Anne Koskinen, Navy Command Finland
Jorma Jormakka, National Defence University, Helsinki, Finland

USABILITY MONITORING – EXTENDING QUALITY OF SERVICE MONITORING FOR DECISION MAKING

Abstract:

The paper presents a new concept, Usability Monitoring, and applies it for situational awareness applications in military Command, Control, Communication, Computers, Intelligence, Surveillance and Reconnaissance systems (C4ISR). Usability Monitoring means taking measurements of technical Quality of Service (QoS) parameters in end systems and comparing them to target values of reference cases. The concept differs from QoS monitoring in the goal and in the placement of measurement points: In QoS monitoring the goal is to verify that the network provides the promised service quality in the user system – network interface. QoS monitoring does not capture the actual end user experience, which is influenced also by the end system, and therefore it does not directly correspond to the service quality that a user sees. Usability Monitoring has exactly this goal. The Observe-Orient-Decide-Act-Loop (OODA) is a decision making concept that is widely used in the network-centric approach and it emphasizes fast decision making. The presented model for Usability Monitoring is based on the OODA-loop. It includes QoS measurements not only in the Act-phase, i.e., can the user perform the actions s/he wants or are there delays and losses that make the system less usable, but also in the Observe-phase, i.e., does the user get the information s/he subscribes to, and in the Orient-phase, i.e., does the user get confusing information and cannot orient, and also the Decide-phase, i.e. is the user able to make right decisions based on the previous steps. The measured technical QoS parameters are connected to user experience by Mean of Score (MOS) functions that are obtained by usability tests.

DOI 10.1515/jms-2016-0173

A case study for partially evaluating the Usability Monitoring concept is from MNE5 MSA (Multinational Experimentation 5, Maritime Situational Awareness) Experimentation Event 3 that was conducted in partnership with the Navy Command Finland, Naval Warfare Centre of Sweden and NATO Allied Command Transformation (ACT) and the Singaporean Armed Forces (SAF) Future Systems Directorate. In the MNE5 MSA experimentation we were able to monitor end user experience, how the user sees the services and is able to work with the current tools and capabilities. In this article we describe how Usability Monitoring was addressed in the MNE5 MSA case study: meters for Usability Monitoring were selected and we investigated what aspects of usability affect the phases of the OODA-loop.

Keywords:

Usability, Monitoring, QoS, OODA, Situational Awareness, Decision Making

1. Introduction

This article presents a new concept of Usability Monitoring and some results of its application in a case study. Usability Monitoring merges ideas from QoS monitoring, usability research and the OODA-loopQuality-of-service (QoS) as defined in ITU-T Recommendation E.800 is a general term for all parameters that are visible to a user of a networked system [14]. There is a solid engineering approach connected with technical QoS parameters. It involves defining QoS parameters and reference connections where target values for QoS parameters are given for normal and high traffic load. The target values for QoS are used as constraints in network dimensioning. They are also included in Service Level Agreements (SLA). QoS parameters are monitored for network management purposes and for checking the agreed QoS in SLAs. Well dimensioned networks usually satisfy the QoS target values and QoS monitoring notices problems, such as faults or configuration errors. Many IP networks are not dimensioned and thus QoS monitoring is sometimes seen as a way to manage performance through feedback control. In the military environment QoS monitoring is seen as a way to detect problems in exceptional conditions, e.g. the Finnish Defence Forces use QoS monitoring in the IP core network for this purpose

Usability is presently treated as a factor that should be considered in the design phase, and usability research focuses on user trials. Then, it does not address changes in usability caused by network problems, different hardware/software configurations, or actions of the adversary. The QoS approach can potentially fill this gap. User experience is connected to QoS parameters by Mean of Score (MOS) measurements. MOS (Mean of Score) is a subjective measure given by test persons, typically on the scale 1-5, for perceived quality. In Usability Monitoring all qualitative tests and measures of usability correspond to MOS measurements, thus the QoS approach does not replace usability research. The methods of usability research (see [12] for a list of typical methods) remain as the ways to evaluate user experience, find improvements to usability, and to obtain the MOS functions. The gain from the QoS approach is that it enables monitoring of usability and mathematical evaluation and optimization of some aspects of usability through technical QoS parameters when the system is operational. There seems to be much potential in extending this approach, especially since usability of networked systems is often relatively poor and something should be done to improve it. It is stated in [1] that many current network architectures address QoS, focusing to the providers perspective to analyze the network performance without taking into account the quality needs from the user's perspective. This paper proposes a wider application of the QoS approach in usability research. It measures usability experienced by the end user indirectly by looking at technical parameters that are influencing usability. In this article we develop a more general model of Usability

Monitoring and apply it to a more restricted case of a C4ISR system, where a set of technical parameters can be selected, though they do not necessarily measure all aspects of usability.

The Usability Monitoring concept has been partially evaluated in a case study in the Multinational Experimentation 5 MSA (Maritime Situational Awareness) track. MNE5 MSA Experimentation Events were conducted in partnership with Navy Command Finland, Naval Warfare Centre of Sweden and NATO Allied Command Transformation (ACT) and the Singaporean Armed Forces (SAF) Future Systems Directorate. The main goal for MNE5 MSA Experimentation was to study how MOC (Maritime Operation Centre) teams are able to achieve and maintain Maritime Situational Awareness in two separate Events that included two scenarios with the used technology. From the Experimentation we were able to get input for the concept development of the Usability Monitoring by creating meters that can measure OODA-loop steps based on QoS ideology. Service is understood in this article as a subjective definition of the end user; how the end user feels the system supports the work done by the user.

In this article we focus on presenting the new concept of Usability Monitoring. Based on the data from MNE5 MSA Event 3 we were able to confirm basic principles of four different Meter levels corresponding to the OODA-loop. This new concept will support system developers in identifying issues that affect the end user's performance and decision making throughout the decision making loop. With the combination of quantitative and qualitative methods, we are able to support the information systems development especially in the design and implementation phase.

2. The QoS approach in networking

The QoS approach is originally developed for telephony, but has been extended to all network services standardized by ITU-T. It is much less understood and used in the TCP/IP protocols. The main idea is that there is a connection between user experience and some measurable parameters, called the QoS parameters. While QoS according to ITU-T Recommendation E.800 includes a large selection of user visible parameters, including the goodness of customer service of the operator (*operator* here refers to the organization operating a network, such as a telephone company, not to the person who is called the operator of the C4ISR application), most of the QoS approach focuses on technical *QoS parameters*. [14] In the early times the most important factor was the bit error ratio (BER) but in TCP/IP protocols these QoS parameters are measured on the IP level and usually include the end-to-end delay of IP packets, the delay variation (jitter) of IP packets in a stream, and the packet loss ratio. *Throughput* is traditionally a network performance (NP) parameter, i.e., a traffic parameter that is not visible to the user. This is because in the telephony service, the user always got the same bit rate and throughput was only relevant for trunks. In the TCP/IP world, as in ATM, throughput can vary and the parameter is visible to the end user. Therefore throughput of a connection is a QoS parameter in IP and ATM networks. There is another set of QoS parameters related to availability, such as *Mean-Time-Between-Failures* (MTBF) and *Mean-Time-To-Repair* (MTTR) [24].

The main idea is that the technical QoS parameters are measurable and they can be connected to the user experience by user tests. The user tests give the MOS tables and the goal is to set target values to the QoS parameters in such a way that a sufficiently high MOS is reached. This idea is clearest in voice quality. If the MOS value is at least 4.5/5, users experience the voice as very good. The idea is the same in all of these cases: we can select some target values to the QoS parameters that guarantee good user experience. Obviously, technical QoS parameters depend on the connection and on the traffic. If the connection is for instance routed through very many nodes, connection establishment delay is bound to be longer. Therefore the target values for QoS parameters are given for a set of reference connections. Without defining the reference connections it is not possible to require that the QoS values are measured in the same way and the target values for QoS are reached. In this ITU-T approach it is essential that good QoS is obtained by agreeing that the target values that are declared in the recommendations are reached by all operators. The technical QoS parameters are measured and monitored by the network management by the measurements, usually done by the Operations and Maintenance (OAM). In the TCP/IP world, QoS measurements are not readily available and may need to be implemented, as in e.g. [15]. Some of the target values for the QoS parameters can be agreed in international recommendations. Service Level Agreements between operators and between operators and customers specify the target values also for non-standardized parameters, such as the throughput. The ITU-T approach is an engineering practice, which has never been clearly described in any document. ITU-T Recommendations by Study Group 2 show applications of the approach [14].

The connection between the QoS approach and usability is that the relevant QoS parameters must be chosen and target values for the QoS parameters must be set. Setting the parameters is best done by usability trials. When the target values for the QoS parameters are available, the problem of reaching good user experience reduces to a technical problem of fulfilling the target values. In the ITU-T approach this problem is basically solved by dimensioning, i.e, the network is built to give good QoS. Thus, the reference model is as in Figure 1.

There are a number of problems with QoS monitoring as in Figure 1. One is that traffic measurements from different points should be correlated. Such correlation, if done correctly, requires moving large data files containing packet headers and time stamps. It is too difficult to match requests and responses of user traffic by observing network traffic and therefore test traffic is often used. Test traffic consumes some capacity, and especially if there is congestion QoS of test traffic may be different from QoS that the user sees for various reasons. For all these reasons QoS monitoring as in Figure 1 is usually not a continuous activity but done periodically to check the QoS level that is promised in the SLA.

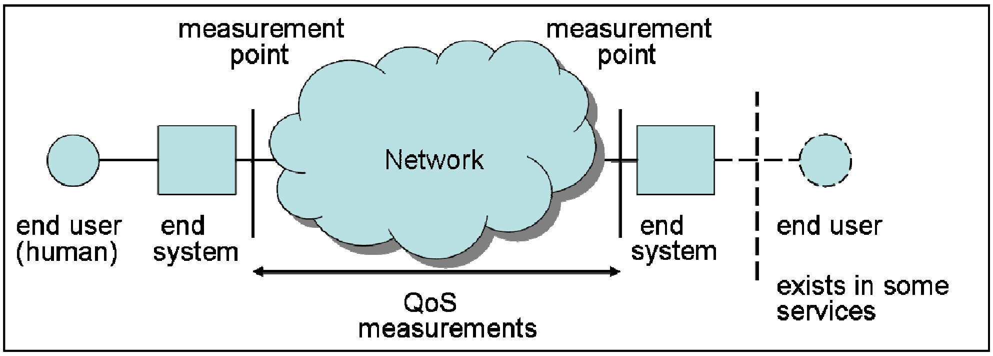

Figure 1. Reference model for QoS monitoring. The QoS parameters are traffic parameters (losses, delays, jitter, errors) or availability parameters (e.g. Mean-Time-To-Error).

The goal of this paper is to widen the area of applications. In order to do this we must find a better reference model. In the QoS approach, the focus is on user visible problems caused by the network. The user visible problems in Figure 1 derive from losses, delays, and errors in the network, and lack of availability, delays and errors in the peer end system. If we want to enhance the concept to Usability Monitoring, we must consider also the end user as a part of the system. The end user can get confused, make errors, or not notice something. To a large extent we can find an analog between user errors and network/end system errors, and between user not noticing something and network losses. There is also similarity between the user being confused because of too many inputs and with the network being congested because of too high traffic. The case study supports these intuitive ideas: We found an analog between social and technical network problems that can be seen on a higher level. We also discovered different types of errors and problems in MNE5 MSA Event 3 while observing different MOC operators interacting with each other.

3. A reference model for Usability Monitoring

Let us start from the basic concept that a user is using the system in order to achieve some goal. Reaching the goal gives a reward. What can happen is that the goal is not attained and the reward is not obtained. Another thing that can happen is that reaching the goal takes too much time. Thus, the delay is important. If the delay is too long, the effort is typically abandoned. If the delay is short enough, it does not bother the user. Between these two values is some gray area where waiting for the task to finish causes some degree of irritation in the user. A third aspect is the effort needed for reaching the goal. The effort may be counted in some applications e.g. by the number of clicks, opened windows or menus etc. If the effort is too large it causes irritation. An application is, or at least should be, tested by a rather extensive test set before it is taken to wider use. Therefore, we have a set of test cases that cover much of typical usage of the system and can take these as the reference cases. We can enhance the test set by assigning the effort and reward to a reference case. So far our model is very simple:

The user has a goal – The user performs some tasks – The user gets a reward

The usability problems are:
Failure to reach the reward
Delay in reaching the reward
Effort in reaching the reward
Difficulty in understanding the situation
Difficulty in deciding what to do next

We have a set of reference cases where the effort and the reward have been given.

This simple model does not describe all relevant aspects of the user's experience and it does not give measurable parameters. One approach is to enhance the model with some existing model describing observation and to try to get to measurable parameters. Every model emphasizes different aspects and none of them fully describe the reality. A model must be sufficiently simple, but we can add some aspects of some other model to our initial model without complicating it too much and gaining better insight to the problem. We will enhance the simple model by the Observe-Orient-Decide-Act (OODA) loop of John Boyd. It should be understood, as is pointed out by [10], that the OODA-loop is not the only model and several arguments have been made against it from a cognitive point of view. Nevertheless the OODA-loop includes the actions of the adversary to the system in a natural way. A main goal in network-centric warfare is to get inside the OODA-loop of the adversary, and the adversary tries to mix up or to slow down decisions. Most decisions are made through a networked computer system and in this sense usability of such systems is of crucial importance. We want to know if the system supports fast decision making and if the adversary can influence the behavior of the system for confusing the decision making process. In the civilian sector, usability of an information system is mainly important for customer satisfaction, and there is no adversary who tries to disturb the system. In military C4ISR systems, the important tasks are decision making, poor usability results in poor decisions, and there is an active adversary who tries to gain on poor decisions. User satisfaction in usability of the system is still a secondary goal: the main goal is that the tasks can be done well: the system does not slow down decisions, cause mistakes, or make decisions harder to take.

The main concepts of the OODA model must be given a meaning in Usability Monitoring: *Observation* for the user of a networked information system is what user sees on the screen. *Orientation* means understanding the information that user sees. *Decision* means deciding what to do next. *Action* is the set of responses the user makes, but here we restrict actions only to the new input the user gives to the information system.

Focusing on the OODA-loop illuminates several drawbacks both of the simple goal-reward model and of QoS monitoring as in Figure 1, and it stresses the importance of time: the key to success is fast decision making. We mention two main observations. Firstly, in Figure 1 we must monitor traffic in several places and correlate the measurements. This difficulty is the consequence of trying to solve the wrong problem. QoS monitoring in Figure 1 is for verifying that the cause of the user's problems is not the operator's network. The correct problem is to monitor if the user has any difficulties in his decision making process. Then, we should compare user's experiences to his expectations. This can be done if we have a set of reference cases for actions that the user can take and we compare the real message exchanges at the user end with the message exchanges in the reference cases. The next observation by focusing on to the OODA-loop is that the simple model of goal and reward is actually a model only for the *Act*-phase. If the system is slow, *Observe* and *Act*-phases are slow. The main problem is that the user cannot perform well in the decision phase but e.g. abandons the system. If the system has errors, the *Act*-phase does not result in a predictable outcome and the decision maker does not have control. The adversary can try to deny the actions. Clearly, we should monitor delays, losses and errors, but such measurements only cover the *Act*-phase. Problems in the *Observe*-phase are that the user does not get information or it is corrupted. We should also monitor delays, losses and errors of this data. In order to do it in the user end system, we must know when the information should be coming. In the *Orient*-phase the problems are that the user cannot understand the situation. Finally, there is the *Decide*-phase. Some decision systems support decision making by calculating different scenarios that may result from a choice of actions.

In the general case, it is impossible to know in advance what information is coming to a user, but in Service-Oriented Architecture (SOA) it is in many cases possible. In a SOA based C4ISR messages often follow the publish/subscribe Message Exchange Protocol (see e.g. [7]) where the end user subscribes to periodic updates of data. The Service Level Agreement for the SOA service gives the promised update period and we can monitor that the updates arrive. The adversary can try to deny observation by influencing the network but QoS measurements can detect these efforts. Integrity can be guaranteed by cryptographic means, thus corruption of data by the adversary can also be detected. SOA based C4ISR applications are of current interest in many countries [3], [16], [19], [21], [22], thus this advantage of SOA can be used in the future.

4. Desirable characteristics of a C4ISR application for situational awareness

The nature of a particular system is an important aspect in selection of QoS parameters for Usability Monitoring because software systems have different purposes and the users have different abilities and goals. The meaning of good usability is different if we talk about a difficult computer game or of a bank automaton. C4ISR systems are networked applications that have a particular set of desirable characteristics. We should take these characteristics into account when considering usability. Some conclusions of what is important for usability can be drawn from these characteristics. Let us select the following set of characteristics for a C4ISR application for situational awareness: 1) The system is transparent: the user does not spend effort in the system but can focus on the task. 2) The system demands a task to be done correctly even if it reduces usability. 3) The system helps the user to understand the situation correctly.

The system is transparent - Many users want the system to be totally transparent and let them achieve their goal as easily and fast as possible. However, it depends on the particular system if the system should be totally transparent or if a part of a good user experience of the system is that it is suitably challenging and the user experiences good command of the system as a reward, like often is with a single person computer game or an operating system. We assume that military C4ISR systems should be as transparent as possible because the primary task is too important to take any risks of failure. This is reasonable and may be true, but one should keep in mind that a fully transparent system is not always the system that gives the best user experience.

The system demands a task to be done correctly – It is not necessarily the same thing if the user finds the system usable and if the task is done well. For instance, handling classified information is clumsy and time demanding but it must be so if the task is done correctly. This is not quite the same as functionality versus usability as in [11]. We want a system that does not allow a task to be done incorrectly, even if it is clumsy. This situation often appears with security.

User understands the situation correctly - Situational awareness is a central concept in all network-centric approaches, also in the Finnish Network Enabled Defence (NED). Situational awareness has three levels: seeing the situation, understanding the situation, and being able to predict the development of the situation; the last level being very difficult to reach. A system should try to assist the second level: understanding the situation.

It follows from *transparency* that all effort in doing tasks is only a nuisance to the user. We can assign a positive value to the reward and assign a negative value to the effort to reach the reward. The effort may be number of clicks, opened windows etc. The numerical value of the effort can be evaluated by usability tests. From the characteristics of *demanding correct operations* follows that users sometimes must follow certain procedures. Therefore we may assume that users also in other tasks easily accept that they have to follow certain procedures. This means that the user interface of the system should not offer many ways of doing the same thing, which reduces the possible cases to be measured. Since a main goal of the system is that the *user understands the situation correctly* we should monitor understanding in some way. Understanding a situation is not a directly measurable parameter but we can assume that if the number of events that are visible to the user in a given time increases too much, then the user may find it harder to understand correctly. Thus, we take a measure of events shown to the user as an indirect measure of understanding. The end system often can be configured to take logs of events and therefore the measure is easy to implement. We need user tests in order to connect the measure to user understanding.

As a conclusion, the special characteristics of C4ISR applications for situational awareness are quite suitable for the presented model. These systems do not try to present as many choices to the user as possible for better usability but the users are accustomed to following fixed procedures. Therefore the set of reference cases that have to be monitored is rather small. We can keep track of the parameters for effort and reward for a representative set of reference cases, possibly for all. It is also easier to match the responses of the system to each request of the user when there is a small set of reference cases. In Usability Monitoring we can compare the logs of the events and identify the message chain as one of the reference cases. Then we can evaluate the effort and the reward of this message chain to the use by comparing delays, additional messages etc. to the target values of the reference case.

The users do not like additional effort and we do not have to consider any deeper cognitive aspects of using the system than only to look at the effort the user must exert in order to reach the reward. The user effort is derived from the delays that he experiences and how many events he must generate. The reward is seen from the way the message chain completes. If it does not complete in a similar way as in the reference case then the user does not get a reward. A simple example is that the user sends a message but it is answered with an error message. We can detect this case. Another example is that the user gets no reply. We can also detect this case. These examples show that we can form some measure of effort and reward to the message chains.

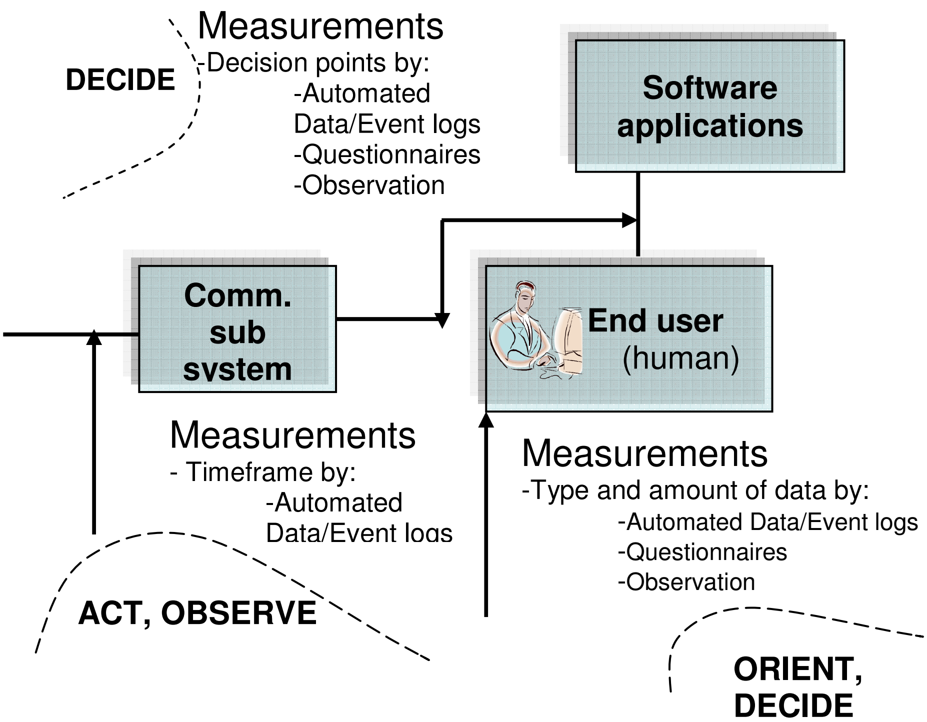

Figure 2. Local usability measurements missing from the QoS monitoring of Figure 1

In Figure 2 we present local usability measurements enhancing the QoS measurement of Figure 1. It is crucial to recognize that the OODA-loop and measuring the decision points occur behind the end system. This is why the new aspect for Usability Monitoring is needed.

We recognize that the case study and the environment was much more complex and human factors affect the results, but still we are able to point out factors that affect the performance and by that the QoS. Those factors can be generalized and transformed into the new concept of Usability Monitoring. As stated, there have been attempts to measure SA but not adequate techniques to tackle the C4 environment. It has been recognized that in order to measure SA we need a technique that measures SA only, it has the required level of sensitivity so that it detects possible changes is SA, and that it does not change SA during the process of the measurement [23]. In order to fully understand the requirements of the monitoring, we need to acknowledge that from the systems level, SA enables decisions to be made in real time. When focusing on, for example, the maritime environment, these types of socio-technical systems need to be orientated towards the dynamics of the environment [25]. It is important that we provide the operators with tools that support them building and maintaining SA [6], but this does not mean only technological innovations. We are not just focusing on technical details but widening the scope of monitoring social aspects and issues that need to be tackled in order to gain situational awareness in evolving environments with their own challenges.

5. Usability Monitoring in the MNE5 MSA experiment

There are several causes to usability problems: System design problems (e.g. software is poorly designed), Hardware and/or software configuration causing permanent problems (e.g.

too slow machine), Transient problems (e.g. errors or delays because of network load, software updates), and Intentional errors/adversary action. In the Usability Monitoring approach we assume that system is well designed in the opinion of those who introduced it but it does not work well in the opinion of those who use it. Thus, the usability problems are caused by poor configuration, network problems, or by adversary action, and they can be found by Usability Monitoring. Usability Monitoring does not measure usability. The quantitative methods used in the Usability Monitoring notice signs that indicate poor usability. The qualitative methods provide a tool for understanding the reason behind the poor usability. In the case study [2], [17], conducted in MNE5 MSA, we focused on the system design and implementation phases. We identified meters for Usability Monitoring and performed qualitative usability tests that can be used for defining MOS functions. We did not continue to setting reference connections and target values, that is, the case study does not verify the whole Usability Monitoring approach. The scope of Usability Monitoring in the MNE5 MSA Experiment was to identify characteristics, which are important when we are monitoring the system and to see if it is possible to implement measurements to every level of the OODA-loop.

The general goal in the MNE5 MSA Experimentation was to help MOC (Maritime Operation Centre) teams to detect, determine, recognize and identify possible suspicious behaviour in the maritime environment. The main idea was information sharing between the MOC teams to prevent behaviour harmful to the security, wealth and economic stability of all the involved partners. During the Experimentation planning Standard Operating Procedures (SOP), recommendations and guidelines were created for carrying out maritime operations mainly from the information sharing perspective. Technological development, scenario design and concept development were also required. Each participating nation used its own sea surveillance systems in an unclassified environment. The operators were also provided with other tools for information sharing, such as chat, email and voice. The MOC structure included at least one Intel Officer and one Operator. Each MOC had their own AOR's (Area Of Responsibility). Before the Experimentation the MOC teams were given technical training of their systems and also SOP and problem solving guidance including information about other MOC team's technological capabilities. The storyline for the scenarios was to effectively identify the COIs (Contacts of Interest). The scenarios were created so that no MOC team alone could solve the given tasks without receiving information from other participating MOC teams [2]. In Event 3 the scenarios were 7 hour long and were followed by data collection through surveys and interviews based on observations during the scenario run. Quantitative methods used were the NASA Task Load Index (TLX) and the Social Technical Organizational Rating Scale (STORS) to capture operators' subjective views. The Analyst Assessment Report Performance Rating Questionnaire (AAR PRQ) was used by the analysts as a subjective measure of MOC performance and workload. Amplifying information from surveys was reflected to information from interviews and observation. Interaction diagrams were used to visualize how the interaction occurred within and between MOC teams. From this research we got confirmation about the necessity to expand the Usability Monitoring and to develop a new concept to meet the demands for adequate response to usability problems. Information sharing was studied from social, technological and organizational perspectives. With this framework we were able to point out issues such as different channels (formal/informal) for communication, types of information, and usage of tools. We were able to identify issues that affect operators' way of using technical systems, why and how they used certain social networks and to follow the information flow.

Based on the MNE5 MSA Event 3, we found it possible to identify situations when it is crucial to look at the timeframe and focus on issues concerning QoS. When looking at the OODA-loop, in the *observe* stage we are able to see usage of different resources; own technical system, open databases or email, chat or voice to contact other MOC teams. At this stage we can analyze the type of data received and also the channels used. In the *orient* stage we are scoping the task and from the analyzing perspective focusing on the amount of data in the given time frame. The *decide* phase includes decision points when the team or user decides to act based on the information they received. From the analyzing point of view this means counting the number of decision points. In the *act* phase we are observing the actions based on the previous steps and analyzing the time to complete the task (sending information if requested or finishing other type of action).

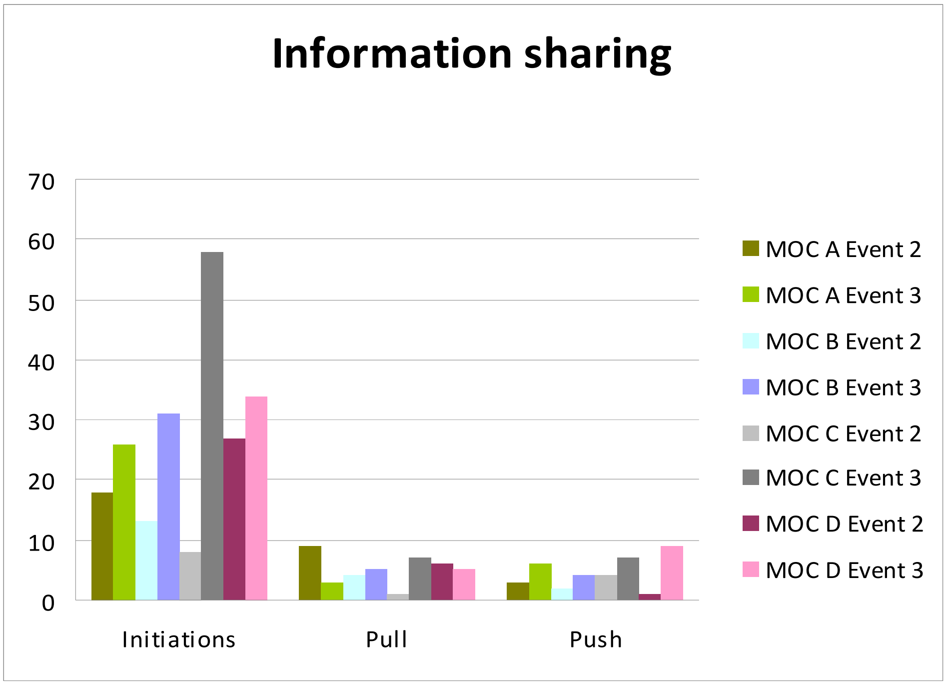

Figure 3 An example of information exchange

Figure 3 is an example of the gathered data from the MNE5 MSA Experimentation. This figure presents the amount of information sharing in two separate Events. It shows the amount of initiations and number of push and pulls from each MOC team. This type of information was crucial when we are searching reasons behind actions. Initiations represent how many times the MOC team was initiating an exchange of information. Pull is referred to situations when MOC asked for example for more information about some particular topic. Push refers to situations when MOC team has sent more information to others.

Reflecting back to the OODA-loop, we are able to identify aspects and meters that measure how we can capture and eventually measure the time between the different steps of the OODA-loop.

- **Meter 1 Observe (Type of data received)**
  - Counting the types of information and the means of receiving and sending information.
  - How many times the MOC was unable to receive or react to the sent information (what means of communication causes delays)?
- **Meter 2 Orient (Amount of data)**
  - Too much data leads to mental overload.
  - Too little data means that the system does not offer adequate SA.
  - How many times there were failures (reasons for delays, social or organizational)?
- **Meter 3 Decide (Number of decision points)**
  - How many decision points are found (the number of decision points)?
- **Meter 4 Act (Time to act from the first step)**
  - Time counted from the sent to the action (information about time delay).

With the list of Meters and the figure 4 of an example of different meters in different stages of the OODA-loop we present a way to create meters that collect the needed data in certain stages of the OODA-loop in order to monitor usability. The QoS parameters are selected by taking all phases in the OODA-loop into account. They are not only measures of delays, jitter, errors, losses and availability but also contain parameters related to understandability. It should be understood that most of the variables that were measured in the experimentation were not technical performance variables: there is a lot of crucial information that needed to be collected in more qualitative ways. At a later stage, MOS functions should be created and target values for reference connections set. At that later stage we ideally would only need quantitative data, such as the number of messages or response times to a message, i.e., only measurable numbers. Before defining the MOS functions, in order to fully understand the given task and the result, we needed to look deeper into the process that the user proceeded with. The basic meters help to get pieces of the information, but in order to measure true performance, the pieces must be put together by using qualitative methods. In the MNE5 MSA experimentation quantitative data was interpreted with observations notes, interviews and surveys. In design and deployment stages of a new or updated information system, the QoS monitoring concept gives the basic guidelines where to look for problems, and it provides a framework for understanding the complexity of performing different types of tasks with technical systems. The new concept helps evaluators and designers to focus on the actual challenges and how to take an advantage of the user feedback.

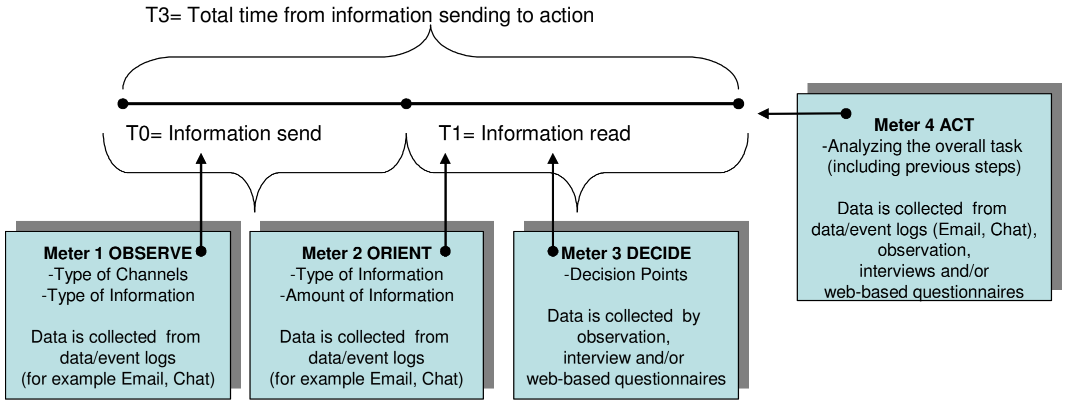

Figure 4 An example of different meters in different stages of the OODA-loop

Figure 4 explains the points that we can capture and count when measuring QoS in the Act stage. In order to understand why the response took too much time or the action was false, we need to identify the phases of information sharing. We can measure the time between different actions with time stamps, and deeper analysis can be performed through observation and interviews. Data collection can also be arranged during run-time by collecting the data for example by online-questionnaires that the users take part in. In the original QoS monitoring model of Figure 1 the problems are limited to the Act stage only. Our new concept extends the scope to cover all of the stages. The Act stage is simply the final phase which uses the results of previous stages, and that amplifies the importance of the other steps even more. Table 1 is an example of information sharing delays in the experimentation.

Measuring the time from sending a request to receiving a reply naturally does not tell us much about the actual quality of service. It is not providing us with information about the situation where the information exchange took place and how the information was created. By observation we were able to capture not only the log files, but the actual times when the operators were able to read the received messages and reasons for any delays. It was interesting to discover that reasons for delays could be a technical, organizational or social issue. Difficulties with the technology can cause distractions and make access to the information more difficult.

Scenario 1 **issue**	**time of sent**	**time of read**	**time of action started**	**time of action finished**	**mean of communication**
RFI reply from another MOC	10:21:00	10:34:00		no action	Email
IR from another MOC	10:40:00	10:55:00		no action	Email
another MOC asks info about mmsi number	11:10:00	10:59:00		no action	Chat

Scenario 2 **issue**	**time of sent**	**time of read**	**time of action started**	**time of action finished**	**mean of communication**
RFI from FIN MOC to WC (coast guard)	8:04:00	------	7:48:00	8:22	Email
RFI from FIN To WC (National Board of navigation) cc to ALL	9:07:00	------		8:06	Email
info from another MOC	9:48	10:04		no action	Email
another MOC send information	10:47	11:45		no action	Email
Reply to another MOC	13:32	13:35		13:37	Chat
Information from another MOC	13:34	13:40		no action	Email

Table 1 An example of information sharing delays in the Act stage

On the other hand, from the organizational perspective we were able to discover that limited authority caused delays because operators had to wait for a response from the higher-level headquarters in order to respond. From the social aspect the actors' own prioritizing also affected the response time. If the operators received a lot of information and requests at the same time, they prioritized and acted based on their own judgment.

TIME	SUBJECT	MEDIA
746	RFI for Vessel X	Email
849	FAILED CALL TO NATO for clarification	Voice
851	RFI Vessel X	Voice
853	RFI Vessel X	Voice
853	Vessel X position	Voice
856	Asking CATES in WC detention records of Vessel X	Email
904	No information from CATES	Email
911	For MOC 4 there is nothing we found on Vessel X besides the position we given you	Chat
915	For MOC 3 TY for looking	Chat
919	MOC 4: my information shows two similar ships with the same IMO number with Vessel X, propable name changes Vessel Y, Z	Chat
934	Re: RFI for Vessel X	Email
940	COI DESIGNATION FOR VESSEL X	Email
949	Action for Vessel X	Email
1021	Re: Action for Vessel X	Email
1048	Vessel X information	Email
1311	Re: Action for Vessel X	Email

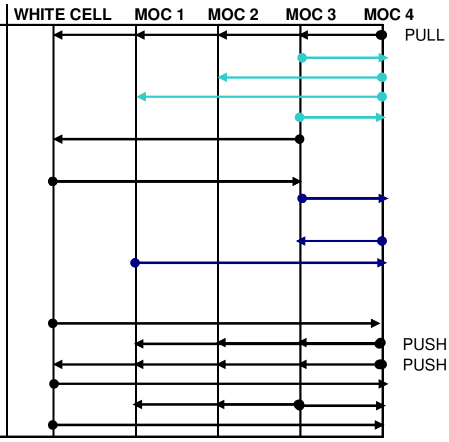

Figure 5 Interaction diagram of information sharing

Figure 5 is an example of the interaction between different actors sharing information about one particular subject. From the interaction diagram we are able to see in a certain time frame, what type of interaction happened, what kind of tools were used and with whom the actual interaction occurred. To reflect the findings back to the QoS, we can actually see factors affecting the collaboration and usage of different tools. Based on the MNE5 MSA Event 3 it is possible to identify situations when it is crucial to look at the timeframe and focus on issues concerning QoS and Usability Monitoring. When looking at the OODA-loop, we are able to see that in the *Observe* stage we are using different resources that can be our own technical system, open databases or contacting other MOC teams via email, chat or voice. We can analyze the type of data received at this stage and also the channels used. The *Orient* stage includes scoping the task, and from the analyzing perspective we are focusing on the amount of data in the given time frame. The *Decide* phase includes decision points when the team decides to act based on the information they received, and from the analyzing point of view this means counting the number of decision points. In the final *Act* phase we are observing the actions based on previous steps and analyzing the time to complete the task (sending information if requested or finishing other types of action).

Based on the data from MNE5 MSA Event 3 we have explained by examples the basic principles of four different meter levels corresponding to the OODA-loop. In the design stage and during the deployment of new information systems the importance of Usability Monitoring cannot be highlighted enough. According to our concept by using four meters to measure QoS, it is possible to identify usability problems and challenges of the new system. The actual testing is done in a test environment with a test scenario where users are given a couple of tasks. Meters capture the data and interactions that occur during the test run. By following the concept and adding surveys and interview with observation we get crucial information about issues that hinder information sharing. After testing the new system, reference values for Usability Monitoring can be created and compared in future usage of the system. By combining quantitative and qualitative data we are able to show the influence of delays in information sharing, whether it is caused by the technical system or human error or is simply a delay from using a formalized report form, divided to social, technical and organizational factors.

The *Observe* stage is measured by a different type of logs that record actions like sending emails, chat logs, and phone records. From that data we gather statistical information about the channels of communication and also the amount of shared information. The *Orient* stage focuses more on the amount of data and information management; i.e., how the user reacts to the received data. A simple technical measurement is not covering the entire truth but we get important information with reference cases for example about the amount of data that can cause overload and also can capture the number of failures in connections. In the *Decide* stage we gather the number of decision points during one task. There is no simple way to do that automatically, but by end-user questionnaires and observation supported by interview we gather that type of information. In the final stage, *Act*, we can measure the time to proceed with the task and possible delays. Most of the Meters can be formed quite easily for a run-time evaluation and collected for most parts automatically but in order to gain the maximum benefit of the method, observation and interviews are required. Although the level of significance of observation and interviews can be minimized for example by using run-time web-questionnaires in order to collect user's impressions.

In the MNE5 MSA experimentation we were able to obtain information that showed how the tools affected the operators' decision making and what type of process they went through while solving the task at hand. As observed in [25], the focus of analysis is in information; how information is held, exchanged, represented and transformed by users regardless of the existing technological infrastructure and organizational framework. In the MNE5 MSA Experimentation we presented the quantity of information exchange, examples of the actions taken by the MOC teams and interactions between them in order to provide a better understanding of the situation. This case study demonstrates that by monitoring these types of measurable variables we can measure the level of usability and analyze the user's level of situational awareness with the respect to information sharing. Figure 6 represents OODA-loop stages and methods that can be used during each step.

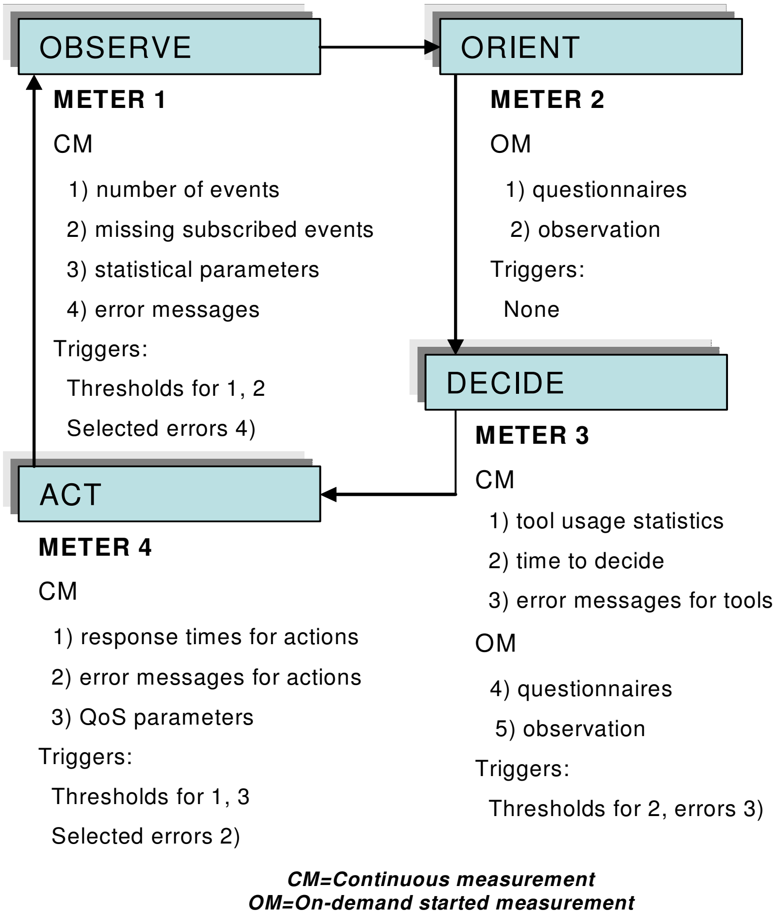

Figure 6 Usability measurements points in the OODA loop.

6. Reflections from related studies from the end user perspective

As Redmiles (2002) stated in [20] development goals generally include end user views distributed across many disciplines, yet there has not been enough research in order to monitor the usability of a system from the end user perspective. Redmiles (2002) ideology of activities in Human-Centered Software Development does not address issues that are central to the Usability Monitoring concept even though he brings up the importance of the workplace environment and expands the meaning of end user to a much wider area. Cardoso, Sheth & Kochut (2002) presented ideas for workflows with QoS [5]. They focus on business processes and for them QoS means analyzing time, cost, reliability, and fidelity metrics. This does not cover the ideology of reflecting QoS to users' views or support for example decision making. Georgievski & Sharda (2003) presented a real-time management of QoS with the Three Layer QoS model including User Perspective, Application Perspective, and Transmission Perspective [8], [9]. One aim was to investigate how the user can interact with the QoS Processing System in real-time. The user element was presented, but not in the same context as we present it in our Usability Monitoring concept. It is also possible to study usability from other point of views, for example from the system perspective, and use models like ITIL [13]. Nevertheless we chose to focus on the human actor, the actual user, and developed the concept to support the end user by using the OODA-loop. End user point of view is crucial because the actual users are the key actors in the organization processing the given information with the usable tools. By supporting the end user, we support the entire organization in achieving its goals. Related studies cover some parts of the metrics of QoS and represent many different ways to measure usability. Our new concept is based on the user's point of view for monitoring usability of systems in the operational phase, and, as the case study shows, it is useful also on the design and implementation phase of new information systems by introducing collected methods following the steps of the OODA-loop.

According to Salmon, Stanton, Walker & Green (2006) existing monitoring methods do not address the problem of Situational Awareness (SA) because current SA measurement techniques focus only on individual SA and approaches have issues that can detract from obtained SA data [23]. As we recognize, there are three levels of awareness that are situation awareness, team awareness and organizational awareness [4]. We are focusing to the situation awareness by also acknowledging that the level of situation awareness of one user affects the team's awareness and the overall awareness of the organization. That is why we are focusing on the actual user and his/her level of SA in order to improve the overall SA of the current ongoing task. Information sharing and collaboration has been recognized to be crucial elements also in air traffic situation and it has been stated that researchers have neglected looking at SA from a team perspective. We need to also understand that SA is applicable to dynamic situations with changing variables such as in the maritime environment [18]. The definition of team dynamics is also a problematic: what do the actors know about their own and other actors' workload and how is this supported by technology. The crucial question has been to see and understand that Team SA needs to be more than collective average of SA of the individual actors in the team [26]. This is the reason why we are focusing with our Usability Monitoring to the individual actors SA, because by monitoring certain steps, we are able to gather valid information from all the actors involved in the specified team. This model is scalable from one individual and his/her SA to team SA by taking into account each actors SA to build up the entire picture of the situation.

7. Conclusions

Usability is an aspect involving human factors and one may ask if the proposed model measures usability, or if any fixed set of technical parameters can measure usability. This question is irrelevant since the proposed model of Usability Monitoring does not intend to measure usability. Usability Monitoring is an extension of QoS monitoring. QoS monitoring does not measure the QoS level, and does not measure all aspects of QoS, but it takes continuous or periodical measurements of certain technical parameters in order to detect indications that the system does not offer adequate QoS. If such parameters are well chosen, the small set of monitored parameters indicates a large range of underlying QoS problems without specifically measuring each of them. Furthermore, if the system is well designed it should give good QoS unless there are problems, thus the lack of problems can be taken as an indication that the system offers good QoS. Similarly, Usability Monitoring does not measure usability but detects by technical measurements signs that the system does not operate in the way as it is intended. If the system is originally designed to have good usability, Usability Monitoring measurements satisfying target values indicate that usability of the system is as good as designed. This article presents a Usability Monitoring model that suits to C4ISR applications for situational awareness. Usability of such an application is closely related to the ability to make good decisions. If the system is slow, causes mistakes, is prone to errors, or is confusing, it cannot be effectively used in decision making. The improvement to the QoS monitoring model is the extension of the scope. Usability Monitoring tackles each stage of the OODA-loop and gives the higher level management a tool to see if the OODA-loop slows down.

MNE5 MSA Event 3 was a multinational experimentation where operators needed to collaborate in order to complete their tasks. This case study gave us a platform to partially evaluate the concept of Usability Monitoring. Especially, we were able to select Meters for each stage of the OODA-loop. In the *Observe* stage we analyze the type of channels we are using and the data we are receiving. From the *Orient* stage we identify the amount of data that causes need for information management. In the *Decide* stage we analyze how many decision points the operator has and does s/he have enough information. And finally in the *Act* stage we collect information in order to analyze successfulness of the operation and how much time it took for the operator to act and finish the task. The MSA experimentation collected plenty of usability information and we could see what usability problems appear and how they are connected with measurable technical Usability Monitoring parameters. The basic technical level data is collected automatically but much of the descriptive qualitative usability data is obtained by questionnaires, observation etc. methods that cannot be collected automatically. The Usability Concept was mainly developed for the design and implementation phase when it is easier with a test scenario to analyze the Meters and evaluate the system with the quantitative and qualitative methods. In this case study we did not continue to definition of target values and reference connections because target values need to be set after each test scenarios in different environments. The Usability Monitoring is a guideline to proceed with a test during the design or implementation phase and collect the data with the given methods and set target values after identifying the gaps and solving the causes of errors or delays. Focusing on the OODA-loop and by looking at technical measurable parameters gave a way to systematically observe the usability of the used system in order to find gaps that are affecting the user's Situational Awareness. This partial evaluation shows that Usability Monitoring is a promising approach, but the QoS approach for Usability Monitoring is a new idea and there is still much work before the ideas can be fully realized in operational systems.

References

[1] C. Aurrecoechea, A.T. Campbell & L. Hauw, *A survey of QoS architectures Multimedia Systems 6,* 1998*,* p. 138–151.

[2] C. Brunett, K. Choo, C. Eshelman-Haynes, A. Koskinen, B.K. Soh, and F. Utterstöm: *MNE MSA White Shipping Experiment Final Report,* 2008, available at http://tide.act.nato.int/mediawiki/index.php/MNE5_MSA_White_Shipping_Experiment_ Final_Report

[3] R. Bunge, S. Chung,, B. Endicott-Popovsky, and D. McLane: "An Operational Framework for Service Oriented Architecture Network Security*,"* in *Proc. 41st Hawaii Int. Conf. on System Sciences*, 2008. IEE Database.

[4] P. Carayon, "Human factors of complex sociotechnical systems," *Applied Ergonomics*, Volume 37, 2006, p. 525–535.

[5] J.Cardoso, A. Sheth and K.Kochut, "Implementing QoS Management for Workflow Systems,"
Technical Report, LSDIS Lab, Computer Science, University of Georgia, July, 2002,

http://lsdis.cs.uga.edu/lib/download/CSK02-QoS-implementation-TR.pdf. *referenced 29. Feb. 2010.*

[6] F. T. Durso & A. Sethumadhavan: ”Situation Awareness: Understanding Dynamic Environments,” *Human Factors,* 2008, Volume 50, p. 442-448.

[7] T. Erl: *Service-Oriented Architecture, Concepts, Technology, and Design*, Prentice-Hall, 2006.

[8] M.Georgievski & N.Sharda: “Usability Testing for Real-Time QoS Management,” in *24th IEEE International Real-Time Systems Symposium,* RTSS2003, Dec 3-5, 2003, Cancun, Mexico, http://sci.vu.edu.au/~nalin/RTSS2003GeorgievskiShardaPaper.pdf.

[9] M. Georgievski & N. Sharda: “A taxonomy of QoS parameters and applications for multimedia communications,” 2003, http://eureka.vu.edu.au/~nalin/IMSA2003GeorgievskiShardaPaper.pdf.

[10] T. Grant: “Unifying Planning and Control using an OODA-based Architecture,” *Proceedings of SAICSIT 2005*, p. 159-170.

[11] N. C. Goodwin: “Functionality and usability,” *Communications of the ACM*, March 1987, Vol. 30, No. 3, p. 229-233.

[12] J. Gulliksen, I. Boivie, J. Persson, A. Hektor, and L. Herulf: ”Making a difference – a survey of the usability profession in Sweden,” *NordiCHI’04,* October 23-27, 2004 Tampere, Finland, p. 207-215.

[13] K. Ishibashi: “Maintaining Quality of Service Based on ITIL-Based IT Service Management,” *FUJITSU Sci. Tech. J.,* 2007, 43, p. 334-344. http://www.fujitsu.com/downloads/MAG/vol43-3/paper11.pdf.

[14] *ITU-T Recommendations.* http://www.itu.int/itu-t/recommendations/index.aspx?ser=E

[15] J. Jormakka and K. Heikkinen: “QoS/GOS parameter definitions and measurement in IP/ATM networks,” *QofIS’2000*, Berlin, in *Lecture Notes in Computer Science 1922*, (ed. J. Crowcroft, J. Roberts, M. Smirnow), Springer-Verlag, 2000, p. 182-193.

[16] J. Jormakka and J. Lucenius: “Architectural Aspects of Migrating C^4I^2SR Systems to Service-Oriented Architecture,” *IEEE/IARIA ICDT 2010*, 2010, June 13-19, Athens.

[17] A. Koskinen: “Maritime Situational Awareness and usability – Operators point of View,” *User’s view on battlespace systems (eds. J. Jormakka, S. Oksa),* Finnish National Defence, University Department of Military Technology Series 3 No 9, 2009, p. 23-41.

[18] J. Langan-Fox, M. J.Sankey, and J. M. Canty: “Human Factors Measurement for Future Air Traffic Control Systems, Human Factors,” *The Journal of the Human Factors and Ergonomics Society,* 2009, Volume 51, p.595 - 637.

[19] J. F. Meyer: “Service Oriented Architecture (SOA) Migration Strategy for U.S. Operational Naval meterology and Oceanography (METOC),” 2007, no 070313-001, *IEEE DataBase*.

[20] D. F. Redmiles: “Supporting the End Users’ Views,” in *AVI,* May 22-24, 2002 Trento, Italy, http://www.ics.uci.edu/~redmiles/publications/C036-Red02.pdf.

[21] D. Russel, N. Lookes, L. Lu, and J. Xu: “Service-Oriented Integration of Systems for Military Capability,” *11th IEEE Symposium on Object Oriented Real-Time Distributed Computing* (ISORC), 2008, p. 33-41.

[22] D. Russel, N. Lookes, and J. Xu: “SOA, Dependability, and Measures and Metrics for Network Enabled Capability,” *IEEE Database*, 2008.

[23] P. Salmon, N.Stanton, G. Walker & Damian Green: “Situation awareness measurement: A review of applicability for C4i environments,” *Applied Ergonomics,* 2006, Volume 37, p. 225-238.

[24] D. P. Siewiorek and R. S. Swarz: *Reliable computer systems: Design and evaluation*, third edition. 1998, ISBN 1-56881-092-X.

[25] G. Walker, N. Stanton, R. Stewart, D. Jenkins, L. Wells, P. Salmon and C. Baber: ”Using an integrated methods approach to analyse the emergent properties of military command and control,” *Applied Ergonomics,* 2009, Volume 40, p. 636–647.

[26] C. D. Wickens: “Situation Awareness: Review of Mica Endsley's 1995 Articles on Situation Awareness Theory and Measurement,” *Human Factors: The Journal of the Human Factors and Ergonomics Society,* 2008, Volume 50, p. 397-403.