Bachelor thesis
Computing Science

Radboud University

Access Control in a Task-Based Methodology

Author:
Romy Stähli
s1000061
r.stahli@student.ru.nl

First supervisor/assessor:
dr. ir. Erik Poll
erikpoll@cs.ru.nl

Internship supervisor:
Michael de Vos
m.d.vos.05@mindef.nl

Second assessor:
dr. Peter Achten
p.achten@cs.ru.nl

June 27, 2020

Abstract

Maritime IT, the in-house IT supplier of the Royal Netherlands Navy (RNLN), focuses on implementing the Task-Based Methodology within the development of the new Command and Control (C2) systems. Yet, within this methodology, no research has been conducted on the expression of access control regarding "secure by design". Therefore, this thesis investigated the ACM, ACL, RBAC, ABAC, and TBAC access control models. By adapting the ABAC model, this thesis proposes a model that fits best within the Task-Based Methodology as applied by Maritime IT.

Contents

1 Introduction **3**

2 Background **6**
2.1 Command and Control (C2) systems 6
2.2 Task-Based Methodology for C2 8
2.3 Relevant basics of the deployment of personnel on board of RNLN ships 10

3 Access Control Models **14**
3.1 Guidelines and Requirements 14
3.2 Access Control 15
3.3 Access Control Models 17
3.3.1 Access Control Matrices (ACM) 17
3.3.2 Access Control Lists (ACL) 18
3.3.3 Role-Based Access Control (RBAC) 20
3.3.4 Attribute-Based Access Control (ABAC) 22
3.3.5 Task-Based Access Control (TBAC) 24
3.4 Recommendation Extended Task-Based Methodology 24

4 Model Selection and Implementation **26**
4.1 Adapted ABAC model 26
4.2 Combining ABAC Model with Task-Based Methodology . . . 30

5 Conclusion **34**

Chapter 1

Introduction

From 2019, Maritime IT focused on implementing the Task-Based Methodology within the development of the new C2 systems. Yet, within this methodology, no research has been conducted on the expression of access control regarding "secure by design". Therefore, this research investigates how access control can be added to the Task-Based Methodology through the following research question:

> *To what extent is it possible to adjust the Task-Based Methodology such that the corresponding model contributes to "secure by design" in the new C2 systems by using access control?*

This research will seek to answer this question by analyzing the access control models available that can incorporate elements, such as current location and time, that are crucial to ensure "secure by design" and are adaptable to implement within Maritime IT's Task-Based Methodology. The goal of this research is to find an access control model that can be implemented within the Task-Based Methodology. Also, there will be envisaged how such a model could be implemented within this methodology.

For over 50 years, the RNLN has an in-house (non-commercial) IT supplier: Maritime IT. It develops C2 systems for most of its military platforms. Maritime IT wants to include security requirements during the design of software. It does so by applying the so-called "secure by design"-approach, which incorporates the security level of a software product from the start of development and throughout the lifecycle of the product.

Michael de Vos [3] described a task-based modeling approach in 2019, which later has been applied by Maritime IT for the development of the new C2 systems. The concept of this methodology was named "Task-Based Methodology", and it describes how to generate practical and explainable plans by using pre- and post conditions assigned to tasks. The precondition deter-

mines when a task can be performed. When a task is completed, a particular goal is accomplished, and thus a specific post condition holds.

The resulting plan can be used to accomplish goals by (automatically) performing tasks in the order as prescribed by the generated plan. The model of the Task-Based Methodology automatically transforms into working software systems by the use of code generation and is directly the implementation.

In order to implement access control, this thesis will analyze access control models that are able to assign access right to subjects: Access Control Matrices (ACM) [6] [7], Access Control Lists (ACL) [6] [7], Role-Based Access Control (RBAC) [5] [6], Attribute-Based Access Control (ABAC) [8] and Task-Based Access Control (TBAC) [11] [13].

An ACM is a table with on one axis the subjects and the other axis the objects. Each cell is filled with the access rights for the combination of subject and object. An ACL defines, for each object o a list L, which is called o's ACL. This list enumerates all the subjects that have access rights for o and for each subject s gives the access rights. Within RBAC, users are assigned to roles, and access rights are assigned to roles. In this way, no access rights have to be given to users directly. ABAC supports boolean logic, in which rules contain "If ... Then" statements about which user requests access, the object it applies to, and the action that must be executed. This access control model can deal with environmental conditions such as location and time. Within TBAC, tasks are a group of permissions and divided into two classes, common tasks, and professional tasks. The common tasks are assigned to the organization unit to which all employees belong who are allowed to execute the task. The professional tasks are assigned to a particular role in a particular organizational unit.

The access control models will be analyzed based on the requirement that the model needs to be flexible in terms of the access control policies. This flexibility is required because, in some situations, the access control policies need to be changed. For example, when someone is not able to fulfill his role. In this case, someone else needs to take over his role. Thus, the model must allow changes during the use of the system. Besides the need for flexibility, the model must deal with the current location and time, and the type of device of a user.

Chapter overview In Chapter 2 we give background information about Command and Control systems, the Task-Based Methodology, and relevant basics of the deployment of personnel on board of RNLN ships. In Chapter 3, the requirements for the access control extension for the Task-Based

Methodology are mentioned, the possible access control models are reviewed, and a conclusion is given about which access control model fits best and why. How the access control model should be adapted such that it can be added to the Task-Based Methodology is described in Chapter 4 together with examples. Finally, we conclude this thesis and look outside the scope to interesting ideas to extend this research in Chapter 5.

Chapter 2

Background

In this chapter, we give background information about Command and Control (C2) system in Section 2.1. The Task-Based Methodology is explained in Section 2.2. In Section 2.3, relevant basics of the deployment of personnel on board of RNLN ships are explained.

2.1 Command and Control (C2) systems

C2 is a function in military operations and consists of the leadership and direction given to a military organization to accomplish its mission. It is one of the most critical functions because C2 serves to integrate the other functions in military operations (such as intelligence, maneuver, fire power, combat service support, and force protection). C2 enables military capabilities to be employed effectively and efficiently.

At the operational level, it is about designing and directing campaigns and major operations to achieve the military-strategic objectives. The operational level translates the military-strategic objectives into concrete, feasible tasks for the tactical deployment of forces in a given area of operations. The military contribution is planned and implemented with other, non-military actors and organizations in a comprehensive approach designed to achieve the desired result [9].

C2 is a means toward creating value (e.g., the accomplishment of a mission). C2 is about focusing on the efforts of some entities (both individuals and organizations) and resources (including information) toward the achievement of a task, objective, or goal. The purpose of C2 has remained unchanged since its military inception, but the challenges encountered and the way C2 is understood have changed significantly over time. These changes are the result of a combination of the coevolution of C2 approaches and technology, the nature and type of military operation, the emergence of new capabilities

of forces globally, and the changing environments in which militaries try to achieve mission success [1].

There are many definitions of C2. In this thesis, we use the definition from Vassiliou et al. [12], which encompasses everything needed to accomplish missions:

> *"Command and Control"(C2) denote the set of organizational and technical attributes and processes by which enterprise marshals and employs human, physical, and information resources to solve problems and accomplish missions. ∼ Vassiliou et al. (2014)*

The definitions in the figure below were used as a foundation and context for the Task-Based Methodology. It is a summary of C2 and the most common terms and relations in the literature.

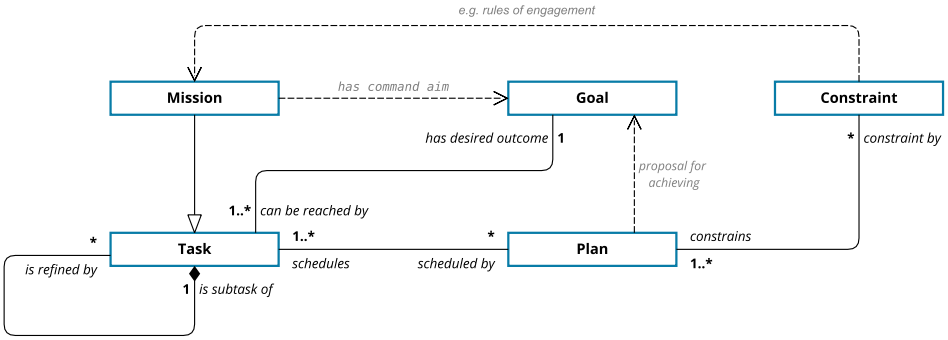

Figure 2.1: Derived relationships between mission, goal, task and plan [3]

A **mission** is defined as a high-level, composite or complex **task**. This means that the task can be decomposed into multiple sub-tasks. Each task has a **goal**. To reach this goal, sub-tasks are selected and ordered by a **plan**. This plan will accomplish a **goal** when performed according to the ordering. The **constraints** are used for generating the plan [3].

2.2 Task-Based Methodology for C2

The Task-Based Methodology presents a method and supporting modeling framework for preserving the operational knowledge (i.e., the why, what, and how to the system's capabilities) while developing C2 systems [3]. The resulting models are suitable for the use in a model-driven engineering approach, which means that the resulting models are automatically transformed into working software systems by the use of code generation.

Task The term task refers to an activity that is performed to accomplish a goal. A task can be of varying levels of complexity and abstraction. Furthermore, a complex task can be split into more concrete sub-tasks until the level of atomic actions, which can not be divided any further. Each time a sub-task is completed (e.g., its goal has been reached), a contribution is made toward achieving the goal of the composite task.

Pre- and post conditions Most often, a task can only be executed when the preconditions hold true. Because preconditions determine when a task can be performed, they appear to be usable for access control. When a task is completed, it is expected that a particular goal has been achieved, or in other words, a particular post condition holds true. A post-condition regarding access control can be seen as to whether someone got access to something or not.

Combining the concept of pre- and post conditions and the definition of a task, the task itself becomes easy to read and understand (see Figure 2.2). Here, the input is the information needed for the task. The output of the task will be the goal of the task when it has finished.

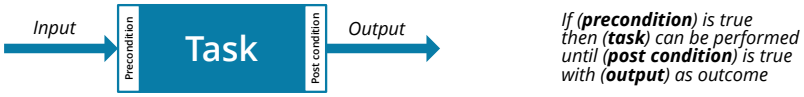

Figure 2.2: A task defined with input, precondition, post condition and output [3]

When combining this with the use of a hierarchical task composition structure and pre- and post conditions, we get Figure 2.3. This model captures the why, what, how and when (i.e. what is possible under which condition).

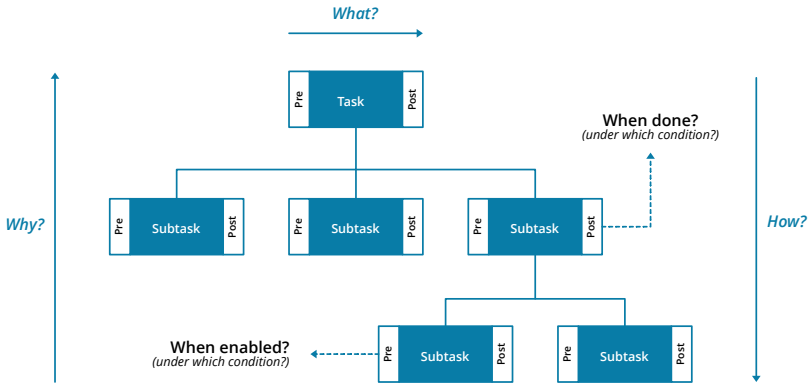

Figure 2.3: Hierarchical task model [3]

Planning support Planning is one of the central activities in C2, which is being performed at the strategic, tactical, and operational levels. Planning is the process of deciding beforehand, **what** is to be done (i.e. formulating a command aim or goal), **when** it is **enabled** (i.e. determining the precondition) and **when** it is **done** (i.e. determining the post condition), **how** it is to be done (i.e. formulating tasks) and **why** the task needs to be done. The outcome of this planning process consists of one or more (alternative) plans. Each plan describes a course of action. When this plan is being executed, it leads to the achievement of a goal.

2.3 Relevant basics of the deployment of personnel on board of RNLN ships

In this section, we explain relevant information that is needed before we investigate the access control models. With this knowledge, it is easier to explain the different access control models.

We start by explaining the term "arbeidsplaats", followed by "gereedheidsgraden", "scheepsrollen" (situation dependent groupings) and finally the "scheepsrollenplan" (situation dependent grouping plan).

"Arbeidsplaats" Each crew member is assigned to one unique primary "arbeidsplaats" and may be assigned to one (or more) secondary "arbeidsplaats". Below a few primary "arbeidsplaatsen" on board of a RNLN ship are explained:

- "Officier van de wacht op de brug": safely navigate the ship.
- Cook: prepare meals for the crew.
- Commanding Officer: overall leadership.
- Chief of Administration: take care of the personnel and financial administration.

While each crew member is assigned to an "arbeidsplaats" with accompanying primary task(s), he usually has multiple secondary tasks on board of a ship based on the current situation on board. A secondary task is, for instance, that he is a fire-fighter when there is a fire on board. Fire-fighters among the crew members are needed because there is no fire brigade around on the sea. So, every crew member has to be multi-functional deployable during different situations. These different situations are defined by "gereedheidsgraden" and "scheepsrollen" as explained below.

"Gereedheidsgraden" These define the extent to which a team can respond to an assignment or an event [10]. The higher the readiness (where one is the highest), the faster one can take action to perform certain tasks. Every "gereedheidsgraad" has its specific accompanying tasks. Below we explain the different "gereedheidsgraden".

5. "Reewacht" - this applies when a ship is anchored or moored in a safe harbor. Only a limited part of the crew is on duty for surveillance, security, and initial emergency response.
4. "Verlichte Zeewacht" - this applies to a sailing ship that, in principle, does not perform any tasks other than safe navigation in open water. A small part of the crew is on duty for safe navigation, regular business operations, and initial emergency response.

3. "Zeewacht" - this is taken when there are additional activities or increased risk. The crew and resources needed to perform the requested activity or to pose an immediate threat or danger are directly available. In principle, this "gereedheidsgraad" can be sustained for a longer period (several weeks, up to a few months). Normally this means that a quarter to a third of the crew is on duty and divided in shifts of six hours. Besides being formally on duty, the crew does a lot of other work.

2. "Oorlogswacht" - this provides the highest possible readiness, which can be sustained for a longer period (two to three weeks). Normally this means that half of the crew is on duty (for this, the entire crew is divided into two divisions) and that as much as possible systems are available immediately or at very short notice.

1. "Gevechtswacht" - the readiness is maximal. Generally, this means that the division which was on duty for "oorlogswacht" remains on their posts. The other division comes on duty too, to fill in the battle- and calamities specific roles. Thus, the complete crew is on duty, and immediate deployment of all systems and functionalities are possible. This readiness can be sustained for a limited time.

"Scheepsrollen" (situation dependent groupings) Each "scheepsrol" specifies a grouping (a tailored package) of crew members and resources to safely carry out the activity and shorten the response time to occurrences [10]. As soon as such a "scheepsrol" is started, crew members know what role to perform at that time. A few "scheepsrollen" are listed below.

- Maneuver grouping - comes to action when the ship has to navigate in narrow or shallow waterways or the immediate vicinity of other ships or objects.

- Replenishment at sea grouping - comes to action for refueling or transfer of other goods or personnel at sea.

- Flight grouping - comes to action for take-off and landing of a helicopter.

Which roles need to be executed during which "scheepsrol" is described in a "scheepsrollenplan" as explained below.

"Scheepsrollenplan" (situational dependent grouping plan) A "scheepsrollenplan" is a combination of occupancy and/or "gereedheidsgraden" and occupancy of additional "scheepsrollen" [2]. An example is given in Table 2.1. This tables shows of seven crew members their different roles during different "gereedheidsgraden" and "scheepsrollen".

When looking at crew member a, the "scheepsrollenplan" shows that he is a MAD ("medische actie dienst") during "gevechtswacht", a baker (the primary reason why he is on board) during "oorlogswacht" and "zeewacht", and a MAD in crash-boat during flight grouping.

	"Gevechtswacht" division A	"Oorlogswacht" division A	"Zeewacht" (not division specific)	Flight grouping	Maneuver grouping
Crew member a	MAD	Baker	Baker	MAD in crashboat	-
Crew member b	Operationsroom officer	Operationsroom officer	Operationsroom officer on standby duty	Operationsroom officer	Navigational assistance
Crew member c	Airpicture compilation	Airpicture compilation	Airpicture compilation	-	Lookout
Crew member d	Chief technical control room	Chief technical control room	Chief (A1) technical control room	-	-
Crew member e	Chief damage control team 1	Chief damage control team	Chief (A2) technical control room	-	-
Crew member f	Chief engineroom	Off duty	Chief (B1) technical control room	-	Chief engineroom
Crew member g	Chief damage control team 2	Off duty	Chief (B2) technical control room	-	-

Table 2.1: Example "Scheepsrollenplan" (situational dependent grouping plan)[a]

[a]This is not in line with reality, but this is to give an impression of the setup of a "scheepsrollenplan"

Chapter 3

Access Control Models

This chapter first describes the guidelines and requirements for extending the Task-Based Methodology with access control in Section 3.1. Second, general information about access control and terminology is described in Section 3.2. Third, an overview of the possible access control models that can be used to extend the Task-Based Methodology is given in Section 3.3. Finally, a recommendation regarding the access control models is described in Section 3.4.

The access control models which are reviewed are: *Access Control Matrices* (Section 3.3.1), *Access Control Lists* (Section 3.3.2), *Role-Based Access Control* (Section 3.3.3), *Attribute-Based Access Control* (Section 3.3.4), and *Task-Based Authorization Controls* (Section 3.3.5).

3.1 Guidelines and Requirements

The *Identity and Access Management* (IAM) within Maritime IT manages digital identities and user access to data, systems and resources within a system. Within this environment, *static attributes* and *role types* are used to grant certain access to applications and are obtained by several source administrations. One of the source administrations is *Peoplesoft* where personal information is gathered, such as the "arbeidsplaats" and employee number.

Static attributes of a crew member are, for example:

- "Arbeidsplaats" (explained in Section 2.3)
- Employee number
- Rank
- Department

Even though every application has its policies regarding access rights, there are a few guidelines, namely granting access rights based on the following *role types*:

- "Arbeidsplaats" - Each "arbeidsplaats" has its accompanying tasks and access rights.
- Department - a specific division, building, or field of knowledge.
- Authorization profile - clustering of the most common set of access rights and tasks, so that those do not have to be given to individual "arbeidsplaatsen".
- Personal working relationship - access rights that are needed to work on a, for example, personally bounded project, working group, participation committee, in-house emergency response.

Besides these static attributes and guidelines, the role types, there are a few requirements for the extended Task-Based Methodology from the RNLN. These requirements should make the deployment of personnel and the execution of tasks more efficient:

- Current location, which is a *dynamic attribute*
- Current time, which is a *dynamic attribute*
- Type of device
- The model needs to be flexible because, in some situations, the policies might need to be changed. For example, when someone is not able to fulfill his role. In this case, someone else needs to take over his role. Thus, the model must allow changes during the use of the system.

3.2 Access Control

For this thesis we use the definition of access control from RFC 4949[1]: "*Access control is a process by which use of system resources is regulated according to a security policy and is permitted only by authorized entities (users, programs, processes, or other systems) according to that policy.*" A policy within the Ministry of Defense is, for example, *crew members of a ship are allowed to read the crew list.* Within the Task-Based Methodology, a policy can be: *only the Commanding Officer is allowed to execute task "fire missile".*

[1] https://tools.ietf.org/html/rfc4949

All access control models assume that there are data administrators and data owners who define the access control specifications. When taking the Task-Based Methodology into account, this can be seen as administrators who are setting the policies, which is a task. The intent is that they should be restricting a *performer*, which is a user or a system, from executing a task under a particular condition. This performer is comparable with the term *subject*, which is mostly used in the access control model we discuss later. The administrators should be applying the principle of *least privilege* to minimize damage from intrusions [6]. This principle requires that every performer can only execute tasks that are necessary for its legitimate purpose.

Before we start looking at the access control models, we summarize the commonly used definitions in Table 3.1. Afterward, we discuss multiple access control models and describe how they can be used within the Task-Based Methodology. This is done by using military roles and scenarios.

Subject	An entity that can perform actions on the system, such as a person, group or system.
Object	An entity representing resources to which access may need to be controlled. For example files, directories, devices, resources, records, tables, processes, programs, networks or information.
Attribute	The characteristics of the subject or object. For a person, this can be a name, date of birth, or home address. For a file, this can be a name, date of last modification, size in bytes, or format.
Environmental conditions	Operational or situational context in which access requests occur. Environmental conditions are detectable environmental characteristics. Environmental characteristics are independent of subject or object and may include the current time, location of a subject, or the current threat level.
Action	In general, this can be reading, writing, editing, deleting, copying, executing, or modifying an object. In the Task-Based Methodology, this is the task itself.
Access rights	Granting permission to perform an action on an object.
Policy	The representation of rules or relationships that makes it possible to determine if requested access should be allowed, given the values of the attributes of the subject, object, and possibly environment conditions.

Table 3.1: Commonly used definitions about access control

3.3 Access Control Models

3.3.1 Access Control Matrices (ACM)

Access Control Matrices [6] [7] are a useful tool for determining access control rights. An ACM, in general, is a table that defines permissions. Each row is associated with a *subject* and each column is associated with an *object*. Each cell is filled with the access rights for the associated combination of subject and object. Table 3.2 gives an example of an ACM. An empty cell means that there are no access rights granted.

	file 1	directory 1	device 1	resource 1
person 1	read	read		
person 2	read, write			
group 1	execute			
group 2	read			
system 1		read		
system 2				

Table 3.2: General example Access Control Matrix

Advantages One of the advantages is that it is quite easy to check a cell for access rights and, thus, whether a subject is allowed to perform the action it is requesting. Just check the cell for a particular subject and object.

Another advantage is that the ACM gives administrators a simple, visual way of seeing the entire set of access control relationships all at once.

Disadvantage The main disadvantage is that within this access control model, there is no possibility to use the current time or location. It is also not possible to use the type of device of the subject.

Another disadvantage is that the ACM can get very big. If we have n subjects and m objects, then the ACM has $n \cdot m$ cells. Imagine a computer server with 1,000 subjects and 1,000,000 objects. This would imply an ACM with 1 billion cells. Nobody would be able to fill in all those cells and to view these all at once.

Using the ACM model in the Task-Based Methodology When trying to use this access control model with tasks, we see that a subject can be seen as a performer, and an object remains the same. Tasks can be added to the list of access rights on a particular object to make sure that a performer is allowed to perform a task on an object. When a performer is allowed to execute a task on a particular object, this can be added to the ACM.

This model can be used within the Task-Based Methodology, but the disadvantage remains. When determining which performers are allowed to execute which tasks and put that in a matrix, it would still become enormous.

3.3.2 Access Control Lists (ACL)

The *Access Control List* model [6] [7] takes an object-centered approach. It defines, for each object o a list L, which is called o's ACL. This list enumerates all the subjects that have access rights for o and for each subject s gives the access rights. This model takes each column of the ACM and compresses it into a list by ignoring all the subject-object pairs which are empty. When we change the ACM of Table 3.2 into an ACL we get Figure 3.2 where r, w and e are read, write and execute respectively.

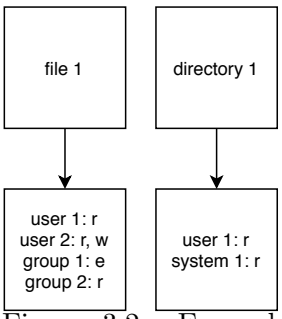

Figure 3.2: Example Access Control List

Besides the object-centered approach, there is also a subject-centered approach, called *capabilities*. However, this works the same as the ACL, but instead of a list for each object, there is a list for each subject.

Advantages One of the advantages of ACLs over ACMs is the size because the empty cells from a matrix are not in the ACL. Thus, the size of an ACL is much smaller than an ACM.

Another advantage is that the ACL of an object can be stored directly with that object as part of its metadata. So, when an operating system is trying to decide if a subject has the access rights it is requesting, it only has to check the ACL of that object.

Disadvantage The main disadvantage is that within this access control model, there is no possibility to use the current time or location. It is also not possible to use the type of device of the subject.

Using the object-centered or subject-centered approach has the disadvantage that in the first case, when a subject needs to be removed from the system, it would have to search for all the ACLs of every object and remove the subject from the lists. For the subject-centered approach, the same holds when an object needs to be removed from the system.

Although it is possible to use both the object-centered and subject-centered approach, it requires keeping the lists synchronous, which might take a lot of time [6].

Using the ACL model in the Task-Based Methodology In this access control model, the subject can be seen as a performer, and we can extend the list of access rights with tasks. This enables that a performer can only execute a task on a particular object if he is allowed.

The object-centered approach can be mapped into a task-centered approach. See Figure 3.3, where each task has a list of performers who are allowed to execute the task.

The subject-centered approach can be mapped into a performer-centered approach. See Figure 3.4, where each performer has a list of tasks they are allowed to execute.

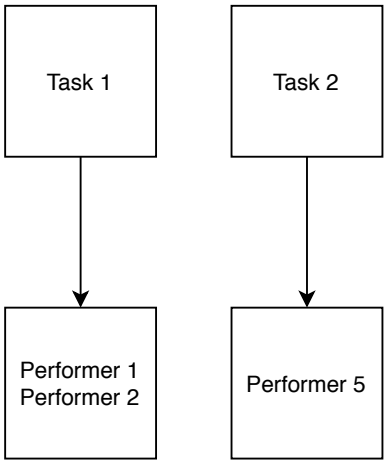

Figure 3.3: Task-centered approach

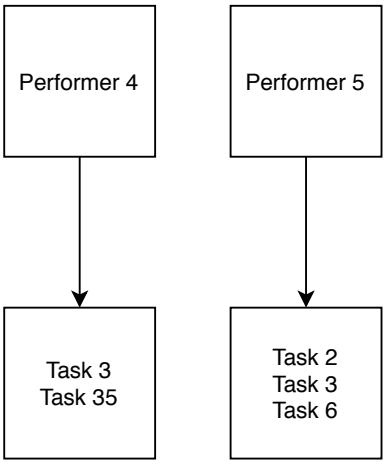

Figure 3.4: Performer-centered approach

The task-centered approach is a better fit to use compared to the performer-centered approach. To us, it makes more sense to determine which performers are allowed to execute a particular task because this can be done when defining the task.

3.3.3 Role-Based Access Control (RBAC)

With *Role-Based Access Control* [5] [6], an administrator defines *roles* and then specifies the access rights for these roles, instead of for subjects directly. In Table 3.3, we give an example of roles during the normal daily operations on a ship of the RNLN. These roles apply during the "zeewacht" (when on sea) and "reewacht" (when ashore). Crew members can have multiple roles, for example, a supervisor is also a crew member.

Role	**Access Rights**	**Notes**
Administrator	Read the crew list	Defines the roles based on the crew list and specifies the access rights for these roles, as described below
	Distributes the roles from the "scheepsrollenplan" to crew members who are actually on board[2]	
Crew member	Read the crew list	Every crew member has this role
	Read the availability and Guard list[3]	
	Change own and other's availability	
Administrative crew member	Change the crew list	This role is connected to a function on the ship
Crew member on duty	Change the Guard list	This role differs every day, based on the scheduled Guard
Supervisor	Change the status (details) of team members.	Someone who is the leader of a team. Status can be available, sick or limited employable. The status detail show the reason of unavailability
Non-crew member		This role is not part of the crew

Table 3.3: Roles during normal daily operations on a ship

Once the roles are defined, and access rights are assigned to the role-object pairs, subjects are assigned to various roles. The access rights for a subject is the union of the access rights for the roles that they have. A crew member who is on duty would have the union rights of both roles "crew member" and "crew member on duty".

[2]Crew members of a ship may not be actually on board for various reasons, i.e., illness, childbirth or personal education

[3]Contains the crew members who are on Guard

Role Hierarchies In addition to the RBAC model, a hierarchy can be defined over roles such that access rights propagate up the hierarchy. If a role r_1 is above r_2 in the hierarchy, then r_1 inherits the access rights of r_2. In other words, it means that the access rights of r_1 include those of r_2. In the example of table 3.3, the role "supervisor" would be above the role of "crew member". Thus, the role of "supervisor" would also get the access rights of the "crew member" role.

Advantages One of the advantages is that it is possible to separate subjects from objects directly by assigning roles to objects and subject to roles, which is currently being used within the RNLN as explained in Section 2.3.

Another advantage is that determining whether a subject is allowed to perform a particular task is relatively easy, just checking if the current role of the subject contains that task.

Another advantage is that when using role hierarchies, the storing of tasks becomes even more efficient, and it is natural when looking at the military.

Disadvantages The main disadvantage is called *role explosion*. Due to the increasing number of roles, it might be hard to manage all those roles. Elliot and Knight [4] challenge this notion, that the number of roles far exceeds the subjects found in enterprise systems. They explain why role explosion occurs in medium to large organizations employing RBAC. Furthermore, they introduce a role-centered approach for dynamically constraining access to data and their concept for *managed role explosion* in medium to large organizations, which is making use of role hierarchies.

Another disadvantage is that within this access control model, there is no possibility to use the current time or location. It is also not possible to use the type of device of the subject.

Using the RBAC model in the Task-Based Methodology In Section 2.3 we explained what an "arbeidsplaats" is. Each "arbeidsplaats" has its accompanying access rights and tasks that can be used in the Task-Based Methodology. Beside the "arbeidsplaatsen" we can also use the roles from the "scheepsrollenplan" as showed in Table 2.1. Each "scheepsrol" consists of multiple roles who are each assigned to a crew member. Since each role has its accompanying access rights and tasks, we can use this in the Task-Based Methodology.

3.3.4 Attribute-Based Access Control (ABAC)

ACLs and RBAC are special cases of *Attribute-Based Access Control* [8] in terms of the attributes they use. ACLs work with an object-centered or subjected-centered approach and RBAC with roles. The difference with ABAC is the concept of policies that combine attributes. This access control model supports boolean logic, in which rules contain "If ... Then" statements about which subject requests access, the object it applies to, and the action that must be executed.

In general, ABAC avoids the need for operations and objects to be directly assigned to subjects or their roles or groups. Instead, when a *subject* (1) requests access, an *Access Control Mechanism* (ACM) determines what operations the subject may perform upon the *object* (3). This decision is based on *policies* (2a) that are specified in terms of attributes and conditions, assigned *subject's attributes* (2b), *object's attributes* (2c) and *environment conditions* (2d). In this way, policies can be created and managed without direct reference to potentially numerous subjects and objects, and subjects and objects can be added without reference policy.

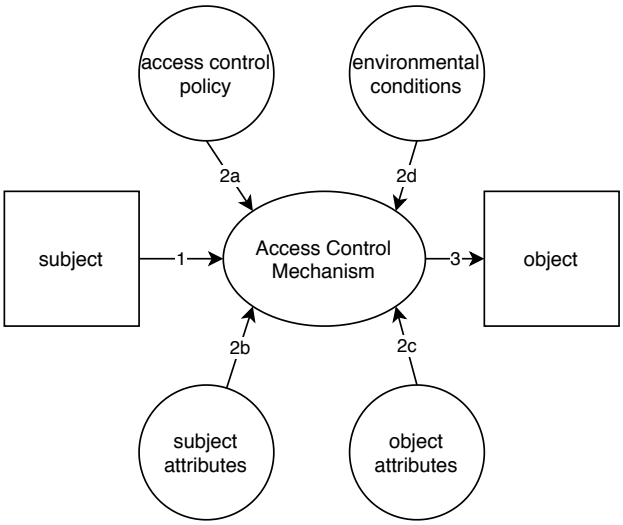

Figure 3.5: Attributes Based Access Control

Advantages The main advantage is that, in contrast to the other access control models, this model can deal with environmental conditions, such as location and time. And the type of device of a subject can be seen as an attribute of the subject. ABAC avoids the need for explicitly assigning access rights to roles and subjects to roles. This model also enables flexibility in a large enterprise where management of ACLs or RBAC would be time consuming and complicated.

Disadvantages The main disadvantage is that it might be difficult and time-consuming to define all policies. However, this does not have to be a significant disadvantage because a complex C2 system is always being improved and modified. This makes the ability to alter or add policies a much more substantial role than making sure all policies are defined from the start.

Using the ABAC model in the Task-Based Methodology This access control model can be used within the Task-Based Methodology when we map a subject unto a performer. The attributes in Figure 3.5 are characteristics of the performer and object. Within this model, policies need to be defined to make sure a performer who grants access to an object is allowed to get access. This model allows the usage of environmental conditions. By environmental conditions, we mean the "gereedheidsgraden" and "scheepsrollen" (situation dependent groupings) on a ship that are explained in Section 2.3.

Access control policies These policies are statements that bring together a performer, object, the attributes of both, and environmental conditions to express what is (not) allowed. Policies can be granting or denying policies. Policies can be local or global and can be written in a way that they override other policies. Below a few examples are given where the numbers match the number of Figure 3.5.

- A specific performer (1) is allowed to enter the compartment (3) (space/-room on a ship) he needs to be working in.
- A specific performer (1) is allowed to access the documents (3) he needs to perform his tasks.
- A specific performer (1) is allowed to operate the helicopter refueling hose (3) only when flight grouping (2d) is in action.
- A specific performer (1) is allowed to fire the ship's missiles (3) during "gevechtswacht" (2d).

3.3.5 Task-Based Access Control (TBAC)

Thomas and Sandhu [11] developed a paradigm for access control and authorization management in computerized information systems, called *Task-Based Authorization Controls*. TBAC models access control from a task-oriented perspective instead of the traditional subject-object ones like ACMs and ACLs. This TBAC approach was motivated by the need to automate authorization and related access controls. Although the core concepts of TBAC are discussed, the languages to model authorization policies and the runtime mapping of these policies to enforcement mechanisms are not mentioned in this research.

Wang and Zhang [13] proposed a task-based access model for workflow by introducing the notion of tasks into RBAC. They take tasks as a group of permissions and divided tasks into two classes, common tasks, and professional tasks. The common tasks are assigned to the organization unit to which all employees belong who are allowed to execute the task. The professional tasks are assigned to a particular role in a particular organizational unit.

In Section 3.3.3, we already mentioned that using RBAC it is possible to assign an "arbeidsplaats" to a crew member. It is also possible to assign crew members to roles from the "scheepsrollenplan". Since this is similar to the task-based access model of Wang and Zhang, this paper does not add much to our research.

3.4 Recommendation Extended Task-Based Methodology

The advice is to use the ABAC model. We take ABAC as the basis because it can deal with environmental conditions. We interpret the "gereedheidsgraden" and "scheepsrollen" (situation dependent groupings) explained in Section 2.3 as environmental conditions. When such an environmental condition is on action, the crew needs to deal with it. To be able to do that, all necessary tasks for a certain environmental condition are divided over a set of roles for that specific environmental condition. Next, for each environmental condition, every role is assigned to a crew member. This results in a matrix (i.e. "scheepsrollenplan") with environmental conditions on one axis, crew members on the other axis, roles on the intersections. An example is given in Table 2.1.

Explicitly defining policies for each role onboard is not necessary because this is already done within the RNLN. Each role in a "scheepsrollenplan" and each "arbeidsplaats" have their accompanying tasks and access rights. These tasks and access rights are used to check if a crew member is allowed to perform a certain task.

Beside the roles from the “scheepsrollenplan” and the “arbeidsplaats” we also use the role types mentioned in Section 3.1. Using these enables, for example, giving access rights to an entire department instead of every crew member individually.

Chapter 4

Model Selection and Implementation

This chapter explains how the adapted ABAC model looks like and explain all the different components of the model, together with examples in Section 4.1. In Section 4.2, we explain how our adapted ABAC model can be added to the Task-Based Methodology.

4.1 Adapted ABAC model

We start with the ABAC model as described in Section 3.3.4 and shown in Figure 3.5 and adapt it. We use the term performer instead of the subject because the Task-Based Methodology uses performers. We add *spatio-temporal constraints* to the model, which represents the current time and location (of the performer and object). Because current time and location are different from the environmental conditions "gereedheidsgraden" and "scheepsrollen", we decided to add these to the model instead of interpreting it as a subset of the environmental conditions.

We decided not to add the *type of device* to the model, but interpret it as an attribute of a performer because it is something the performer owns. Then the model looks like:

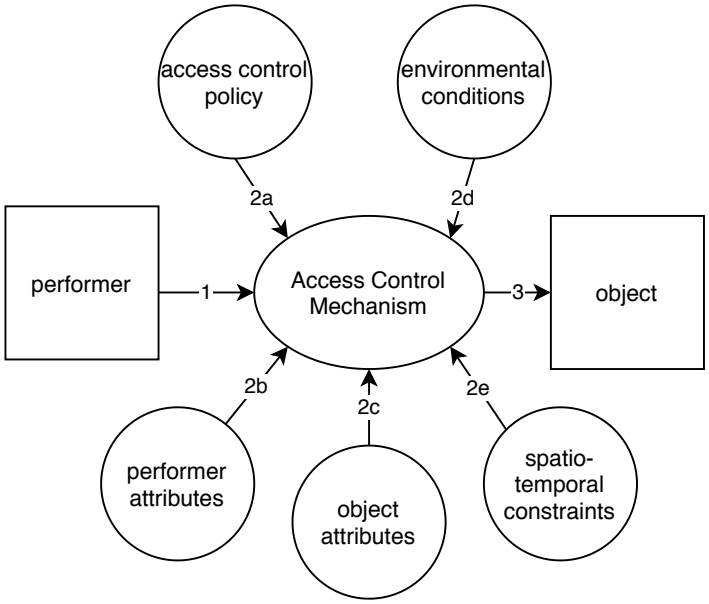

Figure 4.1: Adapted the subject unto a performer and added spatio-temporal constraints to the ABAC model

Below we discuss the different parts of the adapted ABAC model and give a few examples.

1. Performer
   - Person (i.e. a crew member)
   - System component
2. The *Access Control Mechanism* evaluates the access control policy (2a), performer attributes (2b), object attributes (2c), environmental conditions (2d) and spatio-temporal constraints (2e) to compute a decision
   a. Access control policy
      This is explained later on.
   b. Performer attributes
      - Person: employee number, certificates, current device using, list of other devices the person owns[4], rank, "arbeidsplaats", role(s), department(s)
      - System: type, manufacturer, date of delivery, software version, purpose

[4]This is helpful when a person wants to execute a task on his current device, but it only works on another device. He will be notified about which device he should use

c. Object attributes
    - Compartment: compartment type, compartment number
    - System: type, manufacturer, date of delivery, software version, purpose
    - Information: information type, source, classification, date of modification, date of creation, modified by whom, size

d. Environmental conditions - these are divided into "gereedheidsgraden" and "scheepsrollen" (situation dependent groupings) and are both discussed in Section 2.3
    - "Gereedheidsgraden"
        5. "Reewacht"
        4. "Verlichte Zeewacht"
        3. "Zeewacht"
        2. "Oorlogswacht"
        1. "Gevechtswacht"
    - "Scheepsrollen" (situation dependent groupings)
        - Buddy check grouping
        - Maneuver grouping
        - Replenishment at sea grouping
        - Flight grouping

e. Spatio-temporal constraints
    - Current time
    - Current location of the performer
    - Current location of the object

3. Object
    - Compartment (a space/room on a ship)
    - System component
    - Information

Access Control Policy The Access Control Policy that we use is derived from the "scheepsrollenplan" (explained in Section 2.3) on board of a ship. For each cell in the "scheepsrollenplan", a set of tasks and access rights is captured by the RNLN. A few examples from the "scheepsrollenplan" in Table 2.1 are elaborated below together with its task and access rights:

- Navigational assistant: his task is to navigate the ship, and thus he has access to the user part of the navigation system, not the technical part.
- Baker: his task is to prepare bread, pastries or cake on the ship and thus is allowed to access the stocks.

Besides the "scheepsrollenplan", tasks and access rights are also granted based on the different role types that are listed in Section 3.1. Below we give examples per role type together with access rights and tasks of that role.

- Department - every crew member of a department is linked to that department
  - Technical weapons department: these crew members have as a task to maintain the weapons on the ship. Thus, they are allowed to access the technical part of the weapon systems and classified information that is needed for maintenance.
  - Nautical department: tasks of these crew members are, amongst others, mooring, anchoring, and replenishment at sea. Thus, they have access to the tools they need to execute these tasks and for the maintenance of this gear.
- Authorization profile - individually assigned to a crew member
  - Multiple crew members need access to a particular system and compartment. These access rights can be clustered to a group. Then, this group can be assigned to the crew members who need these access rights.

4.2 Combining ABAC Model with Task-Based Methodology

In this chapter we explain how the adapted ABAC model of Figure 4.1 can be added to the current Task-Based Methodology we discussed in Section 2.2. Recall that a task is defined with an input, precondition, post condition, and output, as shown in Figure 4.2.

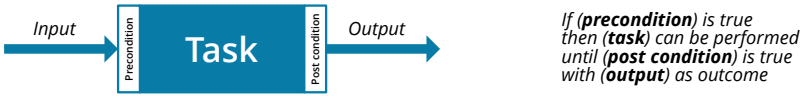

Figure 4.2: A task defined with input, precondition, post condition and output [3]

The precondition, as shown in Figure 4.2, makes sure that a task can only be performed if the precondition holds true. We add to this precondition a condition about when someone is allowed to perform that task. Since this condition needs to be checked before the task is being performed, we decided to add this to the precondition of a task and call this condition *access control preconditions*. These access control preconditions can change during the use of the system. This is in contrast to the essential preconditions of the task itself. This will have to be taken into account during implementation (sort of runtime injection mechanism).

The access control preconditions can be based on the role types mentioned in Section 3.1 or on the roles of a "scheepsrollenplan" (situation dependent grouping plan) mentioned in Section 2.3. These access control preconditions can also be based on environmental conditions, spatio-temporal constraints, and the attributes of both the performer and object. Below we elaborate two examples to make clear how the adapted ABAC model can be used within the Task-Based Methodology. These examples are formulated in such a way that those are realistic tasks.

For each task, we mention the preconditions which state when the task can be performed. Besides this, we also mention the post condition, which holds true after the successful completion of the task. We also mention the access control preconditions that states when someone is allowed to perform the task. For each task, it is also mentioned which components of the adapted ABAC model of Figure 4.1 are used. Note that the access control policy component is described in Section 4.1.

We assume that a mechanism is in place that checks the authorization of a performer.

The first task is: *open door nautical gear compartment* and is shown in Figure 4.3 together with its preconditions and post condition. The precondition *door is closed* checks if the task can be performed. In addition to this, access control can be added in the form of a precondition to make sure that the one granting access is allowed to perform the task. The *access control precondition* for this task is: *crew member is part of the nautical department or "gevechtswacht" is in action.* This precondition means that only a crew member of the nautical gear compartment is allowed to open the door of the compartment, but during "gevechtswacht" every crew member is allowed to open the door of the compartment.

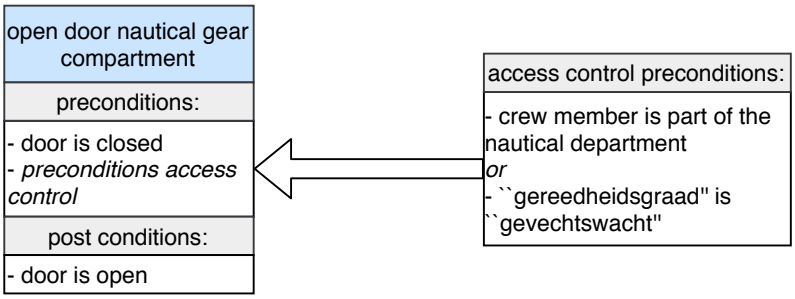

Figure 4.3: Task *open door nautical gear compartment*

Taking the task *open door nautical gear compartment* and the adapted ABAC model (Figure 4.1), Figure 4.4 shows which components of the ABAC model are being used. The performer of this task is a *crew member* and wants to perform this task on the object *door nautical gear compartment.* The performer has the attribute *nautical department* and the object has as attribute the compartment type *nautical gear compartment.* The environmental condition used is the "gereedheidsgraad" *"gevechtswacht"*. The spatio-temporal constraints are not used within this task.

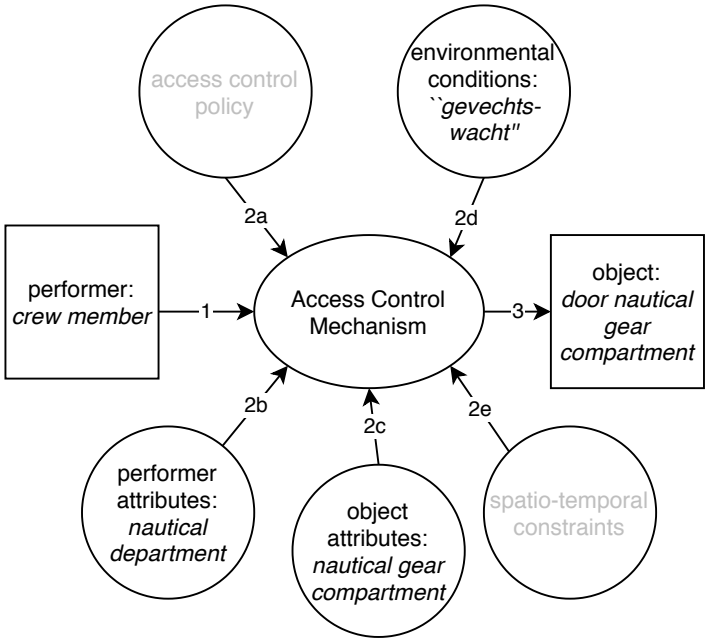

Figure 4.4: ABAC component used for task *open door nautical gear compartment*

The second task is: *fire missile for training purposes* and is shown in Figure 4.5 together with its precondition *missile is ready*. The precondition *missile is ready* checks if the task can be performed. In addition to this, access control can be added in the form of a precondition to make sure that the one who wants to perform the task is allowed to perform the task. The *access control precondition* for this task is: *current location is a practice area, and the commanding officer gives his permission, and crew member is air defense officer*. This precondition means that only in a practice area when the commanding officer has given his permission, the air defense officer is allowed to fire a missile.

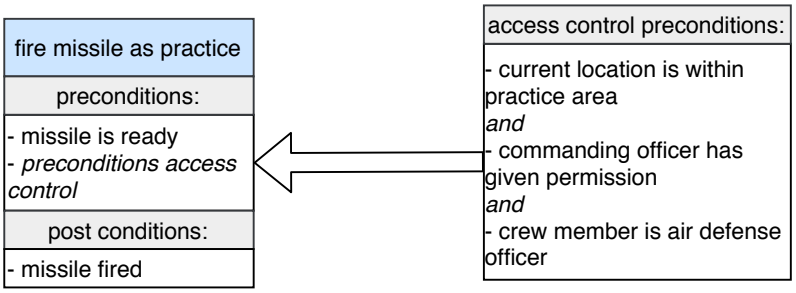

Figure 4.5: Task *fire missile for training purposes*

Taking the task *fire missile for training purposes* and the adapted ABAC model (Figure 4.1), Figure 4.6 shows which components of the ABAC model are being used. The performer of this task is a *crew member* and wants to perform this task on the object *missile*. The performer has the attribute *air defense officer*, which is his role. The object has as an attribute: *permission of commanding officer*. When the commanding officer does not give this permission, the task can not be executed. The spatio-temporal constraint used is the *current location*, which must be in a practice area. The environmental conditions are not used within this task.

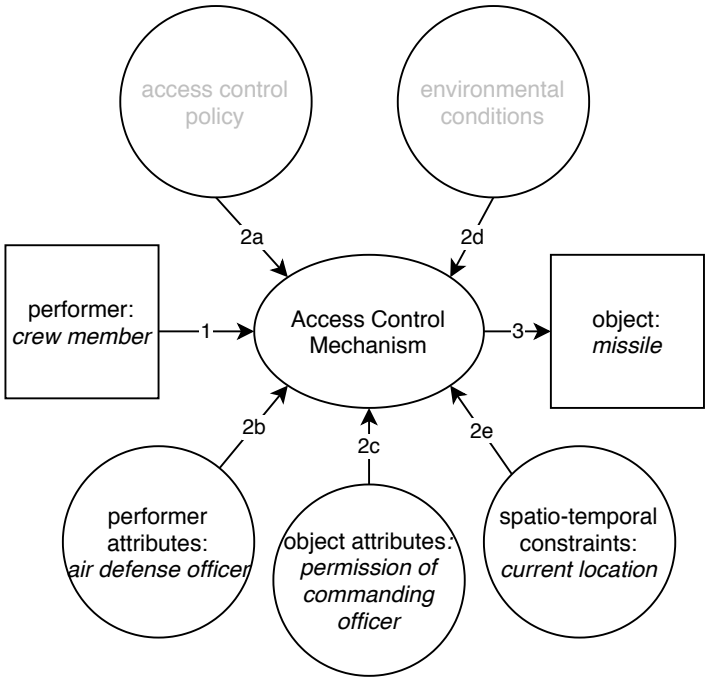

Figure 4.6: ABAC component used for task *fire missile for training purposes*

Chapter 5

Conclusion

This research investigated to what extent it is possible to extend the Task-Based Methodology such that the corresponding model contributes to "secure by design" in the new C2 systems by using access control.

We investigated the ACM, ACL, RBAC, ABAC, and TBAC access control models in Chapter 3. These access control models were analyzed by taking the following requirements into account: the model can deal with current location and time, the model needs to be flexible in usage, and it must deal with the type of device of a performer, such as a phone or a tablet.

From this, we concluded in Section 3.4 that ABAC gives us what we needed. We have chosen for this because ABAC can deal with the environmental conditions such as "gereedheidsgraden" and "scheepsrollen" (situation dependent groupings). These environmental conditions are both part of the "scheepsrollenplan" of a RNLN ship. An example is given in Table 2.1. This plan contains per environmental condition a set of roles divided over the entire crew on board. Each role has its accompanying tasks and access rights and is assigned to a crew member.

Besides the environmental conditions, we added spatio-temporal constraints, which enables the usage of current time and location (of the performer and object). The adapted ABAC model can be found in Chapter 4.

A task in the Task-Based Methodology [3] is defined together with pre- and postconditions as described in Section 2.2. The precondition states when the task can be performed. The post condition indicates when the task is accomplished. We add to this *access control preconditions* to define when someone is allowed to perform the task.

When the access control preconditions are defined for a task, it is a matter of checking the access rights of the performer who wants to execute that task. Based on his roles and accompanying access rights, a decision can be made whether he is allowed to perform that task.

Since the research we have done was focused on the RNLN and access control, we only reviewed five specific access control models while there are more known.

As a future project, we can implement the proposed extension in the Task-Based Methodology.

Acknowledgements

I would like to sincerely thank everyone that has helped me with writing and gave me feedback, especially Peter Achten and Erik Poll. I would also like to thank my supervisor at Maritime IT, Michael de Vos, and my colleagues.

Bibliography

[1] David S. Alberts and Richard E. Hayes. *Understanding command and control.* Ministry of Defense of The United States, Jan 2006.

[2] Hoofd bureau Scheepsbedrijfsvoering bij STC. *Voorschrift Commando Zeestrijdkrachten, Directie Operaties 170.5, Standaard Orderboek Commandant Grote Bovenwatereenheden "Rollenplannen".* Ministry of Defence of The Netherlands, 2017. Only available within the Ministry of Defense.

[3] Michael de Vos. Toward a Knowledge Preserving, Task Based Modelling Approach for Command and Control. 2020. Nederlandse Defensie Academie.

[4] Aaron Elliott and Scott Knight. Towards Managed Role Explosion. *Proceedings of the New Security Paradigms Workshop on ZZZ - NSPW 15*, 2015.

[5] David Ferraiolo and Richard Kuhn. Role-based access control. *Proceedings of the 15th National Computer security Conference*, pages 554–563, 1992.

[6] Michael T. Goodrich and Roberto Tamassia. *Introduction to computer security.* Harlow, Essex: Pearson Education Limited., 2014.

[7] G. Scott Graham and Peter J. Denning. Protection - Principles and practice. *Proceedings of the Spring Joint Computer Conference*, pages 417–429, 1972.

[8] Vincent C. Hu, David F. Ferraiolo, D. Richard Kuhn, Arthur R. Friedman, Alan J. Lang, Margaret Mary Cogdell, Adam Schnitzer, Kenneth Sandlin, Robert Miller, and Karen Scarfone. *Guide to Attribute Based Access Control (ABAC) Definition and Considerations.* NIST Special Publication 800-162, 2014.

[9] Doctrine Branch Netherlands Defence Staff. *Command and Control - Joint Doctrine Publication 5.* Ministry of Defense of the Netherlands, 2012.

[10] Maritime Warfare Centre of the Royal Netherlands Navy. *Grondslagen van het Maritieme Optreden - Nederlandse maritiem-militaire doctrine.* Ministry of Defence of The Netherlands, 2014. Available at `https://www.defensie.nl/downloads/publicaties/2014/02/13/grondslagen-van-het-maritieme-optreden-nederlandse-maritiemmilitaire-doctrine`. Accessed on May 17, 2020.

[11] Roshan K. Thomas and Ravi S. Sandhu. Task-based authorization controls (TBAC): a family of models for active and enterprise-oriented authorization management. *Database Security XI: Status and Prospects*, page 166–181, Jan 1997.

[12] Marius S. Vassiliou, David Stephen Alberts, and Jonathan Russel Agre. *C2 re-envisioned: the future of the enterprise.* CRC Press, 2014.

[13] Baoyi Wang and Shaomin Zhang. An Organization and Task Based Access Control Model for Workflow System. *Advances in Web and Network Technologies, and Information Management*, page 485–490, 2007.