.

5.7.1 Off-line or on-line system

The approach currently used in the TOCC concept (see section 3.1) is to do the TCI calculation and aggregation fully in an off line system. This is illustrated in Figure 18.

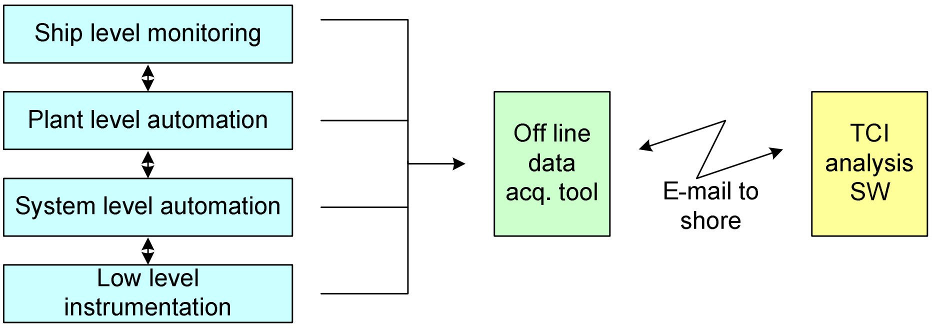


Figure 18 – Off line aggregation

In this system, input data (alerts, control and status as well as manual readings) is collected by an on board data acquisition tool and sent as a batch file (in XML format) to shore. There the TCI analysis tool reads all input data into a database, uses the already stored TFB and calculates the relevant TCI values.

This system could also be implemented as an on-line system, but using the same architecture. This was done in the DSS_DC project and is documented in [DSS06] and [DSS07]. In this case, the acquisition tool was an on-line Java application that directly interfaced to the TCI analysis software.

Other variants of the basic principle can be described in two bullets:

- The data is collected from the automation systems, possibly on different levels and sent to a special software system.
- The TFB tree is fully contained inside the destination software system and used to calculate all TCI and/or TSI within the software system.

5.7.2 As hierarchical system

Another approach that is probably better suited to TSI aggregation is a hierarchical system as exemplified in Figure 19. Here, the TSI aggregation is built into each layer of the automation system, but limited to the part of the tree that the automation system level controls.

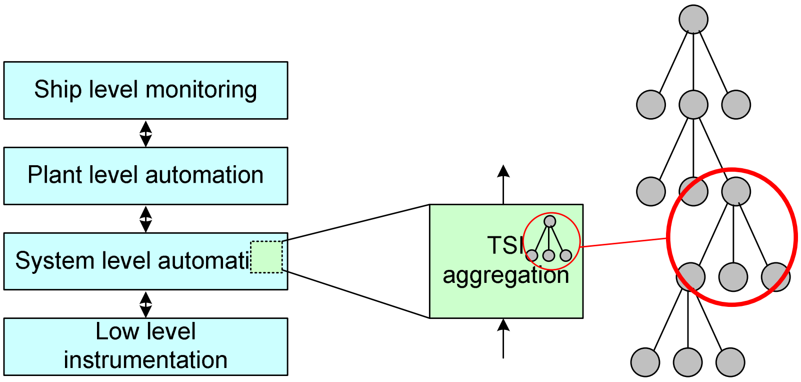


Figure 19 – Partly distributed TSI aggregation

One has two options for sending data into and out of the TSI aggregation module:

1. One can only send the TSI values and only let each level know about its own hierarchy in terms of knowing where the root cause for a problem is.
2. One can also let the lower level TFB be available to the modules so that the user can get direct information about where an indicated problem stems from.

Both approaches are possible and sensible and they can also be implemented in various ways, e.g., by sending the TFB as dynamic data up through the hierarchy, only sending the part of the TFB that represents a problem propagation or to statically configure layers in the system (possibly only the top most layer) to know all relevant lower level TFB.

5.7.3 As fully distributed system

The final example is described in Annex C and is a system that completely distributes the aggregation mechanism, also inside one automation layer. This is probably a mostly theoretical approach as current automation systems typically are fairly layered and this principle would not fit that model. However, with intelligent units, field buses and other technology one could perhaps envisage such an approach in the future. This line of reasoning will not be developed further in this particular document.

5.7.4 Event triggered or periodic recalculation

Also, when one considers the implementation possibilities, one need to recognize the difference between a system that recalculates indexes when anything changes (event triggered) or a system that does the recalculation based on a periodic timer.

In general one can say that periodic timers, with very long periods – weeks or months, is normally appropriate for TCI calculation while time triggered approaches are appropriate for TSI. The reason, obviously enough, is that TSI reflect a rapidly time varying state that need to be reassessed when anything significant changes in the system. The TCI, on the other hand, represent a very slowly varying technical condition that has no reason to be reassessed more rapidly than minimum half the time constant of the changes one wants to observe (according to the sampling theorem). More rapid sampling will not add any benefit to the analysis and may introduce artefacts.

This principle is also apparent in the current implementations of the TFB-like systems. The TOCC system uses a reporting and recalculation rate of about a month while the DSS_DC status assessment module runs continuously and samples data at one-minute intervals. In the case of the DSS_DC, one minute was sufficient, but in the general case one should normally recalculate TSI as soon as there is any significant change in input values, i.e., it should be event triggered.

5.8 Voyage phases and ship modes

One needs to consider the concept of voyage phases and or ship modes as moderator for the presentation of TSI information to the user. As the voyage phases will have significant impact on importance of various functions it should also impact, e.g., the priority of TSI data.

One example is a diesel electric propulsion plant where the overall technical capability of the plant may be presented as a TSI, but where minimum requirements will depend heavily on the voyage mode. During narrow water manoeuvring one will need more reserve power being readily available than in open sea.

However, currently it is suggested that the TSI does not use the voyage phase as input parameter, but rather uses this information when the importance of the TSI is presented.

6 Information structure for TFB and TSI

This section will look at the required information content of the different elements discussed in this document. The presentation will focus on various elements of the high-level data model shown in Figure 10. The corresponding XSD file is listed in Annex B.

6.1 Tree and node identities

The node identity (NodeID) is a text string that gives an unambiguous name to each node in the TFB tree. The node ID of the topmost node identifies the tree itself. All nodes will have an additional free text description to allow the use of more human readable names or descriptions. Again, the topmost node describes the tree. There is no need to enforce any special restrictions on the string format and software-making use of the node identity should be prepared to accept any form of string.

One major issue with the node is that it often will be associated with a physical piece of equipment, and there may be a need to represent this association. This is an optional attribute in the technical node called Tag. If the component or assembly is tagged, one will find that the same component can have at least the following different types of tags:

- *Yard tag:* The tag allocated by the yard to keep track of the physical components installed on a ship. This may, e.g., include information about the position (deck, main vertical zone) and type of equipment (pump, detector, etc.).
- *Ship tag:* The owner may develop a separate tag scheme to support the processes on the ship. Thus, a damper that closes airflow in a certain fire zone may be labelled with a corresponding tag although it is actually located in another fire zone.
- *Automation tag:* Automation system will often tag measurement values related to what electronic component of the system it belongs to. So, if the remote control unit that monitors the above damper is located in another fire zone than the damper itself, the automation tag may reflect that.

There may also be other tag schemes. Typical ships normally have four or five different tag schemes. The ones presented above are the most commonly used. One example of an additional tag scheme is an automation system that may give an additional tag to an object dependent on its connectivity to the automation system network structure.

The TagType entity will have the possibility to support any of these tag types and each node can, if necessary, be associated with several tags of different types.

In addition, it is also necessary to identify the system from which the tag is collected. This can be done through a code (if available) or just a free text name of the external system. Using a code will make it possible to automatically merge trees together.

The top node contains flags to indicate if the tree contains TSI and/or TCI type information. A number of Role flags can be added for this purpose. The list of roles can in principle be extended to also allow, e.g., environmental or safety indexes to be included.

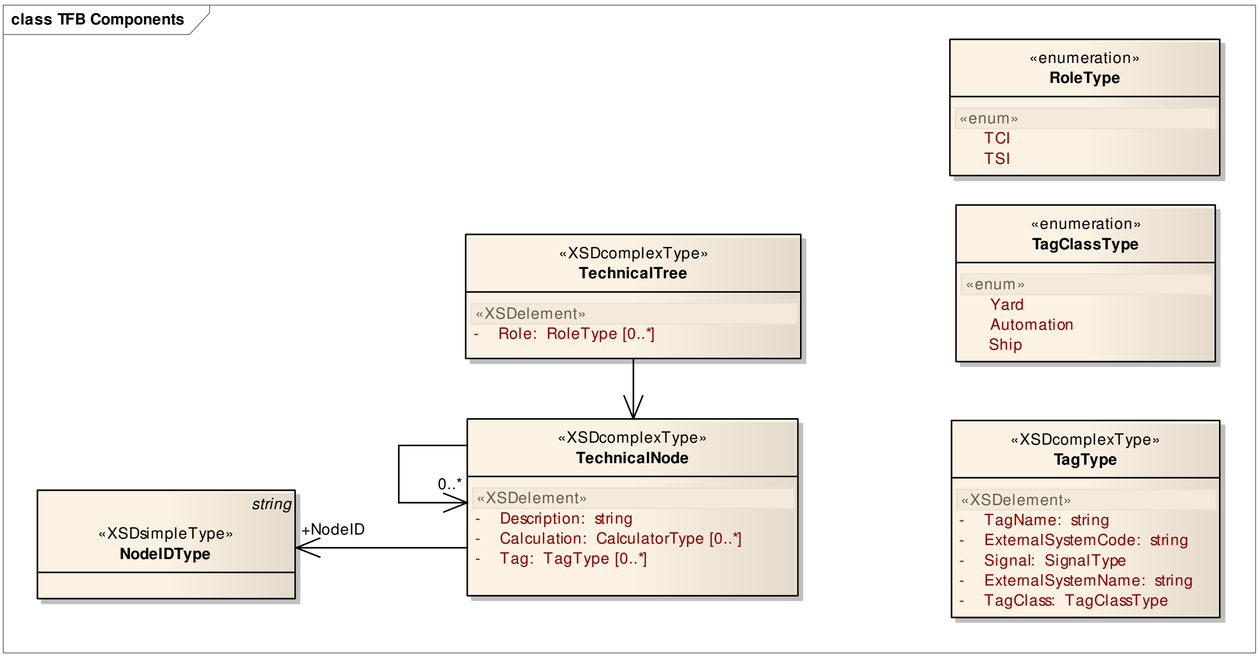


Figure 20 – Identity tags

Figure 20 shows how this information is structured when using the UML notation. The tree has a specific entry that points to the top most nodes in the tree. Attributes in the top node apply to the whole tree. The tree entity also says if the tree can calculate TCI, TSI or other types of indexes through the Role flags. Note that this determines if the tree can be used. Individual nodes may specify that they have, e.g., TCI capabilities although the tree may not have. In this case, one should normally not try to calculate TCI.

The tree itself is a created by letting each node include a number of lower level nodes and so on.

Each node has an identity as discussed above as well as a free text description. It contains a calculator element, which contains the role codes (TCI, TSI and/or other indexes) as well as information related to formulas etc. This is discussed in the next section. Finally, the node also may have one or more references to external tags describing the physical unit the node corresponds to.

The tag element contains the tag name and a code for the external system in question. It also has room for a free text description. The tag class is coded as discussed above and the tag is given a type that is not relevant in this context. The type is described in the next section and is related to how it can be used in formulas.

6.2 TBF node

Each tree node is structured as shown in Figure 21. Again, the notation is UML.

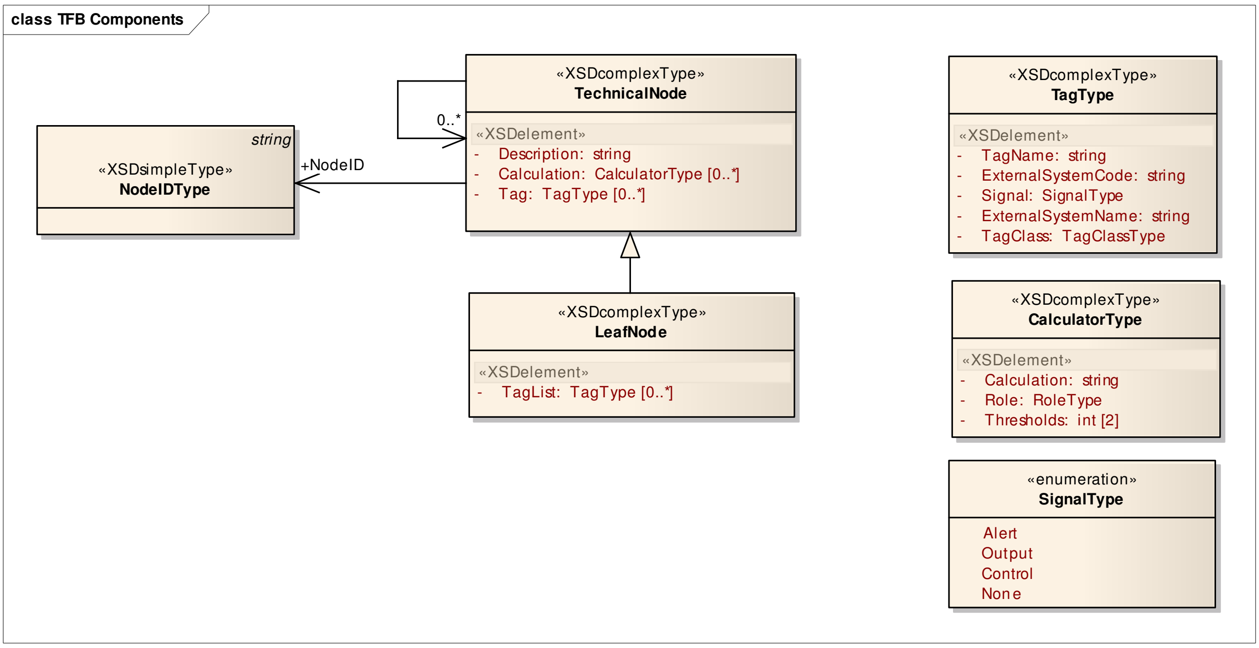


Figure 21 – The TFB tree node

Most of the node attributes were discussed in the previous section. They created the tree structure and gave references to external physical components the nodes were related to.

The calculator element specifies how the TCI, TSI or other index can be calculated. The node allows from zero and up calculator elements. Each element shall have a role flag specifying if it is used for TCI, TSI or other type of index calculation. In addition it has a string containing the actual formula or algorithm. The format of this string is discussed in the next section. Finally, the calculator node also has two thresholds. These are used to determine when the corresponding index value shall be interpreted as normal/green (below or equal to threshold 0), abnormal/yellow (below threshold 1) or dangerous/red (above or equal to threshold 1). For binary TSI components, one would expect these values to be respectively 1 and 100

For leaf nodes an additional list of tags can be added. The tag structure was discussed in the previous section. These tags are those referenced in the calculation of the initial index value as discussed in the next section. Note that the tags can be classified according to signal type, i.e., alert, output, control or none. None will typically be used for tags that point to physical entities that are not necessarily directly monitored. This is the case for the tag references embedded in the tree nodes.

6.3 Representation of the index calculation formula

The index calculation formula can be represented in many ways. Below is a slightly modified listing from another project that implements a similar strategy. The code is java like and is embedded in a CDATA XML element. The code is in the BeanShell[10] scripting language with is very similar to java.

---

[10] http://www.beanshell.org/

```
<Calculation>
    <![CDATA[
    int CalculateTCI(tags, nodes) {
      double totRes = 0;

      for (tag: tags) {
          totRes += tag[0].getDoubleValue();
        }
        return totRes;
    }
    ]]>
</Calculation>
```

Without going into details, the procedure gets two set of input values, lower level nodes or any tags listed in a leaf node. One of these structures should normally be empty, dependent on the node being a leaf node or a proper tree node.

Then various classes are available to retrieve values from each of the list and do the necessary calculations. The final result is returned as an integer to the higher level processing system.

Other structures can easily be accommodated. For systems written in, e.g., C or C++, one would probably select other scripting languages.

6.4 On the portability of the calculation formula

With reference to section 5.7 (practical implementation), any of the three approaches to implementation will allow implementation specific formats for the calculation formula. This is also important because the formula or rather set of formulas may also include intellectual property rights (IPR) that needs to be protected.

One exception to the above observation is when a sub-tree for some reason is exported from one manufacturer to another for the purpose to build a new calculation tree. In this case the manufacturers need to agree on the formula format. See also section 6.6.

6.5 TSI and TCI value

The TSI and TCI values are represented as separate elements related to the corresponding TFB node with a NodeID attribute. This is illustrated in Figure 22. This allows the system to exchange index values without sending the full tree structure.

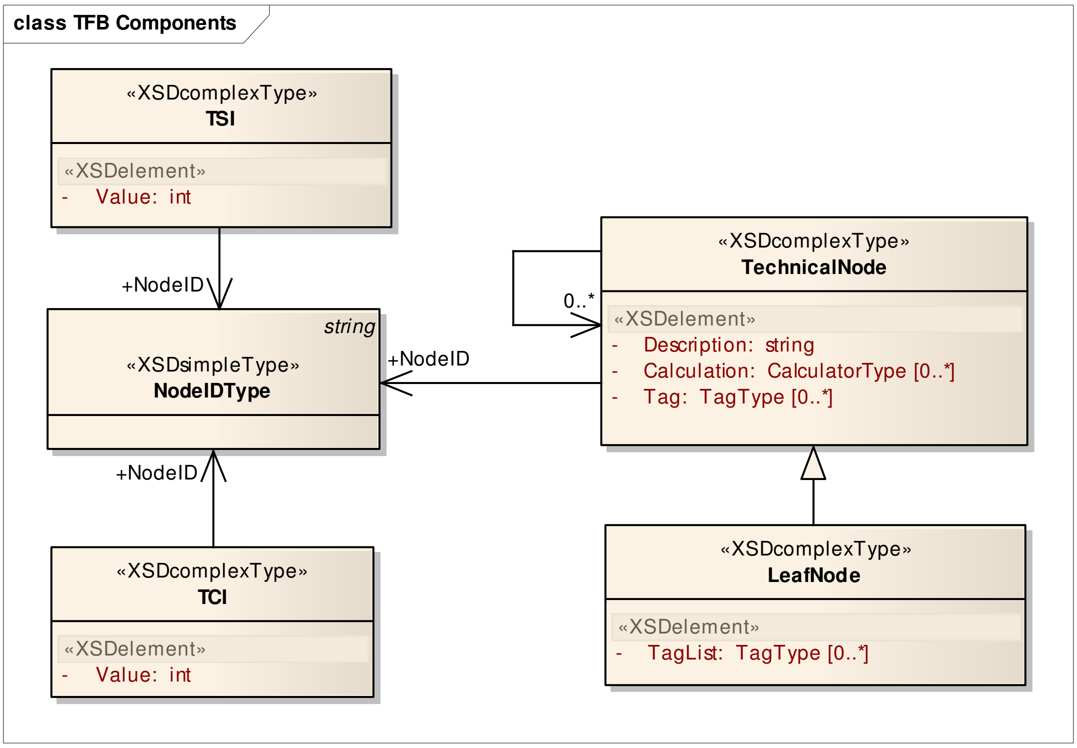


Figure 22 – TSI and TCI elements

Other index values can be added in the same way.

6.6 Export of TSI and TCI values and trees

There are three ways to export TSI, TCI and corresponding tree structures:

1. One can just send the actual index value with the corresponding node identity attribute. This works well within one system, but one needs to add a system identifier if it is exported out of the system.
2. One can send a simplified tree structure with the corresponding index value. The simplified structure would not require tag lists (leaf node information), nor calculator information. This would allow the receiving system to recreate the graphical representation of the tree and all warnings associated with it, but it would not be possible to recalculate new index values based on, e.g., an "what if" type functionality.
3. Finally, one can obviously exchange all information. This would allow recalculating index values and in general manipulation of the tree.

The XML schema defined allows all three methods. Data that is not included in method 2 has cardinality from zero and up and can be omitted from the XML file.

6.7 Alert and measurement structure

For a simple implementation of the indexing scheme as presented here and implemented according to the method described in 5.7.1 or 5.7.2, there are no specific requirements to alerts nor to general measurement data.

The leaf node calculation algorithm will require some formal definitions regarding representation and tagging of data, but that can be implemented on a per system basis.

If one wants to add index information to an alert, the easiest way is to attach the corresponding equipment tag to the alert and then reverse index from the list of tags associated to nodes in the TFB tree. It would be possible to include node identity information in the alert, but that would be fairly costly in terms of engineering, at least until a standard for representation of the relationship between TFB nodes and physical equipment is established.

7 Conclusions

To paraphrase the text from the scope section, the purpose of this report is to describe how to make technical information (fault reports, alarms, and status information) available from the technical systems to the technical management systems – on ship and on shore. This section discusses the main concepts developed and documented in the report and how these concepts support the stated purpose.

7.1 General ship system hierarchy

This report has discussed the issue of developing a standard hierarchical ship system model and has briefly described a few initiatives in this direction.

It is obvious that such a model, e.g., based on the SFI or DNV system (see 3.5 and 3.6) can have a significant benefit, particularly in the life cycle management of ships. The model would allow a common framework for structuring information about the ship and its components. Information produced during the design of the ship can be used for automatic configuration of both control systems and maintenance systems as discussed in the Shipdex initiative. The information will also be important to ensure safe scrapping of ships.

However, development of such models requires a significant amount of development and standardisation. The SFI system has a long story and much effort behind it. One can also assume that the newer DNV model has been costly to develop.

The D1 partners are of the opinion that the DNV model is most appropriate for further consideration, as it is completely open and not protected by copyright. This makes it economically attractive as well as more interesting also from an academic perspective. It remains to be seen how well DNV will respond to change requests, but it is the recommendation that this model is followed up in Flagship. This will be done also in the D1 sub-project in the area of network standardisation.

It should also be pointed out that the reference model most likely will be inappropriate as basis for detailed modelling of technical systems as suggested for the low level TCI or TSI hierarchies. It is very difficult, if not impossible, to generalize technical systems to such a degree that it is applicable to all ships and variants of each technical system.

7.2 TCI information structure

The data structures developed for the TSI and TFB and the corresponding XML schemas can more or less unchanged be used by the TeCoMan tool used to implement the TCI system in sub-project A2. Implementation of a standardised information structure allows easier configuration of the tool as well as the development of generic sub-system models.

Thus, it is recommended that the results from this work be implemented in TeCoMan. This will at least partly be done in the scope of sub-project A2.

However, as pointed out in the previous section, this model cannot be based on the standard technical hierarchies discussed there.

7.3 The TFB and TSI concept

The concept of the TFB is, as mentioned above, applicable to the development of TCI description structures. However, also as pointed out in the report, it is not trivial to capture both technical, functional and control/monitoring aspects in one relatively simple hierarchical model. In fact, it is not clear that this is at all possible. Investigations so far – see e.g., the example in section 4.7 – points to some of the complexity in this idea.

Thus, it is recommended that the concept of TFB be investigated further before it is being implemented in any other subproject in Flagship.

7.4 Usage on alert management

The ability to map technically oriented alerts to functional consequences will be critical to a more efficient management of ship system alerts. In this perspective, both the TFB and the TSI has very interesting possibilities. However, the problem with the TFB as it stands now is the problem of capturing both functional and technical properties in one hierarchical system. It may, e.g., be necessary to restructure the idea of the TFB to capture high-level functionality first and technical implementations thereafter.

One may also argue that a hierarchy is not the best way to represent the technical function block, but this argument is, at least partly, offset by the need to make the system easy to use and understand as well as the need to make it general. If the system is to be applied to a wide range of ships, it is necessary that it is generally applicable and that it does not require very detailed engineering for each ship or even ship type.

In spite of this, it should be clear that the general principles of a hierarchical system breakdown, links from technical alerts to this system and then assessment of functional impact is important.

Subproject B4 needs to investigate this issue as it has the expertise on alert management. From this report one can suggest that the general functional breakdown (most likely the DNV hierarchy) is used as basis for this work in B4.

7.5 The maintenance message standard

One of the main tasks of D1 was to develop message standard for integrating technical information from the online ship systems with the on board maintenance functions. The work on the TFB started with this premise. As it turned out, the Shipdex initiative (see section 3.9) was to some degree a competitor to the suggestions developed by Flagship sub-project D1.

However, Shipdex does not have a very firm foundation in the shipping market and it is, at least for the moment, advocating the use of the SFI system as the architectural reference model. SFI is a proprietary model and this may make it difficult to get general acceptance of the idea.

Also, the complexity of the S1000D model may be an obstacle to general implementation. However, the complexity is more caused by the wide range of the S1000D applications rather than by the system itself. Thus, it should be possible to pick out the components of S1000D that are most applicable to maintenance related messages. This is, at least partly what Shipdex is doing.

It is recommended to follow the Shipdex initiative closely and see how this can be used in the Flagship context.

7.6 Impact on protocol standards

D1 has had an explicit activity towards IEC TC80/WG6 on the development of new protocol standards for alert management. Part of this work is documented in Annex E and further results will be forthcoming in the next deliverable from D1.

8 References

[DSS05] DSS_DC Deliverable 3.1, detailed specification of total system, interfaces and requirements to the various modules, EC Contract TCT3-CT-2003-506354.

[DSS06] DSS_DC Deliverable D08.1c, Software documentation for DSS_DC status assessment module, TCT3-CT-2003-506354.

[DSS07] DSS_DC Deliverable D8.1b, TCM Software Documentation, EC Contract TCT3-CT-2003-506354.

[FSAP07] The FSAP/NuSMV-SA Safety Analysis Platform, International Journal on Software Tools for Technology Transfer (STTT), Volume 9, Issue 1 (February 2007).

[IEC 61158] Digital data communications for measurement and control - Fieldbus for use in industrial control systems.

[IEC61162-1] Maritime navigation and radiocommunication equipment and systems - Digital interfaces. Part 1: Single talker and multiple listeners.

[IEC61162-4] IEC 61162-4 series of standards: Maritime navigation and radiocommunication equipment and systems - Digital interfaces. Part 400: Multiple talkers and multiple listeners - Ship systems interconnection.

[IEC80/506/NP] Maritime navigation and radiocommunication equipment and systems - Digital interfaces - Part XXX: Multiple talkers and multiple listeners – Lightweight ship systems interconnection.

[MSC.64] Resolution MSC.64(67) - Adoption of New and Amended Performance Standards - (Adopted on 4 December 1996) - Annex 1 - Recommendation on Performance Standards for Integrated Bridge Systems (IBS).

[NAV 53/WP.2] Sub-Committee on Safety of Navigation, 53rd session, Agenda items 4, 7, 9, 11, 18 and 21A, NAV 53/WP.2, 26 July 2007, Navigational Aids and Related Matters - Report of the Technical Working Group.

[NR81] Fault Tree Handbook (NUREG-0492), United States Nuclear Regulatory Mission, January 1981.

[OPCDA] *OPC DA 2.05.1.0.1 specification*, OPC Foundation, 2002-06-28. See additional specifications at http://www.opcfoundation.org/.

[PRIAM] The EIAMUG project - intelligent actuation and measurements as users need them, Computing & Control Engineering Journal, December 1995. Volume 6, Issue 6.

[RFC3500] Request for Comments: 3550, July 2003, RTP: A Transport Protocol for Real-Time Applications.

[S1000D23] International specification for technical publications utilizing a common source database, S1000D, Issue 2.3, 2007-02-28.

[SX08] Digital Ship special report, Shipdex - providing ship technical information electronically, 2008.

[TAv34] Flagship Annex I to the contract (Technical Annex).

[TM99] TeCoMan User Manual ("TeCoMan brukermanual" – In Norwegian), Document 1999-HA5-NOT-014, Tom Anders Thorstensen, MARINTEK 1999.

[Vindøy08] Vindøy, Vidar. A Functionally Oriented Vessel Data Model Used as Basis for Classification, 7th International Conference on Computer and IT Applications in the Maritime Industries, COMPIT'08, Liège, 21-23 April 2008.

[Wikipedia] www.wikipedia.org.

[Xantic01] SFI group system, product Description, Version 05-2001, Xantic.

[Zadeh65] Zadeh, L.A. (1965). "Fuzzy sets", Information and Control 8 (3): 338-353.

9 Annexes

9.1 Annex A: TSI and alert related information exchanges

The concepts of TCI, TSI and alerts are closely intertwined as discussed elsewhere in this report. In addition to these, there is also the issue of maintenance orders that should be considered. Figure 23 shows how these information blocks can be used and exchanged.

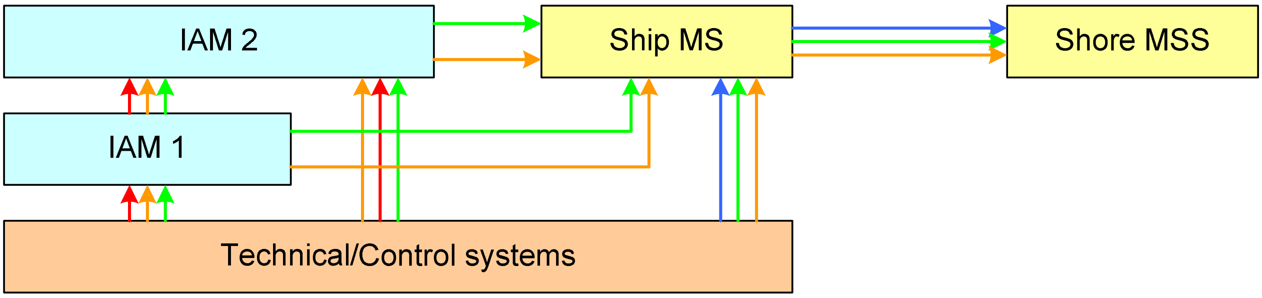


Figure 23 – General technical information flow

The technical systems are normally connected to some form of integrated alarm and monitoring (IAM) system. These systems will often be "hierarchical", e.g., the automation system can provide some alerts (red lines) to the integrated bridge alert management system as illustrated in the figure. This part of the system is discussed in more detail in subproject B4.

Some alerts may be associated with technical status information (TSI – orange lines) as discussed elsewhere in this document. Alerts may also be associated with maintenance instructions or other maintenance related data (green lines). These two types of information should also be sent to the people responsible for technical maintenance. On some ships this can be done through, e.g., the periodic maintenance systems (PMS), but on many ships there may not be an actual computer to take care of this. However, in the figure such a system is indicated as the "Ship Maintenance System" (Ship MS).

There may also be an exchange of information related to TCIs (blue lines). This information is also of interest to both the Ship MS and the shore side superintendent's maintenance and supervision system (Shore MSS),

The rest of this chapter will discuss these information flows and suggest suitable interface standards for them.

9.1.1 Alert information

General open interfaces for alert transmission must most likely be based on the [IEC 61162-1] interface standard. It is a fairly old-fashioned text based message system using serial lines between sender and receiver. However, the telegrams are fairly well defined and in extensive use. Also, it is almost certain that the new light-weight Ethernet variant of this standard will use the same message structure, but send messages on a local area network instead of on serial lines.

Today, this standard has one "official" mechanism for transmission of alert information. This is included in Annex E. However, this is a relatively simplistic way to transmit such information and is not appropriate for future control system with a higher degree of integration.

Some of the problems with the current approach that is already known and are being addressed in subproject B4 and in part two of subproject D1 are briefly described below. Updates on the protocol will be published in future B4 and D1 deliverables.

9.1.1.1 More complete functional breakdown

Alert management functions are actually divided into a set of more or less distinct steps. An overview is shown in Figure 24.

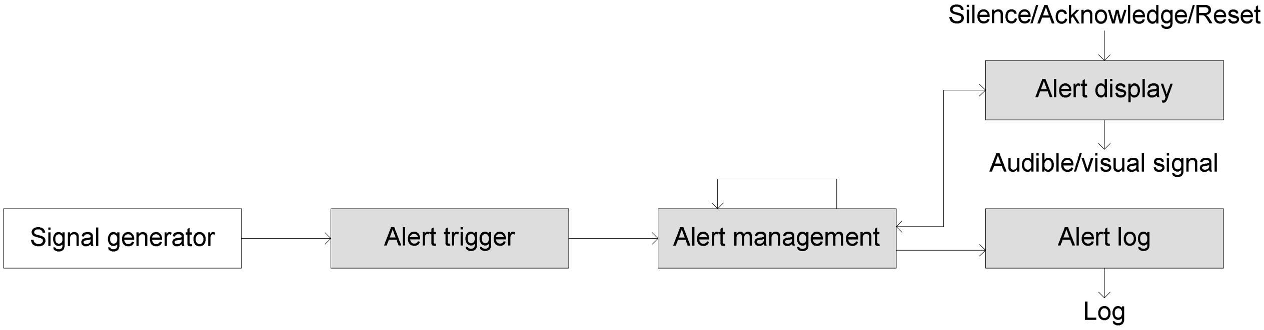


Figure 24 – Alert management functions

The basic functions are:

- *Signal generator:* This is normally outside the alert management system, but it illustrates that the alert signal exists as some form of signal or other information before some function decides that it is an alert.
- *Alert trigger:* A function to decide if a certain logical combination of signals and conditions is to be considered as an alert.
- *Alert management:* Functions associated with maintaining the state of an alert, i.e., latch it so that it is remembered, take it through states unacknowledged, acknowledged and reset and other related functions. Note that a system may contain several such functions.
- *Alert display:* This is the conversion of the alert signal to visual or audible signals as well as provision of methods for silencing, acknowledging or resetting the alert.
- *Alert log:* Storing information about alerts and alert states in a permanent log.

Note that these functions can be combined in different ways into one or more modules. The functional breakdown presented here is necessary to provide consistent protocols for alert management and information transmission.

9.1.1.2 Additional information in alert message

The alert message will normally also need additional data compared to what the current IEC proposal contains. Some elements that have been mentioned, e.g., in the technical annex are:

- *Time tag:* information about when the alert was first triggered. This may also contain information about acknowledgement or other important state changes.
- *Instance code:* Some alerts may reoccur frequently. In such cases it may be necessary to have an instance counter so that two subsequent alerts can be distinguished between. This may in most cases be implemented by the time tag.
- *Severity:* It is as a minimum necessary to distinguish between warnings and alarms. It may be coded into the alert number, but that requires specific knowledge about the alert in the display function.
- *Technical tag:* If an alert is related to a technical problem, it is in most cases necessary to link the alert to the relevant technical component. This can provide a link to the TSI as well as to any other technical maintenance information that can be associated with the alert.

9.1.1.3 Additional messages

The current proposal as shown in Annex E may fail for some scenarios where messages are lost and where there are more than one alert management function in the system. There should either be some form of connection management between sender and receiver or some additional messages to cater for more controlled changes in states.

9.1.2 TSI data

TSI data structure has been discussed in Chapter 6 and this discussion will not be repeated here. The conclusion in section 5.7 was that it is probably better to do the TSI aggregation in a centralized manner and this means that it is not so important to create open standards for TSI transmission.

However, the data structures presented in Chapter 6 and listed in annex B can be used. In that case, one would create a new message as indicated below, just containing the TSI value and the node identity.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="TSI" type="TSI" minOccurs="1" maxOccurs="1"/>
  <xs:complexType name="TSI">
    <xs:sequence>
      <xs:element name="Value" type="xs:int" minOccurs="1" maxOccurs="1"/>
      <xs:element name="NodeID" type="NodeIDType" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="NodeIDType">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
</xs:schema>
```

The node ID needs to be standardised in the system. Further more, it needs to be consistent with the tag code used in the alert message (see previous section).

9.1.3 TCI data

The TCI data is most likely only applicable inside a TCI aggregation system as discussed previously in this report. Thus, it should not be necessary to specify message formats for transfer of TCI data. However, if for some reason one needs to send a TCI value, one can use the same principle as in for TSI and create a TCI message from the relevant portion of Annex B.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="TCI" type="TCI" minOccurs="1" maxOccurs="1"/>
  <xs:complexType name="TCI">
    <xs:sequence>
      <xs:element name="Value" type="xs:int" minOccurs="1" maxOccurs="1"/>
      <xs:element name="NodeID" type="NodeIDType" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="NodeIDType">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
</xs:schema>
```

As previously, the node ID needs to be standardised in the system. Further more, it needs to be consistent with the tag code used in the alert message (see previous section).

9.1.4 Technical maintenance data

The transmission of technical maintenance data may be highly relevant, particular where alert shelving or filtering is done. In these cases, one will normally have a technical problem that cannot immediately be solved and the alert is temporarily "disabled" while the engineers or electrician fixes the problem. It is obviously of great help if the information about the alert can be sent to the relevant personnel.

To transmit information about maintenance actions from the technical system to the management system, it is suggested to use an XML message as described below. This message could also be implemented as an IEC 61162-1 telegram, by inserting tag data into sentence fields instead.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="TechnicalMaintenance" type="TMType" minOccurs="1" maxOccurs="1"/>
  <xs:complexType name="TMType">
    <xs:sequence>
      <xs:element name="FaultCode" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="NodeID" type="NodeIDType" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="NodeIDType">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
</xs:schema>
```

The node ID is the general tag for the equipment in question and the fault code must be a unique code specified by the manufacturers and which can be linked to a maintenance procedure through, e.g., Shipdex descriptions of equipment.

9.2 Annex B: XSD listing

The following listing is the XSD files generated from Enterprise Architect, based on the UML design presented in this paper.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="TechnicalNode">
    <xs:sequence>
      <xs:element name="Description" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="Calculation" type="xs:string"
             minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="Tag" type="TagType" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="TechnicalNode" type="TechnicalNode"
             minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="NodeID" type="NodeIDType" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="SignalType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Alert"/>
      <xs:enumeration value="Output"/>
      <xs:enumeration value="Control"/>
      <xs:enumeration value="None"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="TechnicalTree" type="TechnicalTree"/>
  <xs:complexType name="TechnicalTree">
    <xs:sequence>
      <xs:element name="Role" type="RoleType" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="TechnicalNode" type="TechnicalNode"
             minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="TagType">
    <xs:sequence>
      <xs:element name="TagName" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="ExternalSystemCode" type="xs:string"
             minOccurs="1" maxOccurs="1"/>
      <xs:element name="Signal" type="SignalType" minOccurs="1" maxOccurs="1"/>
      <xs:element name="ExternalSystemName" type="xs:string"
             minOccurs="1" maxOccurs="1"/>
      <xs:element name="TagClass" type="TagClassType" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="TagClassType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Yard"/>
      <xs:enumeration value="Automation"/>
      <xs:enumeration value="Ship"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="TSI" type="TSI" minOccurs="0" maxOccurs="unbounded"/>
  <xs:complexType name="TSI">
    <xs:sequence>
      <xs:element name="Value" type="xs:int" minOccurs="1" maxOccurs="1"/>
      <xs:element name="NodeID" type="NodeIDType" minOccurs="1" maxOccurs="1"/>
```

```
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="NodeIDType">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
  <xs:element name="TCI" type="TCI" minOccurs="0" maxOccurs="unbounded"/>
  <xs:complexType name="TCI">
    <xs:sequence>
      <xs:element name="Value" type="xs:int" minOccurs="1" maxOccurs="1"/>
      <xs:element name="NodeID" type="NodeIDType" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="LeafNode">
    <xs:complexContent>
      <xs:extension base="TechnicalNode">
       <xs:sequence>
         <xs:element name="TagList" type="TagType" minOccurs="0" maxOccurs="unbounded"/>
       </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:simpleType name="RoleType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="TCI"/>
      <xs:enumeration value="TSI"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="CalculatorType">
    <xs:sequence>
      <xs:element name="Calculation" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="Role" type="RoleType" minOccurs="1" maxOccurs="1"/>
      <xs:element name="Thresholds" type="xs:int" minOccurs="2" maxOccurs="2"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

9.3 Annex C: Implementing a fully distributed TSI system

This annex gives a description of an alternate approach to the design of the TSI system, based on implementation in a fully distributed system. This is not the approach selected by Flagship, but the text is included here to provide some background on how this can be done.

9.3.1 Introduction

The Technical Condition Index that is needed for the DSS is built from events coming from ship systems. Between the initial event and the final TCI, attributes shall be added to the event to complete the initial data. These attributes could come from different sources. They could be internal of the system, from interfaced systems or linked to the state of the ship. So from a single value generated, for example by a pump, we need at the end complete information that provides a kind of “context” to the value.

As attributes of the value are filled by the system itself and other systems, it is necessary to define a data structure that is known by all the parts of the DSS.

9.3.2 System structure

Any system aboard ships is composed by elements that are not on the same level. The lower level could be considered to be the simplest, sensor, switch … and the higher the most complex, supervision, management, etc. We can also consider a hierarchic vision between the lowest and the higher. This vision could be represented by a pyramid with the sharp side located in two possible ways. The first way could be up and this means that all the lower information go up to the higher part of the system. Information is concentrated in the top of the system. The second way could be down and means that the value provided by the sensor needs to be improved with attributes from upper levels. This last way is very different from the first one. The way of thinking is different because in this case, the information itself is more important than its use.
It is of course this last vision that we consider for FLAGSHIP model.

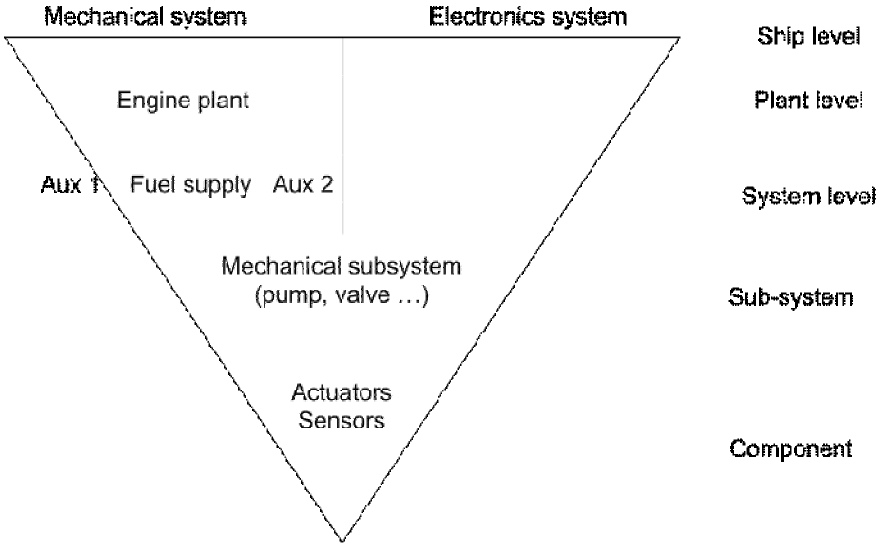


Figure 25 - System hierarchical description

Inside the system, each level will provide additional information to the initial value in the datagram. After, the datagram needs additional information from interfaced system.
It is important to notice that these internal attributes shall be standardized. Generally, it is only one supplier that manages all the items of a single system but we must keep in mind that this datagram could be use in other systems to fill attributes of their own datagram. The structure of the datagram shall be known by all the system in order to find easily the information needed.

9.3.3 Internal attributes

The internal attributes of a value are provided by the system itself. In an ideal world, all these attributes shall be standardized whatever the system. We think that it could be difficult. We could imagine some of them:

- Name of the system
- Normal range of the value
- Redundancy of the item that generate the value
- Abnormal mode
- Previous value
- …

When we analyze the potential attributes, the result is that some of them could be applicable to all the cases but others are specific. What we propose is to define “public” and “private” attributes. “Public” attributes shall be standard and have to be present in all the datagram describing internal attributes. The supplier of the system will define “Private” attributes. The supplier will have to provide a detail description of these attributes and provide it to other systems that need to be interfaced with him.

We consider that the following attributes are mandatory:

- ID of the system
- Criticism
- Type of alert
- Time and date
- Label / value

These are the bases for the mandatory attributes. Discussion with other SP could define new ones.

9.3.4 External attributes

These attributes could be of two types. They could be provided by other systems that interact with the first one, like HVAC and Fire detection, or could be provided by other means to be defined and provide general information on the ship, it operational activity and its vicinity. In the first case, the attributes could be:

- Status of the other systems (in use, damaged, …
- Exchange of data (datagram, audio, video)
- Etc.

In the second case, the attributes could be:

- Ship position (harbor, sea, maneuver, dry dock…)
- Ship status (Normal, Emergency, Abnormal, Evacuation…)
- Passengers location (excursion, dinner, room, …)
- Weather forecast
- Crew status (day, night, drill, …)

As for internal attributes, it is difficult to provide a standard structure. May be some are not available by now; maybe some are finally not interesting for ship Owner.

May be discussion with other WP will help to define the datagram with more details.

9.3.5 Operational modes

9.3.5.1 Introduction

The previous chapters are presenting the workflow of a datagram from the activator / detector of a system to the operator. In today ships, a dedicated supplier specialist provides each system in its system. This architecture is completely different of the operational use of the ship. The final user is managing process and not systems. As an example, the safety officer manage safety and not fire detection. It is the same for the security officer. But these two officers are using common systems like CCTV.

Their point of view is based on operational processes and their modes or status.

9.3.5.2 Processes

During the generation of a TSI, it is important to notice that the processes are an additional layer of information that could modify the status of values providing by systems and their interpretation by the operator. The processes are the level of information that is the most important in the final decision regarding what is, at the beginning, only a modification of status of a sensor.

9.3.6 Technical proposition

9.3.6.1 Introduction

The base of our proposition is made on the MIB structure used in computer network equipment. This structure is hierarchical, tree-structured, and entries are addressed through object identifiers. In the case of the MIB, the information is pulled by the supervision station and not pushed by the equipment like we propose to do in this project.

This architecture can’t be used in our project; the objects defined in a MIB are not interesting. However, the structure could help us. The MIB tree-structure could be reused for the TSI. It is a very simple structure that could be easily implemented in all the systems. For the description of the objects themselves, the MIB is in fact defined by a METADATA registry that is in one part standardised and on another part customised for proprietary use.

The principle of a METADATA registry could be a solution in our case. It is independent of the final language that will be chosen for the demonstrator and could be discussed with the other WP later. Some examples are provided in annex.

9.3.6.2 Structure

In this chapter we make a proposition for the organisation of the data. It is based on a tree structure that identifies at each layer a special a group of similar information.

If we consider that the first value of the datagram is the ID of the system, the first step for the design of the data should be to split the first layer of the tree regarding the source of the data. So we propose these three groups for the first branch of the tree:

- Internal data
- External data
- Processes

For the lower branch of the tree, we don’t want to have too much layer. It is too much complicated to manage and in fact, in a system, the initial data generated by the sensor is often aggregated directly in the supervision and not on the lower layer. The constraint is that we could have many branches on one level.

The second layer of the tree will depend of the upper branch. For the “Internal data” branch this layer will be the list of all sources of data that could provide the system. The question is do we consider the sub-system or do we consider the component for this layer? For the “External data” branch, this layer will be the list of sources of data of systems that have interactions with the component / sub-system emitting the datagram. For the “Processes” branch, this layer will be the list of information coming from processes using the component / sub-system emitting the datagram.

The third layer depends of the second layer. For “internal data” and “processes”, it is the list of the values provided by the sensor / detector / activator. For “external data” the structure is now the same as “internal data” branch, starting at the second layer.

For each layer, the simple principle is to associate a label to a value. For example, “ID=FIRE_DETECTION” or “GROUP:INTERNAL_DATA”. The detail and final sentence will be defined by the protocol choosen.

The list of all the labels is difficult to define in this document. Labels depend of the systems and the information we want to raise up to the operator. On the other hand, the labels of the same system from several suppliers shall be the same. In this frame the proposition is to define a kind of standard designed commonly by all the supplier of the same kind of system.

9.3.6.3 Example

The following table represents how the information could be organised.

System ID	Group/Internal data	Sub-system1	Data 1	
			Data 2	
			…	
			Data X	
		Sub-system2	Data 1	
			Data 2	
			…	
			Data X	
		…	…	
		Sub-system X	Data 1	
			Data 2	
			…	
			Data X	
	Group/External data	System ID 1	Sub-system1	Data 1
				Data 2
				…
				Data X
			…	…
			Sub-system X	Data 1
				Data 2
				…
				Data X
		…	…	...
		System ID X	Sub-system1	Data1
	Group/Processes	Process 1	Data 1	
			Data 2	
			…	
			Data X	
		…	…	
		Process X	Data 1	
			Data 2	
			…	
			Data X	

We can also consider for “external data” to merge “System ID X” and sub-system. Normally, the datagram should not need too much information from other systems.

This also permits to keep the same number of layers for the three main groups.
In this representation, we don't consider the tools to implement it. May be protocols are able to improve this design.

9.4 Annex D: DNV Function list

This is an extract of the first two levels of the DNV function list. It consists of 95 elements. The third level elements are about 590 and the complete list is somewhat more than 2500. The list is for information only. The reader should get the up to date and authoritative list from DNV directly.

Table 1 – DNV Function list (Level 2)

000a	General	
	000	Administration general
	010	Services
	020	Economy
	030	Reference requirements
	040	Planning and reporting
	050	Certificates
	060	Quality assurance
	070	Technical information
	080	Vessel operation
	090	General, other
100a	Main structure	
	100	Main structure general
	110	Ship structure
	120	Column-stabilised unit structure
	130	Self-elevating unit structure
	140	Tension leg installation structure
	150	Deep draught installation structure
	190	Other structures
200a	Stability, watertight and weather tight integrity	
	210	Stability
	220	External watertight and weather tight integrity
	230	Internal watertight integrity
	290	Stability, watertight and weather tight integrity, other
300a	Hull equipment	
	310	Access
	320	Anchoring, mooring and towing
	330	Lifting
	340	Special purpose functions
	390	Hull equipment, other
400a	Propulsion and steering	
	400	Propulsion and steering general
	410	Conventional propulsion
	420	Conventional steering
	430	Thruster propulsion and steering
	440	Manoeuvring
	450	Position keeping
	460	Motion and trim control
	490	Propulsion and steering, other
500a	Electric power	

	500	Electric power general
	510	Main electric power
	520	Emergency electric power
	530	Cabling
	540	Earthing
	590	Electric power, other
600a	Machinery and marine piping systems	
	600	Machinery and marine piping systems general
	610	Steam and thermal oil generation and distribution
	620	Fuel and lubrication oil storage and distribution
	630	Sea and fresh water systems
	640	Compressed air generation and distribution
	650	Bilge handling, ballasting and drain
	660	Discharge and disposal
	670	Air, sounding and overflow systems
	690	Machinery and marine piping systems, other
700a	Navigation, communication and control	
	700	Navigation, communication and control general
	710	Navigation
	720	Collision and grounding avoidance
	730	External communication
	740	Internal communication
	750	Vessel control and monitoring
	790	Navigation, communication and control, other
800a	Safety	
	800	Safety, general
	810	Fire prevention
	820	Structural fire protection
	830	Fire and gas detection and alarm
	840	Fire fighting
	850	Escape
	860	Life-saving
	890	Safety, other
900a	Environment	
	910	Ventilation
	920	On board environment
	930	Working environment
	940	External environment
	950	Pollution prevention
	960	Service facilities
	970	Accommodation
	990	Environment, other
1000a	Dry cargo	
	1000	Dry cargo general
	1010	Dry cargo loading and unloading
	1020	Dry cargo storing
	1030	Dry cargo support systems
	1040	Cargo securing
	1090	Dry cargo, other

1100a	Liquid and gas cargo	
	1100	Liquid and gas cargo general
	1110	Liquid and gas cargo loading and unloading
	1120	Liquid and gas cargo containment
	1130	Liquid and gas cargo support systems
	1140	Oil and gas production process
	1150	Oil and gas production process support
	1190	Liquid and gas cargo, other
1200a	Drilling and well intervention	
	1200	Drilling and well intervention general
	1210	Drilling structures
	1220	Well control
	1230	Heave compensation and tensioning
	1240	Hoisting and rotating
	1250	Blowout preventer and tubular handling
	1260	Bulk storage, drilling fluid mixing and circulation, and cementing
	1270	Well testing
	1280	Well intervention
	1290	Drilling and well intervention, other
1300a	Diving	
	1300	Diving general
	1390	Diving, other

9.5 Annex E: IEC Alert recommendation

The below text is the basis for 80/520/INF, an internal IEC document describing a suitable way to transmit alert and acknowledgement information over [IEC 61162-1]. This document has in substantial parts received contributions from Flagship subproject D1.

9.5.1 Definitions

Alert – Term used for Alarm, Warnings or Cautions. Ref. IMO Res. MSC.283 (83). An Alert may be either active or inactive. If active, it may be either acknowledged or unacknowledged. The inactive condition represents normal operation.

Active alert – Alerts, which have not been acknowledged, and alerts with alarm condition active (A), which have been acknowledged, but still indicates an abnormal condition.

External Alert Management Device – (EAMD a device that supports handling of alerts reported to it from connected alert sources such as Sensor Devices (SD). The EAMD displays alerts, and may provide remote means to manually or automatically acknowledge alerts.

Alert number – The identification (3 digits) of one particular alert as defined in IEC 61162-1 for the ALR sentence. This document will implicitly limit the meaning of *alert number* to an alert of a defined type.

Alert type – An alert related to one specific physical occurrence.

Event – Any change of alert/acknowledgment status of a device that generates the transmission of an ALR message

Clearance – From the active state an alert is cleared when the underlying cause is no longer present AND it has been acknowledged.

Sensor device – (SD) an instrumentation type device with relatively limited functions with respect to alert management. These functions are “known” by the External Alert Management Device (EAMD), in the sense that the latter can reliably understand the meaning of a specific *alert number*.

Alert message – ALR sentence describing a single alert condition of the SD

Alert-list message – A set of ALR sentences, preferably transmitted without time delay between them, describing the complete alert state of the SD.

Alert-list message interval – the time the equipment standard specifies as the maximum time between two consecutive transmissions of the alert-list message. The maximum interval is 60 seconds.

9.5.2 Background

Sentences for exchanging alerts and for acknowledging alerts have been specified in IEC 61162-1 since year 2000. However, detailed instructions and guidelines on how to use these sentences have been missing, causing a potential for differences in implementation between manufactures. Until now this has not been a significant problem, as these alarm and acknowledge related sentences have been used only to a limited extent. In future, however, these sentences will have to be used in conjunction with the revised IMO performance standards for Radar equipment (in its clause 8.3.4) as well as the two new IMO performance standards for Galileo receiver equipment (in its clause 3.18) and for Navigation lights and Navigation light controllers (in its clause 5.8) that require a bi-directional interface to transfer alarms to external systems and for receiving acknowledgement from external systems. This interface is required to be in compliance with IEC61162 series.

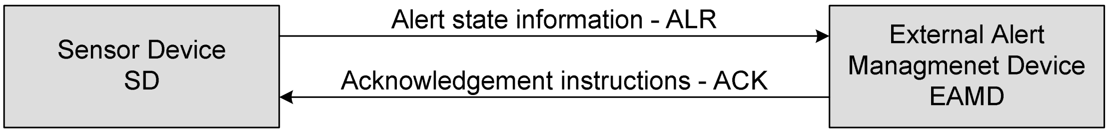
There is furthermore and at present an IMO correspondence group working on developing performance standards for Bridge Alert Management. This document may provide useful guidance to this correspondence group on capabilities and limitations of the alarm and acknowledge related sentences specified in the present standard IEC 61162-1 (2007).

9.5.3 Scope

This specification aims to avoid misunderstandings by giving guidance to manufacturers on how the ALR and ACK sentences should be used to transfer alert related information between a source of alerts, typically a sensor device (SD), and a display/acknowledgement unit (external alert management device, EAMD).

The ALR and ACK sentences have certain limitations that are discussed towards the end of the document. Thus, the specification focuses on how sensor devices (SDs) can exchange alert related information with an external alert management device (EAMD). The data connection should normally be bi-directional, but can also be used in a unidirectional mode for distributing information about alert states in the sensor device without giving any possibility for acknowledging alarms. In the unidirectional case, acknowledgement at the sensor device by direct means provided by the manufacturer (e.g. depress button) should cause the device to transmit at least one ALR message with the acknowledgement field ‘active’.

Currently, the ALR/ACK sentence definitions do not explicitly allow differentiation between silenced and acknowledged state. This document specifies rules and guidance for the acknowledgement process. A differentiation of the silence and acknowledged states may be accomplished either implicitly, e.g. by alarm number encoding, or with sentence extension.

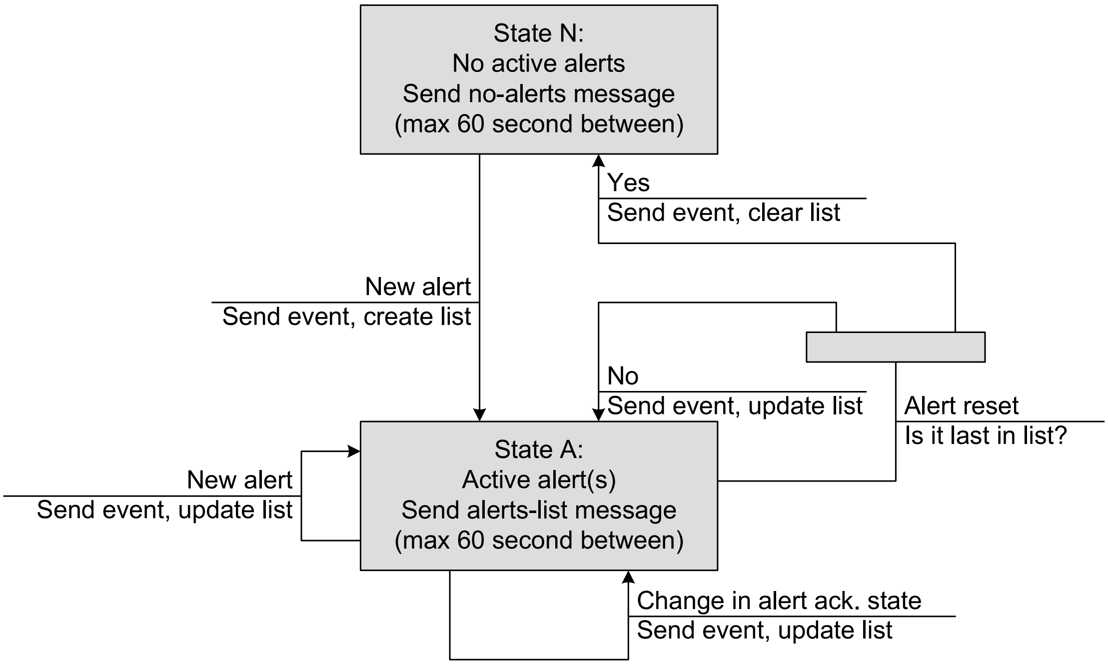


This specification relies on the EAMD having implicit knowledge of the internal functionality of the SD. Thus, these instructions will only cover a limited part of general alert management (see final section).

9.5.4 Alert information distribution

9.5.4.1 Main device states

The below state diagram shows the two main states the sensor device can be in with respect to alerts.



The sensor device has two main states:

- *State N:* No active alerts. The device should send a "no-alerts" message (see below) with an interval of maximum 60 seconds.
- *State A:* The device has one or more active alerts, of which zero or more may be acknowledged and the rest (possibly zero) are unacknowledged. In this state, the device shall send all active alerts at interval not to exceed 60 seconds (Note: The AIS standard specifies 30 seconds here). When multiple alerts are active in the SD, it is recommended to transmit all active alerts as "a list" of alerts (alert-list message).

In addition to the periodic transmissions as mentioned above, the SD shall immediately send an Alert message (ALR) to the EAMD, when (Values for alert condition and acknowledge state in parenthesis):

- A new alert is raised in the SD - (A,V)
- An existing alert is acknowledged in the SD (either on the SD itself or by remote acknowledgement from EAMD) – (A,A)
- An existing alert condition becomes non-active (V,V or V,A)).

The alert message may include the time stamp when the alert last changed status (normally current time) and include the alert number, explanatory text as well as appropriate alert and acknowledgement flags. It may optionally be followed by a TXT message to give additional contextual information. The TXT message should be contiguous with its associated ALR. An example is included below.

```
$--ALR,123456,906,A,V,Sensor fault*hh<CR><LF>
$--TXT,02,01,06,Selftest error 17*hh<CR><LF>
$--TXT,02,02,06,See service manual*hh<CR><LF>
```

Note: This specification does not put any restrictions on the transitions that are reported through an event message. Thus, receivers should be prepared to receive and process all possible combinations and sequences of alert state events.

9.5.4.2 No-alerts message

The *no-alerts* message is intended to inform the EAHD that the SD has no active alerts. It shall be sent at an interval not exceeding 60 seconds. This message may be used to clear the EAHD alert list.

This message is sent as an ALR message, but without time stamp, and shall include a ‘V’ flag in both the alert condition and acknowledgement field. The *no-alerts* (list empty) message is included below.

```
$--ALR,,,V,V,*hh
```

9.5.4.3 Alerts-list message

The alert/alert-list message is intended to periodically refresh the alert list so that the listener can verify that it has the correct internal list of active alerts. This will in turn help to remedy problems that may occur due to lost telegrams at earlier stage, synchronization of recently added receivers, etc.

The alert/ alert-list message shall be sent with an interval not to exceed 60 seconds (Note that the AIS performance standard specifies 30 seconds as maximum).

The alert/ alert-list message consists of the same message(s) sent when the corresponding event occurred, but all active alerts shall be reported, and preferably with no delay between messages. An example with two messages in the list is included below:

```
$--ALR,123456,123,A,A,Battery power in use*hh<CR><LF>
$--ALR,130507,456,A,V,Self test failure*hh<CR><LF>
```

Note: The time stamp will wrap around after 24 hours. For alerts that are active longer than 24 hours, the EAHD will need to keep track of the original event time.

9.5.5 Alert acknowledgement

9.5.5.1 General principles

If the alert management device has a bi-directional data link to the sensor device, it is possible to send remote acknowledgements to alerts. In principle, this can be implemented in two modes:

- *Single acknowledgement*: If each acknowledgement is based on user action, e.g., through an acknowledgement button, one can leave the resolution of lost messages to the user. The user should note that the acknowledgement was not effected and, if necessary, repeat the acknowledgement at the local or remote station.
- *Multiple acknowledgements*: The management device acknowledges the alert based on built in rules. In this case, it needs to consider if the acknowledgement was received or not by the sensor device. Due to the possibility of message loss in the IEC 61162-1 standard, it is possible that the acknowledgement or even the response (event) to the acknowledgement is lost.

Note: The latter case adds significant complexity. Example: One GPS lost gives a number of secondary alarms that are acknowledged at the same time based on one action by the user.

9.5.5.2 Alarm acknowledge capability

In some cases, the sensor device needs to know if the alert management device is still able to communicate with it. This may, e.g. be used to implement silent alerts on the sensor device.

In this case, it is necessary to send an alarm acknowledge message from the external alert management device (EAMD) to the sensor device (SD).

```
$--ACK,,*hh<CR><LF>
```

Note: This needs to be investigated further.

The message should be sent at an interval not to exceed 60 seconds.

The alert management device shall not send any messages, including heartbeat, if the heartbeat message from the sensor device has not been received in a period of maximum 130 seconds. This time shall be reduced appropriately if the maximum repetition time is shorter.

9.5.5.3 Single alert acknowledgement

If single alert acknowledgement is implemented, exactly one acknowledgement message shall be sent each time the operator requests an acknowledgement.

```
$--ACK,xxx*hh<CR><LF>
```

9.5.5.4 Multiple alert acknowledgement

If multiple alert acknowledgement is implemented, the acknowledge message may have to be repeated if an event message is not received immediately after the acknowledgement.

```
$--ACK,xxx*hh<CR><LF>
   | 10 second delay, if no new event has been received
$--ACK,xxx*hh<CR><LF>
   | 10 second delay, if no new event has been received
$--ACK,xxx*hh<CR><LF>
```

Note: The 10 second delay is inserted to avoid confusion when the SD is in the process of sending an alert-list message when the ACK is transmitted. In this case the EAMD may first receive an unacknowledged alert (after the ACK is sent) and then an acknowledged or cleared alert message at a later stage.

Note: No more than three attempts at acknowledgement should be done. Missing returned status massage will then normally indicate that the device is not able to acknowledge the alert.

9.5.6 Limitations in ALR and ACK that should be noted

The IEC 61162-1 description of the ALR sentence uses the term “threshold exceeded” in the notes. This should normally be read as “alarm condition active”. When a threshold is exceeded causing an alarm to be raised, and the signal later goes back to normal (i.e. threshold not exceeded), the alarm state should still be kept until the alarm is acknowledged. This means that the ALR sentence can only show alarm and acknowledgement states, not the underlying signal’s relationship to the alarm threshold (see also ALA).

In general, ALR has limited capabilities in terms of transferring detailed information about alerts. Limitations include to the transfer of information on:

- Multiple active alarms generated as a result of the same alert type.
- Alert priority (caution, warning, alarm), this has to be made explicit in the alert number.
- Alert category (category A, category B – see INS performance standards), this has to be made explicit in the alert number.
- The state of the physical event that originally caused the alert (threshold exceeded, threshold normal, etc.).

Dependent on application, other limitations will most likely also apply. However, for the purpose discussed in the scope section, it is believed that the specified examples represent best practice.