LAPPEENRANTA-LAHTI UNIVERSITY OF TECHNOLOGY LUT

School of Engineering Science

Software Engineering

MARITIME VESSEL CONTROL SYSTEM DEVELOPER PREFERENCES REGARDING THE INDICATION OF SYSTEM STATUS IN A REMOTE DIAGNOSTICS SERVICE

Examiners: Professor Jari Porras
Doctor of Science (Technology) Risto Tiainen

ABSTRACT

Lappeenranta-Lahti University of Technology
School of Engineering Science
Software Engineering
Samuli Siitonen

Maritime Vessel Control System Developer Preferences Regarding the Indication of System Status in a Remote Diagnostics Service
Master's Thesis 2020

80 pages, 9 figures, 8 tables, 6 appendices

Examiners: Professor Jari Porras
Doctor of Science (Technology) Risto Tiainen

Keywords: control system, remote diagnostics, maritime, vessel, remote diagnostics service, software developer, system status, user interface

This thesis has its basis on a potential need for a maritime vessel control system remote diagnostics service in the case company. A remote diagnostics service could provide cost savings to the company and its customers through more effective use of control system developer time during the performing of system diagnostics. To support the creation of a remote diagnostics service, the main goal of this thesis was to find out maritime vessel control system developer preferences regarding the indication of control system status in a remote diagnostics service. This goal was supplemented by the goal of finding out what control system values are important to system developers, especially when the overall system status is to be evaluated. As a result, it was determined that the inclusion of different control system values for monitoring depends on the problem to be solved, but the monitoring of control system software state machine states could have potential in this regard. The indication of system status as a collection function of time graphs on a so-called vessel timeline was deemed useful by control system developers. This finding was formed through semi-structured developer interviews and the usability testing of a developed remote diagnostics service prototype's user interface.

TIIVISTELMÄ

Lappeenrannan-Lahden teknillinen yliopisto LUT
School of Engineering Science
Tietotekniikan koulutusohjelma
Samuli Siitonen

Maritime Vessel Control System Developer Preferences Regarding the Indication of System Status in a Remote Diagnostics Service

Diplomityö 2020

80 sivua, 9 kuvaa, 8 taulukkoa, 6 liitettä

Työn tarkastajat: Professori Jari Porras
Tekniikan tohtori Risto Tiainen

Hakusanat: ohjausjärjestelmä, etädiagnostiikka, merenkulku, alus, etädiagnostiikkapalvelu, ohjelmistokehittäjä, järjestelmän tila, käyttöliittymä

Keywords: control system, remote diagnostics, maritime, vessel, remote diagnostics service, software developer, system status, user interface

Tämä diplomityö pohjautuu case yrityksen mahdolliseen merikäyttöisten alusten ohjausjärjestelmien etädiagnostiikkapalvelun tarpeeseen. Etädiagnostiikkapalvelu voisi mahdollistaa yritykselle ja sen asiakkaille kustannussäästöjä ohjausjärjestelmien kehittäjien diagnostiikkatoiminnan aikaisen ajan käytön tehostamisen kautta. Etädiagnostiikkapalvelun tuottamisen tukemiseksi tämän diplomityön päätavoitteena oli selvittää ohjausjärjestelmien kehittäjien mieltymyksiä järjestelmän tilan esittämiseen etädiagnostiikkapalvelussa. Tätä tavoitetta täydennettiin tavoitteella selvittää, mitkä ohjausjärjestelmän arvot ovat tärkeitä järjestelmäkehittäjille erityisesti järjestelmän yleistilaa arvioitaessa. Työn tuloksena ilmeni, että ohjausjärjestelmien eri arvojen mukaan tuominen tarkasteluun riippuu ratkaistavasta ongelmasta, mutta ohjausjärjestelmän ohjelmiston tilakoneiden tilat voisivat toimiva potentiaalisina järjestelmän tilan yleisinä indikaattoreina. Ohjausjärjestelmän yleistilan esittäminen joukkona ajan funktioiden kuvaajia niin kutsutulla aluksen aikajanalla vaikutti hyödylliseltä ohjausjärjestelmien kehittäjistä. Tämä tutkimustulos muodostettiin kehittäjien teemahaastattelujen ja kehitetyn etädiagnostiikkapalvelun prototyypin käyttöliittymän käytettävyystestauksen kautta.

ACKNOWLEDGEMENTS

I want to thank professor Jari Porras from LUT University for proving me with mentoring and advice during the planning and writing of this thesis. I also wish to thank D.Sc. (Tech.) Risto Tiainen for the support and feedback that he provided to me on the behalf of the case company. I want to thank my superior, M.Sc. (Tech.) Ilkka Toivanen for proofreading the thesis. A special thank you is reserved to the maritime vessel control system developers who took part in the semi-structured interviews and the usability testing of the developed remote diagnostic service prototype. Additionally, I wish to thank the two colleagues who helped me in the setting up of the programmable logic control hardware that was used in the internal testing of diagnostics data communication. Lastly, I wish to thank everyone else who supported me during this undertaking.

Samuli Siitonen
22.06.2020
Lappeenranta, Finland

TABLE OF CONTENTS

LIST OF SYMBOLS AND ABBREVIATIONS

CFAM	Controller-focused Attack Monitoring
CPS	Cyber-physical System
CPU	Central Processing Unit
CS	Control System
ECS	Energy Control System
EMS	Energy Management System
FBP	Flow-based Programming
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
I/O	Input/Output
IAS	Integrated Automation System
IP	Information Packet
JSON	JavaScript Object Notation
MCS	Marine Control System
MQTT	Message Queuing Telemetry Transport
OWEC	Ocean Wave Energy Converter
PLC	Programmable Logic Controller
PMS	Power Management System
PTO	Power Take-off
RD	Remote Diagnostics
RDS	Remote Diagnostics Service
RO	Research Objective
RQ	Research Question
SFAM	System-focused Attack Monitoring
SSC	Successful Completion Criteria
UI	User Interface
URV	Underwater Robotic Vehicle
VPN	Virtual Private Network
XSS	Cross-site Scripting

1 INTRODUCTION

The motivation for authoring this thesis is based on a potential need for a remote diagnostics service (RDS) in a case company operating in the global maritime vessel control systems market. An RDS implementation could provide cost savings to the company based on more effective use of control system developer time during the performing of system diagnostics activities. The case company's customers could benefit from the inclusion of RDS support in their vessels through potential vessel operability improvements; preventive control system maintenance and more effective use of system developer time could reduce the amount of time that vessels need to be out of operation during maintenance and problem troubleshooting.

This chapter is divided into the background, goals and delimitations, and structure of the thesis subchapters. The background chapter describes today's maritime vessels, characteristics of vessel automation systems, and how these systems relate to remote diagnostics. Goals and delimitation chapter presents the research questions and objectives coupled with the high-level steps to be completed in answering the research questions. Delimitations regarding the testing of the developed RDS prototype, and the depth of cyber security considerations are also described. The structure of the thesis chapter provides a high-level view into the contents of each remaining main chapter.

1.1 Background

Today's advanced high-end maritime vessels can be characterized as having complex operational functions coupled with high number of input and output connections (Rensvik, Sørensen et al. 2003, 1-2). The operation and automation of such vessels is achieved through the implementation of integrated automation systems (IAS). These systems incorporate vessel automation, power management, and process automation systems in addition to other complex systems (Rensvik, Sørensen et al. 2003, 1-3) required in vessel operation.

The control and monitoring systems associated with modern IAS implementations can be characterized as complicated and sophisticated, which may work against the vessel operators' ability to maintain vessel operability. RDS implementations that work on collected control and monitoring system data may be deployed to mitigate this; immediate expert assistance in problem troubleshooting, early detection and warning of defects, and enablement of proactive maintenance are some of the potential advantages associated with the use of RDS implementations. (Pepliński 2019, 111)

Based on the apparent diversity of modern vessel designs and control systems, it would seem beneficial for RDS implementations to accommodate configurability in the forming of data representations and visualizations. Flow-based programming's (FBP) ability to support changing workflows (e.g. transformations on diagnostics data) through the graph-like linking of modular data transformation processes would seem to make it a suitable solution for supporting this configurability.

1.2 Goals and delimitations

The goal of this thesis is to find out maritime vessel control system developer preferences regarding the indication of vessel control system status in an RDS implementation. This goal is achieved through the conducting of system developer semi-structured interviews in addition to the design, implementation, and usability testing of an RDS prototype. The analyzation of system developer interview results functions as a basis for the design of this prototype.

Applicability of FBP concepts in marine remote diagnostics is tentatively gauged as a secondary interest during the developer interviews. The driving force behind this is on the forming of a basis for potential future RDS development efforts. It should be emphasized that the focus of the practical work is on the development and validation of an RDS prototype that is developed for the case company.

The following research questions (RQs) are answered:

1. What maritime vessel control system values indicate vessel status or are otherwise important to control system developers during control system diagnostics activities?
2. What form of a remote diagnostics service user interface is preferred by maritime control system developers for indicating vessel control system status (through the presentation of control system values) during system diagnostics activities?

The following research objectives (ROs) are met through the answering of the RQs:

1. To find out what control system values indicate vessel status or are otherwise important to maritime vessel control system developers during the conducting of control system diagnostics activities.
2. To find out how maritime vessel control system values should be indicated in a remote diagnostics service user interface so that control system developers are able to view the system status during the performing of system diagnostics activities.

The high-level steps to be completed during the answering of the RQs are shown in Figure 1. Here the perceptions and opinions of the maritime vessel control system developers function as the main driver for the RDS prototype design and implementation work. Applicability of the designed and created RDS UI for indicating control system status in remote diagnostics activities is later evaluated through the usability testing of the RDS prototype.

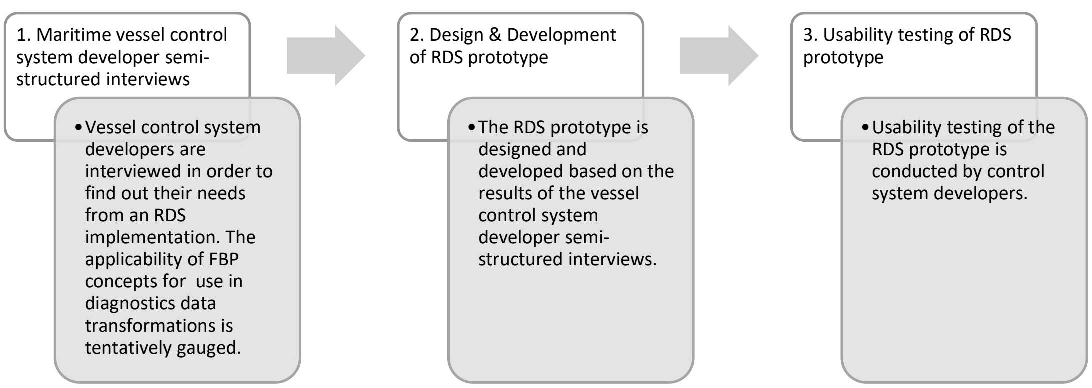


Figure 1. High-level steps to be completed during the answering of the research questions.

The RDS prototype is tested locally in connection with a Beckhoff GmbH programmable logic controller (PLC). The PLC is executing a mock implementation of a vessel control system configuration. Diagnostics data is communicated from the PLC to the RDS local web server through a physical ethernet connection. This data transfer functionality is not part of the usability testing conducted by system developers, and it is intended as a potential future basis for a fully-fledged RDS implementation for the case company. The RDS prototype will be populated with imaginary control system data during usability testing.

Cyber security issues relating to vessel control system remote network access are briefly considered as a basis of discussion for the case company. Cyber security of other parts of an RDS implementation are tentatively discussed but the emphasis is on vessel control systems external access.

1.3 Structure of the thesis

This thesis is divided into the following chapters: introduction, control systems, marine remote diagnostics, research process, results, discussion and conclusions, and summary. The control systems chapter introduces general control systems and marine control systems. The marine remote diagnostics chapter introduces general remote diagnostics and marine remote diagnostics. The research process chapter describes the planning of the semi-structured maritime vessel control system developer interviews and usability testing of the RDS prototype by the system developers. The results chapter describes the findings of the research process. The discussion and conclusions chapter contain analyzation of the research results in addition to their synthesis and presentation of future research possibilities. Results and conclusions of the research are summarized in the summary chapter.

2 CONTROL SYSTEMS

According to Groover (2015, 78), control systems (CS) are one of the three basic elements of automated systems together with programs of instructions and power sources. These systems can be divided into closed-loop and open-loop types based on the existence of a feedback loop for the system output variable (2015, 84-85).

This chapter is divided into the following subchapters: control systems as an element of automated systems, marine control systems, and marine control systems cyber security. The aim here is to provide a general description of control systems in the automated systems context, and then describe marine control systems as a type of control system. The marine control systems cyber security chapter function as a basis for the remote diagnostics cyber security concerns presented in chapter 3.2.

2.1 Control systems as an element of automated systems

The main task of control systems is to execute programs of instruction in order for a given process to fulfill its purpose (Groover 2015, 84). These programs can be described as processing steps that denote the actions that an automated process performs during its operation. The processing steps are contained within work cycles based on the contents of work cycle programs. Moreover, the work cycle programs can be divided into the following categories based on their complexity: set-point control (use of a single process parameter), logic control (process parameter has dependencies to other variables), sequence control (process parameter value is dependent on time), interactive program (human-based control system interaction), and intelligent program (control system is able to perform decision making, learning, and other more demanding tasks typically associated with human-like behavior). (Groover 2015, 80-81)

In an automated system, the control system, programs of instruction, and process are all powered by a power source. It is typical for the power systems powering different control systems to output power in the form of electricity. Advantages of this are that electricity may be converted to other forms, it is usually available for a relatively reasonable cost, it has low power signal transmission capabilities, and that electricity may be stored in batteries for future use. (Groover 2015, 78)

According to Groover (2015, 84), the controls in automated systems can be either closed-loop or open-loop systems. In a closed-loop system, the output variable of a process is compared against the controller input variable; this difference between values is used by a controller to adjust a given process. As its name suggests, open-loop system controllers do not perform input/output comparisons, but rather a model of the controller-based effects on the process is used as guidance. (Groover 2015, 84-85)

Skjetne and Egeland (2006, 240) define control systems based on the performing of a required control function; control systems encapsulate all other systems, components, hardware, software, and user interfaces (UI) that are needed to perform such a function. Example control system subsystems presented by Skjetne and Egeland include power, actuator, sensor, and control computer systems. The definitions of these subsystems are presented in Table 1 where the different, but complementing characteristics of each system would seem noteworthy; a single subsystem can be considered as a part of a larger whole working towards a common system goal.

Table 1. Control system subsystems based on Skjetne and Egeland (2006, 240-241).

Subsystem	**Definition**
Power System	Electric power supplying systems and equipment for essential consumer units.
Actuator System	Physical effort/action supplying components and subsystems. Cause the plant (physical process) to behave as desired.
Sensor System	Hardware, software, and algorithms of the measurement equipment in addition to equipment used in signal communications.
Control Computer System	CPU-processing and I/O capacity equipped processors/computers together with operator stations and power supply units. Signal communication support, HW/SW platform and its associated controllers.

2.2 Marine control systems

According to Sørensen (2005, 125), marine control systems (MCS) can be defined as the science of techniques and methods used in the analysis, monitoring, and control of marine systems. Sørensen presents the following marine industries as the main application fields of MCS:

- Sea Transportation
- Offshore Oil and Gas Exploration and Exploitation
- Fisheries and Aquaculture

Sørensen (2005, 125) divides control system structures into operational and business enterprise management, and real time control and monitoring areas. The operational and business enterprise management area is housed within office systems that are connected to the real time control and monitoring area. Fleet and ship operational management is housed within office systems, while ship local optimization, plant, and actuator control are housed in the real-time control area. Integration of these areas can be defined as industrial IT. (Sørensen 2005, 125-126)

According to Skjetne and Egeland (2006, 241-242), marine control systems can be placed into a three-tiered hierarchy where the control system is situated at the top, its subsystems in the middle, and various components at the bottom of the hierarchy. Here the control system is connected to different subsystems (e.g. the power and actuator systems defined in Table 1), which in turn are connected to specific groups of components (e.g. the power system is connected to power equipment units). The control system at the top of this hierarchy is usually one of many such systems in marine vessels, and they may be integrated into marine automation systems (Skjetne, Egeland 2006, 242).

A vessel control system may for example comprise of a power management system (PMS), a human-machine interface (HMI), PLCs, and communications busses (Sørensen 2013, 10-11). The PMS may further be contained within an energy control system (ECS) that also handles energy management through the energy management system (EMS) (Sørensen 2013, 22).

Some examples of advanced marine control system designs have been presented by Shi et al. (2017) in the context of the following six different types of marine mechatronic systems: ocean wave energy converters (OWECs), profiling floats, underwater gliders, surface vessels, offshore floating wind turbines, and underwater robotic vehicles (URVs). Example control systems belonging to each of these types are presented in Table 2. The dynamic positioning control system example presented in Table 2 would seem to be an area of interest in the marine control systems -related research based on the relatively high number of articles available compared to other marine control systems research. Other general examples of marine control systems include the roll motion reduction and trim systems presented by Ertogan et al. (2018) in their marine control systems case study.

Table 2. Examples of advanced marine control systems belonging to different types of mechatronic systems based on Shi et al. (2017).

Type of mechatronic system	Example control system(s)
Ocean wave energy converters (OWECs)	Power take-off (PTO) systems.
Profiling floats	Buoyancy control.
Underwater gliders	Navigation and maneuvering.
Surface vessels	Dynamic positioning systems for keeping vessel position and heading fixed to a point.
Offshore floating wind turbines	Control of turbine blade pitch and generator torque.
Underwater robotic vehicles (URVs)	Systems for path following, tracking of trajectory and dynamic positioning.

2.3 Marine control systems cyber security

The goal of cyber security can be defined as the protection of data and assets confidentiality, integrity and availability as these data and assets are used in cyber space. This goal can be achieved by following approaches and actions that are associated with security risk management processes. (Schatz, Bashroush et al. 2017, 66)

Skjong et al. (2016, 534) divide cyber-attacks aimed at marine vessels into external and internal forms. External cyber-attacks may be considered particularly interesting from the viewpoint of RDS cyber security. The connecting (at least as output-only) of a vessel control system to a cloud hosted web server requires that the control system has some degree of internet connectivity. This is a cyber security risk to be acknowledged and managed carefully if an RDS implementation is to be used in connection with vessels operating in real life conditions.

Skong et al. (2016, 534) present multiple strategies that may be used in protecting a vessel from external cyber-attacks. In this context, they mention a network switch that can be thought of as an access point between the vessel and the rest of the world. A basic security feature of such a switch is to provide authentication and virtual private network (VPN) capabilities. Further security may be provided by limiting the network communication to output only together with the use of data encryption. The limiting of router input and output ports are also mentioned as ways of reducing the risk of internal cyber-attacks (e.g. malware insertion by a passenger or even a crew member) by Skong et al. (2016, 534)

Khorrami et al. (2016) describe real-time attack monitoring and threat mitigation mechanisms in the context of complex cyber-physical systems (CPSs). According to them, process aware approaches to CPS cyber security can be used in conjugation with general purpose computer and network security approaches. These process aware approaches can be used to detect changes in the dynamic behavior of the process that could not be detected using the more traditional approaches such as code execution monitoring; an example case for this are attacks that change controller parameters. (Khorrami, Krishnamurthy et al. 2016, 76)

The process aware approaches aim at detecting cyber-attack –based effects in the behavior of close-loop control systems. Khorrami et al. mention dynamic state variable values in the control algorithm as an example of something that could potentially be out of reach of process aware monitoring. To distinguish the effects of state changes from cyber-attacks, a two dynamic monitoring system is proposed. This system is composed of a controller-focused attack monitor (CFAM) and a system-focused attack monitor (SFAM) that monitor the dynamic process and controller feedback loops separately. (Khorrami, Krishnamurthy et al. 2016, 78-79)

3 MARINE REMOTE DIAGNOSTICS

Jonsson and Holmström (2005, 154) describe remote diagnostics (RD) as a type of ubiquitous computing where sensors installed on physical products communicate performance parameters using network access. Here RD is presented as an enabler of continuous product performance and timely maintenance. The placement of RD into ubiquitous computing is based on its perceived embeddedness and mobility; sensors are embedded into products, and either the product or its remote monitor may be mobile (Jonsson, Holmström 2005, 155).

In the vehicle context, Kuschel and Ljungberg (2005, 211) have described RD as the remote accessing, diagnosing, and programming of vehicle control and support systems. They also note that RD does not have an agreed upon definition, but it may be considered as a part of telematics; RD is concerned with the part of telematics that deals with remote diagnosing and solving of problems. It should be however noted that Kuschel and Ljungberg indicate that telematics itself is lacking an agreed upon definition, but it can be understood as services that are enabled by vehicle wireless communications. (Kuschel, Ljungberg 2005, 212-213)

According to Pepliński (2019, 111), the use of RD systems in marine vessels has benefits for both the owners and crews of vessels. The crew benefits include early defect warnings that support undertaking preventative actions, enablement of proactive maintenance, and troubleshooting assistance from engineers and experts. The benefits for vessel owners are presented in the context of improved profitability: early warnings of failures lower service costs, and vessel performance is increasement through reduced fault-finding durations. These are in addition to the expected maximization of uptime, performance, and profitability of vessels.

This chapter is divided into the marine diagnostics services and marine remote diagnostics cyber security chapters. Here the aim is to first introduce diagnostic services in the marine context and then move onto the discussion of solutions to cyber security issues raising from communicating vessel diagnostics data through the internet to an RDS web server.

3.1 Marine remote diagnostics services

A prototype interactive webpage for marine remote diagnostics developed by Zorin et al. (2016) may be considered as a partial example of an RDS implementation. The prototype was tested on support center employees and it had the following main components: dashboard, map and fleet, and analytics tools.

The prototype's dashboard was constructed as a user-customizable starting point into the prototype that showed an overview of the most important data. The map tool contained interactable rectangles with information drill-down capability which denoted vessels and their operating state. The fleet tool allowed users to quickly view vessel detailed information in addition to having access to a more detailed and modifiable listing of all vessels with information drill-down capability. The analytics tool made it possible for multiple people to collaborate on a customizable canvas where data sources could be connected with drag-and-drop -ready widgets that could then be used to visualize data. (Zoric, Domova et al. 2016)

Examples of commercial marine remote diagnostics implementations from ABB, Kongsberg and Valmet companies are presented in Table 3. Each of these implementations support remote diagnostics in some form, but ABB's RDS4Marine diagnostics tool seems unique in its approach. This is based on the modular nature of its diagnostics offering for different vessel components and subsystems.

Table 3. Examples of commercial marine RDS implementations.

Company	**Remote Diagnostics Implementation**	**Description**
ABB	ABB RDS4Marine	Marine-tailored diagnostics tool that is used by authorized service engineers in remote troubleshooting, analysis, and monitoring activities. Provides modular diagnostics capabilities for vessel components and subsystems (e.g. a module for vessel propulsion diagnostics). (ABB 2013)
Kongsberg	Kongsberg Remote Services	Tool used by Kongsberg certified engineers in connecting to vessel control systems for remote support. Supports diagnostics (checking of

		live parameters) from offices within the Kongsberg Global Secure Network. (Kongsberg n.d.)
Valmet	Valmet DNA Machine Monitoring	Enables a remote connection to the Valmet DNA system. Supports machinery unit mechanical condition and performance measurement together with associated analysis activities. Provides critical machinery diagnostics. (Valmet 2015)

A high-level view of an example RDS implementation is shown in Figure 2. Here a vessel communicates diagnostics data to an RDS web server using a satellite uplink or cellular network. User computers query the web server for vessel diagnostics data.

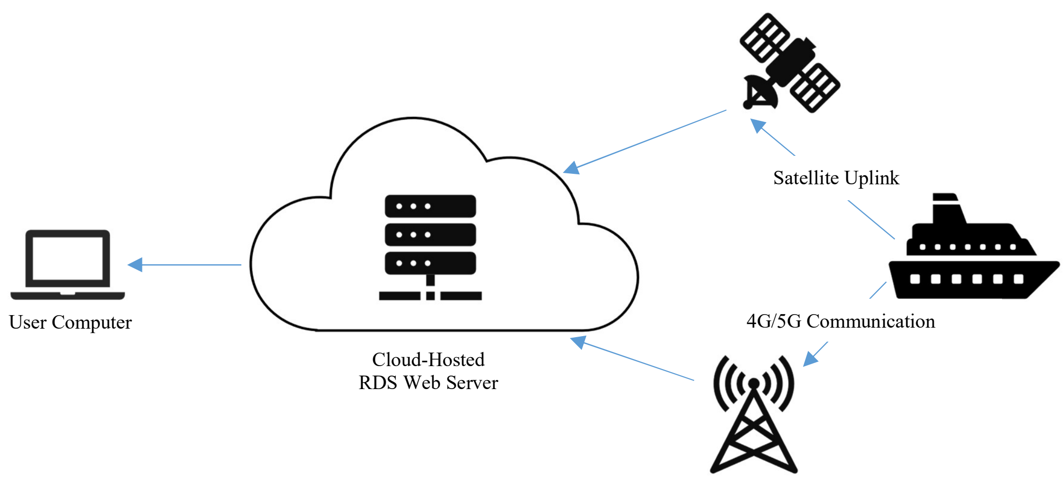


Figure 2. High-level view of an example RDS implementation. Diagnostics data is transferred from a vessel to an RDS web server through a satellite uplink or mobile 4G/5G network. User computers query the web server for gathered diagnostics data.

3.2 Marine remote diagnostics cyber security

As Figure 2 shows, RDS implementations may be perceived as the high-level linking of user computers, web servers and vessel control systems using multiple different networking mediums. Based on this top-down view, the cyber security considerations of RDS implementations could be divided into the following four areas: the user computer (web browser), RDS web server, vessel control system, and network communication.

Chapter 2.3 briefly described control systems cyber security and the external and internal forms of system cyber-attacks. The use of output-only network router was mentioned as an external attack mitigation method. The use of such a switch would seem especially logical in the context of RDS cyber security as diagnostics data is to be gathered from vessels remotely through a network connection. The cyber-security of an RDS-connected vessel could further be enhanced using internal cyber-attack mitigation methods that are also described in chapter 2.3.

The connectivity requirements of an RDS implementation shown in Figure 2 indicate a need for considering information and cyber security at network communications level. Here the use of message encryption would seem to be the minimal requirement in securing confidential diagnostics data. Basic information and cyber security of the RDS web server could be upheld using traditional security methods. These include access authentication (both physical and remote), limiting of open ports and use of up-to-date software. Basic information and cyber security of the user computer (web browser) may be achieved using technologies that prevent cross-site scripting (XSS), query injection and other types of attacks commonly associated with web applications.

4 RESEARCH PROCESS

Semi-structured maritime vessel control system software developer interviews and usability testing of the RDS prototype were selected as the research methods for finding answers to the research question presented in chapter 1.2. As Figure 1 in chapter 1.2 shows, the interviews functioned as a basis for the design and implementation of the RDS prototype. Usability testing of the RDS prototype was conducted to assess the applicability of designed UI representation in indicating system status during the performing of vessel control system remote diagnostics activities undertaken by control system developers.

This chapter begins with an introductory subchapter on semi-structured interviews. The next subchapter introduces flow-based programming concepts as their applicability in remote diagnostics was tentatively gauged during the interviews. A subchapter describing the planning of developer interviews is presented next. Subchapters describing usability testing and the way in which it was planned conclude this chapter.

4.1 Semi-structured interviews

Wilson (2014, 21) describes semi-structured interviews as an interview method that uses structured questions together with unstructured exploration. He bases its usefulness in the gaining of a deeper understanding of complex issues through the asking of probing and spontaneous questions. Use cases for semi-structured interviews listed by Wilson include the gathering of attitudes and opinions, gathering of data relating to behaviors that cannot be observed directly, understanding of user goals, and gathering of information on task flows (Wilson 2014, 24-25).

According to Wilson (2014, 24), semi-structured interviews commonly use an interview guide that contains the following items: a topic and purpose for the interview, a number of topics and their associated questions to be asked, suggestions for interviewee probes and prompts, and comments for closing the interview. An interview guide based on this structure was used in the developer interviews that are further described in chapter 4.3.

Hove and Anda (2005) provide advice on how to handle four challenging areas relating to the planning and conducting of semi-structured interviews in software engineering research.

They base this advice on 280 interviews that were conducted for a total of 12 software engineering studies. These challenging areas are described in Table 4, and the advice regarding the use of appropriate tools and artifacts was planned to be used in the semi-structured maritime vessel control system developer interviews. This meant that graphical illustrations of FBP concepts and RDS UI mockups were planned be shown to the interviewees in order for them to better understand the context of different questions presented to them.

Table 4. Four challenging areas identified by Hove and Anda (2005) in planning and conduction of semi-structured interviews in software engineering research.

Challenging Area	**Relevant Phase**	**Description**
Interview Effort	Planning the Interview	Understanding of the necessary effort required in an interview study. This understanding functions as a basis for resource allocation and interview scheduling.
Interviewer Qualifications	Conducting the Interview	Interviewer must have extensive research area knowledge and good interviewing skills.
Interviewer and Interviewee Interaction	Conducting the Interview	Handling of challenging interviewees and ability to gain answers to sensitive questions.
Use of Appropriate Tools and Artifacts	Conducting the Interview	Elicitation of important information using tools (e.g. audio recording) and artifacts (e.g. pictures relating to the research area and questions).

4.2 Introduction to flow-based programming concepts

The applicability of FBP concepts in marine remote diagnostics was tentatively gauged during the asking of questions belonging to the fourth topic of control the system developer interviews. This topic and its associated questions are presented in subchapter 4.3 together with presentation the of other three topics and their questions. The remaining of this subchapter focuses on providing a succinct introduction to FBP.

FBP is a programming paradigm functioning as a special case of dataflow programming that supports the design and building of applications in a component-oriented manner using a *data processing factory* metaphor. In FBP, information packets (IPs) travel between connected processes (so called *black boxes*) and across networks of these processes in a similar manner that items travel across conveyor belts in a factory. The component orientation of FBP is based on the programmer's apparent ability to form new applications without needing to alter the internals of the processes used. (Morrison n.d)

A simplistic network of connected FBP processes is shown in Figure 3. Here two processes (process A and process B) have their outputs connected to process C inputs. Connections from and to the outside network have been left out for clarity. The flow of information packets between the processes is directed from output to input ports indicated as black and grey circles, respectively.

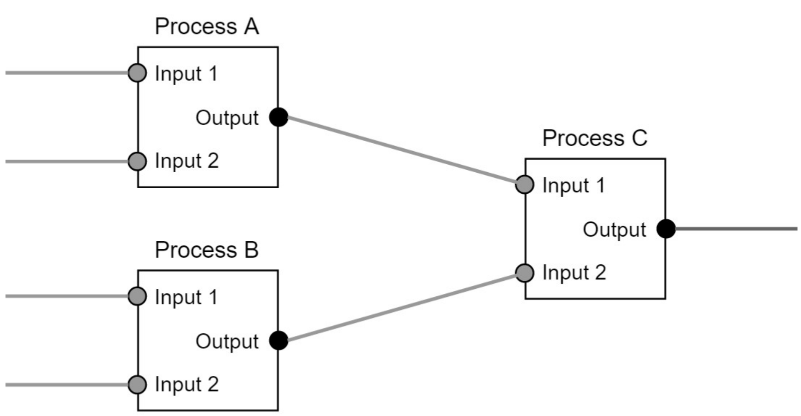


Figure 3. A simplistic network of connected FBP processes.

A more sophisticated process network compared to the one shown in Figure 3 is shown in Figure 4. Here prebuilt processes are used to calculate answers to

$$(x + y) \times (w - z) \tag{1}$$

In this equation, the x, y, w, and z terms are user-configurable parameters. All processes in the figure (designated as Summation, Subtraction and Multiplication) take in two input parameters and output a single output value. Each process performs its calculation after all its input parameters have been received.

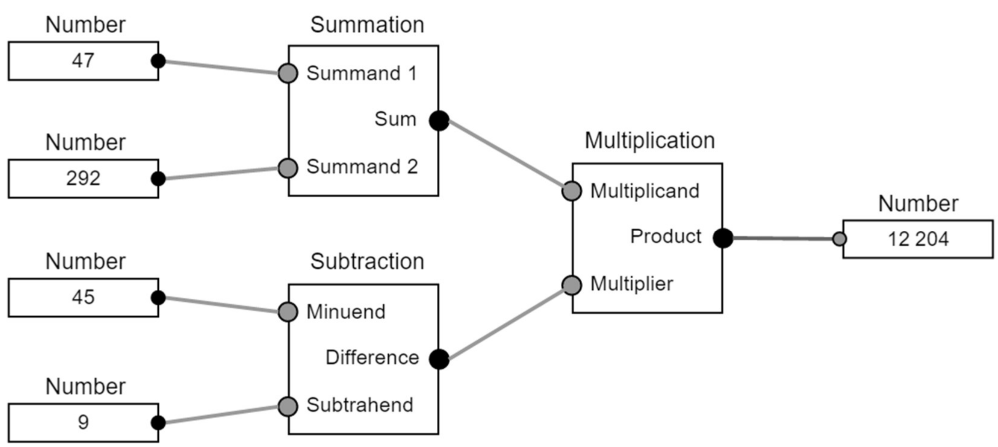


Figure 4. Example FBP-inspired network of a program for calculating (x + y) × (w - z). The parameters x (summand 1), y (summand 2), w (minuend), and z (subtrahend) are user configurable.

FBP-inspired systems do not involve the paradigm shift from the von Neumann computer architecture associated with proper (or classical) FBP. Examples of FBP-inspired systems include NoFlo (Bergius n.d.n.d), Node-RED (JS Foundation n.d) and Flowhub (Flowhub n.d). (Morrison n.d) The usage of FBP concepts in potential future development efforts relating to the RDS prototype would likely results in the creation of a new FBP-inspired system.

4.3 Maritime vessel control system developer semi-structured interviews

Originally the main goal of the maritime vessel control system developer interviews was centered on finding out how FBP concepts could be applied in marine remote diagnostics activities. The practical testing of these concepts' applicability was deemed to be out-of-scope for this work based on the amount of development effort required in it on top of the RDS prototype implementation work. The new goal of this work was centered on finding out what control system variable values and value representations are relevant to control system developers during control system diagnostics activities as an indication of system status. The gathering of RDS prototype functional and non-functional requirements was a secondary goal from the beginning and it was left untouched.

The selection criteria for interviewees was that each interviewee had to have previous experience with real-word maritime vessel control systems development. This criterion was set to focus the interview efforts in gathering data bound to real-world experiences where the use of remote diagnostics could have made the vessel software maintenance process more effective.

The interview guide used in the interviews is presented in Table 5 based on an rough outline provided by Wilson (2014, 34). The guide divides each interview into three main activities. Each activity has a short description together with an initial time estimate for its completion.

Table 5. Interview guide for the maritime vessel control system developer semi-structured interviews. Based on an outline by Wilson (2014, 34).

Activity	**Description**	**Time Estimate**
Brief the Interviewee	Provide a context for the interview. Inform the interviewee regarding the use of interview data: notes are taken and analyzed in order to form a basis for the design and implementation work, collected data is handled anonymously, the data is not connectable to individuals, and the interview notes are destroyed after the completion of the master’s thesis. Describe how the interview is to be conducted. Gather verbal or written consent from the interviewee for taking part in the research.	10 minutes
Go through the Topics and	Follow the list of topics and their associated questions. Note down interviewee answers and any	30 minutes

Associated Questions	relevant details. Expand on questions and interviewee answers when there is potential for gaining valuable information.	
Conclude the Interview	Thank the interviewee for their time and responses. Note down any missing details.	5 minutes

4.3.1 Interview topics and questions

All interview participants were Finnish speakers and thus the interviews were conducted in Finnish. The Finnish format interview topics and their associated questions are presented in Appendix 1. The remaining content of this subchapter focuses on describing the English format topics and questions.

The interview topics were ordered in a sequence that began with the gathering of background information regarding the conducting of traditional diagnostics activities undertaken by the control system developers while in the service of the case company. The potential of remote diagnostics was approached next, after which the interviews moved into finding out the relevancy of different control system variables in remote diagnostics. The interviews were concluded with the asking of potential variable value transformation possibilities through the usage of FBP concepts. The following topics were discussed in order:

1. The traditional vessel control systems diagnostics approach
2. Remote diagnostics for vessel control systems
3. The role of vessel control system variable values
4. Possible transformations on vessel control system variable values

Questions belonging to the first topic (The traditional vessel control systems diagnostics approach) were asked to gather background information regarding traditional diagnostics activities. The following questions were asked:

1. How are vessel control system software issues diagnosed ordinarily?
2. What tools are used in the diagnostics work?
    a. Could you describe how that tool is used?
    b. What data is provided by the tool?
3. What is the most difficult part of the diagnostics work?
    a. What is the cause for this?

4. What is the most time-consuming part of the diagnostics work?
    a. What is the cause for this?
5. What is the most diagnosed issue?
    a. How is this issue diagnosed?

Questions belonging to the second topic (Remote diagnostics for vessel control systems) were asked to gather developer perceptions of remote diagnostics and their expectations from diagnostics solutions. The following questions were asked:

1. What does the term *remote diagnostics* mean to you in the vessel control systems context?
2. What effect could remote diagnostics have on your work if it would be available?
    a. What problems could it be used to solve?
    b. Can you think of any previous situations where the usage of remote diagnostics could have been beneficial?
3. What would be your requirements from a remote diagnostics solution?
    a. How often would the diagnostics data need to be updated?
    b. Would you use the solution to view live data, or should previously gathered data be accessible?
    c. How customizable should the solution need to be to different control systems or problematic situations?
    d. What could cause you to use the traditional diagnostics approach instead of a remote diagnostics solution?

Questions belonging to the third topic (The role of vessel control system variable values) were asked to gather developer opinions on the most important diagnostics data (control system variable values) to be used in diagnostics activities. The following questions were asked:

1. What are the most important variable values that should be accessible remotely during diagnostics activities?
    a. What variables provide information on the vessel overall condition?
2. How often would the variable values need to be refreshed if they were accessible remotely?

    a. Are there be some variable values that need to be refreshed less or more often?
3. Based on your experience, what variable values could be used to diagnose the most common issues?

Questions belonging to the fourth topic (Possible transformations on vessel control system variable values) were asked to gather developer opinions on the transformation of variable values into potentially easier to understand or otherwise more suitable forms in the remote diagnostics context. The basis for this topic was in the tentative gauging of FBP concepts' applicability in remote diagnostics. The following questions were asked:

1. Can you think of any cases where it would benefit the diagnostics activity to transform system variable values into another form (data and its representations) based on computations?
    a. Can you think of any existing transformations that you are using already?
2. How likely would you be willing to graphically "pipe" the variable value transformations into results?
    a. If "pipeable" transformations would be offered, then what transformations would likely see the most use?
    b. If there was a choice between the use of transformations and the viewing of raw variable values, which one would you prefer to use and to what degree (overall percentage of times)?

4.3.2 Role of the interviews in enabling RDS prototype design and implementation work

The control system developer interviews functioned as a basic for the design and implementation of the RDS prototype. Developer perceptions and requirements from the RDS implementation were taken into consideration in the prototype design work. The role of variable value data transformations and how FBP processes could be deployed in this context was a secondary consideration that did not affect the design and implementation work.

The results chapter describes this practical work in more detail together with the presentation of interview results. Usability testing of the RDS prototype was conducted based on the results of the practical work.

4.4 Usability testing

According to Rubin and Chisnell (2008, 21) usability testing can be understood as a research tool and a process. In the usability testing process, the target users of a product function as testing participants in the evaluation of how well the products meets usability criteria (Rubin, Chisnell 2008, 21). Rubin and Chisnell (2008, 64) describe the following steps within this process: test plan development, testing environment setup, finding and selecting of test participants, conducting of the testing sessions, debriefing the test participants and observers, analyzation of data and observations, and reporting of the findings and recommendations.

Rubin and Chisnell (2008, 66-67) describe the following typical sections of a test plan: purpose, goals and objectives for the tests, research questions, characteristics of the participants, the testing method, task list, test environment equipment and logistics, role of the test moderator, data to be collected and its evaluation measures, and the contents of the report together with its presentation. They also note that there is room for changing the composition of a test plan based on the test type used and previously set formality requirements.

In the context of usability testing methods, Rubin and Chisnell (2008, 75) describe the within-subjects method. In this method, each usability testing participant performs the same test tasks using the product that is being tested. Rubin and Chisnell note that there exists a possibility that the testers learn the product being tested which may affect the results of later tests (this is called the transfer of learning effects). This may be mitigated through the randomization of the order of tasks to be performed by the testers (this is called counterbalancing). (Rubin, Chisnell 2008, 75)

4.4.1 RDS prototype usability testing plan

The usability testing plan of the RDS prototype was created based on the sections of a testing plan as described by Rubin and Chisnell (2008, 66-67). The research questions section was replaced with a listing of testing objectives that were intended to be met for the research question 2 presented in chapter 1.2 to be answered. This choice was made to focus the research efforts in finding answers to the second main research question of this work. Additionally, the usability testing report section was chosen to be left out as the result of the testing will be presented in the results chapter of this work.

The remaining of this subchapter presents the sections of the RDS prototype usability testing plan as separate subchapters. The answering of the research question 2 presented in chapter 1.2 acted as the driving force behind the creation of this plan.

4.4.1.1 Purpose, goals, and objectives

The applicability of the designed and implemented RDS prototype's UI for communicating system status in remote diagnostics is to be gauged through its usability testing. The testing functions in validating the requirements gathering and design choices made based on the control system developer semi-structured interviews. Additionally, the gaining of further insight on system status indication possibilities is expected based on test participant feedback. This process is intended to lead to the answering of research question 2 presented in chapter 1.2.

4.4.1.2 Testing objectives

The testing objectives to be met through usability testing are based on validating and further detailing a potential answer to research question 2 presented in chapter 1.2. The following testing objectives were formed for the usability testing plan:

1. Find out if the users naturally use the values and events search functionality when they look for values and events on specific dates and ranges, or if they prefer to manually navigate on the vessel timeline using the playback rewind and fast-forward controls in addition to the date picker control.
2. Find out if the search and focusing of a specific control system event on the vessel timeline using the events search functionality is apparent to the users.
3. Find out if the search and focusing of control system value range timestamps on the vessel timeline using the values search functionality is apparent to the users.

4. Find out if users able to tell what value/event is indicated by control system value/event timestamp when search results are being focused upon on the vessel timeline.
5. Find out if users face difficulties in differentiating control system event indicators from value graphs.

4.4.1.3 Participant characteristics

All test participants are to be to be maritime vessel control system developers with previous experience in developing vessel control systems. The pool of semi-structured interview participants is intended to be used as a basis for selecting the participants. The limited number participants are to be promised anonymity and thus their characteristics are not specified.

4.4.1.4 Testing method

Exploratory (formative) testing as it is described by Rubin and Chisnell (2008) is selected as the testing method based on its fittingness to the early phase of product development cycle. This meant that the tests are to follow an informal approach where the verbal user and moderator interaction is to be conducted freely. The asking of feedback on the prototype implementation as a basis for its further design and development is intended.

The usability testing is to be conducted remotely based on reducing the risk of spread of the Covid-19 coronavirus global pandemic of 2020. Users are to be provided with a digital copy the RDS prototype UI implementation that house imaginary (mock) system data. It is planned that users will share their screens using the Microsoft Teams application's screen sharing functionality during a voice call that is to be used to conduct the testing remotely.

4.4.1.5 Task scenarios (task lists)

Two task scenarios and their associated tasks are to be completed by each test participant. The first scenario represents the beginning steps of system diagnostics based on shipyard feedback. The second scenario acts as a continuation of diagnostics activities where an initial guess regarding the possible cause for an issue occurring needs to be further validated.

Both task scenarios are designed to be completed based on the presentation of mock control system data with pre-determined values and events for target dates and time periods. The tasks scenarios will be read aloud to the test participants during testing. Each participant is to begin the completion of tasks from the same view where the control system timeline is zoomed out and basic vessel information is shown. The tasks are to be completed in order with no data or UI resets between them. The UI shall be reset after the completion of scenario 1. The result of completing the tasks in the task scenarios is that the user could gain information regarding the behavior of system values as they relate to a control system problem occurring. At a high level, this would mean that the user is able to inspect the control system status in multiple points of time.

Scenario 1

An issue report from a shipyard is delivered to you (the control system developer). The report indicates that something has gone wrong with the vessel control system operations on Sun Jun 14, 2020 between 13:00 and 13:30. Please find out if a control system error event has been logged during that time. If an even is found, check the value of *variable0* when the event occurred.

Scenario 1 tasks

1. Find out what error events have occurred on Sun Jun 14, 2020 between 13:00 and 13:30. Successful completion criteria (SCC): user was able to view the events listing for Sun Jun 14, 2020 13:00 - 13:30.
2. Find out the value of control system variable *variable0* when the even occurred. SCC: user was able to read out aloud the value of a *variable0* at the point of time when the event occurred.

Scenario 2

Based on previous diagnostics activities, you have understood that the cause for the error events could be that *variable0* reaches the value threshold of 30. Find out how many times this has almost occurred on Sun Jun 14, 2020 by checking how many times *variable0* has had a value between 20 and 29 (inclusive). Check if any error events have been logged 15 minutes prior or after *variable0* value has been within this range.

Scenario 2 tasks

1. Find out how many times *variable0* values have come close to the value threshold. SCC: user was able to read out aloud how many times the value of *variable0* has come close to the specified value range.
2. Find out if any error events have occurred just after *variable0* value has come close to the specified threshold. SCC: user was able to conclude that one error event (the event from scenario 1) has occurred in a 15-minute range when the value of *variable0* was close to the value threshold.

4.4.1.6 The test environment

As the tests are to be conducted remotely, the test participants are expected to be situated at their home offices. Company and personal hardware are to be used by the participants to test the RDS prototype. A version of the Google Chrome web browser is to be used to run the prototype's UI.

4.4.1.7 Role of the moderator

The moderator introduces the testing session and monitors the users as tasks are completed. Moderator assistance in progressing scenarios is provided only as a last resort. The moderator communicates with the participants through a Microsoft Teams application's voice call.

4.4.1.8 Data collection

Performance and preference data as described by Rubin and Chisnell (2008) is noted down in addition to the noting of miscellaneous verbal feedback. Additionally, the participants are asked at the end of each testing session if the current implementation of the RDS prototype would be useful in determining the status of a vessel control system at a point in time.

Performance data

1. Total times that a participant was not able to proceed without guidance.

Preference data

1. Participant preference on manually navigating the vessel timeline instead of using the search functionality.
2. Participant ability to understand and detect the focusing of values or events on the vessel timeline based on event or value search result selection.
3. Participant requests for more functionality.
4. Participant request to changes in functionality.

5 RESULTS

As is described in chapter five, the research process consisted of semi-structed maritime vessel control system developer interviews and usability testing of the created RDS prototype's UI. Results of the developer interviews were used as a basis in the design and creation of the RDS prototype. The applicability of the designed and implemented RDS UI in indicating maritime vessel control systems status was tentatively gauged through the usability testing of the RDS prototype.

This chapter begins with the presentation of maritime vessel control system developer semi-structured interview results in subchapter 5.1. The functional and non-functional requirements of the RDS prototype based on the developer interviews are presented in subchapter 5.2. The design process of the prototype is the content of subchapter 5.3. The prototype's implementation process is presented in subchapter 5.4. Usability testing results for the prototype are presented in subchapter 5.5. A summary of the research results in subchapter 5.6 concludes this chapter.

5.1 Maritime vessel control system developer semi-structured interview results

The maritime vessel control system developer semi-structured interviews consisted of a total of four topics and their associated questions. These topics and questions were described at detail in chapter five. Certain questions included premade prompts (sub-questions) that were asked when appropriate. Further clarification to answers were asked through the asking of improvised questions when the opportunity for this became apparent.

A total of six control system developers with previous experience in working with real-world maritime vessel control systems were interviewed. The interview guide presented in Table 5 found in chapter 4.3 was applied in each interview. The interview questions presented in chapter 4.3.1 were asked from all interviewees in Finnish based on the Finnish-format interview questions available in Appendix 1. Both written and verbal interview consent alternatives were provided to all interviewees. Each interviewee chose to give verbal consent for taking part in the interviews. Verbal consent was asked based on the contents of the Finnish-format interviewee consent form presented in Appendix 2.

All interviews were conducted remotely. The remote nature of the interviews was based on limiting the risk of spread for the Covid-19 coronavirus global pandemic of 2020. Slack desktop application's voice call and screen sharing functionality was used in one interview and the remaining five interviews were conducted using the Microsoft Teams web application's voice call and screen sharing functionality. The screen sharing functionality of both applications was used to show interviewees supplementary material during the answering of questions belonging to the fourth interview topic.

The results of the interviews indicated that there is a need for an RDS implementation that supports the inspection of control system variable values graphically as functions of time. The gauging of developer opinions regarding the use of FBP concepts in diagnostics activities resulted in the conclusion that there is potential benefit for the use of variable value transformations in certain cases.

Answers to the interview questions are presented on per topic basis in subchapters 5.1.1 – 5.1.4 in a summarized format. Effort has been made to indicate the relevancy of answers through the mentioning of how many interviewees gave similar answers or expressed similar opinions. It should however be noted that no other answer relevancy meter (e.g. the mentioning of developer seniority) is included as the interviewees were promised anonymity during the asking of participation consent.

5.1.1 The traditional vessel control systems diagnostics approach

As is described in chapter four, the first interview topic, the traditional vessel control systems diagnostics approach, functioned in gaining background information relating to the traditional way of conducting control system diagnostics activities. Summarized answers to each of the five questions belonging to this topic are laid out below on per question basis.

Question 1. How are vessel control system software issues diagnosed ordinarily?

In answering this question, three interviewees talked about the diagnostics process as it occurs between the case company and its customers. The role of a shipyard as a mediator between a vessel operator and the case company was mentioned in this regard. It was said that a shipyard may act as a middle point in delivering vessel operator issue reports to the case company, but in some cases a more direct form of communication is exercised. The issue reports can be considered as a starting point for solving a problem, and they may contain supplementary material such as videos and log files in addition to a base problem description.

At times more information must be requested for a more complete understanding of the problem to be attained. Two interviewees denoted this process of asking for more information as “detective work”, which becomes difficult when the necessary information is not available. One interviewee said that this difficulty leads to the need for the control system developers to travel to vessels in person. Additionally, one interviewee noted that misunderstandings are a real possibility when a vessel operator is directly communicated with.

Three interviewees mentioned the need to travel to shipyards as a part of the diagnostics work. Two of these interviewees mentioned the need for developers to connect directly to vessel PLCs through the development environment after arriving at the vessel. In relation to the development environment, one interviewee said that the previously attained problem description works as a starting point that helps in pinpointing source code that is thought to have something to do with a particular problem occurring.

One interviewee mentioned that sometimes a shipyard can provide system-level logs that may be inspected during the troubleshooting process. These logs were said to help in investigating what has transpired at which points in time during a vessel’s operation.

Three interviewees mentioned that the PLC software development environment has previously been used to connect to PLCs using a VPN. The possibility for creating VPN connections was said to be dependent on the permissions and hardware support of individual shipyards. One interviewee described the use of the development environment in conjugation with a VPN as taxing, and it was said that this could be worsened by the latency of the VPN connection. It should however be mentioned that the interviewee also told of an example case where this form of remote diagnostics proved successful in troubleshooting a vessel control system issue.

Question 2. What tools are used in the diagnostics work?

The PLC software development environment was characterized as a diagnostics tool by all six interviewees. Its usage was described as the inspecting of program source code and program states with the goal of finding a problem's root cause. The development environment was said to house good debugging functionality by one interviewee. Another interviewee directly mentioned the development environment's scope functionality as a tool used in diagnostics work. It was said to support the monitoring of multiple variable values and their changes relative to each other as a function of time on a graph. VPN capable routers, when available at vessels, were said to be used in conjugation with the development environment in remote troubleshooting.

Four developers mentioned the usage of signal monitoring and debugging tools in different contexts. These have functioned in viewing system network signals but also device signals have been monitored using custom software. Usage of screen sharing software was mentioned, and email was noted as a communication tool.

The use of developer eyesight was mentioned as a form of a diagnostics tool by one interviewee. This was based on the need to view system status lights as they might be helpful in problem troubleshooting.

Question 3. What is the most difficult part of the diagnostics work?

The finding of a problem's root cause was mentioned by two interviewees as the most difficult part of the diagnostics work. The investigation of which series of events lead to the problem occurring was mentioned by two interviewees in the same context as a form of "detective work".

The need to consider multiple potential sources for a given problem was mentioned by three interviewees as a particularly difficult part of the diagnostics work. In this regard, one interviewee noted the need to consider vessel hardware devices in addition to software source code. Additionally, the need to consider multiple layers of software was mentioned as a software-related challenge.

Lack of tools makes things more difficult according to one interviewee. It was also mentioned that individual values must be checked and monitored manually by sight as one tries to reintroduce a problem during debugging.

One interviewee mentioned the gaining of vessel control system access as a difficulty; a potentially high amount of traveling may be required in achieving physical access to a vessel. This was coupled with the amount of effort it takes to negotiate with shipyards for the systems developers to gain access to vessels at opportune times. One interviewee noted the challenge of gaining accurate enough information from vessel operators relating to a problem to be solved as a difficulty. This was coupled with the possibility for receiving factually incorrect information.

Question 4. What is the most time-consuming part of the diagnostics work?

Three interviewees mentioned the finding of a problem's root cause (e.g. the checking of multiple alternative sources for a problem) as a time-consuming part of the diagnostics work. Two interviewees mentioned the investigation of which series of events lead to the problem occurring (repeating of the problem) in the same context.

Two interviewees noted travelling to vessels as a time-consuming part of the diagnostics work. One interviewee gave an example of a task that would have likely taken 30 minutes to fix remotely, which turned out to take multiple hours to fix because of a travel requirement (the return trip back from the vessel also took multiple hours).

The presence and operation of other workers at vessels was identified by one interviewee as a cause for delays in accessing control systems locally. Operation (sailing) of a vessel was also identified as a hurdle in this regard. Additionally, the investigation of a given problem from the customer's perspective was noted as a time-consuming activity.

Question 5. What is the most diagnosed issue?

Multiple interviewees found it difficult to pinpoint a single issue as the most diagnosed one. The need to change system operating parameters as they are reliant on unique device attributes was one example of a more common issue that was mentioned by one interviewee. Other issues mentioned included functionality that did not behave as expected, miscellaneous electrotechnical issues, software bugs, the need to consider software corner cases, and general PLC software and device firmware issues.

One interviewee described system signal issues as a more common issue category. Wrong signal values, missing signals, and unexpected user action leading to the dispatching of unexpected signals were mentioned as examples. Methods such as the inspection of hardware IO interfaces, repeating the transmission of signals, and checking of signal transmission orders were described to be used in solving these issues. The recording of developer computer screens was mentioned as a method used for detecting quick back and forth changes in signal values.

5.1.2 Remote diagnostics for vessel control systems

The second topic, remote diagnostics for vessel control systems, centered on developer perceptions of remote diagnostics in addition to developer expectations and requirements from a potential remote diagnostics solution. Summarized answers to each of the three questions belonging to this topic are laid out below on per question basis.

Question 1. What does the term *remote diagnostics* mean to you in the vessel control systems context?

Two interviewees described remote diagnostics in the context of vessel control systems as the remote access to vessel software variable values. Use of a VPN connection to PLCs through the development environment was denoted as remote diagnostics by three interviewees. One interviewee described remote diagnostics simply as “detective work”. Usage of screen sharing was mentioned by one interviewee as a real-world example of remote diagnostics. Two interviewees mentioned the ability to talk with vessel personnel remotely as a part of remote diagnostics. One interviewee denoted remote diagnostics as remote access to system logs; the system values (e.g. generator RPM values) within these logs could be viewed as function time on a graph during remote diagnostics work. One interviewee described remote diagnostics as not needing any on-site presence before the problem situation has been evaluated. In this sense, remote diagnostics would allow a developer to have more knowledge of a problem before visiting a vessel as a part of the troubleshooting process.

Question 2. What effect could remote diagnostics have on your work if it would be available?

All six interviewees mentioned the result of having to do less traveling to and from vessels as one effect of remote diagnostics being available. Five interviewees mentioned monetary savings to the case company. One interviewee described the source of these savings as developers having to perform less "guarantee work". Another interviewee mentioned time savings to customers, and one mentioned the possibility that a vessel operator would not need to halt vessel operations for problem troubleshooting to be conducted.

One interviewee gave a real-world example of a developer spending "a few days" to perform a software update that could have taken "a few minutes" if done remotely. Another interviewee described remote diagnostics as having a "quite enormous effect" on the work conducted by the interviewee. This was based on the reduction of time spent trying to solve a given problem as less time would be used up in travelling to the vessel.

One interviewee mentioned the possibility of using remote diagnostics technology during the vessel commissioning process. This notion was based on the apparent similarity of problem diagnostics during commissioning and after a vessel has been commissioned. An example use case for this was given in the form of signal transmission checks where it needs to be ensured that values are written in correct memory addresses on PLCs and other devices.

One interviewee noted that remote diagnostics has potential to make it possible to solve problems noticeably faster and with less effort. Easier access to colleagues as they are sitting close by at the same location was mentioned as one effect of remote diagnostics being available.

According to one interviewee, remote diagnostics (as it has been exercised previously) has been extremely beneficial in limiting excess travel to vessels. An example was provided by another interviewee, according to whom, a VPN connection was used to solve a signal transmission problem. Multiple developers situated in different locations (even at another vessel) were said to have participated in troubleshooting the problem.

Question 3. What would be your requirements from a remote diagnostics solution?

Two interviewees mentioned good usability as a requirement from a remote diagnostics solution. One interview noted the requirement of safety in terms of having confirmations for user actions and ensuring that no harm could be caused to vessels during their operation.

Three interviewees noted the ability to read system variables values as a remote diagnostics requirement. The requirement for users to be able to view logs remotely was presented by three interviewees. One interviewee mentioned the ability to change program parameters remotely. Another interviewee said that source code changes should not be possible remotely as the development environment can be considered sensitive, although the performing of parameter changes was said to be plausible. One interviewee mentioned that the vessel operators could be able to toggle diagnostics support on and off. The ability to view vessel signals (signals lists) with groupings (e.g. generator and propulsion signals in separate groups) together with clear alarm signal indications was also mentioned.

The data refresh rate of a remote diagnostics solution was said to depend on the type of phenomenon or signal that is being monitored: it can be in the milliseconds range but system states may not need to update as often. Few milliseconds, 200 ms, 1 s, 1-3 min and 1-3 h refresh rates were mentioned by the interviewees. One interviewee wished for completely live data so that quick changes in values that switch back forth can be noticed; the missing of these changes could cause a given problem to not be solved. Another interviewee mentioned that the refresh rate could be customizable.

One interviewee mentioned the creation of data snapshots automatically in problem situations (data 1-2 h before and n hours after the problem has been detected). The need to view more accurate data for short time periods and the potential to view data from the past with less granularity (e.g. the change in battery health and motor behaviour on long time periods) was also mentioned by one interviewee. According to the interviewee, historical data could be used to plan maintenance and inform the customers of a maintenance requirement or potential problems beforehand.

According to one interviewee, the more adjustable the solution is to different types of vessels, systems, and devices (supplied by the case company), the better. Two interviewees did not see the need for adjustability as devices and systems do not change that much. One interviewee noted that it would be difficult to adjust diagnostics to the level of system software. Another interviewee mentioned the need for adjustability of which system signals are shown in the diagnostics solution together with signal list viewing support. One interviewee noted wide differences in system deliveries, and the way in which they are worked on; in this context, the ability to configure diagnostics from the device level was mentioned. It should be noted that the prompt question on adjustability of the solution, to which the previously presented answers were given for, proved difficult to understand for multiple interviewees.

At the source code level, the traditional way of diagnosing systems through the development environment was described to be more suitable when compared to a potential remote diagnostics solution. Otherwise the preference of using the traditional way of diagnosis was said to depend on the solution’s capabilities. One interviewee understood the solution to be remote access through the development environment and noted that service personnel would benefit from using a separate diagnostics solution. Two interviewees noted that the usage of one’s senses is also important in the diagnostics work; IO cards lights can be inspected visually, and the sound of the diesel generator is also telling. One interviewee noted that developer on-location presence is important in solving the hardest to solve problems, although the nature of the problem to be solved was mentioned as a factor affecting this.

5.1.3 The role of vessel control system variable values

The third topic, the role of vessel control system variable values, was centered on developer opinions on the most important control system variables and their value refresh rates as the values are viewed during diagnostics activities. Summarized answers to each of the three questions belonging to this topic are laid out below on per question basis.

Question 1. What are the most important variable values that should be accessible remotely during diagnostics activities?

Three interviewees mentioned the importance of having access to system measurements data remotely during diagnostics activities. This data was said to include currents, voltages and torques. One interviewee highlighted the importance of being able to view the vessel DC grid voltage in addition to being able to view temperature measurements from the whole electrical system. DC grid voltage was mentioned as an important variable in indicating vessel overall condition.

According to four interviewees, state machine states should be available for inspection remotely. Three interviewees noted the role of state machine states in indicating vessel overall condition. One interviewee indicated that there is a need to consider hardware device state machines in addition to the PLC software state machines. Two interviewees mentioned the importance of having access to software parameters (that have been set during commissioning). One interviewee made a distinction between PLC and hardware device parameters.

At a more general level, four interviewees noted the importance of having access to the PLC signals interface in the form of viewing inbound and outbound signal values (i.e. variable values). Two interviewees identified this interface as an important indicator for vessel overall condition. One interviewee specified this as the access to variable values based on vessel control system signal list. Two interviewees mentioned the need to be able to view connection statuses and identified them as important indicators of vessel overall condition. The first interviewee of the two explained this as PLC-to-PLC communication checks, while the second expanded the answer with the inclusion of PLC-to-device communication.

Question 2. How often would the variable values need to be refreshed if they were accessible remotely?

The refresh rate specified by the interviewees ranged from a few milliseconds to the level of weeks. Certain variable values, such as state machines states were said to need a seconds-level refresh rate (1 - 2s) whereas measurements data would need to be updated in the milliseconds range (e.g. every 200 ms). An example of week-level refresh rate was provided in the form of battery health monitoring where the *state of health* measurement could change 1 - 2% per week.

One interviewee mentioned the creation of data snapshots every 30 minutes. Another described a more granular approach where data would be gathered at higher resolution for a limited time based on developer needs. One interviewee noted that the need for specific values to be updated is dependent on control system software design.

Question 3. Based on your experience, what variable values could be used to diagnose the most common issues?

According to one interviewee, control system problems are multidimensional and thus there are no singular variable values for diagnosing the most common issues. According to another interviewee, the selection of variable values to focus on differs based on the situation. One interviewee said that the selection depends on the problem to be solved and the nature of hardware devices.

According to one interviewee, state machines states are useful in diagnosing problems in 80% of cases. The inspection of values received by PLCs in one form or another have potential in solving common issues according to three interviewees. One interviewee described the inspection of connection statuses in this context.

One interviewee noted the importance of DC grid measurements in solving most common issues. Multiple voltage measurements from different parts of the grid were mentioned as an example of what to focus on. One interviewee mentioned the checking of values received from hardware devices by the system PLCs.

5.1.4 Possible transformations on vessel control system variable values

The fourth topic, possible transformations on vessel control system variable values, centered on gathering developer opinions on the transformation of system variable values into forms that benefitted the performing of diagnostics activities. Summarized answers to the two questions belonging to this topic are laid out below on per question basis.

Question 1. Can you think of any cases where it would benefit the diagnostics activity to transform system variable values into another form (data and its representations) based on computations?

All interviewees thought that the presentation of variable values (e.g. measurements data) as a function of time on graph would benefit the diagnostics activity. It should be noted that this form of graphical representation was mentioned as an example of data representation when the question was asked. Additionally, similar functionality is housed in the PLC software development environment used by the interviewees in their work duties.

The interviewees found the envisioning of use cases for data transformations (without graphical representations) somewhat difficult. One interviewee noted the possibility for using a low pass filter on data, and this notion was also presented by another interviewee during the answering of the second question of this topic. The usage of electrotechnical equations was deemed potentially useful by two interviewees after they were directly asked about the potential benefit of such equations.

Potential benefit from the usage of data transformations to personnel working directly at the electrotechnical hardware level was mentioned by two interviewees. One interviewee noted (during the answering of the second question of this topic) that certain developers have an electrotechnical background and therefore similar transformations could be useful for them too. One interviewee mentioned the transforming of system signal lists into “plain language” as a possible example of a more complex data transformation.

Question 2. How likely would you be willing to graphically “pipe” the variable value transformations into results?

Two pictures representing FBP process communication were presented to the interviewees as clarifications to the “piping” term mentioned in the question. The pictures showed communication between FBP processes (i.e. nodes) at an abstract and more concrete level (Figures 3 and 4 from chapter 4 respectively).

Two UI mockups were also presented to the interviewees so that the interviewees would be able to gauge their willingness to use FBP concepts in remote diagnostics from a more practical perspective. The first mockup (see Appendix 3) represented the use of a potential FBP toolbox implementation in control system diagnostics. The second mockup (see Appendix 4) represented a bare bones tabular UI alternative where control system variable values were shown as rows in a table. The contrast between the mockups was intended as a tool for helping the interviewees think more critically about the potential of FBP in remote diagnostics. The tabular representation was based on a system signal list -like format that was thought to be familiar to the interviewees whereas the FBP toolbox alternative represented a new approach for diagnostics.

None of the interviewees found the “piping” of values (as its was presented in the supplementary pictures) something that they would not be willing to do. When questioned regarding his willingness to use the FBP toolbox implementation, one interviewee was quoted the following: “I could imagine using this.” (Fin. “Kyllä voisin kuvitella tällaista käyttäväni.”). Another interviewee though the FBP representation seemed useful based on his previous experience in using similar node-linking functionality in another piece of software. One interviewee noted that user does not need to be a developer to use the FBP alternative. Another interviewee was quoted: “Yes I would be, depending on the situation of course.” (Fin. “Kyllä mie olisin, tietysti riippuen tilanteesta.”) regarding his willingness to use the FBP toolbox. The same interviewee noted the need for case company specific FBP processes and expressed willingness to iterate on the graphical solution presented. One interviewee was quoted as saying: “Why not” (Fin. “Miksi ei.”) in the context of using the FBP toolbox. It should be noted that the developer also said that its usage could involve more work for selecting a signal to look at when compared to the tabular alternative.

Four interviewees liked the scope-like FBP-like process that was presented in the first FBP toolbox UI mockup (see Appendix 3). This preference for using the scope seemed to be based on the possibility of comparing values as they are presented as functions of time. One interviewee noted that single values could be transformed based on multiple values and then they could be plotted as a function of time using the FBP process.

One interviewee noted a low pass filter as a possible data transformation process to be provided in the FBP toolbox. This process could be used together with a raw value as they are compared in a graph. The interviewee also noted that the tabular system variable values representation alternative could be joined with the FBP toolbox. As a result of a prompt question, another interviewee noted that base math data transformations and custom functions could be useful but could not think of any examples.

One interviewee noted that hardware device firmware developers could benefit from derivation, integration, sum, and difference transformation FBP processes after he was prompted regarding their applicability. The interviewee also noted that a mathematical sum FBP process could prove useful for system software developers in calculating sums that are not precalculated in developed PLC software. Another interviewee noted that firmware developers have more need for math-themed processes as PLC software developers focus more on individual bits. The interviewee noted the usefulness of transforming single bits to the associated fault descriptions (warnings, trips). The same was noted for received IAS commands that can be represented as words of bits. Unit transformations were noted as potentially useful after a prompt question regarding them was asked. The interviewee also mentioned that the there is a need for the proposed FBP-like solution. Another interviewee noted that there are electrotechnical problems to be solved at the software level too and so similar transformations could have potential use for PLC software developers. One interviewee noted that base math data transformations and custom functions could be useful (after a prompt question regarding them was asked) but could not think of any examples.

Most interviewees found it difficult to express how often (as a percentage of times) they would prefer to directly view raw data values instead of using transformations on the data. One interviewee indicated willingness to use the FBP toolbox alternative 100% of times while another expressed that he would use it 10% of times (and the tabular representation 90% of times). One interviewee wished to combine the tabular representation with graphs that show values as a function time; he would use such implementation 70% of times (and the FBP toolbox alternative 30% of times). One interviewee simply expressed that he would use the tabular alternative more than the FBP toolbox.

The interviewee willingness to use the tabular representation would seem to have multiple reasons. One interviewee noted that the tabular representation is useful for quickly checking multiple values at the same time. The same interviewee also said that the FBP toolbox could be used in more accurate analysis and comparison of values. Another interviewee noted that the validity of data should be checked from the tabular representation where data is in its raw format. The interviewee also said that bits and decimal representations in a table could be enough for quick diagnostics. Another interviewee did not see much need for base data transformations if power ratios are not considered. After a prompt about an FBP-inspired process with graphical representation of bits as “LED lights”, the interviewee described it as useful based on the interviewee’s preference to view data graphically.

5.2 RDS prototype functional and non-functional requirements

The first set of functional and non-functional requirements for the RDS prototype were gathered from the summarized answers to each control system developer interview question for each interview topic. These requirements are shown in Table 6 in a format where the name of a requirement is coupled with a requirement description, source (topics and their questions), and identifier (ID). Requirements that supported the answering the research questions listed in chapter 1.2 through the usability testing of the RDS prototype’s UI were selected to be fulfilled. Certain requirements were considered out-of-scope for this work, and these requirements are listed in Appendix 5. The out-of-scope requirements may be inspected in the future as a basis for a more comprehensive RDS to be developed where fully fledged vessel control systems are connected to the RDS.

Table 6. Functional and non-function requirements for the RDS prototype based on maritime vessel control system software developer semi-structured interviews.

ID	Name	Description	Source(s)
101	Signal and variable values as functions of time graphs	Control system signal and variable values can be viewed as a graph where values are represented as functions of time. Values respective to other values at a point of time can be inspected.	Topic 1, questions 2 and 5; topic 2, questions 1, 2 and 3; topic 3, questions 1 and 3; topic 4, questions 1 and 2
102	Checking of system events	Users can see what has happened and when during troubleshooting activities.	Topic 1, questions 3 and 4
106	“Good usability”	Developer-preferred method of inspecting system values is considered (see requirement 101).	Topic 2, question 3

The questions of the first interview topic, the traditional vessel control systems diagnostics approach, were not directly focused on requirements gathering. Regardless of this, interviewee answers to the topic’s questions gave some indication of implementation requirements through the gauging of developer troubleshooting workflows.

The questions of the second interview topic, remote diagnostics for vessel control systems, focused more on the requirements of the RDS prototype. Certain requirements formed from the answers of the first topic were identified again for this topic.

The questions of the third interview topic, the role of vessel control system variable values, focused on the importance, preferred refresh rate, and relevance of different control system variables and their values if the variables were to be read remotely during diagnostics activities. The answers indicated the importance of system signal values inspection and the requirement for dynamic variable refresh rates (although more precise rates were also provided). Network connection statuses, state machine states, software parameters, and signal values were mentioned as particularly important targets for inspection during diagnostics activities. It should be noted however that the selection of which values to inspect was said to be dependent on the problem to be solved.

The questions of the fourth topic, possible transformations on vessel control system variable values, focused on gauging the potential benefit of control system variable value transformations, and the graphical “piping” of data transformation in the remote diagnostics context. The interviewee answers indicated developer willingness to use FBP-related concepts during diagnostics activities. The listing of useful data transformations proved difficult to the interviewees, but transformations such as low pass filtering, electrotechnical equations, metadata injections, base match operators, and custom transformations were deemed potentially useful.

The presentation of control system values as functions of time on graph was deemed important by all interviewees in the performing of diagnostics activities. The basis for this seemed to be in developer previous experience in using similar functionality within the PLC software development environment.

The requirements in Table 6 were further refined during the user interface design of the RDS prototype. The resulting requirements from this second and final iteration of requirements gathering are shown in Table 7 where the source for each requirement is either a requirement from Table 6 (i.e. the requirement was gathered through developer interviews) or the user interface mockup created during the prototype’s UI design phase.

Table 7. RDS prototype final requirements based on the requirements gathered from maritime vessel control system developer interviews and user interface design of the RDS prototype.

ID	Name	Description	Source(s)
1001	Vessel control system values on a coordinate plane	Control system values are shown on a two-dimensional coordinate plane as functions of time. The x-axis indicates the passing of time and the y-axis indicates system values.	Requirements 101 and 106 from the requirements that were sourced from the PLC software developer interviews.
1002	Vessel control system events on a time axis	Control system events are gathered, and they are shown on the coordinate plane (requirement 1001) x-axis. Events can be selected for focusing them on the x-axis.	Requirement 102 from the requirements that were sourced from the PLC software developer interviews.
1003	User-adjustable y-axis	Users can adjust the position of the coordinate plane (requirement 1001) y-axis relative to the time value shown on the x-axis. Value graphs that intersect the y-axis indicate their values at the point in time indicated by the x-axis. The period of time indicated by the intersection of the two axis is shown.	RDS prototype UI mockup.
1004	Placeholder vessel selection	A placeholder UI element for selecting a specific vessel for diagnostics is provided.	RDS prototype UI mockup.
1005	Placeholder signed in user	A placeholder UI element for the currently signed in user is included.	RDS prototype UI mockup.
1006	Vessel control system events in a tabular format	Events can be inspected in a tabular format through a menu.	RDS prototype UI mockup.

1007	Ability to toggle vessel control system values visibility	Value visibility on the coordinate plane (requirement 1001) can be toggled in a menu.	RDS prototype UI mockup.
1008	Placeholder vessel info	Vessel information can be read through a menu.	RDS prototype UI mockup.
1009	Control system events search	Control system events can be searched for a specific time span and event type.	RDS prototype UI mockup.
1010	Control system values search	Control system values can be searched for a specific time span and value range.	RDS prototype UI mockup.

5.3 RDS prototype design

The RDS prototype's design work consisted of its user interface and architecture design phases. The user interface design phase was directly based on the fulfilling of requirements listed in Table 6. The architecture design of the prototype had the following two goals: a diagnostics UI implementation was to be technically supported for the answering of research question 2, and the communication of diagnostics data from PLCs to an RDS web server was to be testable for potential future benefit to the case company.

In the beginning, the user interface design step was based on the fulfilling of requirements listed in Table 6. As these requirements were being fulfilled in the UI design, more concrete functionality-based requirements became apparent. These requirements are shown in Table 7.

5.3.1 RDS prototype user interface design

Functional requirement 101 in Table 6 describes the need for users of the RDS prototype to be able to view system signal values as functions of time. The preference for viewing data in this manner became increasingly apparent during the developer interviews. Based on this data representation's applicability in remote diagnostics, it was decided that the RDS prototype's main UI should consist of a so called vessel timeline presentation of control system variable values.

It was identified during the design of the timeline presentation of control system values that a search functionality for control system values and events could be beneficial. This was based on the aim of easing user navigation to points of interest in the UI (e.g. specific values at a point in time). The combination of the timeline representation and more traditional tabular views and search functionality is shown in Figure 5. Here vessel control system events are shown on the horizontal axis, and values at a specific point in time may be inspected from a movable vertical axis. A collection of menus is provided for viewing system date in a tabular format in addition to other potentially beneficial functionality such as events and values search.

Figure 5. RDS prototype main UI mockup. Control system events are visible on the horizontal axis, control system values are visible at a point in time at the movable vertical axis, and menu content for the Events menu is visible.

The design of the RDS prototype main UI, and thus the inclusion of the new requirements was validated through the informal asking of maritime vessel control system developer feedback. Three developers were asked if the presented UI would help them in indicating system status and each developer indicated this to be true. The applicability of the *values on a graph as a function of time* functionality was a focal point during this validation of design decisions.

5.3.2 RDS prototype architecture design

The architecture of the RDS prototype was designed using the C4 model for architecture developed by Simon Brown. The C4 model is intended to provide a set of common abstractions (Brown 2019, 20) and vocabulary that can be used to visualize the structure of software in a static manner (Brown 2019, 32). It provides the following four abstractions where each abstraction can be perceived as zooming into the preceding abstraction: system context, containers, components, and code (Brown 2019, 32).

The C4 model system context diagram for the RDS prototype is presented in Figure 6. This diagram type is intended to be used in showing the software system as a part of an existing environment coupled with the users of the system (Brown 2019, 36).

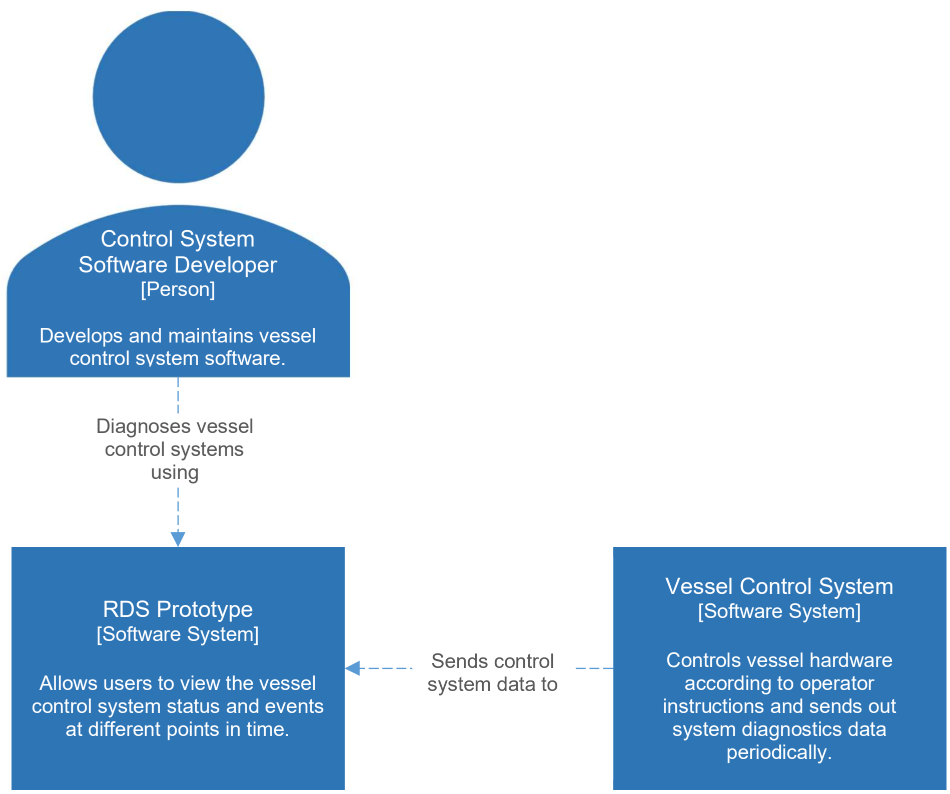


Figure 6. C4 Model system context diagram for the RDS prototype.

The C4 model container diagram for the RDS prototype software system is presented in Figure 7. This diagram type is intended to be used in showing the high-level architectural structure of the software together with implementation technology choices, division of responsibilities, and communication between containers (Brown 2019, 41).

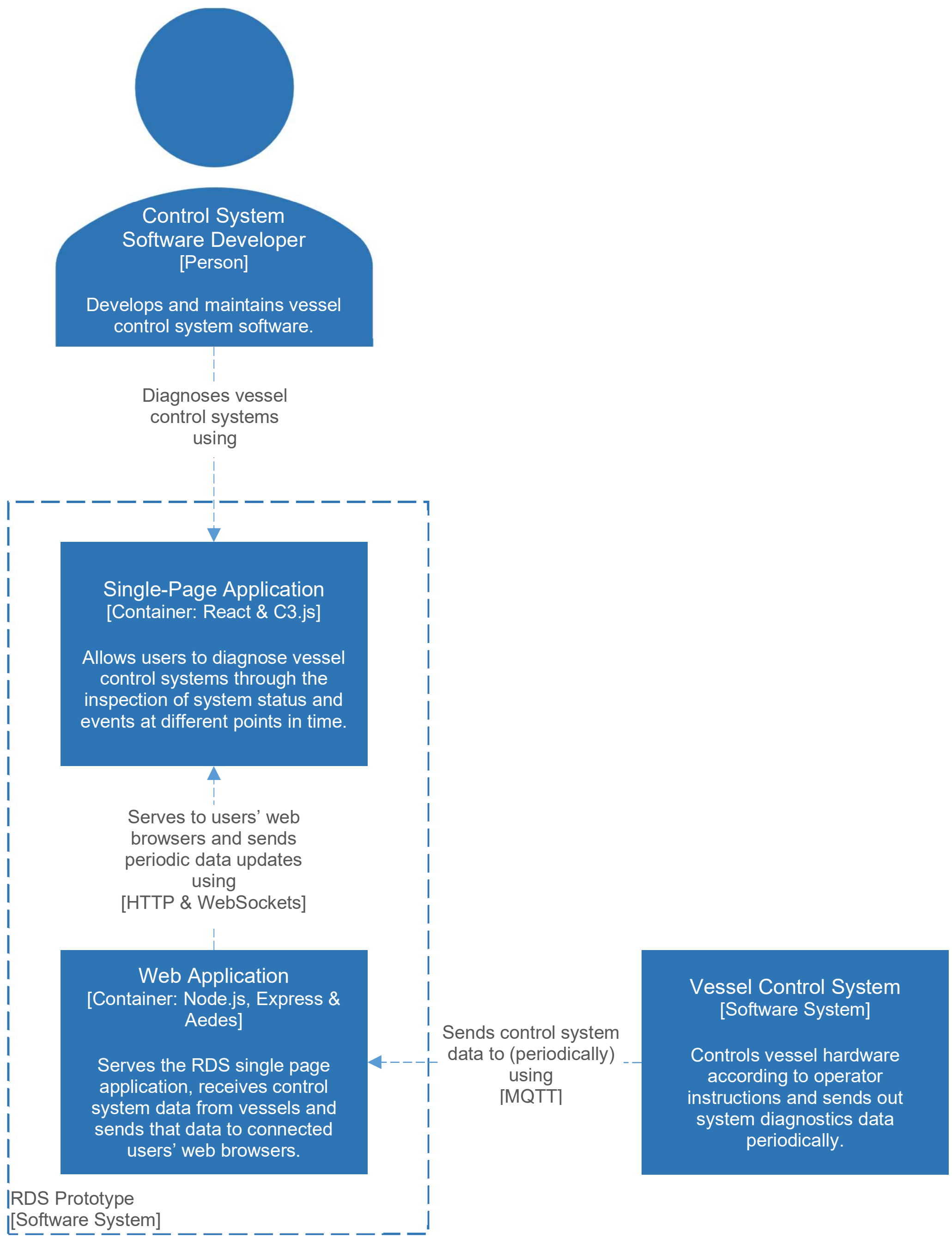


Figure 7. C4 Model container diagram for the RDS prototype software system.

5.4 RDS prototype implementation

The implementation of the RDS prototype began with the establishment of data transfer between a PLC, RDS web server, and a web browser. A simplified program was developed on the PLC side that communicated JavaScript object notation (JSON) format data from the PLC. This was done using software libraries for JSON document creation and message queuing telemetry transport (MQTT) protocol communication supplied by the PLC manufacturer. Data from the PLC was received by a Node.js web server that ran an MQTT server using the Aedes package. The MQTT messages were parsed on the web server after which they were transferred to a client web browser using the WebSocket protocol. Before the messages could be received by the client, it had to make an initial hypertext transfer protocol (HTTP) request for the RDS prototype single page application to be transferred. This application would then attempt to create a WebSocket connection to the web server.

The prototype's user interface was constructed as a single page application using React and C3.js as its main technology components. It should be noted that the vessel timeline representation (collection of vessel values on an XY plane) was based on the C3.js library and only limited modifications were made to the default outlook of its elements. W3.CSS framework was used to style the UI's HTML elements with some modifications.

A screen shot of the finished RDS prototype is show in Figure 8. In this screenshot, events have been searched for within a specific time span, and a single event has been focused on the vessel timeline. The event is indicated by a red vertical line perpendicular to the horizontal x-axis. The selected event is indicated with a grey background in the righthand event search results table. Left mouse click of an another event row in the table would focus the vessel timeline starting point at that event's timestamp.

Figure 8. RDS prototype main UI. Control system events have been searched for a specified time span and one event has been focused (righthand menu content search result row is darkened, and a vertical event line is highlighted on the left-hand side).

The result of focusing a single control system value based on a timestamp is shown in Figure 9. The focused value is indicated by an orange circle on the vessel timeline (x-axis). The selected timestamp for this value is shown with a grey background in the righthand value search results list.

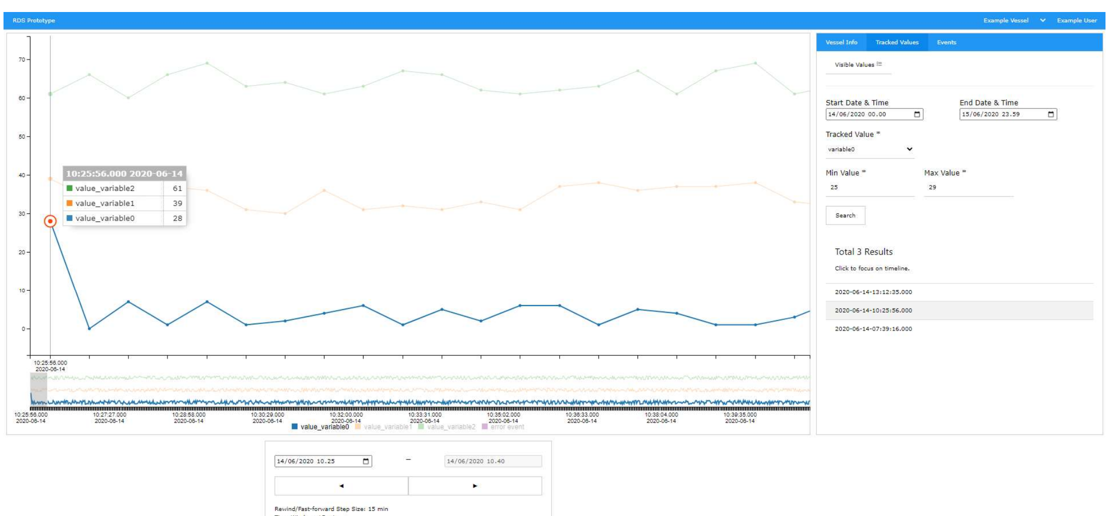


Figure 9. RDS prototype main UI. A value search result has been selected and focused on the vessel timeline and in the righthand menu content.

5.5 RDS prototype usability testing results

The RDS prototype usability testing sessions were conducted according to the usability testing plan presented in chapter 4.4.1. This chapter begins with a sub chapter on the general characteristics of the recruited usability testing participants. Results of the first and second usability testing scenarios are presented next, after which the meeting of set usability testing objectives is described. This subchapter ends with the presentation of the participant feedback regarding the need for changes or new functionality in the RDS prototype.

5.5.1 Usability testing participant characteristics

Three PLC software developers with maritime vessel control system development experience participated in the usability testing sessions. Each participant had taken part in the semi-structured interviews that functioned as a basis for the requirements gathering and design of the prototype RDS implementation. The recruitment of only three participants was based on scheduling challenges related to personnel vacations in addition to the limiting of time resources spending. All participants were Finnish speakers and thus the testing sessions were conducted in Finnish. The participants were provided both written and verbal consent alternatives and each participant chose to give verbal consent for taking part in the testing. The Finnish format consent form that was used as a basis for verbal consent is presented in Appendix 6.

5.5.2 Results of the first usability testing scenario

The first usability testing scenario presented in chapter 4.4.1.5 had each test participant attempt to find a control system error event during a specific date and time span. If an event was found, the participant was instructed to find out the value of a specific variable at the time when the event occurred.

All test participants were able to find the error event and the value of the specified variable, although one participant had to be guided in the resetting of a graph's visibility (this graph indicated the value of the searched for variable). Each participant attempted to use the *Tracked Values* search functionality of the prototype to search for the event in question. When this failed, the participants move onto manual navigation on the vessel timeline, which in turn lead to them being able to find the looked-for event. After the event was found, each participant was able to read out the value of the variable mentioned in the scenario description, which resulted in the completion of the scenario.

The usage of the time inputs for values search and manual date and time-based navigation on the vessel timeline seemed to be mostly straightforward to the participants. It should however be noted that none of the participants clicked the calendar icon on the inputs to select a date and time from the browser-provided UI element; all participants filled in the dates and times manually.

5.5.3 Results of the second usability testing scenario

The second usability testing scenario presented in chapter 4.4.1.5 had each test participant attempt to search for timestamps that indicated when a specific control system value was within a specified range. The reading out of the total count of these timestamps enabled the participants to move onto the next stage of this scenario; participants were instructed to check if an error event was to be found within a 15-minute window of each timestamp.

All test participants were able to search for the timestamps (that indicated when the specified variable was within the instructed range). Each participant was able to note the total count of the timestamps as the count of times that the specified variable was within the range mentioned in the scenario description. Two participants were able to immediately check for the occurrence of error events within a 15-minute time window of each timestamp. One participant had to be guided to click a timestamp search result for the result to be focused on the vessel timeline. It should be noted that a delay in the timestamp focusing functionality was noticeable during the testing session, and this may have affected the participant's ability to acknowledge the provided functionality.

5.5.4 Meeting of the usability testing objectives

The listing of usability testing objectives presented in chapter 4.4.1.2 contains a total of five objectives that are centered on finding out test participant preferences and areas of difficulty regarding the use of the RDS prototype's UI. These objectives and the results for meeting each objective are presented in Table 8. One noteworthy point that was identified during the meeting of these goals was that test participants quickly moved to the use of manual vessel timeline navigation controls if the provided search functionality did not meet their needs; the participants were able to circumvent limited difficulties in using the prototype.

Table 8. Results from meeting the usability testing objectives.

Objective	**Result**
1. Find out if the users naturally use the values and events search functionality when they look for values and events on specific dates and ranges, or if they prefer to manually navigate on the vessel timeline using the playback rewind and fast-forward controls in addition to the date picker control.	Test participants first leaned toward the use of the *Tracked Values* search functionality, but then switched to manual timeline navigation if the search did not function as expected. Interestingly, the events search functionality saw no use during the testing.
2. Find out if the search and focusing of a specific control system event on the vessel timeline using the events search functionality is apparent to the users**.**	The search and focusing of events on the vessel timeline were not apparent to the test participants as not a single tester used the events search functionality (manual navigation on the vessel timeline was preferred).
3. Find out if the search and focusing of control system value range timestamps on the vessel timeline using the values search functionality is apparent to the users.	The search of value range timestamps was clear to all test participants. The focusing of specific value timestamps was immediately clear to two participants.
4. Find out if users able to tell what value/event is indicated by control system value/event timestamp when search results are being focused upon on the vessel timeline.	During the testing, it seemed that the test participants did not pay much attention to the value and event focus indicators. Manual checking of values on individual graphs seemed more natural to the participants. The vertical line event indicator was not immediately apparent to at least one participant who relied more on

	graph tooltips for event occurrence inspection.
5. Find out if users face difficulties in differentiating control system event indicators from value graphs.	One participant was directly able to understand the role of the vertical lines as event indicators. Another participant was not able to see a vertical event indicator in the mix of value graphs at one point during testing.

5.5.5 Test participants requests for more functionality or changes in functionality

Each participant was prompted with a question regarding the need for more functionality in the prototype. Two of the three participants found it difficult to envision new functionality, although possibilities for changes in the existing functionality were mentioned during the asking of this prompt. These included the possibility for a separate vessel events timeline, easier to use time inputs, and the ability to change the time window and time step sizes for the vessel timeline. One participant noted the possibility for diagnostics data export and scaling of individual UI elements as potential examples of new functionality.

The prompt regarding changes for functionality did not receive as many answers as the one regarding the possibility for adding new functionality into the prototype. A likely cause for this was that changes to functionality were already mentioned during the asking of new functionality requirements. One participant noted that the time inputs could be easier to use; the appearance of AM and PM time labels in the time inputs was not preferred by this participant (a 24-hour representation was wished for).

At the end of each testing session, the participants were prompted regarding the ability of the RDS prototype to help the participants determine vessel control systems status at different points in time. Each participant though that this would be possible using the base functionality provided in the prototype. One participant was more concerned with access to relevant data as an enabler of the prototype’s usage in this manner. Another expressed willingness to use the prototype in inspecting live control system signals.

5.6 Results summary

The two main phases of research result generation were the maritime vessel control system developer interviews and usability testing of the RDS prototype. The design and implementation work of the prototype may be considered sub phases that enabled the usability testing of the finished prototype.

The control system developer semi-structured interview answers indicated that there exists a multitude of different values that may or may not be useful in the conducting of remote diagnostics activities depending on the nature of the problem to be solved. Certain commonalities in the answers could however be found; state machines states, signal interfaces and general measurements data would seem important to the interviewees in the general diagnostics circumstances in addition to other data. The ability of different values to communicate vessel overall condition better than others was difficult to gauge for the interviewees but state machine states seemed a be a potential candidate for this.

A point of interest indicated by the developer interviews was that it would likely benefit the diagnostics activities if control system values were indicated as graphs where individual data points would be plotted on a time axis. This concept was further developed in the RDS prototype design phase to include more traditional search and lookup of mock (i.e. imaginary) control system values and events.

During the usability testing of the RDS prototype, it became apparent that the provided UI is likely to be suited to the task of finding out the status of a control system at a specific point in time. This result has a basis on the opinions provided by the usability testing participants and their ability to complete the testing scenarios with limited or no guidance.

6 DISCUSSION AND CONCLUSIONS

This chapter contains the discussion of the research results and description of the conclusions that were drawn from the results. Answers to research question 1 and its associated research objective are provided in subchapter 6.1. Subchapter 6.2 presents the answer to research question 2 and its associated research objective. Subchapter 6.3 describes the implications of the research. Limitations of the research are presented in subchapter 6.4. The recommendations for future research activities are presented in subchapter 6.5. The presentation of research conclusions relating to the understanding of user workflows and development tools usage in chapter 6.6 concludes this chapter.

6.1 The importance of different control system values in system diagnostics

Research question 1 presented in chapter 1.2 was aimed at finding out what maritime vessel control system values indicate vessel status or are otherwise important to control system developers during control system diagnostics activities. This question was answered through the planning and conducting of semi-structured control system developer interviews presented in chapters 4 and 5, respectively.

The completion of the semi-structured interviews resulted in the determination that there exists a multitude of control system values that are important to control system developers during the performing of system diagnostics activities. These include, but are not limited to state machine states, signal interface values, and hardware measurements data. The interview results indicate that a range of control system values must usually be considered for the vessel control system status to be clear to system developers. Linked to this is the notion that the inclusion of different values for consideration is dependent on the control system issue to be solved.

As a special consideration, system state machine states may have potential in indicating control system status in overall higher number of cases when compared to other types of system values. This notion is based on the answers of three control system developers to the first question of the third interview topic presented in chapter 5.1.3; each of these developers indicated that state machines states have a role to play in indicating vessel overall condition.

Based on the interview results, the inclusion of multiple control system values for the indication of control system status seemed to have a basis on the potential complexity of control system and the wide range of possible issues to be solved in the operation of these systems. It was envisioned during the design of the RDS prototype's UI that multiple values could be inspected at the same time using the UI, but a possible complementing approach could be to aggregate multiple values into easy to understand visual elements. These elements may in turn be able to provide a higher-level view into the status of a given control system.

The research objective associated with the answering of research question 1 was presented in chapter 1.2 as research objective 1. It was aimed at finding out what control system values indicate vessel status or are otherwise important to maritime vessel control system developers during the conducting of control system diagnostics activities. As can be seen based on the previous discussion, this objective was met through the answering of research question 1.

6.2 The representation of control system status in an RDS UI

Research question 2 presented in chapter 1.2 was aimed at finding out what form of an RDS UI is preferred by maritime control system developers for indicating vessel control system status (through the presentation of control system values) during system diagnostics activities. As a result of the system developer interviews and developer participation in the usability testing of the RDS prototype, the form of remote diagnostics UI that would seem to be preferred by control system developers became evident. In this UI, the indication of system status was designed to be a representation of a so called vessel timeline together with system values search functionality.

The source of the timeline-based representation of vessel diagnostics data was on control system developer answers to the semi-structured interview topic 4, questions 1 and 2 that are presented in chapter 5.1.4. The questions in this topic were able to generate answers regarding the potential of system variable values transformations, and the possibility for representing system values as function of time graphs became relevant through developer answers. This temporal representation of values was indicated as potentially beneficial in the conducting of diagnostics activities by all control system developers that took part in the interviews.

The visual function of time graph approach to diagnostics would seem to have its basis on developer previous experience in using the PLC software development environment. A so-called scope functionality for inspecting software variable values as graphs on a time axis is provided in the environment. Developer willingness to use the development environment in this manner for remote diagnostics did seem particularly high; the taxing nature of the development environment usage (e.g. because of network latency) in the diagnostics context seemed to be the cause for this based on the results of the system developer interviews.

The research objective associated with the answering of research question 2 was presented in chapter 1.2 as research objective 2. It was aimed at finding out how maritime vessel control system values should be indicated in a remote diagnostics service user interface so that control system developers are able to view the system status during the performing of system diagnostics activities. As can be seen based on the previous discussion, this objective was met through the answering of research question 2.

6.3 Implications of the research

The main implication of this research is that a potential basis for the envisioning of a more complete RDS UI to be used by control system developers in remote diagnostics activities has been established. The gauging of system developer needs from an RDS implementation as these needs relate to specific control system values inclusion and general remote diagnostics requirements is provided as a secondary implication.

6.4 Limitations of the research

One limitation of the research is that the UI of the RDS prototype was tested by a limited number of participants; a total of three control system developers took part in the testing. Inclusion of more participants could have provided more insight regarding the ability of system developers to use the UI in finding out the status of a given control system at a specific point in time. Another limitation associated with the testing is that mock control system data with no live data monitoring functionality was used during the testing sessions.

It should be noted that the number of control system developer semi-structured interview participants was limited to six developers. I it is possible that further insight regarding developer requirements from an RDS implementation could have been gained through the recruitment of additional interviewees.

The communication of data from a PLC to the developed RDS web server, and then finally to a web browser client UI was implemented locally for an early version of the RDS prototype. The data that was sent from the PLC during internal testing consisted of randomly generated floating-point values and hard-coded event strings as JSON documents. The locally run RDS web server was fitted with support for multiple simultaneous browser connections, but this was not tested in action. The communication of live diagnostics data to the web browser-based UI was not tested by control system developers.

6.5 Recommendations for future research

A potential recommendation for future research is the use of the gathered control system developer preferences as a basis for researching the real-world implementation requirements for vessel control system diagnostics data storage and transfer. This research could attempt to envision a method for collecting the required data in a manner that satisfies both the vessel operators and the providers of vessel control systems. A potential issue to be solved in this regard is the plausible vessel operator unwillingness in establishing outside network connections to the vessel control system.

6.6 Concluding remarks

As a conclusion of the research efforts, it can be said that future remote diagnostics service implementation efforts could benefit from the understanding of their target audiences' methods of working and how different tools are used to accomplish task. In other words, the implementation of the vessel timeline in the designed and developer RDS prototype had a basis on developer opinions which were likely affected by their use of the graphical scope functionality found in the PLC software development environment used by the developers.

7 SUMMARY

The results of the semi-structured maritime vessel control system developer interviews indicated that multiple different control system values interact in the indication of system status, and that the nature of system issues to be solved affect the selection of specific values for monitoring. Control system states were provided as a potential candidate as a type of values that may be able to indicate system status in a less limited number of cases when compared to other types of values.

The semi-structured interview functioned in the forming of a basis for the representation of system status in the designed and developed RDS prototype implementation. The presentation of control system values and events on an XY plane where the x-axis is used to indicate a so called vessel timeline, and the y-axis denoted variable values, can be used to indicate control system status at a specific point in time. Usage of this timeline representation was tested by control system developer though usability testing of the RDS prototype's UI., This UI implementation was deemed useful regarding its ability to indicate control system status at specific points in time based on the testing results and prompted test participant feedback.

REFERENCES

ABB, 2013, *Remote Diagnostic Services (RDS)* [Brochure]. Available: https://new.abb.com/marine/systems-and-solutions/service/troubleshooting-and-repair [Accessed 27 January, 2020].

BERGIUS, H., n.d., *Flow-Based Programming for JavaScript*. Available: https://noflojs.org/ [Accessed 29 January, 2020].

BROWN, S., 2019. *Software Architecture for Developers - Volume 2.* n.p.: Leanpub. Available: https://leanpub.com/b/software-architecture [Accessed 27 January, 2020].

ERTOGAN, M., TAYYAR, G.T., WILSON, P.A. and ERTUGRUL, S., 2018. Marine measurement and real-time control systems with case studies. *Ocean Engineering,* **159**, pp. 457-469.

FLOWHUB, n.d., *Flow-based programming for the full stack*. Available: https://flowhub.io/ide/ [Accessed 29 January, 2020].

GROOVER, M.P., 2015. *Automation, Production Systems, and Computer-Integrated Manufacturing.* 4rd edn. USA: Pearson Higher Education, Inc.

HOVE, S.E. and ANDA, B., 2005. Experiences from Conducting Semi-structured Interviews in Empirical Software Engineering Research, *IEEE International Software Metrics Symposium* 2005, IEEE Computer Society, pp. 10-23.

JONSSON, K. and HOLMSTRÖM, J., 2005. Ubiquitous Computing and the Double Immutability of Remote Diagnostics Technology: An Exploration into Six Cases of Remote Diagnostics Technology Use, C. SØRENSEN, Y. YOO, K. LYYTINEN and J.I. DEGROSS, eds. In: *Designing Ubiquitous Information Environments: Socio-Technical Issues and Challenges* 2005, Springer US, pp. 153-167.

JS FOUNDATION, n.d., *Node-RED*. Available: https://nodered.org/ [Accessed 29 January, 2020].

KHORRAMI, F., KRISHNAMURTHY, P. and KARRI, R., 2016. Cybersecurity for Control Systems: A Process-Aware Perspective. *IEEE Design Test,* **33**(5), pp. 75-83.

KONGSBERG, n.d., *Kongsberg Remote Services*. Available: https://www.kongsberg.com/maritime/products/information-management-system/Infrastructure-and-Security/kongsberg-remote-services/ [Accessed 27 January, 2020].

KUSCHEL, J. and LJUNGBERG, F., 2005. Decentralized Remote Diagnostics: A Study of Diagnostics in the Marine Industry, S. FINCHER, P. MARKOPOULOS, D. MOORE and R. RUDDLE, eds. In: *People and Computers XVIII --- Design for Life* 2005, Springer London, pp. 211-226.

MORRISON, J.P., n.d. *Flow-based Programming*. Available: https://jpaulm.github.io/fbp/ [Accessed 29 January, 2020].

PEPLIŃSKI, H., 2019. Chapter 14 - Remote diagnostic systems. In: HENRYK PEPLIŃSKI, ed, *Ship and Mobile Offshore Unit Automation.* n.p.: Gulf Professional Publishing, pp. 141.

RENSVIK, E., SØRENSEN, A.J. and RASMUSSEN, M., 2003. Maritime Industrial IT B . 5, paper presented to *the 9th International Conference on Marine Engineering Systems*, Helsinki University of Technology (HUT) Ship Laboratory and on board MS SILJA SERENADE, 19-21 May 2003.

RUBIN, J. and CHISNELL, D., 2008. *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests.* 2 edn. Indianapolis, IN: Wiley Publishing.

SCHATZ, D., BASHROUSH, R. and WALL, J., 2017. Towards a More Representative Definition of Cyber Security. *Journal of Digital Forensics, Security and Law,* **12**(2), pp. 53-74.

SHI, Y., SHEN, C., FANG, H. and LI, H., 2017. Advanced Control in Marine Mechatronic Systems: A Survey. *IEEE/ASME Transactions on Mechatronics,* **22**(3), pp. 1121-1131.

SKJETNE, R. and EGELAND, O., 2006. Hardware-in-the-loop testing of marine control system. *Modeling, Identification and Control,* **27**(4), pp. 239-258.

SKJONG, E., VOLDEN, R., RØDSKAR, E., MOLINAS, M., JOHANSEN, T.A. and CUNNINGHAM, J., 2016. Past, Present, and Future Challenges of the Marine Vessel's Electrical Power System. *IEEE Transactions on Transportation Electrification,* **2**(4), pp. 522-537.

SØRENSEN, A.J., 2013. *Marine control systems: Propulsion and Motion Control of Ships and Ocean Structures,* lecture notes*,* Norwegian University of Science and Technology. Available: http://folk.ntnu.no/assor/publications/marcyb.pdf [Accessed 22 June, 2020]

SØRENSEN, A.J., 2005. Structural issues in the design and operation of marine control systems. *Annual Reviews in Control,* **29**(1), pp. 125.

VALMET, 2015. *Valmet DNA Machine Monitoring* [Brochure]. Available: https://www.valmet.com/automation/condition-monitoring/ [Accessed 27 January, 2020].

WILSON, C., 2014. Chapter 2 - Semi-Structured Interviews. In: CHAUNCEY WILSON, ed, *Interview Techniques for UX Practitioners.* Boston: Morgan Kaufmann, pp. 23.

ZORIC, G., DOMOVA, V., RALPH, M., VARTIAINEN, E., BJ ORNDAL, P. and MU NOZ, A.A., 2016. Supporting Maritime Remote Experts Working Over Distance.

Proceedings of the 9th Nordic Conference on Human-Computer Interaction (NordiCHI '16) 124, pp. 1–6. DOI:https://doi.org/10.1145/2971485.2995347

APPENDIX 1. Maritime vessel control system software developer interview topics and questions in Finnish

Alusten ohjausjärjestelmien diagnosoinnin perinteinen lähestymistapa

1. Miten alusten ohjausjärjestelmien ohjelmisto-ongelmia diagnosoidaan tavallisesti?
2. Mitä työkaluja käytetään diagnostiikkatyössä?
    a. Voisitko kuvailla, kuinka kyseistä työkalua käytetään?
    b. Mitä dataa työkalu tarjoaa?
3. Mikä on vaikein osa diagnostiikkatyötä?
    a. Mistä tämä johtuu?
4. Mikä on kaikista aikaa vievin osuus diagnostiikkatyössä?
    a. Mistä tämä johtuu?
5. Mikä on kaikista eniten diagnosoitu ongelma?
    a. Kuinka tämä ongelma diagnosoidaan?

Etädiagnostiikka alusten ohjausjärjestelmille

1. Mitä termi *etädiagnostiikka* tarkoittaa sinulle alusten ohjausjärjestelmien kontekstissa?
2. Mikä vaikutus etädiagnostiikalla voisi olla työssäsi, jos se olisi saatavilla?
    a. Mitä ongelmia voitaisiin ratkaista sen avulla?
    b. Tuleeko mieleesi yhtään aikaisempaa tilannetta, jossa etädiagnostiikan käyttämisestä olisi voinut olla hyötyä?
3. Mitkä olisivat vaatimuksesi etädiagnostiikkaratkaisulta?
    a. Kuinka usein diagnostiikkadatan tulisi päivittyä?
    b. Käyttäisitkö diagnostiikkaratkaisua reaaliaikaisen datan tarkastelemiseen, vai tulisiko aikaisemmin kerätyn datan olla saatavilla?
    c. Miten mukautettavissa diagnostiikkaratkaisun tulisi olla erilaisiin ohjausjärjestelmiin tai ongelmatilanteisiin?
    d. Minkä johdosta käyttäisit perinteistä diagnosoinnin lähestymistapaa etädiagnostiikkaratkaisun sijasta?

(continues)

APPENDIX 1. (continues)

Alusten ohjausjärjestelmien muuttujien arvojen rooli

1. Mitkä ovat kaikista tärkeimpiä muuttujien arvoja, joiden tulisi olla saatavilla etänä diagnostiikkatoiminnan aikana?
    a. Mitkä muuttujat antavat tietoa aluksen yleisestä tilasta?
2. Kuinka usein muuttujien arvoja tulisi päivittää, jos ne olisivat saatavilla etänä?
    a. Onko olemassa joitakin muuttujien arvoja, joita tulee päivittää harvemmin tai useammin?
3. Mitä muuttujien arvoja voitaisiin käyttää yleisimpien ongelmien diagnosointiin kokemuksiesi perusteella?

Alusten ohjausjärjestelmien muuttujien arvojen mahdolliset muunnokset

1. Tuleeko mieleesi yhtään tilannetta, jossa diagnostiikkatoimintaa voisi hyödyttää muuttujien arvojen muuntaminen toiseen muotoon (data ja sen esitystavat) laskennan perusteella?
    a. Tuleeko mieleesi yhtään olemassa olevia muunnoksia, joita käytät jo nyt?
2. Miten todennäköisesti olisit halukas graafisesti ”piiputtamaan” muuttujien arvojen muunnoksia tuloksiin?
    a. Jos ”piiputettavia” muunnoksia tarjottaisiin, niin mitkä muunnokset olisivat todennäköisesti eniten käytössä?
    b. Jos olisit vapaa valitsemaan muunnosten käytön ja raakojen muuttujien arvojen tarkastelemisen välillä, niin kumpaa suosisit ja kuinka usein (prosenttiosuus kokonaiskerroista)?

APPENDIX 2. Finnish format maritime vessel control system software developer semi-structured interview consent form

Haastattelusuostumuslomake

Annan suostumukseni siihen, että Samuli Siitonen haastattelee minua osana hänen LUT-yliopiston tietotekniikan koulutusohjelman ohjelmistotuotannon pääaineen diplomityönsä (2019–2020) teemahaastatteluja. Suostun vastauksieni hyödyntämiseen edellä mainitussa diplomityössä ja siihen liittyvässä ohjelmallisessa toteutustyössä. Vastauksiani käsitellään anonyymisti ja ne eivät ole yksilöitävissä minuun. Haastattelusta tehdään muistiinpanot, jotka Samuli Siitonen tuhoaa diplomityön valmistuttua.

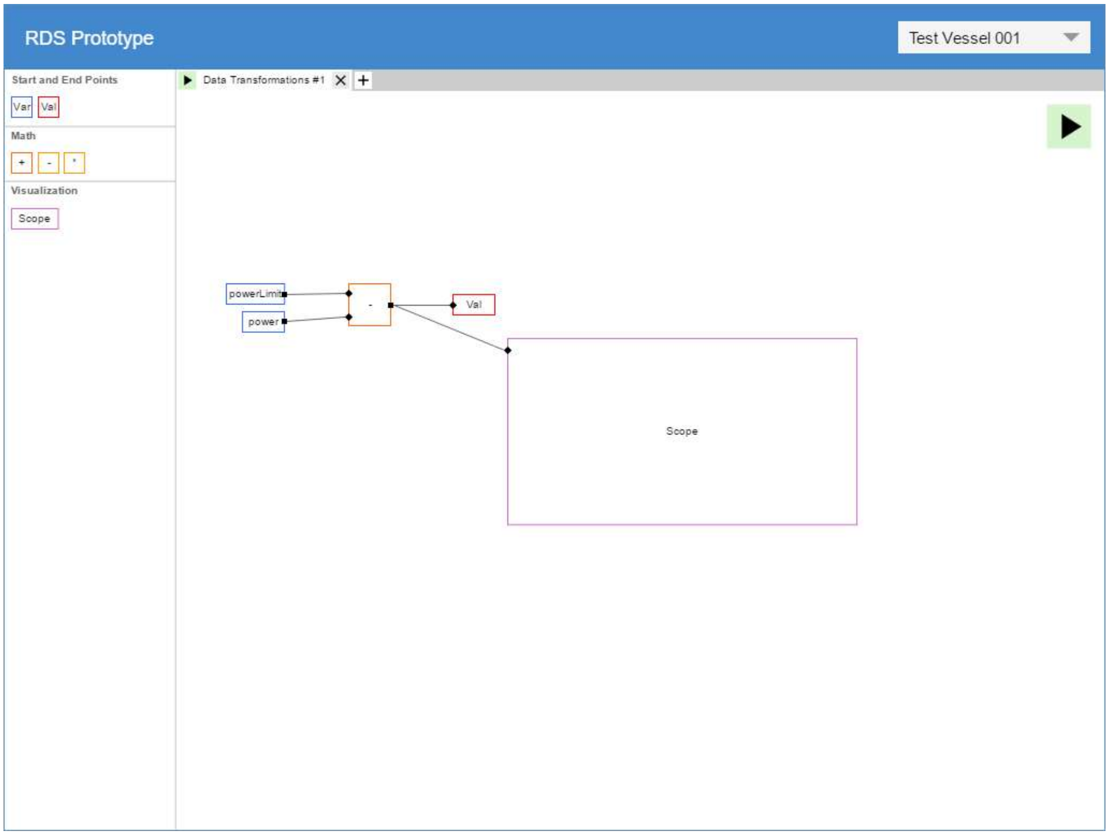
Minulla on oikeus perua osallistumiseni haastatteluihin ja vastauksieni hyödyntämiseen milloin tahansa ennen diplomityön valmistumista.

Paikka ja päivämäärä ___________________________

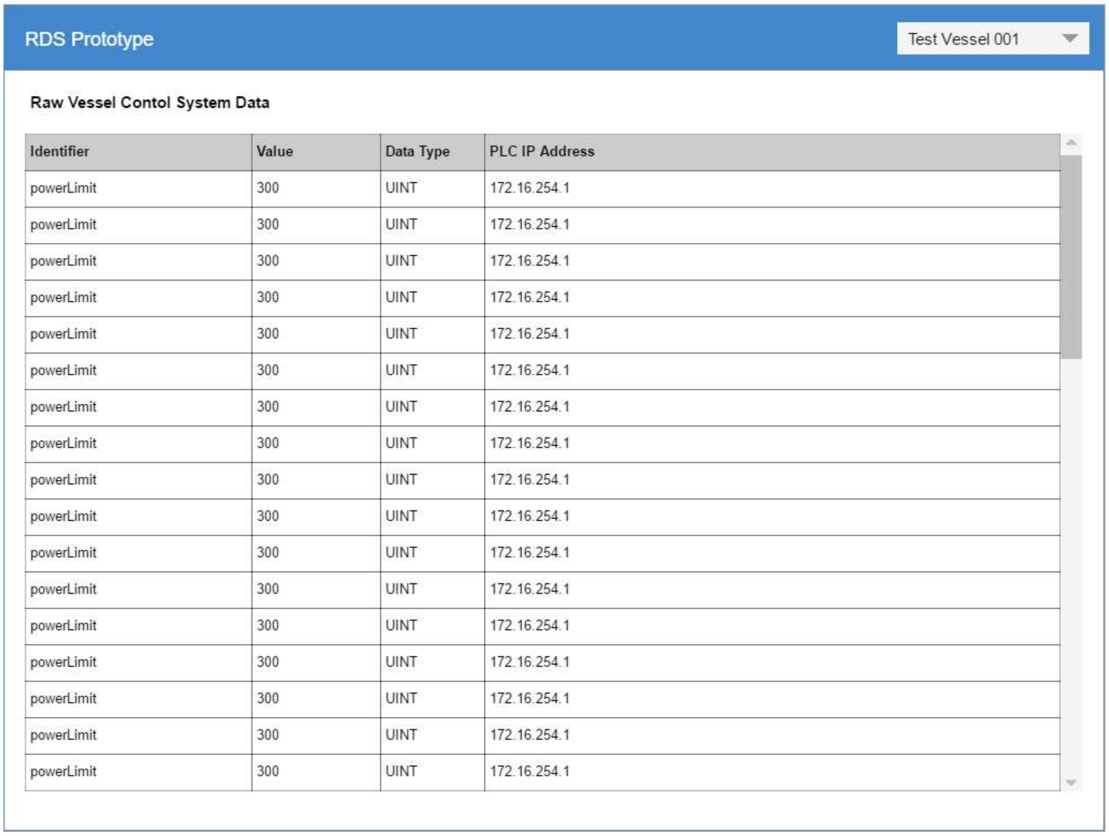
Haastateltavan allekirjoitus ___________________________

Haastateltavan nimenselvennys ____________________________

APPENDIX 3. Simplistic UI mockup of a potential FBP toolbox implementation and its usage in an RDS prototype



APPENDIX 4. Simplistic UI mockup of tabular control system variable value representation in an RDS prototype



Identifier	Value	Data Type	PLC IP Address
powerLimit	300	UINT	172.16.254.1
powerLimit	300	UINT	172.16.254.1
powerLimit	300	UINT	172.16.254.1
powerLimit	300	UINT	172.16.254.1
powerLimit	300	UINT	172.16.254.1
powerLimit	300	UINT	172.16.254.1
powerLimit	300	UINT	172.16.254.1
powerLimit	300	UINT	172.16.254.1
powerLimit	300	UINT	172.16.254.1
powerLimit	300	UINT	172.16.254.1
powerLimit	300	UINT	172.16.254.1
powerLimit	300	UINT	172.16.254.1
powerLimit	300	UINT	172.16.254.1
powerLimit	300	UINT	172.16.254.1
powerLimit	300	UINT	172.16.254.1
powerLimit	300	UINT	172.16.254.1
powerLimit	300	UINT	172.16.254.1

APPENDIX 5. Out-of-scope functional and non-functional software requirements for the RDS prototype

ID	Name	Description	Source(s)
103	Communication of vessel issues	Standardized communication of vessel issues (together with relevant event data).	Topic 1, questions 1,3 and 4; topic 2, question 1
104	Collaboration between users	Users can collaborate during the troubleshooting process.	Topic 1, question 2; topic 2, questions 1 and 2
105	Ability to view system logs	System logs can be accessed remotely, and they may be inspected in a similar manner as values based on requirement 101.	Topic 1, question 1; Topic 2, questions 1 and 3
107	"Good safety"	Confirmations for user actions and limited ability to affect live vessel operations.	Topic 2, question 3
108	PLC software parameter changes	Ability to change PLC software parameters remotely.	Topic 2, question 3
109	Remote diagnostics on/off toggle	Ability for vessel operators to toggle the remote diagnostics support on/off.	Topic 2, question 3
110	Data refresh rate	Data refresh rate is based on variable metadata (configured at the PLC software level).	Topic 2, question 3; topic 3, question 2
111	Data snapshots	System data snapshots are created at increased granularity when a problem is identified.	Topic 3, question 2
114	View internal system data	Users can view state machine states.	Topic 3, questions 1, 2 and 3

112	Gathering of device sensor data	Hardware device sensor data is gathered and viewable. This may include engine sound in addition to more traditional measurements.	Topic 1, question 3; Topic 2, question 3; topic 3, questions 1, 2 and 3
113	Ability to use the RDS during vessel commissioning	Affordances are made so that the RDS may be used during vessel commissioning with limited extra effort.	Topic 2, question 2
115	Value metadata	The meaning of individual values can be clarified through the inclusion of user-specifiable metadata. For example, the mapping of individual bits to specific labels.	Topic 4, questions 1 and 2

APPENDIX 6. Finnish format RDS prototype usability testing participant consent form

Suostumus etädiagnostiikkapalvelun prototyypin käytettävyystestaukseen osallistumiseen

Annan suostumukseni prototyyppi etädiagnostiikkapalvelun käytettävyystestaukseen osallistumisestani osana Samuli Siitosen LUT-yliopiston tietotekniikan koulutusohjelman ohjelmistotuotannon pääaineen diplomityön (2019–2020) ohjelmallisen toteutuksen käytettävyystestausta. Suostun vastauksieni hyödyntämiseen edellä mainitussa diplomityössä ja siihen liittyvässä ohjelmallisessa toteutustyössä. Vastauksiani käsitellään anonyymisti ja ne eivät ole yksilöitävissä minuun. Käytettävyystestauksesta tehdään muistiinpanot, jotka Samuli Siitonen tuhoaa diplomityön valmistuttua.

Minulla on oikeus perua osallistumiseni käytettävyystestaukseen ja vastauksieni hyödyntämiseen milloin tahansa ennen diplomityön valmistumista.

Paikka ja päivämäärä ___________________________

Testaajan allekirjoitus ___________________________

Testaajan nimenselvennys ____________________________