THE ECOLOGICAL DOMAIN OF WARFARE

Jens-Olof Lindh
Kockums AB
SE-371 82 KARLSKRONA
SWEDEN
+46 455 68 3043
jens-olof.lindh@kockums.se

Abstract

This paper suggests that the physical, information, cognitive and social domain layers of warfare should be extended to also include an ecological layer. We compare the extended domain-model to the four-layered architecture for open computational systems (OCS). Implementations of (i) the interaction platform *Service oriented layered architecture for communicating entities* (SOLACE) and (ii) the *Distributed interaction system for complex entity relation networks* (DISCERN) are introduced. The demonstrator system *Trustworthy and sustainable operations in marine environments* (TWOSOME) shows that it is possible to create and control distributed systems that are exposed the evolutionary behaviors of an ecological environment. The relevance of TWOSOME to Network-Centric Warfare is supported by a comparison of the OCS architecture to the domains of warfare. We also argue that systems based on the OCS architecture are both compliant with requirements on essential operational capabilities of network enabled forces and conforming to the Global Information Grid.

Introduction

The recognition of ecological properties in network centric systems is by no means new. For example Cebrowski and Gartska (1998) suggested that the three main themes of the emerging Network-Centric Warfare (NCW) would be (i) the shift in focus from the platform to the network, (ii) the shift from viewing actors as independent to viewing them as part of a continuously adapting ecosystem, and (iii) the importance of making strategic choices to adapt or even survive in such changing ecosystems. [3] Concurrently, Czerwinsky investigated the bounds of military performance in terms of e.g., unpredictability, nonlinearity and self organized criticality. [4] The issues of complex interactions between tightly coupled units have also been highlighted. [9, 10] However, it seems that contemporary models of NCW and Network Enabled Capabilities (NEC) have not quite managed to include the concepts of sustainability and changing ecosystems.

We suggest that the four-layer domain model of warfare is added a fifth layer; the ecological domain. We argues that the introduction of the ecological layer is relevant since technologies developed on the open computational systems (OCS) architecture shows mature Network-Centric Warfare behaviors, and the ecological layer enables a mapping between the domains of warfare and the OCS architecture.

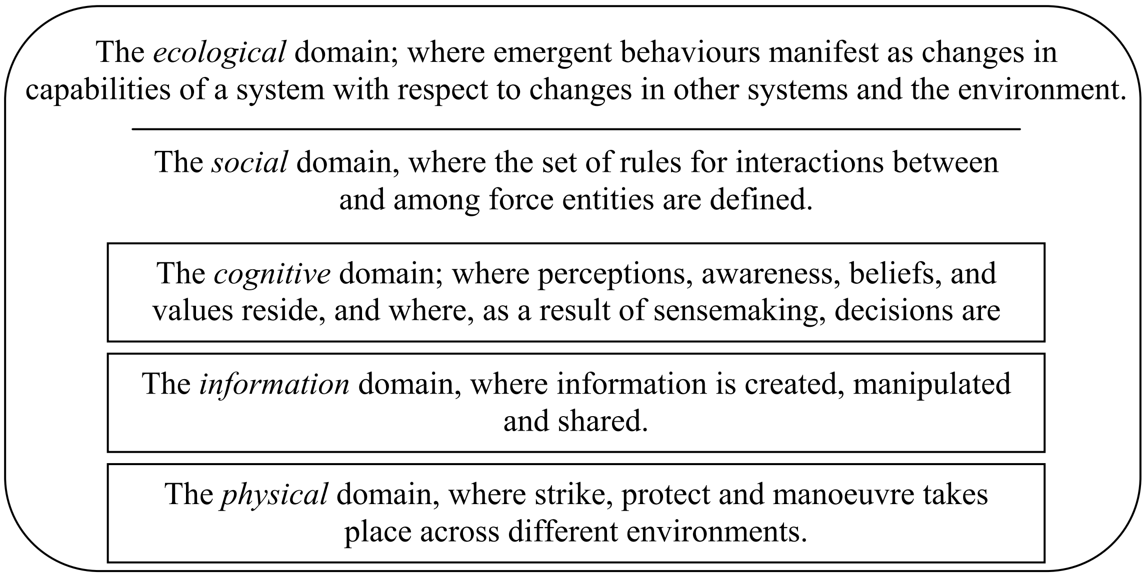

Figure 1: A five-layered domain model of warfare.

This paper is divided into four blocks. The first block introduces the suggested ecological layer and some challenges that follow. The second block introduces the architecture for open computational systems. The third provides a brief dissemination of technologies and a demonstrator that provides examples of mature network enabled capabilities. This is the empirical support for the claims of relevance. The fourth block discusses the conformity of the architecture for open computational systems to the domain model of warfare and the applicability to the requirements posed on information age forces.

The paper can be confusing in the use of the word "domain". In the five-layered model of warfare, every layer is labeled a *domain*. However, in the architecture for open computational systems, the *Domain* is one specific layer. In order to differ between these two meanings, the layers of the open architecture are referred to with capital first letter.

The fifth layer of warfare; the ecological domain

The domains of warfare are presented as means for understanding how information affects our ability to perform military operations. The physical, information and cognitive domains are defined and elaborated in Alberts et al (2001). [1] The social domain is added in Alberts and Hayes (2003). [2] We suggest that an ecological domain is added to this model (see *Figure 1*). The ecological domain would serve as a connection between the social and the physical domains.

The ecological domain is defined as where emergent behaviours are manifested as the changes in capabilities of a system with respect to changes in other systems and the environment. The major implication of setting network enabled systems in an ecological context is that they can be regarded as evolving with the real environment in which all physical actions take place. This evolution is a consequence of the behaviors of all related systems, which in turn are defined by a combination of the rules for interactions as defined in the social domains, and the course of events in the physical domain. Therefore, the main benefit of adding the fifth, ecological, domain is that it connects the states of the systems to the progress of events in the physical world.

An evolutionary perspective on artifacts poses specific challenges on the engineering tasks. For example, the ever changing environment will pose continuously changing requirements on the systems. When the requirements change faster than the development process can cope with, we have to recognize that relevant requirements are unknown at the system design time. An evolutionary system is neither created nor decommissioned. Rather, we have a situation in which maintenance may be needed at any time, for components that are continuously up and running. There is a need for on line engineering. In addition to this, e.g., agent based services in systems of systems may create behaviors and show emergent properties of clearly cognitive character. A consequence of this is that the command and control situation will have to cope with multiple feed-back-feed-forward loops necessary for assessing the current states and predicting the possible consequences of available control actions, in systems of systems where the distribution of cognitive abilities between men and machines has a major impact on events. We address this situation with the architecture and technologies for open computational systems.

An architecture for open computational systems

One major problem in the transfer from platform-centric to network enabled systems is the conception of the environment. The platform centric perspective assumes that systems are *closed*, meaning that that there is some distinguishable border between the system and its surroundings. In closed systems, stable interactions are assumed and the goal is to achieving local (internal) coherence. On the other hand, the anticipated flexibility of network enabled systems implies that they would be *open*. Open systems are designed to anticipate and support interactions and structures that evolve over time. They strive to maintain a sustainable global coherence. Since systems in the open perspective are *parts of* the environment rather than *parts in* the environment, the context in which the system resides (and thus the requirements facing the system) are considered to change continuously. Studies of these properties has lead to the methodology and architecture of open computational systems (OCS). [5]

Open computational systems are in this context to be understood as systems that are evolving over time in such a way that (i) there is no real starting or stopping the system, (ii) it is not possible to know in advance, what the operating circumstances will be and (iii) there is a significant amount of computers and software in the loop. [6]. Note that human operators by no means are excluded in this view.

The architecture for open computational systems is based on the four levels *Domain*, *System*, *Fabric* and *Environment* (see *Figure 2*).

The Domain level contains the definitions of all the words that are to be used in solving an addressed mission. Domains are defined based on the tasks of the connected forces. The networks are established by the participation of actors (entities, software agents, operators, etc.) that declare membership of a specific Domain. This is the same as saying that they can communicate using the same language. Note that this implies that the network is not built on the ability to *connect*, but on the ability and need to *address a common task*.

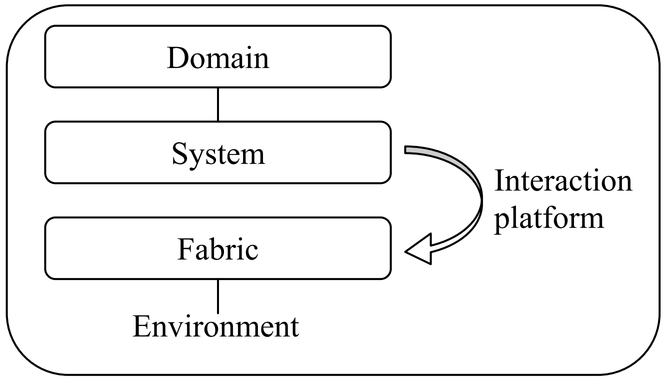

Figure 2: A layered architecture according to the methodology of open computational systems.

The System level contains all the code (in computers) and knowledge (in people) needed to assess situations and create behaviours. The open architecture is inherently service oriented. All services are related to the definitions at the Domain level.

The Fabric level contains all the hardware that is necessary for performing computations and maintaining communications; that is, the information technology. Physical connections between the nodes of the network are created and established at this level. The services at the System level are decoupled from the hardware at the Fabric level by the interface(s) created by the interaction platform. One consequence of this is that (System level) software and (Fabric level) hardware can be developed and maintained independent of each other. Another consequence is that a need for communication created at the System level can make use of a number of different communication technologies. This adds redundancy and robustness to the interaction between the nodes of the network. Also, any given computer can host a number of System level services from a number of Domains.

The Environment level corresponds to the ever evolving reality that provides the context for any and every operation and activity. As declared above, the purpose of the open architecture is to enable sustainable behaviours of real world man-machine interaction systems. Therefore, the environment is both the prerequisite and the arena for the networks that are established. It is a prerequisite in terms of providing both the basis for the task dependent Domain definitions and the series of often unanticipated events that create the situations that need to be handled. It is also the arena for the network, since the components of the systems usually are situated on artefacts (vessels, sensors, weapons, supply systems etc.) that really are (or will be) out there, in the environment.

Technology and demonstration

The properties of open systems in an NCW/NEC setting tell us that the problem is not to get connected, but to provide the required quality of service over extended periods of time; i.e., meeting sustainability criteria. In order to have sustainable open systems we need for example (i) the ability to observe the accessible participants of a network, (ii) to observe the emergent behaviours of the connected entities and (iii) the ability to interact

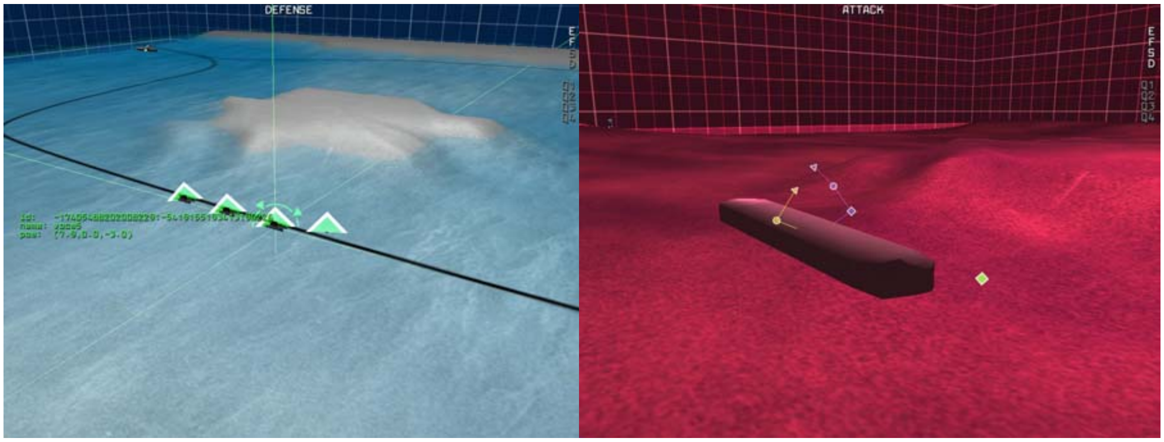

Figure 3: Screenshots from the TWOSOME demonstrator. The blue team tool shows that three vessels are synchronized in order to emulate a large vessel. At the same time, the red team tool indicates that the mine has classified the present signatures as a large vessel.

with various participants in the network. Interaction is necessary both for the purposes of command and control as well as for maintenance. We have to manage on line, real time supervision and engineering of the up-and-running complex system. This section describes three principally different pieces of technology addressing these challenges. These are the *interaction platform*, the *interaction tool*, and the *demonstration system.*

Interaction platform

The *Service oriented layered architecture for interacting entities* (SOLACE) is an interaction platform that provides the interfaces between the System and the Fabric layers. The purpose of the interaction platform is to make sure that services that reside at different nodes in the network (i) get to know about each other and (ii) can establish communication. As such, SOLACE

- enables the observation and recognition of available Domains, which in turn enables the *ad hoc* network construction needed in an open environment,
- provides interfaces between System level services and Fabric level hardware, and,
- makes sure that the System level services can find other services declared using the same Domain vocabulary, so they can interact.

Interaction tool

The *Distributed interaction system for complex entity relation networks* (DISCERN) is a tool that provides visualisations based on the information that can be made available through the network. The purpose of the interaction tool is to provide a graphical interface to human users of the system. Some properties of DISCERN are

- the three-dimensional visualisation of some geography and/or virtual space, in which we can observe the evolving behaviour of events (e.g., moving vessels),
- the visualisation of qualitative and/or abstract constructs such as Fabric level nodes, System level services and the available Domains,

– the point-and-click ability of services. This means that we can open a real time command interface to any available service in the network.

Any defined word of a Domain can be mapped to a specific graphic expression. Since the visuals are defined and produced in the tool, only the underlying information needs to be distributed over the network. As different scopes of visualisation are selected, only the specific information needed for presenting the current state is asked for. Therefore, the amount of information needed may vary substantially depending on design and operational needs. In any case, the amount of information transferred over the network is substantially less than the content of the graphics actually shown to the operator.

Demonstration system

A system is here meant to be the group of interacting services and entities that can provide a coherent behaviour. The *Trustworthy and sustainable operations in marine environments* (TWOSOME) is a demonstrator built with the technologies indicated above[1]. [7] The purposes of TWOSOME are to investigate (i) the behaviours of open computational systems and (ii) the technology in applications related to naval tasks. The goals are to establish and visualize the behaviour of distributed, asynchronous service-based autonomous entities that forms a network.

TWOSOME shows how a naval transport vessel receives a mine-clearance task. This task is communicated to a cluster of three autonomous mine-sweepers that are activated, introduced to the network, launched to sea and set to work by the transporter. The cluster works autonomously showing shared situation awareness and self coordination in solving the task of emulating different sizes of naval vessels. They can also react to performance degradation (e.g., loss of services) by autonomous reconfiguration of the cluster.

DISCERN provides two different but parallel views of the events (see *Figure 3*). The visualisation makes behaviour and events in the networked system easier to understand and possible to control. The blue defence-view presents the world as understood by an operations centre. The red attack-view presents the world as understood by a mine. In the blue view it can be seen that the three mine counter-measure vessels are on track. In the red view it can be seen that the mine has classified the current pattern of signatures as a large vessel. The following main entities are modelled in TWOSOME:

– Sensor - a sonar buoy that reports bearings to the operations centre.
– Operations centre **-** collects sensor data, articulates tasks and allocate resources.
– Transporter – a vessel that receives the tasks of transporting and deploying the defenders to the area currently under threat.
– Attacker – a mine that classify received acoustic and magnetic signatures.
– Defenders – three mine counter-measure vessels (SAMs) that are deployed by the transporter. The cluster of SAMs share a common goal and coordinate in order to deceive the attacker into believing that there is one big vessel, rather than three small. They also have the ability to respond to changes in both the environment

[1] Read more about TWOSOME and other systems at http://www.soclab.bth.se/system.html .

and in their own functionality. The SAMs may share and exchange information within the cluster, with the transporter and/or with the operations-centre.

Some of the capabilities of TWOSOME (as built on SOLACE and DISCERN) that are directly related to network enabled operations are:

- The ad hoc insertion, observation and extraction of nodes to the network.
- The activation and allocation of services that enable predefined behaviours.
- The ability to share information, enabling pattern classification and abstraction of concepts (i.e., information fusion).
- The shared situation awareness in terms of (i) tools that enable distributed and parallel human interaction with systems and networks and (ii) System level code that enable machine-to-machine interaction.
- The implementation of services that enable collaboration and self-synchronization between asynchronous and distributed nodes.

A classification of the maturity of NCW systems is presented in Alberts et al (2001). [1] Admitting the explicit simplicity of the mine sweeping scenario, it can be argued that TWOSOME shows NCW behaviour at maturity level four. [7] It is a step towards on line control and engineering of distributed evolutionary systems.

Discussion

Requirements on control

The architecture for open computational systems provides a distinction between some features of dynamic systems. The Environment is under constant evolution. The Fabric is active in the mediation of information, thus providing connectivity and interaction in physical terms. Interaction in logical terms is provided by the services at the System level. Any new entity can be observed and recognized as a member of a network, if and when it declares that it belongs to some specific Domain. In this way, any number of networks can be constructed in terms of the groups of entities that share the same definitions of the used symbols. Also, any entity can belong to zero or more Domains.

This interaction between the network and the reality is worth some consideration. One observation is that we have to create and maintain components, systems and networks in a large scale, prepared for situations and technologies that we do not know and can not design for in advance. Another observation is that control should be maintained in continuous feed-back-feed-forward loops that update the system(s). A final observation is that there really exist a number of such loops, nestled into each other; environment-to-system interaction, human-to-system interaction, and system-to-system interaction. This indicates that the real challenge of establishing sustainable systems in real world environments is not as much the ability to physically connect nodes to each other, as the ability to control the resulting behaviour – that is, creating the network enabled capabilities. [8]

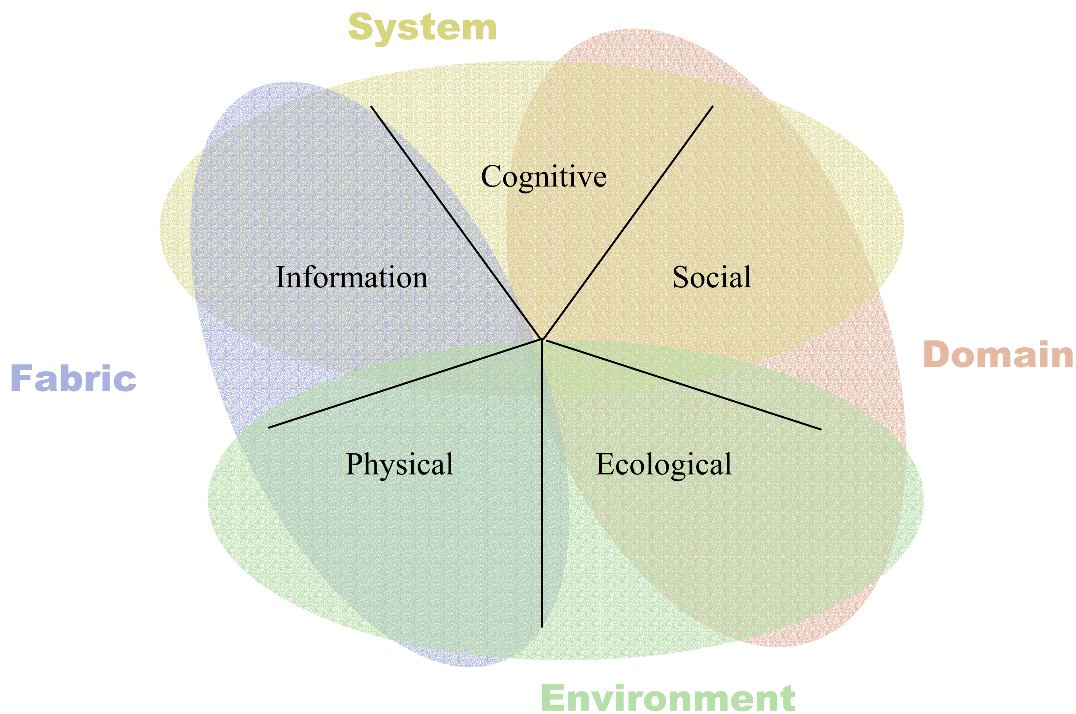

Figure 4: The relationships between the five-layered domain-model of warfare and the architecture for open computational systems.

Comparing models

Figure 4 presents the relationship between the five-level domain perspective of warfare and the architecture for open computational systems.

If we take the perspective of the five-level domain model perspective we see that the vertical model of *Figure 1* is converted to a pie-chart diagram. This is a consequence of the ecological domain bridging between the social and physical domains. Interaction between the domains is possible over the borders. The joint hub allows e.g., jumps of direct sensations from the physical to the cognitive domains, without violating the model.

If we take the perspective of the OCS architecture, we see that the Fabric layer corresponds to the physical and the information domains. This makes sense, since the Fabric layer contains the hardware and physical circumstances of communication and computation. The System layer corresponds to the information, the cognitive and the social domains. This makes sense, since this is where the human and/or software based services reside. These services take cognitive actions based on available information and decision criteria. The Domain layer corresponds to the cognitive, the social and the ecological domains. This makes sense, since the cognitive actions have to make sense in the grand scheme of developing events. Obviously, the social aspects would benefit from being defined with the same vocabulary as the services and the surrounding world. Finally, the Environment layer corresponds to the ecological and the physical domains.

This is assuring, since the ability to handle the evolutionary aspects of the real world was the very reason for establishing the methodology for open computational systems.

The open architecture and the characteristics of information age forces

In order to assess the applicability of the extended domain model and open architecture to the current understanding of NCW/NEC systems, we will relate the findings above to the characteristics of information age forces and the Global Information Grid.

Alberts and Hayes ([2], p.98) presents the four minimum essential capabilities required for a given network operation. The first is the ability to make sense of the information. This ability is supported by the Domain level definitions that encompass both the social and cognitive domain layers. The second is the ability to work in a coalition environment including partners from different organisations. This ability is supported twofold, by (i) the common language as defined in the Domain and (ii) the provision of interaction platforms and tools that enable communication and control. The third is the possession of appropriate means to respond using the tools of war and policy implementation. These means would not be affected by the open architecture. The fourth is the ability to orchestrate the available means to respond in a timely manner. This ability is supported by (i) the interaction tool enabling operator control and (ii) with the shown ability to use services on distributed nodes in a network for algorithmic self-synchronisation.

A comparison to the requirements on the Global Information Grid ([2], p 188) gives that implementations in accordance with the OCS architecture are highly compliant. The Domain layer enables the discovery of new nodes in a network. The System layer enables modular implementation of e.g., software based agents, as services. The Fabric layer enables the mediation of information between nodes. The interaction tool enables shared awareness and situated responses to perceived situations. The Environment layer provides the grounding of the emergent properties of self-synchronised systems to the realities of the physical world.

The evolutionary properties of open systems could be established if it was possible to make on line real time observations, measurements, instrumentation and maintenance on e.g., the services and concepts available in the open architecture. If so, the high maturity exemplified by the TWOSOME demonstrator would be supported by the sustainability of adaptable systems, providing mature network enabled capabilities. Such systems could be compliant with requirements on both essential operational capabilities and the ability to conform to the Global Information Grid.

Summary

While rules of interaction can be formulated in the social layer, the result of applying those rules to the decision processes of the cognitive layer will result in real world behaviours that evolve over time. The loops of a controlled decision process would be aiming at modifying the system in accordance with specified goals. Consecutive modifications lead to new states and abilities. An ecological behaviour emerges, one that would be consistent within the system and exposed to the competitive physical world in a manner best described as an evolutionary development.

Adding the ecological domain to the four-layered domain model of warfare enables describing a relation between the current physical context and the abilities of dynamic systems in an open environment. Architectures and technologies are needed, that can establish networks, make networks possible to maintain over extended periods of time, and provide the necessary control functionality. Examples of such technologies are implemented in the TWOSOME demonstrator, which shows that the open computational system architecture can enable mature network enabled behaviours. This is used as evidence supporting the claim that an ecological domain can connect the social domain to the physical. It is also argued (i) that the five-layer domain model of warfare and the architecture for open computational systems are highly correlated to each other, and (ii) that systems based the open architecture complies with requirements on both essential operational capabilities and the ability to conform to the Global Information Grid.

Acknowledgements

SOLACE and DISCERN are developed by SOCLAB at the School of Engineering at Blekinge Institute of Technology (BTH/TEK). TWOSOME is developed by SOCLAB and the property of Kockums AB.

References

1. Alberts, D.S., et al., *Understanding Information Age Warfare*. 2001, CCRP. p. 312.
2. Alberts, D.S. and R.E. Hayes, *Power to the Edge - Command and Control in the Information Age*, in *DoD Command and Control Program*. 2003, CCRP: Washington D.C. p. 259.
3. Cebrowski, A.K. and J.J. Gartska, *Network-Centric Warfare: Its Origin and Future.* Naval Insitute Proceedings, 1998. **124/1/1, 139** (January 1998).
4. Czerwinski, T.J., *Coping with the Bounds; Speculations on Nonlinearity in Military Affairs*. 1998, Washington DC.: DoD Command and Control Research Programme.
5. Fredriksson, M. and R. Gustavsson. *A Methodological Perspective on Engineering of Agent Societies*. in *Proceedings from ESAW 2001*. 2001. Prague Springer-Verlag Berlin Heidelberg.
6. Gustavsson, R. and M. Fredriksson. *Sustainable Information Ecosystems*. in *International conference on Software Engineering*. 2002. Orlando, Fl.
7. Lindh, J.-O. *Managing Systems for Network Centric Warfare*. in *Proceedings from Undersea Defence Technology Europe (UDT Europe-03)*. 2003. Malmö, Sweden June 24-26 2003.
8. Lindh, J.-O. *Cognitive Aspects of Network Enabled Operations*. in *21st Intl System Safety Conference*. 2003. Ottawa, Canada System Safety Society.
9. Moffat, J., *Complexity Theory and Network Centric Warfare*. DoD Command and Control Reserach Program. 2003, Washington D.C.: CCRP. 161.
10. Perrow, C., *Normal Accidents*. 2nd ed. 1999, Princeton, NJ: Princeton University Press. 451 p.