University of Leipzig
Faculty of Mathematics and Computer Science
Institute of Computer Science

Bachelor Thesis

A Diagnostic System for Remote Real-Time Monitoring of Marine Diesel-Electric Propulsion Systems

Abstract: An innovative diesel-electric power supply and propulsion system is a highly integrated electric system consisting of power generation, power distribution, and electrical drives. Every component of this system constantly produces operating data, which is read and evaluated by several programmable logic controllers, which in turn produce control signals.
This thesis presents a diagnostic system that collects the operating data and control signals from all power supply, propulsion, and control components on board, saves them to a hard drive, and enables an engineer to view the data remotely over the internet, in real-time as well as in retrospect. An industrial computer, certified for on-board use, is embedded into the ship's control unit, autonomously running the software that retrieves the data via an industrial Ethernet connection and makes it available to a remote user through a web interface or a database connection.

Leipzig, September 2011	Andersen, Björn Student of Computer Science
Presented to:	Prof. Dr. Martin Bogdan Institute of Computer Science Department of Computer Engineering

Acknowledgement

I would like to express my gratitude to all those who supported me in completing this thesis, especially to those mentioned below.
I am deeply grateful to Prof. Dr. Martin Bogdan for agreeing to supervise the writing of this thesis, for his interest in my work, and for his patient support throughout the process. Thank You!
I would also like to thank Jörn Hoffmann for his endless supply of suggestions for improvement, for his invaluable advice on architectural matters, and for his thorough search for gaps and inconsistencies in this work.
This thesis would not have been possible without the support of E-Powered Marine Solutions GmbH & Co. KG. I would like to show my gratitude to the director, my father Peter Andersen, at whose invitation the development of the system presented in this thesis came to pass. Although economical and technical difficulties often slowed the progress, he never ceased to support the project.
It was a pleasure to do the interface engineering with Michael Franck, project engineer with Interschalt AG and responsible for the propulsion control unit. Thank You for the prolific teamwork on and off the cruise vessel.
I am indebted to Sarah Mellor for copy-editing this thesis in an amazingly short period of time and reliably uncovering its linguistic imperfections.
I also owe my gratitude to Max Rockstroh, with whom I discussed innumerable problems and who often helped me in finding efficient and/or elegant ways to solve them.
I especially thank my wife Wiebke for her patient love, supporting me throughout all stages of this work and enabling me to complete this thesis.

Contents

List of Figures

1 Introduction

'Is it possible to observe the ship's propulsion from afar?'

This question arose after the development of an experimental diesel-electric marine power supply and propulsion system. As proven in this thesis, it can now be answered with a definitive 'yes'.

With the first deployment of a then unproven system architecture for power generation and distribution on a ship, the idea for a remote diagnostic system was spawned. Observing the operating data from afar, thus gaining more system knowledge and being able to analyse failures remotely, was the desired result of this work. As there is no adequate ready-to-use solution available on the market, a new system was to be developed. This thesis gives an introduction to diesel-electric propulsion technology in chapter 2, together with a discussion on the challenges of maintenance of such systems. Chapter 3 presents other research in the field of shipboard diagnostics and explains why the results are only partly applicable to the new system architecture.

Following a requirements analysis it was decided to build an embedded system that autonomously collects operating data from the power supply and propulsion control units, continuously records it locally, and enables a remote user to view live data as well as the complete log in retrospect. The results of the requirements analysis and the derived concept for realisation are shown in chapter 4.

For the implementation, an industrial computer was configured with a Linux operating system for operation within the switchboard of the control unit. It runs the newly developed software packages for data collection and remote retrieval, and the database server that ensures safe storage of the operating data history. The computer operates whenever power is available and automatically recovers from errors or power interruptions. Chapter 5 describes the implementation in detail.

This diagnostic unit was completed and installed together with the second deployment of the power supply and propulsion system on a river cruise vessel in June 2011. It has since facilitated remote failure investigation and system improvement through log history analysis, as outlined in chapter 6.

2 Basic Information

This chapter briefly describes the concept of diesel-electric propulsion systems and their use aboard motor vessels. It also explains why maintenance can be very cost-intensive and how a remote diagnostic system can significantly reduce time and effort for maintenance.

2.1 Marine Diesel-Electric Propulsion

For more than 100 years, military and civilian shipping has used diesel-electric transmission for propulsion systems. Their considerable advantages over traditional direct diesel engine propulsion are outlined in this section. In addition, one particular new system architecture is presented, into which the diagnostic system proposed in this thesis is integrated.

2.1.1 Conventional Diesel-Electric Systems

A diesel-electric power supply and propulsion system comprises one or more diesel engines, each connected to an electrical generator. The electrical power is used to drive one or more electric motors and to supply transformers that power the ship's network. This architecture allows the diesel engines constantly to be run at their optimal speed while making a mechanical transmission obsolete. Operation at optimal speed drastically reduces diesel consumption compared to variable speed operation and allows for optimised noise and vibration insulation. This insulation was an especially important reason for an early marine implementation of diesel-electric propulsion systems aboard United States Navy submarines in 1929 [Fri95], the first having been aboard the Russian river tanker *Vandal* [Tol76, Mot03].

In conventional diesel-electric systems, an independent switchboard system with fixed voltage and AC frequency is used for power distribution. Figure 2.1 depicts such a topology, consisting of three generator sets, six propulsion drives, and four transformers for the ship's network, arranged in two units.

The diesel generators need to be synchronised with each other and with the

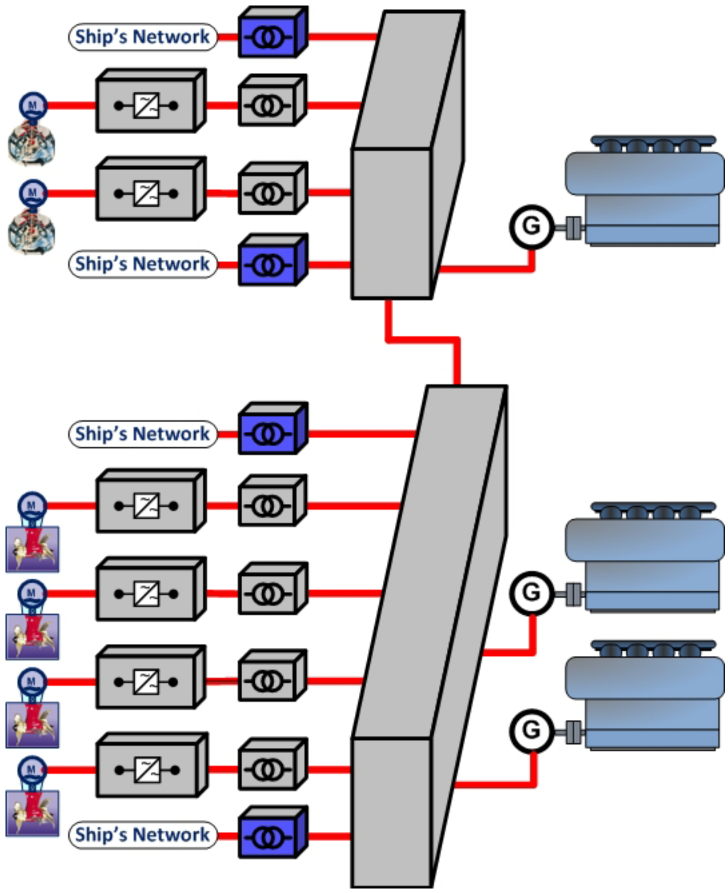

Figure 2.1: Conventional diesel-electric system (courtesy of E-MS)

ship's network. In addition, each motor has its own transformer and inverter to power the variable speed drive [And09, ATB09b], as can also be seen in figure 2.1. A significant part of the space freed by omission of a mechanical transmission is taken up by these components.

2.1.2 Innovative System by E-MS

Recent research on power generation and distribution, together with the present-day availability of highly efficient power converters, has facilitated the development of innovative network topologies. A review of current technological advances can be found in [MP07]. Among other things, the article presents a DC bus-based power distribution solution utilising rectifiers, inverters, and variable-frequency drives.
The Hamburg-based system integrator *E-Powered Marine Solutions (E-MS)* has

developed such an innovative diesel-electric system that eliminates many shortcomings of conventional systems. Their highly integrated solution is in operation on the river cruise vessels *MS Viking Legend* (since July 2009, figure 2.2) and *MS Viking Prestige* (since June 2011, figure 5.2). Both ships are 134.9 m

Figure 2.2: The MS Viking Legend (courtesy of Schiffstechnik Buchloh)

in length and 11.4 m in width, and can accommodate 190–200 passengers along with 50 crew members. Due to the high efficiency of their generator sets and propulsion drives, their fuel consumption and exhaust gas emission have been reduced by approximately 20% compared to similar river cruise vessels with conventional propulsion [Bog09].

Technology The high environmental compliance and small space requirements of the switchboards, converters, and power supply are attributed to the new network topology. Generator sets, propulsion drives, and transformers that power the ship's network are all connected to a DC energy bus via latest-generation *IGBT converters*, frequency converters controlling the AC devices with inverter/rectifier that is based on the insulated-gate bipolar transistor. The DC energy bus makes it unnecessary to synchronise the generators to the ship's

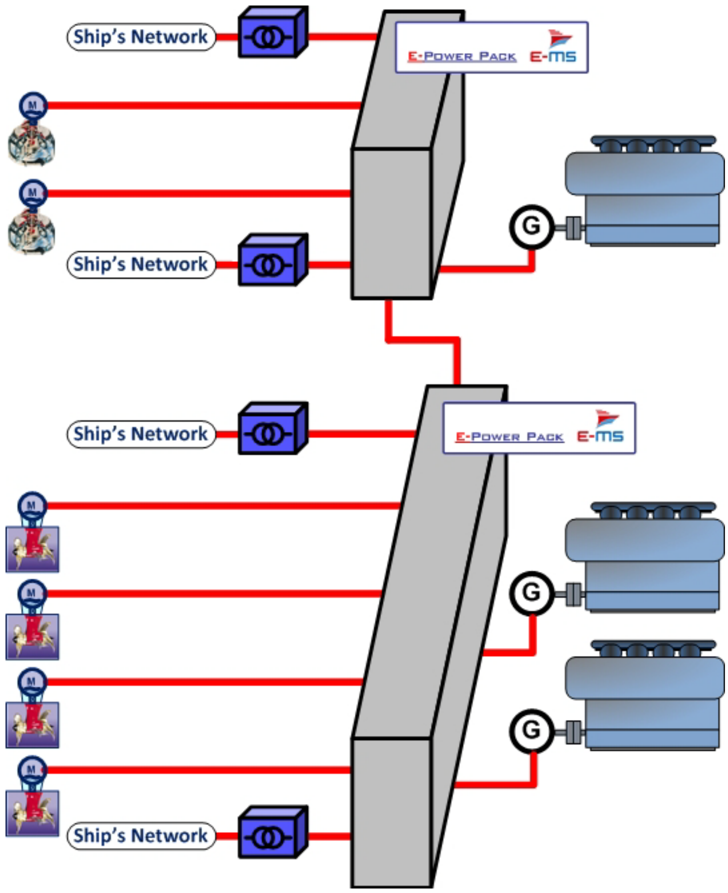

Figure 2.3: Diesel-electric system on the MS Viking Legend (courtesy of E-MS)

network, which allows for the use of cost-efficient asynchronous machines instead of complex synchronous generators.
Figure 2.3 shows the new topology of the switchboard system as deployed on the MS Viking Legend. In contrast to the conventional topology in figure 2.1, there is no need for additional components like individual transformers for each AC drive.
On both river cruise vessels, two independent systems, the converter units at the bow and stern, form the redundant propulsion and power supply system: the stern unit, consisting of four Schottel twin-screw rudder propellers with a power of 330 kW each, is the main propulsion system; whereas the two 300 kW Schottel pump-jet drives of the bow unit serve as bow-thrusters. Each unit is also equipped with generator sets energising the DC bus and with transformers supplying the ship's network.
Under normal operating conditions the DC buses of both units are connected so that any supplier can power any consumer. In case of an emergency, however,

the units are disconnected. The bow unit then serves as the statutory redundant emergency propulsion system and operates independently of the stern unit. A shore connection can power the DC bus when the ship is in port, so that the generators can be disconnected. Like the generators and motors, the shore connection does not have to be synchronised to the ship's network, by reason of the DC bus topology.
Figure 2.4 illustrates the complete system topology on the MS Viking Prestige. The converter units (dark blue boxes) with the high voltage DC bus distribute the power provided by the generators (top) or through the shore connection (light blue box, top right) to the propulsion drives (bottom left and right) and to the transformers (bottom central), which supply the ship's network main distribution bars (light blue boxes, bottom). The emergency bus bar (light blue box, central) is either powered by the main distribution bow or, in case of a blackout, by one of the emergency generator sets (top central).

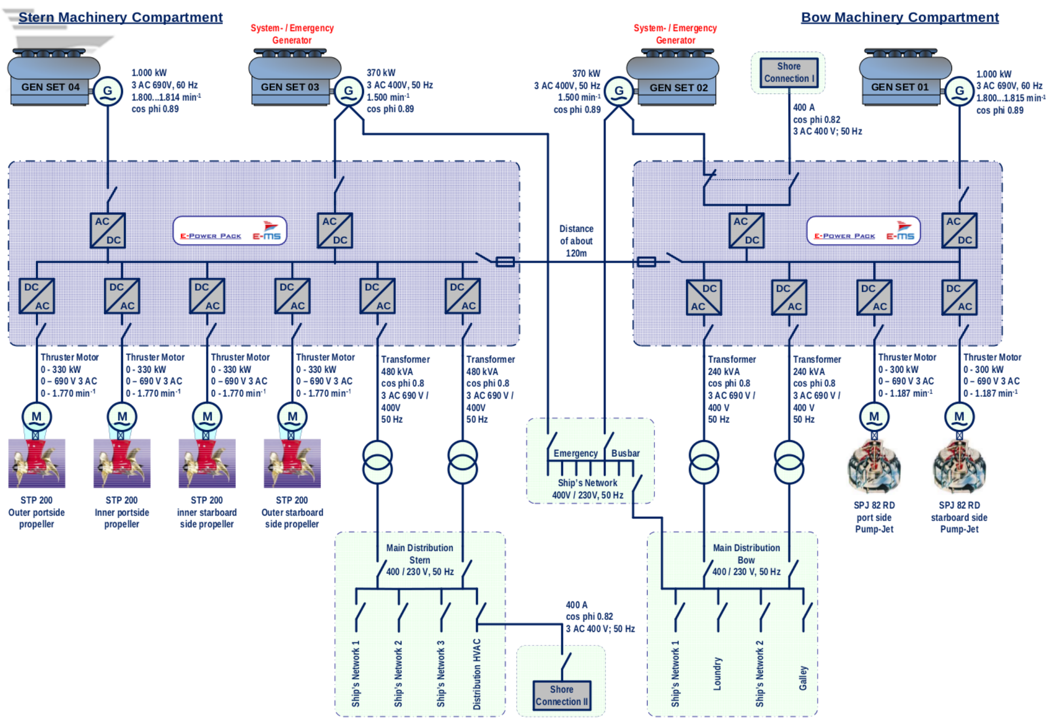

Figure 2.4: Network topology on the MS Viking Prestige (courtesy of E-MS)

How many and which generator sets are running is controlled by the *power management system*. This provides exactly the power needed at any given time by selecting the appropriate number and dimensions of the generator sets to be run. Within 15 seconds of an increase in power demand, the load is shared

with a newly started generator. The power management system also seeks to balance operation hours between the generator sets.
Furthermore the stern propulsion drives are equipped with a dynamic *torque control* system, as opposed to a conventional speed control. Especially in winding rivers, this dynamic control ensures a thrust distribution as intended by the captain, taking into account dynamically changing operating conditions like water depth and rudder angle [And09, ATB09b, ATB09a].
The system's instances on the river cruisers constitute only two topology examples of the power supply and propulsion system. Many different application-specific solutions are conceivable, based on the same modular architecture. The components for power generation, distribution, and the propulsion drives can be arranged in an arbitrary number of converter units, whose DC buses are connected to each other.

2.2 Maintenance

Although masked to the outside world by its high level of integration, the power supply and propulsion is an inherently complex system. Many components from several different manufacturers operate together under the hood, and every component constantly produces operating data. Some devices, like the IGBT converters, have their own integrated data recorder, but other devices' data is only processed by the *programmable logic controllers (PLCs)* and immediately discarded. This absence of recorded data leads to a high maintenance complexity in case of a problem, which can be detected by the crew or the alarm and monitoring system. Depending on the nature of the issue, many components may have to be inspected manually. Since the ship's alarm and monitoring system can record only warnings and failures, but not the circumstances leading up to them, it is often difficult, if not impossible, to find the source of a problem. Possibly unreliable statements of the crew further complicate the task. Inspecting the system on site can be very time-consuming and consequently, acquisition of replacement parts, if needed, is also delayed.
All of these factors contribute to a fairly cost-intensive maintenance operation.

2.2.1 Remote Diagnostics

This work proposes a remote diagnostic system to counter the maintenance difficulties outlined above. It provides the facts in every detail and in a structured manner, so that maintenance engineers can remotely identify the cause of a problem and can spend their time finding a solution rather than on find-

ing the problem. The diagnostic unit is specifically tailored to the needs of the diesel-electric power supply and propulsion system developed by E-MS and thus achieves a high level of integration.
In addition to processing every incoming piece of information, the PLCs now forward all relevant data to the diagnostic unit in very high temporal resolution. The data is autonomously saved and made available for analysis — both locally and remotely. It can be accessed through a web interface or a database connection, to monitor the vessel's current operating state, to examine events leading up to a failure retrospectively, or to aggregate operating data for statistical purposes. Given sufficiently large bandwidth, it is also possible to stream operating data to a remote terminal in real-time.
Chapter 6 illustrates the cost reduction that can be achieved by using remote diagnostics.

3 State of the Art

This chapter explains why existing concepts are only partly applicable to the situation at hand. Additionally, it briefly summarises the results of other scientific works on the concept of diagnostic systems in marine applications.

3.1 Applicability of Existing Concepts

Due to the new and complex architecture of the power supply and propulsion system developed by E-MS, the applicability of existing monitoring and diagnostic schemes is very limited. Widely-used solutions like the DEXTER system by Macsea Ltd. extract data from every single component and aggregate them in an external diagnostic unit. This approach, however, does not take advantage of the highly integrated structure of the power supply and propulsion solution at hand. The diagnostic system proposed in this work makes extensive use of in-depth knowledge of the architecture it monitors. It is thus possible to collect considerably more data, including internal system states, control flags, and command signals. From the memory of the unit, the entire system state at the time of a possible failure can be reconstructed, whereas an external system could only reconstruct the scenario as perceivable by an outside observer. With this integrated approach, examining the time line up to the point of failure is made easy and thereby the chance to find the error source is theoretically 100% (given that the data is consistent, i.e. there are no software errors in the PLC and all sensors are functional). As a result, the internal diagnostic unit can help in finding out *why* the error occurred, while an external system can generally only find out *where* it occurred.
The hardware is deployed as part of the switchboard and receives the operating data directly from the power supply and propulsion control units. Not only does the software monitor physical properties like voltages, currents, or machine speeds, but it also saves all input signals, internal and external indicator flags, warnings, and alarms. All sensors are built redundantly to reduce the chance of missing a critical value or generating a false positive due to a sensor defect. The full internal system state at any given point in time can therefore be reconstructed reliably from the database.

Established concepts and automation systems are used as subsystems, doing preliminary work for the diagnostic unit. These include the Siemens industrial automation modular PLC *SIMATIC S7*, variants of which are used as main and auxiliary control units on board, the *PROFINET* industrial communication interface, and, in future versions, Vacon's IGBT converter diagnostic interface, based on the controller area network (CAN), to collect even more detailed data from these devices.

3.2 Scientific Literature on Shipboard Diagnostics

With the recent emergence of integrated power supply and propulsion systems, there is a growing interest in the matter of shipboard diagnostics. One work partially applies to the architecture at hand. It is discussed further in this section. Other efforts are being conducted to aid the design of a diagnostic system architecture by the use of *failure mode and effects analysis (FMEA)* on the notional medium voltage direct current shipboard power supply system proposed by Hegner and Desai in 2002 [HD02]. At the current preliminary stage, however, none of these efforts are qualified to help the development of the diagnostic system presented in this work. Manufacturers of different kinds of shipboard power distribution systems presumably conduct similar analyses in the design stage of their products.
There are currently no publications on a *remote* system. Due to competition in the field of electrical ship propulsion, it is, however, likely that proprietary systems are under development, and will become commercially available in the years to come. For reasons explained in section 3.1, it is probable that these realisations will be coupled closely with a specific electrical system solution. Before a standardised component interface for diagnostic purposes can be established, generic diagnostic units that can integrate with any system architecture are not expected to be placed on the market.

Requirements of a diagnostic system In 2007 the president of Macsea Ltd., K. P. Logan, published an article about the requirements of shipboard diagnostics [Log07]. His work focuses on the survivability of the power supply and propulsion system and its automatic reconfiguration when sustaining battle damage on the future all-electric warship. Many of these requirements also apply to other shipboard systems though, as battle damage is of course not the only type of possible component failure on any type of electrically-propelled ship.
The author agrees with this work about the necessity of embedded diagnostics.

According to him, an ideal system acquired information about failures from component-level embedded diagnostic software agents. Together with sensor data and information from control units of all subsystems, it was able to provide a detailed report on the system state. Pervasive connectivity between embedded diagnostics and the control unit could be implemented cost-efficiently by existing industrial automation infrastructure solutions. From the equipment manufacturer's viewpoint, this also required diagnostic intelligence in every device and, most importantly, a standardised interface for exchanging diagnostic data.

Component-level diagnostic intelligence should also comprise device-specific diagnostic techniques, e.g. motor current signature analysis, partial discharge pattern analysis, or leakage flux analysis for motors and generators. Different techniques apply to power source or controller components. In addition to their built-in self-diagnostic capabilities, according to the author, they also had to provide connectivity to ship-wide diagnostics, to evaluate their operating data together with independently gathered sensor data.

Logan states that control software was usually closely coupled with the automation hardware. Engineering an external diagnostic system thus required knowledge of control theory, including hard- and software solutions, and industrial communication. Hardware dependencies furthermore made such a system difficult to port over to other control architectures.

Distributed intelligence approaches as presented by K. F. Drew in 2004 [Dre04] are also considered in the article, especially in the context of dynamic reconfiguration and survivability. For a power supply and propulsion system with a central control unit, it does, however, not stand to reason that a collaboration of device-level diagnostic agents be chosen over a central diagnostic unit.

The more complex future electric ships will be, the more the diagnostic knowledge must be managed, and methods be updated. According to Logan, diagnostic knowledge management required not only diagnostic intelligence in every electric component of the ship, but also the design of the diagnostic system early in the ship's life cycle, i.e. before construction, and the possibility to update both the control and diagnostic system during operation, when new system knowledge has been accumulated.

4 Design

In the first part of this chapter, the requirements to the diagnostic system are analysed. Based on these requirements, the design choices are then explained in the second part.

4.1 Requirements

Before designing the diagnostic system, it was necessary to specify two types of requirements it has to fulfil: the desired range of functions the system has to perform and a set of constraints it has to comply with.

4.1.1 Functional Requirements

Data collection It is imperative that data collection does not interfere with the regular operation of the power supply and propulsion system. Hence the data transmission has to be controlled by the PLCs, to ensure that no external control logic can manipulate the ship's routine operation in any way. Consequently the diagnostic unit has to be the passive recipient of incoming data. It must thus implement the server side of a connection protocol to which the PLCs connect as a client.
The communication solution needed to be chosen from the technologies inherently available to the PLCs: field bus or industrial Ethernet. Since the hardware of the diagnostic unit is to run a Linux operating system, for which device drivers for field bus interfaces are not freely available, industrial Ethernet is the method of choice.

Safe storage and data integrity in case of failure As one of the unit's primary tasks is to provide information about cases of system failure, it is essential that it is robust against faults affecting itself. Every set of data therefore has to be written to disk immediately. Power failure may under no circumstances corrupt recorded data. A *transaction concept* that prevents inconsistent database states is therefore absolutely necessary.

Remote access To make efficient use of the diagnostic unit, the current system state and any recorded data must be made available to a remote user. This requires an internet connection on board and the implementation of means to access the information. To view the current state of all components, a web-based interface is desired that presents the available attribute values in a structured way. Furthermore direct database access is needed to readout the operational history. In some cases it can also be helpful to stream operating data to the remote user in real-time.

4.1.2 Constraints

Stability, availability, and autonomous operation The diagnostic system is intended to serve as an objective observer, independent of any action by the ship's crew. User interaction or manipulation is thus neither necessary nor desired. Consequently the computer, integrated into the control unit, must boot, initialise, and function completely on its own under normal as well as under unintended conditions. This includes automatic resumption of operation after a failure like a power interruption. The system must never be locked into a faulty state, but needs to recover from every incident autonomously.

Industry standards As part of an automation system deployed on a maritime vessel, the diagnostic unit is required to comply with a variety of industrial and maritime safety restrictions. It may neither physically nor operationally endanger the control unit, the power supply and propulsion system, or the vessel in general. In particular, the supervision and safety surveys of the *Germanischer Lloyd* apply. In practice, the choice of hardware to embed into the control unit and the choice of connection protocol are thus restricted (see *Data collection* in section 4.1.1).

User-based access control The commercial use of the diagnostic system demands different levels of access to the live and recorded data. Information is not to be available to the general public, but to select groups of users. E-MS engineers must be able to request every detail, while their intellectual property, like internal control signals and indicator flags, is invisible to shipyard personnel and shipowner. User- and group-based access control in both the web interface and the database connection need to make sure everybody is provided only with what they are entitled to observe.

4.2 Concept

Careful consideration of all requirements has led to the concept for the diagnostic unit as described in this section.

Industrial computer To meet the physical demands made on the hardware, an industrial computer was chosen to host the software of the diagnostic system. The device deployed on the MS Viking Prestige is a Siemens SIMATIC IPC427C Microbox computer (figure 4.1). It is certified for on-board use in automation systems by the Germanischer Lloyd [Llo11], and has a high tolerance for environmental conditions. Built for maintenance-free operation and being embedded into switchboards, the device handles a high vibration and shock load well, is sufficiently electromagnetically compatible with any switchboard components, and tolerates the ambient temperatures and moist heat of the control cabinet [Sie10].

Figure 4.1: Siemens SIMATIC IPC427C [Sie11]

Its compact form factor (262 mm $\times$ 134 mm $\times$ 47 mm), DIN rail mounting, and 24 V DC isolated power supply make the Microbox well suited for installation into the control unit of the stern switchboard. A hardware *watchdog* ensures reset to intended operation in case of an unforeseeable software failure.

Universal operating system High stability and availability demand an established stable server operating system. The universal nature of *Debian GNU/Linux* and available expertise made it the OS of choice for the computer aboard the MS Viking Prestige. Although the diagnostic software is written entirely OS-independent, positive experience and the range of possibilities for future expansion provided by Debian convinced E-MS decision makers to further rely on it for upcoming deployments of the diagnostic unit. Hardware compatibility certified by Siemens [Sie09] further promotes GNU/Linux as a viable OS solution for embedded industrial computers.
Debian GNU/Linux inherently supports daemon processes and interaction-free operation, while providing very high stability and a comfortable means of user- and group-based access control. To date there has been no faulty system state or any OS-related failure in the unit installed on the MS Viking Prestige.

Data connection A switch connects the diagnostic computer to the bow and stern control units' PLCs via industrial Ethernet. There are no other devices connected to this network as of yet. In a future version however, the IGBT converters may also be connected via field-bus-to-Ethernet coupling, in order to collect even more detailed operating data than the converters currently transmit to the PLCs. On this isolated network, the diagnostic unit acts as a server to which the PLCs connect, using TCP to send their data frames. Thus, it is ensured that the PLCs' operation is not influenced by the computer in any way, which is an important requirement by the Germanischer Lloyd.
The connection to the ship's Ethernet network — and through it to the internet — is handled by the Microbox's other physical Ethernet interface. The computer thereby acts as a hardware firewall, shielding the control unit from outside manipulation.

Preprocessing In order to save hard disk space, only changes to any operating data value are recorded. An image of the last received data set is thus kept in the main memory for fast comparison. In one transaction per set of data, any changed values are written through to the hard disk, after updating the memory image.

Database storage and access By using the transaction concept, it is ensured that in case of a failure only the current data set is lost and the database is left in a consistent state. The *PostgreSQL* relational database was chosen not only for this feature, but also for its well-proven stability and scaling performance. The diagnostic software and the web interface connect to the database's server process, which can also be accessed remotely. Due to the low bandwidth of the 3G- and satellite-based internet connection of the river cruise vessel, it has, however, proven to be more time-efficient to compress any output to a remote query locally and send it to the remote client via the Secure Copy Protocol (SCP).

Web-based interface For an intuitive overview of the current system state, as kept in the main memory, a web-based interface was developed using *JRuby on Rails*. With a web browser of their choice, currently up to ten individuals can simultaneously monitor the ship remotely by connecting to the the interface, which implements PostgreSQL's *connection pooling* concept. The web application is written in Java and Ruby and is deployed on a *WEBrick* application server.

5 Implementation

This chapter describes in detail how the design was realised: from the hardware, operating system, and software environment to the main software packages, and finally to the live operation of the diagnostic system on board. The deployment diagram (figure 5.1) illustrates the interrelationships between these components.

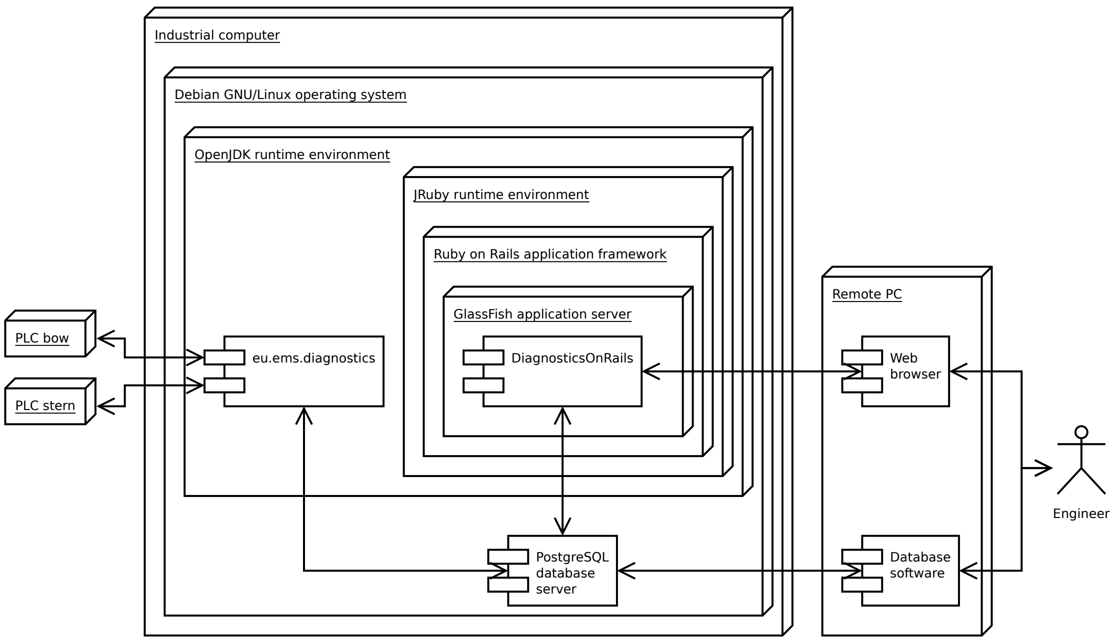

Figure 5.1: Deployment diagram of the diagnostic system on the MS Viking Prestige

5.1 System Configuration

Part of the engineering work went into the configuration of the computer system that runs the diagnostic software. While the reasons for the choice of hardware and software environment are outlined in the previous chapter, the sys-

Figure 5.2: The MS Viking Prestige [Kos11]

tem configuration itself, as in operation on the MS Viking Prestige (figure 5.2), is presented here in detail.

5.1.1 Hardware

The Siemens SIMATIC IPC427C Microbox industrial computer Although there are more powerful industrial computers available than the IPC427C, it is the reasonable choice for the task: its low voltage Intel Core2 Duo processor SU9300 has proven to have sufficient computational power to handle the data collection and processing whilst running on less than 10 W of power. The processor has a clock speed of 1.2 GHz and a second-level cache of 3 MB. The front-side bus clocks at 800 MHz. Also equipped with 2 GB of DDR3-1066 main memory, the IPC is capable of running the operating system and diagnostic software without the use of swap space.
The low power consumption also made it possible to wire the Microbox to the

existing 2 A circuit breaker of the control unit's HMI panel. Their combined consumption is below 48 W.

Figure 5.3: Interfaces of the SIMATIC IPC427C [Sie11]

Interfaces In regular operation the Microbox's only connected interfaces are the power supply and both network interfaces (figure 5.3). As user input or direct graphical output are neither required nor wanted, all other physical interfaces are unused. They can however be utilised in special cases: for example, when it has not been possible to connect to the remote backup server for a long time, the resulting large amounts of saved data can be transferred manually to a USB memory stick or an external hard disk.

5.1.2 Operating System

The industrial computer runs on Debian GNU/Linux 6.0.2 'squeeze' with an x86-64 kernel. Due to Debian's modular structure, only the required components had to be installed. Besides the standard system utilities, these include a set of tools for use on laptops and low-power embedded systems, providing advanced power management functionality; and the OpenSSH server for remote access to the machine. The other installed packages are only those needed to run the diagnostic software.

5.1.3 Environment

The diagnostic software is written entirely in Java and uses a relational database back-end to save the collected data. In the development of the web interface Ruby and Java were used. For the system to work, a runtime environment had to be provided. This section briefly examines the chosen packages.

Sun Microsystems OpenJDK6 Most importantly a Java implementation is needed for the diagnostic software to run. The installed *OpenJDK6* is the free and open source implementation offered by the Oracle Corporation (formerly Sun Microsystems). Version 6b18-1.8.7-2, built with the *GNU Classpath IcedTea* to replace proprietary code, is used on the Microbox.

PostgreSQL As for the database management, *PostgreSQL 8.4.8* was chosen to host the diagnostic database. In addition to the advantages described in section 4.2, its connection pooling support is used for reading access to the database. Furthermore, PostgreSQL provides comfortable tools for exporting and backing up data, as well as for automatic indexing and routine database maintenance concurrently with regular operation.
The PostgreSQL *JDBC* driver is compiled with the diagnostic software. As a direct-to-database pure Java driver, also known as a JDBC type IV driver, it requires no additional software to run and thus does not suffer from performance loss over a middleware like ODBC.

JRuby and Ruby on Rails The implementation of the diagnostic system in Java suggests a web application which can reuse connection routines and access the database in exactly the same way the diagnostic software does. Consequently *JRuby*, a Java implementation of the *Ruby* programming language that allows for combined use of both languages, and the associated *Ruby on Rails* web application framework, are the logical choices for the development of the web interface.
The Microbox runs JRuby 1.5.1 with Ruby 1.8.7 support and Rails 3.1.0. The built-in WEBrick application server, version 1.3.1, is used to deploy the web interface. A future release of the application may be deployed on the *GlassFish* application server, which provides better scalability and speed.

5.2 The Diagnostic Software

This section describes the core part of the diagnostic system: the software developed for the project. It is organised in two packages: the `eu.ems.diagnostics` Java package and the `DiagnosticsOnRails` web application. The former handles the data collection, preprocessing, and database input; whereas the latter provides one possible way to access the saved data locally or remotely. Both packages are written OS-independently, and require only the software environment detailed in the previous section at runtime.

5.2.1 The `eu.ems.diagnostics` Package

The package is divided into three sub-packages: a `controller` package that manages the application's control flow, a `model` package that contains all information about the components and topology of the power supply and propulsion system, and a `utility` package with auxiliary classes that provide static tools used throughout the application. While model and controller are strictly separated, the *model-view-controller* design pattern is only partially implemented by the `diagnostics` package, as the view component is sourced out to the self-contained application `DiagnosticsOnRails`.

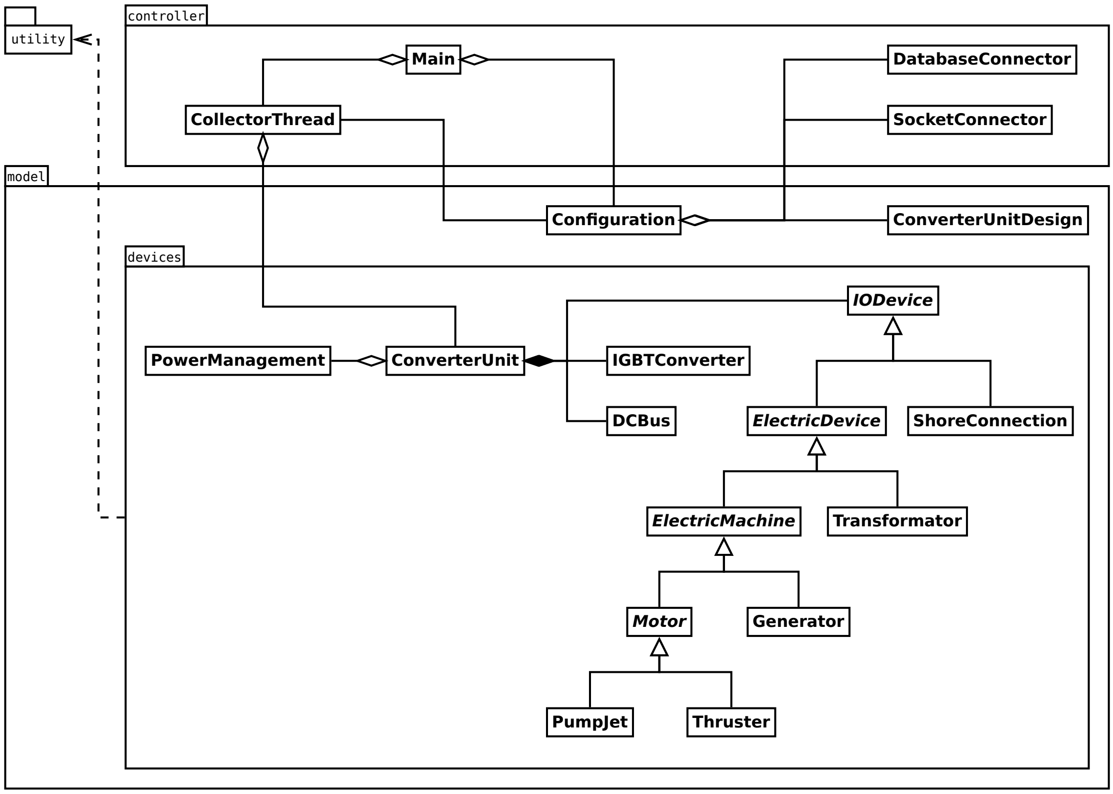

Figure 5.4: Class diagram of the package `eu.ems.diagnostics`

As can be seen in figure 5.4, the `controller` sub-package contains the application's main class. It is the entry point for execution, and it handles the control flow as seen in figure 5.5.

Control flow Upon application start, files for application log and error log are created. Since the program runs without any user supervision, these log files are needed to verify clean execution, analyse load spikes, and determine the source of runtime errors. Any irregular behaviour is recorded to the log files. This includes, but is not limited to, the loss of connection to a PLC or to the database, unusually high receive-buffer usage, and any runtime error that may occur. Timestamps allow for detailed examination and cross-referencing to vessel data in retrospect.

The next step is the loading of ship-specific data: the converter unit design. Naturally, not every vessel that the system will be deployed on has the same network topology. It might, in fact, be very different between, for example, a river cruiser and an offshore supply vessel. Hence the converter unit design is modular: a ship has at least one up to an arbitrary number of converter units, each of which contain up to sixteen IGBT converters and associated I/O devices. Those devices may be any combination of generator sets, motors, transformers, and shore connections.

Subsequently configuration parameters are set and stored in a configuration object. They define access method and user data for the database back-end, as well as host addresses of the PLCs. Furthermore, the configuration object serves as a container for references throughout program execution, for example, to allow all model objects to access the log file streams.

After initialising the configuration, the software connects to the database server. Whereas in the current implementation both run on the same physical machine, the software can also connect to remote databases. In future deployments of the system, scenarios are conceivable in which several physical sub-units of the diagnostic system connect to one dedicated database server. Whenever a connection to the database cannot be established, the software retries until connecting is successful. This behaviour is intended, since the main purpose of the diagnostic system is to record the operating data. That cannot be achieved without a database connection.

According to the ship-specific configuration, the data objects, which store the in-memory image of the current set of operating data, are now constructed. Every object, modelling exactly one logical component of the power supply and propulsion system, checks for the existence of a corresponding table in the database and creates one if necessary. It then constructs and initialises any sub-component it may have.

When all components are properly initialised, one thread is started for each converter unit that is supplying data to the diagnostic system. In the case of the MS Viking Prestige, the topology consists of two such units, each controlled by a PLC (see section 2.1.2). If a thread is killed for any reason, it is immediately recreated to resume data collection. It is recommended to run the diagnostic software on a CPU with as many cores as converter units or, alternatively, on a CPU with simultaneous multi-threading, supporting as many threads as converter units.

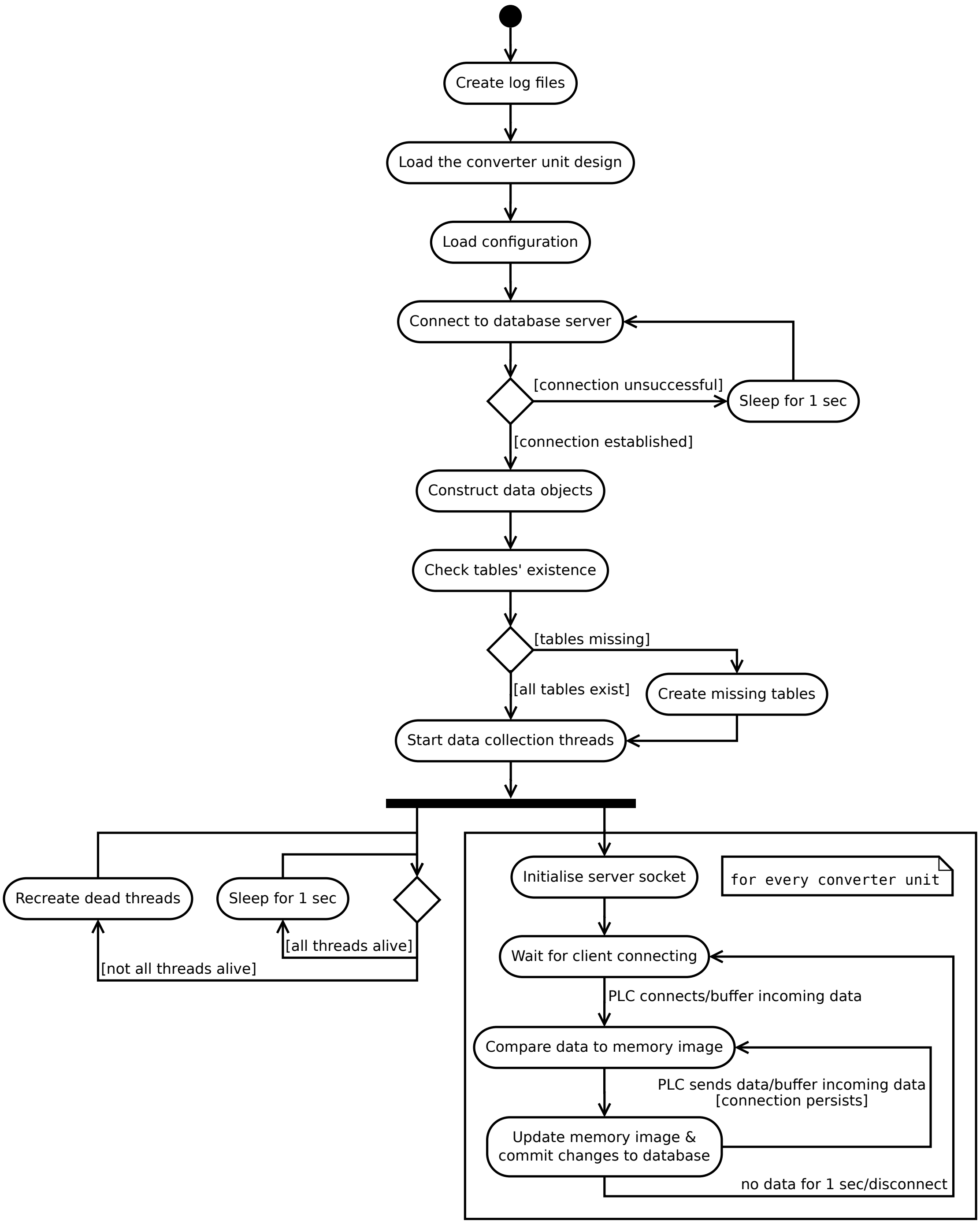

Figure 5.5: Activity diagram of the package `eu.ems.diagnostics`

Upon creation, a collector thread initialises the server side of its communication with the PLC. It then awaits a client connection and incoming data on a specific local port, which is known to the PLC. The PLCs transmit an image of the full system state every 50 ms. As long as the connection holds, the thread reads incoming data into a buffer, whose contents are then compared to the in-memory data set. For each table in the database, an SQL statement is dynamically generated to insert only the changed values. All table statements are then grouped into a single transaction that is written through. Thus, database consistency is achieved. If there is no data transmission for one second, the thread assumes a connection error, disconnects, and waits for a new connection to be established by the PLC. This behaviour continues indefinitely.

The data model Figure 5.4 also depicts the data model of the application, which is contained in the `eu.ems.diagnostics.model` package.
In the converter unit's data object, values are stored that cannot be attributed to a specific component, mainly control signals and status flags. Its child object representing the DC bus stores data directly related to the converter unit's energy backbone, for example, the bus voltage. Physically connected to the DC bus are the IBGT converters, whose data object representations store all data these devices send to the PLC. These include, among others, power, voltage, current, AC drive frequency, and the converter's temperature. Data provided by each of the connected I/O devices is stored in its own object. As many devices share a set of variables, inheriting them from a common abstract supertype suggests itself:

- All I/O devices but the shore connection (herein called *electric devices*) share, for example, winding temperatures and associated alarms.
- Electric machines (generators and motors) have a set of common values, for example, bearing temperatures and associated alarms.
- Both types of motors that are in use on the MS Viking Prestige share, for example, many status flags.

One converter unit object per ship also has a power management data block. Every piece of information and every control signal related to the automatic power management and distribution system is stored therein, e.g. available power and current load, indication of the current operating mode, and status/control signals for the generator sets.

5.2.2 The `DiagnosticsOnRails` Package

This application provides a simple view of live data recorded by the `eu.ems.diagnostics` package (figure 5.6). There is only as much control logic in it as needed to access the database through a pooled connection, and to retrieve the data for the component that is selected for viewing.

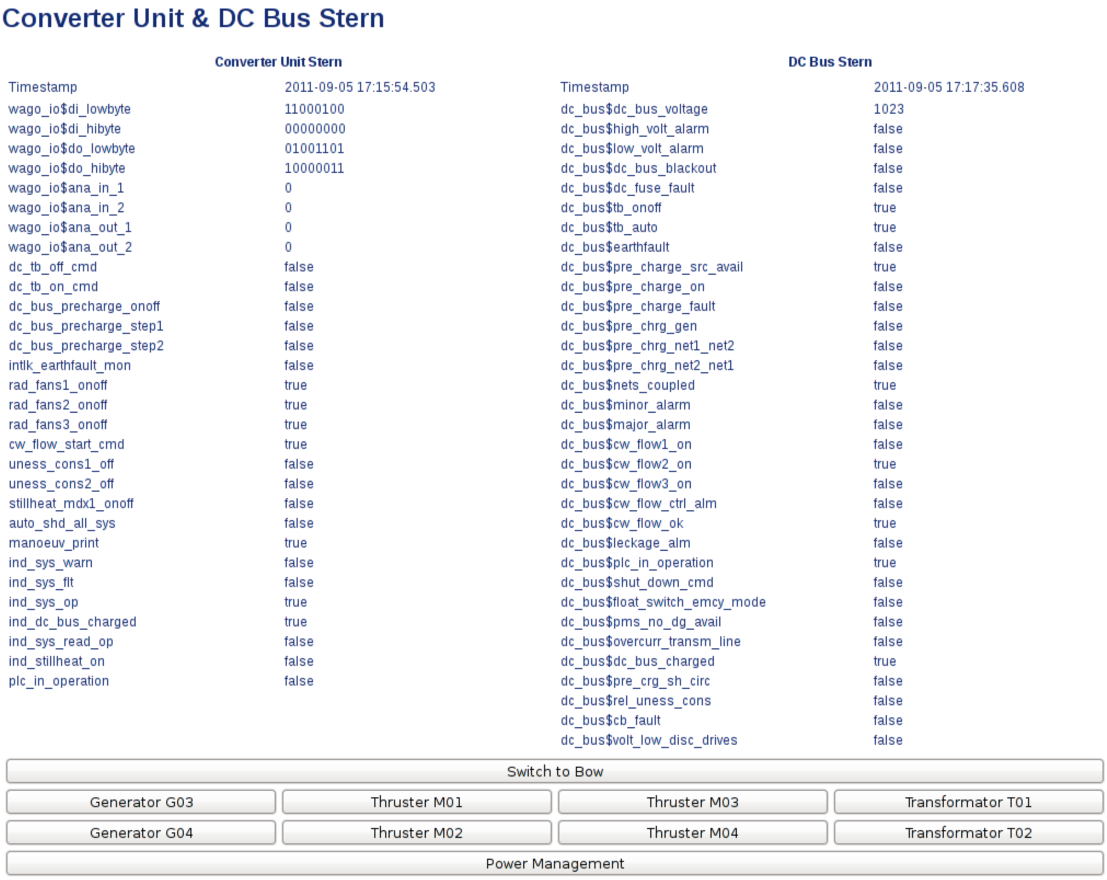

Figure 5.6: The simple web-based interface to access live vessel data

For every type of component, there is an HTML view with embedded JRuby code that dynamically generates a web page with current vessel data, which is retrieved from the database. Given database access, the index page of each ship and every component view automatically identifies the network topology, and offer navigational elements for the existing components in the ship-specific layout.

This web interface will be integrated into a user-restricted area of the company website `http://www.e-ms.eu/`, for customers to remotely view live data of their vessels.

5.3 The System in Operation

Having described the configuration and the software implementation above, this section focuses on the live operation of the diagnostic system.

5.3.1 Start-Up

The Microbox computer is connected to the 24 V circuit supplying the switchboard control unit. It is thus started automatically whenever the switchboard power comes up. This however, rarely occurs, as the switchboard power is supplied by a 24 V UPS in case of a ship blackout, and should therefore never be interrupted in the first place.
The Microbox then starts the Debian GNU/Linux operating system. On operating system boot, the necessary software environment is also started: the PostgreSQL database server is brought up by its init script; the Java and JRuby on Rails runtime environment, together with the WEBrick application server, are loaded at time of program execution, which is triggered by the cron job scheduler at start-up.

5.3.2 Regular Operation

At runtime there are only two relevant tasks for the diagnostic system: data collection must run at any give time and accessing the data must be possible simultaneously. The former is achieved by the control flow described in the previous section. The latter is provided by the application server running the web-based interface for browser access, and by the database server an engineer can connect to for querying the diagnostic database. To avoid slowing down the data collection, the number of simultaneous connections is currently limited to ten.

5.3.3 Error Recovery

Should regular operation be interrupted by an unforeseeable error, the system is programmed to recover automatically. If a data collection thread exits unexpectedly, it is recreated. Should the whole diagnostic software stop responding, a hardware watchdog restarts the computer, which then resumes operation as described in section 5.3.1. In any scenario the database is left in a consistent state, since any unfinished transaction is automatically rolled back.

Examination of application logs has shown that the only error recovery that has ever occurred was due to a power failure in the converter unit and operation was correctly resumed. Other scenarios have been tested, but have never occurred during live operation. Those include, for example, loss of physical data connection to the PLC, transmission of invalid data blocks by the PLC, and manually depriving the software of system resources by a simulated denial-of-service attack.

5.3.4 Maintenance

Under normal conditions the diagnostic system does not need any manual maintenance, neither local nor remote. Nevertheless, the diagnostic computer runs a *Secure Shell (SSH)* server to allow remote login. It is thus possible to update the software remotely and apply bug-fixes if necessary.
At night time, when the internet connection usage (in case of a passenger vessel) is usually lowest, an automatic backup and maintenance script is run. It compresses all data older than two weeks to a backup file on the hard disk, and deletes the corresponding rows from the database tables. The backup file is then copied to a company FTP server in Hamburg, and deleted on the diagnostic computer upon successful transmission. At the same time, a garbage collection routine is run on the database, concurrently with ongoing write operations, to reclaim free space for new usage; and the index tables are updated according to the deletions.
The data collected within 24 hours on the MS Viking Prestige amounts to approximately 3 GB in the uncompressed database. After compression with an improved version of the Lempel-Ziv-Markov chain algorithm, it is reduced to roughly 100 MB per 24 hours of data.

6 Evaluation

The diagnostic system has been deployed on the MS Viking Prestige since June 2011. In the past months it was thus possible to perform analyses of several failures that occurred on board. In each case, the source of error could be identified beyond doubt, be it an internal or an external fault. Faults are considered internal when the responsible unit is part of the power supply and propulsion system. External faults are those caused by systems that were deployed by other contractors, for example, if the main distribution switchboard of the ship's network issues a command for transformer shut-down while the transformer is still needed to power the distribution.
In all above-mentioned cases of failure, E-MS was able to reduce its reaction time to failure reports significantly. Identification of the error source was quickly performed remotely and time to responsive action was drastically reduced, compared to regular maintenance procedure. Whereas for remote diagnostics only an internet connection is required, the prerequisites for a conventional maintenance procedure can be extensive:

- The chief engineer (Chief) of the vessel must be present and must be able to report to the maintenance engineer on the failure scenario.
- The power supply and propulsion system must be at the engineer's disposal for unrestricted switching operations.
- The ship must be well secured to allow for the propulsion drives to be run at high load for testing purposes.

Furthermore the conventional maintenance procedure itself is naturally more time-consuming and more expensive. Table 6.1 compares the conventional procedure to the simplified procedure using remote diagnostics. As can be seen therein, no travel expenses or time delays are incurred for the remote procedure. Additionally, the identification of the faulty component is always possible, because the fault scenario can be reconstructed in full with data from the diagnostic system, including internal command signals that allow for establishing an exact time line. The data is also less prone to error than the memory of the crew and is, in contrast, always available.
Using the diagnostic system for failure source identification on the MS Viking Prestige, E-MS has resolved at least four issues since June 2011 remotely that

conventional maintenance	maintenance with remote diagnostics
1. The crew reports a failure to the owner, who in turn reports it to the shipyard, who in turn reports it to E-MS. 2. The vessel travels to the nearest port. Meanwhile an engineer is sent to investigate the claim. 3. The engineer inspects all involved components to identify the error source. This step may involve extensive testing and/or a system shut-down. 4. If the third step was successful, responsive action is taken.	1. The diagnostic unit notifies E-MS about a failure. 2. An engineer starts downloading vessel data around the time of failure. 3. The error source is quickly and reliably identified using the downloaded data. 4. Responsive action or remote crew assistance is performed immediately.

Table 6.1: Conventional vs. remote diagnostics-assisted maintenance procedure

would otherwise have required sending for an engineer. The time for resolution of the claims was thereby decreased from an average of at least two days to approximately two hours, thus saving significant amounts on travel expenses and costly man hours.

Moreover, it has been possible to analyse vessel data over a sufficiently long period of time to acquire more knowledge about the system behaviour, and to notice trends within the data. Both have revealed opportunities to optimise the control software of the vessel at hand, and to optimise the system configuration for future vessels.

Having proven its usefulness, the diagnostic unit was incorporated into the portfolio of E-MS. It can be purchased as an optional component in the power supply and propulsion system, and existing systems can be retrofitted with it as part of a service contract. Due to the unit's high potential for saving time and expenses on maintenance, it is expected to be attractive for shipyards and shipowners. It will thus likely be deployed with every new power supply and propulsion system delivered by E-MS.

Shortcomings of the diagnostic unit One of the major advantages of the diagnostic system, the high level of integration into the control unit, can also be a disadvantage in future deployment scenarios. Due to the large amount of system knowledge incorporated into the data model, major adjustments to the software are required if the control unit is modified in any way other than changing the topology of the components. Any changes to the control unit's data model of the components will have to be reflected in the diagnostic unit's data model.

Additionally, in the current implementation, retrieving data by other means than the web-based interface requires basic knowledge of relational database systems. To ease the use of the diagnostic unit and to make comfortable access available to a wider range of company personnel, a platform-independent data retrieval tool is desired for the near future.

7 Conclusion

7.1 Summary

The maritime industry uses diesel-electric propulsion systems to substitute mechanical transmissions. The conventional synchronised diesel-electric systems, however, are in the process of being replaced by alternative architectures. One of these alternative systems is the direct-current energy-bus solution offered by E-Powered Marine Solutions as an integrated power supply and propulsion system. Due to the high complexity of such a system, maintenance can be time-consuming and costly.

This thesis presents a diagnostic unit for remote real-time monitoring, which is integrated into the control unit of the system developed by E-MS. In contrast to retrofitted diagnostic solutions, this diagnostic system thus takes advantage of in-depth knowledge of the control unit, and is able to reproduce the complete system state at any given point in time. Its data model is closely related to the physical system topology and the control unit's internal data model.

An industrial computer, which tolerates the control cabinet's environmental conditions, is running a highly stable Linux operating system, and is deployed with the power supply and propulsion system. It serves as the platform for the diagnostic software that receives operating data from the ship's control units via industrial Ethernet without interfering with their operation. Autonomously, i.e. without user interaction and automatically recovering from failure, the software processes the data and saves any changes to a relational database, which is protected against inconsistency by the use of a transaction concept. Access to the data is provided through a user-restricted database connection and by means of a simple web-based interface, which can both be accessed remotely over the internet.

Being deployed on a river cruise vessel since June 2011, the diagnostic system presented in this thesis has facilitated the remote analysis of several failures in the power supply and propulsion system. The error sources were identified without the presence of a maintenance engineer on board. Thus, the time to responsive action could be decreased by magnitudes. Significant amounts of travel expenses and man hours have thereby been saved.

7.2 Future expansion and research

The next deployment of the diagnostic unit is likely to see some changes in device coverage as well as in topology. The connection of the IGBT converters' diagnostic interface to the computer is currently under development and will probably be implemented by a field-bus-to-Ethernet coupling. While a big part of the converters' system state is already recorded by the diagnostic system through the data sent by the PLC, there remain internal signals that are not needed by the control unit, and are thus not yet read, but can be helpful for failure diagnosis. The direct connection of other devices to a future version of the diagnostic unit is imaginable but currently not projected.
The possibilities and benefits of a distributed architecture are, however, being explored at the moment. A possible future topology comprises even smaller embedded data collection computers in the control cabinets, connected to one central more powerful database and web server. The server hardware would not be restricted by environmental conditions and could thus offer large computing and storage capabilities inexpensively.
The main focus for future research is on the development of an intelligent system for preventive maintenance. The appliance of machine-learning algorithms to the recorded data is anticipated to reveal non-trivial interrelations within the data, and thus hopefully to facilitate failure prediction. This feature can potentially replace post-failure maintenance by scheduled preventive maintenance in some cases.
Furthermore, the data that is acquired by the diagnostic unit will be examined carefully to gain more knowledge about the specific power supply and propulsion system, and about direct-current energy-bus solutions in general. This knowledge is expected to improve future deployments and possibly spawn new ideas for future power distribution solutions.

Glossary

Connection pooling Database connections are maintained in a pool to be reused for future requests. This concept saves the time it takes to create connections.

Debian GNU/Linux A free and open source computer operating system that includes the GNU operating system tools and the Linux kernel.

Germanischer Lloyd The German classification society and technical supervisory organisation for the maritime, offshore and energy industries.

Failure mode and effects analysis A systematic analysis of a system and/or process to identify failure modes and possible consequences based on past experience. The analysis is used to redesign the system and/or process in question to minimise the risk of failures.

GlassFish A free and open source application server for the Java platform, Enterprise Edition.

IcedTea A free and open source build and integration package for OpenJDK, developed and maintained by the GNU Classpath project.

IGBT converter Herein used to describe the IGBT-based device that serves as an inverter, rectifier and frequency converter between the DC energy bus and a connected AC device.

Java Database Connectivity An application programming interface that provides database connectivity for the Java programming language.

JRuby A Java implementation of the Ruby programming language.

JRuby on Rails Ruby on Rails built with and supporting JRuby.

Model-view-controller A design pattern in software engineering that isolates application logic and domain-specific data from the user interface.

OpenJDK6 A free and open source implementation of the Java programming language by Sun Microsystems.

PostgreSQL A free and open source object-relational database management system.

PROFINET An open industrial Ethernet standard for automation.

Programmable logic controller An automation device that is used to control electromechanical processes in hard real-time.

Ruby on Rails A free and open-source web-application framework for the Ruby programming language.

Secure Shell A network protocol that allows users to login on a remote computer via a secure channel over an insecure network.

Torque control In propulsion, the concept of controlling variable frequency drives by setting the speed to gain a specified torque value.

Transaction concept A concept in database engineering: units of work are grouped into transactions that are either entirely completed or have no effect at all. The transaction concept ensures a consistent database state and correct recovery in case of a system failure.

Watchdog (timer) A timer, implemented in hardware or software, that resets the operation of a computer system when a program fails to service it regularly, due to a faulty system state.

WEBrick A free and open source Ruby library that provides an HTML web server to deploy Ruby on Rails applications.

List of Abbreviations

3G 3rd generation mobile telecommunications

AC Alternating current

CAN Controller area network

CPU Central processing unit

DC Direct current

DDR3 Double data rate type three

DIN Deutsches Institut für Normung (German Institute for Standardisation)

E-MS E-Powered Marine Solutions

FMEA Failure mode and effects analysis

FTP File Transfer Protocol

GNU GNU's Not Unix, recursive acronym for a Unix-like operating system

HMI Human machine interface

HTML Hypertext Markup Language

IGBT Insulated-gate bipolar transistor

I/O Input/output

IPC Industrial (personal) computer

JDBC Java Database Connectivity

JDK Java Development Kit

MS Motor ship

ODBC Open Database Connectivity

OS Operating system

PLC Programmable logic controller

SCP Secure Copy Protocol

SSH Secure Shell

SQL Structured Query Language

TCP Transmission Control Protocol

UPS Uninterruptible power supply

USB Universal Serial Bus

Bibliography

[And09] Peter Andersen. E-MS News - MS Viking Legend. News, August 2009.

[ATB09a] Peter Andersen, Claas Tepel, and Heiko Buchloh. For the sake of the environment: The first Diesel electric River Cruise Vessel – "MV Viking Legend". Press release, November 2009.

[ATB09b] Peter Andersen, Claas Tepel, and Heiko Buchloh. »Viking Legend« – Flusskreuzfahrtschiff mit dieselelektrischem Antrieb. *HANSA International Maritime Journal*, 146(9):36–42, September 2009.

[Bog09] Th. Bogler. Viking Legend Cruise Ship. Website, July 2009.

[Dre04] K. F. Drew. Distributed Machine Intelligence for Automated Survivability. Technical report, DTIC Document, 2004.

[Fri95] Norman Friedman. *US Submarines Through 1945: an Illustrated Design History*. Naval Institute Press, 1995.

[HD02] H. Hegner and B. Desai. Integrated fight through power. In *Power Engineering Society Summer Meeting, 2002 IEEE*, volume 1, pages 336–339. IEEE, 2002.

[Kos11] Peter Kosztolicz. Viking Prestige: Ship Photos. Website, July 2011.

[Llo11] Germanischer Lloyd. Certificate No. 45 201 - 07 HH. Type Approval Certificate, April 2011.

[Log07] K. P. Logan. Intelligent Diagnostic Requirements of Future All-Electric Ship Integrated Power System. *Industry Applications, IEEE Transactions on*, 43(1):139 –149, jan.-feb. 2007.

[Mot03] Trials and Tribulations of the Marine Diesel - A Look Back at the History of Diesel Ships. *The Motorship*, 2003.

[MP07] Víctor M. Moreno and Alberto Pigazo. Future Trends in Electric Propulsion Systems for Commercial Vessels. *Journal of maritime research*, 4(2):81–100, 2007.

[Sie09] Siemens AG, Industry Section, Postfach 48 48 90026 Nürnberg Germany. *SIMATIC IPC427C Manufacturer's Declaration „Suited for Linux"*, July 2009.

[Sie10] Siemens AG, Industry Section, Postfach 48 48 90026 Nürnberg Germany. *SIMATIC IPC427C Operating Instructions*, October 2010.

[Sie11] Siemens AG, Industry Section. Siemens Industry Automation and Drive Technologies: Image Database. Website, 1998–2011.

[Tol76] Robert W. Tolf. *The Russian Rockefellers: the Saga of the Nobel Family and the Russian Oil Industry*. Number 158. Hoover Press, 1976.

Affidavit

I declare that I wrote this thesis independently and on my own. I clearly marked any language or ideas borrowed from other sources as not my own and documented their sources.

I am aware that any failure to do so constitutes plagiarism. Plagiarism is the presentation of another person's thoughts or words as if they were my own — even if I summarise, paraphrase, condense, cut, rearrange, or otherwise alter them. I am aware of the consequences and sanctions plagiarism entails. Among others, consequences may include nullification of the thesis, exclusion from the BSc program without a degree, and legal consequences for lying under oath. These consequences also apply retrospectively, i.e. if plagiarism is discovered after the thesis has been accepted and graded.

place date signature