



Article

Towards a Reference Architecture for Cargo Ports

Virginia M. Romero and Eduardo B. Fernandez *

Department of Electrical Engineering and Computer Science, Florida Atlantic University,
Boca Raton, FL 33431, USA; vromero@techno-transfers.com

* Correspondence: fernande@fau.edu

Abstract: Cyber-Physical Systems (CPS) are physical systems whose operations are coordinated, monitored, and controlled by computing and communication functions. These systems are typically heterogeneous, including Internet of Things and information technology subsystems, and can present a myriad of implementation details, making them very complex systems. An important type of CPS is a maritime container terminal (cargo port), which is a facility where cargo containers are transported between ships and land vehicles for onward transportation and vice versa. A cargo port performs four basic functions: receiving, storing, staging, and loading for both import and export containers. We present here process patterns that describe the functional aspects of cargo ports and a pattern that describes their structural properties (patterns are encapsulated solutions to recurrent problems). These patterns describe semantic aspects found in any cargo port and can be adapted to describe other CPSs. We decompose these process patterns into use cases that describe their interactions with the system. We then integrate the process patterns with structural patterns to assemble a partial reference architecture (RA) that shows the interactions of all the patterns while also indicating the typical stakeholders found in all ports. We validate the proposed reference architecture, highlighting its theoretical and practical value. Software and system designers of cargo ports need to start from a conceptual and abstract view that is subsequently refined to add more details. The use of reference architectures and patterns is an effective way to organize and describe the functional and non-functional aspects of a system, as well as to unify the design of all its aspects. This is, until now, the only published RA for cargo ports, and it can be a useful guideline for the designers of any type of cargo port.



Citation: Romero, V.M.; Fernandez, E.B. Towards a Reference Architecture for Cargo Ports. *Future Internet* **2023**, *15*, 139. <https://doi.org/10.3390/fi15040139>

Academic Editors: Paolo Bellavista and Eirini Eleni Tsiropoulou

Received: 15 February 2023
Revised: 27 March 2023
Accepted: 30 March 2023
Published: 4 April 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: maritime transportation; cargo port; cyber-physical systems; IoT; software patterns; process patterns; design science

1. Introduction

Cyber-physical systems (CPSs) are systems that integrate computational resources and communication capabilities with the control of activities in the physical world. Another definition is [1]: “CPSs are integrations of computation with physical processes. Embedded computers and networks monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa”. The components of a CPS can be centralized or distributed and typically include embedded devices, sensors, actuators, and wireless links (see Figure 1). CPSs are often safety-critical because their failure could endanger lives or cause significant economic losses. CPS may include as components Internet of Things (IoT) and Information Technology (IT) subsystems and can be studied as a system of systems, although we do not do that here. CPSs can be structured into at least five architectural levels [1]: a process layer involving field instrumentation, a basic control layer, a supervisory control layer, a process management control layer involving application servers, and a corporate network layer. CPSs require a high level of adaptability because of continuously evolving conditions and need situation awareness and modifiability [2]. They may include legacy systems, and they increasingly include humans in the loop. Typically, they use combinations of commercial-off-the-shelf (COTS) components

and real-time operating systems, as well as products from different vendors using different protocols. Many CPS also need to follow government or industrial regulations [3]. The design of CPS systems requires consideration of several disciplines, such as embedded systems, computers, communications, and others, and the software is embedded in devices whose principal function is not only computation. However, we are concerned here only with software aspects that are pervasive in modern systems.

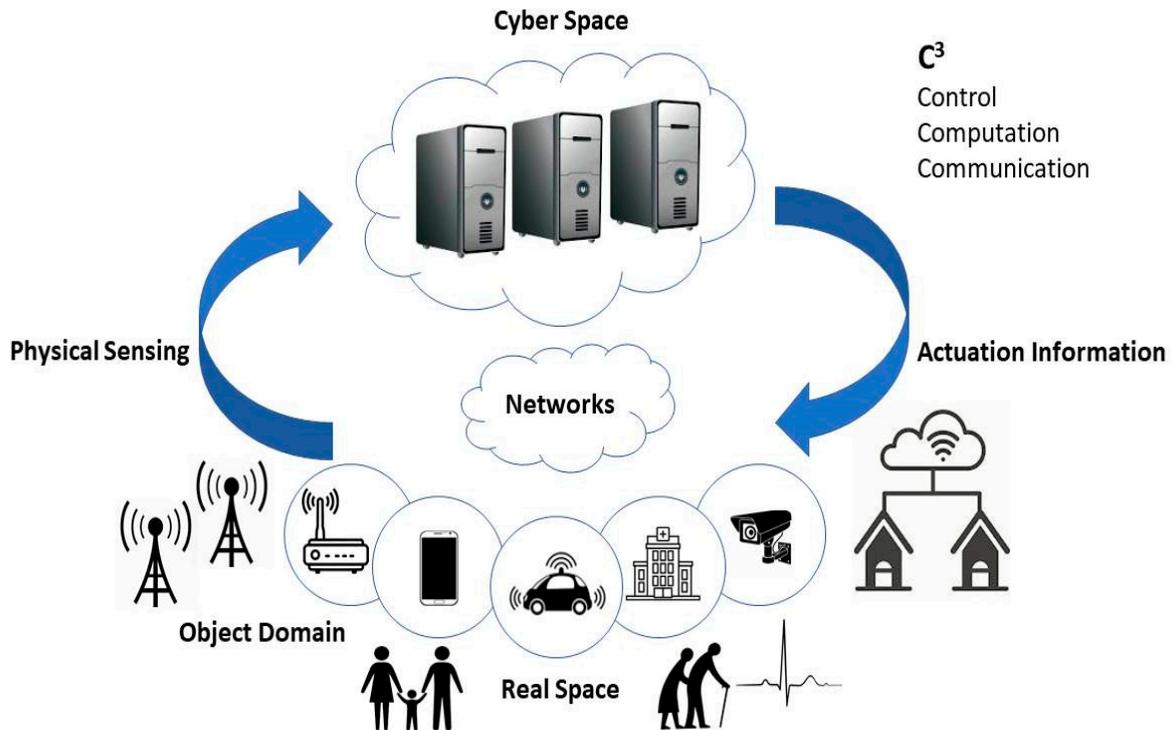


Figure 1. A cyber-physical system.

CPSs can be very complex systems and may comprise a multitude of systems, components, and implementations; they can be classified as massively interconnected, complicated systems [4]. To be able to understand their interactions and perform their design, we need to consider their global architectures and apply abstraction, focusing on their fundamental properties. A reference architecture (RA) is a generic software architecture, based on one or more domains, that does not include implementation details [5,6]. An RA is reusable, extensible, and configurable; it also provides guidelines for interoperability and portability. RAs can be seen as patterns for complete architectures and can be instantiated into concrete software architectures by adding platform aspects [5]. A software pattern is an encapsulated solution to a recurrent software problem and defines a vocabulary that concisely expresses requirements and solutions, as well as provides a communication vocabulary for the involved stakeholders [7,8]. A reference architecture built using patterns is a useful way for designers to describe the system and analyze security and other non-functional aspects. Specifically, for CPSs, these abstraction artifacts have become not only very useful to understand and building these complex systems but also have the potential to unify the design of the computational, communication, and control aspects of CPSs, especially in the presence of the many implementation details of their component units.

There are many types of CPSs, such as those in aerospace, automotive, chemical processes, healthcare, manufacturing, and transportation. Our objective is not to define an RA for any type of CPS, but we will consider a specific type of system, a transportation system, specifically a maritime container cargo port. As we discuss in Section 7, an analysis of the relevant literature found that there are not many reference architectures for cargo ports ([9] and Section 7). We show here a cargo port RA built of patterns described using

UML models. We consider this effort the first attempt to define a precise and semiformal architecture for these systems. We believe that an abstract, semiformal approach is the most practical design approach given the complexity of the systems we are considering. That is not to say that parts of the architectures cannot be formally modelled; UML models can be complemented with formal descriptions such as OCL [10], and with them, we can make this architecture more formal if needed [11]. Furthermore, patterns and RAs can be converted into ontologies, which are even more precise descriptions and allow for proving some formal properties [12]. Note that for CPSs, it is also possible to define reference architectures that include several levels [4,13,14]. The lower levels of those architectures have many common aspects, but we are concerned only with the semantic aspects of port operations, so we will focus only on the higher levels.

We first assemble a set of process patterns (a CPS process describes a physical process, e.g., the purification of water). A software process is more general; it describes a sequence of steps to accomplish some objective, e.g., teaching a course. Here, process patterns (which describe software processes) describe the functions performed by cargo ports. These patterns include *ship arrivals and departures*; *port drayage* (the transport of containerized cargo by specialized trucking/railway companies between a maritime port terminal and an inland distribution point or a rail terminal [15]); *container terminal physical structure* (the physical units used in the port); and *port loading/unloading* (the activities of loading and unloading containers) [16]. Each process pattern includes one or more use cases. A use case represents a complete interaction between the users and the system; each use case is associated with a set of actors that correspond to some of the stakeholders of the system. The complete patterns have been published in conferences; we show here only their important sections; some have been modified. These patterns result in a partial RA for a cargo port. We used this architecture as the basis for a security reference architecture, where defenses against identified threats have been added [17], so this article is a complement to that article.

We make it clear that we are not presenting a model for a specific port or type of port. Patterns and RAs are generic models intended for system and software designers. Our models are abstractions of fundamental aspects found by studying the structure and operations of several cargo ports around the world. Because of this, regulations, standards, and trade models (e.g., single-window strategy) are not relevant. A complete model is also beyond the possibilities of two people, so this is a partial model, and we do not claim this model to be complete.

Our contributions include:

- A set of process/structure patterns that describe the functional aspects of a reference architecture for cargo ports. These patterns describe semantic aspects found in any cargo port and can be adapted or extended to describe other CPSs.
- A decomposition of the cargo port process patterns into use cases that describe its actor interactions with the system. The use cases are dynamic models that complement the structural models of the port.
- A partial reference architecture that shows how to integrate these process patterns with structural patterns, indicating the typical stakeholders found in all ports. We know of no other reference architecture for cargo ports (see Section 7).
- A validation of the proposed RA that highlights its theoretical and practical value.

Although we describe only a partial RA, we show the method to build it, which can be used by others to complete it. From an academic point of view, this partial RA is sufficient, but in an industrial environment, it should be completed. Our main audience includes software architects working with CPS and port experts building cargo port systems. The rest of the paper is structured as follows: Section 2 presents some background information on patterns and reference architectures as well as a description of the basic aspects of container cargo ports and their operations. In Section 3, we introduce the process patterns and list their main actors and use cases (UCs). Section 4 describes in detail each process pattern, while Section 5 shows a partial RA for a cargo port using these patterns. Section 6

validates the RA, while Section 7 inspects related work and puts our work in perspective. We end with conclusions and future work in Section 8.

2. Background

2.1. Patterns and Reference Architectures

We present a summary of the concepts of software patterns and reference architectures, which are fundamental to understanding our work. A pattern is a solution to a recurrent problem in a given context; it embodies the knowledge and experience of software experts and can be used for new applications and to guide inexperienced designers. Patterns are described using structured templates that contain sections that define specific information about the problem being solved and its solution, along with recommendations about its use. A frequent way to describe a pattern solution comes in the form of a UML class diagram, complemented with some sequence diagrams that describe dynamic aspects of the use cases and possibly activity or state diagrams. A problem section defines a recurrent problem and the set of forces that constrain its solution. Once the solution is presented, a section on the consequences indicates how well the forces were satisfied by the solution as well as the possible negative aspects of it, such as the extra overhead. An implementation section provides hints on how to use the pattern in an application, indicating what steps are needed and possible realizations. A section on related patterns indicates other patterns that complement the pattern or that provide alternative solutions [18]. This level of detail and precision allows system designers and developers to use them as guidelines to build systems, and for users to understand the effect of the mechanisms they represent. Patterns are also useful for communication between designers and to evaluate and reengineer existing systems [7]. Patterns are applied to designs by instantiating them, which requires tailoring them to the rest of the design, e.g., removing some classes, changing names, or deleting attributes; that is, a pattern behaves like a type in a programming language. The use of patterns for system design is grounded in the principles of design science [19], and several pattern conferences each year introduce new patterns as well as applications of this design style.

There are several types of patterns, such as design and architecture patterns, that can be used to build flexible and extensible systems [7,8]. *Security patterns* are used to build secure systems by describing ways to control their threats, fix their vulnerabilities, and provide security attributes [20]. *Misuse patterns* describe how attacks are performed from the point of view of an attacker [21,22]. They define the environment where the attack is performed, what security mechanisms are needed as countermeasures to stop it, and how to find forensic information to trace the attack once it happens. *Process patterns* describe the workflows of the activities required to perform some business objective, such as order parts, schedule activities, or similar.

A reference architecture (RA) is a standardized, generic software architecture based on a particular domain that does not contain implementation details [5,6,23]. It can be seen as a type of pattern for whole architectures that can be instantiated into concrete architectures by adding platform aspects [24]. There is no general agreement about what an RA should contain [23]. Avgeriou [5] presents an example and describes what should be included in one. Some groups define other types of reference architectures, including hardware and other aspects, e.g., the Smart Grid Architecture Model (SGAM) [4,25] (see the section on related work), but we are not concerned with those types here; we think of this RA as a metamodel for concrete architectures defined as a PIM (Platform Independent Model) in an MDE (Model Driven Engineering) methodology.

Security patterns can be added to the RA to handle their identified threats, resulting in a security reference architecture (SRA). An SRA is an abstract architecture describing a conceptual model of security for systems and providing a way to specify security requirements for a range of concrete architectures. The SRA is an extended RA where security solutions have been added in appropriate places to provide some degree of security [17,26].

As indicated earlier in another paper [17], we have defined an SRA for cargo ports that complements this RA.

2.2. Cargo Ports

We present an overview of cargo ports as a general background on this type of port. As indicated, our models are abstractions of real ports, and we have not intended to describe every detail of ports.

2.2.1. Overview

For millennia, mankind has shipped goods across the oceans, from one land to another. The loading and unloading of a ship have always been very labor-intensive. A ship could easily spend more time in port than at sea while dock workers handled cargo in and out of tight spaces below decks. The introduction of containerization has greatly simplified this process, and nowadays goods can be moved seamlessly between ships, trucks, and trains [27]. The U.S. Bureau of Transportation reports that more than seventy-seven percent of freight tonnage entering the U.S. came by water, compared to 22 percent by land and only 0.3 percent by air [28]. By any measure, marine transportation is the primary means of moving goods and raw materials to and from the U.S. While the U.S. represents only 4.5 percent of the world's population [29], it accounts for 9 percent of the worldwide container traffic, with one container out of every eleven engaged in global trade either bound for or originating in the U.S. [30]. The largest amount of cargo traffic in the world is done by China [31]. The largest container ships can currently carry over 20,000 TEUs (a TEU is the volume of a 20 ft. container).

Port automation has been playing an increasingly important role with the introduction of robots, artificial intelligence, and other digital tools that keep the goods flowing into and out of major ports. This technology is widely seen as the most efficient way for seaports to cope with rising global shipping traffic and massive new ships that haul more and more containers. By digitizing and automating activities once handled by human crane operators and cargo haulers, seaports can reduce the amount of time ships sit in port and thus boost port productivity by up to 30%, according to some estimates [32]. IoT advances have had a strong impact on the automation of port activities, allowing the remote control of cranes and vehicles.

2.2.2. Container Terminal Overview

Container terminals provide many services, i.e., container loading/unloading to and from vessels and feeder ships for import or export purposes, internal container movement from ships to stacking areas and vice versa, stacking containers in dedicated areas distributed in the terminal area, container inspection for customs requirements, refrigerated containers handling and storage, etc. All the above processes require several shared and reusable resources and equipment to fulfill the tasks involved in handling and transporting containers, such as dock cranes, yard cranes, transport vehicles, yard stacking deposits, automatic stacking cranes or automatic storage/retrieval systems, railway tracks, and human operators. All processes and operations are usually planned, scheduled, monitored, and controlled by a central supervisor and make use of information technologies and IoT to allow fast shipping operations, optimize the usage of facilities, and reduce lag times [33].

A typical modern container terminal installation is represented in Figure 2 using UML package notation. It includes [34]:

- a. An area for port physical security and access control; this incorporates the physical entrance to the terminal, video surveillance, gates, and the infrastructure necessary to check the credentials of the maritime transportation workers requiring unescorted access to the secure areas of the port. Credentials are typically biometric identification cards, also known as TWICs (Transport Worker Identification Cards);
- b. The cargo handling equipment at shipside and landside, which includes the container terminal cranes, transport vehicles, and similar conveyances;

- c. An area for intermodal transportation of the containers, such as commercial long-haul trucks, railways, or even other ships;
- d. A terminal operating center, incorporates the financial, communications, customs, IT security, and other back office functions.

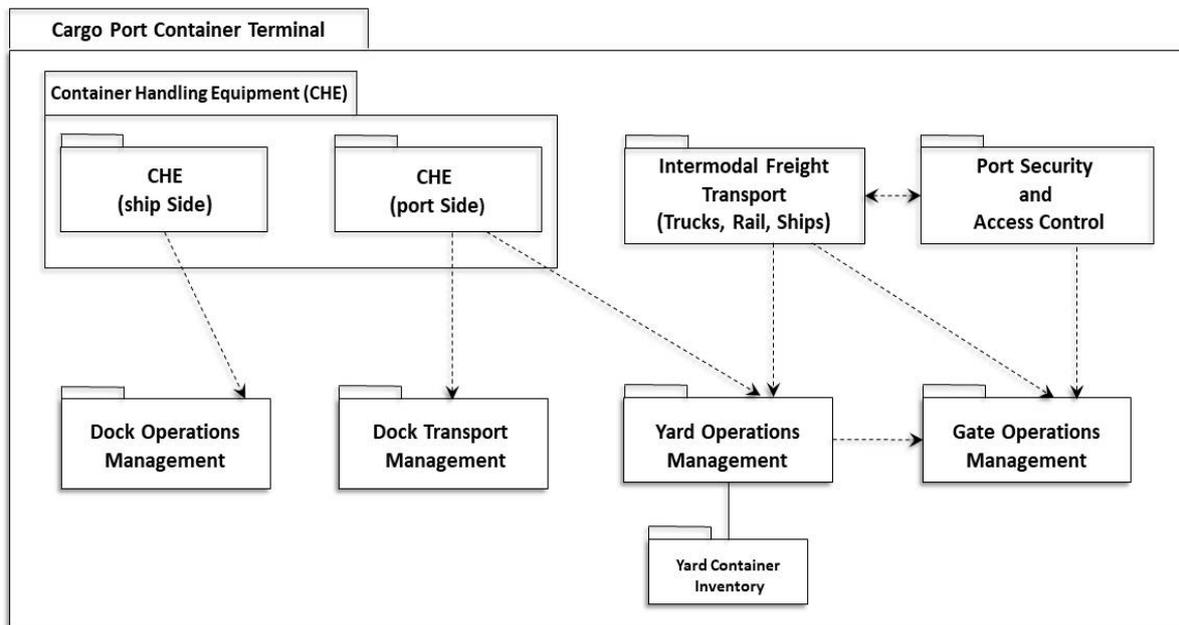


Figure 2. Cargo port system package diagram.

2.2.3. Container Terminal Operations

Figure 3 presents a modern container terminal [34]. Container terminals work day and night, and their workload depends on the number of containers in transit. Containers arrive at the terminal by trains, ships, or trucks and are stored in the terminal yard. Then, they leave the terminal by the same means to reach their final destination [35]. A ship-to-shore crane (STS) is a crane that services a container ship by shifting on a rail to reach the assigned stowage within the same ship and also to move from one ship to the next once the first one has been completed. STS cranes provide the single most important operation (called a move) associated with the loading and unloading of a ship and are the only means of moving containers to or from a ship [36]. To unload a ship, one or several STS cranes pick up containers from the ship and put them on trailers or shuttle trucks that move them to the assigned yard positions within the terminal storage area. Automated guided vehicles (AGV) are also used for the transport of containers from the dock to the yard. To load a ship, the STS crane unloads a container from the trailer and puts it in the ship. In automated or semi-automated container terminals, this operation is performed by crane operators controlling ship-to-shore cranes (STS) from a control room. Optical character recognition (OCR) technology is used to automatically read and record the container's ISO code identification number as it is handled by an STS crane. By automating the handoff of containers from cranes to vehicles, the process becomes safer for people as there is no need for checkers or tally workers under the cranes who are responsible for recording container numbers. More advanced features include damage images, door direction, hazardous labels, and reading and reporting the drayage truck numbers onto which the container is transferred [37].

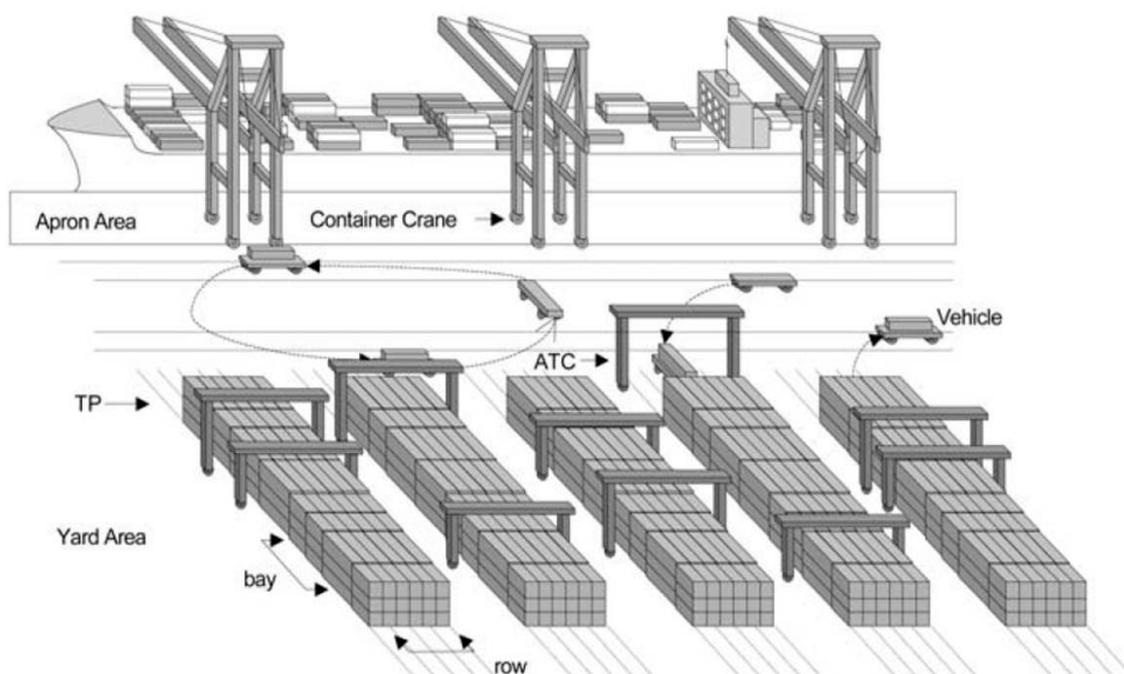


Figure 3. Modern container terminal.

Containers stacked in a ship are secured in several ways to prevent them from being damaged at sea. Locking corner castings are placed between stacked containers in non-cellularized ships to align the containers and provide a place to brace them. The cross braces are then secured to the floor of the ship, and, finally, the hatch covers are put back in place. Cellularized ships do not require corner castings or cross braces since permanent guides and locks (which allow containers to be stored more densely than in non-cellularized cargo vessels) are already on board [35,36].

Operations in the storage yard are more flexible than STS crane operations. In a storage yard, gantry cranes, top-pick loaders, or straddle carriers are used to stack containers. Automated stacking cranes (ASC) are also used. Other transport vehicles include automated lifting vehicles and unmanned shuttle carriers able to locate the containers and pick them up from the ground [37]. Automated vehicles may share a common software control module at the equipment level, often referred to as the equipment control system (ECS), for handling, for example, safety features and intra-vehicle coordination. An ECS is defined as the software that monitors and controls all events and processes at the equipment level, either for a single piece of container handling equipment (CHE) such as an AGV or a group of CHE. ECS coordinates interactions between different types of automated equipment [33,37,38].

The container storage area is usually separated into different stacks (or blocks) that are differentiated into rows, bays, and tiers. Some stack areas are reserved for special containers like reefers (refrigerated containers), which need electrical connections, dangerous goods, or overheight/overwidth containers, which do not allow for normal stacking. Often, stacks are separated into areas for import, export, and empty containers. Even with these many subdivisions, the efficiency of storage yard equipment is greatly increased by being able to store only one portion of the yard at a time. To prevent multiple restows or misplaced containers, the efficient assignment of the location or address of the container is of primary importance. Without efficient ways to assign container addresses, multiple restows are likely [39]. Containers are assigned specific addresses before entering the storage yard. In automated or semi-automated container terminals, intelligent stacking cranes operate using the optimum path, overlapping horizontal and vertical motion. Multiple ASC cranes in the yard are coordinated to avoid waiting times, so stacking cranes are able to deliver variable volumes of containers in a short time. Automated equipment is operated without

unnecessary accelerations and decelerations, and huge energy savings can be made. With intelligent automation, all the operations of the ACT can be comfortably handled from a central control room, which makes terminals safer and the working environment more ergonomic. Remote operation and on-board cameras provide better views than what is possible from a crane cabin in situations like landing containers on a ship or vehicle or handling hatch covers close to the ground [37,38].

Another important element of a container terminal is the movement of containers between the STS cranes and the storage yard. STS cranes unload a ship and place the containers in trailers or shuttle trucks. These shuttle trucks move the containers to storage locations in the yard. This operation is a closed loop, as their only function is to shuttle the containers from the ship to the storage yard. These trucks are local, and they usually do not leave the terminal. A collection of shuttle trucks is called a gang, and they may be automated guided vehicles (AGV) in larger terminals. The road network for these unmanned robotic transport vehicles is defined by electric wires or transponders in the ground, which enable the accurate positioning of these vehicles [40]. The number of shuttle trucks also needs to be considered. Too many trucks in the system cause long lines at the crane and long waiting times for service; conversely, fewer trucks in the system will result in idle stacking equipment. Containers, which are stored in the storage yard, leave the terminal on input/output trucks to reach their final destinations. Because of the high cost of keeping a ship in port, it is important to keep the dock crane operating without delay to turn the ship around as quickly as possible [38].

A great variety of container terminals exist, mainly depending on what type of handling equipment is used in the system. The decision on which equipment to use depends on several factors. Space restrictions, economic reasons, and even historical reasons play an important role. A basic factor is the size of the space that can be used for a terminal. If space is restricted, gantry cranes to store the containers are preferred. A decision for AGVs and automated gantry cranes can be made in cases of high labor costs and new terminal construction. Historical and cultural reasons must be considered if container terminals are enhanced or modernized. Because space is becoming a scarce resource, a tendency toward higher storage is to be expected [41].

3. Building a Reference Architecture (RA) for a Cargo Port

In this section, we present several process patterns of cargo ports used to build the architecture and their relationships; we define their use cases and identify their primary actors. A process pattern may contain several use cases.

3.1. Process Patterns

Figure 4 presents a pattern diagram relating some of the patterns we would find in a reference architecture for cargo ports (a pattern diagram shows relationships between patterns). This diagram includes three process patterns: *Ship arrivals or departures* describe the port traffic; *port drayage* delivers (from local sources) or receives (from ships) the containers to be stored in the yard, while *port loading/unloading* performs the ship container handling activities. The *container terminal's physical structure* describes the physical zones and buildings found in a container terminal, and it is an architecture pattern, not a process pattern.

Customs Manager, *Financial Services Manager*, and *Berth Assignment Manager* patterns are not shown for conciseness but are presented in this figure to give a broader perspective. The *Customs Manager* pattern may include use cases such as “inspect ship” and “inspect container”; the *Financial Services Manager* pattern may include use cases such as “bill for service to ship” and “bill for transportation services”; the *Berth Assignment Manager* pattern may include use cases such as “receive cargo manifesto” and “assign berth”.

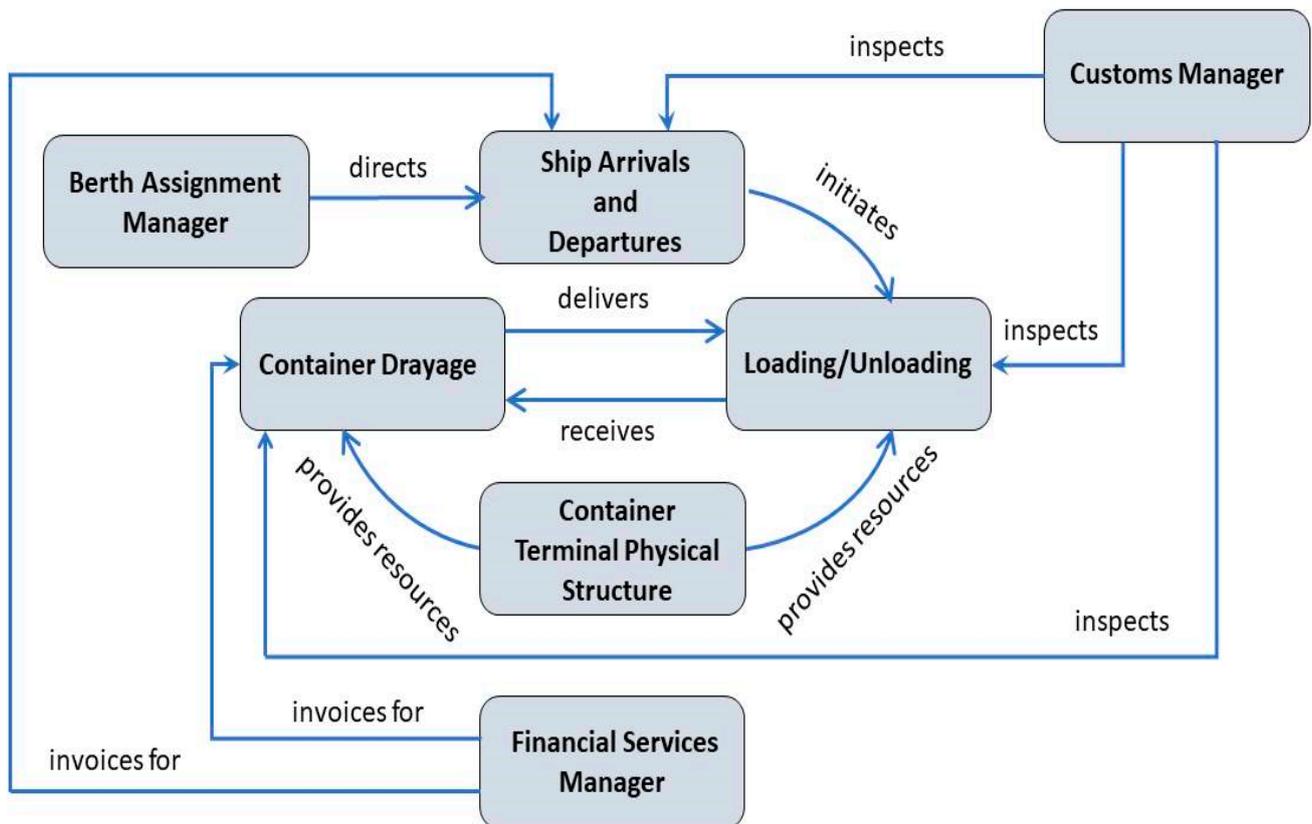


Figure 4. Relationship of cargo port patterns.

3.2. Use Case Model

A use-case model shows how users interact with the system to solve a problem [42]. A use case represents a complete interaction between a user and the system and describes how the system should respond under various conditions to requests from the user [43]. Each main function in a cargo port is represented as a use case, and each use case is associated with an actor or a set of actors. An actor represents a user or automated system that may interact with the system [42,44]. Generally, an actor is a role rather than a specific person; an actor can be distinguished through his tasks, and a single actor could be associated with one or more use cases and vice versa [44,45]. In these models, actors correspond to some of the stakeholders (see Section 3.3). Figure 5 presents a general overview of the system functions using a UML use case diagram, where the use cases are related to the process patterns (in color). In automated container terminals, use cases UC5 (*Assign Trailer Driver*) and UC10 (*Assign Crane Operators*) (shaded in purple) in Figure 5 may not apply, as these activities are substituted by automated guided vehicles (AGVs) and unmanned container terminal cranes. Use cases not related to the patterns in this figure are part of other cargo port patterns not covered here.

3.3. Stakeholders

Following is a description of the stakeholders (we show only the stakeholders relevant to the use cases described here and according to the level of abstraction of this paper). A more complete port description would have more of them in a cargo port, some of which are also actors in the use cases we will show later:

1. **Port Authority:** responsible for the overall administration of the property, terminals, and other facilities on the port complex.
2. **Port Operations Manager:** responsible for the efficient use of port facilities and resources with specific responsibilities for health, safety, and security. The individual

- works closely with regulatory authorities, port operations, and personnel of shipping lines. Manages resource allocation within the terminals and monitors expenditures.
3. **Terminal Operator (leasing company):** legal entities that lease from port authorities their terminal facilities. Either the port authority or the terminal operator will supply the cranes and other cargo-handling equipment. It depends on the lease agreement between the port authority and each terminal operator.
 4. **Terminal Operations Manager:** responsible for the planning and administration of the operations at a port terminal in order to optimize resource use, minimize costs, and maintain quality standards. Activities associated with the transportation of cargo include the operation of vessels, cargo handling equipment, locomotives, trucks, vehicles, and storage and warehousing facilities related to the transportation of cargo. Assigned by the terminal leasing company.
 5. **Gate Operations Manager:** manage all gate, inter-, and intra-terminal transfer activities. Liaison with customers, port authorities, road container haulers, and truck drivers regarding terminal gate container transactions and transfer activities. Assigned by the terminal leasing company.
 6. **Drayage Company:** legal entities in charge of intermodal transportation, i.e., taking containers in and out of warehouses, rail terminals, ocean ports, and harbors. Assigned by shipping lines or terminal operators.
 7. **Drayage Truck Driver Scheduler:** an individual responsible for scheduling truck drivers for the drayage transaction (either drop or pick up a container). Assigned by the drayage company.
 8. **Drayage Truck Driver:** an individual responsible for dropping off or picking up a container.
 9. **Terminal Trailer Driver:** an individual responsible for moving the container to/from the ship to the storage area inside the terminal.
 10. **Crane Operator Scheduler:** an individual responsible for assigning crane operators to load/unload containers to/from the ship. Depending on the contract that the terminal operator has with the port authority, the crane operator scheduler can be assigned by the terminal operations manager or the port operations manager.
 11. **Crane Operator:** an individual that controls the operation of the STS crane when loading and unloading containers to/from a ship from either the crane's cabin or a remote control room.
 12. **Operations Officer:** a member of a marine operations team involved in the safe transit and handling of vessels into and out of the port. The individual may also be involved in the berthing of vessels, port control and marine services, operation of harbors and marinas, conservancy, and environmental protection.
 13. **Security Officer:** an individual responsible for the protection of people, properties, and information at the port.
 14. **Gate Control Employee:** a member of the port authority organization.
 15. **Harbor Master:** an individual or system establishing a berthing schedule consisting of berthing times and berthing positions of containerships in port container terminals. The berth schedule must be constructed in a way to satisfy requests from carriers on berthing times and minimize handling efforts during ship operation.
 16. **Marine Pilot:** an individual employed by a port or harbor to ensure the safe navigation of ships in their waters. Pilots board ships entering or exiting the port and navigate them safely, avoiding hazards. Pilots may be faced with high-risk cargo, poor maneuverability, and communication difficulties.
 17. **Tugboat Pilot:** the pilot of a small vessel that helps larger crafts steer in tight spaces where their engines cannot safely reach full power.

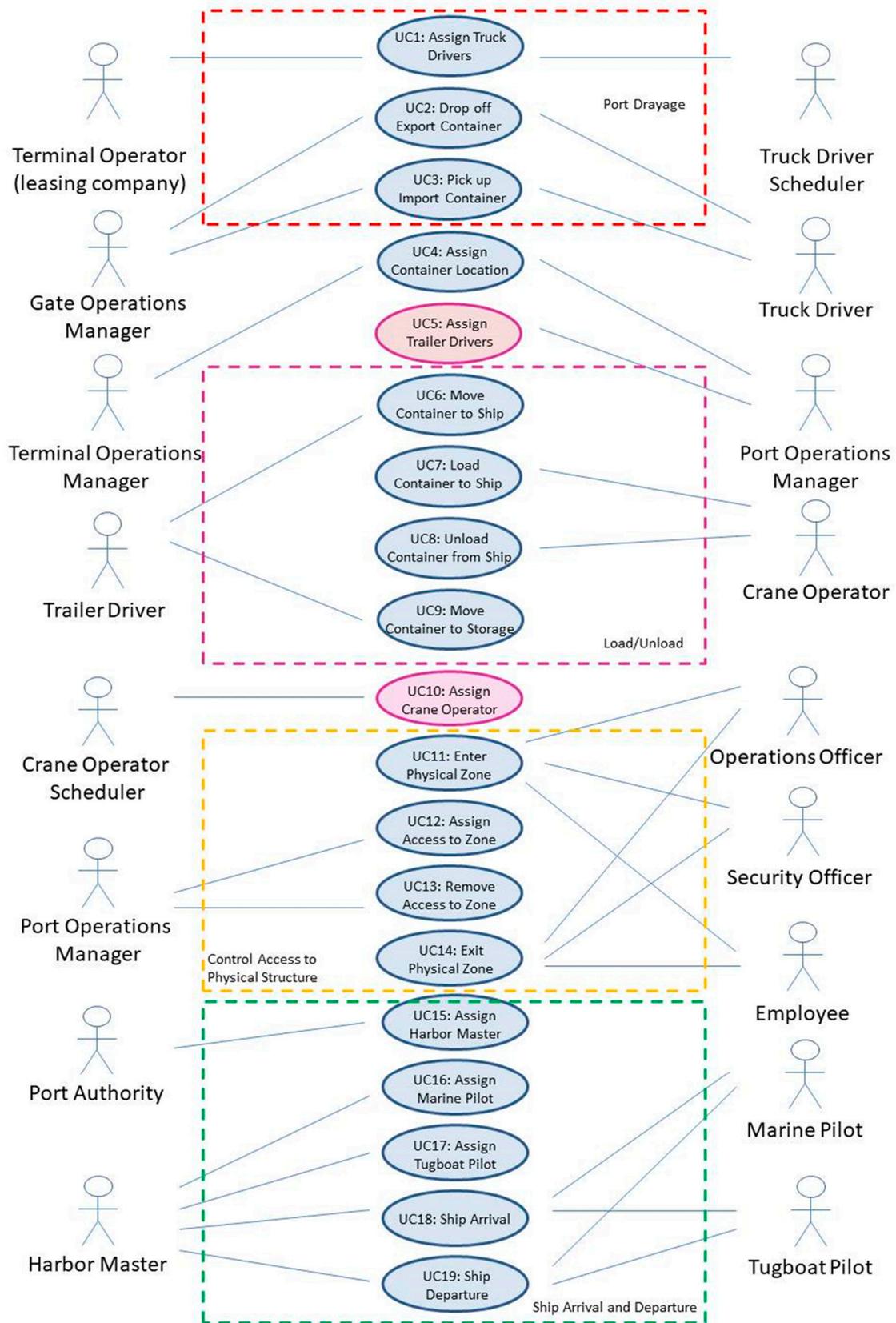


Figure 5. Use cases for a cargo port.

3.4. Use Cases (UCs)

A description of the use cases in the system follows. The relationship of these use cases to the process patterns in the previous section is shown in Figure 5, where patterns are delineated by dashed colored lines.

UC1—Assign Drayage Truck Drivers: Actions taken by the drayage company truck driver scheduler to assign truck drivers to drop off or pick up containers from the cargo port. The drayage company may be assigned by the terminal operator (leasing company) or the shipping line. Actors: terminal operator (leasing company), drayage company truck driver, scheduler.

UC2—Drop off Export Container: Actions taken by the drayage truck driver and the gate operations manager to allow entrance to the port and drop off an export container. Actors: gate operations manager, drayage truck driver.

UC3—Pickup Import Container: Actions taken by the drayage truck driver and the gate operations manager to allow entrance to the port and pick up an import container. Actors: gate operations manager, drayage truck driver.

UC4—Assign Container Location: Actions taken by the terminal operations manager or port operations manager controlling the logistics of the container terminal to assign container locations. Actors: terminal operations manager; port operations manager.

UC5—Assign Trailer Drivers: Actions taken by the port operations manager to assign trailer/shuttle drivers to take the containers from the STS cranes to the drop-off point in the storage yard. Actors: port operations manager.

UC6—Move Container to Ship: Actions taken by the trailer/shuttle driver to take the container from the storage yard to the ship. Actors: trailer driver.

UC7—Load Container to Ship: Actions taken by crane operators to load outbound containers onto ships utilizing STS cranes. Actors: crane operator.

UC8—Unload Container from Ship: Actions taken by crane operators to offload inbound containers from ships utilizing STS cranes. Actors: crane operator.

UC9—Move Container to Storage: Actions taken by the trailer/shuttle driver to take the container from the ship to the storage yard. Actors: trailer driver.

UC10—Assign Crane Operator: Actions taken by the crane operator scheduler to assign a crane operator to a specific STS crane. One crane operator may be assigned more than one STS crane. Actors: crane operator scheduler.

UC11—Enter Physical Zone: Actions taken by an operations officer, a security officer, or an employee to have access to a specific area in the cargo port. Actors: operations officer, security officer, employee.

UC12—Assign Access to Zone: Actions taken by the port operations manager to allow access by an individual to a specific zone. Actors: port operations manager.

UC13—Remove Access to Zone: Actions taken by the port operations manager to remove access for an individual to a specific zone. Actors: port operations manager.

UC14—Exit Physical Zone: Actions taken by an operations officer, a security officer, or an employee to exit a specific area in the cargo port. Actors: operations officer, security officer, employee.

UC15—Assign Harbor Master: Actions taken by port operations to assign a harbor master to schedule vessel arrivals, departures, and berthing assignments. Actors: port authority.

UC16—Assign Marine Pilot: Actions taken by the harbor master to assign a marine pilot to direct the ship to the port area. Actors: harbor master.

UC17—Assign Tugboat Pilot: Actions taken by the harbor master to assign a tugboat pilot in case the ship requires the assistance of a tugboat for performing maneuvers in the port area and completing entrance and mooring operations. Actors: harbor master.

UC18—Ship Arrival: Actions taken by the harbor master, marine pilot, and tugboat pilot to bring the ship to port. Actors: harbor master, marine pilot, tugboat pilot.

UC19—Ship Departure: Actions taken by the harbor master, marine pilot, and tugboat pilot when the ship is departing the port. Actors: harbor master, marine pilot, tugboat pilot.

4. Patterns in a Cargo Port

Now we show the patterns that we used as building blocks for our cargo port reference architecture. Patterns may contain one or more use cases. We present partial and updated descriptions of the patterns using a POSA template [7]; for full descriptions, see the corresponding references.

4.1. Cargo Port Drayage

The cargo port drayage pattern includes several use cases: *UC1: assign truck drivers*; *UC2: drop off export containers*; and *UC3: pickup import containers*. In this subsection, we discuss *UC2: drop off export container* in detail. This pattern is an update of [15].

4.1.1. Intent

Provide all the needed functions for the delivery and pickup of containers at a maritime cargo port.

4.1.2. Context

Port drayage refers to the movement of containers between a port terminal and an inland distribution point or rail terminal. A typical drayage assignment involves either delivering an export container to a marine terminal or picking up an import container that has arrived by ship [46]. According to the Intermodal Association of North America (IANA), there are more than 60 million drayage movements each year in North America.

4.1.3. Problem

Drayage companies take containers in and out of warehouses, rail terminals, ocean ports, and harbors. Drivers arriving at the entrance gate are anticipating one of the following transactions: picking up a container or dropping off a container. There may be several variations in this process, such as dual transactions, trouble window transactions, and equipment issues. It requires extra time to inspect and accept a container, to receive instructions on where to take the container within the terminal, and to leave the load and either start another transaction or leave. Depending on the type of transaction, gate processing time and associated queueing vary.

The solution to this problem is guided by the following forces:

- Flexibility: several internal and external roles may be involved, i.e., truck operators, storage area workers and supervisors, crane operators, etc. Resources and devices used, as well as their corresponding operations, must be flexible to accommodate this variety of roles [47];
- Usability: the software used to identify the individuals entering the port, their trucks, and their container contents should be easy to use by roles that do not have technical backgrounds;
- Alerting: any attempt to deviate from the normal operations of the terminal must produce an alert and should be logged;
- Logging: any activity should be recorded and logged for future auditing. In general, every visit should be logged to keep track of any access to the facility. All containers must be registered and logged;
- Location tracking—we need to be able to find every container. A container in the wrong place can delay operations.

4.1.4. Solution

Every maritime port container terminal should include port security and access control functions. Drivers arriving at a maritime terminal entrance gate intend to either drop off a loaded export or an empty export container and/or pick up a loaded or empty import container. The variety of external users and the fact that the contents of the containers are usually not visible present many threats. Continuous checks are required, not only of the individuals entering the terminal but also of the contents of their trucks. We must

authenticate drivers and their loads before they enter the terminal. We must log every container move. All activities need to be recorded for future auditing in case of a security violation. All truck and container moves need to be recorded using videos to improve security and operations at the port. The containers need to be tagged, tracked, and their locations recorded for identification and to prevent the illegal entry of weapons, chemicals, or biological agents, as well as human trafficking. In general, every visit should be logged. Other transactions may be possible (such as moving trucks without containers or chassis), but these movements of empty containers or bare chassis are the result of loaded movements.

4.1.5. Structure

Figure 6 presents the class diagram of a cargo port drayage operation. *Quay* represents the platform for loading and unloading ships. *GateAccess* represents the entrance to the terminal. Each quay has only one *GateAccess* since each port terminal has only one gate for entrance. *DrayageTransaction* represents a single drayage transaction, where a transaction implies the delivery or pickup of a *container* to/from the quay. *DrayageTransaction* consists of entities *truck*, *truck driver*, and *container*. For each drayage transaction, there is one truck driver in charge of one truck and one to two containers, depending on the size of the containers and whether they drop off and pick up a different one. *GateAccess* must verify that the drayage operation is a legitimate one. *ImportArea* and *ExportArea* represent the surfaces on the terminal assigned to stack the containers. Each container has a unique location in the storage yard, which is represented by *ImportLocation* or *ExportLocation*. These unique locations are kept and updated in the log.

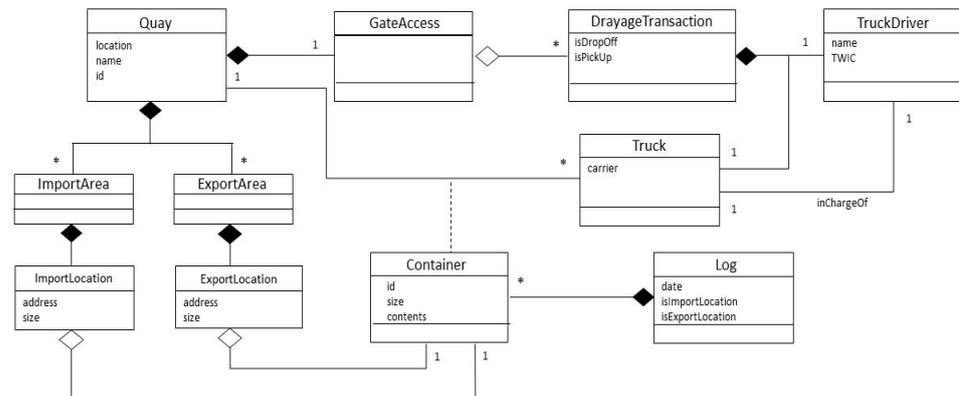


Figure 6. Class Diagram of Cargo Port Drayage.

4.1.6. Dynamics

Figure 7 describes the dynamic aspects of the Secure Cargo Port Drayage pattern using an activity diagram for the following use case.

Use Case: Drop-off Export Container:

Summary: a truck requests entrance to the maritime cargo port. *GateAccess* verifies the identity of the driver, the truck, and its load. *GateAccess* also verifies that the driver’s transaction is a legitimate one. The driver obtains the location and proceeds to drop the container off in the storage yard.

Actors: *TruckDriver*, *GateAccess*.

Precondition: The drayage operation information has been previously entered into the system.

Description:

1. *TruckDriver* arrives at the terminal entrance.
2. *GateAccess* authenticates the driver.
3. *GateAccess* authenticates the truck.
4. *GateAccess* authenticates the container load.

5. TruckDriver obtains the location for the drop and proceeds to the storage yard.
6. TruckDriver drops the container in the appropriate location in the storage yard, exits the terminal, or picks up another container. The log is written, indicating the container ID and its location.

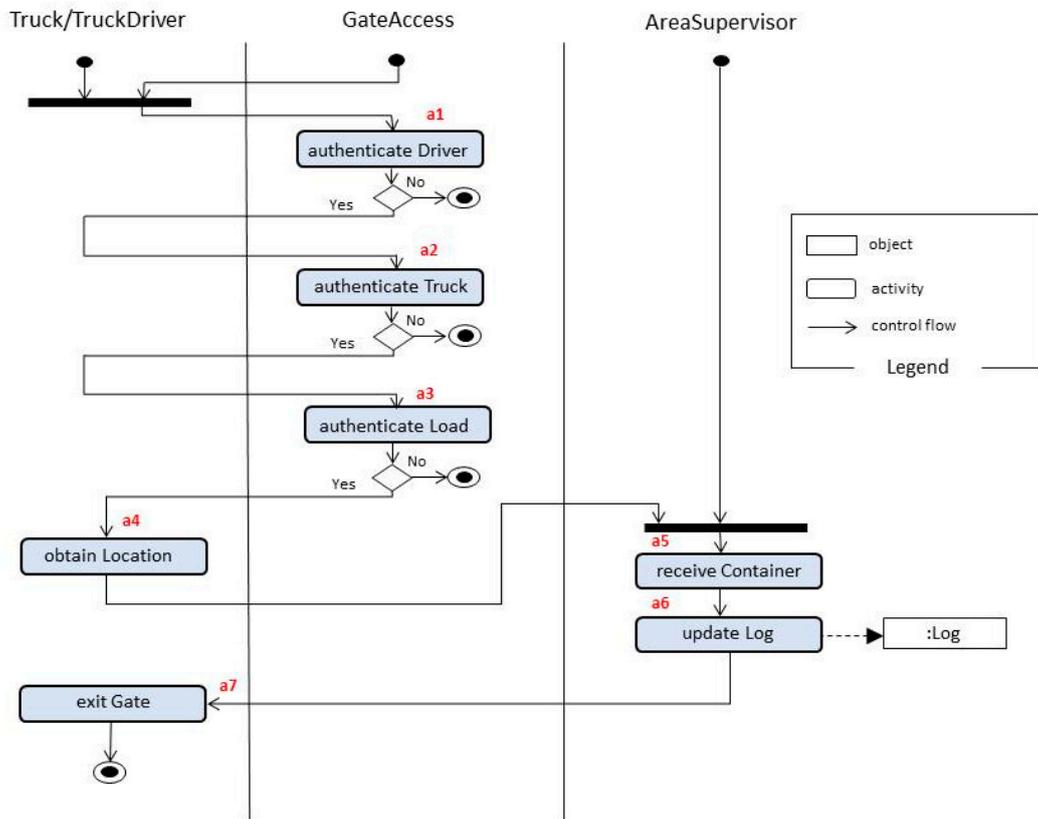


Figure 7. Activity Diagram for use case drop-off export container.

Postcondition: A container has been deposited in the port, and the transaction is recorded.

Other typical use cases include picking up import containers, assigning locations to the containers, and assigning drivers to the trucks. Typical roles include gate attendant, storage yard supervisor, storage yard worker, truck driver, and gate access worker.

4.1.7. Implementation

This pattern can be implemented at all gate entrance operations for all ports. Roles and access rights need to be standard across port terminal locations. A code for each port drayage operation can be implemented.

To prevent the forgery and image manipulation of documents, we can use image processing applications that detect irregularities in font and design of inserted words, spaces between letters, discrepancies in size, crowding, and non-uniformities in the background [48].

Access to the storage yard should be limited only to the area indicated by the authenticating system [49]. The different areas may have physical separations and gates. Any deviation should be alerted, reported, and logged.

4.1.8. Known Uses

The ports of Los Angeles and Long Beach, California, USA, are using a similar process for their port drayage operations. In fact, we used their port drayage process as the basis for this pattern [50,51].

The ports of Miami and Fort Lauderdale (Port Everglades), Florida, USA, also apply a similar process for their port drayage operations.

4.1.9. Consequences

The advantages of this pattern include:

- Flexibility: role-based access control (RBAC) [50] allows us to accommodate several roles that participate in the system; it allows for all types of users. People performing the same tasks are given the same rights;
- Usability: easy to use by individuals that do not have technical backgrounds;
- Alerting: we can use the alarm monitoring, security logger, and auditor patterns to record all activities that are security relevant. Any deviation from normal activities will be logged, and if necessary, an alert will be displayed;
- Logging: the places where the containers have been placed are registered. All transactions are logged and later audited to assure compliance with security regulations;
- Location Tracking: logging will keep track of locating the containers.

The pattern has some liabilities:

- All the mechanisms needed for the process require extra personnel and maintenance, and they have an associated extra cost. Containers may have dangerous loads, and physical measures are needed to detect them, such as X-rays and radiation portal monitors that may be costly.

4.2. Cargo Port Loading and Unloading

The cargo port loading and unloading pattern includes several use cases, including UC6: Move Container to Ship; UC7: Load Container to Ship; UC8: Unload Container from Ship; and UC9: Move Container to Storage. In this subsection, we discuss UC8: Unload Container from Ship and UC9: Move Container to Storage in detail. This description is an update of [16].

4.2.1. Intent

This pattern provides the typical functions of a container terminal facility: loading and unloading of containers to/from a ship.

4.2.2. Context

Port operations and logistics have a significant impact on the economy and international trade. As international trade increases, ports face pressure to improve their infrastructure in order to maintain their operations and respond to market demands. In addition, given the high level of competition, ports must use their resources efficiently and effectively, which has resulted in increased automation. Port operations refer to the activities and processes necessary to manage and control a port. Container terminals are commonly open systems of material flow. In a container terminal, there are three main areas of operation: seaside operations, yardside operations, and landside operations. A port logistic system must keep track of many container positions. A log to keep track of all the container movements must also be kept.

4.2.3. Problem

Traditional ship unloading in ports and terminals varies. There are fully integrated ship-to-stockyard systems, incorporating STS cranes, AGVs, and automated stacking cranes for fully dedicated berths. Alternatively, the system for multi-cargo berths operates cranes, trailers, trucks, and hoppers in the specific stockyards. The main issue is how to do it in an efficient, safe, and environmentally friendly manner.

The solution to this problem is guided by the following forces:

- Flexibility: a variety of users are involved in ports. They are required to operate the facility and have assigned roles. Some external users can also have access for

administrative or management purposes. We need to accommodate this variety of roles;

- Safety: the equipment has physical limits that cannot be exceeded because they would get damaged or endanger the port workers;
- Records: we need to record all activities that may need to be audited later to record any type of violation.

4.2.4. Solution

Ports include loading/unloading facilities such as cranes and storage units. During unloading, operators move containers from ships to storage units, and during loading, operators move containers from storage units to ships.

4.2.5. Structure

Figure 8 shows the class diagram of a port loading system, Port-Load. The system is composed of a set of cranes, several crane operators, and some storage units (warehouses, bins). In a given moment, this system loads or unloads a set of ships, each one carrying a set of containers.

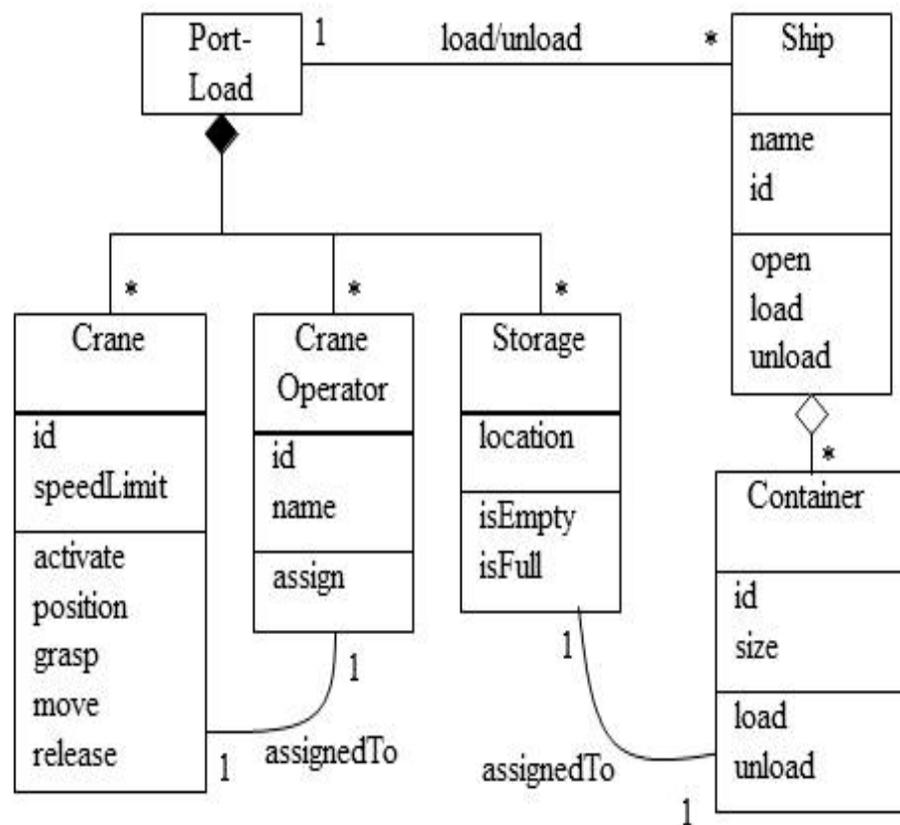


Figure 8. Class diagram of port loading system.

4.2.6. Dynamics

We describe the dynamic aspects of the load/unload container process in a cargo port terminal using an activity diagram for the following use cases: unload container from ship and move container to storage.

Activity: Unload Container from a Ship and Move to Storage

Summary: this activity diagram presents all the steps necessary to unload a container from a ship and move it to an appropriate location in storage. See Figure 9.

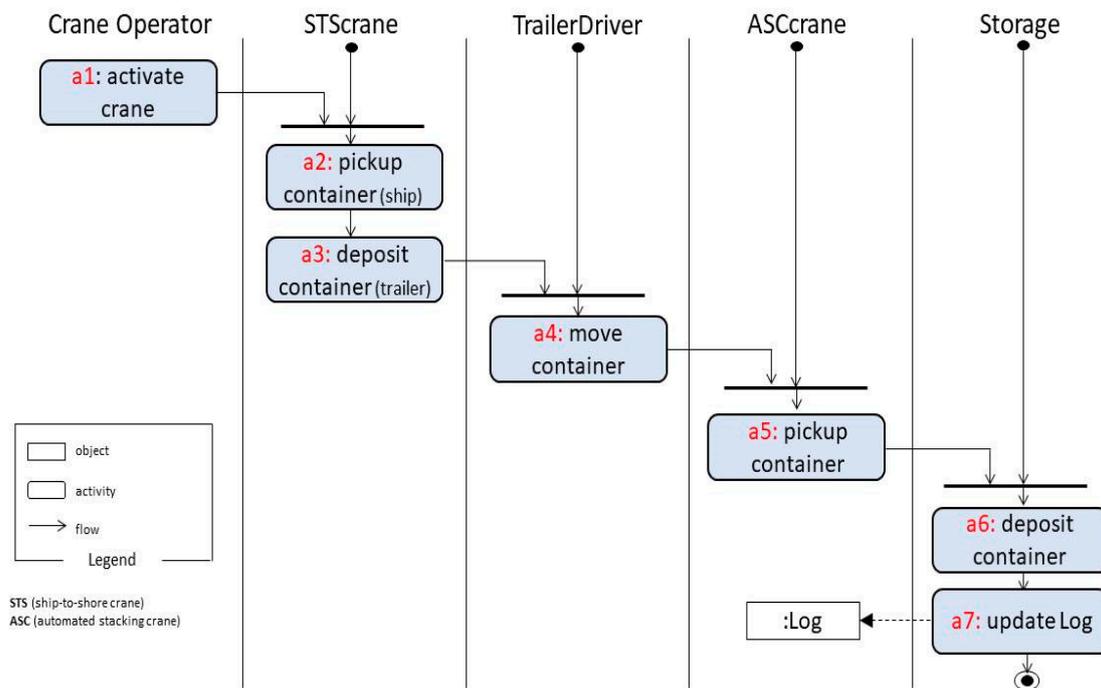


Figure 9. Activity diagram for UCs unload container and move to Storage.

Actors: crane operator, trailer driver.

Precondition: the location of the ship and which crane to operate have been previously entered in the system.

Description:

1. Crane operator activates the crane.
2. Crane operator picks up container from the ship.
3. Crane operator deposits the container in the trailer.
4. Trailer driver receives the container.
5. Trailer driver takes the container to the stacking crane transfer point.
6. ASC picks up the container and finds its location in storage.
7. ASC deposits container in an appropriate place in storage.
8. Transaction is logged.

Postcondition: a container has been unloaded from a ship and deposited in its corresponding place in storage. The log is updated.

4.2.7. Implementation

We can first consider the physical units. We need to give unique identifiers to all the cranes, operators, ships, and storage units. These components are relatively unchanging.

After this, we need to define rights for operators and other roles. Their specific members and rights may change frequently.

While RBAC is not the only option, it is the most used security model in industry because of its general advantages [52].

Ships are the most variable components; they come and go. Their identifiers may be outside our control, but we need to know their IDs to authenticate them when they arrive.

4.2.8. Known Uses

Most ports worldwide use this process for the loading and unloading of containers. In automated container terminals, “trailers” have been replaced by automated guided vehicles (AGVs), and most container terminal cranes are unmanned and operated by crane operators from a control room.

4.2.9. Consequences

The advantages of this pattern include:

- **Flexibility:** RBAC allows for accommodating any kinds of roles appropriate for all kinds of users;
- **Safety:** The equipment's physical limits can be controlled with safety assertions. Other assertions can prevent situations that would be dangerous for humans;
- **Records:** We can use the security logger/auditor pattern to record security-relevant activities.

The pattern also has some liabilities:

- There is the possibility that containers have dangerous loads, and we need physical measures to detect them.

4.3. Container Terminal Physical Structure

4.3.1. Intent

Describe the structure of a container terminal in a cargo port, including its zones and their subdivisions.

4.3.2. Context

Due to the operational characteristics of maritime transportation, port locations are constrained to a limited array of sites defined mostly by geography. Container terminal infrastructure, on the other hand, is determined by the cost and size of the port. Container terminals work day and night, and they need specific areas to load and unload ships. They also need areas for storage of the containers and an area to accommodate transshipment activities.

4.3.3. Problem

To define an infrastructure to keep track of the containers while they are being loaded or unloaded to/from ships and/or transshipped to other places for distribution. We need to identify the different areas and their functions. We also need to have a way to control access.

The solution to this problem is guided by the following forces:

- **Flexibility:** the description of the physical structure should be flexible and scalable. The physical units may change their functions, within restrictions, and we may need more physical units of some type;
- **Scalability:** the number of buildings and their divisions can change up or down depending on the amount of work required;
- **Upgradability:** should allow for changes in the structure to add or remove port functions;
- **Physical Security:** need to provide a good level of security for cargo port functions.

4.3.4. Solution

Once "maritime access", which refers to the physical capacity of the site to accommodate ship operations, and "maritime interface", which indicates the amount of space that is available to support these operations, are satisfied by a proposed port site, the surrounding land is divided for the construction of the port. The port site must have infrastructure such as docks, stacking or storage areas, warehouses, and equipment such as cranes, some by the water to unload the ships and some by the storage areas and warehouses to move and store the containers. Depending on the size of the port and the services provided, we determine the number of employees and the number of buildings needed for offices.

4.3.5. Structure

Figure 10 shows the static structure of a container terminal and the relationships among its components. Class *Port* represents all the terminals or docks in our cargo port. *Terminal*, *TransportMean*, *Area*, and *Crane* classes represent the physical components.

Terminal represents the docks of the cargo port, and TransportMean represents all the transportation means within the port: trucks moving containers in and out of the terminal, railroad trains carrying the containers in and out of the terminal, or possibly another ship. Class Area represents several surfaces in the terminal; StorageArea represents the area to park the containers; BufferArea, a temporary storage area; Crane represents all types of cranes: ship-to-shore cranes that service the ship; YardCrane and StackingCrane employed to stack the containers.

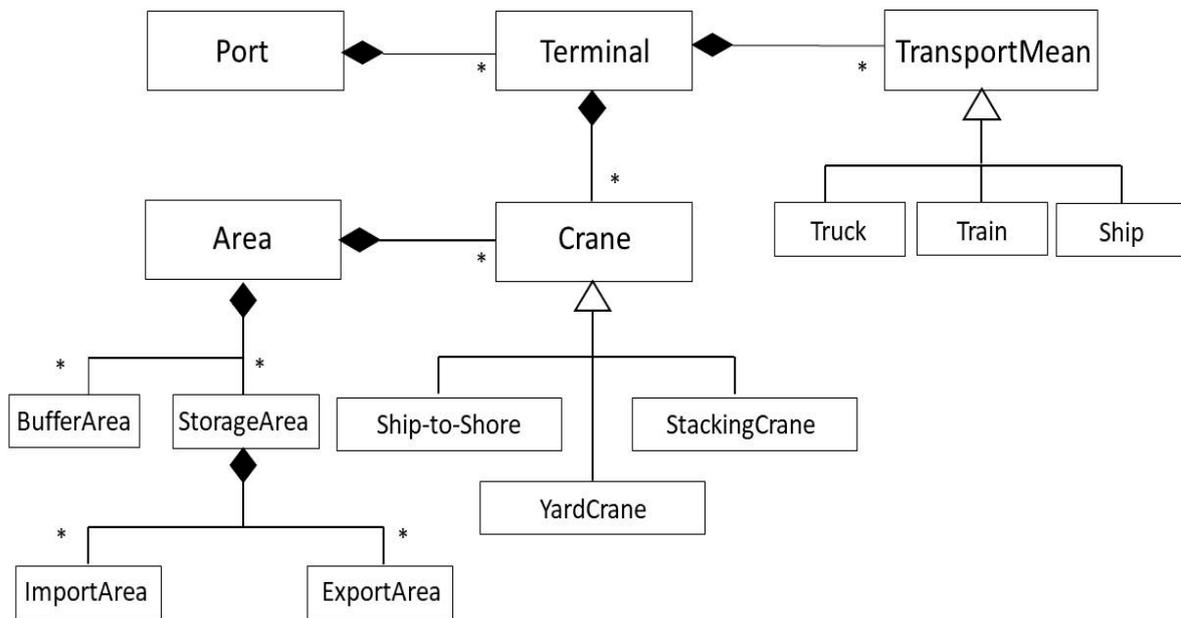


Figure 10. Class diagram of a container terminal’s physical structure.

4.3.6. Implementation

This pattern can be implemented for all existing and new container terminals. There are a variety of possible implementations to take advantage of technological advances.

Remote crane operations are becoming more popular in yard operations and in ship-to-shore cranes. The cranes are being fitted with advanced automation and remote control.

4.3.7. Known Uses

Most ports worldwide have the infrastructure and container handling equipment presented in this pattern. The ports of Los Angeles [50] and Long Beach in California, USA [51], as well as the ports of Miami and Port Everglades in Fort Lauderdale [53], present the described infrastructure.

Port Everglades is the #1 seaport in Florida by revenue as well as one of the top container ports in the state. The port is also South Florida’s main seaport for petroleum products, including gasoline, jet fuel, and alternative fuels. It has 44 available berths with a draft depth of 44 ft. Port Everglades’ existing seven gantry cranes in the Southport area, where most of the containerized cargo operations take place, are 151 feet high and can reach containers stacked six high on deck and sixteen containers across [53]. As part of its expansion program (fall 2020), they have currently purchased an additional six Super Post-Panamax container gantry cranes, the largest of their kind worldwide. On November 17, Broward County’s Port Everglades received three. In comparison with the existing ones, the three 175-foot-high gantry cranes, valued at \$13.8 million each, have the ability to handle containers stacked nine high from a ship’s deck and reach 22 containers across. This will enable the port to handle larger ships and gain economies of scale [53].

4.3.8. Consequences

The advantages of this pattern include:

- **Flexibility:** this pattern is flexible enough that we can add other types of cranes and other types of areas to store the containers, i.e., areas to handle refrigerated containers, areas to store liquids, etc;
- **Scalability:** this pattern provides for any amount of container handling equipment, vehicles used to move the containers around the terminal, and storage areas, depending on the amount of work required;
- **Upgradability:** this pattern allows for any changes in functionality, such as adding or removing port functions;
- **Physical Security:** access of personnel to specific areas can be controlled using authorization models [17,52].

The pattern has some liabilities:

- This pattern does not detect dangerous loads, and physical measures are needed to detect them, such as X-rays and radiation portal monitors that may be costly.

4.4. Ship Arrivals and Departures

Ship arrival and departure patterns include several use cases, including *UC15: Assign Harbor Master*; *UC16: Assign Marine Pilot*; *UC17: Assign Tugboat Pilot*; *UC18: Ship Arrival*; and *UC19: Ship Departure*. In this subsection, we discuss *UC18: Ship Arrival in detail*. The activities pertaining to the ship and the ship crew prior to arrival are not covered in this pattern; only the actions required when arriving at port are.

4.4.1. Intent

Arriving and departing ships must be taken to/from their predefined port locations (berths) in a safe manner.

4.4.2. Context

Cargo ports have several locations to hold ships for loading/unloading (berths). These locations must be assigned in advance but can be changed depending on the conditions of the port or berth availability.

4.4.3. Problem

Arrival at a port and departure from a port are two separate but very important aspects of a ship's voyage. Both of these procedures are critical and involve a number of different complexities in them. Arriving at some ports may require going through narrow or busy seaways where navigation may be difficult.

The solution to this problem is guided by the following forces:

- **Availability:** berth facilities must be assigned in advance and in such a way as to minimize ship stays in the port;
- **Scalability:** the number of provisioned, designated locations to dock in a port should vary depending on the traffic of vessels to the port. New berths can be provisioned for loading/unloading as needed;
- **Berth Scheduling:** berth scheduling should be based on the priority of resource allocation and a comparison of the vessel and berth data. How to assign its berth time and position is of great importance to improving the level of port services and reducing the port's cost of production;
- **Safety and Port Integrity:** arrival and departure procedures should ensure the safety of the ship's arrival and departure, both for the people on board as well as those at the seaport. Most of the collisions and groundings of ships are reported during the maneuvering of the vessels in port, and hence the maneuvering operation at port arrival or departure is considered the most crucial time a ship faces in her voyage. The

availability of marine pilots and tugboats should ensure the avoidance of physical damage [54].

4.4.4. Solution

Before entering the port area, the ship is required to contact the port authority regarding its estimated time of arrival (ETA), give particulars of the ship and her cargo, check for berth availability, and request a marine pilot to direct it to the port area. The ship may also require the assistance of a tugboat for maneuvering inside the port area and completing entrance and mooring operations. Similar operations are carried out when the ship leaves the port area [53].

All berthing, unberthing, and maneuvering of vessels within the port limits must be done under the supervision of a marine pilot. It is the duty of the Harbor Master to inform the pilot of any special conditions, difficulties, or weather conditions.

4.4.5. Structure

In Figure 11, the class *Port* contains a set of berths—the multiple areas for docking in a port. For each object of class *Berth*, there are multiple stays of a ship in a berth. Each instance of *ShipStay* can be an *Arrival* or a *Departure*, representing either a ship’s arrival or departure. Multiple objects of the class *Tugboat* can be associated with multiple objects of the class *Ship*. The associations between ships and tugboats are temporary and described by instances of the mediator pattern [8]. Each object of *Berth* is assigned a schedule, *BerthSchedule*.

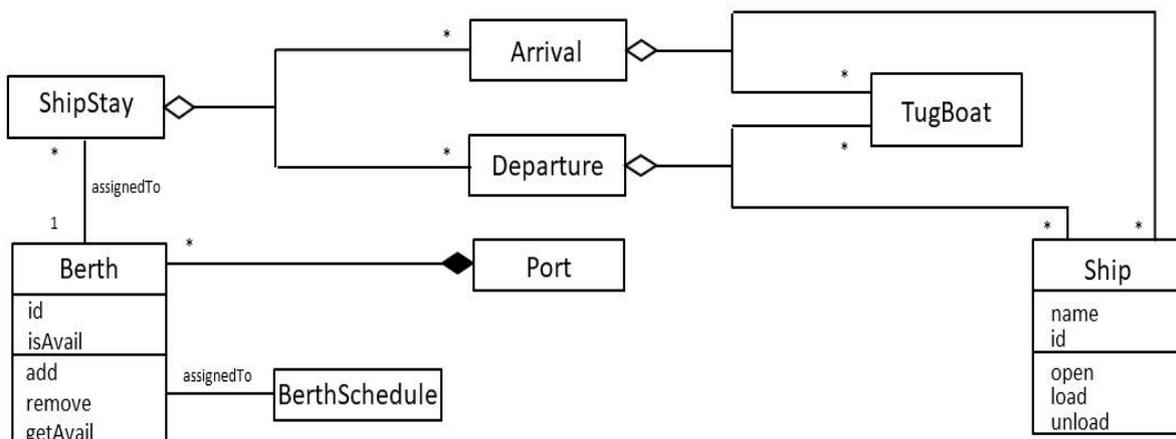


Figure 11. Class diagram of ship arrival and departure operations.

4.4.6. Dynamics

UC “ship arrival” is executed as a series of steps described below. Figure 12 shows the sequence diagram for the arrival of the ship at the port.

UC: Ship Arrival

Summary: a ship arrives at a port.

Actors: ship master, harbor master, marine pilot, and tugboat pilot.

Preconditions: ships’ times of arrival are known. Assignments of berths to arriving ships, of pilots to ships, and/or of tugboats must be predefined.

Description:

1. Ship master contacts harbor master and requests access to the port.
2. Harbor master checks the berthing schedule.
3. Harbor master grants access for ships to berth.
4. Ship master requests a marine pilot to bring the ship to port.
5. Harbor master grants request to send marine pilot.
6. Ship master requests a tugboat for maneuvering vessel in port.
7. Harbor master grants request to send tugboat pilot and tugboat to assist ship.

- 8. Ship master informs ship docked in berth.
 Postcondition: A ship successfully docks at the port.

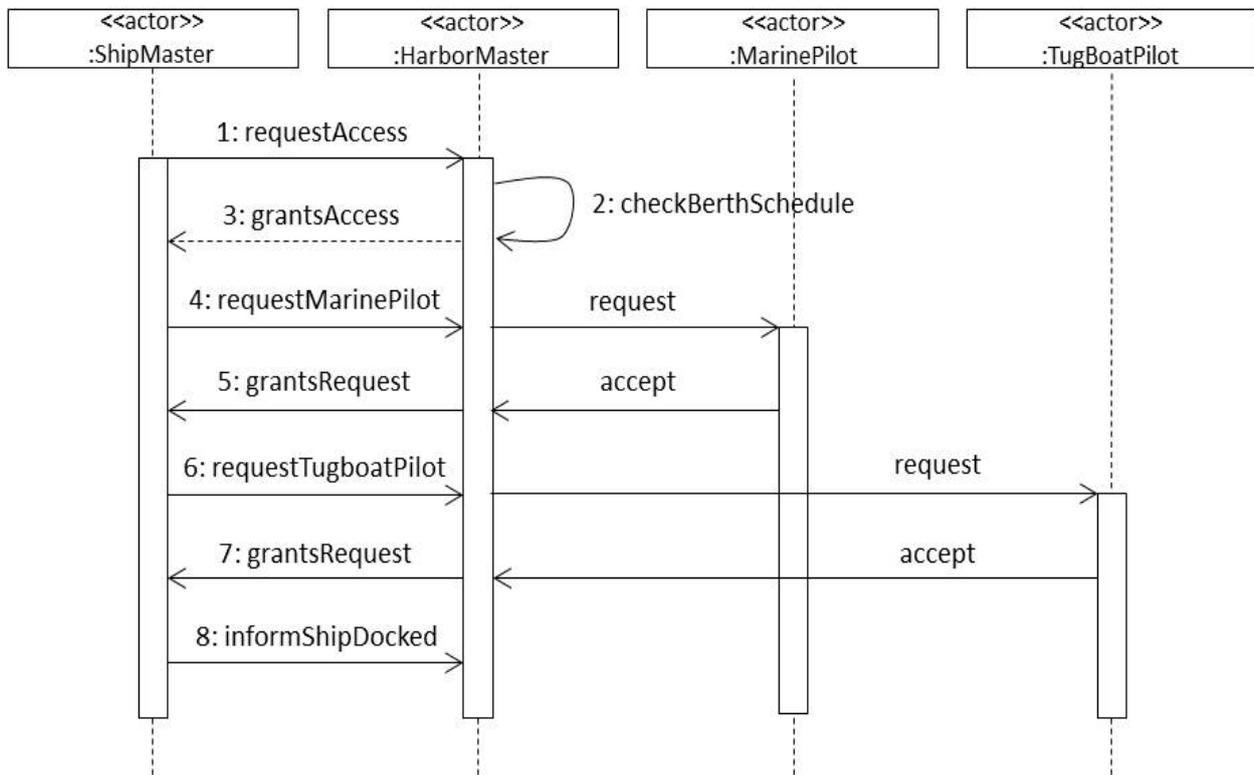


Figure 12. Sequence diagram for UC ship arrival.

4.4.7. Implementation

Starting in 2001, ships and ports started to install automatic identification system (AIS) equipment. AIS transmitters on board the ships automatically report the arrival and departure times to the port authorities. This technology is primarily used to avoid collisions and increase port security. At the present time, AIS devices have not been installed in all ships and ports, but most major ports and the largest ships are included. AIS provides a means for ships to electronically send data, including vessel identification, position, speed, and course. AIS uses global positioning systems (GPS) in conjunction with shipboard sensors and digital VHF radio communication equipment to automatically exchange navigation information electronically and to monitor vessel location and movement primarily for traffic management, collision avoidance, and other safety applications [55–57].

4.4.8. Known Uses

The ports of Los Angeles and Long Beach in the USA use this process for ship arrivals and departures. Most ports worldwide, such as the port of Shanghai and the port of Singapore, also use this process for ship arrivals and departures.

Additionally, Port Everglades and the Port of Miami use AIS for vessel tracking [53]. AIS tracking websites (e.g., Marine Traffic) operate over 2000 AIS stations in over 165 countries across the globe.

4.4.9. Consequences

The advantages of this pattern include:

- Availability: port facilities work 24 h a day. By previously scheduling with the harbor master, vessels can arrive at a port at any time. Ports are always open and available to vessels when needed;

- Scalability: ports and harbor masters provision berths for vessels to dock depending on traffic and the number of vessels entering the port;
- Berth scheduling: proper berth assignment and scheduling ensure that the vessel characteristics are compatible with the characteristics of the berth to which the vessel is scheduled. It improves the level of port services and reduces the port’s cost of production;
- Safety and port Integrity: arrival and departure procedures ensure that the essential safety steps are followed both for the people on board as well as those at the seaport. Most of the collisions and consequent grounding of ships are avoided by utilizing experienced marine pilots and tugboat pilots to aid in the mooring of a vessel.

The pattern has some liabilities:

- Having experienced marine pilots and tugboats reserved and waiting to help the ships dock increases the cost of running the port.

5. A Reference Architecture for a Cargo Port

Now that we have seen the different conceptual units that together make up a cargo port system, we can show its corresponding reference architecture. See Figure 13. This RA combines the patterns shown earlier and includes the process patterns, stakeholders, and use cases described above. In this RA, we included the patterns for: loading and unloading of containers to/from a ship at a cargo port loading facility [16]; a pattern for the delivery of cargo port containers (drayage) [15]; a pattern for ship arrivals and departures; and a pattern for a container terminal physical structure [58]. Patterns may exhibit overlapping classes. As indicated earlier, this is not a complete RA for cargo ports.

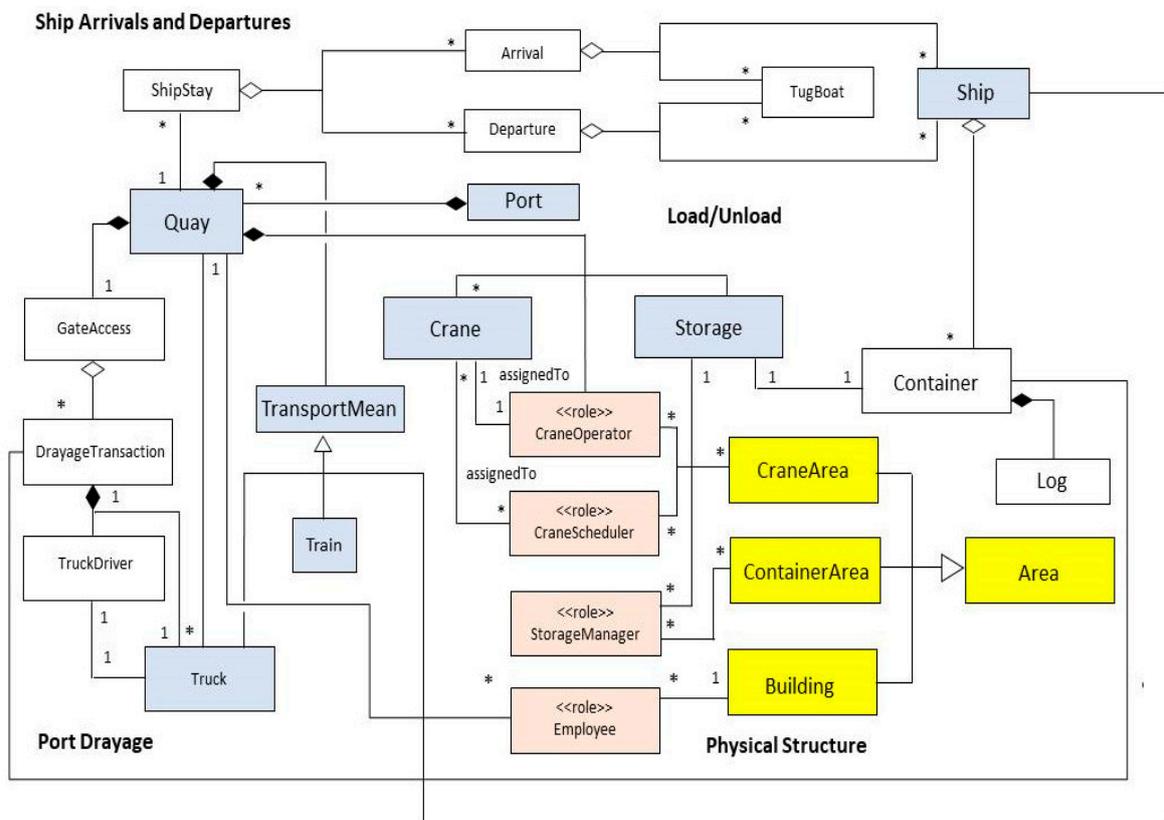


Figure 13. Partial reference architecture for a cargo port.

A brief explanation of the architecture follows: a typical cargo port, *Port*, contains many container terminals, *Quay*. Each container terminal should include port security and access control functions. Drivers arriving at the entrance, *GateAccess*, intend to either pick

up or drop off a container, destined for import or export, respectively. The movement of containers between a port terminal and an inland distribution point or rail terminal is referred to as port drayage. Each single port drayage transaction is an instance of *DrayageTransaction*. For each drayage transaction, there is one *TruckDriver* in charge of a *Truck* and a *Container*. Ports include loading and unloading facilities such as *Cranes* and *Storage* units. During unloading, *CraneOperators* move containers from *Ships* to storage units; during loading, crane operators move containers to ships. *CraneSchedulers* are in charge of assigning crane operators to cranes. *StorageManager* is in charge of assigning storage locations and warehousing facilities. *Employees* are members of the Port Authority organization. Arrival at a port and departure from a port are two separate but very important aspects of a ship's voyage. Both of these procedures are critical and involve a number of different complexities. *ShipStay* represents each stay of a *Ship* in a berth, either an *Arrival* or a *Departure*. The ship may also require the assistance of a *Tugboat* for maneuvering inside the port area and completing entrance and mooring operations. Similar operations are carried out when the ship leaves the port area. Multiple tugboats can be associated with multiple ships. The associations between ships and tugboats are temporary. *TransportMean* represents all the transportation means: trucks moving containers in and out of the terminal, railroad trains carrying the containers in and out of the terminal, or possibly other ships. We do not claim completeness in this RA, but it can be used as a framework to guide the design of the cargo port system.

6. Validation of the RA

RA and patterns are abstract models, not software, and such models cannot be evaluated in terms of performance, reliability, or security using traditional experimentation or testing methods [20,26]; it does not make sense to experimentally validate an instance of a pattern or RA because evaluating its properties verifies them only for that instance, not for all the instances of the pattern or RA. On the other hand, other criteria can be used to validate RA and patterns. We can validate abstract models' completeness by comparing an RA model to models of commercial cargo port systems, explicitly comparing the fundamental components of our cargo port RA with the elements of these systems. If the RA includes all the fundamental features and functions offered by existing cargo port architectures, it means the abstractions faithfully represent their key aspects. Our patterns include "Known Uses" subsections, which indicate examples of commercial cargo port models where we have verified, pattern by pattern, that all the components of our RA are present in these ports. We studied descriptions of the ports of Shanghai, China; Singapore; Rotterdam, Netherlands; Los Angeles, Long Beach, Miami, and Everglades, USA; all these ports represent modern ports, including automation and other advances. Following a standard way to find patterns [7,8], the first four ports were used to build the patterns, and the other three were selected to confirm that real ports used these patterns. None of these descriptions used patterns or UML, so we needed to abstract their components and functions.

In general, specific patterns are normally evaluated by submitting them to a pattern conference. In these conferences, a pattern paper is developed with the help of a shepherd and then discussed in a workshop of about 8–12 subject matter experts. The pattern is exposed for criticism, revised, and then published. In terms of precision, we created our RA through UML modeling and a well-defined pattern template for describing the pattern solutions. This method is widely used in the software engineering discipline and is regarded as more precise as compared to other representations such as textual or block diagram-based representations.

In addition, we have presented our work in several architecture-oriented meetings, such as Pattern Languages of Programming (PLoP) conferences, open Web application security project (OWASP) meetings, and in the pattern language community, where we received significant feedback on the usefulness and benefits of this work. Ultimately, the final validation of this work will come from practitioners using it to build concrete architectures.

Another type of validation refers to the usefulness of the RA [23,26,59], shown in Section 7.

7. Value and Use of the Reference Architecture in Cargo Port Design and Operation

The design of a cargo port is a major endeavor and is performed in three basic stages: A conceptual design is done first for deciding aspects such as location, capacity, personnel, cost, regulations, etc. [60]. This is followed by a detailed design indicating the number and placement of buildings, number of cranes, level of automation, functional distribution of zones, interactions with external organizations, compliance with regulations, security requirements, etc. [61,62]. Finally, the software design of the complete facility is done, for which our reference architecture would be used. This stage involves experts in ports, CPSs, and software (architects and developers). The increasing automation of ports requires an increasing amount of software to coordinate and monitor operations as well as describe the functions of the components [63].

As indicated earlier, Martínez-Fernández et al. [59] addressed the benefits and drawbacks of using RAs in industry practice from the perspective of different stakeholders involved in their design and usage. They estimated the value and risks of building and using an RA for a particular domain. Their study revealed that not all organizations realize the theoretical benefits of RAs; most only realize a subset of these benefits. They revealed that one main drawback was the additional learning curve experienced by application developers who use the RA, causing higher time to market and high complexity when they do not aim to facilitate the development of applications. They concluded that different stakeholders present different concerns. For software architects, standardization and reliability were the key benefits of using RAs, whereas for application developers, it was the use of the latest technologies. Buccaioni et al. [23] give a list of the advantages of having a reference architecture for enterprises.

Some of the practical uses of a model like ours include:

1. RAs aggregate knowledge and design expertise in a specific domain [23]. This knowledge can be reused to build new ports and improve existing ones.
2. Holistic and unified models such as RAs are useful for gaining understanding of a complex system and its limitations. Cargo ports are very complex systems, and having an abstract architecture free of implementation details is very important to understanding the system.
3. An RA serves as a way to decompose a complex system into simpler and more coherent parts that can be assigned to different types of specialists.
4. Automation is now a basic objective to increase efficiency and reduce costs, and an RA can provide a perspective on how the automation of parts of the system contributes to the efficiency of port operation.
5. UML models used to represent the RA provide more precision as compared to other descriptions such as block diagrams.
6. We can contribute to the standardization of concrete architectures by using the RA as a template. Standardization can accelerate automation and reduce costs [63].
7. An RA serves as a way to unify terminology and make it easier to train the developers, system architects, and administrators so they become familiar with the platforms and processes.
8. Abstract representational models of the cargo port components in the RA help address heterogeneity; this is accomplished by focusing on the general categories of the system and grouping them on the basis of similarities in structure and functionality.
9. We can identify common functionalities and configurations to encourage reuse in new projects.
10. RAs support interoperability among different RA applications and their components by establishing common mechanisms for information exchange.
11. RAs can serve as a guideline to indicate where to add defenses for the expected threats and for adding security monitoring in the physical port.

12. Security, compliance, privacy, safety, reliability, and governance can all be improved by the use of these models. For example, the analysis of threats in complex systems can be done systematically with the help of RAs [17].
13. RAs can guarantee the compliance of real architectures with RAs and regulatory requirements [3,23]. Once a new architecture is built, we can test if it follows the structure and policies of its corresponding RA.

8. Related Work

We enumerate now work that helps provide a perspective on our work.

8.1. Papers That Use Patterns and/or UML Models to Describe CPS Architectures

Maidl et al. [64] describe a UML RA for CPSs oriented to analyze security. It describes typical aspects of CPSs such as sensors, interfaces, and networks, from which a deployment diagram for a control system is derived.

Gottschalk et al. [4], using UML models and MDE, defined an RA for smart grids. Another paper in that group relates CPSs to systems of systems [25].

Hehenberger et al. [65] is a survey paper that provides an overview of different types of CPSs and the transition process from mechatronics to CPS. It indicates that methodologies for CPS design should be part of a multi-disciplinary development process within which designers should focus not only on the separate physical and computational components but also on their integration and interaction.

Tan et al. [66] use a service-oriented architecture (SOA) style to design CPSs, including patterns, but their approach is oriented toward the dynamic composition of services [67]. They also propose a SOA architecture described using a block diagram.

La et al. [68] present a 3-tier Software Oriented Architecture (SOA) and define in detail the components of this architecture. Their tiers include the environmental tier for the target physical environment, the control tier for making decisions about networked physical devices, and the service tier for managing reusable services. They use UML sequence diagrams to describe the system components.

RAs using patterns have been very effective in describing other complex non-CPS and IoT systems. A pattern-based RA for a cloud system is presented in [69,70], and a network function virtualization (NFV) RA is presented in [71]. In Syed et al. [47], the authors defined an RA for a cloud ecosystem that described at an abstract level the main features of the cloud system and showed the relationships between containers, the cloud, and IoT ecosystems.

Seiger et al. [14] consider models for CPS processes, described using UML class diagrams; they could be used in the structure of a CPS RA.

None of these architectures considers cargo ports, but [64] has aspects that would apply to the implementation aspects of ports. Seiger's models [14] could be useful to extend the RA.

8.2. Architectures of Transportation Systems

A work discusses the construction of the infrastructure as a collaborative network of stakeholders offering business services based on a single contract [72]. Other authors discuss a service-oriented approach to modeling Port Community Systems (PCS) as an electronic platform linking the multiple systems operated by private and public organizations (enterprises and government institutions) in a complex service system network required to coordinate flows of merchandise, property rights, payments, and information in the global supply chain. Their approach is to provide tools to describe the service-based relationships of the actors, and to understand how each contributes to improving the efficiency of the network [73–75]. The last two studies use a collaborative multi-agent framework. Thurston et al. [67] present a port layout to reduce the time spent by ships in a port. This last work can be considered a complement to the port structure pattern shown in this paper, but the others have very different objectives.

Cargo ports are part of maritime transportation systems, and these papers could be useful to extend our models, although none of them provides architectural models.

8.3. Architectures Proposed as Part of Descriptions of CPSs or Development Methodologies

The literature mentions the possibility of using domain models for CPSs, but no details are given. Refs. [76–81] all give basic block diagram architectures with short descriptions of the units. Spichkova in [82] introduces a development methodology for CPSs focusing on the abstraction levels of the system model, based on the idea of refinement-based development of complex, interactive systems. Krotofil and Gollmann describe the layered structure of CPSs [83].

8.4. Architectures for Other Types of CPSs

Microsoft defined an RA for smart grids [84], and a more up-to-date architecture for smart grids is given in [4], mentioned above. Verissimo et al. [73] describe a security architecture for smart grids emphasizing reliability. Autosar and Industry 4.0 are important architectures for vehicles and manufacturing, respectively. Several RAs for IoT are given in [13], while Guth et al. used an RA to compare IoT platforms [85].

All these architectures are too general (CPSs or IoT in general) or apply to other CPSs, not cargo ports.

8.5. Papers about Quality Aspects of CPSs

A very important quality aspect is security. Our complementary paper [17] shows the design of a security reference architecture (SRA). We discussed a variety of threats, including:
 Drop container, which could destroy the container, kill humans, and damage the ship.
 Misplace container, which would make it very difficult to find it later.
 Accelerate the crane beyond its limit, which could destroy it.
 Denial of service [17,86], where the system is flooded with messages that do not allow it to perform useful work.

Other threats, indicated by one of the referees, complement our threats and include:
 False data injection [87,88], can affect sensor readings and provide a crane operator with incorrect crane positions that could cause a crane to place a container in the wrong place or cause a crane crash.

Preventing actuation attacks [89] prevents the exchange of information between a controller and its actuators.

More threats are discussed in [90].

Martínez-Fernández et al. [59] addressed the benefits and drawbacks of using RAs in industry practice from the perspective of different stakeholders involved in their design and usage. We used their work in Section 7.

8.6. Papers about Development Methodologies for CPSs

Reference architectures and patterns are artifacts used in model-driven development methodologies. Some authors have proposed specific methodologies for CPSs: Hehenberger et al. [65] use SysML (a variant of UML), J.C. Jensen et al. [91] propose a model-based approach, Spichkova et al. [82] propose a multilevel semiformal model, Jacky uses formal models [92], and Karsai et al. propose a layered architecture [93].

In summary, our patterns and RA, as well as the methodology we used to build them, are original and useful for architects and developers building software for cargo ports. The RA captures semantic aspects unique to this type of CPS and has been the basis of a security RA [17].

9. Conclusions and Future Work

We have applied architectural modeling using patterns and reference architectures to build the reference architecture of a specific cyber-physical system, a transportation system, namely, a cargo port. We have verified that no other reference architecture exists for cargo

ports. Existing RAs model the lower levels of a CPS but do not capture the semantics of specific port operations, a basic feature of our models. The method presented in this paper of using patterns to build the reference architecture provides not only a holistic and unified view of the system to users, developers, and researchers, but it is also fundamental to handling the cross-domain complexity of this type of system. A reference architecture is an ongoing process. The model presented here is just a step toward achieving a precise architectural representation; other use cases and components need to be defined as patterns to complete the architecture. Building a complete RA is a very large enterprise, not feasible for a two-person team; therefore, we just intended to define the methodology to build such architectures and build part of it to illustrate this methodology. The complete architecture has many similar components that just differ from others in small details, so building a complete architecture, while important for an enterprise, is not particularly interesting from a research point of view. In parallel work, we have built a security reference architecture for cargo ports [17]. Our architecture considers only the application level, but it can be extended to become a multilevel architecture such as Industry 4.0 or SGAM. RAs can be used as part of a model-driven methodology to build systems; we have proposed such a methodology for distributed systems [94], where our patterns and RAs could be used to build CPSs. We have used UML for our models, a semi-formal approach, but ontologies can be derived from patterns and RAs to have formal descriptions of this RA, which may be in our future work.

Author Contributions: Conceptualization: V.M.R. and E.B.F.; Writing of original version: V.M.R.; Supervision and writing of revision: E.B.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Received from the other author.

Data Availability Statement: No data used.

Acknowledgments: We thank Evangelos I. Kaisar for his valuable comments. The referees provided valuable suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lee, E.A. Cyber Physical Systems: Design Challenges. In Proceedings of the 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing, Orlando, FL, USA, 5–7 May 2008; pp. 363–369.
2. Denker, G.; Dutt, N.; Mehrotra, S.; Stehr, M.-O.; Talcott, C.; Venkatasubramanian, N. Resilient dependable cyber-physical systems: A middleware perspective. *J. Internet Serv. Appl.* **2012**, *3*, 41–49. [\[CrossRef\]](#)
3. Yimam, D.; Fernandez, E.B. Building Compliance and Security Reference Architectures for cloud systems. In Proceedings of the IEEE International Conference on Cloud Engineering (IC2E) 2016, Berlin, Germany, 4–8 April 2016.
4. Gottschalk, M.; Delfs, C.; Model, T.S.G.A. *The Use Case and Smart Grid Architecture Model Approach*; Springer: Berlin, Germany, 2017; pp. 41–61.
5. Avgeriou, P. Describing, Instantiating and Evaluating a Reference Architecture: A Case Study. *Enterp. Archit. J.* **2003**, *342*, 1–24.
6. Taylor, R.N.; Medvidovic, N.; Dashofy, E.M. *Software Architecture: Foundations, Theory, and Practice*; Wiley: London, UK, 2009; ISBN 0470167742, 9780470167748.
7. Buschmann, F.; Meunier, R.; Rohnert, H.; Sommerlad, P.; Stal, M.; Architecture, P.-O.S. *A System of Patterns*; John Wiley Sons Ltd.: New York, NY, USA, 1996; Volume 1.
8. Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software*; Addison-Wesley: Boston, CM, USA, 1994.
9. Romero, V.M.; Fernandez, E.B. Towards a Security Reference Architecture for Cyber Physical Systems. In Proceedings of the Fifteenth Latin American and Caribbean Conference for Engineering and Technology, Boca Raton, FL, USA, 19–21 July 2017.
10. Warmer, J.; Kleppe, A. *The Object Constraint Language: Getting Your Models Ready for MDA*, 2nd ed.; Addison-Wesley Longman: Boston, MA, USA, 2013; ISBN 0321179366.
11. Garavel, H.; Graf, S. Formal methods for safe and secure computer systems. In *Technical Report. BSI Study 875*; Federal Office for Information Security: Bonn, Germany, 2013.

12. Pereira-Vale, A.; Fernandez, E.B. An Ontology for Security Patterns. In Proceedings of the 38th International Conference of the Chilean Computer Science Society (SCCC 2019), Concepción, Chile, 4–8 November 2019.
13. Weyrich, M.; Ebert, C. Reference architectures for the internet of things. *IEEE Softw.* **2015**, *33*, 112–116. [[CrossRef](#)]
14. Seiger, R. Modelling complex and flexible processes for smart cyber-physical environments. *J. Comput. Sci.* **2015**, *10*, 137–148. [[CrossRef](#)]
15. Romero, V.M.; Fernandez, E.B. A Pattern for Secure Cargo Port Drayage. In Proceedings of the 7th Asian Conference on Pattern Languages of Programs, Asian PLOP'18, Tokyo, Japan, 1–2 March 2018; p. 9.
16. Fernandez, E.B.; Monge, R.; Carvajal, R. A pattern for a secure and safe port loading facility. In Proceedings of the 10th Latin American Conference on Pattern Languages of Programs—SugarLoafPLOP, Rio de Janeiro, Brazil, 20–23 November 2014.
17. Fernandez, E.B.; Romero, V. A security reference architecture for cargo ports. *Internet Things Cyber-Phys. Syst.* **2022**, *2*, 120–137. [[CrossRef](#)]
18. Buschmann, F.; Henney, K.; Schmidt, D.C. *Architecture, Pattern-Oriented Software Architecture: On Patterns and Pattern Languages*; John Wiley Sons Ltd.: Hoboken, NJ, USA, 2007; Volume 5.
19. Wieringa, R. Design Science as nested problem solving. In Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology, New York, NY, USA, 7–8 May 2009.
20. Fernandez, E.B. *Security Patterns in Practice: Building Secure Architectures Using Software Patterns*; Wiley Series on Software Design Patterns: Chichester, UK, 2013.
21. Fernandez, E.B.; Pelaez, J.; Larrondo-Petrie, M. Attack Patterns: A New Forensic and Design Tool. In *Advances in Digital Forensics III*; Springer: New York, NY, USA, 2007; pp. 345–357.
22. Fernandez, E.B.; Yoshioka, N.; Washizaki, H. Modeling Misuse Patterns. In Proceedings of the 2009 International Conference on Availability, Reliability, and Security (ARES 2009), Fukuoka, Japan, 16–19 March 2009; pp. 566–571.
23. Bucaioni, A. Reference architectures modelling and compliance checking. *Softw. Syst. Model.* **2022**, 1–27. [[CrossRef](#)]
24. Stricker, V.; Lauenroth, K.; Corte, P.; Gittler, F.; Panfilis, S.D.; Pohl, K. Creating a reference architecture for service-based systems—A pattern-based approach. In *Towards the Future Internet—Emerging Trends from European Research*; IOS Press: Amsterdam, The Netherlands, 2010; pp. 149–160. [[CrossRef](#)]
25. Uslar, M.; Rohjans, S.; Neureiter, C.; Andrén, F.P.; Velasquez, J.; Steinbrink, C.; Efthymiou, V.; Migliavacca, G.; Horsmanheimo, S.; Brunner, H.; et al. Applying the Smart Grid Architecture Model for Designing and Validating System-of-Systems in the Power and Energy Domain: A European Perspective. *Energies* **2019**, *12*, 258. [[CrossRef](#)]
26. Fernandez, E.B.; Monge, R.; Hashizume, K. Building a security reference architecture for cloud systems. *Requir. Eng.* **2016**, *21*, 225–249. [[CrossRef](#)]
27. Romero, V.M.; Fernandez, E.B. Building a Reference Architecture for Cargo Ports using Patterns. In Proceedings of the Sixteenth Latin American and Caribbean Conference for Engineering and Technology, Lima, Peru, 18–20 July 2018.
28. US Dept. of Transportation, Research and Innovative Technology Administration. In *Freight Transportation: Global Highlights*; Bureau of Transportation Statistics: Washington, DC, USA, 2010.
29. Available online: http://www.rita.dot.gov/bts/sites/rita.dot.gov/bts/files/publications/freight_transportation/pdf/entire.pdf (accessed on 14 February 2023).
30. Bureau, P.R. 2011 World Population Data Sheet. (Washington, DC: 2012). Available online: http://www.prb.org/pdf11/2011_population-data-sheet_eng.pdf (accessed on 1 May 2013).
31. The World Bank, Container Port Traffic. Available online: <https://data.worldbank.org/indicator/IS.SHP.GOOD.TU> (accessed on 14 February 2023).
32. US Dept. of Transportation, Research and Innovative Technology Administration, Bureau of Transportation Statistics. America's Container Ports: Linking Markets at Home and Abroad; Washington, DC, USA. 2011. Available online: https://www.bts.gov/publications/americas_container_ports/2011/pdf/entire.pdf (accessed on 1 May 2013).
33. Magazine, F. Internet Reference. Available online: <http://fortune.com/2018/01/30/port-automation-robots-container-ships/> (accessed on 14 February 2023).
34. Yang, C.H.; Choi, Y.S.; Ha, T.Y. Simulation-based performance evaluation of transport vehicles at automated container terminals. *OR Spectr.* **2004**, *26*, 149–170. [[CrossRef](#)]
35. YorkWallischeck, E. *ICS Security in Maritime Transportation A White Paper Examining the Security and Resiliency of Critical Transportation Infrastructure*; DOT-VNTSC-MARAD-13-01; Bureau of Transportation Statistics: Washington, DC, USA, 2013.
36. Bielli, M.; Boulmakoul, A.; Rida, M. Object Oriented Model for Container Terminal Distributed Simulation. *Eur. J. Oper. Res.* **2006**, *175*, 1731–1751. [[CrossRef](#)]
37. Boulmakoul, A.; Rida, M. Discrete event simulation component specification for container terminals operational management. In Proceedings of the 2nd IEEE ISSPIT (International Symposium on Signal Processing and Information Technology), Marrakesh, Morocco, 18–21 December 2002; pp. 266–271.
38. Organization, P. Internet Reference. Available online: <https://www.pema.org/wp-content/uploads/downloads/2016/06/PEMA-IP12-Container-Terminal-Automation.pdf> (accessed on 14 February 2023).
39. Steenken, D.; Voß, S.; Stahlbock, R. Container terminal operation and operations research—a classification and literature review. *OR Spectr.* **2004**, *26*, 3. [[CrossRef](#)]

40. Cheng, Y.L.; Sen, H.C.; Natarajan, K.; Teo, C.P.; Tan, K.C. Dispatching Automated Guided Vehicles in a Container Terminal. In *Supply Chain Optimization. Applied Optimizatio*; Geunes, J., Pardalos, P.M., Eds.; Springer: Boston, MA, USA, 2005; Volume 98.
41. Zaffalon, M.; Rizzoli, A.E.; Gambardella, L.M.; Mastrolilli, M. Resource allocation and scheduling of operations in an intermodal terminal. In Proceedings of the 10th European Simulation Symposium and Exhibition, Simulation in Industry, ESS98, Nottingham, UK, 26–28 October 1998; pp. 520–528.
42. Günther, H.; Kim, K.H. *Container Terminals and Automated Transport Systems: Logistics Control Issues and Quantitative Decision Support*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2005.
43. Lethbridge, T.C.; Laganière, R. *Object-Oriented Software Engineering: Practical Software Development Using UML and Java*; McGraw-Hill: New York, NY, USA, 2001; Volume 102.
44. Angelov, S.; Grefen, P.; Greefhorst, D. A framework for analysis and design of software reference architectures. *Inf. Softw. Technol.* **2012**, *54*, 417–431. [[CrossRef](#)]
45. Pankowska, M. Stakeholder Oriented Enterprise Architecture Modelling. In Proceedings of the 12th International Conference on e-Business, Colmar, France, 20–22 July 2015.
46. Notteboom, T.; Winkelmans, W. Stakeholder Relations Management in ports: Dealing with the interplay of forces among stakeholders in a changing competitive environment. In Proceedings of the IAME 2002 Maritime Economics: Setting the Foundations for Port and Shipping Policies, Panama City, Panama, 12–15 November 2002.
47. Syed, M.H.; Fernandez, E.B. A reference architecture for the container ecosystem. In Proceedings of the 13th International Conference on Availability, Reliability and Security (ARES 2018), Hamburg, Germany, 27–30 August 2018; ACM: New York, NY, USA, 2018.
48. Saini, K.; Kaur, S. Forensic Examination of Computer-Manipulated Documents using Image Processing Techniques. *Egypt. J. Forensic Sci.* **2016**, *6*, 317–322. [[CrossRef](#)]
49. National Cooperative Freight Research Program, N.C.R.P. *Report 11. Truck Drayage Productivity Guide*; Grant No. DTOS59-06-00039; The National Academies Press: Washington, DC, USA, 2011.
50. The Port of Los Angeles Port Drayage. Available online: <https://www.portoflosangeles.org/> (accessed on 14 February 2023).
51. The Port of Long Beach Port Drayage. Available online: <https://www.polb.com/> (accessed on 14 February 2023).
52. Stallings, W.; Brown, L. *Computer Security: Principles and Practice*, 4th ed.; Pearson: London, UK, 2018.
53. Port Everglades Website, Broward County, FL, USA. Available online: <https://www.porteverglades.net/articles/post/port-everglades-inspects-new-super-post-panamax-cranes/> (accessed on 14 February 2023).
54. Fernandez, E.B.; Sorgente, T.; VanHilst, M. Constrained Resource Assignment Description Pattern. In Proceedings of the Nordic Conference on Pattern Languages of Programs, Viking PLoP 2005, Otaniemi, Finland, 23–25 September 2005.
55. Longo, F.; Padovano, A.; Baveja, A.; Melamed, B. Challenges and opportunities in implementing green initiatives for port terminals. In Proceedings of the 3rd International Workshop on Simulation for Energy, Sustainable Development and Environment, SESDE 2015, Bergeggi, Italy, 21–23 September 2015.
56. Marine Insight. Available online: <https://www.marineinsight.com/guidelines/general-procedure-of-preparing-ships-for-entering-ports/> (accessed on 14 February 2023).
57. Kaluza, P.; Kölzsch, A.; Gastner, M.T.; Blasius, B. The complex network of global cargo ship movements. *J. R. Soc. Interface* **2010**, *7*, 1093–1103. [[CrossRef](#)]
58. Fernandez, E.B.; Ballesteros, J.; Desouza-Doucet, A.C.; Larrondo-Petrie, M.M. Security Patterns for Physical Access Control Systems. In *Data and Applications Security XXI. DBSec 2007. Lecture Notes in Computer Science*; Barker, S., Ahn, G.J., Eds.; Springer: Berlin/Heidelberg, Germany, 2007; Volume 4602.
59. Martínez-Fernández, S.; Ayala, C.P.; Franch, X.; Marques, H.M. Benefits and drawbacks of software reference architectures: A case study. *Inf. Softw. Technol.* **2017**, *88*, 37–52. [[CrossRef](#)]
60. Group, T.B. Container Port Design and Planning. Available online: <https://tba.group/en/services/bulk-container-terminal-design> (accessed on 14 February 2023).
61. Wikipedia, Container Port Design Process. Available online: https://en.wikipedia.org/wiki/Container_port_design_process (accessed on 14 February 2023).
62. Cisco, Ports and Terminals Design Guide. Available online: <https://www.cisco.com/c/en/us/td/docs/solutions/Verticals/CCI/Ports/DG/cci-ports-dg/cci-ports-dg.html> (accessed on 14 February 2023).
63. Martin-Soberon, A.M.; Monfort, A.; Sapina, R.; Monterde, N.; Calduch, D. Automation in port container terminals. *Procedia-Soc. Behav. Sci.* **2014**, *160*, 195–204. [[CrossRef](#)]
64. Maidl, M.; Wirtz, R.; Zhao, T.; Heisel, M.; Wagner, M. Pattern-based modeling of cyber-physical systems for analyzing security. In Proceedings of the 24th European Conference on Pattern Languages of Programs, Irsee, Germany, 3–7 July 2019; Volume 23, pp. 1–10.
65. Hehenberger, P.; Achiche, S. Design, modelling, simulation and integration of cyber physical systems: Methods and applications. *Comput. Ind.* **2016**, *82*, 273–289. [[CrossRef](#)]
66. Tan, Y.; Goddard, S.; Perez, L.C. A prototype architecture for cyber-physical systems. *ACM Sigbed Rev.* **2008**, *5*, 1–2. [[CrossRef](#)]
67. Thurston, T.; Hu, H. Distributed agent architecture for port automation. In Proceedings of the 26th Annual International Computer Software and Applications, Oxford, England, 26–29 August 2002. [[CrossRef](#)]

68. La, H.J.; Kim, S.D. A service-based approach to designing cyber-physical systems. In Proceedings of the 9th IEEE/ACIS International Conference on Computer and Information Science, Washington, DC, USA, 18–20 August 2010; pp. 895–900.
69. Hashizume, K.; Fernandez, E.B.; Larrondo-Petrie, M. *Cloud infrastructure pattern. First International Symposium on Software Architecture and Patterns*; LACCEI: Panama City, Panama, 2012; pp. 23–27.
70. Hashizume, K.; Fernandez, E.B.; Larrondo-Petrie, M.M. Cloud service model patterns. In Proceedings of the 19th International Conference on Pattern Languages of Programs (PLoP2012), Tucson, AZ, USA, 19–21 October 2012.
71. Alnaim, A.K.; Alwakeel, A.M.; Fernandez, E.B. Towards a Security Reference Architecture for Network Function Virtualization. *Sensors* **2022**, *22*, 3750. [\[CrossRef\]](#)
72. Nota, G.; Bisogno, M.; Saccomanno, A. A service-oriented approach to modeling and performance analysis of Port Community Systems. *Int. J. Eng. Bus. Manag* **2018**, *10*. [\[CrossRef\]](#)
73. Verissimo, P. CRUTIAL: Towards a reference critical information infrastructure architecture. *Eur. CIIP Newsl.* **2007**, *3*, 6–8.
74. Osório, A.; Afsarmanesh, H.; Camarinha-Matos, L. Towards a Reference Architecture for a Collaborative Intelligent Transport System Infrastructure. In *Collaborative Networks for a Sustainable World, IFIP Advances in Information and Communication Technology*; Springer: Berlin, Germany, 2010; Volume 336, pp. 469–477. [\[CrossRef\]](#)
75. Zambrano, G.R.; Vera, L.O. Reference architecture for an intelligent transportation system. *Int. J. Innov. Appl. Studies* **2016**, *15*, 2028–9324.
76. Shi, J.; Wan, J.; Yan, H.; Suo, H. A survey of cyber-physical systems. In Proceedings of the International Conference on Wireless Comm. and Signal Processing, Nanjing, China, 9–11 November 2011. [\[CrossRef\]](#)
77. King, M.; Zhu, B.; Tang, S. Optimal Path Planning. *Mob. Robot.* **2001**, *8*, 520–531.
78. Cardenas, A.A.; Amin, S.; Lin, Z.-S.; Huang, Y.-L.; Huang, C.H.; Sastry, S. Attacks against process control systems: Risk assessment, detection, and response. In Proceedings of the ASSIACS'11, Hong Kong, China, 22–24 March 2011; pp. 355–366.
79. Miller, A. Trends in process control systems security. *IEEE Secur. Priv.* **2005**, *3*, 57–60. [\[CrossRef\]](#)
80. Rahman, H.A.; Beznosov, K. SPAPI: A security and protection architecture for physical infrastructures and its deployment strategy using wireless sensor networks. In Proceedings of the 10th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2005), Catania, Italy, 19–22 September 2005; pp. 885–892.
81. Liu, Y.; Peng, Y.; Wang, B.; Yao, S.; Liu, Z. Review on Cyber-physical Systems. *IEEE/CAA J. Autom. Sin.* **2017**, *4*, 17–40. [\[CrossRef\]](#)
82. Spichkova, M.; Schmidt, H.; Peake, I. From abstract modelling to remote cyber-physical integration/interoperability testing. *arXiv* **2014**, arXiv:1403.1005.
83. Krotofil, M.; Gollmann, D. Industrial control systems security: What is happening? In Proceedings of the 2013 11th IEEE International Conference on Industrial Informatics (INDIN), Bochum, Germany, 29–31 July 2013; pp. 664–669. [\[CrossRef\]](#)
84. Microsoft Power and Utilities, Smart Energy Reference Architecture, Oct. 2009. Available online: <https://www.microsoft.com/Utilities> (accessed on 14 February 2023).
85. Guth, J.; Breitenbücher, U.; Falkenthal, M.; Leymann, F.; Reinfurt, L. Comparison of IoT platform architectures: A field study based on a reference architecture. In Proceedings of the 2016 Cloudification of the Internet of Things (CIoT), Paris, France, 23–25 November 2016. [\[CrossRef\]](#)
86. Prakash, A.; Satish, M.; Bhargav, T.S.S.; Bhalaji, N. *Detection and Mitigation of Denial of Service Attacks Using Stratified Architecture*; Elsevier: Amsterdam, The Netherlands, 2016. [\[CrossRef\]](#)
87. Ganjkhani, M.; Fallah, S.N.; Badakhshan, S.; Shamshirband, S.; Chau, K.-W. A Novel Detection Algorithm to Identify False Data Injection Attacks on Power System State Estimation. *Energies* **2019**, *12*, 2209. [\[CrossRef\]](#)
88. Liang, G.; Zhao, J.; Luo, F.; Weller, S.R.; Dong, Z.Y. A Review of False Data Injection Attacks Against Modern Power Systems. *IEEE Trans Smart Grid* **2016**, *8*, 1630–1638. [\[CrossRef\]](#)
89. Hosseinzadeh, M.; Sinopoli, B. Active Attack Detection and Control in Constrained Cyber-Physical Systems Under Prevented Actuation Attack. In Proceedings of the American Control Conference (ACC), New Orleans, LA, USA, 25–28 May 2021. [\[CrossRef\]](#)
90. Loukas, G.; Vuong, D.G.T. A Review of Cyber Threats and Defence Approaches in Emergency Management. *Future Internet* **2013**, *5*, 205–236. [\[CrossRef\]](#)
91. Jensen, J.C. A model-based design methodology for cyber-physical systems. In Proceedings of the 7th International Wireless Communications and Mobile Computing Conference, Istanbul, Turkey, 4–8 July 2011; pp. 1666–1671.
92. Jacky, J. Specifying a Safety-Critical Control System in Z. *IEEE Trans. Softw. Eng.* **1995**, *21*, 388–402. [\[CrossRef\]](#)
93. Karsai, G.; Balasubramanian, D.; Dubey, A.; Otte, W.R. Distributed and managed: Research challenges and opportunities of the next generation cyber-physical systems. In Proceedings of the 2014 IEEE 17th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing, Reno, NV, USA, 10–12 June 2014; pp. 1–8.
94. Uzunov, A.; Fernandez, E.B.; Falkner, K. ASE: A Comprehensive Pattern-Driven Security Methodology for Distributed Systems. *J. Comput. Stand. Interfaces* **2015**, *41*, 112–137. [\[CrossRef\]](#)

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.