Abstract

A major trend in ship control is to interface all vital systems on board to make an integrated system. The goal is to make better system, e.g., increase safety and reduce the operator’s workload. Traditional ship control systems are not prepared for this integration. Most systems use point to point communication with proprietary protocols.

An alternative to the above approach is the “open” system where components share a network and the interface to the network. We believe that this approach has great benefits compared to proprietary systems, both for users and producers of such systems. Norwegian research authorities, a number of Norwegian ship automation suppliers, and SINTEF have joined forces to at least take a few steps towards the open system.

We have tried to be pragmatic in our approach to the problem and have decided to focus on four areas: 1) The use of established standards, e.g., Ethernet, TCP/IP, and X. 2) The use of a distributed operator interface based on X. 3) The selection of low level protocols for system communication instead of high level protocols like MMS. 4) An integrated framework for operator support, e.g., alarm filtering, decision support, on line data analysis etc.

Introduction

This paper gives an overview of MITS - “Maritime Information Technology Standard”. MITS is an open standard for integrated ship control systems. It is being developed by Norwegian industry in cooperation with users, governmental agencies, and research institutions.

The goal of MITS is to enable the users of maritime control systems to utilize the potential of open systems. The potential is mainly in the increased availability of information (e.g., new functionality) and in the open definition of interfaces (e.g., mix components from different vendors).

To reach this goal we have decided to focus on three areas:

- **System architecture - SA.** SA covers structured data movement between modules in the system and the administration of data access (configuration, exception handling, etc). SA is the integrating “glue” in MITS.

- **Man machine interface - MMI.** A MITS system will more often than not consist of modules from different producers. A standardized MMI approach from all producers of MITS equipment will make the use of the system easier.

- **Operator support - OS.** An integrated control system offers more information to the user. The third part of MITS is a framework for advanced operator support, e.g., alarm filtering, decision support, and on line data analysis.

The principles and ideas underlying MITS is not confined to maritime control systems. Most of them are equally valid for process, manufacturing, and other control application. Thus, we will normally use just control system instead of maritime control system in the text.

The paper is divided into the following parts: Section 1 tries to analyze the typical control system used today and find their advantages and drawbacks. Section 2 gives an overview of the emerging “open systems” and their advantages and drawbacks. Section 3 describes MITS and compares it to the traditional systems and the new “open systems”. Section 4 looks at projects similar to MITS, sums up the preceding sections, and tries to identify the factors that we believe makes MITS a viable concept. It does also give a short overview of the current status of the project. Section 5 ends the paper with acknowledgments to the different participants and funding partners in MITS.
PROPRIETARY SYSTEMS TODAY

Today's control systems have few openings for easy integration of third party equipment. There is a trend towards the use of networked solutions, but the interfaces and protocols are usually proprietary. There are virtually no standards or guidelines for communication with other vendors' equipment. Thus, a considerable amount of work and energy is required to integrate third party equipment. This is in strong contrast to the marketplace of office computers, where mix and match capability is a prerequisite for most customers.

The maritime and process control systems used today are characterized by one or more of the following points:

- **Point to point communication.** Many systems consist of several modules that communicate by dedicated point to point links (typically serial line).
- **Proprietary communication protocols.** Third party producers have difficulties in using data available in the system.
- **Proprietary hardware and software.** There is little use of standard components.
- **Proprietary MMI.** Every producer of equipment has his own standard for the design of MMI. Every system exhibits its own advantages, quirks, and idiosyncrasies.

Lately there have been some significant changes to this picture. There is a trend today that more and more systems use some type of network for data exchange between modules. However, even in these cases the protocol is usually proprietary. Another interesting development is the use of the PC with its DOS operating system. It has established itself as a kind of common platform for several independent vendors. This does not, however, imply any significant type of compatibility between the vendors' components.

ADVANTAGES AND DRAWBACKS

There are advantages to the proprietary approach. They roughly fall into three categories:

1. **Producers economy.** High volume production may give higher profit or lower prices when you produce components yourself. Established technology (e.g., what we used last time) may be more cost effective than new technology.

2. **Protection of market.** If you are a large producer, there is little sense in making it easier for your competitors to sell alternatives to your own components. A good MMI solution is an important sales argument. Most vendors put a lot of effort into MMI and it has to be, to a certain degree, vendor specific.

3. **Consistent quality.** One integrator with full control of all components should give a fully integrated solution. This applies to MMI as well as module integration.

There are, however, serious drawbacks to the proprietary approach. A rough grouping of drawbacks could be something like the following.

1. **Inability to use standard components.** The producer may have difficulties in utilizing developments made in other market segments. Use of standard hardware and software may lower development costs.

2. **The user is hostage.** A proprietary standard may force the customer to buy all subsystems from one vendor. These subsystems may not always be the best available for their particular jobs. The supplier can more easily press prices up.

3. **Difficult tailoring.** Third party subsystems may have to be used in certain cases, e.g., when the vendor can not supply a specific function. This may lead to problems in the overall consistency of the system, particularly in the MMI.

4. **No external quality control.** Consistent quality do not necessarily mean good quality. Proprietary and "secret" standard may hide bad implementations and dubious principles from the public view.

An important advantage in proprietary systems is the consistency inherent in having only one supplier. However, a similar effect can be got in open systems by selecting a "good" standard with sufficient support in the market to ensure a consistent and continuous development.

The decision to develop MITTS was based on our opinion that the proprietary system has more drawbacks than advantages. We felt that the only parties that have a real benefit from proprietary systems are large vendors that can supply complete system solutions. Norway has a lot of users of control systems and only small to medium size producers. Thus, we felt that the time was right to look forward from proprietary systems and current technology to the solutions of tomorrow.
OPEN SYSTEMS TOMORROW

The main characteristics of an open system is that the interfaces and protocols between parts of the system, and between the system and the environment are standardized, open and well documented. Such standards may evolve either because the industry adopts certain products which become de facto standards due to a large user base, or as a result of the work in standardization committees, vendor or user groups. Typical examples of the first are DOS, TCP/IP and Windows 3.0. Examples of the second type are the OSI protocol layers, ANSI C, the X Window System and the POSIX standard for application program interface to operating systems.

ASPECTS OF OPEN SYSTEMS

The interfaces in an open system are of different types. The most important for a control system is the interface between the modules of the control system and the interface between modules and the operator, i.e, data communication and MMI.

Other important interfaces in a control system are inside the modules, particularly between application software and the hardware platform and operating system. These interfaces are, however, not as crucial as the communication and MMI interfaces. The primary gain in standardizing hardware and operating system is in reuse of software, not so much in system performance or user friendliness.

Communication

Module to module communication must be standardized so that different vendors can deliver components to one single control system. The selected standardization approach must cover all seven layers of the Open Systems Interconnection (OSI) model, see e.g. [1].

<table>
<thead>
<tr>
<th>Companion standards</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
</tr>
<tr>
<td>Application</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>Presentation</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>Session</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>Transport</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>Network</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>Data link</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>Physical</td>
</tr>
</tbody>
</table>

Figure 1: The OSI Layers

In addition it has to cover an additional “eight layer”. The OSI model does not include a layer that describes the interpretation of data passed between communicating entities. This is, however, of great importance in an application oriented system like MITS. It is not enough to know how to read data from an application unit. We do also have to know how to extract the relevant information from the data.

There are some established standards for communication in industrial environments, the best known is probably the Manufacturing Message Specification (MMS) [2]. MMS is the top layer in the Manufacturing Automation Protocol (MAP), i.e., level 7 in the OSI model. MMS “companion standards” take care of the eighth level. Companion standards for MMS in the process industry is being worked at by the ISA SP72 working committee [3]. A short overview of MMS can be found in [4].

Fieldbus is also an emerging open standard for communication in control systems. Whereas MAP is used in “high level” control, Fieldbus is intended for simpler devices on a low cost network or bus. The upper protocol levels in Fieldbus will probably be based on MMS, but there is still a lot of confusion about what exactly Fieldbus is going to be [5].

The most popular open protocols are probably TCP/IP (Transmission Control Protocol/Internet Protocol) and UDP (User Datagram Protocol - one of the IP family of protocols). Both are well described in the literature, see e.g. [1]. They cover the four lower layers of the OSI model. These two protocols has gained their popularity through their status as de facto standard in the UNIX environment. Virtually any type of UNIX computer (and most other computers too) offer one or more implementations of these protocols. The implementations are low cost, or not uncommonly, free.
The problem with these protocols is that to use them in a control system we have to design the upper three (or four) levels ourselves. This is, however, not too difficult.

MMS and Fieldbus was considered for use in MITS. We found that the ISO protocols did not have the widespread use that TCP/IP and UDP have. An effect of this was that the cost of buying an MMS implementation for MITS would have been prohibitively high. We decided to use the IP family of protocols and use a relatively simple special purpose top level protocol.

**MMI**

There are at least four important aspects of openness for the MMI part of an application:

1. The look and feel as it appears for the end user, i.e., how smooth and safely the user may switch between different parts of the system. Typical examples of look and feel standards are the style guides provided for Windows [6], OSF/Motif [7], and the Open Look GUI [8].
2. The application program interface (API) that implements a particular look and feel standard, i.e., an API to windows, buttons, function keys etc. Examples here are the Motif widget set and the XView toolkit.
3. Standards for drawing, manipulating and storing graphics. Examples are GKS and PHIGS for 2D and 3D graphics.
4. Standards for handling of graphics and user commands in a network environment. The X Window System is the major standard in this category.

The first point is the most crucial. It embodies the actual interface between the system and the user. Point two and three is important for the movement of software from one platform to the next, it is not an requirement as such. The fourth point, however, is worth noticing. X includes an open communication protocol and is important for systems where we want to use graphic user interfaces in a network environment.

**Hardware and Operating Systems**

This aspect of open systems is not an absolute necessity in a general control system. It has, however, an impact on the producers benefits from the system. Standardized hardware and software platforms give the developers more opportunities for reuse of application software. This will lead to shorter development time, less software bugs, and lower cost. Keywords in this context is DOS, UNIX, and PCs. One could also include standard program languages like ANSI C and C++.

**BENEFITS AND DRAWBACKS**

Many of the benefits of open systems are the opposite of the drawbacks of proprietary systems. The most important is possibly that the users becomes independent on a single system vendor. This and other aspect have already been described in the section on proprietary systems.

A new aspect of open systems is the free information exchange. Standardization of the communication between modules opens for new types of functionality and improvements by letting one module use data from other modules. Old functions may also be performed better due to better access to data. This is particularly interesting in conjunction with new techniques such as decision support and other types of “artificial intelligence”.

The most obvious benefit for the producer is the availability of standard hardware and software components. This reduces development time and cost and enables the company to compete where it is good. It does not have to waste resources on developing standard components like networks, communication software, and basic graphic capabilities.

Open systems means that several companies can deliver components to one system. This is a very important benefit for smaller companies. Today they usually have great problems penetrating the market dominated by their bigger cousins.

Finally, it should be noted that open systems have potential to benefit small and large vendors alike. The openness gives a potential for adding more functionality at the same cost. This may entice the owners of ships to renew their old control systems more often than today. It may also cause builders to add more functionality, at higher prices, than they would today. The net result may be that the size of the market increases and that all producers get a larger share.

Benefits of proprietary systems are the drawbacks of open systems. The main problem is the vulnerability to divergences within the standards. The different “nearly” compatible variants of the UNIX operating system is a typical example of this. It is therefore important to rely on well established standards with a large and stable user group.
Another special problem is the security risks inherent in open systems. The standardization of the communication protocol makes it easier for hostile persons or organizations to sabotage or listen in on an open system. This may be remedied by limiting the access to the physical system or by building security into the transmission protocol.

MITS

MITS is an example of an open system. We have taken the advantages of open systems, while keeping an eye on potential problems, and tried to merge this into a consistent whole which is suitable for users as well as producers. This sections shows some of the highlights of MITS.

SYSTEM ARCHITECTURE

The basic unit in the MITS system architecture is the MITS Application Unit (MAU). A MAU contains one relatively independent piece of the application, e.g., conning display, radar, electronic chart, etc. The MAUs are interconnected through a logical network where all MAUs have access to all other MAUs. This is shown in figure 2.

Figure 2: The MITS system architecture

The logical network makes it possible for a MAU to read data, write data or call functions on all other MAUs. This can be done independently of the underlying transport mechanism. Furthermore, the interface between the MAU and the logical network is simplified so that most of the interface can be pulled out of standard libraries. This is shown in figure 3.

Figure 3: The MITS system software

The shaded areas are part of the generic software that is independent of application. It is developed once and can be reused in all MITS systems. The different parts of the generic software is:

- **MAPI.** MITS Application Interface. It defines the interface between the real application and the rest of the system. Several MAPIs will be available. The differences will cater to differences in operating systems, program languages and so on.

- **MCL.** MITS Connection List. This is a description of a MAUs imported and exported data and functions. The MCL is used by the generic software as a kind of table based description of the MAU.

- **LNA.** Local Network Administration. This unit takes care of most administrative tasks for the MAU. It transmits data from and to the MAU, it takes care of MAU configuration, start up, exception handling and so on.
- **NWI.** Network Interface. This is essentially a TCP/IP or UDP implementation. It has extensions for handling redundant network and different transport medium.

- **GNA.** Global Network Administration. This unit takes care of general administrative services in the system. It is used to register new MAUs and to inquire about a specific MAU’s location.

The main task of the MITS system software is to support data transfer between MAUs. In addition to the pure transport aspects it also has to provide services for exceptions, e.g., start up, shut down, configuration, error handling, etc.

### General Functionality

The following principles are some of the most important in the design of MITS. The principles are embedded in the software structure as shown after the list.

1. **Minimum overhead on MAU.** It should in principle be possible to create a MAU on a single chip microprocessor. The interface between MAU and logical network shall be as simple as possible.

2. **Automatic identification.** It shall be possible to add or reconfigure a MAU to a working system without changing existing MAUs or shut the system down.

3. **Simple configuration.** The configuration of a MAU, i.e., determine from where it can import data and to where it can export data shall be as simple as possible. There should preferably be tools available for this task.

4. **Consistent “companion standard”.** Modules with equivalent functionality should be interchangeable. It is a goal that different producers can make compatible “plug in” modules.

5. **Different types of transport medium shall be supported** (serial, ether, satellite). This includes remote login from, e.g., shore, for diagnostic and administrative purposes.

6. **Software Reuse.** The system shall be made in a way that ensures reuse modules.

A MAU need not be located at the same computer as its LNA. The MAU can use Ethernet, serial line, or Fieldbus for communication with the LNA. The MAPI is “light weight” and can be implemented on, e.g., small 8 bit microprocessors. Most of the administrative overhead is taken care of by the LNA (see points 1 and 4).

The establishment of a GNA is automatic at power up. Mechanisms exist to handle automatic restart and redundant GNAs. The communication to the GNA uses broadcast so that new MAUs or LNAs can be added without knowing the physical location of the GNA (see point 2).

The table driven description of the MAU through the MCL enables automated configuration tool and makes it possible to delegate administrative tasks from the MAU to the LNA (see point 3). The MCL is also the vehicle for the “companion standards”. We will have to organize the definition of MCLs so that vendors can rely on getting necessary data from a module just by inspecting the MCL. It is a goal to define a kind of class hierarchy of MCLs. This will make it possible to define minimum sets of functionality for a given class of modules, e.g., alarm modules (see point 4).

The use of TCP/IP and UDP in the NWI makes it simple to interconnect LNAs on completely different networks. The TCP/IP and UDP protocols is supported on a wide range of media, from fiber optic high speed net, satellite voice and data channels, and serial lines (see point 5).

Most of the system consist of generic parts that are independent of application. We have emphasized a structured and modular approach to the design of the software and this ensures a large degree of flexibility (see point 6).

### Data Transport Functionality

The data transport mechanism in MITS is based on using two different types of networks. Ethernet with Internet protocols (TCP/IP and UDP) are used for high level “soft” real time control. Some type of simple instrument bus, e.g., Fieldbus, is used where there are hard real time requirements. This mix is easily supported on all levels of the MITS architecture. All MAUs will be integrated by one logical network regardless of what type of physical network they are connected to.

Ethernet has traditionally been viewed as rather suspect by most real time programmers. This has partly its basis in a thorough investigation of different network types, performed by Bux in 1981 [10]. He shows that Ethernet can saturate at a sustained load of about 40%. Another aspect of Ethernet that is somewhat hard to accept is its statistical nature. Ethernet does not do any arbitration between connected units. It uses statistical principles to determine when a unit may use the network. This means that it is difficult to determine an absolute upper boundary for the time it takes to send a message from a unit to another. The saturation effect is partly a function of Ethernet’s statistical nature.
Ethernet has several good sides. It is completely flexible with respect to connecting and disconnecting units. It is relatively simple to use, it has high bandwidth, and it is cheap. It is also one of the dominant network types in data processing and office automation systems. TCP/IP and UDP is implemented on virtual all hardware systems with an Ethernet interface, and quite a few that uses other types of communication interface. A very good point in favor of TCP/IP today is that the X window system can use it for its communication.

A lot of experience has been gained in the use of Ethernet since its introduction in 1976. It has also gained acceptance in real time control systems. Experience and more careful analysis have shown that the potential problems inherent in Ethernet are more or less “non-problems”, in the sense that they do not show up in a correctly designed system [11], [12].

Our opinion is that the stability, robustness, and usefulness of Ethernet in “soft” real time applications have been sufficiently well established. Ethernet is probably not useful where there are short and bounded response times are needed. For these applications we will be using a simpler type of network which support these demands better. We have not yet selected a particular implementation of an instrumentation bus. A major contender would have been the Fieldbus [5], except that no Fieldbus standard have been defined. However, MITS should be able to support most transport protocols the emerging instrumentation bus will use.

MMI

The MMI recommendations in MITS consists of two parts. First, a style guide describing a minimum set of look and feel guide lines for MITS applications. Second, a standardization on the X Window system protocol for transfer of graphics between the applications and the consoles in the MITS Network.

Look and feel

Safe and smooth operation of systems from different vendors in an open environment requires a minimum set of common look and feel in the different application’s MMI. MITS defines a set of guide lines for look and feel based on a subset of the OSF/Motif Style Guide [7].

The Motif style guide is chosen as a reference of the following reasons:

1. It is closely related to the Windows 3.0 GUI for PCs.
2. It has strong support for operation by function keys.
3. Implementations are available on a wide range of computers.
4. It has a large user group.

The MITS Style Guide specially emphasizes a consistent framework for traversing, selecting and activating components in the MMI. This includes navigation by pointer or function keys, highlighting of the selected window and component and activation of components. Less attention is paid to the appearance of the different components.

The MITS Style Guide defines a set of function keys and interface components which includes: menus, buttons, scroll bars, labels and simple input of commands and text.

The MITS Style Guide does not cover application specific symbols such as valves, pumps etc. typically found in applications for machine and cargo monitoring. Neither is the graphic layout of such applications covered by MITS.

Implementation

The X Window system is today an established standard for network transparent distribution of applications using raster graphics. The MITS network structure combined with the X Window system, thus, facilitates a flexible allocation of functionality between the different consoles connected to a MITS Network.

Incorporation of the X Window system in MITS removes the device dependent code from the applications and provides the user and system integrator with flexibility in choosing consoles and network topology. The software developer is also freed from maintaining and updating a set of different device drivers. X servers are available today on a wide range of hardware including workstations, PCs and X terminals. The machine running the server is usually connected to a LAN, but one producer has just introduced an X terminal which can be connected to its host via a 9600 baud serial line. The speed is reported to be acceptable for multi window applications without too much graphics. A 38.4 kilobaud line will support graphic applications [9].
The widespread use of X has resulted in a rapid development of tools for building X applications, especially based on the Motif widget set. There is, however, also available tools for building the more dynamic oriented diagrams typically found in control and monitoring applications.

MITS only defines that the X protocol shall be used for network transparent connection between applications and consoles. The system developer is thus free to choose the most appropriate layer in the X software hierarchy. There may even be systems connected to the MITS network which not supports connections to other consoles and thus does not use X at all.

The use of the X Window system requires network bandwidth, memory and CPU resources. [13] describes a set of tests on a system configuration based on Ethernet that show that X only has minor influence on the network capacity.

OPERATOR SUPPORT

MITS provides an architecture which supports integration of operator support systems. Open distribution of data between modules and a common look and feel standard opens for integration of DS both within specific applications and as new more general modules.

Advanced operator support is to a large degree application specific. It is therefore not as well covered in the MITS standard as the system architecture and the MMI. We have found, however, that the full potential inherent in an open system like MITS hardly can be utilized without the help of modern operator support systems. The emphasis in MITS regarding operator support has been to make a suitable framework for the inclusion of such systems in MITS.

WILL MITS MEASURE UP?

Eventually the success of MITS depend on whether the approach taken addresses real and general problems associated with ship automation systems. Traditionally, ship automation systems is delivered world wide by relatively small specialist companies, maintaining high ship automation process knowledge. More of these companies have launched totally integrated ship automation systems, and has succeeded as long as they have had control of the major modules of these systems. This is of course expensive. On the other hand, larger companies in the business have not been able to gain dominance in this market, even though they should be able to write of their investments on more areas of application as process industry etc. All known applications so far has been in the creation of proprietary systems, and designing ad hoc interfaces to other proprietary systems when needed.

SIMILAR APPROACHES

Most other projects known has had as a goal to create systems with new or improved functionality for the operation, improving operational costs and/or safety. Such projects, such as "Projekt Skib" of Denmark, "Schiff der Zukunft" of Germany and "The Intelligent Ship" of Japan, have had a much wider scope than MITS, but it does not seem that any of them has put the same emphasis on basic technology of automation systems as MITS does. One reason for that can be that the success of the new open technologies particularly in office automation is quite recent. Although the systems emerging from these projects provide the wanted functionality and partly reduce yard installation costs, they are rather complex in system design, and demands much competence of yards, owners and classification societies. Also, they demand a continuous development work if they are to follow the rapid changes in technology in the IT world.

MITS set out to change this by connecting the ship automation technology to renowned international standards, which are maintained by others, In this way it is also less vulnerable to technology shifts. Therefore MITS offers possibilities for cost reductions in systems development. It does also provide easier handling by yards, owners and classification societies. as well as provide these parties more security in specification, commissioning and certification of such systems. One aspect of particular interest is the future of ship-shore and even ship-ship communication. as demonstrated by the "RACE Marine" project. This is a part of the European Commission's RACE Program. The main objective for RACE Marine is to develop and evaluate the multi media maritime communications by means of high speed data links between ship and shore.

RACE Marine and MITS thus complement each other because MITS mostly concerns data communication on board, while RACE Marine covers communication between ship and shore. A well established network on board will be a presupposition to profit by the possibilities provided by RACE Marine, being on-line diagnosis of trouble onboard, follow-up and guidance in critical situations etc.

TECHNICAL SOLUTION

Our technical solution is based on the use of standards from the "mass markets", i.e., Ethernet, TCP/IP, X, Motif and so on. We believe that this approach is:
1 **Technically sound.** We have found enough documentation to convince us that the selected standards are good enough for their intended use. They may in fact be better than the more traditional industry standards in some respects. This is an effect related to the widespread use of these standards. They are continuously being upgraded and optimized.

2 **Economically sound.** These standards are low cost and widely available. They make economical sense both because they are “cheap” and because they will be supported by their producers for a long time. Investments in such standards will live longer.

We believe that these points have been sufficiently well made earlier in this paper. Both our and others experience show both the economic and technical feasibility of these standards.

**MARKET ACCEPTANCE**

For MITS to gain market acceptance all parties involved must see probable benefits:

Ship owners are today paying more attendance to life-cycle costs, system operational safety and costs of modification. MITS offer concepts to make updates and modifications easier, as well as enabling consistent MMI for a total system.

Yards put emphasis on ease of installation and commissioning, flexibility in order to use the same system on more vessels, all in order to contribute to their competitiveness. By using a common standard as basis, reuse of basic competence is easier, and flexibility in adaption to the various customers needs is improved.

Classification societies are concerned about the possibilities of getting insight in the complexity of ship automation systems to be sure of safe operation under all circumstances. MITS do offer possibilities of higher quality products and safer operation through consistent MMI.

Shipping Authorities will in the future be more concerned about the quality of ships operating under their flag and along their coasts, MITS will contribute to this in the same way as for classification societies.

Ship Automation Companies are continuously seeking means of widening their market place and increasing their productivity. MITS will contribute to this by development cost reduction and by simplifying marketing efforts by referencing to a common standard.

By offering such benefits MITS have the chance of widely acceptance, in the same way as particularly office automation has succeeded, as it serves all parties involved.

**CREATED BY PRODUCERS, IN COOPERATION WITH USERS**

Another thing that puts MITS apart from some other standardization efforts that has been pursued elsewhere is the wide basis of the project organization.

What we feel is the main difference, is:

- Several leading ship automation companies are pursuing the project.
- Deliverers of machinery equipment follow the project closely, as they see possibilities for ease of connection to the automation system.
- The project was originally initiated by user interests, in order to overcome their difficulties in having a proper system for their vessels. Thus the Norwegian Ship owners Association is supporting the project.
- NTNF provides funding to the project, and have found it of particular importance that normally competing industries unite in this project.

Two pilot ship are planned, where the cooperating companies are welcomed to offer their products adapted to the MITS standard. These pilot ship will eventually confirm success of the technical concepts behind MITS.

**CURRENT STATUS**

The first version of the MITS network and MMI was demonstrated in February 1992. This included basic communication software, as described in the MITS overview, and a suggested look and feel based on the Motif standard. It did also incorporate a PC based decision support system.

During the final stages of the project in 1992 the MITS ideas will be tested in the laboratory with real ship automation equipment from the companies involved, After a short period of test result discussion, a set of rules and interface descriptions are to be issued from the project. These are to be further tested as part of the of the Norwegian Maritime IT programme in 1993-94, comprising two pilot installations of systems conforming to these rules.
ACKNOWLEDGMENTS

This project is a part of the Norwegian Maritime IT programme, covering many aspects of future ship automation. The work has been sponsored by NTNF (The Norwegian Council for Scientific and Industrial Research) as well as the Norwegian Ship Automation Companies SIMRAD ALBATROSS A/S, ROBERTSON TRITECH A/S, AUTRONICA A/S, and NORCONTROL AUTOMATION A/S. Personnel from these companies as well as from Det norske VERITAS has actively taken part in the project group. The main research partner of the project has been SINTEF Automatic Control of Trondheim, Norway, with the assistance of MARINTEK, also of Trondheim, Norway.

References
1 Tanenbaum, Andrew S., “Computer Networks”, Prentice-Hall 1989
3 Instrumentation Society of America (ISA) DS72.02, “Process Messaging Service”.