# An Intelligent Operator Support System for Dynamic Positioning

Jurriaan van Diggelen[1], Hans van den Broek[1], Jan Maarten Schraagen[1] and Jasper van der Waa[1]

[1] TNO, Kampweg 5, 3769 DE Soesterberg, The Netherlands
{jurriaan.vandiggelen, hans.vandenbroek, jan_maarten.schraagen, jasper.vanderwaa}@tno.nl

**Abstract.** This paper proposes a human-centered approach to Dynamic Positioning systems which combines multiple technologies in an intelligent operator support system (IOSS). IOSS allows the operator to be roaming and do other tasks in quiet conditions. When conditions become more demanding, the IOSS calls the operator to return to his bridge position. In particular, attention is paid to human factors issues such as trust misalignment, and context-aware interfaces.

**Keywords:** Cognitive Systems Engineering · Personal Assistants · Dynamic Positioning · Predictive Analytics

## 1 Introduction

Dynamic positioning (DP) is a computer-controlled system which aims to maintain a vessel's position and heading using dedicated propellers and thrusters. DP operations form the basis of Floating Production, Storage and Offloading (FPSO) platforms and are a typical example of a highly automated control task that still requires human supervision: four operators are working in shifts 24/7 to monitor the system and resolve malfunctions in the rare case that this cannot be done automatically by the DP system. Sensor values that exceed threshold values lead to alarms, and serve as the primary means to trigger operators to solve malfunctions and abnormalities.

Because the DP operator (DPO) is not busy most of the time, relatively high personnel costs are spent on little work, and the operator could suffer from problems like drowsiness and boredom. A more self-sufficient control system capable of dealing with an increased range of conditions, would not solve this problem by itself. The DP operator would be even less occupied during his work shift, but would still be required to solve 'left over' incidents.

We believe that alarm-based DP systems cannot be advanced further to solve this impasse (which is sometimes referred to as the automation paradox [1]). Therefore, we propose a human-centered approach to DP systems which combines multiple technologies in an intelligent software agent, called IOSS (intelligent operator support system). IOSS functions as a team-partner of the DPO [2] and allows the operator to be roaming and do other tasks in quiet conditions. When conditions become more demanding, the IOSS calls the operator to return to his station position at the bridge. Ultimately, this

could save costs by deploying personnel more efficiently. Furthermore, it creates a more varied job description for DPO's than just system monitoring.

We have followed a systematic approach that integrates technological, human factors (HF), and operational perspectives (i.e. situated Cognitive Engineering (sCE) [3]) to develop the first prototype of IOSS. The four steps in the process are depicted below.

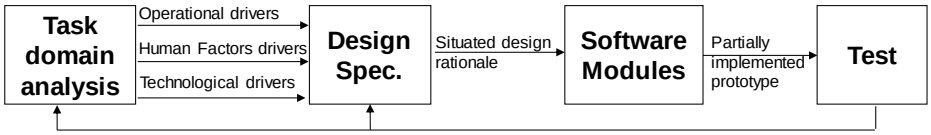


**Fig. 1.** Steps in the situated Cognitive Engineering methodology

In the first phase, we conducted a task domain analysis, and identified the most important operational, human, and technological drivers. From a technological perspective, we have identified predictive analytics as a crucial technology to enable a roaming operator. Predictive analytics can be used to predict future situations based on data from the past using machine learning algorithms. For example, to predict whether conditions are expected to remain stable, allowing the operator to leave, or to predict when alarms are likely to appear, requiring the operator to return to stationary position. From a human factors perspective, we identified a number of potential problems, related to trust, cognitive overload, and other issues well known in the HF literature [4]. For example, misalignments of operator's trust in the system could occur, because the performance of predictive analytics changes over time as more training data is used. These concerns must be adequately addressed in the design specification. In the second phase, we have specified the design specification which aims at providing a solution to the problems in the task domain. We will present these results in terms of user requirements, design patterns, and claims (which specify the rationale behind a design decision). The design specification is implemented in software modules in the third phase. We have implemented the most important patterns and user requirements in an early prototype of IOSS, which enables us to test if they bring about the expected results. To test IOSS (the fourth phase), we have set up a simulation environment, which allows us to give feedback on the earlier phases of the design process based on experiences in a semi-realistic end user environment.

The paper is structured around the four phases of the methodology. Section 2 describes the task domain analysis of the DP domain, followed by the design specification of IOSS in Section 3. Section 4 describes the software components and their interaction, followed by a description of the demonstrator which we used as a way of early testing IOSS (Section 5). The results of this test and our experiences while developing IOSS are described in the Section 6 (Conclusion).

## 2 Task Domain Analysis

As described in the introduction, the operational innovation we aim for is to allow an operator to be roaming and perform other tasks if conditions allow. This section describes relevant human factors and technological considerations when designing a system that allows these type of operations.

### 2.1 Human Factors

We have identified the following human factors issues to be relevant for IOSS.

Firstly, IOSS should address the issues of *maintaining operator's Situation Awareness* (SA). As the human operator is increasingly supported by intelligent technologies, the role of the human has evolved from direct system operator, to controller of automation, to supervisor of automation. Overall, this has had great positive effects on performance and costs, but research has shown that negative effects may arise due to lack of situation awareness [5] and out-of-the-loop problems [6]. This prevents operators from making effective decisions and causes errors [5]. These issues can lead to disastrous incidents in case automation fails. Particularly for roaming operators, we cannot take for granted that an operator's SA is at an appropriate level when (s)he returns to the task of operating DP after having been away for a while.

Secondly, IOSS must ensure the establishment of an *appropriate level of trust,* i.e. avoiding situations of overtrust and undertrust [7]. Overtrust occurs when the operator's trust in the system exceeds the system's capabilities. This situation could result in erroneous behavior as system mistakes are not corrected by the human supervisor. Undertrust occurs when operator trust falls short of the system's capabilities. This situation could lead to unnecessary operator workload, which in turn could lead to errors. As IOSS aims at establishing a higher degree of automation, the range of tasks that are performed by the system increases, and addressing trust concerns with respect to these tasks becomes even more important.

Thirdly, IOSS must establish an *appropriate level of cognitive task load (CTL),* avoiding both cognitive overload and underload. Cognitive overload occurs when the human cannot process all information that is provided by the system. An example of this in DP operations is known as alarm flooding, when the operator cannot timely respond to each alarm anymore [8], leading to suboptimal performance. Cognitive underload occurs, when the operator experiences insufficient workload, leading to drowsiness and inattentive behavior. This is a common problem for operators of highly automated DP systems, especially at night.

Fourthly, IOSS must adapt the interaction to the dynamic context of use, also known as *context aware interaction*. This issue becomes particularly relevant when we adopt the notion of a roaming operator. Because, the question of how often and in which way notifications should be sent highly depends on what the operator is doing [9].

### 2.2 Technological Drivers

We have identified a number of technological trends that will play a major role in future DP operations. Three of these technologies are outlined below.

Firstly, *predictive analytics techniques* are expected to have a major impact on the maritime world [10]. One possible application is predictive maintenance where large quantities of sensor data is collected and used as input for a machine learning algorithm. Over time, the algorithm should be able to recognize system failures before they occur, using data of the past. Such a classifier would be useful to our DP application where potential component failures are important to the operator. Many other applications of predictive analytics to DP are conceivable, for example, predicting position-loss based on weather data, or predicting operator's drowsiness based on physiological data (see [11] for an example in the automotive domain).

Secondly, *Internet of Things* [12] can be regarded as having a major impact in the maritime domain by allowing an unprecedented amount of data to be gathered and shared on a vessel. Virtually every component of a ship could become an information processing node in a large network. Applications in the DP domain could be monitoring the location of an operator, and disclosing vast amounts of additional information sources to the DP system to enable it to function more accurately.

Thirdly, computers are becoming more and more used as *personal assistants* (e.g., Siri[1], and google home[2]), which changes the relation between human and computer from that of a reactive tool to a more proactive entity (e.g. teammate [13]). As explained in the remainder of this paper, IOSS should be viewed as a personal assistant.

### 2.3 Combining Perspectives

The different drivers discussed in the previous sections are summarized in the following table.

**Table 1.** Operational, Human Factors, and Technological drivers

| Operational | Human Factors | Technological |
|---|---|---|
| Roaming operator | Maintaining SA | Predictive analytics |
| | Appropriate level of trust | Internet of Things (IoT) |
| | Appropriate level of CTL | Personal assistants |
| | Context aware interaction | |

In this early phase of design, it already becomes apparent that the technological drivers of IOSS do not straightforwardly match with human factors drivers. For example, we could expect operator mistrust in a system that is based on predictive analytics algorithms. This is because the performance of such a system changes over time and is dependent on the amount of training data it has used. We cannot take for granted that the operator is capable of making proper judgements of the prediction's trustworthiness. Another problem could be information overload of the operator, caused by the massive amount of data made available by the IoT. Also, the use of mobile devices could lead to smaller graphical interfaces that convey fewer information than the stationary displays, having a direct effect on situation awareness.

A solution to these problems is proposed in the next Section.

[1] http://www.apple.com/ios/siri/

[2] https://madeby.google.com/home/

## 3 Design Specification

Following the sCE methodology, the design specification is described from multiple perspectives. From a functional perspective, the Human-machine team functions are specified in terms of high level user requirements. From an interaction design perspective, different parts of the design solutions of IOSS are described in terms of design patterns. From an ontological perspective, the most important concepts and relations are defined that are used in the knowledge representation of IOSS [14].

### 3.1 Functional Perspective

Following the sCE methodology [15], the functional design is specified using use cases (that specify relevant environmental context, i.e. situatedness), core functions (specifying the main functionality of IOSS), and claims (specifying the reason why the function is required, i.e. design rationale).

An excerpt from envisioned core functions for IOSS is shown in **Fig. 2** :

**Adaptive Automation**
- IOSS should be adaptable w.r.t. task division and communication style
- IOSS should adapt its communication style according to user state
- IOSS should prevent cognitive overload of its user
- IOSS should behave according to a mixed initiative interaction style

**User interface**
- IOSS should support mobile and stationary UI's

**Situational Awareness**
- IOSS should support prediction of future situations
- IOSS should support change detection
- IOSS should support procedure awareness

**Trust calibration**
- IOSS should be able to explain itself
- IOSS should have a recognizable appearance

**Agent architecture**
- IOSS should be capable of acting in an open system
- IOSS should be capable of integrating information from multiple sources

**Fig. 2.** Excerpt of core functions of IOSS

The core functions are divided in five parts, each of which will be briefly discussed below. The requirements for *adaptive automation* aim to ensure a balanced workload which is tailored to the current situation of the user. This impacts the density of information that is communicated between user and IOSS, and finding a proper balance is regarded to be a responsibility of both, i.e. mixed initiative interaction. This means that the user is capable of instructing the computer when and how it wishes to be notified about which information by making *working agreements* [16]. The system also adapts its communication style to match the user's state (e.g., being brief when the operator is busy, and being more elaborate when the operator is not that busy). The *user interface* requirements state that both mobile and stationary user interfaces are needed to allow the concept of a roaming operator. The *Situation Awareness* requirements are intended

to provide the operator with a sufficient level of SA [4]. At their most fine grained level (not shown in **Fig. 2**), these requirements specify exactly which information must be communicated in which types of situations. However, as stated above, these are adaptable to the user's preferences using working agreements. The requirements regarding *trust calibration* aim to prevent distrust by ensuring that the agent is capable of explaining the outcomes of the predictive analytics algorithms (i.e., explainable AI [18]). Because IOSS is used complementary to the DP-system (and its alarm system), a different trust relation should be built up with the DP system (which produces alarms that legally require a response [17]), and IOSS which learns over time and could mistakenly produce wrong predictions. To make it clear to the operator if he is interacting with IOSS or with the DP system, the IOSS must have a recognizable appearance. The last set of requirements deals with architectural issues, such as openness of the system, and access to digital information sources.

### 3.2 Interaction Design Perspective

Whereas the functional specification describes what IOSS should be capable of, the interaction design patterns specify how this must be established [19]. For IOSS, we have specified multiple design patterns. For example, one design pattern describes a notification (called a *smart notification*) in which interactive dialogues can take place to achieve explainability. Unlike an alarm, which contains a brief text statement about a problem, a smart notification presents the message in layers that can be exposed using a dialogue. Another design pattern describes how a user can deal with multiple smart notifications, set irrelevant notifications to inactive, and relevant notification to monitor mode. A detailed discussion of design patterns for human agent teams is beyond the scope of this paper. For more information about the specification and implementation of these design patterns, the reader is referred to [19]. In the following sections, the implementation in the software prototype is discussed in more detail.

### 3.3 Ontological Perspective

To realize the adaptable interaction between IOSS and the DPO, ontologies are required that specify the shared concepts that enable communication. IOSS utilizes several ontologies at different levels.

The most basic ontology behind IOSS can be seen in **Fig. 3**. This ontology defines several basic concepts, such as what an agent and action are and how these relate to each other. In addition, this ontology specifies the concept of a policy decision. A policy is an implementation of a working agreement in the form of *"if <condition> then <create PolicyDecision>"*. A *PolicyDecision* can be about prohibiting or obligating a certain *Action* for a certain *Actor* (e.g., that IOSS must initiate a dialogue with the DPO to communicate a certain piece of information.

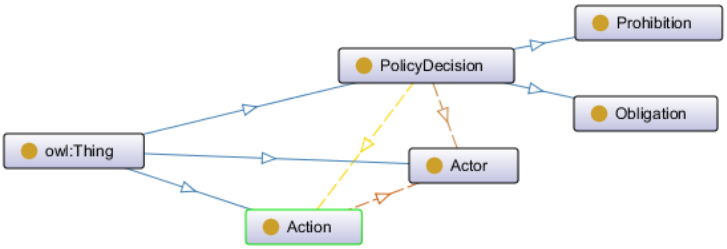


**Fig. 3.** One of the basic ontologies behind IOSS

Besides the basic ontology discussed above (which is very generic and abstract), domain specific ontologies are used to represent the knowledge required to create adequate working agreements. This removes the need for a large generic ontology of everything that is often difficult to comprehend. An example of a domain specific ontology is an ontology that defines the incoming sensor-data of the ship. With this, the user can create working agreements with IOSS that trigger on sensor values. Another ontology used by IOSS is the ontology of interactions (or *smart notifications*) that is used to describe the different ways in which the DPO can be informed.

## 4 Software implementation

To implement the system described above, we have chosen to abandon the paradigm of alarm-based control systems for DP, in favor of the agent based paradigm [20]. An agent functions as a standalone component which monitors the user and provides assistance based on the current context. Instead of sending only factual information to its user, the agent has the capability of participating in a meaningful dialogue, express judgements, provide advice, or discuss remarkable situations. The agent may even be mistaken at times, just like humans. This is why it is important that the operator is allowed to develop accurate levels of trust in the agent. To facilitate this, the agent has a recognizable appearance, a so-called avatar.

For the IOSS, we designed an agent-based system architecture that allows multiple technologies to be combined. As shown in Figure 4, we distinguish between three components within the social-technical system: the DP operator, the agent (IOSS) that supports the user, and the shared environment that contains digital information that can be accessed by both the operator and the agent.

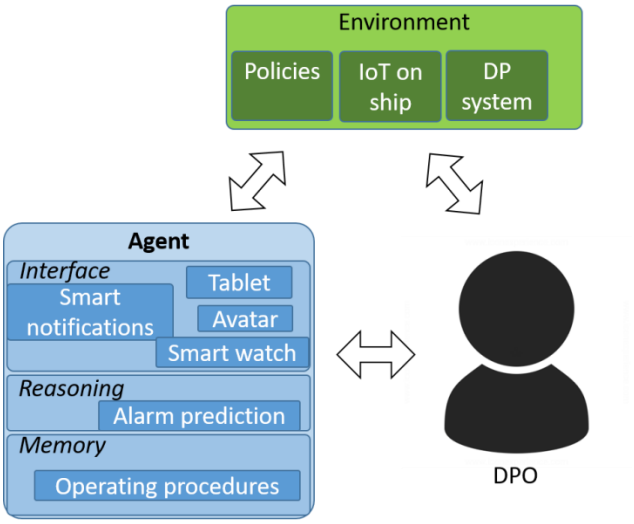


**Fig. 4.** Architectural overview of the human agent team.

The DP system remains unchanged and becomes part of the shared environment. In this way, all legacy systems are kept intact and IOSS is built on top as a layer of additional functionality. Other parts of the environment are the IoT (e.g., containing location tracking sensors) and a policy engine. As argued by Bunch et al. [21] , a policy engine can be used to specify notification rules that allow users to adapt when and how the user is notified. We follow a similar approach and specify policies in the Drools expert system language[3]. An example of a policy in our case is:

```
If wind speed is greater than 6 bft and the operator is
roaming then IOSS must suggest to the operator to come
back to stationary position
```

An important feature of our policy engine is that these rules are understandable for non-programming experts, which allows them to adapt these rules to their liking.

### 4.1 Demonstrator

With the monitoring ability of IOSS, mobile devices, proximity sensors, and notifying devices we demonstrated that the concept of a roaming operator is feasible[4].

The agent, called IOSS, is the smart component that combines machine learning, IoT, and intelligent interfaces (as discussed in the previous section). We are currently developing the machine learning components with which it is possible to predict alarms in an early stage. That is to say, to learn to detect events, circumstances and weak signals which in the past have led to problems. With early detection on the basis of weak

[3]https://www.drools.org/
[4]Human Enhancement by Maritime Adaptive Automation, TNO, https://youtu.be/MH0Vj-rChrM

signals and data analytics, we hope to achieve that the window in which a problem is detected and solved can be enlarged and that the attention of the operator is aroused and is 'drawn' into the loop.


**Fig. 5.** Screenshot of IOSS

The IoT technology is used to locate the position of the operator on board of the vessel. For example, to determine whether the operator is sitting in front of the DP workstation or is roaming. If the DPO is roaming, what is the distance to the workstation, and what is the estimated 'return time' of the operator? This is important input for the intelligent interface which must decide if the interaction should take place on the operators' mobile device (a tablet or smartwatch), or on the stationary interface. Figure 5 depicts a screenshot of the interface where the agent (shown by the avatar in the lower left corner) engages in dialogue with its user.

The next figure (Figure 6) shows the use of IOSS in the stationary condition (the left screen), where the operator uses IOSS in combination with traditional DP interfaces (the two large screens on the right).

**Fig. 6.** IOSS in stationary condition

In stable conditions, the operator can be roaming and do other things. A photo of the roaming condition is provided below (Figure 7).

**Fig. 7.** Using IOSS in mobile condition

In the roaming condition, IOSS can advise the operator to return to stationary position**,** i.e. the bridge. If the operator is not looking at his screen (because he is busy doing other things), IOSS will notify the operator using a tactile signal on the smartwatch.

## 5 Conclusion

This paper proposes an intelligent operator support system (IOSS) for dynamic positioning systems. The IOSS functions at a high level of autonomy allowing the operator to be roaming in stable conditions. IOSS is aware of its limitations, and calls back the operator to stationary position in more demanding conditions. We have developed a first design specification and prototype and dealt with a number of (conflicting) operational, technological, and human factors demands. When designing a highly autonomous system such as IOSS, the classical risk of automation paradox (i.e. that the system's disadvantages overshadow its advantages) is very relevant. This paper proposes an agent-based approach that not only considers the problem-solving technology itself (e.g. *predictive analytics*), but also considers the technology that is required to team up with humans to avoid the automation paradox (e.g. *personal assistants* technology such as *smart notifications, working agreements; IoT* technology such as *location tracking, multiple mobile devices).* This paper demonstrates the possibility to combine these components in a meaningful way as a start to develop an IOSS that acts as a true teammate of DPO's.

Because the design of these different components is highly interdependent, much work remains to be done to evaluate and refine the working of IOSS. We are currently performing tests in a controlled end user environment allowing us to refine the system. Ultimately, these tests should prove that DP operations can be performed as safe with a roaming operator and IOSS as they can using a stationary operator.

## References


1. Bainbridge, L.: Ironies of automation. Automatica 19(6), 775–779 (1983)
2. Klein, G., Woods, D. D., Bradshaw, J. M., Hoffman, R. R., & Feltovich, P. J. (2004). Ten challenges for making automation a" team player" in joint human-agent activity. IEEE Intelligent Systems, 19(6), 91-95.
3. Neerincx, M. A., & Lindenberg, J. (2008). Situated cognitive engineering for complex task environments. In J.M. Schraagen, L.G. Militello, T. Ormerod, & R. Lipshitz (Eds.), Naturalistic Decision Making and Macrocognition (pp. 373-389). Aldershot: Ashgate Publishing Limited.
4. Endsley, M. R. (2016). From Here to Autonomy: Lessons Learned From Human–Automation Research. Human Factors, 0018720816681350.
5. Sarter, N. B., & Woods, D. D. (1995). How in the world did we ever get into that mode? Mode error and awareness in supervisory control. Human factors, 37(1), 5-19.
6. Kaber, D. B., & Endsley, M. R. (1997). Out-of-the-loop performance problems and the use of intermediate levels of automation for improved control system functioning and safety. Process Safety Progress, 16(3), 126-131.
7. Lee, J. D., & See, K. A. (2004). Trust in automation: Designing for appropriate reliance. Human Factors: The Journal of the Human Factors and Ergonomics Society, 46(1), 50-80.
8. Bullemer, P. T., Tolsma, M., Reising, D. V. C., & Laberge, J. C. (2011). Towards improving operator alarm flood responses: Alternative alarm presentation techniques. Abnormal Situation Management Consortium.
9. van Diggelen, J., Grootjen, M., Ubink, E. M., van Zomeren, M., & Smets, N. J. (2013). Content-based design and implementation of ambient intelligence applications. In Ambient Intelligence-Software and Applications (pp. 1-8). Springer International Publishing.
10. Lee, H. G. (2013). A study on predictive analytics application to ship machinery maintenance (Doctoral dissertation, Monterey California. Naval Postgraduate School).
11. Singh, H., Bhatia, J. S., & Kaur, J. (2011, January). Eye tracking based driver fatigue monitoring and warning system. In Power Electronics (IICPE), 2010 India International Conference on (pp. 1-6). IEEE.
12. Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey. Computer networks, 54(15), 2787-2805.
13. Bradshaw, J. M., Feltovich, P., Johnson, M., Breedy, M., Bunch, L., Eskridge, T., ... & van Diggelen, J. (2009, July). From tools to teammates: Joint activity in human-agent-robot teams. In International Conference on Human Centered Design (pp. 935-944). Springer Berlin Heidelberg.
14. Guarino, N. (1995). Formal ontology, conceptual analysis and knowledge representation. International journal of human-computer studies, 43(5-6), 625-640.
15. van Diggelen, J., van Drimmelen, K., Heuvelink, A., Kerbusch, P. J., Neerincx, M. A., van Trijp, S., ... & van der Vecht, B. (2012). Mutual empowerment in mobile soldier support. Journal of Battlefield Technology, 15(1), 11.
16. Arciszewski, H. F., De Greef, T. E., & Van Delft, J. H. (2009). Adaptive automation in a naval combat management system. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 39(6), 1188-1199.
17. ANSI/ISA–18.2–2009, Management of Alarm Systems for the Process Industries. http://ipi.ir/standard/STANDS/ISA/18.2.pdf, 2009

18. Van Lent, M., Fisher, W., & Mancuso, M. (2004, July). An explainable artificial intelligence system for small-unit tactical behavior. In Proceedings of the National Conference on Artificial Intelligence (pp. 900-907). Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press.
19. Neerincx, M. A., van Diggelen, J., & van Breda, L. (2016, July). Interaction Design Patterns for Adaptive Human-Agent-Robot Teamwork in High-Risk Domains. In International Conference on Engineering Psychology and Cognitive Ergonomics (pp. 211-220). Springer International Publishing.
20. Wooldridge, M., & Jennings, N. R. (1995). Intelligent agents: Theory and practice. The knowledge engineering review, 10(02), 115-152.
21. Bunch, L., Breedy, M., Bradshaw, J. M., Carvalho, M., Danks, D., & Suri, N. (2005, March). Flexible automated monitoring and notification for complex processes using KARMEN. In Proceedings. 2005 IEEE Networking, Sensing and Control, 2005. (pp. 443-448). IEEE.