**A pattern for a secure and safe port loading facility**

Eduardo B. Fernández1, Raúl Monge2, and René Carvajal2

1 Florida Atlantic University, Boca Raton, FL, USA

2 Universidad Técnica Federico Santa María, Valparaíso, Chile

**Abstract**

A cyber-physical system (CPS) integrates computing and communication capabilities with the monitoring and control of entities in the physical world. An important type of CPSs are maritime ports. Ports are vulnerable systems and a variety of attacks are possible; they are also safety-critical systems. We present here a pattern for a secure and safe port loading facility, where we have added security and safety patterns on a model for a port loading facility that describes the basic operations of such a system. The specific security and safety patterns are deduced following a systematic methodology.

**Keywords**. Cyber-physical systems, security patterns, safety patterns, port loading, maritime transportation

**1. Introduction**

A cyber-physical system (CPS) integrates computing and communication capabilities with the monitoring and control of entities in the physical world. Such control systems are normally distributed, real-time, and usually include embedded devices, sensors, and wireless links [Shi11]. Many system components are remotely deployed, have unique constraints, and may be physically inaccessible for maintenance, and/or physically accessible for attack. Examples include transportation systems, smart power grids, patient monitoring, smart buildings, flexible manufacturing systems, and many others. Often, CPSs are safety-critical[[1]](#footnote-1), because their failure could endanger lives or cause large economic losses. Their safety can be compromised by poor design, low reliability, or security attacks.

For economic and productivity reasons, open networks are an attractive communication medium, but the possibility to connect to physical systems through cyberspace increases their vulnerability to intentional attacks. The objectives of attacks may vary, from terrorist objectives to economic objectives such as collecting private information. The complexity of modern CPSs makes their design and maintenance very difficult: we need to determine all forms of attacks and failures, which can use any of the possible inputs to these systems. Cyber-Manipulation of container logistics in a port has become a realistic attack vector that is even reported on in the news [Bat13], *Stuxnet* demonstrated the feasibility of attacking physical systems using worms [Den12] and similar attacks are possible and expected [Kat11, Lee13]. Security attacks in CPSs such as transportation systems can affect the reliability and safety of those systems and may have catastrophic consequences [GAO, NIT12, Por].

Secure systems need to be built in a systematic way where security is an integral part of the lifecycle [Fer13, McG06], and the same applies to safety. If, when we build applications, we also consider the effect of middleware, operating systems, and networks as a whole, we can build systems that exhibit a unified architecture where we can identify attacks and apply global solutions. The platform should match the type of application, and all safety and security constraints should be defined at the application level, where their semantics are understood and propagated to the lower levels. The lower levels must provide the assurance that the constraints are being followed, i.e., they implement these constraints and enforce that there are no ways to bypass them. Following these ideas, we developed a secure systems development methodology [Fer06a, Fer13a], which considers all lifecycle stages and all architectural levels. We expanded its architectural aspects [Fer11], and recently added process aspects. The only way to provide this unification in the presence of myriad implementation details of the component units is to use abstraction. In particular, we can apply abstraction through the use of *patterns*. The description of architectures and mechanisms using patterns makes them easier to understand, provides guidelines for design and analysis, and can define a way to make their structure more secure. Their abstraction properties make them ideal for dealing with complex and heterogeneous systems such as CPSs. Other approaches include [Ban12, Mon11]

The application of our methodology requires a relatively complete catalog of patterns, covering all architectural levels, as well as Computation, Communications, and Control aspects. To select patterns we need to enumerate the threats to the system. We proposed a method that analyzes the activities in a use case to see how they can be subverted by an attacker [Bra08]. We showed that this approach works well with IT systems. However, its application to CPSs is not so clear. CPSs have a larger variety of threats and these threats are specific to the type of system.

We present here a pattern for a port loading facility. Today, U.S. port facilities rely as much upon networked computer and control systems as they do upon stevedores to ensure the flow of maritime commerce that the economy depends upon. We analyze here the specific threats to this type of CPS and we provide defenses against them. This approach results in a Secure Port Loading Facility, where all the identified threats have been handled. This pattern is what we call a *Secure Semantic Analysis* pattern (SSAP) [Fer00, Fer13a]. A SSAP is a secured analysis pattern implementing a small coherent set of use cases.

We use the template of [Bus96] to present our pattern. Our audience includes CPSs designers as well as designers of applications that need to run in CPSs. For conciseness, we only show in detail one of its use cases.

**2. Secure and Safe Port Loading Facility**

**Intent**

Provides the typical functions of a port loading facility (loading and unloading of containers to/from a ship) including security and safety mechanisms that can defend against all identified threats.

**Context**

Port operations and logistics have a significant impact on 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 an increased automation. For this reason, ports are sensitive to different types of threats and challenges. One type of threats is natural disasters such as earthquakes and hurricanes. Another type of threat is defined by terrorist attacks or data security breaches. 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 operations which are: seaside operations, yardside operations, and landside operations. Each area could be sensitive to different types of threats [Kra13]. Threats that a container terminal may face include cyber attacks, radiation attacks, biological attacks, and explosions. We specifically consider cyber attacks—with the possibility of other attacks being triggered in this way**.** A port logistic system which must keep track of many container positions is an example of a data-driven safety-critical system. Security is especially critical in these systems because an unauthorized modification of their data could affect the safety of people, have a serious economic impact, or do extensive damage.

**Problem**

The lack of security and safety in port loading facilities can be a serious problem because of the importance of these activities and the potential high impact of attacks. How can we build a secure and safe port loading facility?

**Forces.**

* ***Flexibility.*** A variety of users are involved in this system. 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.
* ***Threats.*** The variety of users and the fact that the facility is connected to the Internet bring many threats. We should be prepared for a variety of attacks.
* ***Safety.*** The equipment has physical limits that cannot be exceeded because they would get damaged or endanger the port workers.
* ***Secure interoperation***. We need to verify that the ship has an appropriate level of security, able to interact securely with our facility.
* ***Records.*** We need to record all activities that may need to be audited later to investigate security or safety violations.

**Solution**

Ports include loading/unloading facilities such as cranes and storage units. During unloading, operators move containers from ships to storage units, during loading operators move containers from storage units to ships. Figure 1 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**. Typical use cases include Load container into a ship, Unload container from a ship, Assign crane operator, and Assign locations to the containers. Typical roles include: load supervisor, crane operator, ship worker, and storage worker.

***Enumerating threats***

Figure 2 shows the activity diagram for the use case Unload a container. For each activity we can analyze its possible threats. We do not try to be exhaustive, only to show the procedure to enumerate threats.

Some of the threats are: a1: activation threats: no activation or unwanted activation; a2: crane control threats: accelerate beyond limit, wrong movements; a3: pick up threats: wrong container; a4: deposit container in the wrong place; a5: hatch is not opened, and a6: log threats: write wrong entry or delete log.



Figure 1 Class diagram of Port-Load system



Figure 2. Activity diagram for use case Unload a container from a ship

***Secure solution***

The identified threats can be stopped by using Authentication and Authorization patterns (RBAC) if they come from external attackers (a1,a5, a6). If they come from internal attackers (a2,a3,a4) we can use the Security Logger/Auditor pattern. To enforce the constraint that only supervisors can assign crane operators we use the Content-Dependent Authorizer pattern.

We need to add OCL assertions for defining constraints for security to stop the identified threats and safety assertions that avoid unsafe conditions. These assertions realize pre-specified policies based on regulations and equipment constraints. Possible assertions and security constraints include:

***Safety:*** ∀ crane [active] 🡪 ∃ crane\_operator (every crane in active state must have a human operator)

aCrane.speed ≤ Speed\_Limit ; aCrane\_acceleration ≤ Acceleration\_Limit

***Security:*** aCrane\_operator.assign = aPortLoadSupervisor.assign (only port loading supervisors can assign crane operators)

We add security mechanisms realized as patterns to stop the identified attacks. If we can cover all the possible threats we can consider the system as secure. We also need to realize the predefined policies.

**Implementation**

We can consider first 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 the other roles. Their specific members and rights may change frequently.

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.

**Consequences**

The advantages of this pattern include:

* ***Flexibility.*** RBAC allows to accommodate any kinds of roles, appropriate for all kinds of users.
* ***Threats.*** We can enumerate threats systematically. We do not know if we found all of them but we can control those we find by matching them to their defenses (security patterns). Using the semantics of the application we can also estimate their impact.
* ***Safety.*** The equipment physical limits can be controlled with safety assertions.
* ***Secure interoperation***. We can list security requirements for the ships in the form of security patterns.
* ***Records.*** We can use the Security Logger / Auditor pattern to record security-relevant activities.

The pattern also has some liabilities:

The security and safety mechanisms have a cost, have some overhead, and require maintenance.

**Known uses**

* Container companies in the port of Antwerp, Belgium, after suffering a serious attack, are applying some of these defenses [Bat13].
* The US and China are exchanging best practices on how to defend port facilities, which include defenses of the type discussed here [Col14].

**Related patterns**

***Authenticator*** [Fer13a]. When a user or system (subject) identifies itself to the system, how do we verify that the subject intending to access the system is who it says it is?

# *Role-Based Access Control* [Fer13a]. Describe how to assign rights based on the functions or tasks of people in an environment in which control of access to computing resources is required and where there is a large number of users, information types, or a large variety of resources.

***Security Logger/Auditor*** [Fer13a]. How can we keep track of user’s actions in order to determine who did what and when? Log all security-sensitive actions performed by users and provide controlled access to records for Audit purposes.

***Content-Dependent Authorizer*** [Fer14]. How can we restrict subjects (users, systems, parties) to access (read, write) only data with specific values? Filter the values obtained from applying authorization rules to the data according to a predicate (condition) that selects specific values.

**Conclusions**

This pattern provides a starting point to build secure port facilities. We can build a Secure Port Reference Architecture [Tay10], which would combine this pattern with patterns for container logistics, ship scheduling, and similar. We need more safety patterns, few exist [Bit01, Cof11].

**Acknowledgements**

We thank our shepherd, Eduardo Guerra, who under time pressure was able to provide insightful comments that significantly improved this paper.

**References**

|  |
| --- |
|  |

[Ban12] A. Banerjee, K.K. Venkatasubramanian,T. Mukherjee, and S.K.S. Gupta, “Ensuring safety, security, and sustainability of mission-critical cyber-physical systems”, *Procs. of the IEEE*, vol. 100, No 1, January 2012, 283-299.

[Bat13] Tom Bateman, “Police warning after drug traffickers' cyber- attack”,

[http://www.bbc.co.uk/news/world-europe-24539417](https://exchange.fau.edu/owa/redir.aspx?C=5VbHE1kEYEi6n001OCANrtCKvIxv6tAIZyjmP7O7k7URg4ENQlR4jaJaVtAZB-5GoKusXxhjTKE.&URL=http%3a%2f%2fwww.bbc.co.uk%2fnews%2fworld-europe-24539417), 16 October 2013.

[Bit01] Friedemann Bitsch, “Safety Patterns – The Key to Formal Specification of Safety Requirements”,

SAFECOMP 2001, LNCS 2187, pp. 176–189, 2001.

<http://theory.sci.univr.it/papers/Isa/Sicurezza/SafLiv/safety2.pdf>

[Bra08] F. Braz, E.B.Fernandez, and M. VanHilst, "Eliciting security requirements through misuse activities" *Procs. of the*  *2nd* *Int. Workshop on Secure Systems  Methodologies using Patterns (SPattern'08)*.  In conjunction with the *4th  International Conference onTrust, Privacy & Security in Digital Business (TrustBus'08)*, Turin, Italy, September 1-5, 2008.  328-333.

[Bus96] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, M. Stal. *Pattern-Oriented*

*Software Architecture: A System of Patterns, Volume 1*, Wiley, 1996.

[Coc07] T.J. Cockram and S.R. Lautieri, “Combining security and safety principles in practice”, *Procs. 2nd IET Int. Conf. on Sys. Safety,* 2007, 159-164.

[Cof11] Darren Cofer, “Complexity-reducing design patterns for cyber-physical systems”, Rockwell Int. Rept., 2011

[Col14] Columbia University, School of International and Public Affairs, “U.S.-China Cybersecurity Cooperation”, Capstone Report, June 2014.

https://sipa.columbia.edu/sites/default/files/AY14\_CyberCooperation\_FinalReport.pdf

[Den12] D. E. Denning, “Stuxnet: What has changed”, *Future Internet*, vol. 4, 2012, 672-687. doi: 10.3390/fi4030672

[Fer00] E. B. Fernandez and X. Yuan, “Semantic analysis patterns”, *Proceedings of the* *19th Int.*

*Conf. on Conceptual Modeling, ER2000*, 183-195. Also:

<http://www.cse.fau.edu/~ed/SAPpaper2.pdf>

[Fer06a] E. B. Fernandez, M.M. Larrondo-Petrie, T. Sorgente, and M. VanHilst, "A methodology to develop secure systems using patterns", Chapter 5 in "*Integrating security and software engineering: Advances and future vision",* H. Mouratidis and P. Giorgini (Eds.), IDEA Press, 2006, 107-126.

[Fer11] E. B. Fernandez, Nobukazu Yoshioka, Hironori Washizaki, and Michael VanHilst, "An approach to model-based development of secure and reliable systems", *Procs.* *Sixth International Conference on Availability, Reliability and Security (ARES 2011)*, August 22-26, Vienna, Austria.

[Fer13] E.B.Fernandez, *Security patterns in practice: Building secure architectures using software patterns*, Wiley Series on Software Design Patterns, 2013.

[Fer14] E. B. Fernández, Raúl Monge, René Carvajal, Oscar Encina, Juan Hernández, and Paulina Silva, “Patterns for Content-Dependent and Context-Enhanced Authorization”, *Procs. of EuroPLoP 2014.*

[GAO14] U.S. Government Accountability Office, Rept. 14-459, “Maritime infrastructure protection: DHS needs to better address port cybersecurity”, 2014.

[Kat11] S. Katzenbeisser, “Can trains be hacked? Die Technik der Eisenbahnsicherungsanlagen*.”*, presentation at *28th Chaos Computer Congress*, 2011, http://www.youtube.com/watch?v=C3sGgRtsTbg

(in German).

[Kra13] J. Kramek, “The Critical Infrastructure Gap: U.S. Port Facilities and Cyber Vulnerabilities “, Brookings Inst., 2013, <http://www.brookings.edu/~/media/research/files/papers/2013/07/02%20cyber%20port%20security%20kramek/03%20cyber%20port%20security%20kramek.pdf>

[Lee13] M. Lee, “Hackers could control Brisbane traffic controls: Report I”

<http://www.zdnet.com/au/hackers-could-control-brisbane-traffic-controls-report-7000023405/>

[McG06] G. McGraw, *Software security: Building security in*, Addison-Wesley 2006.

[Mon11] M. Montanari, R.H. Campbell, K. Sampigethaya, and M. Li, “A security policy framework for sEnabled fleets and airports”, *IEEE Aerospace Conference*, 2011, 1-11.

[NIT12] Executive Office of the President, National Science and Technology Council, *The Networking and Information Technology Research and Development (NITRD) Program, 2012 Strategic Plan,* July 2012.

[Por] Port security, <http://en.wikipedia.org/wiki/Port_security>

[Shi11] J. Shi, J. Wan, H. Yan, and H. Suo, “A survey of cyber-physical systems”, *Proc. Int. Conf. on Wireless Comm. and Signal Processing,* Nanjing, China, Nov.9-11, 2011

[Tay10] R.N.Taylor,N. Medvidovic, and N.Dashofy. *Software architecture: Foundation, theory, and practice,* Wiley, 2010.

[Van09] M. VanHilst, E.B.Fernandez, and F. Braz, "A multidimensional classification for users of security patterns", *Journal of Research and Practice in Information Technology,* vol. 41, No 2, May 2009, 87-97.

1. Safety is the freedom from unacceptable risks, including threats to human lives, the environment, or to costly facilities. Safety is often expressed with assertions on how to avoid unsafe conditions, e.g., an elevator must not open its doors when moving. These assertions are usually related to specific states of a control system. [↑](#footnote-ref-1)