

PAPER • OPEN ACCESS

Integrating accountability in the systems design of autonomous and remote-controlled operations

To cite this article: Bård Myhre *et al* 2020 *IOP Conf. Ser.: Mater. Sci. Eng.* **929** 012020

View the [article online](#) for updates and enhancements.

The 17th International Symposium on Solid Oxide Fuel Cells (SOFC-XVII)
DIGITAL MEETING • July 18-23, 2021

EXTENDED Abstract Submission Deadline: February 19, 2021



SUBMIT NOW →

Integrating accountability in the systems design of autonomous and remote-controlled operations

Bård Myhre¹, Ørnulf Jan Rødseth² and Stig Petersen³

¹ SINTEF Digital, Oslo, Norway

² SINTEF Ocean, Trondheim, Norway

³ SINTEF Digital, Trondheim, Norway

Corresponding author: bard.myhre@sintef.no

Abstract. In a previous paper we have suggested that the transfer of human accountability from an on-site human actor (such as the captain) to a remote human actor (such as the creator of the autonomous control system) could be regarded as the defining characteristic of autonomous systems. In this paper we take this approach one step further, by suggesting a methodology for how accountability can be used as a basis for systems design of autonomous and remote-controlled operations. Furthermore, the suggested methodology is applied on a hypothetical case of a vessel supporting both autonomous and remote-controlled operation.

1. Introduction

The concept of autonomy is by many considered to be the next paradigm shift within transportation. Brought forth by technological advances in sensor technology, machine learning, and computing power, the vision of self-driving cars, vessels, and airborne drones has seemingly only been held back by the final technological break-through. However, over the last couple of years, some concern has been raised about the challenges that remain for the autonomous revolution to happen. In the automobile industry, executives of both Ford and Waymo have stated that fully self-driving cars may be decades away [1]. Within the maritime domain, IMO is performing a regulatory scoping exercise for the use of Maritime Autonomous Surface Ships (MASS), with targeted completion by 2020. The reported results from IMO [2] so far seem optimistic, but several complex challenges are to be resolved before autonomous vessels will be a commercial reality.

From a technical perspective, much effort is put into making self-driving cars and vessels detect obstacles, react properly to various situations, and enabling operation under a broad range of conditions. However, although autonomous devices technically speaking operate by themselves, the responsibility for their actions will still be held by humans. This expectedly means that either a driver, an operator, a technician, or an organization will be held accountable for any accidents caused by an autonomous car or vessel. This aspect is rarely touched upon by the technical community, although the question of responsibility is described as a "fundamental issue to be addressed" by IMO [2].

In a previous paper [3] we have suggested that the transfer of *human accountability* from an *on-site human actor*, such as the captain, to a *remote human actor*, such as the creator of the autonomous control system, could be regarded as the defining characteristic of autonomous systems. Even though questions of responsibility and liability have received increased attention over the last years [4], most research seems to focus on how fast humans can take over in emergency situation, ethical aspects and transparency of machine learning decisions, and how to properly test autonomous systems. Our notion



however implies that the concept of autonomy may benefit from also being associated with legal aspects rather than solely technical ones. Also, it may suggest that legal aspects such as certification should be the starting point of any design process involving autonomous systems, instead of being the final hurdle after the technology development is completed.

As classification schemes for autonomous ships are an area in constant development [5], we do not think that an accountability-centered approach towards autonomy will necessarily be the holy grail within the field of autonomy. However, we do hope that our perspectives may contribute to expanding the understanding on how autonomy could be regarded, and on how the design of autonomous systems may be approached. On this background, this paper aims to take the ideas from [3] one step further, by suggesting a methodology for how accountability can be used as a basis for systems design of autonomous and remote-controlled operations. Furthermore, the suggested methodology will be applied on a fictitious case of a vessel supporting both autonomous and remote-controlled operation.

To establish a common ground, section 2 of this paper will first give a brief summary of how accountability could be regarded as a starting-point for autonomy, as introduced in [3]. Section 3 will then suggest some definitions for central concepts related to non-conventional navigation and control. In section 4 we introduce the above-mentioned methodology for integrating accountability in the systems design of autonomous and remote-controlled operations, while section 5 walks us through an imaginary example case illustrating the practical use of the methodology. Section 6 will summarize our considerations and findings, before we conclude the paper in section 7 while also pointing towards some unresolved challenges and future work.

Accountability as it is presented in this paper can thus be seen as a variant of transfer of control as discussed e.g. in the context of automated cars [6]. However, as the paper will show, the novelty here is to use this as a high-level design principle rather than as a consequence of design decisions or as a fallback solution.

2. Accountability as a starting point for autonomous and remote-controlled operations

As mentioned in section 1, our previous paper [3] suggested a definition for *autonomy* that have been modified as follows:

An automated system is considered *autonomous* within a specified operational envelope if it can *lawfully accept accountability* for an operation, thereby assuming the accountability that was previously held by either a *human operator* or another *autonomous system*.

Note: As an automated system itself cannot be held *legally* accountable, the factual accountability for the actions of an autonomous system will necessarily reside with, e.g. the creators, installers or operators of the system, dependent on how the liability regime develops [7]. Furthermore, this definition implies that a human operator most likely can be considered to have exercised due care if she or he leaves control to the autonomous system, as long as the autonomous system has accepted accountability. The human operator will however be required to remain in control if the autonomous system declines to take control, and she or he must retake control if the autonomous system later alerts the operator of a possible future exit from the automatic control part of the operational envelope [5].

From the perspective of a system designer, the suggested definition of *autonomy* may have implications on how to describe, design and implement an autonomous system.

First, the capability of *lawfully accepting accountability* could be handled through a type of certification. Such a certification could be limited, for instance to certain operations, operational conditions, time of day, etc. It also means that some actors could be certified for parts of an overall operation only, and not for the entire operation.

Second, accountability can be transferred but not shared. This interpretation is in line with the concept of "Responsibility charting" introduced in [8], where the accountable person is the individual

who is ultimately answerable for an activity or decision. This means that there at any time must be one (and only one) actor holding the accountability for the ship.

Third, as accountability can be transferred between both human operators and other autonomous systems, this means that several actors can be certified to hold the accountability for an operation, although only one at the time.

Fourth, an autonomous system does not have to be physically located on the vessel it controls. Depending on its certification, autonomous systems could also be located remotely.

Fifth, the definition above implies that an accountable system is "on its own" in case of problems, when it is inside the automatic control part of the operational envelope [5], and that it cannot rely on other systems or actors for help if the situation comes out of control.

Sixth, if the accountable actor is not able to fulfil its obligations of controlling the vessel due to communication loss or other failure, the system should have a fallback to a Minimum Risk Condition.

3. Defining central concepts in non-conventional navigation and control

Although terms such as *autonomous* and *remote-controlled* may be used colloquially in everyday talk, defining concepts as stringent as possible is important to avoid ambiguities and misunderstandings. For the remainder of this paper, we will adhere to the following definitions:

- **Navigation and control (N&C) system.** A technical on-board system that enables navigation and control of a vessel, but without the functionality to do this automatically.
- **Classic Automated Ship Controller.** A control system that is technically able to provide input to a vessel's N&C system, but that cannot lawfully accept accountability for an operation. A traditional autopilot would fall into this category, as the captain would still be accountable for the vessel's operation.
- **Autonomous Ship Controller (ASC).** A control system that is technically able to provide input to a vessel's N&C system, and that can lawfully accept accountability for an operation, thereby assuming the accountability that was previously held by either a human operator (e.g. the captain) or another autonomous control system.
- **Remote-controlled vessel.** A vessel whose accountable control is held by someone not physically present on the vessel. A vessel can be remote-controlled both by remote-located human operators and remote-located ASCs.
- **Remote Control Centre (RCC).** An off-vessel location for remote-controlling vessels.
- **Remote Control Centre Operator (RCCO).** A human that is certified to remote-control a vessel under specific conditions and limitations, working physically or logically through an RCC.
- **Crewed vessel.** A vessel with a human operator on board that can lawfully accept accountability for the vessel.
- **Uncrewed vessel.** A vessel without a human on board that can lawfully accept accountability for the vessel.
- **Minimum Risk Condition (MRC).** A minimum risk condition (MRC) is a state that the vessel should enter in situations that are outside those in which it can operate normally, but is still expected to deal with in one way or another [9].
- **Minimum Risk Condition Function (MRC Function, or MRCF).** An MRC Function is an Automated (but not Autonomous) Ship Controller whose task is to navigate the vessel into an MRC if no accountable operator (either human or ASC) is in control of the vessel.

According to these definitions, there exist a wide range of variants of how a ship can be controlled. One could for instance imagine a **crewed vessel being controlled by an ASC**, enabling the on-board captain to be off duty and without any formal accountability for the vessel. Furthermore, in such a case the ASC could either be located on the ship or at the RCC, where the latter in practise would constitute an *Autonomous Remote Controller (ASC)*.

4. A 3+3 step methodology for high-level design of autonomous and remote-controlled systems

To successfully develop autonomous and remote-controlled systems, we have suggested the idea that accountability should be considered a fundamental capability. Integrating accountability in technical design is not necessarily intuitive, and we have therefore attempted to specify a two-phased design process. Each phase consist of three distinct activities, and the entire methodology could therefore be regarded as a 3+3 step procedure, as illustrated in Figure 1.

The first phase is the *concept phase*, whose objective is to establish the overall requirements for the system with regards to accountability. This phase consists of the following three stages: *i) actors and capabilities*, *ii) accountability states*, and *iii) control states*. The stages are logically to be developed in sequence, but iterations should be expected as the insight into the case evolves.

The second phase is the *high-level design phase*, which establishes the top-level design for fulfilling the requirements from the first phase. This phase consists of three more stages: *i) system architecture*, *ii) messaging scheme*, and *iii) module requirements*. Like for phase 1, the stages form a logical sequential order, but iterations should be expected also here.

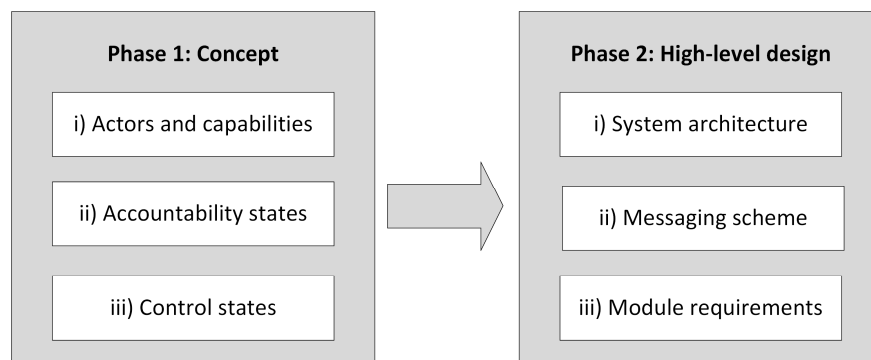


Figure 1. A 3+3 step methodology for high-level design of autonomous and remote-controlled systems

This principle can be embedded in the first design steps, either in a V-type or a more agile development process, e.g. using a spiral model with incremental refinement of the design and implementation.

5. A fictitious case: Autonomous and remote-controlled vessel with contingency function

To illustrate how the methodology can be applied in practise, we will describe and develop a fictitious case along the phases and activities from section 4. This will demonstrate where various design choices have been made during the overall process, and at some points we will also highlight some design choices that we deliberately *have not taken*. Note however that the presented case is not a real one, and that the objective of this section is to exemplify the suggested process. It should therefore be stressed that this paper is more in search of *relevant questions* than *correct answers*, as the actual answers for a real case would inevitably need to be developed with the real-word stakeholders.

5.1. Phase 1: The concept phase

The concept to be developed in this paper could be described as "an uncrewed vessel, with an RCCO that can assume accountability for the entire voyage, an on-board ASC that can accept accountability during sea passage (i.e. outside port), and an MRC Function to assume control from the RCCO if communication with the vessel is lost".

5.1.1. Actors and capabilities

The actors and their fundamental capabilities, along with their interactions, are illustrated in Figure 2:

- One **vessel**, with
 - One **N&C unit**, that enables physical control of the vessel; receiving input from either the **ASC**, the **MRC Function** or wirelessly from the **RCCO**.
 - One **ASC**, that may accept accountability for the vessel *during sea passage only*¹.
 - One **MRC Function**, that shall assume control (without accepting accountability) if input from RCCO is lost.
- One **RCC**, with
 - One **RCCO**, that may accept accountability for the vessel during *the entire voyage*².

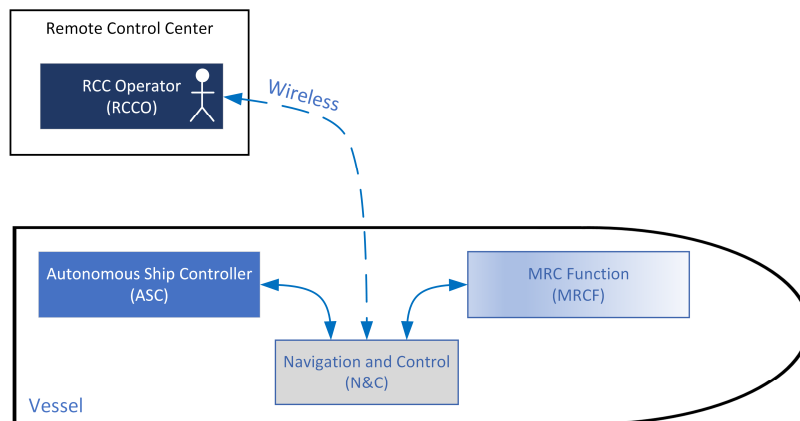


Figure 2. Actors and capabilities

Note that we at this stage only consider overall functions and capabilities, and that the actual system architecture will be left for phase 2. For the remainder of phase 1, we will therefore exclusively consider the entities that have been listed above.

5.1.2. Accountability states

Both the RCCO and the ASC may accept accountability for the vessel, with the ASC having certification for sea passage only. The interaction between the two units is however still to be decided: Will one of the entities have higher rank than the other, meaning that it may reclaim accountability at any time? And if one entity has higher rank than the other; shall the entity with lowest rank be able to offer accountability to the entity with higher rank, or should it simply wait for the entity with highest rank to reclaim accountability?

For the sake of the example case, we have made the following design choices:

- The RCCO shall have higher rank than the ASC, meaning that the RCCO at any time can reclaim accountability from the ASC³.
- The ASC shall keep accountability until the RCCO has reclaimed accountability, and the ASC shall thus not formally offer accountability to the RCCO.

A visualisation of the accountability states and transfer conditions are provided in Figure 3.

¹ Note that the operational limits of the ASC should be formally defined in a real case.

² Note that the operational limits of the RCCO should be formally defined in a real case.

³ Note that if the RCCO wants the ASC to reassume accountability after a reclaim operation, this requires a new acceptance from the ASC.

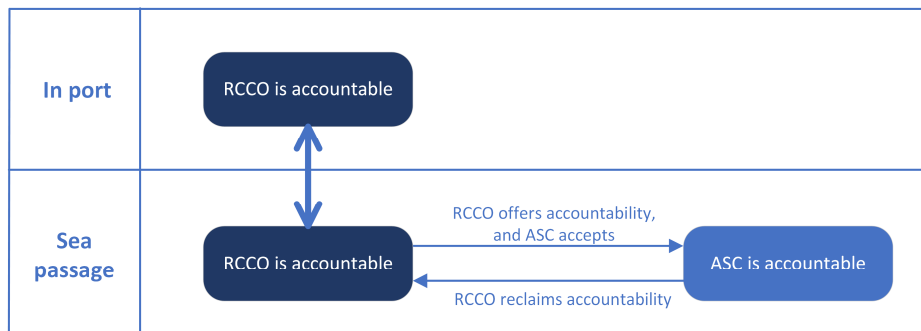


Figure 3. Accountability diagram with states and transfer conditions

Note that the MRC Function is not included in the accountability diagram in Figure 3, as it cannot accept accountability for the vessel. In our imagined case, the MRC Function is solely a backup if the RCCO loses control. Hence, the RCCO should still be considered accountable in cases where the MRC Function is enabled. Also observe that since the ASC cannot enter port areas, this means that the RCCO must actively monitor the situation in order to timely reclaim accountability before the vessel reaches the end of sea passage. If not, the ASC will simply keep the vessel outside the port area. Another design option would be to let the ASC formally request a transfer of accountability back to the RCCO when reaching the end of its operational envelope. In terms of illustrating the methodology, this functionality would however add unnecessary complexity to the accountability diagram in Figure 3 as well as the subsequent design phases without contributing correspondingly in value.

5.1.3. Control states

Having defined accountability as a fundamental capability in an autonomous and remote-controlled operation, it is vital that there is no doubt as to which entity *provides input* to the N&C of the vessel, and which entity holds the *accountability*. As the RCCO, the ASC and the MRC Function may all provide input to the N&C, there must exist a predefined logic of how to select the input for the N&C.

In our case, we define the input requirements as follows:

- Initially, the RCCO holds accountability and thus provides input to N&C.
- If the RCCO provides input to N&C and communication between RCCO and the vessel fails, then the MRC Function shall provide input to N&C.
- If the MRC Function provides input to N&C and communication between the RCCO and the vessel is restored, then the RCCO shall provide input to N&C.
- While in sea passage, the RCCO may offer ASC to assume accountability.
- If, and only if, the RCCO has offered ASC to assume accountability, ASC may accept (or implicitly refuse, by not accepting) the offer for accountability.
- If ASC accepts accountability, ASC shall provide input to N&C.
- If ASC provides input to N&C, the RCCO may at any time reclaim accountability. If the RCCO reclaims accountability, the RCCO shall provide input to N&C.
- The RCCO may at any time withdraw an offer to ASC of assuming accountability. This equals to reclaiming accountability from ASC.

It should be noted that the requirements above do not describe what to do if the ASC would fail. Such a situation could for instance be handled by using the MRC Function as fallback (with the ASC keeping accountability), but one could also imagine that an on-board ASC is required to provide its own, built-in MRC Function. Such considerations would be vital in a real case but are left out of our example to maintain a manageable scope.

The defined requirements can be translated into a more detailed *control state diagram* for the vessel, as presented in Figure 4. Here, the dark blue boxes indicate that the RCCO provides input to the N&C, the medium blue state indicate that the ASC provides input to the N&C, while the light blue states indicate that the MRC Function provides input to the N&C. Note that the RCCO is still considered accountable when the MRC Function provides input to the N&C.

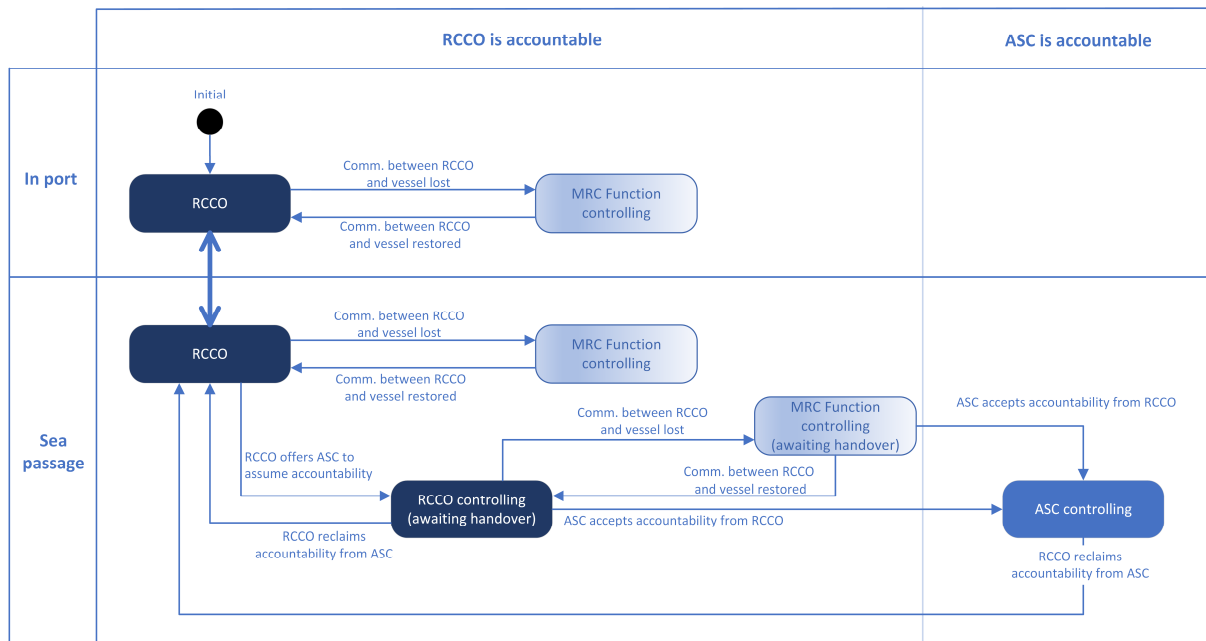


Figure 4. Control state diagram, with states and transfer conditions

In practise, these requirements mean that losing communication between the RCCO and the vessel does not affect a situation when the ASC is accountable.

Note that although these requirements may seem obvious at first glance, there are several other requirements that *could have* been stated. For instance, one could imagine a requirement that enabled the ASC to consider claiming accountability if communication between the RCCO and vessel is lost, instead of simply enabling the MRC Function. This would allow for a smoother operation in the cases where communication is lost during sea passage but is for simplicity left out of our case.

5.2. Phase 2: High-level design

The objective of the high-level design phase is to establish a technical system architecture as well as requirements for module interaction and behaviour. This is achieved by systematically going through the three steps of this phase.

5.2.1. System architecture

There exist numerous possible system architectures for realising the requirements from Phase 1, and it is not within the scope of this paper to perform a thorough evaluation of these. Some architectures still deserve specific attention. Figure 5 shows an architecture where the RCCO communicates directly with the MRC Function, and where the ASC and the MRC Function have separate inputs to the N&C. This scheme is visually intuitive but requires adding logic related to the N&C input. Figure 6 shows another solution, where the MRC Function is included in the ASC, and where the ASC holds the communication with the RCCO. This scheme may be regarded as a natural consequence of the idea that the RCCO and the ASC cooperate in controlling the ship. However, in this case the ASC would serve both as ASC, MRC Function and switching board for the N&C. Figure 7 declutters Figure 6 a bit, by separating the MRC Function and ASC, but still resembles a sequential "switching board" of input signals. This finally

leaves us with Figure 8, which is inspired by both the Central Processing Unit (CPU) of modern computers, as well as the philosophy of "separation of concerns" brought forth by e.g. Dijkstra [10]. Here, we introduce the concept of an N&C Input Manager (NCIM), handling the communication between all the various entities as well as doing the selection of input for the N&C. This centralised architecture ensures that all distinct functions from Phase 1 are kept within separate functional modules, and also enables us to keep the responsibility for selecting the correct input for N&C within one single entity, i.e. the NCIM. From a practical perspective, it should be noted that such a distributed architecture does not hinder combining two or more modules into one single product⁴.

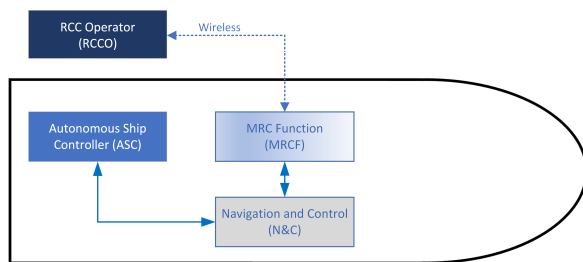


Figure 5. Parallel input management

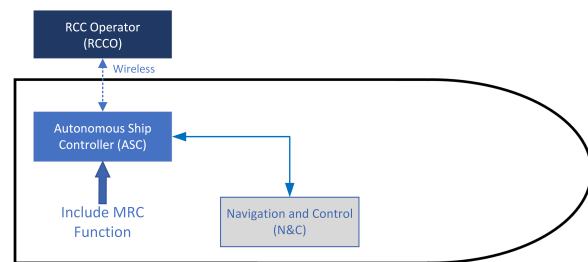


Figure 6. Sequential input management, with ASC providing MRC Function

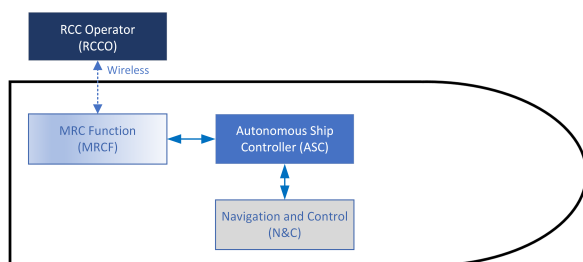


Figure 7. Sequential input management, with separate MRC Function

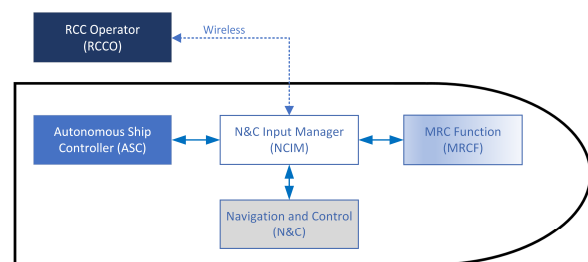


Figure 8. Centralised input management

Within the scope of this paper, we make the following decision for system architecture:

- The requirements stated in Phase 1 shall be realised through a centralised system architecture, where a NCIM will serve as a hub for all communication between the RCCO, the ASC, the MRC Function and the N&C. This architecture is illustrated in Figure 8.

5.2.2. Messaging scheme

One of the benefits of the centralised architecture illustrated in Figure 8 is that the control state diagram from Figure 4 can be directly converted into a state diagram for the NCIM. Each transition must however be investigated with care, to ensure that the relevant triggers are implemented and that corresponding messages are created for the relevant modules. Figure 9 shows one possible implementation of the NCIM state diagram, where each transition is assigned with a triggering event and corresponding action(s). The *triggering events* are denoted in two ways; either with hyphens ("-"), indicating a decision made within the current module, or angle brackets ("< >"), indicating a message from another module. The *corresponding actions* are denoted immediately after the slash ("/"), indicating message recipients (following the @ sign) and a message description.

⁴ Interestingly, the architecture in Figure 6 could be regarded as one possible implementation of the architecture in Figure 8, with the former presenting the ASC, NCIM and MRC Function as one combined unit.

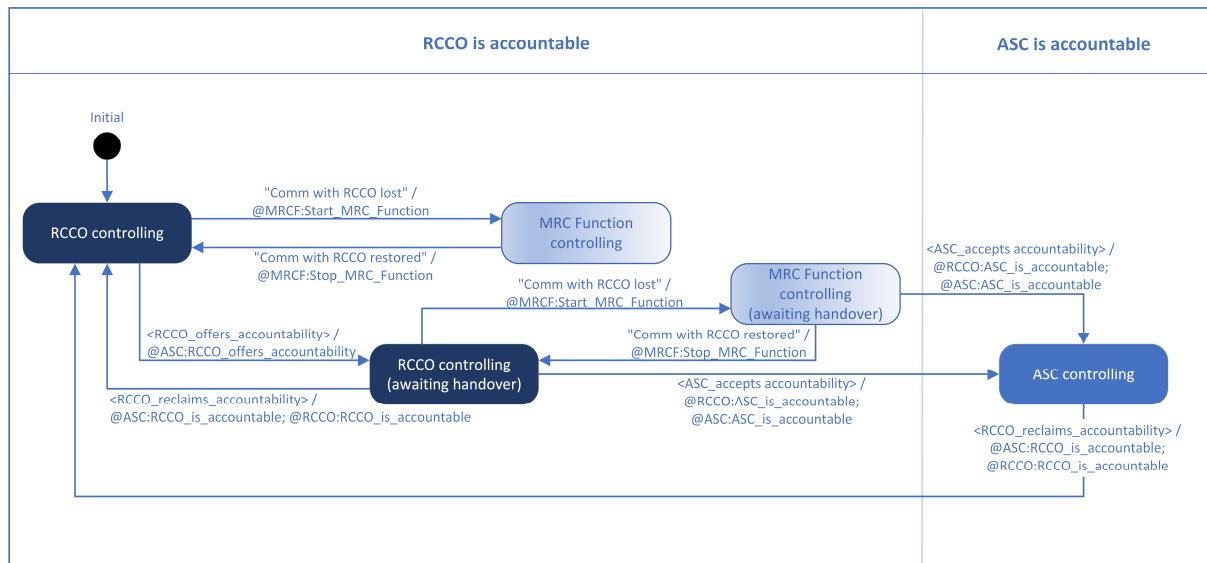


Figure 9. State diagram for the N&C Input Manager (NCIM)

To ensure correct operation, the following decisions must be made by the NCIM:

- "Communication with RCCO lost"
- "Communication with RCCO restored"

The following messages are required from the RCCO:

- <RCCO_offers_accountability>
- <RCCO_reclaims_accountability>

The following message is required from the ASC:

- <ASC_accepts_accountability>

Finally, the following messages must be communicated from the NCIM to the other modules:

- @MRCF:Start_MRC_Function
- @MRCF:Stop_MRC_Function
- @ASC:RCCO_offers_accountability
- @ASC:RCCO_is_accountable
- @RCCO:RCCO_is_accountable

It should be noted that the only two stand-alone decisions to be made by the NCIM are related to the status of the communication with the RCCO. This means that the communication between the NCIM and the RCCO requires a mechanism for timely detection of communication loss, e.g. in the form of a heartbeat. As detecting this communication loss would in most cases be considered safety-critical, one might want to investigate established standards for safe communication within other domains, such as EN 50159 [11] for railway signalling.

5.2.3. Module requirements

The final stage of the 3+3 methodology is to establish requirements for the individual modules. The requirements for the NCIM are already established as a part of the messaging scheme, but the other

modules remain. However, their internal logic can for the most part be derived directly from the NCIM state diagram and message requirements as presented in section 5.2.2.

State diagram for the MRC Function

The state diagram for the MRC Function is illustrated in Figure 10. This module has only two operational states, and it is in practise controlled directly by the NCIM.

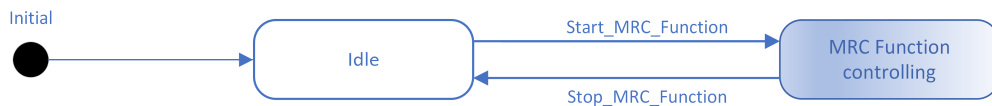


Figure 10. State diagram for the MRC Function

State diagram for the ASC

The state diagram for the ASC is illustrated in Figure 11. This module receives requests for accountability from the RCCO and needs to evaluate whether these requests should be accepted. Note that the ASC will have to provide control information to the NCIM as soon as accountability is accepted, although it will have to wait for the confirmation from the NCIM to know that it has actually received accountability. This is to ensure that an accountable entity will always provide information to the N&C.

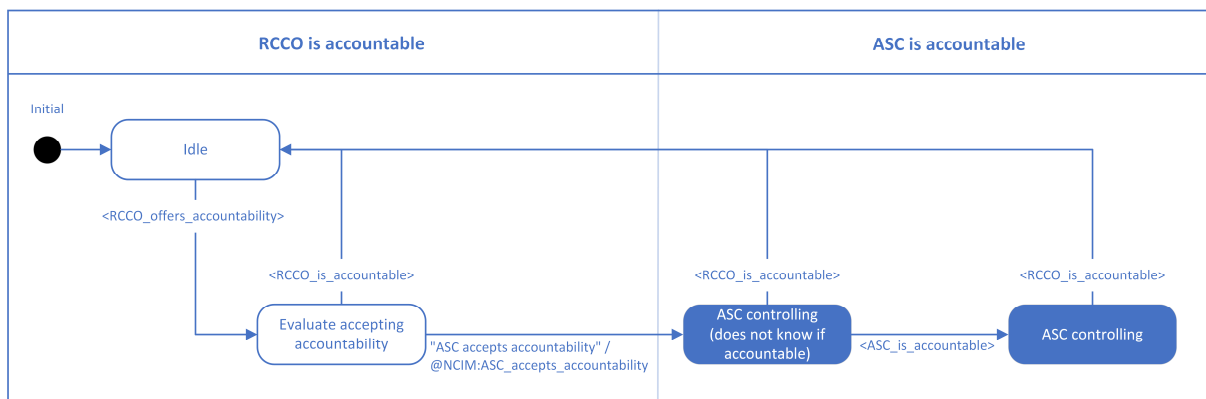


Figure 11. State diagram for the Autonomous Ship Controller (ASC)

To ensure correct operation, the following decision must be made by the ASC:

- "ASC accepts accountability"

Also, the following message must be communicated from the ASC to the NCIM:

- @NCIM: ASC_accepts_accountability

State diagram for the RCCO

The state diagram for the RCCO is illustrated in Figure 12. As the RCCO is a human being, keeping track of the states should ideally be handled digitally by a management system.

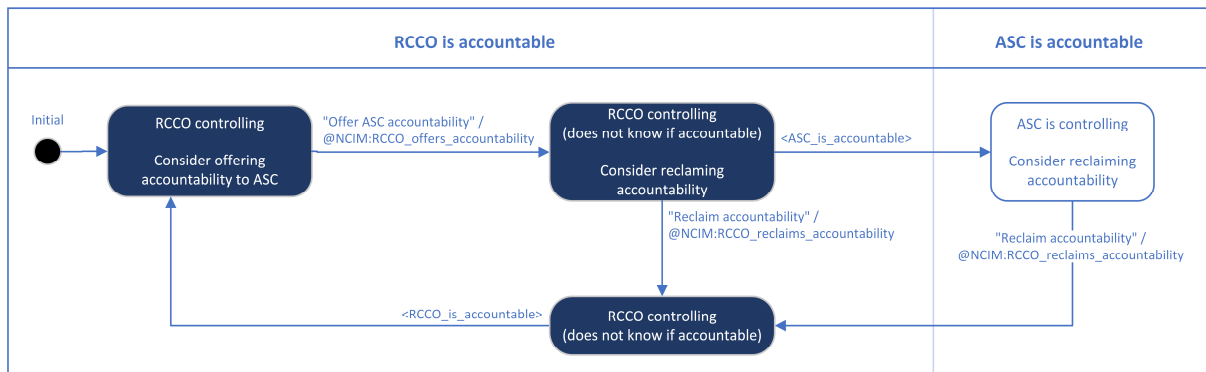


Figure 12. State diagram for the RCCO

To ensure correct operation, the following decisions must be made by the RCCO:

- "Offer ASC accountability "
- "Reclaim accountability"

Also, the following messages must be communicated from the RCCO to the NCIM:

- @NCIM:RCCO_offers_accountability
- @NCIM:RCCO_reclaims_accountability

One might further expect that the RCCO would like to know whether communication with the vessel is working correctly. This can be handled by a separate service (and hence a separate state diagram), as illustrated in Figure 13. Note that *knowing* the communication status will not influence the RCCO's ability to control the vessel.

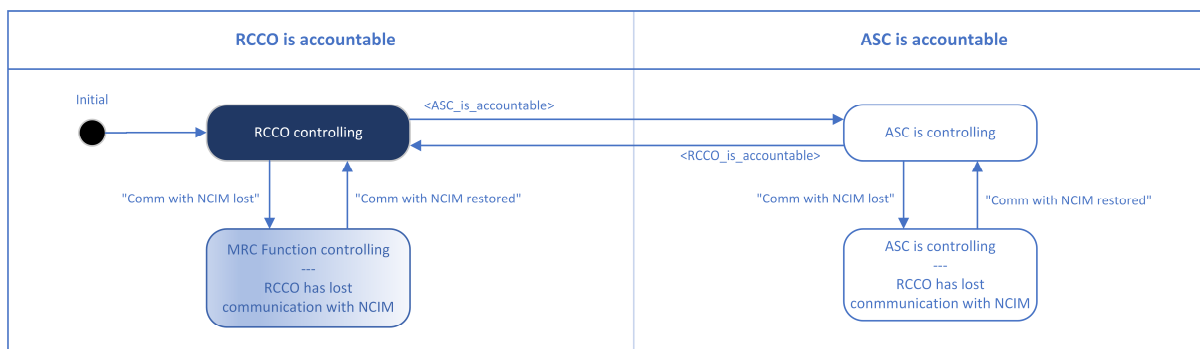


Figure 13. State diagram for the RCCO's communication status

To ensure correct operation, the following decisions must be made by the RCCO:

- "Communication with NCIM lost"
- "Communication with NCIM lost"

6. Considerations on the presented methodology

As mentioned in section 1, much research can be found on specific functions of autonomous systems, and on technologies that are assumed to be relevant for the success of autonomy. This paper has a different approach, and by defining *accountability* as a starting point for technological development, we have aimed to illustrate how it can be possible to integrate various *accountable entities* (or systems) into what could be regarded as an *accountable system-of-systems*. This also means that the presented

methodology should not be considered a substitute for other design methodologies, but rather seen as an attempt to combine various aspects such as remote-control, autonomy and contingency operations into one common framework, with accountability as a common denominator.

7. Conclusions

By employing the methodology introduced in this paper, we have attempted to integrate *accountability* into the high-level technical design of a hypothetical autonomous and remote-controlled vessel. One must however expect new challenges to arise as soon as the methodology is applied to real cases, and this paper should therefore be seen more as an investigation into new ways of approaching autonomy than an actual and definite solution.

From a research perspective, the topics presented in this paper can be followed up in several directions. Naturally, applying the methodology to a real case would be a logical first step. Also, our choice of system architecture deserves further evaluation, both from an implementation perspective and from the viewpoint of systems design and analysis. Finally, to ensure compatibility with existing practices, one should also investigate how the proposed methodology could be integrated with existing legal and technical regulatory frameworks.

Acknowledgments

The work presented in this text has received funding from the European Union's Horizon 2020 research and innovation programme's GSA H2H project under grant agreement No. 775998, the AUTOSHIP project under grant agreement No. 815012 and the AEGIS project under grant agreement No. 859992 as well as from the Research Council of Norway's Sacomas project under grant agreement No. 269510.

References

- [1] Wired Magazine (2019) Ford taps the brakes on the arrival of self-driving cars <https://www.wired.com/story/ford-taps-brakes-arrival-self-driving-cars> (Retrieved 2020-04-26)
- [2] Tunfors H (2020) IMO regulatory scoping exercise on MASS *5th UK Maritime Autonomous Systems Regulatory Working Group Conference* https://www.maritimeuk.org/documents/544/Henrik_Tunfors_suw9wJK.pdf (Retrieved 2020-04-26)
- [3] Myhre B, Hellandsvik A and Petersen S 2019 A responsibility-centered approach to defining levels of automation *J. Phys.: Conf. Ser.* **1357** 012027 <http://dx.doi.org/10.1088/1742-6596/1357/1/012027>
- [4] Alawadhi M, Almazrouie J, Kamil M and Khalil K A 2020 Review and analysis of the importance of autonomous vehicles liability: a systematic literature review *Int J Syst Assur Eng Manag* <https://doi.org/10.1007/s13198-020-00978-9>
- [5] Rødseth Ø J (2018) Defining ship autonomy by characteristic factors *Proc. of the 1st International Conference on Maritime Autonomous Surface Ships (ICMASS 2018)*
- [6] Molnar, L. J., Ryan, L. H., Pradhan, A. K., Eby, D. W., Louis, R. M. S., & Zakrajsek, J. S. (2018). Understanding trust and acceptance of automated vehicles: An exploratory simulator study of transfer of control between automated and manual driving. *Transportation research part F: traffic psychology and behaviour*, 58, 319-328.
- [7] Kuro, J. (2018). *Maritime Product Liability The Case of Unmanned Vessels*, Master's thesis from University in Oslo.
- [8] Smith M L, Erwin J and Diaferio S (2005) Role & responsibility charting (RACI) *Project Management Forum (PMForum)* **Vol. 5**
- [9] DNV GL (2018) Class guideline DNVGL-CG-0264: Autonomous and remotely operated ships
- [10] Dijkstra E W (1982) Selected Writings on Computing: A Personal Perspective (New York: Springer-Verlag) pp 60-66
- [11] CENELEC (2010) EN 50159:2010 Railway applications – Communication, signaling and processing systems. Safety-related communication in transmission systems.