VISUALIZING OPEN COMPUTATIONAL SYSTEMS

TOMAS SAREKLINT
This thesis is submitted to the School of Engineering at Blekinge Institute of Technology in partial fulfilment of the requirements for the degree of Master of Science in Software Engineering. The thesis is equivalent to 20 weeks of full time studies.

Author
Tomas Sareklint
Email: tomas@sareklint.se

University advisor
Martin Fredriksson
Department of Interaction and System Design
Email: martin.fredriksson@bth.se
Phone: +46 (0)457 385825

Contact information
School of Engineering
Blekinge Institute of Technology
Box 520, SE-372 25 Ronneby
Sweden

Internet: www.bth.se/tek
Phone: +46 (0)457 385000
Fax: +46 (0)457 271 25
I’d like to acknowledge some who have supported me during the prolonged time of writing this thesis.

- My supervisor Martin for a countless number of hours of guidance, discussion, and reviewing.
- My family – in particular Johanna for withstanding my constant writing.
- SOCLAB, Martin H, associates, and friends for fruitful discussions and reviewing.

Thank you.
There is an emerging field in the design and development of complex systems, where systems are built upon components which in themselves are large scale systems – system of systems. Among other things, the system of systems viewpoint emphasises on open complex systems. In this thesis, the model of open computational systems is used to convey the constituents, dependencies, and interactions of such complex distributed systems.

These open complex systems are exposed to critical events, occurring in the systems execution environment. Moreover, these events may have negative effects on the system at hand, resulting in system behaviour diverging from intended. Also, to take all possible affecting events in consideration when designing the system is impossible.

By being able to instrument the system at hand in real time, i.e. online, one may be able to compensate the effects caused by critical events. However, to enable online instrumentation, one needs a supporting methodology which handles issues of an online nature and supporting technologies. In this thesis, this support is enabled by the methodology of online engineering and the technology of visualization. Furthermore, these instrumentations can be performed by cognitive agents – both human and software – which may explore and refine a specific system in conformance with their own, or cooperative, agendas and qualitative goals. To be able to perform the instrumentation, the cognitive agents need to be able to observe the phenomenon at hand to gain situation awareness, which in itself lies as a foundation for the decision process, carried out during the instrumentation phase.

With this in mind, one quickly realizes the importance of enabling observation of open computational systems for both human and software cognitive agents. If human cognitive agents are involved in applying the methodology; the requirements on how the system is represented for the observing human agent – how the system is visualized – grows even more important. In this thesis, we emphasise visualization technology as a supporting technology for human cognitive agents in their observation process. By providing human cognitive agents with visualization technology, we may enhance the result of their observation process and thereby also increase the possibility to reach their qualitative goals.

Hence, visualization of open computational systems affects a human cognitive agent’s situation awareness, which in itself lies as the foundation for the decision making process on instrumentation of the specific system at hand in conformance with the agent’s qualitative goals.

This thesis will present an evaluation of a supporting tool for visualization of systemic qualities in open computational systems. Such tool must supply functions which convey the set of requirements put forward by the selected model, method, and technology. Moreover, the evaluation will be accompanied by appropriate recommendations for improvement of such a tool.

Keywords: visualization, situational awareness, systemic qualities, open computational systems, online engineering.
**Visualizing open computational systems**

**Table of Contents**

**Chapter 1**

**Introduction**

1.1 Problem Domain ............................................................... 2  
1.2 Problem Description .......................................................... 2  
1.3 Approach and Contribution .................................................. 3  
1.4 Outline ......................................................................... 3

**Chapter 2**

**Background**

2.1 Critical Events ..................................................................... 5  
2.2 Systemic Qualities ............................................................ 6  
2.3 Technological Availability .................................................... 7  
2.4 Cognitive Processes ........................................................... 8  
2.5 Enabling Technologies ....................................................... 11  
2.6 Concluding Remarks ........................................................ 12

**Chapter 3**

**Phenomenon – Twosome**

3.1 Execution Environment ..................................................... 13  
3.2 Scenario Constituents ....................................................... 14  
3.3 System Behaviour ............................................................ 15  
3.4 Visualization ..................................................................... 16  
3.5 Concluding Remarks ........................................................ 17

**Chapter 4**

**Model - Open Computational Systems**

4.1 Open Computational Systems .............................................. 19  
4.2 Environment ..................................................................... 20  
4.3 Fabric ............................................................................... 21  
4.4 System ............................................................................ 22  
4.5 Domain ............................................................................ 22  
4.6 Concluding Remarks ........................................................ 23

**Chapter 5**

**Method – Online Engineering**

5.1 Online engineering ............................................................ 25  
5.2 Articulation ...................................................................... 26  
5.3 Construction ..................................................................... 27
INTRODUCTION

The term system of systems is widespread and generally recognized as large scale systems which are built upon independent and self contained components. Moreover, these components – or subsystems – are in them self often large complex systems and may be designed and built by different organizations with potentially conflicting goals and principal assumptions. The attributes distinguishing system of systems concept from large monolithic systems is their component independence, a geographic extent which limits component interaction to information exchange, their evolutionary nature, and emergent behaviours [3][4]. Based on these attributes, we have in this thesis chosen the model of open computational systems (OCS), which convey the constituents, dependencies, and interactions of such complex distributed systems [5].

With the presented system of systems attributes in mind, one can understand that these systems to a great extent will be affected by each other and by events occurring in their surrounding environment. Hence, a particular system must necessarily neither be the source of, or able to prevent, the occurrence of critical events – but might still be negatively affected, resulting in deviation from the intended system behaviour. When designing these open complex systems, it is extremely difficult, if not impossible, to take in account for all possible critical events, occurring across all components involved [3].

One way to address this issue of critical events is to enable instrumentation of the system at hand, in real time, i.e. online. Hence, by performing instrumentation, one may be able to compensate for the effects caused by critical events and thereby sustaining the intended system behaviour. To enable this in situ refinement of system qualities, one needs both a supporting methodology, which handles issues of an online nature, and supporting technology. In this thesis; we have chosen the methodology of online engineering (OE) [5] and the technology of visualization. The methodology emphasizes that on the one hand, the establishment of qualitative system behaviour is obtained by performing an iterative in situ refinement of the system at hand. On the other hand, to be able to perform this refinement the functional system design must be made accessible online for adaptation, in the form of an online accessible model of behaviour [5].

The principal assumption at hand is that these instrumentations can be performed by cognitive agents – both human and software. Moreover, the cognitive agents may perform exploration and refinement of system qualities, accordingly with their own, or common, qualitative goals. Also, to be able to perform instrumentation, i.e. adaptation of the system at hand due to affecting critical events in the systems environment, the cognitive agents must be able to observe the current phenomenon. The observation process results in an information acquisition; the cognitive agents gain situation awareness which will lie as a foundation for reasoning and decisions processes regarding how the system should be instrumented in conformance with qualitative goals.

With this in mind, one quickly realizes the importance of enabling observation of system behaviour in open computational systems for both human and software cognitive agents. Furthermore, if a human, as opposed to software, cognitive agent is performing instrumentation and thereby observation of the system at hand; the qualitative requirements on how the system behaviour is represented – how the system is visualized – are a fundamentally important research question to address. In this thesis we therefore emphasize on human cognitive agents performing instrumentation activities and thereby also
emphasize visualization of open computational systems. Alternative characterizations could be software cognitive agents, automation, and/or human-computer collaborative instrumentation.

Continuing, the following sections in this chapter will give the reader an introduction to the problem domain and a description of the problem at hand, which will be dealt with in this thesis. Also, we will present how we will approach the problem at hand as well as the resulting contributions in doing so.

1.1 PROBLEM DOMAIN

As discussed in the introduction, the emerging concept of system of systems depicts large scale integration of multiple independent components with the goal of satisfying a global need. Moreover, these components – or subsystems – are in them self often large complex systems and may be designed and built by different organizations with potential conflicting goals and assumptions. However, even though the subsystems are noted as independent, they are often interdependent – that is; each system may affect the other. An applicable example of this state of affairs is that of an airport. An airport involves a large set of complex systems, such as airplanes, air traffic control, and luggage handling – the airport in it self is a complex logistical system. Furthermore, for the complex airport system to function; every independent component must function, be able to cooperate with other components, and handle affects from possible events occurring in the airport.

In this thesis, however, we will emphasize practical examples considering certain types of system of systems which are described within the domain of network enabled capabilities. Network enabled capabilities (NEC) is an initiative by the United Kingdom which aims to enhance defence capability by exploitation of information [13]. The problem domain of NEC also involves non-military capabilities such as critical support functions for the society, e.g. healthcare and energy. The concept of network enabled capabilities focuses on the idea that a certain organization’s capability can gain a more effective performance by better exploitation of information and with the help of modern network technologies [5]. To that end, in this thesis we consider the user capability of observation and technologies supporting this capability in the application domain of NEC.

1.2 PROBLEM DESCRIPTION

In a general observation process, human cognitive agents perform an activity where assessment of some presented content is conducted. Moreover, this activity leads to the formation of an internal model, representing the assessed information [14]. The internal model is then combined with existing models and experiences, forming a base for the cognitive reasoning process. To be able to enhance the human cognitive agent’s information assessment capability, we advocate the use of visualization technology. Hence, in this thesis we emphasize the use of visualization technology as a supporting technology for human cognitive agents in their performance of refinement of system qualities.
As stated above, since we consider human cognitive agents to perform instrumentation activities on an open computational system with goals to restore deviating system behaviour, i.e. dealing with the consequences of affecting critical events, supporting methodology and technology is needed. In this thesis, we focus on the supporting technologies in means of a methodological tool for human cognitive agents. However, such a tool must fulfil a set of requirements put forward by both the technology of visualization and the capability of observation. Therefore, requirements put forward by the intended system operator(s) must also be taken into consideration. Hence, the problem lies in how to develop a tool which supply functions based on these technical as well as cognitive requirements.

1.3 APPROACH AND CONTRIBUTIONS

Our approach to the aforementioned problem starts with providing the reader with background information regarding a need for visualization of systemic qualities, thus enabling operators to observe such qualities when carrying out recovery processes to restore deviating system behaviour. However, to enable operators to perform such observation and instrumentation activities, we will identify the need for support from a model, method, and a set of techniques. We have selected a supportive model, method, and techniques and by covering and discussing literature reviews concerning these concepts we aim at creating a foundation for forthcoming argumentation and reasoning regarding a supportive tool. A specific tool which is stated to incorporate requirements from each of the discussed concepts of model, method, and techniques will be presented and we will be focusing our contributions on evaluating and providing recommendations regarding improvements for the specific tool. As a foundation for explanatory examples, we will give a thorough presentation of a specific open computational system, which will be used throughout this thesis.

Regarding the contributions, our evaluation concludes that the major flaws in the tool in question is the operator’s inability to select any other cognitive frustum as a starting point for exploration and instrumentation activities than the predefined. Moreover, the tool also presents the operator with complex tree structures which were built upon a good basic visualization concept of using fundamental geometric forms for symbolic encoding, but fail to facilitate operators when applied to actual cases where the level of complexity is high. Our recommendations in this case include, on the one hand, providing the operator with an active selection of frustum, thus enabling a plethora of new possibilities to observe other systemic qualities which may improve the operator’s level of situational awareness and affecting the overall instrumentation quality. On the other hand, we also stress the need for a dynamic filter which facilitates the operator in dealing with the current level of complexity.

1.4 OUTLINE

The outline of this thesis is as follows. Chapter 2, Background will go into further detail regarding the need of visualizing systemic qualities and need of supportive tools for system operators. Continuing,
the chapter will also present the Societies of computation laboratories (SOCLAB) and their readily available software, which this thesis is closely coupled with via the use of practical examples. Chapter 3, Phenomenon – TWOSOME provides such an example when presenting a real life explanatory example of an open computational system to the reader; the TWOSOME phenomenon. Following, chapter 4, Model – Open Computational Systems covers the model of open computational systems and how it supports isolation, adaptation, and validation of phenomena like TWOSOME. Chapter 5, Method – Online Engineering will give insight into the method of online engineering which provides online refinement of systemic qualities by presenting four supportive activities. With phenomenon, model, and method covered, we continue with chapter 6, Technique – Visualization, which presents and discuss visualization techniques, appropriate for visualizing systemic qualities, and human cognition, interwoven with practical examples from our example phenomenon TWOSOME. Chapter 7, Technology will present a visualization tool and we will argue how such a tool supports operators in observation and instrumentation activities. We will also discuss if the specific tool supplies functions conveying requirements put forward by the prior chapter and provide recommendations regarding possible improvements. Finally, chapter 8, Conclusions will summarise and conclude this thesis together with notes regarding future work. Chapter 9, References provides a bibliography of the thesis references and Appendix A provides a set of screenshots regarding different visualization concepts.
BACKGROUND

There is an emerging field in the design and development of complex systems, where systems are built upon components which in themselves are large scale systems – system of systems. These open complex systems are exposed to critical events, occurring in the systems execution environment. Moreover, these events may have negative effects on the system at hand, resulting in system behaviour diverging from intended. As mentioned in thesis introduction, we emphasize on human cognitive agents performing instrumentation and observation activities with the goal of restoring the resulting deviating system behaviour in conformance with the agent’s qualitative goals.

With this chapter, we aim to provide a background for the reader regarding the need for visualization of systemic qualities and the need for tools which support human cognitive agents with visualization technology in these observation and instrumentation activities. Furthermore, we will present related projects and their results; supportive tools for visualization of open computational systems within the problem domain of network enabled capabilities.

2.1 CRITICAL EVENTS

Consider an open complex system – comprised by a dynamic set of subcomponents or subsystems. The attributes of such a system, in conjunction with the term system of systems, is suggested to be independence, evolutionary nature, and emergent behaviour [3][4]. However, a system may also be interdependent – that is; each subsystem can generate events which may affect other systems in some shared execution environment. Exposed to these critical events, a subsystem being affected may deviate from its intended and/or projected behaviour. Consequently, the deviating subsystem behaviour also affects systemic properties – the overall system’s quality.

In the offline design and implementation phases of some information system it is possible to prepare the system for the occurrence of critical events, i.e., to take precautionary actions. However, to take into account all possible critical events, occurring in the system’s online execution phase is impossible [3]. A complementary way to address the matter of critical events is therefore to instrument the system in its online phase [5]. With a supporting methodology and associate technologies, which handles issues of an online nature, an operator agent may perform instrumentation activities on the system at hand with the goal of restoring system behaviour to an acceptable state at runtime.

We regard the instrumentation activity to be based on a basic control process, e.g. observe, reason, decide, and act. Consequently, with this particular control process in mind, observation of system behaviour becomes essential since perceived information regarding such behaviour will lie as a foundation for the operator’s forthcoming process activities [1]. The question which arises is that of how a system operator agent is supposed to gain understanding of system behaviour in a tractable manner.
In this thesis, we consider observation of system behaviour as a matter of human operators observing systemic qualities and their associate quantitative dimensions. Furthermore, we deem that how the systemic qualities are represented will affect the human operator’s control process as well as performance thereof. In doing so, a supporting technology is needed for the acquisition of systemic quantities and subsequent representation of systemic qualities. With the background presented above in mind, we continue our discussion with a focus on the term systemic qualities and delve into certain practical examples of such qualities.

2.2 SYSTEMIC QUALITIES

To describe systemic qualities, we emphasize non-functional requirements of software applications, which are expressed by Stroud as follows [7]:

“In order to be useful, an application must meet the requirements against which it was constructed. Furthermore, it is not sufficient for an application to simply perform some function, it must perform that function "well" in some sense. Thus, it is convenient to group application requirements into two categories, namely functional requirements, which are primarily concerned with the purpose of an application (i.e., what it does), and non-functional requirements, which are more concerned with its fitness for purpose (i.e., how well it does it).”

With respect to the aforementioned system of systems concept and qualities thereof, examples of such non-functional requirements, or non-functional system properties, include dependability, survivability, and adaptability [6]. In contrast to functional properties, non-functional system properties cannot be associated with some specific system component since they involve an aggregate behaviour of the whole system [6]. Hence, these non-functional properties are said to be systemic properties or qualities that cannot be added to a system in the form of an isolated system component alone. Instead, systemic qualities as perceived at runtime, related to fitness functions, are to be correlated with preestablished intentions originating from the offline phases of some particular information system – design and implementation rationales. In our current context, quality of service is another term which is used to describe systemic properties and is characterized by Sluman et al. as follows [8]:

“Quality of service is a general term that covers system performance, as opposed to system operation. A system will be built to perform some set of functions for its users. [...] the performance of each function will take time, require system resources and be subject to occasional system errors or failures: these and other similar features are performance or non-functional features of the system.”

Continuing, quality of service is concerned with the properties of system functions and information such as stability and availability [8]. Moreover, these systemic properties can be further decomposed into additional qualitative and/or quantitative dimensions.

To provide an example, quality of service in military operations, supported by technologies of network centric warfare, is depicted as comprised by the dimensions of information richness, information reach, and network robustness [9]. The concept of quality of service is also commonly used in the domain of telecommunication and networking. Examples of dimensions of systemic
properties and intended qualities, e.g. reliability, found in such domains may be delivery delay, delay variations, and data throughput capacity.

However, to give a practical example, we consider a specific systemic property and its dimensions, which are crucial to systems in a broad spectrum of application domains – availability.

2.3 Technological Availability

Regardless of the intended behaviour of some system, fundamental requirements are imposed on the system and its constitutive components in terms of being available. Hence, regarding systems where user interaction and technological support thereof is emphasized, high requirements are imposed on the systemic quality of availability.

To decompose the systemic quality of availability into its constitutive dimensions, a de facto standard is that of MTTF (mean time to failure) and MTTR (mean time to recovery). Using these quantitative dimensions, measurement of steady-state availability can be described as [1]:

\[ Availability = \frac{MTTF}{MTTF + MTTR} \]

In the offline phase of some system, infinitely large value of availability is one of the principle qualities that we seek to attain, be means of certain design decisions and related implementations of exception handling. However, in the online phase, a system is exposed to actual critical events and failures occurring in some physical execution environment; not only involving the critical events anticipated by some designer and/or programmer but the unanticipated ones as well. We consider that failures occurring in the online phase are inevitable and during this phase of some system we must therefore shift focus from the offline design goal of maximizing MTTF towards the online instrumentation goal of minimizing MTTR. When a failure occurs, the efficiency of system users as well as operators in performing corrective actions is directly related to the minimization of MTTR [1]. Furthermore, a sufficiently fast system recovery may even result in the cognitive agents being unaffected and therefore given the impression of high MTTF - high availability [1].

In the process of system recovery, it is important to quickly identify, decide, and act on what system constituents to restore and/or save and what part not to. Furthermore, in those cases where system complexity limits our capability of automating the necessary corrective actions, Fox recognizes the need for information access and visibility regarding the system at hand combined with supporting tools for human operators performing the recovery process in a manual fashion [1]. Hence, the need for supportive tools which makes systemic properties available for observation by human operators. Consequently, we deem that when human operators perform the activity of instrumentation in the recovery process – a type of control process, the operator will benefit from supportive tools which visualize systemic properties, e.g. the task of minimizing MTTR can be more efficiently dealt with. With systemic qualities and the process of system recovery in mind, we will now continue with the human operator’s cognitive process, with an emphasis on the activity of observation.
When operators carry out the type of recovery process previously mentioned, the resulting quality is based on the individual quality of all previous steps in the process. Moreover, in a recovery process, it is particularly important for the operator to reach a high level of situational awareness since it greatly affects the overall quality of the process’ final result.

Situational awareness includes perception of the elements of the surrounding in time and space, understanding their importance and projecting their status into the near future [10]. Hence, we consider situational awareness to incorporate the recovery process activities of observation and reason. Moreover, situational awareness is the most important aspect of decision making, which must therefore be taken in consideration when designing graphical system interfaces, with the aim of assisting the operator in gaining high situational understanding [11]. Analysis of accidents regarding some critical systems shows that loss of situation awareness has lead to operators’ making mistakes [10]. To assist an operator of a complex system in gaining a high level of situational awareness, we deem that supplied supportive tools must present systemic qualities with appropriate and efficient representation. Of course, operating complex systems also imposes demands on operators in terms of an intellectual nature, sensory abilities, and biological mechanisms of reciprocal connection [10] but these issues will not be discussed in this thesis.

Examples of technologies incorporated into tools which assist operators in information acquisition and distribution is; intelligent decision aiding, computer supported cooperative work, and visualization and mental models [10]. Since this thesis focuses on operator’s information acquisition via observation of systemic qualities, we emphasize the visualization and mental models technology.

The mental models, or visual mental models, is stored in the operator’s memory and is continuously as well as subconsciously updated by perception and interaction with the short-term memory. When events occur, information inconsistent with the current mental model is generated and a conscious process takes over and starts to consciously observe in order to analyze this lack of synchronization [10]. Furthermore, since the human operator’s capability to process large amounts of information simultaneously is limited and if large amounts of data are involved, the data must be summarized, filtered, and presented in such a way that it assists the operator. The actual visualization process takes place in the human operator’s mind; system information is represented to the operator as a visualization that reflects the state of the environment [11]. The aim is to engage the users' models of elements and relationships in a system so that they can take a visual form that can be tied to the surface layout of elements on the display [11]. Consequently, different graphical layouts may be used to suggest different types of system metaphors which support users in their cognitive process. Examples of possible visual mental models are models of physical shape, physical function, functional structure, abstract function, and functional meaning and objectives [10].

Within the area of visualization, there is a large set of visualization techniques which may be used to create and manipulate graphical representations from data sets. Based on the nature of the system and constituents at hand; some techniques will be more appropriate than others in enhancing and supporting the human operator in performing its delegated tasks. Moreover, there are also generic techniques which can be applied to many different applications; such as colour encoding techniques which may be applied in e.g. both office applications and supportive tools for performing instrumentation activities on complex systems.
As discussed in the background section, we consider the concept of system of systems to involve complex distributed systems which are exposed to critical events occurring in some shared execution environment. We deemed online instrumentation as part of a process of system recovery that aims to counteract some undesirable system behaviour. Moreover, we consider the recovery process to be performed by human cognitive agents, i.e. operators, and the term agent to follow the de facto structure of tightly coupled sensor and actuator capabilities – see Figure 2. However, alternative approaches could be emphasizing software cognitive agents, automation, and human-computer collaborative instrumentation.

Regarding the general concerns of critical events, we identified the importance for the operator to be able to quickly identify, decide and act on what system constituents to restore in the recovery process. To explicate this consideration, we regard it to correlate with Boyd’s OODA cycle and its following process activities; observe, orient, decide, and act [12].

The OODA cycle is described as an information strategy concept for information warfare, developed for military purposes, originally for air-to-air combat. However, the OODA cycle has also been successfully applied in business strategy and used to describe the decision-making process in command and control systems [12]. Boyd viewed all acting parties as systems, i.e. subsystems in a greater system, acting through a decision-making process, based on observation of the surrounding environment. Hence, observation is the first activity in the OODA cycle, as showed in Figure 1.

In a recovery process, operators first observe the system at hand, i.e. collect information regarding systemic behaviour and qualities thereof. As follows, operators orient to, or reason about, the collected information in order to increase situational awareness. Hence, a reasoning process inputs the collected information from the observation activity as it was perceived by the operator and interweaves the information with the operator’s mental models of experience and beliefs; resulting in establishment or refinement of situation awareness. Continuing, operators decide on how to affect the current system based on the observed and oriented situation, i.e. as a matter of the operators’ current situational awareness. The final action of the cycle is to act on results from the decision making process, i.e. operators perform online instrumentation of the system at hand. The resulting effect will of course be observable and, subsequently, the process is reiterated.

As mentioned before, we consider cognitive agents with sensor and actuator capabilities, which are vital to performing recovery processes. Hence, it is of importance that we support these capabilities, to enhance the process outcome. Since we consider operators – human cognitive agents – to performing these recovery processes, we can couple the OODA activities of observation and reason with the sensor capability and the activities of decision and act with the actuator capability – see Figure 2. Further, we will denote cognitive agents performing observation and reason activities as performing exploration activities whilst decision and act activities will be denoted as performing refinement activities.

By consistently cycling through the OODA process faster than opposing forces, i.e., the problem at hand, an increasing information advantage with each cycle is gained and thereby possibility to act, resulting in rendering the opposition’s decision process less fruitful. However, Boyd did not emphasize that the cycle should be performed as fast as possible but at a higher speed relative to the enemy. Boyd originally developed the OODA cycle for air to air combat where the pilots themselves performed all of the cycle’s activities. Hence, the time between observing a situation to initiating appropriate action (manoeuvring) is short and limited to the capabilities of the pilot. Applying the OODA cycle to e.g. land operations or organizations involving a large set of actors and operators implies a larger
cycle between gaining reports of observed information to actions being carried out based on issued orders. However, such larger OODA cycles can be decomposed into smaller continuous cycles, occurring simultaneously, within sections of an organization. An operator working within the smaller cycles either proceeds with local decision making or, based on the current operator’s capabilities and mental models, forwards an encountered situation to a higher decision level of the organization.

The OODA cycle can also be correlated with the basic low-level control loop of human response time, where response time corresponds to the time it takes for a human operator to perform a cycle of the OODA loop. Thus, response time is the time it takes for an operator to respond from the detection of a visual signal. However, the response often involves decisions to be made. The task of detecting a signal, making a choice, and reacting based on a decision has been thoroughly studied [25] and Kohlberg states that in an optimal state of readiness, an operator can react to a visual signal in about 130 ms [24]. Moreover, choice reaction time can be modelled as followed [2]:

$$Reaction\ time = a + b \log_2(C)$$

Where $a$ and $b$ are empirical determined constants and $C$ is the number of choices presented. Also, $\log_2(C)$ represent the amount of information processed by the human operator expressed in bits of information. Besides distracting factors such as visual noise, distinctness of the visual signal, and stimulus-response capability, Ware states that optimal conditions allow a response time of about 160ms / bit of information, plus time to set up the response [25]. Thus, by applying a set of fitting visualization techniques that strive towards providing optimal conditions, the operator will be facilitated in the detection of a visual signal – i.e. minimizing the response time.

Consequently, we consider that by providing operators with tools which facilitate the operators’ capabilities – e.g. increasing situation awareness – will result in faster cycling of the overall OODA process, and subsequently, increasing instrumentation quality – e.g. minimizing MTTR.

The observation activity is a crucial activity in the recovery process since all other steps use the gathered information as input, i.e. based on the operator’s situation awareness. Hence, we deem that the quality of representation of the observed systemic qualities will affect the operator’s situational awareness and thereby also affecting the result of the overall recovery process. Supportive technologies which deal with issues of representation of systemic qualities may be enabled to increase the operator’s situational awareness.

To present examples of supportive tools which enable observation of systemic qualities to operators via the technology of visualization, we consider appropriate research and development conducted at Blekinge institute of technology, i.e. a set of projects with resulting operator tools.
Figure 2. Showing a) the de facto agent capabilities of sensor and actuator and b) the OODA cycle coupled with these capabilities.

which support human cognitive agents with visualization technology in the process of observation and instrumentation of open computational systems. Consequently, we continue with an introduction to these enabling technologies.

2.6 ENABLING TECHNOLOGIES

At Blekinge institute of technology (BTH), the Societies of computation (SOC) research group conducts computer science research and theories are put into practice at the Societies of computation laboratories (SOCLAB). The engineering division of SOCLAB, which the author is a part of, has developed a range of tools and platforms as research enabling technologies and all software presented in this thesis is developed by SOCLAB for various projects. Moreover, screenshots displayed throughout the thesis is taken from both readily available software and non-available experimental software developed at SOCLAB. As mentioned, SOCLAB has carried out a number of projects and a large set of these were conducted within the application domain of network enabled capabilities and all with different aims and objectives such as connectivity, autonomy, and organization. Moreover, the set of enabling technologies developed at SOCLAB includes both tools and platforms but also systems which act as experimental applications for the tools and platforms. However, in this thesis we explicitly consider enabling technologies in the form of tools which supports operators in observation and instrumentation activities. Consequently, in the set of developed supportive tools, we have selected an applicable example; Distributed interaction system of complex entity relation networks - DISCERN.

The supportive tool of DISCERN was created with the intent of addressing requirements such as demonstration and comprehension of systemic qualities [5]; i.e. visualization of open computational systems. Thus, the development effort emphasized the notions of cognitive inspections and exploration of open computational systems [5]. The DISCERN development was a first hand experience in the development of a methodological tool for visualization of open computational systems and without any significant knowledge regarding visualization techniques or principles. Moreover, DISCERN will be further presented and reviewed in chapter 7 – Technology.

The open computational system Trustworthy and sustainable operations in marine environments (TWOSOME) was developed with the aim to implement a system, following the model of open

\[1\] For more information, visit: http://www.soclab.bth.se
computational systems and the method of online engineering. Furthermore, in this thesis, we have selected TWOSOME, which was developed in a project named Network enabled capabilities – generation 2 (NECG2), as an example of an open complex system in the application domain of network enabled capabilities. Concurrent development in the NECG2 project included e.g. the supportive tool of DISCERN.

2.7 CONCLUDING REMARKS

Considering complex systems, comprised by multiple subsystems, each subsystem is exposed critical events which may result in deviation from intended and/or projected behaviour, thus, affecting the current set of systemic qualities. To present an example of a specific systemic quality, we presented availability which is crucial to systems in a broad spectrum of application domains. In the offline phase of some system, infinitely large value of availability is one of the principle qualities that we seek to attain by means of certain design decisions and related implementations of exception handling. However, in the online phase, a system is exposed to actual unanticipated critical events occurring in some shared execution environment. We consider that failures occurring in the online phase are inevitable and during this phase we must therefore shift focus from the offline design goal of maximizing mean time to failure (MTTF) towards the online instrumentation goal of minimizing mean time to recovery (MTTR). When a failure occurs, the efficiency of system operators in performing corrective actions is directly related to the minimization of MTTR. Hence, there is a need for information access and visibility of systemic qualities regarding the system at hand. This need can be fulfilled by a supportive tool which enables operators to perform observation and instrumentation activities on systemic properties. Moreover, when performing such activities, it is important for operators to reach a high level of situational awareness since it greatly affects the overall quality of the recovery process’ final result. To assist an operator of a complex system in gaining a high level of situational awareness, we deem that supplied supportive tools must present systemic qualities with appropriate and efficient representation.

Regarding the general concerns of critical events, we identified the importance for the operator to be able to quickly identify, decide and act on what system constituents to restore in the recovery process. To explicate this consideration, we regard it to correlate with Boyd’s OODA cycle and its following process activities; observe, orient, decide, and act. The OODA cycle can also be correlated with the basic low-level control loop of human response time, where response time corresponds to the time it takes for a human operator to perform a cycle of the OODA loop. Thus, response time is the time it takes for an operator to respond from the detection of a visual signal. By applying a set of fitting visualization techniques that strive towards providing optimal conditions, the operator will be facilitated in the detection of a visual signal – i.e. minimizing the response time.

In the last section of this chapter, we considered enabling technologies in the form of tools which supports operators in observation and instrumentation activities. Furthermore, we briefly consider an open computational system, the TWOSOME phenomenon, which we will give a thorough presentation of in the following chapter.

For more information, visit: http://engineering.soelab.bth.se/projects/default.aspx
In this thesis we emphasize tools which support human cognitive agents in performing observation and in situ refinement of systemic qualities. Moreover, a set of such tools has been developed at SOCLAB throughout different projects. In parallel with the development of these enabling technologies, open computational systems has been developed with both the project aims in mind and as experimental applications for the tools and platforms involved. This chapter aims at giving the reader a introduction to the open computational system and network enabled capabilities demonstrator Trustworthy and sustainable operations in marine environments, TWOSOME [15][9] which was developed by the engineering division at SOCLAB in 2002. Moreover, we consider the presentation of TWOSOME to give the reader a real life example of an open computational system. One of the aims for the TWOSOME initiative was to validate the theoretical aspects of the model of open computational systems and the method of online engineering. Furthermore, by developing the system, i.e. applying theory to practice, valuable feedback regarding the methodology was gained.

The reminder of this chapter is structured as follows. We begin with describing TWOSOME’s actual physical execution environment, followed by the scenario and its constituents – e.g. scenario location and resources – and continue with presenting the actual system behaviour occurring during the scenario. Moreover, we will describe TWOSOME in the following sections from a visualization perspective with focus on each of the two system perspectives – i.e. the defence and attack domain.

3.1 EXECUTION ENVIRONMENT

The physical execution environment which hosts TWOSOME is comprised by a set of network-interlinked computers. Each computer is loaded with a specific software tool which enables the computer to be a part of a specific system as a resource entity. However, which specific role each computer node will represent is dynamically decided by settings in the tool. This network of nodes represents the actual system, composed by subsystems residing on each node. Furthermore, two additional computers are connected to this network – each with the predefined role as a visualization node. Each of the visualization nodes connect to the network as a normal node but both contain a different type of tool which enables human cognitive agents – operators – to observe a visual representation of the current system, its resources, and behaviour. To display this visual rendering of a virtual world, each of the visualization nodes are equipped with high-end graphics cards and each is connected to a projector – i.e. the visualization nodes displays are presented on adjacent dual projectors. However, each operator is only able to view its own projection of the selected phenomenon – see Figure 3 for an explanatory view. This physical setup enables two different perspectives of a specific phenomenon to be presented adjacent. Thus, enabling external
Figure 3. The physical execution environment and its constituents; network interlinked-computers, visualization nodes, projectors, operators, and observers.

observers, which are able to observe both projections simultaneously to observe multiple perspectives on the same phenomenon and different values and beliefs of the same set of attributes.

3.2 Scenario Constituents

For the TWOSOME system, we selected a scenario which is conducted in a marine environment with imaginary location and topology which is delimited to the physical bounding space of 1*1*1 nautical miles. The marine environment contains two landmasses that form a natural harbour where naval vessels can start and stop various operations. Furthermore, an island localized between the two landmasses creates two separate navigation routes used as passage for different marine vessels. The environment also involves an operations centre placed close to the natural harbour.

The following paragraphs describe the set of system resources of TWOSOME as articulated by Fredriksson and Gustavsson [15]. However, additional information regarding visual presentation of each resource has been added. Physical representation and proportions of all resources are correctly visualized in the system but the overall size of resources are manipulated to fit the scenario regarding focus, comprehension, and separation of concerns.

**Operations centre** – An operation centre gathers intelligence and is responsible for the overall safety of some physical environment. This resource continuously delegates orders, based upon newly acquired intelligence, to other entities under the operation centre’s control. The centre is visualized as a military building with communication abilities which are visually represented through clearly visible antennas and satellite dishes.

**Transporter** – In the present scenario, multipurpose vessels, e.g., a corvette class vessel, is used to transport a group of defenders from one location to another and subsequently delegate a defence order to the transported entities upon arrival. The vessel is visualized as a Swedish high-technology stealth corvette,
class Visby, which represents modern ship- and computer technology which can also be used in areas besides military applications.

Attacker – The role of an attacker is represented by a mine. The sole purpose of this mine is to detonate when it identifies certain acoustic and magnetic signatures – corresponding to targeted vessel types – in its surrounding environment. A modern mine is visualized as a simple cylindrical shape.

Defender – A defender is an autonomous vessel which is capable of emanating a set of acoustic and magnetic signatures with the aim of simulating different types of naval vessels and their properties. A team of defenders may be coordinated via self-organization to form complex signatures to be able to deceive and trigger an attacker. Moreover, a defender is visualized as a simple autonomous vessel with visual representations of its functionality.

Sensor – The purpose of sensors is to acquire information regarding their surrounding environment and forward gathered data to appropriate resources. In the TWOSOME scenario, the operations centre requires sensor data to gain knowledge of discovered threats and uses gathered data as input in a decision making process of how to counteract the current threat. The sensor is visualized as a floating buoy equipped with an underwater sonic sensor.

3.3 System Behaviour

The TWOSOME scenario begins with the creation of an attacker, an active mine, somewhere within the physical environment. Two different offshore sensors observe the creation and are thereby triggered to forward this information to the operations centre. Based on sensor information, the operations centre sends out a transport carrying a number of defenders, i.e. autonomous vessels emanating deceiving acoustic and magnetic signatures. Moreover, the transport travels out to a particular location, estimated by the operation centre, deploys the defenders and travels back to its origin, e.g. a nearby naval base.

Each of the defenders has the capability to navigate and cooperate with each other resulting in more or less complex and deceiving vessel signatures aimed for a potential attacker. Moreover, the complex and deceiving vessel signatures are very difficult to distinguish from those produced by that actual vessel class. To be able to keep the generated vessel signature correct, the defenders must be able to observe and instrument its current context. For example, if a particular defender’s service of generating some specific signature is downgraded or completely removed, the whole team must be reorganized. The team of defenders is capable of self-organization and is therefore capable of reconfiguring and of reusing the currently available resources and services to be able to keep the deceiving vessel signature. Dependent of the attacker’s configuration, the mine will detonate if a predefined set of conditions is fulfilled. Thus, detonation of the mine can be dependent on a combination of factors such as: ship type i.e. vessel signature, number of ships passed, etc.

The team of defenders coordinates respectively to the common goal – to deceive a potential attacker, i.e. the mine, which will lead to detonation and thereby removing the threat. To be able to
reach the goal, the team of defenders agrees over which type of vessel signatures to generate and starts
to repeatedly sweep an area, selected by the operations centre. When the goal is fulfilled, i.e. the mine
has detonated, the transport travels to the selected pickup point to collect the defenders and return to
its origin.

As described earlier, different events will occur during the scenario – e.g. a potential threat has
been detected or a resource has changed location. Thus, the occurring system behaviour generates
events, of different significance, which require supporting method and technology for presenting these
events in such way that a cognitive operator may observe and take notice of the events. Examples of
more complex events occurring during the TWOSOME scenario range from e.g. resources being
withdrawn or induced into the system to system state changes such as a mine detonation.

3.4 VISUALIZATION

As stated above, the TWOSOME scenario world is visualised as a marine environment of a specific
area. Furthermore, TWOSOME has two different types of perspectives of the same environment; the
defence domain and the attack domain. In the coming section we will describe how an operator may
observe, interact, and explore the two perspectives.

**Defence domain** - When observing the system from a defence perspective, one will gain an overview
over the selected and delimited environment. Moreover, scale indicators positioned in the outer
boundaries of the environment allows the user to gain a sense of proportions. When considering the
overall view of this environment, it is obvious that it was constructed to mimic the real world; e.g.
using natural colours. However, simplification of some details was needed and visual artefacts were set
out of proportions to ease comprehension. Furthermore, the surrounding skybox is in a sky blue
colour with a superimposed grid to enable operators to gain depth recollection. Regarding navigation,
the operator is enabled via move, pitch, and yaw commands to freely navigate and explore the virtual
world. As described earlier, the defence aspect of TWOSOME allows the user to gain an overview of
the environment but in this specific scenario it also acts as the main interaction point of the system.
Hence, the defence perspective allows the user to activate scenario specific actions and interact with
the system, such as deployment of a hostile mine or manipulating the selected area for mine sweep.
From a visualization viewpoint, interaction with the system is performed via an orthographical
projected text based command console. Furthermore, the semi-transparent command console is
activated by affecting the camera and place the view centred pointer over the desired resource’s
communication port and pressing the enter key. Thus, displaying the command console and shifting
mouse and keyboard focus from the virtual world over to the console. With a specified set of
commands one interacts with the selected system resource, i.e. a computer node, in real time and may
relay orders such as to commence a mine sweep. Moreover, a listing and description of the specified
set of commands can be accessed via the “help” command and removal of the console is performed with
the “exit” command.

**Attack domain** – Compared with the defence perspective; the attack aspect of TWOSOME is
visualized with the exact same environment but the view frustum is centred accordingly to the mine’s
underwater location, resulting in a view as seen from the mine. Consequently, the scaling of the
environment is greatly enlarged compared to the defence perspective to generate a more accurate view
of the mine’s possible sensor range from its static position. The view frustum is in a fixed position with
only pitch and yaw navigation commands available, representing the mine’s inability to move. To ease
comprehension of which perspective of TWOSOME is observed, the colour scheme has been replaced with an overall spectrum of red. Hence, when both defence and attack perspective is visualized on adjacent dual projectors, the two distinct opposing colour schemes aid in distinguishing which perspective is observed on which screen.

During the scenario, as the autonomous vessels navigate through the mine’s sensor range, the attack perspective visualizes the result of their task of emanating false vessel signatures. Hence, if the defenders are successful in deceiving the mine, the attack perspective will show a ship hull representing what the mine’s sensors perceive and interpret – not the actual autonomous vessels themselves.

3.5 CONCLUDING REMARKS

The TWOSOME system is a conceptual demonstrator within the application domain of network enabled capabilities and a practical example of an open computational system. As a part of the scenario, the demonstrator shows self-organizing and autonomous vessels acting on orders to generate a complex vessel signature over a defined area – i.e. commencing a mine sweep. Moreover, the scenario takes place in a virtual marine environment, with an imaginary location and topology, which is rendered by a graphics engine.

From a visualization viewpoint, TWOSOME enables two distinct views into one system – i.e. allowing an operator to choose either the defence perspective or the attack perspective as a starting point for exploration activities. Each perspective displays its own values and beliefs of the same attributes of the current system, thus, enabling external observers, which view both perspectives simultaneously, to gain understanding of certain complex systemic qualities, e.g. the level of success in the autonomous defenders’ mission (Figure 2).

The TWOSOME system is an open complex system with component interdependence and emergent behaviours. Thus, to structure observations and dependencies of such a system one needs a supporting model and in the next chapter we will continue with presenting such a model – open computational systems.
Figure 4. The defense perspective of TWOSOME showing a) an environmental overview and b) the scenario naval route with three defenders commencing a mine sweep and a static acoustic sensor buoy.

Figure 5. The attack perspective of TWOSOME showing c) the mine's view of three autonomous defenders, unsuccessful in detecting the mine and d) a medium class ship, detection and the ensuing naval route with the defenses commencing another environmental survey.
The concept of system of systems is generally recognized as large scale systems, built upon interdependent and self contained components. Attributes which distinguish the system of systems concept from large monolithic systems is e.g. their component independence, their evolutionary nature, and emergent behaviours [3]. Based on these attributes, we have chosen the model of open computational systems, which convey the in situ constituents, dependencies, and interactions of such complex distributed systems [5].

In TWOSOME, a cognitive operator performed observation and instrumentation activities on the specific phenomenon at hand. Furthermore, the operator – i.e. a human cognitive agent – was supported by both static and dynamical visual artefacts, presented in an observable manner. Thus, the cognitive agent was supported via both a model for observation and isolation, and a technology used for representing the artefacts. Moreover, to be able to handle events occurring in TWOSOME, with the aim of sustaining system behaviour, the operator was supported with system adaptation and validation. Consequently, in this chapter we will give the reader an insight into the model of open computational systems and how it supports observation, isolation, and adaptation of phenomena via its four abstract layers, i.e. environment, fabric, system, and domain.

With the open computational systems TWOSOME in mind, we continue with the following outline for this chapter. First, we will present further details regarding the layered model of open computational systems. Second, each of the four layers will be presented from both model and visualization viewpoints, i.e. the layers of environment, fabric, system, and domain.

4.1 OPEN COMPUTATIONAL SYSTEMS

In the domain of open computational systems, the main origin of complexity is of the notion of openness. Furthermore, systems which are defined as open can be described as systems which are continuously in focus of that their components, dependencies, and interactions may evolve over time. A system evolution can be the result of some intervention by cognitive agents or unanticipated events. Moreover, the intervention can be the result of the capability of adaptation for some cognitive agents, i.e. the possibility to instrument the selected phenomenon in runtime towards a predefined goal, which is one way of dealing with the issue of complexity. Since the cognitive agents can be of both human and computational nature, there exists a need of observation and isolation of the phenomenon at hand. Consequently, there is also a need of an effective model which supports the notion of cognitive isolation, adaptation, and validation – the model of open computational systems.

A main problem with systems embedded in some environment is their property of being open to change. These changes may take place on different levels of abstraction; i.e. on a structural level, procedural level, and on a level of patterns [5]. Hence, there are three levels of abstraction that could
be considered in observation of open computational systems. Change and dynamics in the structural level may include situations where entities are induced or withdrawn from some system. Moreover, regarding changes at a procedural level, dynamics are result of state changes in some system. The abstraction level of patterns can be used to describe system dynamics deduced by some cognitive agent through observation. However, structures and processes can be observed by means of appropriate instruments whereas the abstract level of patterns may be thought of as the private result of some cognitive agent’s observation, including the observer’s experience. With this in mind, complex situations emerge when multiple cognitive agents explore and refine a specific complex phenomenon, thus creating a need for an applicable model which provides for grounding behaviour semantics. The model of open computational systems aims at providing primitive concepts and dependencies which can be utilized in observation and instrumentation of a specific phenomenon, and shared by all agents involved. Thus, the layered perspective of open computational systems aims to depict the continuous evolution of structures, patterns, and processes.

From a visualization viewpoint, we consider that providing primitive concepts and dependencies by using visual artefacts may support cognitive agents in their observation and instrumentation activities. Moreover, these visual artefacts must be provided in such way that they are equally accessible by all agents involved. Hence by providing a virtual three dimensional universe in real time, cognitive agents may share the same graphical artefacts and behavioural semantics, residing in the virtual universe. With the support of shared visual artefacts and the model of open computational systems, multiple cognitive agents may explore and refine a specific complex phenomenon.

4.2 Environment

The first layer of the open computational systems’ model is the layer of environment which is represented as a physical environment. Cognitive agents, with the goal of exploration and refinement of a specific phenomenon, may identify a physical environment and its constituencies as a matter of the context that helps understanding the behaviour of the system at hand. Moreover, it is within the physical environment, i.e. within a spatial volume, that agents can interact by means of some specific medium and each agent in the particular system resides at some specific spatial location in the physical environment.

As stated, identifying a physical environment may help in understanding system behaviour but will also help in isolating the specific phenomenon. A delimitation of the physical environment may be performed by a distinct cognitive agent who creates a cognitive bounding space, isolating the physical phenomenon. Fredriksson introduces the concept of frustum as the defined cognitive bounding space, depicted as a solid volume and cut by the four levels of the open computational systems model, i.e. environment, fabric, system, and domain [5] – as shown in Figure 6. The concept of frustum enables the possibility to model the specific open computational system involved. Moreover, cognitive agents can use the concept of frustum as a tool for filtering in observation and articulation of online phenomena.

To visually present a whole virtual universe containing a range of phenomena, may not be feasible. However, by supporting cognitive agents in performing an isolation of a specific phenomenon of interest by delimiting the virtual universe to a specific virtual world; enables the possibility to visually represent a specific physical environment and its constituents. This creation of a cognitive bounding space may be performed by supporting cognitive agents with a set of virtual worlds, where
the active selection performed by the agent denotes the starting point of its exploration activities (Figure 7). In TWOSOME, the operator makes an active selection when choosing to explore TWOSOME’s physical environment – i.e. the virtual world of TWOSOME - and in the selection of which of the two available perspectives to explore. With a virtual world identified, there are a set of different approaches for how to visually represent the selected physical environment, depending on the specific phenomenon of interest, ranging from a realistic representation of an existing physical environment to an abstract representation of a non-existing environment. Furthermore, appropriate visualization techniques are discussed in chapter 6, Technique – Visualization.

4.3 FABRIC

When cognitive agents perform activities such as exploration and refinement, they take part in communications with other agents. Hence, the agents performing these activities require the capability of communication and require support for mediation, processing, and distribution of information. The fabric layer supports these basic and essential capabilities and provides an infrastructure which enables agents to observe and interact with both other local agents as well as remote agents.

A set of cognitive agents which all have performed similar creation of cognitive bounding spaces and thereby reside in the same virtual world, observe the same visual artefacts. The fabric layer ensures that the actual information presented in the virtual world is available to both local and remote agents, creating a seamless information flow between agents. However, the fabric layer can also be represented in the virtual world as a physical network structure, containing both nodes and communication links (Figure 8c). By enabling visual artefacts which represent the constituents of the fabric layer, cognitive agents may observe the actual network information flow and attributes thereof, e.g. throughput, status, and capacities.
4.4 **SYSTEM**

The system frustum is concerned with the actual system on which cognitive agents perform activities such as exploration and refinement. Thus, actual system behaviour takes place in this frustum. Fredriksson defines the notion of system as a population of partially unknown and temporally ordered set of physical entities that interact with each other [5]. Interactions and the number of entities present will be of an unanticipated nature due to the openness of the system, so to enable communication between these physical entities, one is required to be able to identify a unique and distinct entity and enable an interface – ports – for transportation of information between the entities.

Visualizing actual system behaviour is dependent on the current systemic qualities and system attributes and may therefore be difficult to represent; however, the existence of physical entities can be presented to an observing cognitive agent by representing a set of physical entities with a corresponding set of virtual artefacts. Moreover, by performing measurement activities on quantitative systemic qualities, a set of results may be presented in such way that a cognitive agent gains understanding regarding the current system behaviour.

4.5 **DOMAIN**

The domain layer is by far the most abstract layer of the model of open computational systems and the main intent is to provide support for cognitive exploration and refinements of some phenomenon at hand. Furthermore, the primary benefit of the domain layer is it’s supporting cognitive constructs which depicts both summative and constitutive characteristics of some phenomenon at hand. These constructs are used in the fundamental activities of exploration and refinement of open computational systems. Moreover, from a cognitive agents perspective, these cognitive constructs are a set of general concepts and relations i.e. domains, concepts, ports, dimensions, and dependencies (Figure 8d). Fredriksson presents the following general constructs and their connections, at the domain layer [5].

**Domain** – The cognitive construct of a domain acts as a starting point for elicitation of relevant concepts involved. Moreover, a particular domain can only be made explicit if at least one cognitive agent considers it relevant. Each domain can connect with internal and external concepts, where the external concepts are connected by means of dimensions and associated with some other domain.

**Concept** – An arbitrary amount of concepts may be connected to each domain and each concept depicts the most relevant features of the particular domain, according to a set of cognitive agents. Besides being connected as internal concepts to a particular domain, a concept can also be connected as an external concept to some other domain via the cognitive relation of dimensions. Both concepts in general and dimensions are products of a set of cognitive agents’ development of public harmony in grounded semantics.

**Manifestation** – Each concept can be connected to multiple manifestations – instances of concepts e.g. the concept of capability can have a manifestation named propulsion. Furthermore, each manifestation can have relations with each other – dependencies. However, a more concrete notion of this construct is that of a port, which will be used in this thesis.
By presenting a cognitive agent with a set of these cognitive constructs, i.e. domains, concepts, ports, dimensions, and dependencies, in a virtual world, the agent may observe existing system constructs and their relations to other constructs in the specific phenomenon at hand. However, to visually represent each of the very abstract constructs in an effective and qualitative manner may be complicated and is dependent on e.g. specific system’s context. To give a practical example of domain layer visualization, we refer the reader to chapter 7 - Technology where the DISCERN domain layer visualization is presented in detail.

4.6 CONCLUDING REMARKS

In the prior chapter, we presented a practical example of an open complex system; TWOSOME. Moreover, complexity in such systems mainly originates from the notion of openness and the need to keep such systems available for exploration and refinement activities. One way to deal with these issues of complexity is to enable and support cognitive agents in observation and instrumentation activities. In this thesis we consider human cognitive agents performing such activities which require an effective model with supports notions of observation, isolation, and adaptation.

The model of open computational systems aims at providing primitive concepts and dependencies which can be utilized in observation and instrumentation activities of a specific isolated phenomenon. Furthermore, by presenting the possibility for human cognitive agents to isolate a specific phenomenon via an active selection of a three dimensional virtual world, multiple cognitive agents may explore and refine a specific complex phenomenon using shared visual artefacts.

Attributes of the type of complex systems we consider in this thesis, i.e. system of systems, are those of component independence, evolutionary nature, and emergent behaviour. Also, we consider such open systems to be interdependent – i.e. each subsystem may generate events which may affect other systems in a shared execution environment. With the support of the method of open computational systems we enable agents to isolate and observe such systems and resulting system behaviour. However, since occurring critical events may result in deviating system behaviour, one requires a supportive method for addressing such issues. In this thesis, we consider the method of online engineering which supports cognitive agents’ in situ – i.e. online – instrumentation activities with the aim of sustaining system behaviour. Continuing, we will in the next chapter present the method of online engineering.
Figure 7. Showing a) isolation of a phenomenon via the active selection of a virtual world and b) the results on isolated physical environment and its constituents.

Figure 8. c) Shows a visual representation of a conceptual physical network structure containing both nodes and communication links. d) Shows an example of a visualization of both cognitive and physical constructs.
As presented earlier, complex distributed systems such as open computational systems will to a great extent be affected by other systems and events occurring in their surrounding environment – i.e. affecting critical events. Consequently, the effect may be that the specific system deviates from its intended system behaviour. Moreover, addressing these issues in the design phase of open complex systems is extremely difficult [3]. A complementary way to address these issues is to enable real time – online – instrumentation of the system at hand with aim of compensating for the effects caused by critical events and thereby sustaining the intended system behaviour [5]. Consequently, in this thesis we have chosen the supportive methodology of online engineering to enable this in situ refinement of system qualities.

The following sections aim at presenting the methodology and the four activities of articulation, construction, observation, and instrumentation. Furthermore, since we deem that human cognitive agents will be performing these online activities, we will also consider visualization aspects of these activities – where applicable.

5.1 ONLINE ENGINEERING

One way to address complex phenomena such as the previously presented TWOSOME phenomenon is to apply the method of online engineering which deals with issues of critical events by enabling online instrumentation. Moreover, the method uses an iterative approach based on four steps to support a cognitive agent in refinement activities. First, a cognitive agent needs to articulate its individual experiences of the phenomenon at hand and map them onto cognitive structures. Second, the articulated structures must be constructed and made accessible online to support adaptation of the phenomenon at hand. Moreover, resulting constructs should be structured following the model of open computational systems – i.e. in the form of an online accessible model of system behaviour. Subsequently, the third step involves cognitive inspection – observation – of the online accessible and structured constructs. This activity of observation is performed by cognitive agents, resulting in a gain of experiences and knowledge regarding the specific phenomenon at hand. Moreover, since we consider human cognitive agents – operators – to be performing in situ refinement, the observation activity becomes an even more crucial activity since the gathered information is used as input for the process of deciding how to affect the phenomenon at hand. Consequently, the final step is to carry out the decided actions – instrument – the specific phenomenon accordingly with the operators qualitative goals, e.g. sustain system behaviour.

In a prior chapter, we discussed recovery processes based on affecting critical events and discussed the OODA cycle from an operator perspective. To correlate this operator perspective with the method of online engineering, we decompose the online activity of observation to comprise the
cognitive OODA operator steps of observe and reason. Thus, considering the online activity of instrumentation to correlate to the OODA activities of decide and act. Moreover, in relation to the agent sensor and actuator capabilities we consider the online activity of observation to involve sensor capability whilst the instrumentation activity involves the actuator capability (Figure 9).

### 5.2 ARTICULATION

Cognitive agents, both software and human, may perform exploration and refinement of a phenomenon at hand. As a point of departure for each individual agent, when performing these activities, is an articulated frustum. Prior, a cognitive agent articulates its individual experiences of the phenomenon, based on the agent’s intentions and type, and maps them onto the conceptual structure of frustums. Furthermore, by articulating individual aspects, based on each agent’s individual experiences, the cognitive agent increases the probability of a more effective outcome when dealing with a certain phenomenon [5].

To give examples of different types of articulation, Fredriksson describes an exploratory agent, refinement agent, and an operator agent which focuses on different aspects of the current phenomenon. An exploratory agent would focus on identifying the current most dominant conceptual structures in its articulation activities [5]. However, a refinement agent would rather focus on identifying physical constructs which convey the current system behaviour [5]. Furthermore, an operator agent would emphasize the need to identify both conceptual and physical constructs which reflects the intended system behaviour [5]. As stated before, in this thesis we focus on operator agents – i.e. human cognitive operators – and their exploration and refinement activities.

Regarding the TWOSOME phenomenon, the exploring operator performed identification of both conceptual and physical constructs via the support of observation of visual artefacts, residing in a virtual world. Moreover, these identifications will be performed at each structural level of the model, i.e. open computational systems’ domain, fabric, system, and environment [5].
When constructing artificial systems, a traditional approach assumes a one-to-one mapping between design and model of the system's behaviour. Furthermore, the assumption is based on that the system at hand is of a closed nature from a stimuli point of view and is never subjected to any unanticipated stimuli from the surrounding environment. However, if the system at hand can be influenced by unexpected stimuli from the surrounding environment; the system is considered to be of an open nature – which is the case with open computational systems. Consequently, the design of a system which is of an open nature contains an incomplete set of the possible states which the system will experience during its lifetime. Instead of applying new design patterns and trying to anticipate future system states, Fredriksson states that it is imperative that we understand the requirements involved in transforming a design into system behaviour and present the following solution regarding constructing an open system [5]:

“The most important aspect of transforming designs into systems is not primarily that the original design is preserved throughout construction of the target system, but rather that it is preserved and made accessible for adaptation during the system’s in situ evolution, in the form of an online accessible model of behavior”

In the articulation phase, all cognitive agents articulate their individual experiences of the phenomenon at hand and map those onto frustums whose characteristic features can be described in terms of models and designs. Following the articulation phase is the construction phase where the involved agents take the frustum constructs identified during articulation as input to the process of turning their individual experiences and expectations into concrete manifestations of system behaviour at each level of the open computational systems model. Moreover, the possibility of online access to the original design is given by instantiating the design in its concrete form at the domain level of the system at hand. Consequently, if observation of the domain level is enabled, observation of the original design is possible, which may be used in restoring deviating system behaviour.

After a cognitive agent has articulated its individual experiences onto a set of frustums, corresponding to the structure of the open computational systems model, the frustums are turned into a concrete set of system behaviour. Thus, setting the specific phenomenon at hand to an online state and thereby making the cognitive constructs accessible online. When a phenomenon is in an online state, cognitive agents are able to perform cognitive inspection – i.e. observation. Hence, cognitive agents may observe the online frustums with the aim of gathering information regarding the phenomenon’s systemic behaviour and qualities thereof.

To be able to perform this observation, cognitive agents need the capability to observe their surrounding environment to, first, gain understanding of the current context which they are situated in and, second, in order to execute the most appropriate actions towards fulfilling their individual qualitative goals.
In this thesis, we consider human cognitive agents with the capability of observation which performs observation and instrumentation activities, with the aim of sustaining system behaviour. Moreover, as stated in the online engineering introduction section, since we consider operators, the online activity of observation can be decomposed into comprising the OODA steps of observation and reason. Hence, first, the operator collects information regarding systemic qualities and, second, the operator reasons about the collected information. The operator carries out a reasoning process where information as perceived by the operator act as input and is interwoven with the operator’s internal mental models of experiences and beliefs; resulting in an establishment or refinement of situational awareness.

The operator’s level of situational awareness is a most important aspect of the following activities, since it greatly affects the overall quality of the whole refinement activity. Thus, a cognitive agent uses the information gathered by observation of online accessible constructs as input in the following online activity of instrumentation.

5.5 Instrumentation

We emphasize systems which are of an open nature, thus exposed to critical events occurring in some shared execution environment, resulting in possible deviating system behaviour. As stated before, we consider addressing these issues by enabling cognitive agents to iterative instrument the system at hand in its online phase towards the agent’s qualitative goals, e.g. sustaining system behaviour. Moreover, since we consider human cognitive agents to be performing this in situ refinement we can decompose the online activity of instrumentation to comprise the OODA activities of decision and act. Thus, the cognitive agent’s decision process uses the situational awareness gained during the observation activity as input and, consequently, carries out its decision – i.e. instrument the phenomenon accordingly.

When performing instrumentation activities on an open computational system, one wants to gain understanding of how the current refinements affect the overall quality of the intended system behaviour, from an online perspective. Hence, to instrument an open computational system in conformance with the method of online engineering, one performs the online activities of articulation, construction, and observation all over again [5]. Also, in this second iteration, some frustum constructs already exist in an online state and may be changed, removed, or new constructs may be created by cognitive agents.

This iterative approach to in situ refinement of open computational systems aim at establishing stable system behaviour, thus, facilitating real time observation and instrumentation of systemic qualities is of essence [5].

5.6 Concluding Remarks

When dealing with open computational systems; one of the major challenges is handling critical events in the systems execution environment, i.e. events that were not planned for when designing,
implementing, and deploying the system. Instead of applying new design patterns which would improve handling of these unanticipated events, Fredriksson advocates the need for a supporting model and method which helps in dealing with their in situ impact [5]. Consequently, Fredriksson suggests the use of the method of online engineering, which advocates articulation, construction, observation, and instrumentation of open and online phenomena in conformance with the model of open computational systems.

The method of online engineering emphasizes an iterative approach to support cognitive agents in their in situ refinement process of a specific phenomenon. With articulated and constructed frustums accessible online, cognitive agents may perform observation with the aim of gathering information regarding the phenomenon’s systemic behaviour and qualities thereof. Moreover, with the gathered information as input, cognitive agents may perform online instrumentation of the specific phenomenon accordingly with their qualitative goals. Continuing, to gain understanding of how the current refinements affect the overall quality of the intended system behaviour, cognitive agents may perform the online activities of articulation, construction, and observation all over again – i.e. an iterative in situ refinement of the specific phenomenon at hand.

The method of online engineering is composed by four main activities, however, to further decompose the actual transition from an offline state phenomenon towards an online state, one may introduce two additional steps after construction and just before observation (Figure 10). The offline activity of configuration would comprise the actual setup to enable the following execution activity. Moreover, when the specific phenomenon has been turned into an online state, the following activities of execution, observation, and instrumentation is considered to be carried out in parallel, compared to the sequential flow of the offline activities. However, in this thesis, we emphasize systems which are executed and therefore in an online state, thus, focusing on observation and instrumentation activities – i.e. exploration and refinement of a specific phenomenon.

In TWOSOME, the cognitive operator was supported with the model of open computational systems for isolation and adaptation, the method of online engineering for supporting exploration and refinement activities, and a technology which enabled presentation and thereby observation of systemic qualities. Hence, in the next chapter we will present supporting techniques which transforms the symbolic into the geometric – visualization.
In prior chapters, we have presented our viewpoint on open complex systems and human cognitive agents. Furthermore, a specific phenomenon, TWOSOME, has been presented as a real life example of the type of systems and needs we consider. Following, we presented a model which supports cognitive agents in isolation and adaptation activities of such phenomenon and a method which supports exploration and instrumentation activities. A major concern of the method of online engineering is to enable cognitive agents to instrument a phenomenon based on the agent’s observations, i.e. its level of situational awareness; in order to deal with issues regarding critical events. Hence, to further support the agent’s situational awareness we proposed the use of a set of techniques that facilitates human cognitive agents in observation and, consequently, instrumentation activities.

In this chapter, we aim at providing a brief introduction to the area of visualization by means of discussing a general rationale – human cognition and information processing. However, the main focus for this chapter lies on presenting in what way a set of visualization techniques may facilitate human operators, in such ways that their response time to a visual signal is minimized and that their capability of instrumentation thereby is enhanced.

The outline for this chapter is as follows. First, the general rationale of visualization is discussed. Second, we will discuss human cognition and internal models with the aim of clarifying the need of technological support of good externalization of data, i.e. visualization. Finally, we will present visualization techniques supporting operators in their essential activities of data exploration as well as data representation. The previously introduced phenomenon of TWOSOME will be used as an explanatory example throughout the entire chapter.

6.1 RATIONALE OF VISUALIZATION

As stated in chapter 2 – Background, we consider that visualization techniques may support cognitive agents in their observation activities. However, the actual visualization activity in itself is a cognitive activity engaged by human beings where an internal model is interwove with external perceptions and manipulations. The potential value of this activity is to gain insight and understanding regarding information which is not evident from raw data concerning some physical phenomenon. Moreover, these insights may be geared towards discovery, decision making, and explanation [16], and as Tufte stated; "often the most effective way to describe, explore, and summarize a set of numbers – even a very large set – is to look at pictures of those numbers."[19]

A landmark publication in 1987 defines visualization as a method of computation; transforming the symbolic into the geometric which enables observation of simulations and computations [20]. Thus, visualization enables insight into scientific methods via supporting visual methods [20]. The use of
visual methods or graphical support for research has a long history but our modern advance in computer technology enables a plethora of new possibilities, e.g. supportive tools that may amplify cognition via the use of a set of visualization techniques. Such tools should use visualization techniques to offer support which strives towards providing as optimal conditions as possible for an operator in detecting visual signals, i.e. minimizing human response time.

The area of visualization is often divided into two main branches; scientific visualization and information visualization. Moreover, these separate sub branches were defined at the first IEEE Symposium on Information Visualization with the statement that [21]:

“Information visualization is a process of transforming data and information that are not inherently spatial into visual form, allowing the user to observe and understand the information. This is in contrast with scientific visualization, which frequently focuses on spatial data generated by scientific processes”

The actual defining of the two sub-branches helped the development of both scientific and information visualization and opened up for new fields beyond e.g. high performance computing and computational modelling [18]. Hence, the application fields for modern visualization include, but are not limited to, medical visualization, flow visualization, geographic information systems visualization, and architectural visualization [16]. In this thesis, we emphasize information visualization due to the aforementioned emphasis on systemic qualities, e.g. availability, and will therefore refer to information visualization when using the term visualization.

As stated in the beginning of this section, visualization is a cognitive activity performed by human beings, so to understand the actual need and proper application of visualization we need to understand why the cognitive agents require visualization support. Therefore, in the next section of this chapter, we introduce a discussion regarding human cognition and perception.

6.2 Human perception and information processing

Above, we stated that visualization techniques should support human cognitive agents in their observation activities; this is accomplished via the use of visual artefacts – external visual artefacts from a human point of view. The human visual system is capable of rapidly absorbing and interpreting visual information and about 70% of all information that a person absorbs is acquired through the visual system [22]. Thus, the use of such visual artefacts is of importance and external artefacts which aid or enhance human cognition already exist all around us. For example, with the support of the external artefacts of pen and paper, the time for performing a multiplication is reduced even though mental multiplication is not in itself very difficult; storing partial results is [16]. Hence, supporting the user with external storage of partial results in a visual form will facilitate the working memory and improve the overall process. The use of tools or computational devices that provide visual artefacts may facilitate mental abilities. However, by presenting the result from the usage of such tool, e.g. a million calculation results may not be easily comprehended by humans. By applying suitable visualization techniques on the result set, an observer may rapidly gain insight of the overall result. Thus, in this thesis we consider the term visualization to be about exploiting the dynamic, interactive, and inexpensive medium of computer graphics to device new external tools for enhancing human cognitive abilities [16].
To further understand how to aid cognitive abilities, we must first describe cognition. We consider cognition to be the mental process of knowing, including aspects such as awareness, perception, reasoning, language, memory and judgment [26]. Hence, cognition includes all of the human brain’s mental input and output, e.g. from basic activities such as using language and arithmetic to complex decisions, creativity, and to understand another person’s perspective [26]. The human memory is a part of cognition and is much more than a passive storage system; memory is a set of active processes that encode information to be stored and associated with possible existing memory. Cognition assists memory by identifying relevant information to be remembered. Hence, perception performed by an operator is transformed into a representation in memory but without connected meaning. The representation is then subjected in to the process of cognition where an internal model and understanding is created in memory. However, the area of perception and cognition is very complex and not yet fully understood [14]. In the following two subsections, we discuss two concepts which are closely coupled with cognition – externalization and browsing.

Externalization – By performing perception and consequently a process of cognition, an operator gains data created from such activities which are then used in the formation of internal mental models. These models are the mental representation of information such as artefacts, situations, and connections to the visually presented data. Moreover, by interpreting the internal models, further insight may be gained which might lead to e.g. an informed decision [14]. When we are familiar with some data, the internal model is said to be robust and well established. However, the internal model is not a single homogenous entity; it can be incomplete. Nevertheless, the incompleteness may be of value in some situations. There may exist a resistance to change the internal model when the externalization changes and when the externalized data contradicts the internal model, the user tries to simultaneously understand both views which results in construction of a new internal model for the externalized data. Electronically displayed data have the possibility to easily change the data externalization, which may cause problems if an internal model is continuously demolished, the operator may instead loose the gained situational awareness. However, in some cases the externalization must be changed to enable the operator to gain further insight in some data. Spence et al. presents an example where a trade-off in-between changing externalization and maintaining existing internal models is accomplished by applying an inertial movement to a cone-tree representation of hierarchical data when changing the externalization, thus gaining spatial awareness and therefore not demolishing the existing internal model [14].

Browsing – As stated above, the combination of perception and cognition leads toward the forming of internal models. However, we emphasize this process as the term browsing – the activity of registration or elicitation of content which leads to the formation of an internal model [14]. Browsing is sometimes implied to be an unstructured and random activity. However, performing the browsing activity in an unstructured manner is only one of many possible browsing strategies; other strategies may include planned browsing and opportunistic browsing [14]. When formulating a browsing strategy there are two main determinants to consider; cognitive and perceptual determinants [17]. The cognitive determinant is based on interpretations made or new ideas but is not directly correlated to what is displayed. As such, the interpretations may result in a formulation of a conscious and cognitive planned browsing strategy or a cognitively initiated opportunistic strategy. The perceptual determinant formulates a strategy that is influenced by the displayed data and perceptual triggers. After the formulation of a browsing strategy, either cognitive or perceptual, the execution of the strategy leads to either an enhancement of a currently considered mental model or an initiation of another mental model. Hence, if one is able to support an operator in the browsing activity, one also facilitates the creation of qualitative internal models which is of major importance for the operator’s overall cognitive decision process. One way to support the browsing activity is to think about how relevant data is displayed and arranged, i.e. externalization. Moreover, externalization is in this thesis
considered to include visual graphics displayed on a computer screen but can also include aural, tactile, or olfactory stimuli [14].

We have discussed human perception and how one can facilitate operators in their cognitive process with the help of external support. Moreover, we consider this support in forms of both strategic browsing support and appropriate externalizing of data – i.e. representation support. In the next section, we will continue with presenting these two types of support with connections to our example phenomenon TWOSOME.

6.3 APPLICATION OF VISUALIZATION TECHNIQUES

In this section we will present visualization techniques and examples where human operators may be facilitated in their instrumentation process, e.g. minimizing MTTR, with the use of such techniques. As stated in the prior section, operator’s cognitive processes may be facilitated by visual external artefacts and may thus also be supported in both performing information gathering activities and from a comprehensible representation of data. In this thesis, we emphasize visualization technology to supply the support needed for operators to perform their online activities of observation and instrumentation.

When an operator is performing the online activity of instrumentation, visualization techniques may facilitate parts of the underlying activity, i.e. observation. The instrumentation activity is based on information gathered during the activity of observation, resulting in the forming of an internal model of the current situation – the operator’s situational awareness. Thus, appropriate visualization techniques may facilitate the operator’s situational awareness by providing activity support such as browsing strategies or navigational frameworks. For example, by providing for an opportunistic browsing strategy, the operator will be encouraged to explore and discover data which were not evident from observation at the exploration starting point. Moreover, by providing for the activity of zooming, the operator may increase and decrease content magnification under the constraint of a view frustum of constant size, to be able to further explore data – with the aim of enhancing the operators current situational awareness.

Browsing – Revisiting our example system from chapter 2; TWOSOME provides examples where visualization supports operator activities by enabling navigation and filtering. In TWOSOME, the operator is presented with the possibility of filtering the current visualization by toggling the information granularity of the visualization. The filter is based on the open computational systems model’s four layers – i.e. domain, system, fabric, and environment – thus enabling the operator to study each layer separate or in combinations, dependent of selected perspective (Figure 11a). Moreover, the operator is also supported with a navigational framework which enables a free form flight to explore the virtual world. With the possibility to dynamically explore, the operator may gather information in an opportunistic way and may e.g. navigate closer to a specific phenomenon for further observation. The combination of exploration and filtering is a basic support for TWOSOME operators when gathering information to be able to increase the current situational awareness – i.e. performing a browsing activity. Thus, with the support of visualization activities, the operator gains the level of situational awareness which will lay as a foundation for the decision making process needed for the following instrumentation of systemic qualities.

In these examples of activity support, human cognitive operators are performing dynamic exploration with the aim of information gathering – improving the current situational awareness. However, when the operator is confronted with data, it is important to consider how to represent this
data in such a way that it minimizes the operator’s time spent on interpreting the data, i.e. the time to transform data into information. Thus, it is important to provide appropriate externalizations of data – how data is represented – to be able to percept emergent properties which were not evident from raw data.

*Externalization* – Selecting the type of representation suitable for a specific set of data is complicated but there exists a great number of possible visualization techniques to apply based on different circumstances. For example, symbolic encoding can be used when creating visual representations of numerical values, ordinal, or categorical data [14]. Thus, it is important to map the computer representation to fit the perceptual representation by choosing encoding techniques which strive towards providing comprehensible conditions and thus maximising human understanding and minimizing cognitive response time. Colour can be a good addition to symbolic encoding, extending the possible amount of information displayed. However, the use of colour should be undertaken with some caution since there is a risk of lowering the visualization quality by applying too much colour [14]. Examples of other symbolic encodings may regard size, shape, or angle (Figure 11c).

Continuing on symbolic encodings, there exists ways to visually represent symbols in such way that they are more likely to be more easily distinguishable. Thus, certain colours and shapes are considered to pop out and be noticed even after very short exposure, prior to conscious attention – i.e. preattentive processing [25] (Figure 11d). The research regarding what determines which visual objects are presented to our attention and an understanding thereof is a very important contribution from visual science to data visualization. Moreover, to be able to identify if some symbol encoding is preattentively processed, experiments can be performed where the response time of finding a target in a set of distractors is measured and the time taken to identify the target should be independent of the number of distractors [25]. When measuring processing rates of 10 msec per item or faster, the processed item is considered to be preattentively processed whereas typical non-preattentive item is considered to be 40 msec or more per item [25]. Thus, if a symbol should be immediately identified, it should be differentiated from all other symbols in a preattentive way. However, preattentive symbols become less distinct as the variety of distractors increase and one should therefore consider the degree of difference between the target items and the non-targets and the degree of difference of non-targets from each other when determining whether the encoded symbol is preattentively processed or not [25].

If one once more revisits the TWOSOME example, from a representation viewpoint, there are a couple of techniques used which can be pointed out. The use of symbolic encoding can be e.g. found in the display of the abstract constructs in the domain layer visualization. Thus, the domain layer is displayed as a combination of selected symbol- and colour-encodings and a structural tree hierarchy, resulting in a visual representation of an abstract view of the current system (Figure 11b). Moreover, this example of a complex visualization is presented in further detail in the following chapter. In TWOSOME, colour is also used as a simple but fruitful encoding to determine which of the two perspectives one is currently observing – i.e. the overall colour scheme is a spectrum of red or blue, dependent on perspective. By consistently reflecting the colour scheme on the virtual environment and each residing object, the operator instantaneously gains understanding of which perspective is observed as a result of preattentive processing.
In chapter 5 – Method, we discussed the method of online engineering and its online activity of observation, followed by the instrumentation activity, with the aim of sustaining system behaviour. The instrumentation activity can be facilitated by the use of appropriate visualization techniques which support human operators in their observation activity; and thus laying the foundation for the decision making process regarding the following instrumentation. Moreover, since the human visual system is capable of absorbing and interpreting visual information rapidly, the support of effective visual artefacts which enhance human cognition in the observation activity is important.

By exploiting the dynamic and interactive medium of computer graphics one can create complex visualization tools which facilitates operators in their struggle to manage critical events. Such tools can be the result or combination of different visualization techniques which enables both activity support and representation support. Activity support may include browsing strategies, navigational frameworks, and further activities to explore data with an aim of improving the operator’s current level of situational awareness. Furthermore, when performing such dynamic exploration, it is imperative that the data which the operator is confronted with is presented using appropriate externalization and encoding techniques which strive towards providing efficient and comprehensible representation, thus, maximizing human understanding and minimizing cognitive response time.

There exists a plethora of possible visualization techniques which may be suitable in different situations, but it is not possible to determine a specific technique which is optimal in all conditions [25]. Thus, specific visualization tools combine sets of techniques to support operators in an optimal way depending on the current domain and the form of the task the operator seeks to perform. In the next chapter, we will continue with presentation of a visualization tool which was constructed to facilitate operators in instrumentation of systemic qualities in open complex systems.
of cognitive encoding symbols in figure 11.

In figure 11, a) a complex tree structure, displayed at three different levels of online filtering. b) A simplified view of the basic symbolic and color encoding applied to physical and cognitive constructs at each of the four open computational systems layers.

c) A random selection from a rich plethora of different symbolic encodings from different cultures and time periods. d) A basic set of PI decimals – how many eights (8) is represented in the text?

e) Preattentive processing allows humans to process the asymmetric color encoded symbols with a lower cognitive response time than the set of symmetric encoded symbols in figure 11d.

Fabric Domain System Environment

36/54
We have in prior chapters presented the need for operators to observe systemic qualities to be able to deal with occurring critical events affecting the intended system behaviour. Moreover, we have suggested a model to support isolation and adaptation of such affected open complex systems, followed by a method which enables observation and instrumentation of such systems. To facilitate the observation process, we presented visualization techniques which strive towards providing optimal conditions for human cognition to maximize comprehension and minimize cognitive response time. Moreover, considering and converging all requirements put forward by these prior chapters, we will continue with presenting a tool with the aim of facilitating operators in sustaining system behaviour.

In this chapter, we will present the visualization tool Distributed interaction system of complex entity relation networks (DISCERN) and discuss how such a tool support operators in observation and instrumentation activities of open complex systems. Moreover, we will discuss qualities of DISCERN with the prior chapters model, method, and techniques in mind. We also aim at giving the reader understandings regarding possible improvements for DISCERN which in itself can lay as suggestions for development of similar tools. This chapter should not be thought of as a complete set of recommendations for constructing similar tools but as a presentation of an example tool and its positive and negative qualities and how it could be improved, in accordance with the suggested model, method, and techniques.

It is difficult to discuss DISCERN’s support for observation and instrumentation activities without interweaving and referring to some concrete implemented system. Thus, our example phenomenon TWOSOME is revisited throughout this chapter to present concrete textual and graphical examples.

In the next section of this chapter, we will present DISCERN and the aims and objectives which were emphasized during the development of the tool. As stated in chapter 4, Model – Open Computational Systems, we consider systems which are executed and therefore in an online state, thus, focusing on observation and instrumentation activities of a specific phenomenon. Hence, following the presentation of DISCERN is sections which regard the tool from an observation and instrumentation viewpoint. The discussions are structured accordingly with the open computational systems model, thus delimited by the four layers of environment, fabric, system, and domain. Moreover, at each layer we will discuss negative and positive qualities of the support that DISCERN’s provides for operators. Moreover, these discussions will be accompanied with suggestions regarding possible improvements, with prior chapters in mind.

7.1 DISCERN

As presented in chapter 2 - Background, the tool DISCERN was created with the intent of addressing requirements such as supporting operators in observation and comprehension of systemic qualities; i.e.
visualization of open computational systems. Moreover, the development of DISCERN was a first experience in development of a methodological tool for visualization of open computational systems at SOCLAB3.

One of the primary requirements imposed on DISCERN was the need of enabling observation of, and interaction with, multiple aspects of the same phenomenon in real time [5]. Furthermore, DISCERN was required to implement this requirement in accordance with the open computational systems model; i.e. supporting the four abstraction levels; environment, fabric, system and domain. Thus, supporting four different qualitative perspectives and dynamic quantification of each of these four perspectives [5].

To be able to observe a system phenomenon in a virtual environment, one needs to be able to isolate the phenomenon so that it can be dynamically rendered and interacted with onscreen [5]. Furthermore, the system must be able to develop and evolve in parallel with the observation. Thus, DISCERN provides real time rendering of an isolated phenomenon without disrupting it. DISCERN also allows multiple operators to observe the same phenomenon which results in an opportunity to gain a large amount of different aspects of that phenomenon. For example, revisiting the TWOSOME phenomenon; DISCERN is used to observe TWOSOME from two different aspects, i.e. the defence domain and the attack domain. This enables operators to comprehend complex systemic qualities by performing observation of multiple perspectives of such qualities (Figure 17b).

As stated, DISCERN enables operators to observe and instrument a specific phenomenon and as a starting point for these online activities, DISCERN provides access via the domain frustum. For example, in TWOSOME, the operator is initially presented with the choices of selecting either the attack or the defence domain as a starting point for online activities (Figure 17a).

In the following sections, we will be discussing the support which DISCERN provide for operators regarding the online activities of observation and instrumentation.

7.2 OBSERVATION

In this section, we will primarily discuss qualities of DISCERN which enable support for observation activities, carried out by human operators. Moreover, the provided support aims at representing cognitive and physical constructs in a delimited frustum. As stated in chapter 4, Model – Open Computational Systems, we consider a frustum to be a cognitive bounding space; defined by some operator, and depicted as a solid volume and cut by the four layers of the open computational systems model, i.e. domain, system, fabric, and environment. Thus, the following subsections will discuss DISCERN’s support for observation activities, following the notion of layer separation accordingly to the model of open computational systems.

Before observing a phenomenon, the operator actively selects a frustum which delimits the phenomenon accordingly to the operator’s goals and will act as a starting point for observation activities. However, DISCERN only provides support for operators to observe a phenomenon from a domain frustum viewpoint (Figure 12). Thus, DISCERN enables observation of the specific open

\[3\] For more information, visit: http://www.soclab.bth.se

38 / 54
The concept of frustum is depicted as a solid volume which cuts through the four layers of the open computational systems model. A cognitive operator (A) may use a defined frustum as a starting point for observation activities of systems behavior (B), ranging from the abstract representation of the domain layer to the concrete physical environment.

computational system at hand via a domain frustum but does also limit the possibility for operators to select another type of starting point, e.g. a fabric frustum. Revisiting the TWOSOME phenomenon, an operator is presented with possibility to select between two different predefined perspectives within the domain frustum; the attack and defence domains. We advise the reader to note the difference between the notion of domain layer and domain frustum – figure 12 depicts the two concepts.

Domain – Since we observe the current phenomenon with a domain frustum starting point, we begin with the domain layer visualization. As stated in chapter 4, Model – Open Computational Systems, we consider the domain layer to be comprised of a set of cognitive constructs which, from an operator perspective, is a set of general concepts and relations; i.e. domains, concepts, ports, dimensions, and dependencies. Using symbolic encoding, each cognitive construct is visualized as a basic geometric form, except for dimensions and dependencies which are represented by single lines between the concepts in question (Figure 11b). Each cognitive construct is connected with each other, resulting in a form of a tree structure. Moreover, this tree structure can be thought of as a branch in a larger tree structure which encompasses the complete visualization of a specific node from a domain frustum viewpoint. By applying colour encoding to both symbols and the branch connecting the symbols, different branches are distinguishable from each other. The visual presentation of the tree structure in a virtual world is facilitated by always turning the symbols towards the observing operator thus assisting the detection of a specific symbol’s geometric shape (Figure 13). The symbol and colour encoding techniques used in the domain layer tree structure is consistently used throughout the forthcoming branches. This visualization concept was developed with the preconditions of a simple tree structure containing one domain, one concept, and one port in a two dimensional presentation. However, the open computational systems model allows multiple cognitive concepts to be associated with a cognitive domain and following, multiple ports to be associated with each concept.

Multiple cognitive concepts and ports combined with other branches in a three dimensional virtual world presentation resulted in the possibility to create very complex and non-comprehensive tree structures. Moreover, the complex tree structure representation combined with the lack of possibility for operators to select other starting frustums than a domain frustum for observation proved to be some of the major flaws in DISCERN. Providing recommendations regarding construction of an improved, dynamic, and comprehensible visualization concept for the domain layer is in itself a very complex task and is therefore one of the suggested future work for this thesis.

At SOCLAB, different conceptual designs for domain layer visualizations have been developed and we refer the reader to Appendix A for further conceptual screenshots.
System — From a domain frustum starting point, the system layer is visualized as a new branch on the existing tree structure. Colour encoding is used to taint the two symbols, physical concept and physical port, which makes them distinguishable from the other layer’s symbols (Figure 14). This type of representation of the system layer does only present physical concepts and ports connected to a specific physical node. However, DISCERN also allows operators to specifically observe the current system behaviour in further detail by supporting measurement activities on quantifiable systemic qualities at the system layer (Figure 17c). By enabling operators to observe such measurement’s results, i.e. to be able to observe selected details of the current system behaviour, an operator’s current level of situational awareness could be improved, thus followed by a more accurate instrumentation of the current systemic qualities with the aim of sustaining system behaviour.

Fabric — The fabric layer provides support for information dispersion to both local and remote nodes, thus enabling communication between nodes. However, DISCERN does not provide any visualization of this information dispersion or any physical network structures because of the domain frustum emphasis. However, the existence of fabric layer entities is represented with a single type of geometric form, using symbol and colour encodings consistent with the prior branches of the tree structure (Figure 15). Moreover, the symbol representing the fabric layer entities acts as the connecting root for the domain and system branches and as a connection downwards to the physical environment layer. To support operators in gaining a cognitive connection to the specific physical node type, a graphical representation of a physical artefact is also displayed.

However, the graphical location of the physical artefact is placed too close to the representation of the fabric layer symbol, resulting in a restricted view. Thus, it is recommended that the fabric layer should be represented as a geometric symbol placed some distance above each corresponding physical artefact and displaying a connection line in-between them. Moreover, a plethora of possibilities arise if one were to provide a visualization of the current physical network structure based on current communication links and information flow. For example, enabling an operator to observe the current information flow, represented as geometric shapes with size corresponding to the quantitative measurement, to be able to identify nodes which are critical from the perspective of sustaining information flow (Figure 17d). A precondition to be able to provide such visualization is to be able to observe a phenomenon from a fabric frustum.
Environment — A reoccurring need discovered when visualizing the type of open computational systems developed at SOCLAB is the need for a physical environment. The environment layer acts as, on the one hand, a cognitive enhancement for the operator when relating to a real-world physical environment, and on the other hand, to cognitively enhance the relation between spatial locations of the current system’s nodes. The environment layer is visualized with the aim to mimic a physical environment in a three dimensional virtual world with graphical artefacts representing system nodes, residing in a virtual world. In TWOSOME, a marine environment with an imaginary location was chosen to fulfil the requirements put forward by the selected scenario. Selecting an imaginary location both helped to fulfil the scenario requirements and eliminated possible problems with inconsistent representations compared to a real-world location. However, the decisions of selecting an imaginary location were also affected by the fact that DISCERN was not capable to present a sufficient consistent visualization of a real-world location.

Dependent on specific scenario and system requirements, a decision must be made regarding mimicking a real-world location to create spatial recognition and thereby also confronting inconsistency problems or selecting an imaginary location. Another dilemma that arises when representing physical artefacts such as the TWOSOME vessel VISBY in a virtual world is that of scale. Applying an enlarged scale to the representation of physical artefacts compared with the virtual environment may facilitate visibility and comprehension of the artefact’s spatial position for the operator. However, the comprehension regarding relation in-between artefacts and environment, distance assumptions, and the precise position of an artefact may be suffering from applying an enlarged scaling.

Prior in this thesis, we have presented multiple examples of physical environments displayed in virtual worlds (Figure 4, 5, and 7).

Combining each of the three layer’s representations, a basic tree structure emerges (Figure 16a). However, in the quite common case where multiple cognitive and physical concepts and ports is displayed, the tree structure become complex and harder to comprehend (Figure 16b). In a phenomenon such as TWOSOME, multiple nodes are often presented with high spatial proximity and displaying a complex tree structure connected to each of these nodes results in a very complex, and in some cases incomprehensible, view (Figure 16c). When presented with such complex views, an operator require support for dynamically filtering information based on the operator’s current goals. DISCERN’s support for such Information filtering and other instrumentation activities are discussed in the following section.


**Figure 15. Showing a physical node and physical artifact, the TWOSOME vessel VISBY, with symbol and colour encoding at the fabric layer.**

The subsections above discuss what support DISCERN provides, in terms of representation, for operators when observing a phenomenon. Moreover, these representations of cognitive and physical constructs aim to fit a perceptual representation by the selection of appropriate visualization techniques. However as discussed, the techniques used in DISCERN can be improved to further provide comprehensible conditions for observing operators and thus maximizing human understanding and minimizing cognitive response time.

In the next section, we will discuss what support DISCERN provides for operators performing instrumentation activities. The decisions carried out in instrumentation activities are based on the operator’s situational awareness and the level thereof is increased by performing observation activities. Hence, it is vital that DISCERN provide support for both observation and instrumentation activities in such way that the level of successes in sustaining and restoring deviating system behaviour is increased.

### 7.3 INSTRUMENTATION

In the prior section, we discussed support for observation activities, provided by DISCERN for operators. With such representation support, operators may gain a level of situational awareness which in itself lay as foundation for the decision process affecting the following instrumentation activity. Based on the decisions taken, operators uses DISCERN to facilitate the realization of the intended instrumentation. The main instrumentation activities that DISCERN provide is that of navigation, information filtering, and a text based command console. Utilizing this support, an operator may affect the observed phenomenon in accordance with the operator’s current aims and objectives.

**Navigation** – DISCERN provides a basic flight type of navigation in each virtual world. Hence, the operator is presented with four degrees of freedom by combining the use of keyboard and mouse. The flight navigation technique enables the operator to easily explore the virtual world and is commonly used in computer games and simulations where navigation of three dimensional worlds is required. Moreover, using DISCERN, if the operator align the view in such way that the operator’s view centre points at a cognitive or physical construct, information regarding the construct and its corresponding physical node is presented textually (Figure 16c). The use of four degrees of freedom enables
experienced user to freely navigate and explore a virtual world, however, inexperienced users may find it problematic to use this type of navigation.

To facilitate navigation for inexperienced users, it is recommended to be able to delimit the navigation to follow a set of predefined paths combined with two degree of freedom. Moreover, providing a set of predefined fixed points with suitable view angels, which the user easily could change position to with a keyboard command, could further support inexperienced users.

However, such fixed view positions could also support experienced users in performing a more time efficient navigation by lowering the travel time when being aware of the spatial position of both appropriate view position and the navigational goal.

Information filtering – As discussed in the prior section, when presenting a large amount of information concurrently, the display may be cluttered with information, thus limiting the level of comprehension for observing operators. The figure 16c provides a good example of such a case at our example phenomenon TWOSOME. However, DISCERN provide online filtering of the current information presented, in correlation with the open computational systems model’s four layers. Hence, an operator is able to enable or disable the representation of each of the four layers, thus being able to filter what information is relevant to observe, dependent on the operator’s current aims and goals. This type of filtering was proven very useful when using DISCERN for demonstrational purposes, thus limiting the amount of presented information to increase the level of comprehension.

However, the filtering in itself is limited since DISCERN only support toggling of visibility of the four layers, thus not supporting e.g. operator selections, intersections, or combinations of interesting information, spanning over multiple layers. With support for a more powerful and dynamic filtering mechanism, it would be possible to handle a larger range of filtering desires from different operators with dissimilar goals.

The prior mentioned need for support for observation via frustums other than the domain frustum would require a very dynamic filter or a set of filters which would fulﬁl requirements put forward by each type of frustum visualization.

Command console – As stated above, DISCERN support operators with the possibility to explore the current virtual world by means of navigation. Moreover, an operator may navigate towards a speciﬁc node and observe the system layer’s branch, thus observing both its physical concept and physical communication port. DISCERN also enables operator to instrument the current phenomenon by accessing a speciﬁc node’s physical port. Accessing the communication port is carried out by aligning the operator’s view to point directly at a speciﬁc communication port and pressing the enter key at the keyboard, the user is then presented with an orthographical projected text based command console (Figure 16d). With the semi-transparent command console active, the mouse and keyboard focus is shifted from the virtual world over to the console. Using a speciﬁc set of commands the operator is able to instrument the selected system node in real time e.g. relaying an order to commence a mine sweep. To enable the operator to instrument the current system at hand via a semi-transparent orthographical projected text based console were proven to be an effective method of interaction without disrupting the operator’s current virtual world spatial context.

However, the instrumentation is limited to textual commands, thus forcing the operator to self consider the contextual connection in-between the commands carried out and the eventual resulting effects on the current systemic qualities, without any direct graphical feedback.
In this chapter, we have been discussing the visualization tool DISCERN and the support it provides for operators when performing observation and instrumentation activities. The tool was developed to support these online activities in accordance with the open computational systems model; i.e. supporting the four layers – domain, system, fabric, and environment. Prior to the observation of a phenomenon, the operator actively selects a frustum which delimits the phenomenon and acts as a starting point for the observation activities. However, DISCERN only provides observation of the phenomenon at hand via one particular domain frustum which limits the possibilities for operators to actively select any other frustum as starting point for their observation activities.

From a domain frustum view, DISCERN presents a complex tree structure which is comprised by cognitive and physical constructs in accordance with the open computational systems model. By applying symbol and colour encoding on each construct and enclosed branches, operators are able to observe a tree structure which encompass the complete visualization of a specific node from the cognitive domain layer down to the physical environment layer, from a domain frustum viewpoint. The fabric layer provides representations of physical artefacts connected to each tree structure’s root and the physical environment layer provides representation of a physical environment which places the current node in a spatial location. These representations of cognitive and physical constructs aim to fit a perceptual representation by the selection of appropriate visualization techniques. However, as discussed in the sections above, DISCERN can be improved to further provide comprehensible conditions for operators performing observation activities and thus maximizing human understanding and minimizing cognitive response time.

The resulting tree structures often become too complex to be easily comprehensible for human operators and in cases where multiple concepts and ports were applied, the resulting visualization becomes even more complex. Thus, observing multiple nodes containing such complex tree structures, with a high level of spatial proximity, result in an enlarged need of information filtering. However, DISCERN does provides support for a limited form of information filtering, based on toggling visibility of the four layers, thus not supporting e.g. operator selections, intersections, or combinations of information, spanning over multiple layers.

Regarding the system layer, DISCERN supports operators with the possibility to observe measurement results of quantifiable systemic qualities, thus observing detail of the current system behaviour. Such observations may improve the level of situational awareness, which may enhance the implementation of the following instrumentation activities. However, regarding the fabric layer, if operators were able to observe the physical network structure, based on current communication links and information flow, another plethora of possibilities arises. For example, enabling an operator to observe the current information flow, represented as geometric shapes with size corresponding to quantitative measurements, to be able to identify nodes which are critical from the perspective of sustaining information flow. A precondition to be able to provide such visualization is to support operators in observation of a phenomenon from a fabric frustum point of view.

The complex tree structure presentation and the restrictive possibility of selection starting frustum for observation activities proved to be some of the major flaws in DISCERN. Thus, by enabling operators to select from a variety of frustums, a plethora of new visualization issues arises and in the section 8.3 Future challenges at the next chapter we will shortly discuss such visualizations. Moreover, we refer the reader to Appendix A for conceptual screenshots regarding such issues.
Figure 16. a) A basic tree structure comprised of the three methodological layers; domain, system, and fabric, each with respective symbolic encoding. b) Displaying a more common example tree structure where multiple cognitive and domain concepts are present, and ports with respective symbolic encoding. c) Displaying multiple nodes with complex tree structures and a high level of spatial proximity. d) An example of the semi-transparent orthographical command console from the TriOsome phenomenon.
Figure 17. a) The menu of selection which operators are provided with when choosing which perspective of TWOSOME to observe. b) DISCERN enables operators to observe TWOSOME from two different aspects, i.e. the defence and the attack domain. c) Observation of the current system behaviour in further detail by displaying measurement results of quantifiable systemic qualities at the system layer. d) A visualization concept of a physical network structure with nodes and communication links. e) The menu of selection which operators are provided with when observing the actual domain and the actual domain.

Image 17. a) The menu of selection which operators are provided with when choosing which perspective of TWOSOME to observe. b) DISCERN enables operators to observe TWOSOME from two different aspects, i.e. the defence and the attack domain. c) Observation of the current system behaviour in further detail by displaying measurement results of quantifiable systemic qualities at the system layer. d) A visualization concept of a physical network structure with nodes and communication links.

Figure 17. (a) The menu of selection which operators are provided with when choosing which perspective of TWOSOME to observe. (b) DISCERN enables operators to observe TWOSOME from two different aspects, i.e. the defence and the attack domain. (c) Observation of the current system behaviour in further detail by displaying measurement results of quantifiable systemic qualities at the system layer. (d) A visualization concept of a physical network structure with nodes and communication links.

Image 17. (a) The menu of selection which operators are provided with when choosing which perspective of TWOSOME to observe. (b) DISCERN enables operators to observe TWOSOME from two different aspects, i.e. the defence and the attack domain. (c) Observation of the current system behaviour in further detail by displaying measurement results of quantifiable systemic qualities at the system layer. (d) A visualization concept of a physical network structure with nodes and communication links.
With all prior chapters behind us, having discussed and presented systems, model, method, techniques, and a tool conveying the presented requirements, we have finally reached the chapter of this thesis where we will present the thesis summary and revisit the initial problem description. Thus, a summary discussion together with directions for roads leading towards future development will conclude this thesis.

8.1 SUMMARY

When considering the system of systems concept, i.e. large scale systems which are built upon interdependent and self contained subsystems, each subsystem is exposed to unanticipated events occurring in some shared execution environment. Such critical events may cause the current system behaviour to deviate from an intended state. Moreover, since it is impossible to consider all possible events at the offline phases of design and implementation, we must address such issues in an online matter. For example, allowing human operators to instrument the current system at runtime towards the goal of restoring system behaviour to an acceptable state. However, such approach requires a supportive model for isolation, adaptation, and validation of such systems and a method for handling online activities such as observation and instrumentation. In this thesis, we have selected the model of open computational systems and the method of online engineering to provide such required support.

Our principal assumption is that the instrumentation activities to restore system behaviour are performed by human operators, thus requiring a comprehensible visual representation of relevant systemic qualities to gain a certain level of situational awareness after performing observation activities. The situational awareness in itself lay as a foundation for the upcoming reasoning and decision process regarding how the system at hand should be instrumented in conformance with the operator’s qualitative goals. To support operators in gaining a sufficient level of situational awareness, we emphasize the use of visualization techniques which facilitates operators in their observation activities. Thus, we consider the use of appropriate visualization techniques which facilitate operators in such ways that their cognitive response time to a visual signal is minimized and their level of situational awareness is increased, which results in enhancing their capability of instrumentation.

The requirements described above boils down to the need of a tool which supports cognitive operators in their observation and instrumentation activities in conformance with the presented model, method, and techniques. Thus, such a tool must supply functions based on technical as well as cognitive requirements.

Throughout this thesis, we have presented the type of systems which we consider, the critical events which the systems are exposed to, a model, method, and techniques used to support intervention of these critical events and the type of agent considered to sustain and restore system
behaviour. Finally, we presented a tool which primarily aims to supply functions conveying all the requirements put forward by these prior chapters. The presentation, evaluation, and discussion of such a tool reveal that the supportive tool of DISCERN did provide a good approach towards fulfilling both the technical as well as the cognitive requirements presented, based on the fact that it was a first experience in implementing such a tool. However, DISCERN do also leave plenty of room for improvements and future development regarding support for observation and instrumentation activities with the aim of maximizing human comprehension and minimizing cognitive response time.

8.2 CONTRIBUTIONS

Using DISCERN to observe abstract constructs, the operator is presented with complex tree structures which were built upon a basic visualization concept, using fundamental geometric forms for symbolic encoding, but fails to facilitate operators when applied to actual cases where the level of complexity is high. Hence, DISCERN provides symbolic and colour encoding to facilitate observation of abstract constructs which should support the operator in detection and identification of separate constructs. However, when applied in cases concerning multiple constructs at each tree structure and multiple nodes with high spatial proximity; the level of distractors were raised and the cognitive gain of the symbol and colour encoding decreased, thus lowering the operator’s overall level of situational awareness. This situation also brought forward the need of more dynamical filters, supporting e.g. operator selections, intersections, or combinations of information, spanning over multiple layers, since the provided filter only support toggling of four levels of information granularity.

DISCERN do not support operators regarding the ability to select a specific frustum, other than the predefined, as starting point for observation and instrumentation activities. Hence, by providing operators with an active selection of frustum enables a plethora of new possibilities to observe other systemic qualities which may improve the operator’s level of situational awareness and affecting the overall instrumentation quality. For example, providing visualization of the current physical network structure based on current communication links and information flow would enable operators to identify nodes which are critical from the perspective of sustaining information flow. A precondition to be able to provide such visualization is to be able to observe a phenomenon from a fabric frustum.

The following points summarize the evaluation and recommendations presented regarding the DISCERN tool and its supportive functions.

- Multiple cognitive concepts and ports combined with other branches in a three dimensional virtual world presentation resulted in very complex and non-comprehensive tree structures.

- The graphical location of the physical artefact is placed too close to the representation of the fabric layer symbol, resulting in a restricted view. Thus, it is recommended that the fabric layer representation should be placed with some distance above each corresponding physical artefact and displaying a connection line in-between them.

- Dependent on specific scenario and system requirements, a decision must be made regarding visualizing a virtual environment to either mimicking a real-world location to create spatial recognition and thereby also confronting inconsistency problems or selecting an imaginary location.
• Applying an enlarged scale to the representation of physical artefacts compared with the virtual environment may facilitate visibility and comprehension of the artefact’s spatial position for an operator. However, the comprehension regarding relation in-between artefacts and environment, distance assumptions, and the precise position of an artefact may be suffering from applying an enlarged scaling.

• To facilitate navigation for inexperienced users, it is recommended to be able to delimit the navigational support to follow a set of predefined paths combined with two degree of freedom. Moreover, providing a set of predefined fixed points with suitable view angels, which the user easily could change position to with a keyboard command, could further support inexperienced users.

• The filtering support in DISCERN only support toggling of visibility of the four layers of the open computational systems model, thus not supporting e.g. operator selections, intersections, or combinations of interesting information, spanning over multiple layers. With support for a more powerful and dynamic filtering mechanism, it would be possible to handle a larger range of filtering desires from different operators with dissimilar goals.

• DISCERN provide support for system instrumentation but the operator is limited to using textual commands at a command console, thus forcing the operator to self consider the contextual connection in-between the commands carried out and the eventual resulting effects on the current systemic qualities, without any direct graphical feedback.

To conclude the evaluation and provided recommendations, we emphasize the use of appropriate visualization techniques to facilitate operators in such ways that their cognitive response time to a visual signal is minimized and their level of situational awareness is increased, which results in enhancing their recovery process – e.g. minimizing MTTR.

8.3 RELATED WORK

Considering a set of different fields, visualization techniques are used to facilitate operators in tasks such as monitoring, analysis, and control process. For example, supporting operators in assessing system stability in complex power systems [23], or in the field of network visualization, several visualization approaches have been presented regarding monitoring and analysis of networks [28][29]. Moreover, an interesting note from the field of complex networks; Livnat et al. presents a novel visualization paradigm for network intrusion detection which facilitates and promotes situational awareness in complex network environments [27]. Livnat et al.’s approach emphasize use of certain visualization techniques to foster rapid correlation and perceived associations of network intrusion events with the aim of supporting the decision making process of identifying problems, characterizing them, and determining appropriate response. However, this paradigm considers enhancing situational awareness via visual correlation of existing intrusion events, thus does not consider real time information or any system intervention, e.g. system instrumentation, at all.
CONCLUSIONS

8.4 FUTURE CHALLENGES

A large part of this thesis presents and describes different concepts, such as a method, model, or techniques, to justify and motivate our statements concerning qualities of a supportive tool for observation and instrumentation activities. Thus, with all this background information prepared, the following subsections present suggestions regarding future work which emphasize on performing experiments regarding human cognition and visualization concepts which provides for new ways of sustaining system qualities and behaviour.

- Enable operators to dynamically construct a frustum to be used as a starting point for their exploration and refinement activities. With such support, a plethora of new possibilities of observation of different systemic qualities arise, thus requiring a set of new visualization concepts which comprise the new requirements put forward by specific frustums. Experimental software has been developed at SOCLAB concerning parts of this area and screenshots of such visualization concepts can be found in Appendix A.

- Performing measurements on human cognitive response time when an operator is set to restore a specific systemic quality using different sets of visualization techniques, e.g. preattentive processing, to derive a set of visualization techniques which provides the most advantageous conditions for observation of that specific systemic quality.

- Performing experiments within an open complex system with multiple operators concurrent carrying out observation and instrumentation activities to observe and derive emergent behaviours regarding the OODA process from an organizational perspective.

8.5 CONCLUDING REMARKS

DISCERN provides operators with the possibility to intervene in an open complex system – to observe and instrument complex systemic qualities with visualization techniques playing the vital role of provider of propitious conditions for operators’ counteracting unanticipated events with a succeeding level of situational awareness. Hopefully, this thesis has provided an insight into the area of visualization of open computational systems, remaining is only an infinite number of possibilities to apply visualization to facilitate operators.
REFERENCES

REFERENCES


REFERENCES

