

## Article

# From Offshore Operation to Onshore Simulator: Using Visualized Ethnographic Outcomes to Work with Systems Developers

Yushan Pan <sup>1,2,\*</sup>  and Sisse Finken <sup>3</sup>

<sup>1</sup> Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology, 6025 Ålesund, Norway

<sup>2</sup> Department of Informatics, University of Oslo, 0316 Oslo, Norway

<sup>3</sup> Technologies in Practices Research Group, Department of Business IT, IT University of Copenhagen, DK-2300 Copenhagen, Denmark; sisf@itu.dk

\* Correspondence: yushan.pan@ntnu.no Tel.: +47-9-735-8841

Received: 19 December 2017; Accepted: 6 February 2018; Published: 9 February 2018

**Abstract:** This paper focuses on the process of translating insights from a Computer Supported Cooperative Work (CSCW)-based study, conducted on a vessel at sea, into a model that can assist systems developers working with simulators, which are used by vessel operators for training purposes on land. That is, the empirical study at sea brought about rich insights into cooperation, which is important for systems developers to know about and consider in their designs. In the paper, we establish a model that primarily consists of a ‘computational artifact’. The model is designed to support researchers working with systems developers. Drawing on marine examples, we focus on the translation process and investigate how the model serves to visualize work activities; how it addresses relations between technical and computational artifacts, as well as between functions in technical systems and functionalities in cooperative systems. In turn, we link design back to fieldwork studies.

**Keywords:** system design; computer-supported cooperative work; social computing

## 1. Introduction

There is a long tradition in which Computer Supported Cooperative Work (CSCW) researchers [1–3] have used insights about cooperative work to inform systems design [4–6]. In this tradition, ethnographic fieldwork has been a driver for creating changes in systems design by way of shifting the focal point from developers’ visions to the inclusion of real-world use [7,8]. CSCW researchers study cooperative work practice and how it is supported by technical systems [9]. They also communicate and work with workers to ensure that the final products fulfill the needs for the specific workspace [10]. In this case, systems design informed by CSCW studies [11] concentrates on descriptions of social aspects of using systems with limited technical guidance, such as systems architectures [12] and patterns of work [7,13] for systems developers. Although there are studies concerned with how to design cooperative systems, including healthcare information systems [9], agile software development [14], cooperative mobile work [15], offshore outsourcing [16], and team coordination training in high-risk workspaces [17], systems developers focus solely on the logic and data access layers of systems’ shapes [18]. This longstanding historical issue is commonly characterized as the difference between CSCW researchers’ point of view and the systems developers’ synthesis solutions [6].

With the growth of maritime technologies, marine operations require operating systems to support marine operators’ cooperative work. The Norwegian ship industry pays attention to cooperative work of marine operators since ship-owners in the industry are concerned with how to update

their mechanical and partially digitalized ship bridges to make them better places for cooperative work. Thus, the present study was conducted as a part of the first author's Ph.D. project offering a method to support these requirements [19,20] and assist the local maritime industry. However, traditionally, maritime technologies have been designed as mechanical engineering systems to make them more automated, thereby leaving out cooperative work of marine operators from systems design [21]. For marine navigation, such an approach may be satisfactory, since most navigation work is automatically carried out by technical systems [22], enabling the vessel to self-manuever from one place to another. However, marine operations are different; they involve offshore operations in which marine operators work cooperatively [22] on the ship bridge as well as with people on oil and gas platforms. Thus, an important layer, when designing systems for marine operations, is to consider the cooperative work of marine operators.

While responding to the demands of cooperative work in marine operations, the maritime industry integrates different pieces of operational systems to support cooperative work, such as developing maritime simulators as prototypes for training purposes before making a real workspace on the ship bridge [23]. For example, systems developers understand the design of simulators as positioning software systems, hardware systems, and machinery in combination to create a space for marine operators' cooperative work, in order to prepare marine operators on land before they work at sea [24,25]. Among other reasons, systems design work is performed to see how each system communicates and cooperates with the other systems [26]. In addition, systems developers understand that cooperative work is a way of assembling individual tasks into a mechanical system to support cooperation in technical systems. Unfortunately, safety issues at sea still do challenge marine operators in their daily work, although they are regularly trained using maritime simulators [27]. Given such challenge, the shipping industry is striving to solve cooperative work problems to provide a safer cooperative work workspace for marine operators. Thus, we argue that systems design should prioritize cooperative work practices. It is important to re-engineer maritime simulators to support the cooperative work of marine operators before they move into the real workspace [2,28–30]. However, it is also important to note that systems developers in the maritime domain are systems architects, software engineers, electrical engineers, mechanical engineers, and telecom engineers. Thus, incorporating support for cooperative work in systems design may require efforts beyond collaboration between systems designers and ethnographers, which is a subject focus within CSCW studies [6,12,31].

In this article, we seek to understand how ethnographic outcomes can be translated and considered useful for systems developers. We utilize the concept of 'computational artifact' [32,33] to visualize the ethnographic outcomes from the fieldwork to conduct three workshops with three systems developers in face-to-face sessions. We point out that, although CSCW may have paid less attention to inspiring systems developers in their hands-on work [30,34], our work can be used as a starting point for guiding interactional relationships in real work settings. For example, CSCW researchers can address relations between artifacts and work practices, and combine them as a computational artifact (named 'object' in this paper). They can deal with relations between functions in technical systems and functionalities in cooperative systems. They can also deploy such relations to support cooperative work in systems design and link design back to fieldwork studies of practice. They could make efforts to ensure that the 'what-you-see-is-what-I-see' rule is followed in interactions between CSCW researchers and systems developers. By adopting such rule, marine operators' cooperative work regarding development requirements [35,36] can be understood as modeling of systems' architectures in the engineering process.

Although CSCW has been trying to establish how work practices could be integrated into design, in this paper, we explore how to fruitfully unfold work practices between systems developers and researchers, while using the working language of the systems developers in tandem with examples from qualitative field material. Thus, this paper not only contributes to CSCW, but also shows how to work *with* engineers to support their work. It is important to note that we do not aim to design maritime simulators, nor do we offer a theory of design for CSCW research; rather, we seek to understand how

systems developers can use ethnographic outcomes from a CSCW-based study in systems design. We examine insights that can serve as mediators between operators at sea (who work in the maritime workspace) and systems developers (who work with designing maritime simulators).

This paper is structured as follows: Section 2 presents related work in both CSCW and systems design. Section 3 delineates the empirical settings, the methods used and the data analysis. Section 4 presents the process of establishing a model that can support the researcher when working with systems developers. Section 5 presents the workshops and findings. Section 6 presents the impact of the study in industry. Section 7 discusses and concludes how our work shifts the emphasis from describing ethnographic findings to visualizing them as resources when working with systems developers. In turn, such working process helps reflect how social-technical solutions can be developed, with insights from social aspects, to support engineering work, thereby linking ethnographic outcomes to support systems developers.

## 2. Literature Review

Suchman [37] argues that systems design should represent human activities in technical systems. In line with this, the European tradition of CSCW [1] aims at understanding cooperation [38] to inform design [6,39]. To accomplish this, researchers seek to represent work practices, workspaces, and problems in technology use for systems developers, expecting that such informed requirement specifications will be implemented [29]. For example, researchers have outlined ethnographic methods and their relationship with systems design [40]. They have noted that approaches like “quick-and-dirty” or “lightweight” ethnography, concurrent ethnography, and evaluative ethnography can each contribute substantially to this endeavor by enabling a shared understanding between ethnography and systems design and providing effective critiques that recognize problems in technology use. Thus, CSCW promotes design work that seeks to discover “something more” [41] from fieldwork than simply accumulating practical guidelines for systems developers [42].

Some CSCW scholars have attempted to encourage the use of more convenient tools by systems developers, such as employing pattern languages [13,43,44], linking ethnomethodology to systems design [45,46], deploying a notepad for supporting and understanding cooperative design between designers [47], modeling human activities to inform design [48], and addressing the paucity of papers detailing specific design guidelines [42]. However, the insights from CSCW studies still lack hands-on guidance, focusing instead on the systems developer’s collaboration [49]. In other words, we may not know how to translate our ethnographic outcomes into systems-relevant materials. Furthermore, we lose the ability to work with systems developers. Studies published to date have yet to identify how cooperative work can be organized as an interactional accomplishment for informing systems design; some researchers have called this the missing “interactional what” [12]. How can those systems’ shapes be broken down into designable units that are suitable for use in engineering work?

Some researchers have argued that CSCW insights can shed light on systems design in different ways [7,8,50], but, in the maritime domain, CSCW researchers have yet to participate in systems design for a variety of reasons. One explanation is that systems design researchers view CSCW methods as insufficient when it comes to making actionable contributions to systems engineering [30]. Systems developers are unable to follow interpretations of cooperative work to program engineering systems [30]. Thus, there is a missing link between CSCW research and systems design [6]. In existing systems design, the technical approaches to supporting cooperative work use a group of computer facilities [51]. From this perspective, cooperative work is understood as supported by digital communication channels and devices, linking various software and hardware systems in product design, and enabling systems developers to evaluate the manufacturability and assembly ability of software and other technical tools at the early stages. This approach focuses on how systems developers are enabled to manipulate current technology for analyzing, simulating, and optimizing individual and controllable technical artifacts to be integrated into the systems to enhance their mechanical functioning [52]. Distributed computing [53], agent-based collaborative computing, and autonomous

and agent computing [54] are the typical approaches used in the systems engineering field when designing cooperative systems. Along with these approaches, design applications are employed by systems developers; for instance, the building information modeling method (BIM) [55] combines some features of the computer-aided design, product data management, and product lifecycle management into mechanical engineering. BIM can also facilitate collaboration among stakeholders during the design, construction, and maintenance of buildings and facilities [51].

Systems design may be effective for processing information or quantitative data of systems architecture [56] by handling the relations among different applications and pieces of information from cooperative work. However, the assembly ability of software and other technical tools to increase a product's mechanical functions may not be as helpful as systems developers anticipate [1] in supporting operators' cooperative work. On the contrary, operators are the ones who can discover new meanings about artifacts in the systems development process [57]; their innovations are more likely to improve the workspace. However, it is not easy to access an offshore vessel without specific permission [58]. This restriction applies to all people outside the maritime industry, including researchers and systems developers who research and design maritime simulators to support cooperative work on the ship's bridge for marine operations. Hence, the first author managed to get access to an offshore supply vessel with permission from a shipping company having an agreement with the university about supporting regional research<sup>1</sup>. In this manner, a ship in the North Sea and Norwegian Sea (the working place) was boarded to learn about the work practices of marine operators. The outcome of the field work at sea was later on used as resources to work with systems developers.

### 3. The Empirical Settings, Methods Used and Data Analysis

Here, we describe the fieldwork settings, the methods used for generating data, and the data analysis. One of the settings is at sea on the bridge of a vessel and this we call phase one; the other is on land, comprising workshops conducted with maritime systems developers and the simulators they work on, and this we call phase two. With this, the work presented here is carried out by the first author as part of his Ph.D. project, which is concerned with simulator design, including operational systems, and physical settings. Overall, the project seeks to move constructively beyond criticism to offer specific solutions for systems design. In line with the Norwegian rules for the research on human-based projects, the first author applied to the Norwegian Center for Research Data (NSD) for approval. After receiving ethical consent from the NSD, the study began in the fall of 2013, and it is now complete. The project was conducted using methods like observations, interviews, photos, and workshops.

The work presented here is primarily from the second phase of the project. However, firstly, a short introduction to the workspace at sea is presented. Then, secondly, the simulators and workshops on land are presented. Following this, thirdly, the methods for data collection and analysis are described.

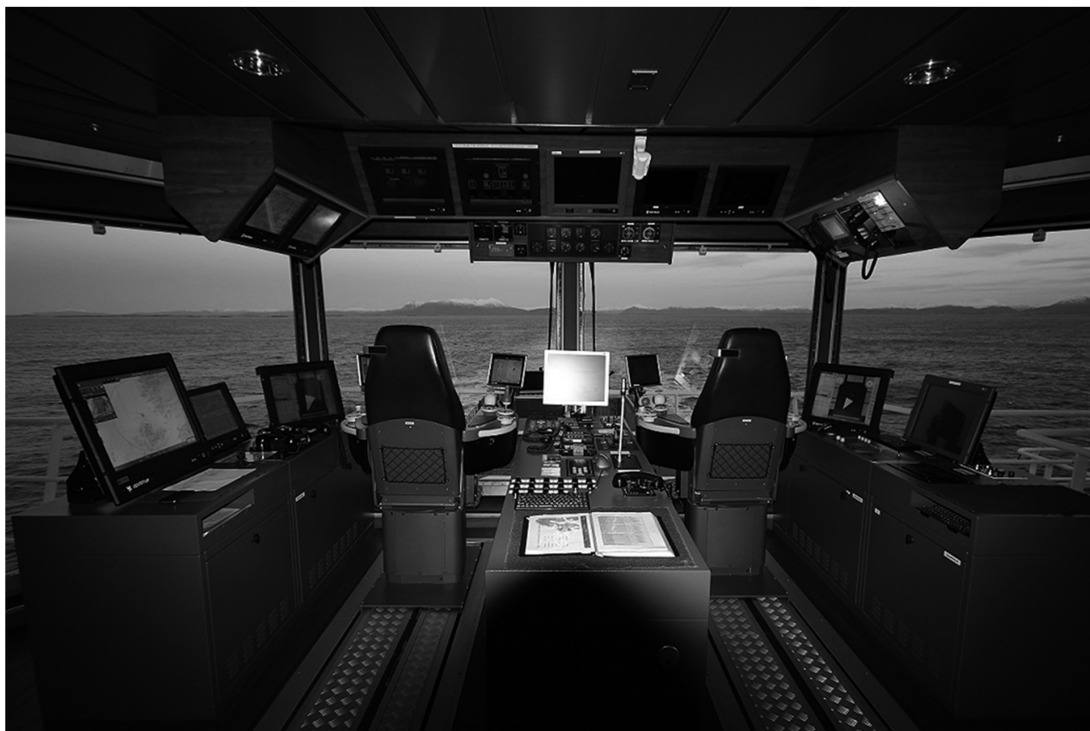
#### 3.1. On Board

On board, observations were carried out for six months, starting in spring 2015, on an offshore supply vessel. The study occurred on 56 days spread over six voyages, for a total of 633 h. Each fieldwork session was eight days on average. The first author interviewed [59] and observed two teams of marine operators working on a ship bridge. The two teams worked alternating 6-h shifts both day and night. The purpose of performing observation [60] was to understand how cooperative work is conducted in each marine operation and how safety issues are considered when carrying out cooperative work. Marine operations are complex and marine operators must cooperate with many other actors; the deck crews and other technicians who work on board and assist the marine operators on the ship bridge were also taken into consideration for this study.

In the workspace on a ship's bridge (see Figure 1), marine operators work cooperatively and simultaneously. Their workspace has two main types of operational systems, namely dynamic positioning (DP) systems and automation integrated systems (AIS). The DP systems are located



on the armrests of two chairs. DP is used to ensure that the vessel is in the correct location at sea. The screen in the middle and the two screens on the top (front) of the left chair are AIS interfaces. The AIS is used for shifting water and mud from under the deck and supplying them to the oil platform. On the top (front) of the right chair, the three screens are the navigation and radar interfaces. The two screens on the left cabinet are interfaces for monitoring systems that check the external workspace of the ship's bridge. The cabinet located between the two chairs has buttons and devices for the fire alarm and communication systems. The two screens on the right cabinet are usually not used for marine operations; instead, they are used for assisting navigational operations. The two screens on the left top (side) are interfaces for the duplication of DP systems. The telephone on the right top (side) is used for communication with the onshore control center. There are other artifacts in the workspace, such as a keyboard used to type in messages to the AIS and an alarm clock and calculator used for computation and notification. This paper focuses largely on the DP, AIS, and artifacts used for marine services for an offshore oil platform.



**Figure 1.** The operational area on a ship bridge (Copyright: Yushan Pan).

### 3.2. Simulators and the Workshop Setting

Before moving into the workshop setting, we first offer a short background to maritime simulators to clarify the relations between simulators and the workspace on a ship's bridge. The simulators are situated on a university campus. They also have DP systems and AIS at similar positions as in the real workspace (see Figure 2). The simulators are used to train marine operators regularly, in accordance with the safety regulations of the International Maritime Organization. They are also used as prototypes for developing maritime products for the future. For these reasons, maritime simulators must replicate the real workspace as closely as possible. With this, fieldwork in the real workspace at sea can provide valuable insights that can serve as resources for improving maritime simulators.



**Figure 2.** Maritime simulators (Copyright: Offshore Simulation Center, used with permission).

It is important to note that the maritime simulators were fully developed and already operating before the first author started participating in the systems design process. Maritime operators were recruited to practice on the simulators in marine courses in 2014 and 2015 by the Department of Nautical Studies at Norwegian University of Science and Technology; each course package lasted about two weeks, for a total of four packets. The purpose of this training was to help marine operators learn how to operate simulators safely and cooperatively before they work on an offshore vessel at sea [61]. As noted above, maritime simulators are developed by way of linking various products in product design with the purpose of supporting cooperative work. In training, the course instructors describe how to use the maritime simulators cooperatively; for example, they describe how the trainees can use the simulators and communicate through their cooperation. At the end of each course, the instructors evaluate how the collaboration among maritime operators occurred and whether it was effective, using metrics like time on task, errors, and communications. Utilizing the instructors' feedback from the maritime courses as requirement specifications, the systems developers gain insights into the simulators and try to debug errors, improve the ability of the interactive interfaces, and solve ergonomic problems concerning the physical tools and equipment, such as the brightness of the displays. It is important to clarify that the issue of safety in this cooperative workspace of the simulators, including the effectiveness of operational systems' functions, is not part of the training.

The first author (henceforward the researcher) met with several maritime systems developers to investigate how CSCW insights could help them in designing simulators for cooperative work purposes. The researcher worked on the empirical material from his fieldwork at sea and tried to convert the outcomes into a form, which can be understood in engineering language, to work with systems developers. Beyond this, he also crafted a model aiming at eliminating the barrier between CSCW and systems design, as well as finding a common place where the model could serve to re-format ethnographic outcomes as system related materials or as an artifact familiar to systems developers.

The model was brought along to three workshops arranged by the researcher. Here, it served as a reminder, to him, on how to describe and explain the concept of artifact to the systems developers. After the workshops, the model was slightly updated. The workshop setting was a small meeting room that can hold 10 people, located at the Norwegian University of Science and Technology. The room had no windows and was selected to simulate the regulations in the industries in which marine systems developers work.

### 3.3. The Methods Used and Data Analysis of the Workshops

The workshops took place on 24 August 2015, 18 February 2016 and 5 August 2016. Although the time interval of three workshops is wide, the researcher always reviewed the material generated at the previous workshop before conducting a new one. The purpose was to recall the previous discussion between the developers and the researcher. At the workshops, the researcher used the whiteboard to make a drawing to present his fieldwork at sea to the three invited maritime systems developers from the marine industry. Notepaper, pencils, and DP system, and AIS systems design process diagrams were placed on the table where the marine systems developers were seated and where we could sit and have conversations. The purpose of using diagrams was to let the systems developers follow and speak if a misunderstanding concerning a specific function in either the DP system or AIS occurred.

The three maritime systems developers voluntarily participated in the workshops [62]. All had master's degrees in systems engineering and systems design. In addition, they all had at least three years of practical experience in designing systems for maritime simulators and vessels. The three workshops lasted for about 14 h in total; the first two were about 12 h combined, and the final workshop was 2 h in length. The location did not change. During the workshops, the researcher also introduced the concepts of computational artifact, and the essence of fieldwork. There was no video recording, and no personal nor corporate information was gathered. Given that the workshops were audio recorded, an oral agreement, between the systems developers and the researcher, was made to ensure that transcripts were rephrased regarding ethical considerations. Furthermore, conversations were noted down to capture particular emotions and body movements. This, importantly, assisted the researcher in identifying which parts of the transcripts that needed particular attention in cases of sensitive and personal expressions and/or perceptions that should not be identified outside this research. Thus, the field material presented in Section 5, is an aggregation of the different material (collected during the workshop), which has been rephrased.

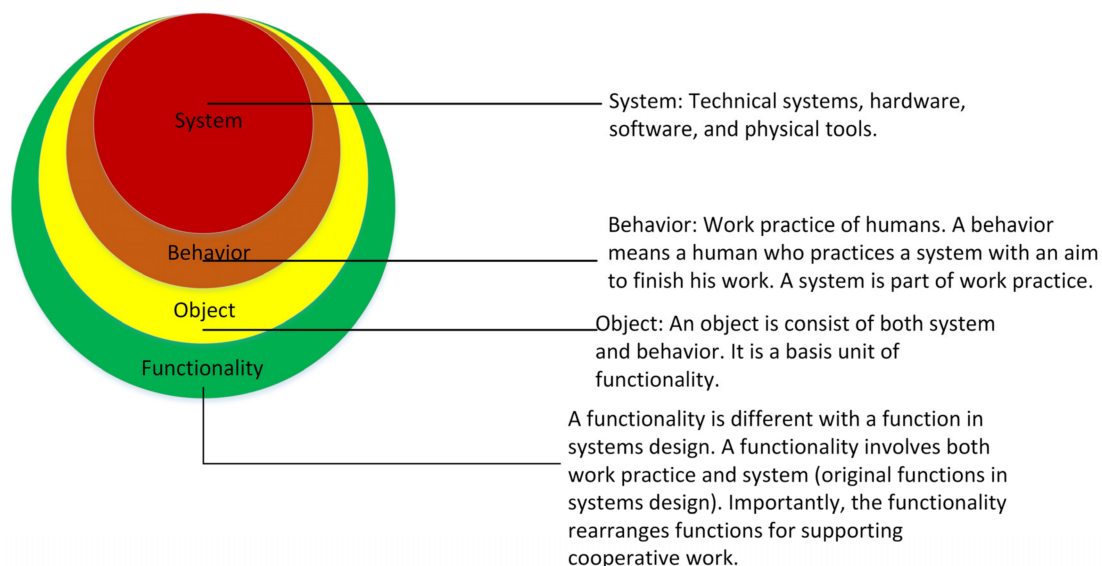
After each workshop, the researcher collected all notes, papers (i.e., diagrams), and pictures to write up the information obtained during the workshops. The researcher was particularly interested in figuring out how the diagrams, pictures, and conversations could be linked to events before and after the concept 'computational artifact' was introduced. This by comparing the systems developers' descriptions of their original work practices, in an engineering context, with the diagrams they made, as well as the conversations between them and the researchers. For example, when writing up the information, the researcher paid specific attention to how the systems developers described artifacts and how they drew diagrams after having been introduced to the model and the concept of computational artifacts. For analytical purposes, different colors were used for particular themes to re-organize the collected materials. By doing so, the researcher tried to find out how to link the technical elements, engineering work practices on artifacts, and the diagram drawn by systems developers, to the colored materials (transcribed texts) in order to find patterns. With reference to Madden [59], this workshop material, along with the material gathered at sea, was later on compared in an effort to delineate relations between and patterns of how these people do their work.

Thus, in summary, at the workshops, the researcher found it important to understand how the systems developers describe their work designing simulators before and after a new concept (like 'computational artifact') had been introduced to them. Furthermore, it was considered fruitful to look for patterns and deviations in the material in an effort to understand how and to extend the model served to convert ethnographic outcomes into a format familiar and useful to the developers. In the next section, we introduce the model in more detail, before moving into the workshops.

## 4. The Model

Schmidt and Bansler [33] state that an artifact should be understood as an object that is surrounded by work practices. We argue that system developers should take those work practices into account when they program an artifact. In this manner, an artifact must be described as the artifact itself, the work practices around it, and the relations with other artifacts (see Figure 3). An example from the

present study is an alarm watch and a calculator on the ship's bridge. These artifacts have their shapes and features. Marine operators use them in their daily life when checking the time and calculating a number. Still, when they bring these artifacts into their work context (set the alarm to keep the balance of the vessel and input a number to manage the shifting water and mud exchanging between the vessel and platform), they are ascribed additional meanings [63]. The internal mechanisms [33] of the watch and calculator do not change. However, the watch and calculator become part of the operating systems on the ship's bridge due to the work practices of marine operators. In this case, if we see the watch and calculator as objects, they become more than an artifact and its original internal mechanism. An object, in this manner, consists of the artifact, the internal mechanism of the artifact, the work practices around it, and the technical support that adds new attributes to it. In this case, the artifacts are designed to act according to the activities in which they are used, to be incorporated into sophisticated semiotic practices and to respond to and integrate unfolding events that go beyond function to address social meaning [3,33]. Hence, an artifact not only supports cooperative work. Moreover, social contexts and the work practices in them enable collaborative work [32]. Thus, here we use 'object', on similar terms as a 'computational artifact'. That is, to represent systems related elements to systems developers.



**Figure 3.** The model: From technical systems to functionalities for supporting cooperative work in marine operations.

In line with this, it is important that systems developers also consider work practices when they program simulators. However, objects in systems design are considered technical elements (tools, machines), as they are supplements that assist routine practices [64]. In this manner, objects as computational artifacts can be identified during the investigation of cooperative work [2]. For example, the researcher came to understand an artifact through the study of marine operators' cooperative work at sea. Most importantly, the idea of combining the material and non-material in relation to work practices can tell a story about the relations of artifacts as the non-material component of work practices [65–67]. Schmidt and Bansler ([33], p. 24) describe such an idea as follows:

*[I]t is we who, by manual control of tools and instruments or by the use of more or less automatic machines, do the work. Sure, the use of automatic machinery as part of our practices may have implications for these practices (educational, organisational, etc.), but they are nevertheless just that: technical compliments of our practices. It only makes sense to talk about this mechanical (or causal) regularity from the point of view and in the context of the normative regularity of our practices in which these artefacts are integral technical complements. In other words, it is we who*



*engage in normatively constituted practices, by using rulers, compasses, and by using machines, computational artefacts included.*

Schmidt and Bansler [33] argue that computational artifacts cannot be identified in terms of time and situations in which they are employed by users—in our case the systems developers. In line with this, computational artifacts become dynamic, and the factors around them, such as social meaning and work practices, are also vital when investigating new attributes of the artifacts in the very cooperation. This is an important issue for us to think about in terms of how we can inspire systems developers to take a broader view of objects. Thus, we further understand objects as the process of shaping interactive relationships regarding each operator's tasks and the connections among the various tasks, that is, the interactive relations. This understanding contributes to shifting the focus from the internal opacity [33] of an artifact to its relations of human practices in a cooperative social context. Therefore, this gave us an opportunity to use computational artifact as 'object' to enhance the translation of ethnographic outcomes, so that the project audience could shift its focus from the internal mechanism of an artifact to understanding it as an object with relations to human practices in the maritime domain. Thus, although this decomposition of a computational artifact may be conceptual, the model itself is an effort to expand the understanding of technical artifacts with the aim of helping researchers working with systems developers regarding simulator design. Moreover, such model can enable researchers to translate their ethnographic material, so that it is as close as possible to working within the engineering context.

The model is built up as follows: first, we acknowledge that "technology" refers to the use of artifacts in practice [68], and a technical system is part of cooperative work [69]. In line with this, functionality is a component of cooperative systems that includes at least one object. An object consists of both the technical system and human behavior. A cooperative system may start as a technical system, but it can become an object when systems developers incorporate cooperative work into systems design. The behavior in the model involves the interactive relations between operators, systems and tools. An 'object' is equivalent to a 'computational artifact', but here it is termed differently to distinguish CSCW research from systems design (see Figure 3). From a systems design perspective, functionality is a connection of objects consisting of interactive relations and the connections between them if we see an object as a part of a cooperative system.

The systems design process becomes a collective creation of interactive relations that connects relationships to actualize an object that allows systems developers to deploy their engineering skills. Hence, we see the model as a tool to translate ethnographic outcomes in order to work with systems developers. In this study, the inside areas are computational artifacts that involve marine operators, the workspace, and their interactive relations. Each task of maritime operators involves computational artifact(s).

According to Schmidt and Bansler ([33], p. 36),

*The key feature of computational artifacts is their capacity to react to and thus be [an] integrated part of unfolding events . . . . In short, the real challenge is that of developing machinery designed to be an integral part of normatively constituted but contingent cooperative work practices.*

For systems developers, the interactive relations inside computational artifacts are the design space inside the object(s) in the model. This translation allows CSCW researchers to convert insights from CSCW to systems design. In line with this translation process, cooperative work can be simplified as a visualized practical kit in normative systems design [70]. Systems developers can see how cooperative work is conducted with whom, which parts of the workspace are involved in the process, and how marine operators interact in the workspace. In addition, they can see how different interactions are connected to whom, with what workspace, and in what ways. Thus, systems design becomes a process of shaping cooperative systems and fulfilling functions to support that shaping work, adopting a bottom-up process that employs insights from cooperative work. As Schmidt and Bansler ([33], p. 35) note,

*In the design of computational artifacts for the purpose of serving in a regulating capacity in coordinative practices, we construct a repertoire of potential electronic circuitry that can be activated*



*and combined in multiple but determinate ways and that then, in step with the unfolding actions of the cooperating actors, are used in the coordination of interdependent activities. In other words, what is designed are mechanisms that function in a strictly causal manner but interactively.*

Schmidt and Bansler argue that we need to find a way to locate the unfolding actions of the cooperating actors and to follow such causal manner for cooperative actions to be designed strictly in computational artifacts. This discussion is quite similar to programming practices where systems developers seek to add new attributes to the technical elements. For example, the electronic circuitry needs to be combined and activated through the unfolding actions. The model presented in this paper comes to serve the unfolding actions of the cooperating actors *and* it illuminates how new attributes can be arranged into the technical elements for supporting the cooperating actors. With this, the newly designed technical elements can better support interactions among the cooperating actors. In line with this, the workshops offer constructive findings for reflecting on the importance of translating ethnographic outcomes when working with systems developers.

## 5. The Workshops

During the workshops, the maritime systems developers were asked questions about how they take cooperation into account. They thought that communication was a factor that should connect maritime operators as a group for cooperative work. In this way, communication was viewed as a channel for making operators' work visible by connecting all facilities. During the workshops, it was evident that the systems developers understood cooperative work as amassing facilities of technical tools to support marine operators' work. Moreover, they considered such work, focusing mainly on a marine operator's tasks, rather than on the cooperative work they were doing. This is understandable because, as they noted, systems developers have never been on board:

*We have never been on board, since we were provided requirement specifications from maritime engineers. Maritime engineers are those people in our companies. They get requirement specifications from domain experts, such as electrical, telecom, and so on (Workshop 1).*

Thus, based on the fieldwork at sea, an example concerning dynamic positioning of an offshore vessel in its assigned location was used to show how cooperative work is carried out. With this example, the systems developers were asked to identify objects in cooperative work. This question aimed at showing the systems developers a process of translating cooperative work from fieldwork into the model in an effort to expand their *modus operandi*. During the discussion, the systems developers saw DP systems as technical artifacts:

*DP systems are technical systems; we know how to approach them and connect them for mechanically controlling a vessel and automatically positioning it in the proper position. However, what do you mean by cooperative systems? Do you mean how each piece of [the] DP system works and connects with one another? (Workshop 1).*

Furthermore, the researcher explained how the cooperative work of marine operators unfolds when positioning a vessel. An operator should first place the vessel at an appropriate point at sea and then communicate with the oil platform to confirm that the ship's position is acceptable. Second, the marine operators should check the paper checklist to confirm what types of services are needed, such as providing fresh water or different types of mud. Then, deck crews connect pipelines that are lowered from the platform. In this step, the marine operators controlling the DP operations should coordinate with the deck crews, as wind and waves could move the pipelines and vessel away from each other. In doing so, as observed in work situations at sea, they need to check the engine status and weather reports, as well as completing the checklist for DP operations, before starting the operations. Normally, two marine operators are involved in DP operations, with an engine operator located in the engine room assisting them. The first officer is primarily responsible for DP operations, but the

chief officer also works with him or her if there are safety concerns. The researcher then explained that technical systems might be sufficient to provide mechanical control, but this dismisses marine operators' work practices:

*If you think DP systems are technical systems, then how about the work of those marine operators? What do you think checklists, forms, and engine status are? (Workshop 1).*

The developers responded,

*They are important. However, they are designed outside of DP systems. They are referred to in the requirement specifications, but we think that those actions are outside of DP operations. As you say, they have paper forms (Workshop 1).*

The researcher replied,

*If I add an unusual situation, how do you think DP systems will work? You can answer later (Workshop 1).*

The researcher displayed several photographs of the real workspace. The photos showed that, when all the DP preparation processes were ready, maritime operators on vessels were required to service fresh water and drill mud on the platform. However, during the fieldwork, an emergency occurred. The platform had a limited capacity for different types of mud drilling, and the oil company changed its need for mud without prior notice. As a result, the vessel operators could not follow their normal work practices, and they were unable to use their systems properly. When a request for changing the type of mud was sent from the platform to the vessel, the maritime operators had to go back and forth to print the forms and check all the necessary information. They also had to wait for confirmation from onshore before they could continue service to the platform. This urgent situation changed many normal work routines. In the processes of changing the forms, each operator had to hand over the work to his co-operator for a moment, which resulted in unsafe conditions, like imbalance of the vessel.

The researcher repeated the question, and the systems developers replied,

*It seems that those checklists and forms are important and probably should be inside DP systems. It would be a challenge because, in the engineering design process, each piece of the DP system is closed [the engineers pointed to the designed DP systems on the table]. If we add those forms, checklists, and so on, the whole engineering process may [be] difficult to integrate (Workshop 2).*

The researcher then referred to the model:

*That may be because you see technical systems as the basis for DP systems. Let's think differently. If I see a technical system as part of cooperative work, then I need to consider how such a technical system is used. For example, a marine operator needs to check the weather, fill in forms and check checklists, and run DP systems. If I model these work practices like this [see Figure 4], what do you think?*

*As you can see in the picture, I group each work practice as a unit that consists of a marine operator, the technical system or tool, and how they work together—their interactive relations. In CSCW, we call such a unit a "computational artifact (Workshop 2).*

The systems developers replied,

*It seems that you are shaping new DP systems according to marine operators' work practices, aren't you? Yes, if the new systems were like this diagram, we could rebuild DP, but we would like to see more examples (Workshop 2).*

Indeed, the diagram did not change the systems developers' entire *modus operandi* regarding how to design DP systems. The researcher only expanded their *modus operandi* to adopt a broader understanding of the term "technical system" and how to display it. However, it still leaves room for them to follow their engineering understanding [71]. In this diagram, DP systems became a connection of several work practices, rather than focusing on specific functions in technical systems in the early phase of systems design.

The researcher then asked the systems developers,

*Now, you know how to build up a technical system according to situated work practice. Could you please show me what you think the relations between [the] DP and AIS systems are in a similar diagram, taking into account the story I told you before? (Workshop 2).*

The maritime systems developers responded,

*Well. They were designed to be separate, since we saw that they functioned differently. Now, though, based on the situated work practices, we think it would be like this . . . According to the fieldwork, DP operations are usually associated with services operations. In marine services operations, operators position the vessel at the right place via DP systems; the operators then work with automation integrated systems. AIS is used for providing mud drilling and water to the platform and for shifting mud drilling and water under the deck to maintain balance. However, such systems do not provide information for marine operators on how much water and mud could be moved from one side of the vessel to the other . . . making it difficult to manage the balance.*

*Marine operators use an alarm clock and calculators from their daily lives and different functionalities of AIS systems, printing systems, and communications to service the oil platform. We know printing systems . . . also not a part of AIS. In addition, one marine operator may work on two systems at the same time. If not, there may be another operator helping with the operation through either verbal or digital communication. (Workshops 1 and 2).*

After this explanation, the systems developers seemed to understand how a "technical system" that incorporates insights about practices in real work situations differs from the traditional understanding of technical systems in systems engineering, as they indicated when they drew the diagram:

*Aha! That is what you mean by cooperative systems! Now, the process for designing cooperative systems is different with systems engineering. If we start with a shape of cooperative work that we can draw for you, then the relationships between two operations are evident. Individual technical systems should connect with each other in meaningful ways that are different from our traditional engineering-based understanding of technical systems. The meaning—or what you called a computational artifact—consists of a person and a piece of software or a tool and their interactional behavior. So, why not call it an object? It would be easy for us to see that an object is different with technical systems in systems engineering. This object is a function that is a part of a system, isn't it?! Those objects make systems work together to support cooperative work (Workshop 2).*

The researcher found that the model added to the systems developers' understanding of cooperative work. The model could also assist when integrating cooperative work into systems design [46]. For example, the alarm and calculator are not only artifacts, but have other meanings, which the marine operators give to them in their daily marine operations. That is, operators use them for more than just telling time or giving an alarm. They are used as functions in DP and AIS operations. In this way, we consider the model valuable for translating insights about cooperative work into visible [72], natural shapes of systems for non-CSCW professionals. This links from the technical systems to the work practices, which make the systems function in each different task. In addition,

the model links accountabilities of from fieldwork studies regarding both texts [45] and practice to the object of inquiry [32].

The systems developers followed with a question:

*If we see those diagrams as new shapes of cooperative systems, then do we make code to implement them? We think that this remains a challenge. The shapes strengthen our thinking about how to connect systems, but how can program code be put into them? The one thing we must do is to make sense of control as some part of the systems (Workshop 2).*

The researcher answered,

*When you are mapping out those diagrams, do you notice the interaction relations inside each object? Let me ask one more question: “When you show me the diagram, what do you mean [by] using communication?” (Workshop 2).*

The three systems developers said that, when they draw a diagram, the communication is a type of interaction. For them, if two operators need to cooperate, there should be some way to support this interactively. They stated,

*Interactive relations are creating “engineering functions,” to make a technical artifact function . . . to interact with operators. In systems engineering, an engineering function is interpreted as a specific process, action, or task that a system can perform (Workshop 2).*

The researcher then highlighted the phrase “a system can perform”:

*Certainly, I also mean interactive relations should be supported by some means—those means giving functionality to a technical system (Workshops 2 and 3).*

The researcher gave an example on how to illustrate interactive relations. This example was from previous piece of work where the researchers transferred cooperative work into systems functions [73]. The researcher again picked up the topic of DP operations and explained,

*The marine operator needs to fill in forms, check the weather and engine, and run DP systems, so if I visualize these activities and their relations with those systems and tools, it would be displayed as in this diagram. This chart aims to create a system that can successfully achieve its function of enabling operators or another system to interact with one another (Workshops 2 and 3).*

When the researcher demonstrated the diagram on the whiteboard, the systems developers were surprised:

*You use UML [Unified Modeling Language] in this way! We also use it! However, how do you explain the relations between two operations? For example, DP requires two people to work in reality (Workshop 3).*

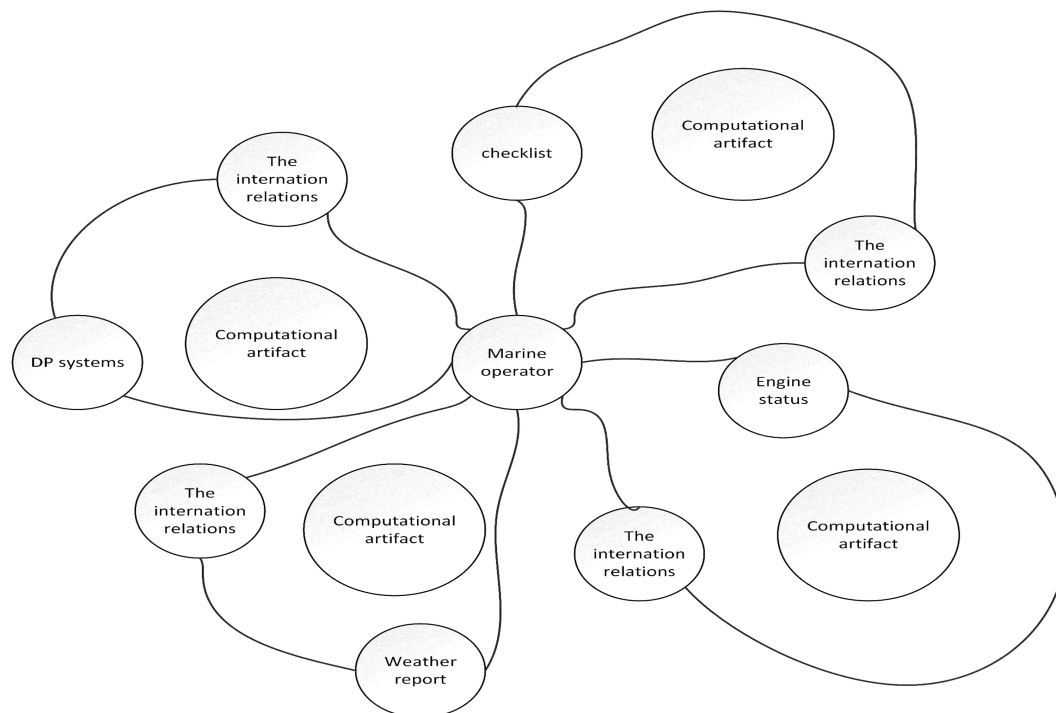
The researcher continued his explanation:

*I connected their cooperative work relationships, such as awareness of other people’s work practices [see Figure 5]. In my fieldwork, two operators were working with DP systems—the first officer and the chief officer. In these operations, the chief officer also checks the weather information [marked “W” in Figure 5]. Meanwhile, the engineer in the engine room is also aware of the weather information. I connected their behavior in a UML model that can be used in requirements engineering to link programming language to these behaviors (Workshop 3).*

The systems developers said excitedly,

*You mean, cooperative systems consist of several functionalities? Each functionality may have one or more objects. An object consists of the behavior of marine operators and software systems or technical tools. When designing cooperative systems, the core notion is to focus on the interactive relations between a marine operator and the system or tool that the operator uses rather than . . . simply the combination of software and hardware. Then, it is important to link the connections of the interactive relations, because they build up objects connected as functionalities. Interactive relations are the design space for us. With some models similar to the one you showed [UML], it is not difficult to translate into a JAVA framework for filling code. Yes, we know more models that can help with such [a] modeling process of interactive relations; UML is just one of them (Workshop 3).*

It is clear that if we see each technical function of a system (marine operating system) and subsystem (DP and AIS) as an artifact, the work practices of marine operators give meaning to that artifact, and in turn, it supports their work practices. In this case, the issue becomes how to translate and illustrate the ethnographic outcomes in a format that may be useful in the other field; the maritime domain is only one example.



**Figure 4.** “New” dynamic positioning (DP) systems.



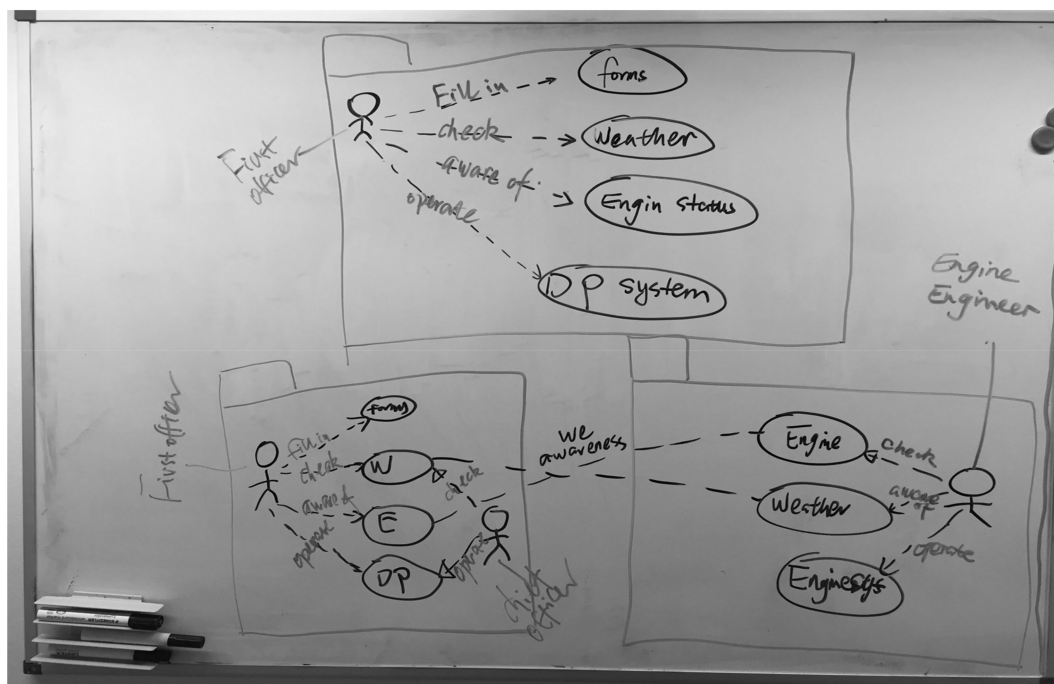


Figure 5. Displaying interactive relations with dynamic positioning (DP) as an example.

## 6. Revisiting the Marine Systems Developers after Six Months

Half a year later, the researcher revisited the three systems developers at their firms. One of the companies was building a new simulator for different marine operations. Surprisingly, the room in which the simulators had been designed now featured multiple surveillance devices, with a software system to record marine operators' cooperative work during maritime training. Furthermore, a debriefing room had been established, where interviews with operators and trainers could be conducted and from which the training process could be observed via a specific software program that could play back recorded videos frame by frame. In this way, the software system allowed the systems developers to replay sessions to compare the results with cooperative work in real work settings at sea. Most importantly, the new systems structures that they had created now followed the model to include the behavior of marine operators (Visiting and meeting at the company, 9 October 2016). The new structures were developed based on observations from experts in their companies. These experts participated in the maritime training sessions in simulators where marine operators use technical systems. After the training sessions, the systems developers highlighted interactive relations and their connections in shaped cooperative systems in their systems models. By running tests with unusual situations on the simulator with different marine operators from local shipping industries, such as the imbalance of a vessel or fire alarm in the engine room, the systems developers proved that cooperative work was now better supported by the new simulator compared with the approach taken before the introduction of situations from work at sea and explanations about concepts used within CSCW (Visiting and meeting at the company, 9 October 2016).

## 7. Discussion

Argyris [74] claims that most people concentrate on problem-solving rather than reasoning when faced with challenges in their fields. This also applies to systems design. When the systems in use face challenges related to supporting cooperative work, maritime systems developers can only imagine how to solve such problems with the current systems, even without sufficient proof that the solutions will address the safety issues in cooperative work [64]. It is understandable that approaches addressing human-related topics in systems design are rare [34]. Although researchers [9] have stated that the

binding of artifacts and locations can help investigate the meanings of computational artifacts in work practices, a “how-to” kit for systems developers is still lacking. When studying cooperative work, CSCW researchers may overlook that systems developers are also users. Smith [75] and Miles and Huberman [76] encourage researchers to “think display”—to show the complexity of qualitative data analysis so that a study’s findings may be essentialized via readable charts, matrices, and graphics rather than simply being the retelling of a story, as in the ethnographic field [12]. Thus, our model enables researchers to engage in the core of their design work to establish hands-on engineering practices [77–80].

Unlike other interdisciplinary research in systems design, which mainly focuses on why systems fail and propose mitigating methods [78], our model supports the inclusion of CSCW in systems design contexts, herein, allowing it to log ideas and make decisions in support of engineering work. Contrary to fieldwork studies [12,79,81], this model enables CSCW researchers to emphasize computational artifacts in the workspace as artifacts and the behavior of marine operators inside those artifacts. With such an effort, the model goes further and links identified computational artifacts as objects to establish functionalities and their interactive relations in cooperative systems.

The model also enables CSCW researchers to report observations and interviews and to show how insights from their work can assist others who have different expertise [72]. Furthermore, it also allows for ideas to be translated (as in our case) from one marine task to the entire picture. For example, researchers can represent the DP system as a technical system to begin with, and then bring social behavior into that special system to expand systems developers’ *modus operandi* regarding the shape of DP systems. It is important that this step-by-step process is concerned with the “interactional what” of work—with how operators’ cooperative work sheds light on the evolution of the shape of systems and the challenge of breaking down those shapes into the designable units undertaken in design work. Our work adds new elements to previous work concerning CSCW design efforts [12], offering a “how-to” kit for systems developers. This contribution informs pragmatic decisions on where to design since it is a fundamental “semantics” [80] for systems developers [30]. This semantics enables researchers to share accountable insights of cooperative work to guide systems developers in creating engineering models, with the mathematical specifications allowing them to communicate that behavior unambiguously to other engineers [30,77]. For example, researchers respect systems developers’ skills and make room for implementing and connecting each system to support cooperative work from the systems engineering perspective, as by using UML models. This approach also emphasizes how each system can be connected by following the diagram of cooperative work for each marine operation. In this way, cooperative work is laid out logically in diagrams that create a common understanding (the object) shared by CSCW researchers and systems developers. This helps systems developers focus on the design space of interactive relations and their connections to realize the implementations. In addition, it helps researchers translate their ethnographic outcome into a presentation that can be understood by others who do not have the same expertise.

In addition, this study acknowledges that cooperative work happens in a flat organization without a hierarchical model [50,77], a central need when developing the cooperative system. By taking into account the non-hierarchical structure of cooperative work, our model implements systems design with a bottom-up innovation that lets operators speak from their work positions in the systems design process [81]. Hence, the insights from this study serve as a mediator between operators and systems developers, and they contribute to systems design by visualizing the shape of cooperative systems, informing systems developers of the importance of marine operators’ work practices. The cooperative work illuminated in systems’ shapes shows that one vital capacity of the visual lies in its malleability [72]. CSCW insights, as mediators, can efficiently add “how-to” skills; it does not work from a low level of modeling interactive relations as functions for technical systems [73]. Rather, such efforts emerge from an ontological level of visualizing what is going on in the workspace and examining interactional work to bring any social events into the shapes of cooperative systems. These “how-to” skills also open spaces for systems developers to use their preferred tools, skills,

and languages to deal with interactions and fill in functions to support those shapes of cooperative systems. Moreover, the model offers demonstrable opportunities for systems developers to work with CSCW researchers.

By doing so, our focus differs from CSCW in terms of the role of materials in cooperative work. The interactive relations concerning the materials used by human actors may not be critical to researchers; however, in the context of the present study, it is important to consider how such interactive relations shape and reshape marine operations to show how to design maritime simulators. This focus, we think, is closer to the role of systems developers, as it can show them how objects could be connected to support specific work practices by borrowing programming thinking. Moreover, such thinking helps convert objects to simulate programming languages where the object involves different attributes. In this case, the translation of ethnographic outcomes is aimed at catering to the systems developers' enterprise. Arguably, some researchers in the CSCW field focus on recognizing different materials that have different qualities depending on the ways they are used in a specific place. In this case, no matter how the material is bounded through time and space in cooperative work, it is static irrespective of the execution of the coordination it prescribes [82].

However, bridging design and engineering is not new in the CSCW field. According to Schmidt ([83], p. 4),

*When engaged in cooperative effect, actors are objectively and materially interdependent. Their interdependence inescapably has causal aspects, and their actions and interactions are both intentional and material. Again, this is not sensational news. Some may refer to this duplicity as the "double character of work process" [84] or by conceiving of it as a socio-technical system [85] or "distributed cognition" [86], or as a network of actors and artefact [87] or whatever. These are merely different ways of stating the problem. The challenge is to develop the conceptual implications of this insight and understand the intricate interplay of the causal and the intentional, of the material and the culture.*

Thus, concentrating on the computational artifact is essential for supporting the interactive relations between actors. In cooperative work, such relations between human and nonhuman should be considered in the design of maritime simulators [88]. They should be evaluated as sociotechnical associations between humans and nonhumans. However, it is important to notice that if we focus on creating a platform for two different work practices (of marine operators and of systems developers) to meet, we must consider how to convince systems developers to take the work practices of marine operators into account. Therefore, we might need to provide a similar method to deliver our insights to developers. If such discussions are present, it reinforces our argument that the solution of interpreting ethnographic outcomes must get back to the engineering context.

Furthermore, this study also responds to the matter that qualitative studies, such as CSCW studies, are time-consuming [30] and may not be effective because system engineers are unable to quickly understand where to begin designing and where to stop shaping their systems [30]. The outcome from qualitative studies can amplify the voices of marine operators. In this case, systems design becomes a process of design *with* them, rather than design *for* them. In return, such cooperative systems support the cooperative work of marine operators as a whole, rather than understanding cooperative work as a collection of individual work supported by computer systems [47]. Thus, our model opens design spaces for systems developers by way of translating situations at sea into engineering diagrams, adding codes, and so on. Importantly, the translation process brings social meaning and work practices into the design process.

Systems developers can work in this space because they design the internal mechanisms of a social-technical system (i.e., artifact or object). Researchers who have gained knowledge, doing fieldwork regarding cooperative work, can help systems developers translate work practices and social meaning in their designs. As discussed in the literature review, in dealing with the complexity of cooperative work, previous studies have focused on human activities, without taking technology

development into account. The present study contributes to the literature by showing that systems developers can include both social and technical aspects in the same model. This effort contributes to the CSCW field. In our case, supporting human activities not only informs design. Importantly, we must point out how to use human activities to formulate engineering work practices. The reason is that it may be insufficient to show systems developers how simulators play an important role in humans' cooperative work. By doing so, researchers can help translate the artifact, its name, attributes, and methods in a technical system into new attributes and methods. Reassembling these new attributes and methods, the artifacts, as computational artifacts, can shape a socio-technical solution to better support cooperative work.

Schmidt [82,83] argues that it is not sensational news when actors' interdependence has causal aspects, as all actions and interactions are intentional and material. Due to CSCW's focus on specific problems, it challenges us to generate the present study into a wider area. However, an important contribution of the study is that its findings have implications for how actors can be treated objectively and materially in the design process, as well as how systems developers can contribute their professional knowledge in working with researchers. We believe that it may be interesting for other research fields to borrow when facing similar challenges.

## 8. Conclusions

We established a model that consists of a 'computational artifact' and insights from the CSCW-based fieldwork regarding cooperative work to enable researchers to work with systems developers. We align with scholars who have argued that, if field workers can effectively use ethnography with their fellows [59], then it is worthwhile to adopt a similar approach for grasping and exchanging ideas between two different elements of the design process. CSCW research is not just an analytical lens through which cooperative work can be viewed; rather, it is an approach that can bring about significant changes in the modus operandi of systems developers. In line with this, our study can be fruitful for CSCW researchers in that it opens up a possibility for focusing on visualizations of interactive work, thereby addressing relations between technical artifacts, computational artifacts and objects; between functions in technical systems and functionality in cooperative systems, and in linking design back to fieldwork studies of practice. The insights from CSCW can be used as resources for bridging technical functions to social-technical solutions in cooperative systems when cooperating with systems developers. Thus, we assert that cooperative systems can be formed by cooperative work that employs an innovative, bottom-up evolution of systems design.

### Note:

1. According to Store norske leksikon [Great Norwegian lexicon] a supply vessel is: "a ship that brings supplies and services (fuel, supply, well equipment, spare parts, etc.) to other ships and installations at sea. It is an important ship type within offshore oil operations." (Translated from Norwegian) [89].

**Acknowledgments:** The Research Council of Norway supported this study with a grant (no. 237929). We greatly appreciate the critical feedback of three reviewers from the 15th European Conference on Computer-supported Cooperative Work and three reviewers from the 21st ACM Conference on Computer-supported Cooperative Work and Social Computing. Thanks also go to the two reviewers who provided invaluable comments.

**Author Contributions:** Yushan Pan conceived and designed the field studies and workshops, analyzed the data, and wrote the paper. Sisse Finken helped improve arguments and writing them up.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Bannon, L.; Schmidt, K.; Wagner, I. Lest we forget. In *Proceedings of the 12th European Conference on Computer Supported Cooperative Work, ECSCW 2011, Aarhus, Denmark, 24–28 September 2011*; Bødker, S., Bouvin, O.N., Wulf, V., Ciolfi, L., Lutters, W., Eds.; Springer: London, UK, 2011; pp. 213–232. ISBN 978-0-85729-913-0.



2. Bannon, L.; Bødker, S. Beyond the interface: Encountering artifacts in use. In *Designing Interaction*; Cambridge University Press: New York, NY, USA, 1991; pp. 227–253.
3. Schmidt, K. The Concept of “Practice”: What’s the Point? In *Proceedings of the 11th International Conference on the Design of Cooperative Systems (COOP 2014), Nice, France, 27–30 May 2014*; Rossitto, C., Ciolfi, L., Martin, D., Conein, B., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 427–444. ISBN 978-3-319-06498-7.
4. Andersen, T.O. Prototyping a Collective: On Ethnography, Design, and Use of a Personal Health Record. Ph.D. Thesis, University of Copenhagen, København, Denmark, 23 November 2012.
5. Bentley, R.; Randall, D. *Tutorial notes, Proceedings of the Computer Supported Cooperative Work, CSCW 2004*; Chicago, IL, USA, 6–10 November 2004; ACM: Toronto, ON, Canada, 2004.
6. Randall, D.; Harper, R.; Rouncefield, M. *Fieldwork for Design: Theory and Practice*; Springer: London, UK, 2007.
7. Kashimura, K.; Hara, Y.; Ikeya, N.; Randall, D. Patterns of work: A pragmatic approach. In *Designing Socially Embedded Technologies in the Real-World*; Springer-Verlag: London, UK, 2015; pp. 35–62.
8. Wulf, V.; Müller, C.; Pipek, V.; Randall, D.; Rohde, M.; Stevens, G. Practice-Based Computing: Empirically Grounded Conceptualizations Derived from Design Case Studies. In *Designing Socially Embedded Technologies in the Real-World*; Springer: London, UK, 2015; pp. 111–150.
9. Bjørn, P.; Østerlund, C. *Sociomaterial-Design: Bounding Technologies in Practice*; Springer International Publishing: Cham, Switzerland, 2014.
10. Avram, G.; Bannon, L.; Bowers, J.; Sheehan, A.; Sullivan, D.K. Bridging, Patching and Keeping the Work Flowing: Defect Resolution in Distributed Software Development. *Comput. Support. Coop. Work* **2009**, *18*, 477. [[CrossRef](#)]
11. Randall, D. Living Inside a Smart Home: A Case Study. In *Inside the Smart Home*; Harper, R., Ed.; Springer: London, UK, 2003; ISBN 978-1-85233-854-1.
12. Button, G.; Crabtree, A.; Rouncefield, M.; Tolmie, P. *Deconstructing Ethnography: Towards a Social Methodology for Ubiquitous Computing and Interactive Systems Design*; Springer International Publishing: Cham, Switzerland, 2015.
13. Martin, D.; Sommerville, I. Patterns of cooperative interaction: Linking ethnomethodology and design. *ACM Trans. Comput.-Hum. Interact.* **2004**, *11*, 59–89. [[CrossRef](#)]
14. Procter, R.; Rouncefield, M.; Poschen, M.; Lin, Y.; Voss, A. Agile Project Management: A Case Study of a Virtual Research Environment Development Project. *Comput. Support. Coop. Work* **2011**, *20*, 197–225. [[CrossRef](#)]
15. Sá, M.; Shamma, D.A.; Churchill, E.F. Live mobile collaboration for video production: Design, guidelines, and requirements. *J. Pers. Ubiquitous Comput.* **2014**, *18*, 693–707.
16. Ambe, A.M.H.; Brereton, M.F.; Rittenbruch, M. Vendors’ Perspectives of Coordination in the Information Technology Offshore Outsourcing Industry: An Exploratory Study from the Philippines. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, San Francisco, CA, USA, 27 February–2 March 2016; pp. 319–334.
17. Desjardins, A.; Greenberg, S.; Wakkary, R.; Hambelton, J. Avalanche Beacon Parks: Skill Development and Group Coordination in a Technological Training Ground. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, San Francisco, CA, USA, 27 February–2 March 2016; pp. 872–886.
18. Garlan, D.; Clements, P.; Little, R.; Nord, R.; Stafford, J. *Documenting Software Architectures: Views and Beyond*; Addison-Wesley Professional: Boston, MA, USA, 2010; ISBN1 0321552687. ISBN2 9780321552686.
19. Hildre, H.P. *Research Arena for Ocean Space Operations (ROSO)*; Technical report; NTNU: Ålesund, Norway, 2016.
20. Hildre, H.P. *AMO—Infrastruktur—Laboratorier*; Technical report; NTNU: Ålesund, Norway, 2010.
21. Fraunhofer CML. *Maritime Unmanned Navigation through Intelligence in Networks*; Fraunhofer CML: Hamburg, Germany, 2016.
22. Kang, H.J.; Yang, Y.S.; Choi, J.; Lee, J.K.; Lee, D. Time basis ship safety assessment model for a novel ship design. *Ocean Eng.* **2013**, *59*, 179–189. [[CrossRef](#)]
23. DNV. *Marine Operations, General*; DVN: Oslo, Norway, 2011.
24. Øvergård, K.I.; Sorensen, L.J.; Hontvedt, M.; Smit, P.N.; Nazir, S. Maritime Bridge Crew Training. In *Simulators for Transportation Human Factors*; Young, M.S., Lenne, M.G., Eds.; CRC Press; Taylor & Francis: Boca Raton, FL, USA, 2017.
25. Vincentius, R. *Human Factors in Ship Design and Operation: Experiential Learning*; Norwegian University of Science and Technology: Trondheim, Norway, 2016.



26. Norges Rederiforbund; Norsk Olje & Gas; NOGEPa; Danmarks Rederiforening; Oil & Gas UK; United Kingdom Chamber of Shipping. *Guidelines for Offshore Marine Operations*; United Kingdom Chamber of Shipping: London, UK, 2013.
27. Forskningsrådet. *Innovation Programme for Maritime Activities and Offshore Operations*; Forskningsrådet: Oslo, Norway, 2012.
28. Anderson, B. Where the rubber hits the road: Notes on the development problem in workplace studies. In *Workplace Studies: Recovering Work Practice and Informing System Design*; Cambridge University Press: Cambridge, UK, 2000; pp. 215–229.
29. Medeiros, W.G. Meaningful interaction and products. *Des. Issues* **2014**, *30*, 16–28. [[CrossRef](#)]
30. Baxter, G.; Sommerville, I. Socio-technical systems: From design methods to systems engineering. *Interact. Comput.* **2011**, *23*, 4–17. [[CrossRef](#)]
31. Blomberg, J.; Karasti, H. Reflections on 25 years of ethnography in CSCW. *Comput. Coop. Work* **2013**, *22*, 373–423. [[CrossRef](#)]
32. Christensen, L.R.; Harper, R. The many faces of computational artifacts. In Proceedings of the 12th International Conference on the Design of Cooperative Systems, Trento, Italy, 23–27 May 2016; pp. 93–106.
33. Schmidt, K.; Bansler, J. Computational artifacts: Interactive and collaborative computing as an integral feature of work practice. In Proceedings of the 12th International Conference on the Design of Cooperative Systems, COOP 2016, Trento, Italy, 23–27 May 2016; pp. 21–38.
34. Lenberg, P.; Feldt, R.; Wallgren, L.G. Behavioral software engineering: A definition and systematic literature. *J. Syst. Softw.* **2015**, *107*, 15–37. [[CrossRef](#)]
35. Ralph, P.; Mohanani, R. Is Requirements Engineering Inherently Counterproductive? In Proceedings of the 5th International Workshop on the Twin Peaks of Requirements and Architecture, TwinPeaks 2015, Florence, Italy, 17 May 2015; pp. 20–23.
36. Sommerville, I.; Lock, R.; Storer, T. Information Requirements for Enterprise Systems. In *Large-Scale Complex IT Systems. Development, Operation and Management*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 266–282.
37. Suchman, L. Making work visible. *Commun. ACM* **1995**, *38*, 56–64. [[CrossRef](#)]
38. Bjørn, P.; Boulus-Rødje, N. The Multiple Intersecting Sites of Design in CSCW Research. *Comput. Support. Coop. Work* **2015**, *24*, 319–351. [[CrossRef](#)]
39. Goulden, M.; Greiffenhagen, C.; Crowcroft, J.; Mcauley, D.; Mortier, R.; Redenkovic, M.; Sathiaselalan, A. Wild interdisciplinarity: Ethnography and computer science. *Int. J. Soc. Res. Methodol.* **2017**, *20*, 137–150. [[CrossRef](#)]
40. Harper, R. The organisation in ethnography—A discussion of ethnographic fieldwork programs in CSCW. *Comput. Coop. Work* **2000**, *9*, 239–264. [[CrossRef](#)]
41. Arminen, I. Workplace studies: The practical sociology of technology in action. *ACTA Sociol.* **2001**, *44*, 183–189. [[CrossRef](#)]
42. Plowman, L.; Rogers, Y.; Ramage, M. What are workplace studies for? In Proceedings of the Fourth European Conference on Computer-Supported Cooperative Work, Stockholm, Sweden, 10–14 September 1995.
43. Denef, S.; Oppermann, R.; Keyson, D. Designing for social configurations: Pattern languages to inform the design of ubiquitous computing. *Int. J. Des.* **2011**, *5*, 49–65.
44. Erickson, T. Supporting interdisciplinary design: Towards pattern languages for workplaces. In *Workplace Studies: Recovering Work Practice and Informing System Design*; Cambridge University Press: Cambridge, UK, 2000; pp. 252–261.
45. Dourish, P. Implications for design. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Montréal, QC, Canada, 22–27 April 2006; pp. 541–550.
46. Dourish, P.; Button, G. On “Technomethodology”: Foundational Relationships between Ethnomethodology and System Design. *Hum. Comput. Interact.* **1998**, *13*, 395–432. [[CrossRef](#)]
47. Twidale, M.; Rodden, T.; Sommerville, I. The designers’ notepad: Supporting and understanding cooperative design. In Proceedings of the Third European Conference on Computer-Supported Cooperative Work, Milan, Italy, 13–17 1993; pp. 93–108.
48. Checkland, P.; Checkland, P. Systems thinking, systems practice: Includes a 30-year retrospective. *Syst. Res.* **1999**, *17*, S11–S58. [[CrossRef](#)]

49. Schmidt, K. The critical role of workplace studies in CSCW. In *Workplace Studies: Recovering Work Practice and Informing System Design*; Luff, P., Hindmarsh, J., Heath, C., Eds.; Cambridge University Press: London, UK, 2000; pp. 141–149.
50. Schmidt, K. The Concept of “Work” in CSCW. *Comput. Support. Coop. Work* **2011**, *20*, 341–401. [[CrossRef](#)]
51. Shen, W.; Barthès, J.-P.; Luo, J. Computer Supported Collaborative Design: Technologies, Systems, and Applications. In *Contemporary Issues in Systems Science and Engineering*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2015; pp. 537–573. ISBN 978-1-119-03682-1.
52. Li, W.; Ong, S.K.; Nee, A.Y.C. *Integrated and Collaborative Product Development Environment: Technologies and Implementations*; World Scientific: Singapore, 2006.
53. Coulouris, G.; Dollimore, J.; Kindberg, T.; Blair, G. *Distributed Systems: Concepts and Design*; Addison Wesley: Boston, MA, USA, 2012.
54. Lander, S.; Corkill, D.; Staley, S. Designing Integrated Engineering Environments: Blackboard-Based Integration of Design and Analysis Tools. *Concurr. Eng.* **1996**, *4*, 59–71. [[CrossRef](#)]
55. Howard, R.; Björk, B.-C. Building information modelling—Expert’s views on standardisation and industry deployment. *Adv. Eng. Inform.* **2008**, *22*, 271–280. [[CrossRef](#)]
56. Eckerson, W. Three tier client/server architecture: Achieving scalability, performance, and efficiency in client server applications. *Open Inf. Syst.* **1995**, *10*, 1–20.
57. Murphy-Hill, E.; Lee, D.Y.; Murphy, G.C.; McGrenere, J. How do users discover new tools in software development and beyond? *Comput. Coop. Work* **2015**, *24*, 389–422. [[CrossRef](#)]
58. Lurås, S.; Mainsah, H. *ACM Interactions*; ACM: New York, NY, USA, 2013; pp. 32–35.
59. Madden, R. *Being Ethnographic: A Guide to the Theory and Practice of Ethnography*; SAGE: Newcastle upon Tyne, UK, 2013.
60. Blomberg, J.; Giacomi, J.; Mosher, A.; Swenton-Wall, P. Ethnographic field methods and their relation to design. In *Participatory Design: Principles and Practices*; Lawrence Erlbaum Associates: London, UK, 1993; pp. 123–155.
61. NTNU. *NTNU Training Courses*. Available online: <https://maritime.hials.no/> (accessed on 23 January 2018).
62. DSDM Workshops. *DSDM Handbooks on the DSDM Agile Project Framework*; Agile Business Consortium Limited: Ashford, UK, 2014; Chapter 9; Available online: <https://www.agilebusiness.org/content/workshops> (accessed on 8 February 2018).
63. Pan, Y.; Komandur, S.; Finken, S. Complex Systems, Cooperative Work, and Usability. *J. Usability Stud.* **2015**, *10*, 100–112.
64. De Weck, O.; Roos, D.; Magee, C.L. *Engineering Systems: Meeting Human Needs in a Complex Technological World*; First Paper; MIT Press: Cambridge, MA, USA, 2016.
65. Ackerman, M.S.; Dachtera, J.; Pipek, V.; Wulf, V. Sharing knowledge and expertise: The CSCW view of knowledge management. *Comput. Support. Coop. Work CSCW Int. J.* **2013**, *22*, 531–573. [[CrossRef](#)]
66. Christensen, L.R. Work practice between the real and the really made up. In *Proceedings of the First International Workshop on Physicality*, Lancaster University, Lancaster, UK, 6–7 February 2006; pp. 16–20.
67. Schmidt, K.; Wagner, I. Ordering systems: Coordinative practices and artefacts in architectural design and planning. *Comput. Coop. Work* **2004**, *13*, 349–408. [[CrossRef](#)]
68. Christensen, L.R. Implications for CSCW. In *Coordinative Practices in the Building Process: An Ethnographic Perspective*; Springer: London, UK, 2013; pp. 121–132.
69. Christensen, L.R. Techno-anthropology for design. In *What Is Techno-Anthropology*; Aalborg University: Aalborg, Denmark, 2014.
70. Vandecasteele, A.; Devillers, R.; Napoli, A. From Movement Data to Objects Behavior Using Semantic Trajectory and Semantic Events. *Mar. Geod.* **2014**, *37*, 126–144. [[CrossRef](#)]
71. Pressman, R. *Software Engineering*; McGraw Hill: New York, NY, USA, 2010.
72. Henderson, K. *On Line and On Paper: Visual Representations, Visual Culture, and Computer Graphics in Design Engineering*; MIT Press: Cambridge, MA, USA, 1999; ISBN 9780585087399.

73. Pan, Y.; Finken, S. Visualising Actor Network for Cooperative Systems in Marine Technology. In *Technology and Intimacy: Choice or Coercion, Proceedings of the 12th IFIP TC 9 International Conference on Human Choice and Computers, HCC12 2016, Salford, UK, 7–9 September 2016*; Kreps, D., Fletcher, G., Griffiths, M., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 178–190. ISBN 978-3-319-44805-3.
74. Argyris, C. *Teaching Smart People How to Learn*; Harvard Business Review Press: Brighton, MA, USA, 2016.
75. Smith, A.D. *Talk to Me: Listening between Lines*; Random House: New York, NY, USA, 2000.
76. Miles, M.; Huberman, M. *Qualitative Data Analysis: An Expanded Sourcebook*, 2nd ed.; SAGE Publications, Inc.: Thousand Oaks, CA, USA, 1994; Volume 20, pp. 159–160.
77. Sommerville, I.; Rodden, T.; Sawyer, P.; Bentley, R. Sociologists can be surprisingly useful in interactive systems design. In *Proceedings of the Conference on People and Computers VII, York, UK, 15–18 September 1992*; pp. 342–354.
78. Rooksby, J. Wild in the Laboratory: A Discussion of Plans and Situated Actions. *ACM Trans. Comput. Hum. Interact.* **2013**, *20*, 1–17. [CrossRef]
79. Hughes, J.A.; Randall, D.; Shapiro, D. From ethnographic record to system design—Some experiences from the field. *Comput. Support. Coop. Work* **1992**, *1*, 123–141. [CrossRef]
80. De Souza, C.S.; de Gusmão Cerqueira, R.F.; Luiz, M.A.; de Mello Brandão, R.R.; Juliana, S.J.F. *Software Developers as Users: Semiotic Investigations in Human-Centered Software Development*; Springer International Publishing: Cham, Switzerland, 2016.
81. Bucciarelli, L.L. *Designing Engineers*; MIT Press: Cambridge, MA, USA, 1996.
82. Schmidt, K. Of maps and scripts—The status of formal constructs in cooperative work. In *Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work: The Integration Challenge, Phoenix, AZ, USA, 16–19 November 1997*; pp. 138–147.
83. Schmidt, K. Distributed collective practices: A CSCW perspective. In *Proceedings of the Conference on Distributed Collective Practices, Paris, France, 19–22 September 2000*; pp. 1–7.
84. Karl, M. *Das Kapital. Zur Kritik der Politischen Ökonomie. Erster Band*; Dietz Verlag: Berlin, Germany, 1867.
85. Emery, F.E.; Trist, E.L. The Causal Texture of Organizational Environments. *Hum. Relat.* **1965**, *18*, 21–32. [CrossRef]
86. Hutchins, E. *Cognition in the Wild*; MIT Press: Cambridge, MA, USA, 1994.
87. Law, J.; Hassard, J. *Actor Network Theory and After*; Law, J., Hassard, J., Eds.; Blackwell Publishing: Hoboken, NJ, USA, 1999.
88. Shiga, J. Translations: Artifacts from an Actor-Network Perspective. *Artifact* **2007**, *1*, 40–55. [CrossRef]
89. Store Norske Leksikon. Supplyskip. Available online: <https://snl.no/supplyskip> (accessed on 8 February 2018).



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).