Building a Reference Architecture for Cargo Ports using Patterns

Virginia M. Romero, Eduardo B. Fernandez
Florida Atlantic University, USA. vromero@fau.edu, *fernande@fau.edu*

Abstract*–*Cyber-Physical Systems (CPS) are physical entities whose operations are monitored, coordinated and controlled by a complex network of computers, software and communication technologies. These systems are heterogeneous and present a myriad of implementation details. In order to design efficient and dependable CPS systems we must abstract their cross domain complexity and develop a common framework, namely a Reference Architecture (RA). RAs describe at an abstract level and with no implementation details, the main functions of the system. The use of reference architectures and patterns is an effective way to organize and describe not only the functional and non-functional aspects of the system but also to unify the design of the computational, communication and control aspects of CPSs. In this paper we propose a method to build a Reference Architecture for CPS using patterns and UML models that go beyond existing models to provide a global view and a more precise description of the system. The design and analysis is given using a Cargo Port as our example, but the approach can be used in other domains as well.

Keywords—CPS, Reference Architectures, Security Reference Architectures, Cargo Port, Patterns, Process Pattern.

I. Introduction

Cyber Physical Systems (CPS) are systems that integrate physical processes, computational resources, and communication capabilities with the control of entities in the physical world. The components of a CPS can be centralized or distributed and usually include embedded devices, sensors, and wireless links. Many system components are remotely deployed, have unique constraints and may be physically inaccessible for maintenance but not for attacks.

CPSs are very complex systems and comprise a multitude of systems, components and implementations. To understand their interaction and solidify their design, we need to start from their global architectures. A *reference architecture* (RA) is a generic software architecture, based on one or more domains, that describes functionality without any implementation details. An RA is reusable, extendable and configurable; it also provides standards for interoperability and portability. It is a pattern for whole architectures and it can be instantiated into a specific software architecture by adding platform aspects [1]. *Patterns* are encapsulated solutions to recurrent system problems and define a vocabulary that concisely expresses requirements and solutions as well as provide a communication vocabulary for the various stakeholders [3, 4]. Building reference architectures using patterns becomes a powerful way to organize and describe security and other non-functional aspects of a system as well. But, specifically for CPSs, these abstraction artifacts have become not only very useful to understand and build these complex systems but have the potential to unify the design of the computational, communication and control aspects of CPSs, especially in the presence of the countless implementation details of the component units of these complex systems.

Digital Object Identifier (DOI): http://dx.doi.org/10.18687/LACCEI2018.1.1.536
ISBN: 978-0-9993443-1-6
ISSN: 2414-6390

There are many types of cyber physical systems and they can be found in areas as diverse as aerospace, automotive, chemical processes, healthcare, manufacturing and transportation; even entertainment and consumer appliances. The construction of a general CPS reference architecture will be rather vague so to be more precise we will take a specific type of system, a transportation system, specifically a maritime container cargo port, and we will build an RA. In previous work, we reviewed the literature [38] and found that there are not many reference architectures for CPS, much less for cargo ports. We propose here an object oriented approach for building RAs using UML models and patterns. We consider this effort the first attempt to define a precise and semiformal architecture for these systems. We believe that a semiformal approach is the only practical approach given the complexity of the system 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 [5], and we can make this architecture more formal if needed, although purely formal methods are difficult for most practitioners [6].

The objective of this paper is to build a reference architecture for container cargo ports using patterns. Pattern based architectural models have been found useful in representing complex systems. Architectural models provide holistic and unified views of the system. We will build a structure based on patterns for this specific type of CPS, which can be used as a systematic approach to building reference architectures for transportation systems and others types of CPS. This is by no means a complete system but we believe that many of its functions are variations or repetitions of some idea and there is no need to show every aspect; we only want to show an approach for building such an architecture, demonstrate its value, and build some of its parts to illustrate the approach. We are including a pattern for a cargo port loading and unloading facility [7] and a pattern for the secure delivery of containers [8] to describe its components. In future work, we will also add security patterns to control threats and define a *Security Reference Architecture (SRA)* for cargo ports .

This paper is organized as follows: Section II presents some necessary background information. Section III presents a description of container cargo ports and their operations. Section IV presents a partial reference architecture using patterns. Related work and Conclusions are presented in Section V and Section VI respectively.

II. Background

An important development in software is the concept of a pattern: a solution constructed to address a recurrent problem in a given context. A pattern embodies the knowledge and experience of software developers that can be reused in new applications. The typical solution provided by a pattern comes in the form of a UML class diagram complemented with some sequence diagrams, and possibly activity or state diagrams. OCL constraints can add formality if needed. A set of consequences indicate how well the forces were satisfied by the solution as well as the possible negative aspects of the solution. 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 ones that complement the pattern or that provide alternative solutions [3]. This level of detail and precision allows designers to use them as guidelines and users to understand the effect of the mechanisms they represent. Patterns are also good for communication between designers and to evaluate and to reengineer existing systems [9]. Patterns are abstractions of best practices and, as such, they are not based on a formal model, although their solutions can be formalized. A process pattern is a specific type of pattern that is used to develop object oriented software. A process is a series of actions in which one or more inputs are used to produce one or more outputs. A process pattern describes a collection of general techniques, actions and/or tasks (activities) for solving a process related problem. It identifies the environment in which the problem has been encountered and suggests one or more proven solutions to the problem [2].

A reference architecture (RA) is a standardized, generic architecture, valid for a particular domain that does not contain implementation details [1, 11], a pattern for whole architectures that can be instantiated into whole architectures by adding platform aspects. There is no general agreement about what an RA should contain. Avgeriou [1] presents an example and describes what should be included in one: A class diagram and a set of use cases with their roles. Previous work on Semantic Analysis Patterns [12], was a step towards building RAs out of patterns, an idea also used in [13, 14].

The Unified Modeling Language (UML) is used for describing this reference architecture. UML is a semiformal language whose syntax is formally defined using a metamodel [15]. It is an industry standard, it is widely used by practitioners and many tools support its use. UML has proved its value for the creation of high-quality software. Being a graphic language, it is highly intuitive and it almost corresponds directly to code. Its diagrammatic notation for visually modeling object-oriented systems is now the accepted standard for software development. There is an extensive amount of design and security patterns and most of them describe their solutions using UML. While using models is not always easy or convenient, we believe that this semi-formal approach is the only practical approach given the complexity of the systems we are considering; of course, parts of the architecture can be formally modelled, especially for safety-critical systems. Purely formal methods cannot handle such complex systems and are not appropriate to be used by most practitioners. Completely formalizing a pattern makes it too rigid and violates the idea that patterns are mostly suggestions. In complex systems, developers do not have the same level of control over every component's development and evolution. A development methodology for secure systems must address the issues posed by legacy subsystems, Commercial Off-The-Shelf (COTS) components, library components, and outsourcing relationships. We believe these heterogeneous architectures can be well described by UML.

III. Cargo Ports

For millennia, mankind has shipped goods across the oceans, from one land to another. The loading and unloading of a ship has always been very labour intensive. A ship could spend easily more time in port than at sea while dock workers handled cargo into 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. 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 [16]. 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 [17], it accounts for 9 percent, of worldwide container traffic, with one container out of eleven engaged in global trade either bound for or originating in the U.S [18].

Port automation has been playing an increasing 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 otherwise boost port productivity by up to 30% by some estimates [39].

A. Container Terminal Overview

Container terminals provide many services i.e. container loading/unloading to/from vessel 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 need several shared and reusable resources and equipment to fulfill the tasks involved in handling and transporting containers; such as, quay cranes, yard cranes, transport vehicles, e.g. multi-trailers or automatically guided vehicles, straddle carriers, yard stacking deposits, automatic stacking cranes or automatic storage/retrieval systems, railway tracks, human operators. All processes and operations are usually planned, scheduled, monitored, and controlled by a central supervisor and make use of information technologies, to allow fast ship operations, optimization of the usage of facilities, and to reduce lag times [19].

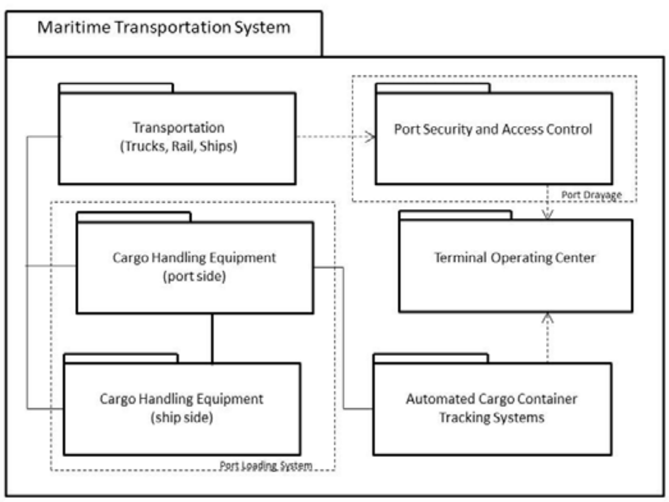

Figure 1. Package diagram of a Maritime Cargo Port System

A typical modern container terminal installation is represented in Figure 1. using UML packages notation. It includes: An area for Port Security and Access Control, this incorporates the physical entrance to the terminal, video surveillance (CCTV), gates, and the infrastructure necessary to check the credentials of the maritime transportation workers that require unescorted access to the secure areas of the port. These biometric identification cards are also known as TWIC (Transport Worker Identification Card). Another component is the Terminal Operating Center. This incorporates the financial, communications, customs, security and other back office functions. An automated Cargo Container Tracking System; an area for intermodal transportation of the containers using commercial long-haul trucks or a railway interface. Cargo Handling Equipment at the port side and the ship side, automated cargo handling equipment, vehicles and similar conveyances [19].

B. Container Terminal Operations

Container terminals work day and night and their amount of work deals with the quantity 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 [20]. A quay crane is a crane that services 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. Quay cranes provide the single most important operation (called a *move*) associated with the loading and unloading of a ship, and represents the only means of moving containers to or from a ship [21]. To unload a ship, one or several quay cranes pick up containers from the ship and put them on shuttle trucks that move them to the assigned yard positions within the terminal storage area. To load a ship the quay crane unloads a container from the shuttle truck and puts it in a ship. Containers stacked in a ship are secured in several ways in order 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 to 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 showed more densely than in non-cellularized cargo vessels) are already on board.

Operations in the storage yard are more flexible than quay crane operations. In a storage yard, yard gantry cranes, top-pick loaders or straddle carriers are used to stack containers. The container storage area is usually separated into different stacks (or blocks) which 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 [23]. Containers are assigned specific addresses before entering the storage yard.

Another important element of a container terminal port is the movement of containers between the quay cranes and the storage yard. Quay cranes unload a ship and place the containers in *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 accurate positioning of these vehicles [24]. The number of shuttle trucks also needs to be considered. Too many trucks in the system cause long files at the crane and long waiting times for service, conversely, few trucks in the system will result in idle stacking equipment. Containers, which are stored in the storage yard, leave the terminal by 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 quay crane operating without delay in order to turn the ship around as quickly as possible.

A great variety of container terminals exist mainly depending on what type of handling equipment is used to form the system. The decision on which equipment to use depends on several factors. Space restrictions, economical reasons, even historical reasons play an important role. A basic factor is the dimension of the space which 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 case of high labour costs and new terminal construction. Historical and cultural reasons have to be considered if container terminals are enhanced or modernized. Because space is becoming a scarce resource, a tendency for higher storage is to be foreseen [42]. Figure 2. presents a schematic side view of a container terminal system.

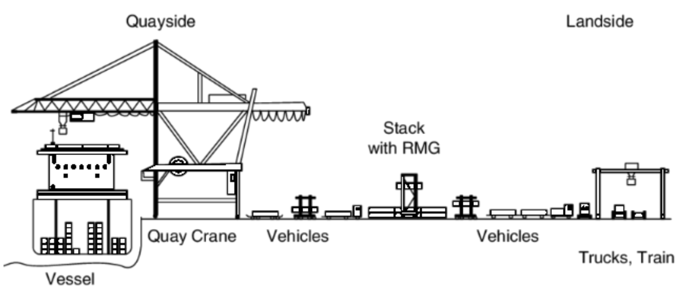

Figure 2. Container Terminal System [24]

C Structure Model of a Container Terminal

Using a UML notation class diagram, shown in Figure 3. we can represent the static structure of the container terminal system and the relationship of its components. The class *Terminal* represents the container terminal in our system. It aggregates multiple objects of classes *Quay, TransportMean, Area and Crane*, these classes represent the terminal components. Quay represents the quays of the terminal, the *TransportMean* is a super class of 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 another ship. *Area* is another super class representing several surfaces in the terminal, *StorageArea* to park the containers, *BufferArea*, a temporary storage area. Crane super class of all types of cranes, *QuayCrane* that service the ship, *YardCrane, GantryCrane* employed to stack containers.

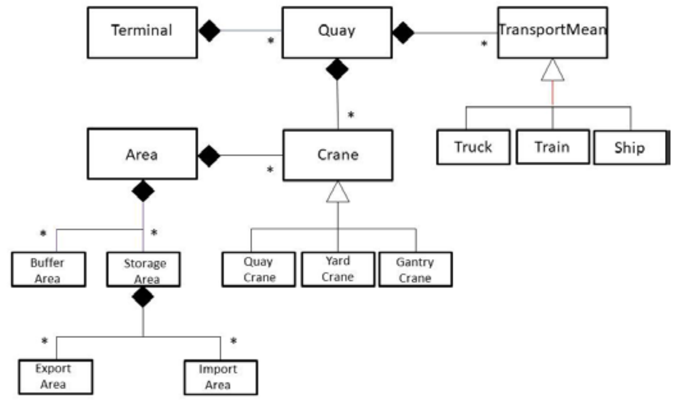

Figure 3. Structure Model of a Container Terminal System

IV. Modeling the Reference Architecture

The description of architectures and mechanisms using patterns provides solid guidelines for design and analysis and makes any system easier to understand. Their abstraction properties make them ideal for dealing with highly heterogeneous systems such as CPSs. The idea here is that we can create process patterns for all the interactions (use cases) in the system and use these process patterns to build their RA. We describe a pattern for the loading and unloading of containers to/from a ship at a cargo port facility [7] and a pattern for the secure delivery of containers at a cargo port to demonstrate our approach [8]. Here we only include pattern descriptions and a diagram to show how they can be used to build a partial reference architecture for a Cargo Port. Complete patterns can be found in their respective references.

A. Secure and Safe Cargo Port Loading Facility pattern

Intent

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

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. 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.

B. A Pattern for Secure Cargo Port Drayage

Intent

This pattern provides all the typical functions for the secure delivery and pick up of containers at a maritime cargo port.

Solution

Every maritime port container terminal includes a Port Security and Access Control component. Drivers arriving at a container terminal entrance gate intend to either drop off a loaded export container, or an empty container; and/or to pick up a loaded import container, or an empty container. The variety of external users and the fact that the contents of the containers are usually not in plain sight bring many threats, so continuous checks are required, not only to the individuals entering the terminal but also to 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 location recorded for identification. All visits should be logged. Typical use cases include: Assign locations to the containers, Assign drivers to the trucks. Typical roles include: gate attendant, storage yard supervisor, storage yard worker, truck driver, gate access worker.

C. Partial RA of a Cargo Port using Patterns

As a way to demonstrate the usefulness of our approach, we present a partial reference architecture for a cargo port as a collection of patterns (see Figure 4). We have included patterns for a port loading facility, a cargo port drayage and a mediator pattern for ship arrivals and departures, demarked by dashed lines. This is not a complete RA for Cargo Ports but can be used as a framework to guide you in the design of a cargo port system.

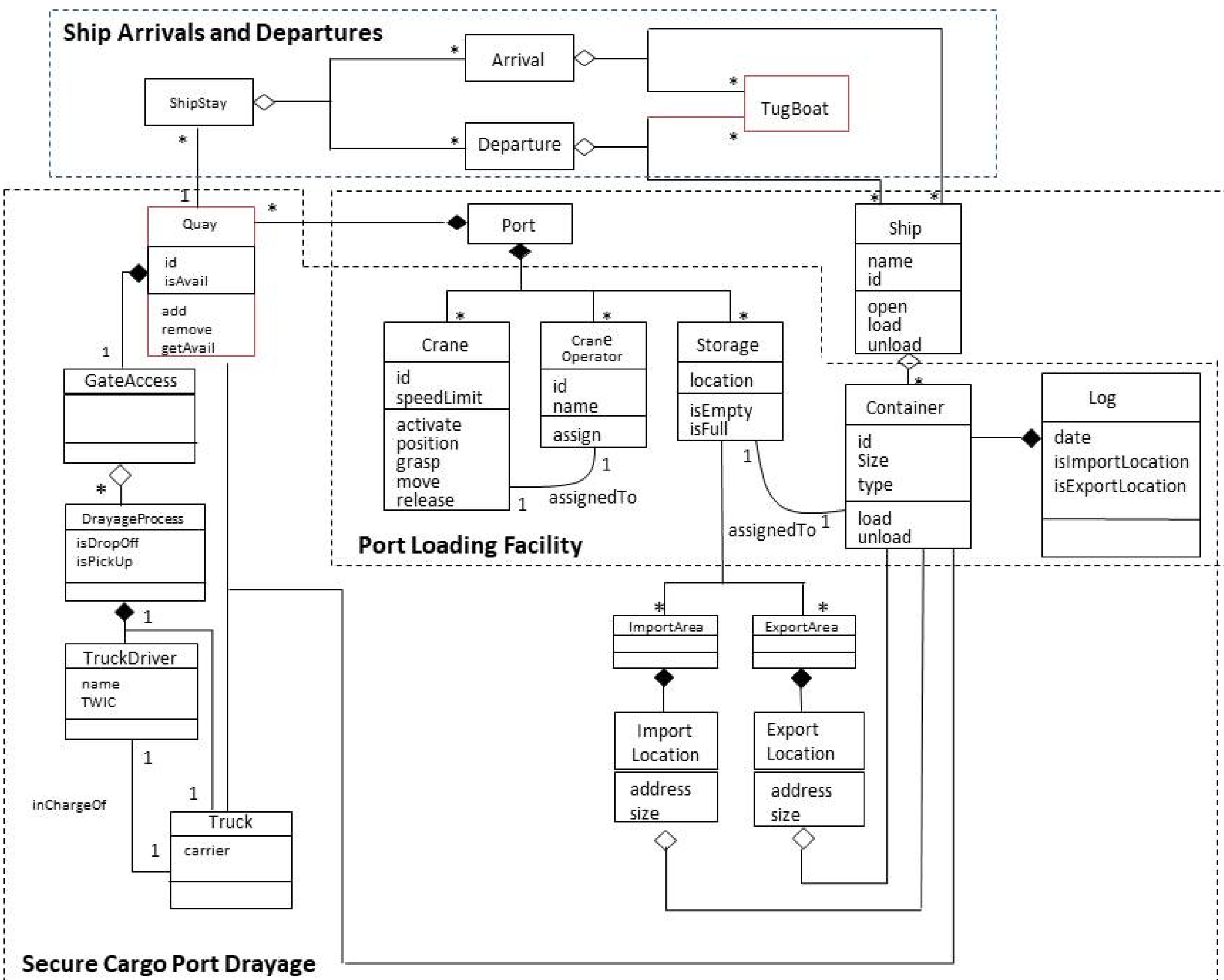

Figure 4. Partial Reference Architecture for a Cargo Port using Patterns

V. RELATED WORK

There are not many reference architectures for Cyber Physical Systems, much less for cargo ports. The literature mentions the possibility of using domain models but no details are given [41]. Refs. [25, 26, 27, 41] all give basic block diagram architectures with short description of the units. A more detailed architecture is given in [28]. Ref. [29] uses a SOA approach to design CPSs, including patterns but their approach is oriented to dynamic composition of services. Microsoft defined an RA for smart grids [30], and [31] described a secure architecture for smart grids. These descriptions are not detailed or formal enough to analyse threats. Our SCADA model [32] goes one step beyond them and it is described as a pattern; we used it as a starting point.

Some papers, e.g., [27], propose implementation-oriented architectures, [27] is based on wireless networks. Our approach is to start at a more abstract level and then consider more concrete architectures using wireless networks, web services and clouds. Ref. [33] describes a pattern language for safe adaptive control. The language includes a type of patterns about cases, where safety is proven by argument. Ref. [34] also used safety case patterns, where each pattern presents a safety argument that can be reused to assert safety in different situations. Ref. [35] describes a case study combining security and safety restrictions and also using arguments. Arguments have the problem that they do not have a clear reference to system units as patterns do; but they could make useful complements to patterns, especially for safety. Ref. [40] presents an approach for the early requirements analyzing and modeling of CPS. Through defining the structure model and the goal model, this approach can help to clarify the users' requirements, the required software domains and physical domains, and what these domains need to do to meet the users' requirements.

Reference architectures using patterns have been very effective in describing complex systems. Ref. [36] presents a security reference architecture based on patterns for Big Data. Following our methodology, a cloud reference architecture is presented in [10, 37] and a security reference architecture for cloud systems in [22].

VI. CONCLUSIONS

Most of the work related to Cyber Physical Systems present simplified models which are not precise. The method presented in this paper of using patterns to build the reference architecture of a CPS provides not only a holistic and unified view of the system to its users, developers and researchers, but it is also fundamental to handle the cross domain complexity of this type of system. The model presented here is just a step for achieving a precise architectural representation, other use cases and components need to be defined as patterns to complete the architecture. In future work, we will continue to build the architecture and include security and threat patterns to build a Security Reference Architecture for CPS.

REFERENCES

[1] P. Avgeriou, "Describing, instantiating and evaluating a reference architecture: A case study", Enterprise Architecture Journal, June 2003.

[2] S. Ambler, "Process Patterns: Building Large-Scale Systems using Object Technology", SIGS Books, Cambridge University Press July 1998.

[3] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, M. Stal. Pattern-Oriented Software Architecture: A System of Patterns, Volume 1, Wiley, 1996.

[4] E. Gamma, R. Helm, R. Johnson, J. Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, Boston, Mass., 1994.

[5] J. Warmer, A. Kleppe. "The object constraint language: getting your models ready for MDA", 2nd edn. Addison-Wesley Longman, Boston. 2013. ISBN 0321179366

[6] H. Garavel, S. Graf Formal methods for safe and secure computer systems. Technical report. BSI Study 875, Federal Office for Information Security, Bonn , 2013

[7] E. B. Fernandez, R. Monge, and R. Carvajal, "A pattern for a secure and safe port loading facility", 10th Latin American Conference on Pattern Languages of Programs - SugarLoafPLoP 2014.

[8] V.M. Romero, and E.B. Fernandez, "A Pattern for Secure Cargo Port Drayage", Procs. 7th Asian Conference on Pattern Languages of Programs, Asian PLoP'18, March 1-2, 2018, Tokyo, Japan. 9 pages.

[9] E.B. Fernandez and N. Delessy, ""Using patterns to understand and compare web services security products and standards", *Proceedings of the Int. Conference on Web Applications and Services (ICIW'06*), Guadeloupe, February 2006. IEEE Comp. Society, 2006.

[10] K. Hashizume, E.B. Fernandez, M. Larrondo-Petrie "Cloud infrastructure pattern", First international symposium on software architecture and patterns. LACCEI, Panama City, Panama, 2012, pp 23–27

[11] Taylor RN, Medvidovic N, Dashofy EM (2009) Software architecture: foundations, theory, and practice. Wiley, London. ISBN 0470167742, 9780470167748

[12] E.B. Fernandez, X. Yuan, (2000) Semantic analysis patterns. In: Laender A, Liddle S, Storey V (eds) Conceptual modeling—ER2000, vol 1920 of lecture notes in computer science. Springer, Berlin, pp 183–195. doi:10.1007/3-540-45393-8_14, ISBN 978-3-540-41072-0

[13] V. Stricker, K. Lauenroth, P. Corte, F. Gittler, SD Panfilis, K. Pohl, "Creating a reference architecture for service-based systems— a pattern-based approach". In: Towards the future internet— emerging trends from European research, 2010, pp 149–160. doi:10.3233/978-1-60750-539-6-149

[14] Muller G, van de Laar P (2009) Researching reference architectures and their relationships with frameworks, methods, techniques, and tools. In: Kalawsky R, O'Brien J, Goonetilleke T, Grocott C (eds) 7th annual conference on systems engineering research (CSER 2009). Research School of Systems Engineering, Loughborough University, Loughborough

[15] Object Management Group (2014) Unified Modeling Language TM (UML) Tech. rep., Object Management Group Inc.

[16] U.S. Department of Transportation, Research and Innovative Technology Administration, Bureau of Transportation Statistics, Freight Transportation: Global Highlights, 2010 (Washington, DC: 2010), accessed May 1, 2013, http://www.rita.dot.gov/bts/sites/rita.dot.gov.bts/files/publications/freight_transportation/pdf/entire.pdf.

[17] Population Reference Bureau. 2011 World Population Data Sheet. (Washington, DC: 2012), accessed May 1, 2013, http://www.prb.org/pdf11/2011population-data-sheet_eng.pdf.

[18] U.S. Department of Transportation, Research and Innovative Technology Administration, Bureau of Transportation Statistics. America's Container Ports: Linking Markets at Home and Abroad. (Washington, DC: 2011), accessed May 1, 2013, http://www.bts.gov/publications/americas_container_ports/2011/pdf/entire.pdf.

[19] E. YorkWallischeck, “ICS Security in Maritime Transportation A White Paper Examining the Security and Resiliency of Critical Transportation Infrastructure” , June 2013, DOT-VNTSC-MARAD-13-01.

[20] M. Bielli, A. Boulmakoul, M.Rida, “Object Oriented Model for Container Terminal Distributed Simulation”, European Journal of Operational Research 175 (2006) 1731-1751.

[21] A. Boulmakoul, M. Rida, “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, December, 18-21, 2002, pp. 266-271.

[22] E.B. Fernandez, R. Monge, and K. Hashizume, “Building a security reference architecture for cloud systems”, Requirements Engineering. Doi: 10.1007/s00766-014-0218-7, June 2016, Volume 21, Issue 2, pp 225-249.

[23]M. Zaffalon, A.E. Rizzoli, L.M. Gambardella, M. Mastrolilli, Resource allocation and scheduling of operations in an intermodal terminal, In: 10th European Simula- tion Symposium and Exhibition, Simulation in Industry, ESS98, Nottingham , UK, October 26-28, 1998, pp. 520-528.

[24] Steenken, D., Voß, S. & Stahlbock, R. OR Spectrum (2004) 26: 3. https://doi.org/10.1007/s00291-003-0157-z

[25] A.A. Cardenas, S. Amin, Z.-S. Lin, Y.-L.Huang, C.H.Huang, and S. Sastry, “Attacks against process control systems: Risk assessment, detection, and response”, Procs. of ASSIACS’11, March 22-24, 2011, Hong Kong, China, 355-366.

[26] A. Miller, “Trends in process control systems security”, IEEE Security and Privacy, Sept/October 2005, 57-60.

[27] H.A. Rahman and K. Beznosov, “SPAPI: A security and protection architecture for physical infrastructures and its deployment strategy using wireless sensor networks”, Proceedings of 10th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2005), Catania, Italy, 2005, 885-892

[28] Y. Tan, S. Goddard, and L.C.Perez, “A prototype architecture for cyber-physical systems”, ACM Sigbed Rev., Vol. 5, No1, 2008, 1-2

[29] Hyun Jung La and Soo Dong Kim, “A service-based approach to designing cyber-physical systems”, Procs. 9th IEEE/ACIS Int. Conf. on Computer and Information Sci., 2010, 895-900.

[30] Microsoft Power and Utilities, Smart Energy Reference Architecture, Oct. 2009, http://www.Microsoft.com/Utilities

[31] P. Verissimo, “CRUTIAL: Towards a reference critical information infrastructure architecture”, European CIIP Newsletter, May/June 2007, vol. 3, No 2, 6-8

[32] E.B. Fernandez and M.M. Larrondo-Petrie, "Designing secure SCADA systems using security patterns", Procs. of the 43rd Hawaii Conf. on Systems Science, Honolulu, Jan.2010, 1-8. March http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5428672

[33] Andre A. Hauge, Ketil Stølen. SACS - A pattern language for safe adaptive control software. Proc. 18th Conference on Pattern Languages of Programs (PLOP'11).

[34] A. Ayoub, B.G. Kim, I. Lee, and O. Sokolsky, “A safety case pattern for model-based development approach”, LNCS Volume 7226, 2012, NASA Formal methods 2012, 141-146. http://shemesh.larc.nasa.gov/nfm2012/talks/NFM2012_141_146.pdf

[35] 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.

[36] J. Moreno, M.A. Serrano, E. Fernandez-Medina, E.B.Fernandez, “Towards a Security Reference Architecture for Big Data”, to be published.

[37] Hashizume K, Fernandez EB, Larrondo-Petrie MM (2012) Cloud service model patterns. In: 19th international conference on pattern languages of programs (PLoP2012), Tucson, AZ

[38] V.M.Romero, E.B.Fernandez, “Towards a Security Reference Architecture for Cyber Physical Systems”, Proceedings of the Fifthteen Latin American and Caribbean Conference for Engineering and Technology, July19-21,2017, Boca Raton, FL-USA

[39] Fortune Magazine, internet reference, http://fortune.com/2018/01/30/port-automation-robots-container-ships/

[40] C. Liu, W. Zhang, H. Zhao, Z. Jin, “Analyzing Early Requirements of Cyber-Physical Systems Through Structure and Goal Modeling”, 20th Asia-Pacific Software Engineering Conference, 2013.

[41] 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, 2011M. King, B. Zhu, and S. Tang, “Optimal path planning,” *Mobile Robots*, vol. 8, no. 2, pp. 520-531, March 2001.

[42] H. Günther, K. Hwan Kim, “Container Terminals and Automated Transport Systems: Logistics Control Issues and Quantitative Decision Support”, Springer Science & Business Media, Dec 6, 2005