Multi-Agent System and Digital Twin Models for Security Study of Cyber-Physical System

Pre-study

October 2022

Student	Zoé Lagache[1]
KTH Examiner	Roberto Guanciale
KTH Supervisor	Musard Balliu
LCIS supervisor	Oum-El-Kheir Aktouf[2]
	Annabelle Mercier[3]

[1] lagache@kth.com

[2] oum-el-kheir.aktouf@lcis.grenoble-inp.fr

[3] annabelle.mercier@lcis.grenoble-inp.fr

Keywords: Multi-Agent System, Digital Twin, Cyber-Physical system, Security

List of acronyms and abbreviations

ANGEL The Automatic Network Guardian for ELectrical systems 16

BDI Belief–Desire–Intention 13

CCLCBM Center for Connected Learning and Computer Based Modeling 13

CIA Confidentiality, Integrity, Availability 6, 9, 10

CPS Cyber-Physical System 1–8, 10, 11, 16, 17

DDoS Distributed Denial of Service 10

DoS Denial of Service 2, 9, 10

DT Digital Twin 2, 3, 11, 14–16

FIPA Foundation for Intelligent, Physical Agent 13, 14

IoT Internet of Things 2

JADE Java Agent DEvelopment 13

MaDKit Multiagent Development Kit 13

MAS Multi-Agent System 2, 3, 11–14, 16

MITM Man-In-The-Middle 8, 10

PADE Python Agent DEvelopment 13

SPADE Smart Python Agent Development Environment 14

SPOF Single Point Of Failure 12

TMS Trust Management System 12

VMNet Virtual Mote Network 16

WSN Wireless Sensor Network 16

Contents

Chapter 1

Introduction

1.1 Background

In a world in an increasing need of control over physical processes, monitoring becomes mandatory. In order to be accurate enough, more an more systems use electronic component for monitoring, or even in order to create a digital interface to manipulate physical processes more easily. This is what we call Cyber-Physical System (CPS). These systems are becoming increasingly common and play a crucial role in many aspects of modern life. As an example, vehicles are now equipped with electronic systems to enhance the safety of their occupants. These systems include sensors for detecting impacts and rollovers, airbags and belt tensioners that deploy in the event of a crash, monitoring of tire pressure, and ACC which automatically adjusts the car's speed to maintain a safe distance from the vehicle in front [1].

1.2 Problem statement

According to the CPS Steering Group [2], CPS are defined as the interaction between physical systems and processes using computations and communication abilities. However, the integration of computing and communication technologies also introduces new security risks such as the potential for unauthorized access, manipulation of system controls, and the disruption of critical infrastructure. These risks can have significant consequences, including loss of life, economic damage, and the compromise of sensitive information [3] [4]. We can imagine, for example, the Cyber-Physical System in an airport monitoring each planes entering or leaving the airport.

If such a system undergoes, for instance, a Denial of Service (DoS) attack, the consequences could be devastating. A DoS attack on such a system could disrupt the communication between the control tower and the planes, leading to confusion and potentially dangerous situations such as collisions on the runway or in the air. It could also cause delays or cancellations of flights, resulting in significant economic losses. Furthermore, the attack could also disrupt other airport operations, such as baggage handling and passenger screening, causing further chaos and disruption. It is important to have robust security measures in place to mitigate these risks and to have contingency plans in place to respond to potential security breaches. Additionally, research and development in the field of CPS security should be ongoing to stay ahead of emerging threats. This is why there is a growing need to study the security of CPS and develop effective strategies for protecting them.

1.3 Purpose

Cyber-Physical Systems can be found in a large range of fields, going from healthcare to the electrical power grid management. Numerous applications exist and could benefit from a stronger degree of security.

Our project, motivated by this need of security, aims at contributing to the field by using innovative approaches such as Multi-Agent System (MAS) and Digital Twin (DT) models. This could help advance the state of the art in Cyber-Physical Systems security and provide new insights and solutions for addressing the challenges described in the previous section 1.2.

By combining the MAS (see 2.2) and the DT (see 2.3) approaches, the project could help at identifying potential security vulnerabilities in cyber-physical systems and develop strategies for addressing them. This could help improve the security of these systems and protect them from cyber threats.

1.4 Goals

In this master thesis, we propose to study how MAS and DT models can help with this aforementioned problem, and how they complement each other. A Multi-Agent System is a system composed of agents collaborating with each other in order to achieve a common goal. The agents are able to communicate with their local neighbours and, most of the time, only have a local view of the system. MAS are now a trend in the Internet of Things (IoT) field thanks to its decentralization aspect. In the other hand Digital Twins are often associated to a way to track and analyse a system in real time, often in order to predict its behaviour. Both models have the potential to be very useful tools for protecting and preventing cyber attacks on CPS.

DT can be used to simulate and analyze the behaviour of the system in various scenarios. This can be very useful in protecting and preventing cyber attacks because it allows for the simulation of different attack scenarios [5] and the evaluation of the system's response to them. This can help identify vulnerabilities in the system and allow for the implementation of countermeasures before an actual attack occurs.

MAS models can be very useful in protecting and preventing cyber attacks on CPS because it allows for the modeling of complex interactions between different system components, as well as the modeling of the behaviour of an attacker [6]. This can help identify potential attack scenarios and the potential impact of an attack on the system. Furthermore, MAS models can also be used to test the effectiveness of different security measures and to optimize the design of the system for better security.

These two concepts complement each other in protecting and preventing cyber attacks on CPSs by providing a dynamic simulation of the system's behaviour in various scenarios from the DT, and providing a more detailed understanding of the complex interactions between different system components from the MAS. Together, they provide a more comprehensive approach to identifying vulnerabilities, testing security measures, and optimizing the design of the system for better security.

Our goals with this project are:

- Developing multi-agent system and digital twin models for studying the security of CPS;
- Identifying potential security vulnerabilities in CPS;
- Developing strategies for addressing these vulnerabilities and improving the security of CPS using the proposed model;
- Evaluating the effectiveness of these strategies through simulation and experimentation.

1.5 Research methodology

In order to establish the state of the art, we are going to use mainly Google Scholar and Scopus to look for papers or articles and Zotero [7] to save them.

First, the results from the state of the art will direct us towards our model creation and several simulation tools. We will establish an attack model from existing CPS vulnerabilities and make a choice among the tools described in sections 2.2.2 and 2.3.2. Then, we will simulate this model with the chosen simulator to check if it works as we want and if we can extract some security claims in relation with the chosen attack model.

From a reproducibility perspective, we will try to provide every sources of paper and tool we use when possible, and will state when it is not possible, which is unlikely in our case. Indeed, most of the tool we will use should be free. However, some papers may require institutional access or a fee to access the full document. The DOI will be given as often as possible to facilitate the access to the different sources.

1.6 Delimitations

Definitions of both Multi-Agent systems and Digital Twins do not have a consensus. We will have to decide of our own. This could limit the set of possible interpretations of these two concepts.

In addition to this, this work will only focus on one specific vulnerability model and will not be generalized to more than the chosen model.

As a limitation, we can also take the duration of the project into consideration. Indeed this is a 5-month project which does not leave much room to study all the possible aspect of the subject.

Finally, the delimitations of this study include the use of simulation to evaluate the proposed solution. Due to the lack of time and access to physical CPS, it was not possible to test the proposed solution on actual CPS. Therefore, simulation is utilized as a mean of evaluating the potential performance and effectiveness of the proposed solution. The results obtained from simulation could be slightly different from the results obtained from testing on actual CPS. The simulation results will only be considered as an approximation of the actual performance, and the results of the simulation should be considered in that context.

Most of these limitations can be the subject of future works in the field.

1.7 Ethics and sustainability

In computer science, simulation is often something that is energy consuming. We will try to tend to optimized solutions but this is not in the scope of this project. We will not focus on a sustainable solution for security of CPS analysis.

Nevertheless, using Digital Twins can also have a positive impact on ethics and sustainability. Creating virtual prototypes instead of physical ones, means less energy and materials consumption, and less waste. Additionally, by identifying and addressing issues early in the design process, it is possible to create products that are more energy-efficient and have a smaller environmental footprint.

Moreover, as an ethical and sustainability perspective, we have to say that this work will discuss attacks on CPS and may present examples that could be used outside of this work context. However, since the project is primarily concerned with theoretical concepts, this is unlikely.

Chapter 2

Background

In this chapter, we are going to introduce the main areas of the subject by providing their basic background. The related work will also be discussed.

2.1 Cyber-physical Systems

In this section, we are going to explain how CPS are defined in this thesis, and focus on what kind of vulnerability this sort of system is facing. More precisely, we will indicate which of the Confidentiality, Integrity, Availability (CIA) properties is affected for each vulnerability.

The CIA model is a framework for evaluating the security of a system. These three elements are considered the core components of information security and are used to assess the overall security of a system.

Confidentiality refers to the protection of sensitive information from unauthorized access or disclosure. It ensures that only authorized individuals can access sensitive information, and that the information is protected from unauthorized disclosure.

Integrity refers to the protection of information from unauthorized modification or destruction. It ensures that the information is accurate and complete and that it has not been tampered with.

Availability refers to the ability of authorized individuals to access the information when they need it. It ensures that the information is available to those who need it, and that it is protected from unauthorized denial of service attacks.

The main goal is to assess the overall security of the system by evaluating the Confidentiality, Integrity and Availability of the system's data and control. By doing so, it will allow us to identify potential vulnerabilities and to develop a strategy for protecting the CPS from cyber attacks.

2.1.1 What are Cyber-physical Systems

As indicated in 1.2, CPS are the combination of the physical world and the cyber space interacting with one another through the use of sensors, actuators, communication, and interfaces. The cooperation between the physical and cyber systems is typically achieved through the use of sensors to monitor the physical system and actuators to control it, as seen in the work of Chen et al. [8], Lei et al. [9] and Tsang et al. [10]. Monitoring refers to the process of gathering data and information about a system, process or environment, by using sensors and virtual models. This allows to identify vulnerabilities and to perform vulnerability study, and it can be applied in various fields and contexts.

An example of a CPS is a smart grid system, which uses sensors to monitor the energy consumption in buildings and homes, actuators to control the flow of electricity, and interfaces to communicate with utility companies and customers. This system uses data from sensors to optimize the distribution of electricity, reduce costs and prevent outages. The smart grid system is a CPS because it enables the integration of the physical power grid with the cyber-space, allowing real-time monitoring and control of the power grid.

2.1.2 Cyber-physical System vulnerabilities

The figure 2.1 represents an abstract view of a CPS. We simplified it as two parts: the physical process(es) part and the cyber system part. Both of these parts are interacting with each other through sensors and actuators, which compose the interface between both worlds. The cyber system is composed of computing devices receiving data from the sensors, processing it, and sending the result to the actuators. The green arrows are indicating the monitoring interaction, while the red ones represent the communication within the cyber system. The numbers point to the parts of the CPS that are subject to vulnerabilities.

By analysing and synthetizing the classifications done by Wang et al. [3], Wazid et al. [11] and Singh et al. [4], we obtained the following vulnerabilities mapping:

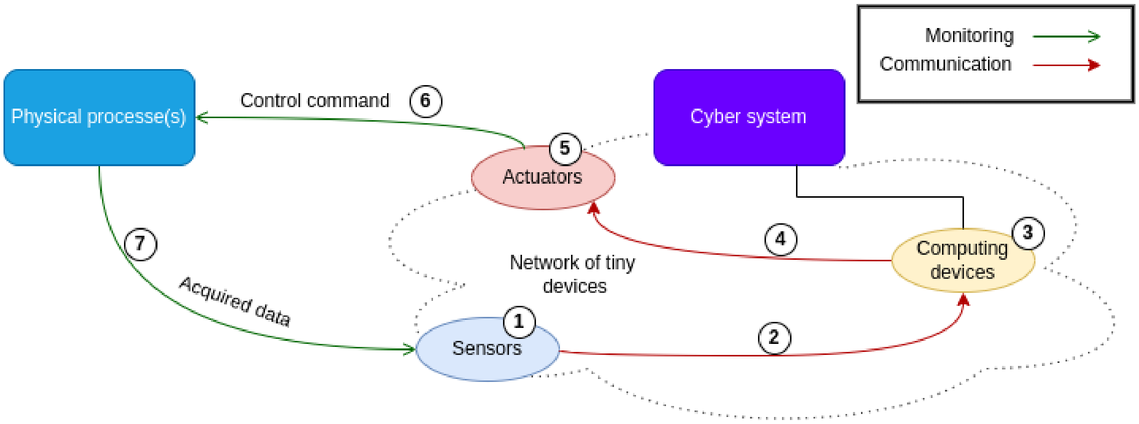

Figure 2.1: CPS main vulnerabilities

2.1.2.1 Communication attacks

These attacks have the potential to be operated on all communication links, i. e. at points 2, 4, 6 and 7 in figure 2.1. Communication attacks can include:

- *Eavesdropping:* a passive attack where the attacker is listening to a communication between the two or more nodes of the system. This harms the *confidentiality* of the communication.
- *Man-In-The-Middle (MITM):* the base concept is the same as the eavesdropping attack except that the attacker is able to intercept the communication packets and thus to modify them. This attack can impact the *confidentiality* and the *integrity* of the communication.

2.1.2.2 Network or routing attacks

We put in this category the attacks that are the result of a changing of behaviour from one element of the system that can impact changes in the rest of the network. All parts of the system from 1 to 5 in figure 2.1 could be impacted by such attacks.

- *Blackhole attack:* the attacker is able to corrupt one or several nodes in the networked system so they advertise their neighboring nodes. This way, they become more attractive in the path-finding algorithm of the nodes. Nevertheless, once the blackhole nodes receive a packet, they drop it. This attack may be called Sinkhole attack in literature while designing Blackhole attacks as several nodes dropping packets, without making themselves more attractive.

This attack may perturb the *availability* of some part of the networked system.

- *Greyhole attack:* this attack is the same as the Blackhole attack but not all the packets are dropped. A filter is used to select which packets to drop. As for the Blackhole attack, the availability is impacted.
- *Wormhole attack:* at least two nodes are required to carry out this attack. These nodes are normally not able to communicate with each other but the attacker will upgrade them so they can. This can be done by several ways, for example by simply adding the route in their routing table to increase their emission and reception ranges, in the case they are too far from each other. This attack does not directly affect any of the CIA parameters by itself but it can help multiple follow-up attacks which could damage one or several of them.

2.1.2.3 Physical attacks

Physical attacks can be done on the devices that are the closest to the real world, thus on the actuators and on the sensors (1 and 5 in figure 2.1). They can be:

- *Side channel attack:* the attacker analyzes physical parameters that may vary depending on algorithms implementations and their inputs and is then able to find secret information. This impacts the *confidentiality* of the system.
- *Fault injection:* the attacker injects a physical fault in the device to change its behaviour to a malicious one. This can be done by injecting quick voltage faults or electromagnetic faults, etc. Such attacks have the potential to extract secret information or bypass system security, for example. As for the Side Channel Attack, *confidentiality* is compromised here.
- *Jamming attack:* the attacker jams the communication hence no packets can be sent and received between the sensor and the signal transforming device. Jamming attacks impact the *availability* of some parts of the system.

2.1.2.4 Miscellaneous

Here are the attacks that have their own location on the map and are not part of the previous classes, or can potentially be part of several of the previous classes:

- *Malware spreading attack:* this is an application attack. The attacker spreads a piece of malicious code into one or more devices of the system. What the code does vary a lot from revealing secret keys to making a service unavailable, and thus create a DoS, for example. This kind of attack is mostly observable

Table 2.1: Summary of CPS vulnerabilities

Attack Class	Attack	Vulnerable Surface	CIA Involved
Communication	Eavesdropping	2, 4, 6 & 7	Confidentiality
	MITM		Confidentiality, Integrity
Network/Routing	Blackhole	1–5	Availability
	Greyhole		Availability
	Wormhole		All
Physical	Side channel	1,5	Confidentiality
	Fault injection		Confidentiality
	Jamming		Availability
Miscellaneous	Malware	3	All
	DoS	2 or 3	Availability

on the computing devices (3 in figure 2.1). As a result of its diversity, any of the three parameters of the CIA model can be harmed.

- *DoS attack:* the attacker puts down a device so it cannot work anymore. This device is often chosen strategically. Ways to achieve this attack are numerous. Depending on how it is carried out, a DoS could be part of each of the above part. For example, it can be by sending an extremely high number of messages so a server receiving them is flooded and cannot work properly. If the DoS is carried out by several devices attacking one target, we call that a Distributed Denial of Service (DDoS) attack. Depending on whether one considers the vulnerable part to be the one that is exploitable or the one that suffers the damage, DoS is located either on point 2 or on point 3 in figure 2.1. It affects the *availability* of the system.

2.1.2.5 Summary

The table 2.1 summarize the vulnerabilities discussed above.

2.1.3 Cyber-physical Systems simulation

In computer science, simulation is the act of digitally representing a phenomenon using a model. It is often used when it is not possible or too expensive to produce the phenomenon in the real world, or when one wants to represent abstract concepts. Simulation is also a way to test our work without having to impact the real world. For example, critical systems failures can put lives in danger.

The value of CPS simulation resides in that CPSs already have information about the real world in a digital environment. They have been studied since several years and are still confronted to simulation questions as shown by Thule et al. [12], a 2019 paper presenting a framework for CPS simulation.

It is also a useful tool for prevention or detection in security. Suppose we want to secure a large-scale industrial system that includes multiple interconnected components, including sensors, controllers, and actuators. We can use a MAS to model the interactions between these components and simulate different attack scenarios to identify vulnerabilities and assess the impact of potential attacks. The MAS can also incorporate autonomous decision-making capabilities to allow for quick responses to attacks and to optimize the system's security.

In addition, we can use DT to create a virtual model of the physical system that captures real-time information about its behavior and performance. This virtual model can be used to simulate different scenarios and test the effectiveness of different security measures before they are implemented in the physical system. The DT can also be updated in real-time based on data from the physical system to improve the accuracy of the model and the effectiveness of the security measures.

The distributed nature of the MAS allows for scalability, as the system can be easily expanded to include additional components or subsystems to the industrial chain. This makes it possible to secure even very large and complex systems. The real-time information gathering capability of the DT enables us to detect and respond to attacks more quickly and accurately, as we can monitor the system in real-time and detect anomalies or suspicious behavior.

2.2 Multi-Agent Systems

A MAS is a system with a collection of, at least, two agents, cooperating with each other [13, 14]. These agents are entities containing an algorithm defining their behaviour and making them autonomous in their decision making to achieve a common goal. If we add capabilities constraints like computing or memory limitations, then we are talking about embedded agents. As specified in section 1.4, MASs have been discussed a lot in the computer science field. It is a model that can be used at different level and in different fields. For example, a MAS could be a software working on multiple autonomous threads, representing the agents, on a computer. In this case, the separation of the agents is at an application level. Another use case could be a fleet of drones interacting with each other with one drone corresponding to one agent. This diversity of application areas comes from the high customisability and scalability of MAS.

2.2.1 Multi-Agent System and security

MASs raise specific challenges for security or safety concerns, due to their autonomy and decentralization that makes them complex, or even impossible, to monitor. However, the approach can solve problems coming from centralization like Single Point Of Failure (SPOF). For example, Cai et al. [15] propose an Intrusion Detection System based on an Agents approach where different kind of agents are responsible for monitoring different kinds of resources in the system.

Moreover, a topic that is often associated to MAS is Trust Management System (TMS). The TMS defines how an agent can decide to trust another agent or not, depending on specified conditions. It also defines the behaviour of the agent in both cases.

Another interesting feature that can be done is a way to protect the communications between the agents. This can be done through message signature and encryption.

Building a MAS integrating a TMS with secured communications makes a system easier to protect in the way that securing one agent is simpler than to monitor an entire system.

2.2.2 Multi-Agent System simulation tools

This section aims at comparing several tools for simulating MAS. To gather the tools we are going to introduce, we based our research on previous works done at the LCIS laboratory [16, 17] in addition to some personal ones as specified in 1.5. Among all these simulators, our first criteria of selection was on the version number. We did not select any tools under version 1.x, in other words that are still at a beta step of development. The second one was on the running platform. We must be able to run the simulator on Debian 11 (Bullseye).

2.2.2.1 First selection

GAMA [18] is an open source tool for agent based simulation allowing to incorporate agent locations thanks to the GAML language. It was created in 2019.

JACK Intelligent Agents [19] is a multi-agent system development framework written in Java and developped in 1999. It has the particularity to use the Belief–Desire–Intention (BDI) model.

Java Agent DEvelopment (JADE) [20] is also a Java framework for agent-based development but with the difference that JADE is compliant to the Foundation for Intelligent, Physical Agent (FIPA). JADE is open source and was developed in 2001.

SARL [21] is an open source general-purpose agent-oriented programming language that comes with the Janus execution platform[1]. SARL was developed in 2014.

Multiagent Development Kit (MaDKit) [22] is an open source library, written in 2000 in Java, for simulating MAS.

Mesa [23] is an open-source agent-based modeling framework. It was written in Python and first released in 2021. Mesa comes with a web-browser visualization and aims at offering an easy and customisable way to create agent-based models.

NetLogo [24] is a simulator with the goal to simulate natural or social phenomena. It is continuously improved by the Center for Connected Learning and Computer Based Modeling (CCLCBM). NetLogo is open source and comes with its own language and a web-browser interface.

Python Agent DEvelopment (PADE) [25] is another MAS framework in Python that is FIPA compliant. It provides an open source tool for developing, executing and managing a MAS. PADE was created in 2019 for application on power grids.

[1] http://www.sarl.io/runtime/janus/

Smart Python Agent Development Environment (SPADE) [26] is a development platform in python using the Jabber communication framework for agents communications. SPADE is also open source and FIPA compliant. It was developed in 2006.

Table 2.2: Summary of the MAS simulation tools

Tool name	Year	Language	Easy to learn	Close support	Open-source
GAMA	2019	GAML	No	No	Yes
JACK	1999	Java	Yes	No	No
JADE	2001	Java	Yes	No	Yes
SARL	2014	SARL	No	No	Yes
MaDKit	2000	Java	Yes	No	Yes
Mesa	2021	Python	Yes	Yes	Yes
NetLogo	1999	NetLogo	Yes	Yes	Yes
PADE	2019	Python	Yes	No	Yes
SPADE	2006	Python	No	No	Yes

2.2.2.2 The chosen one

From the pre-selection we made, we can now summarize the tools found while highlighting our selection criteria: open-sourceness, short learning time, creation year, language of the application programming interface and the availability of technical support at the laboratory. The summary is done in table 2.2.

We can say that Mesa stands out. It is one of the most recent works and aims at being easy to use which is needed in our project since we don't have much time to spend to learn new tools. Python is also a language with a big and active community which makes support on this language more accessible. On top of that, the tool was used in previous and current projects done at the laboratory [16] thus we can get help on how to handle Mesa and the possible difficulties we could encounter.

2.3 Digital Twins

A Digital Twin (DT) is a virtual representation of a physical object or a system. It is a digital replica of a physical entity, which can be used to model and simulate the real-world characteristics and behaviours and to bring feedback to it. DT are typically created by using data from sensors and are updated in real-time as new data becomes available. They can be used for a wide range of applications, including design and engineering, manufacturing and production, operations and maintenance, and more. One of the most common uses is the predictive maintenance.

They add a new dimension to product development by allowing engineers and designers to test and analyze the performance of a product before it is built. This can be a major advantage in product development as it allows for a more efficient and cost-effective process, as well as reducing the need for physical prototypes.

One of the key benefits of using Digital Twins is the ability to identify and address potential issues early in the design process. By simulating the product's behaviour and performance in different scenarios, it is possible to identify and address any potential issues before the product is built. This can lead to a more reliable and high-quality product.

The concept is arduous to frame because most of the subjects working with real-time simulation can, directly or not, be linked to DT. For example, the work done by Mrissa et al. [27] presents a concept of avatars as a runtime environment and a virtual abstraction of physical objects.

2.3.1 Digital Twins for security

Since DTs tend to have the same behaviour as the physical entity they are a twin of, they are an interesting tool to monitor complex systems. For example, they can be used to predict malfunctioning or malicious behaviour, like proposed by the Automatic Network Guardian for Electrical (ANGEL) [28].

They can also be used to simulate and then test these complex systems. Thus we can identify and address potential vulnerabilities existing on the system. These vulnerabilities can then be prevented or a recovery system can be set up, mitigating the effects of cyber attacks.

Additionally, DTs can be used to evaluate how effective are different security measures and strategies, by simulating how a system would behave under different scenarios and with different security measures in place. This can help organizations to choose the most effective ways to protect their systems from cyber attacks.

2.3.2 Digital Twins simulation tools

In this section, we are going to present several tools for simulating a DT. As well as for the MAS simulation tools, we only picked tools that can run on Debian 11 (Bullseye) and with a version number greater than or equal to 1.

Unfortunately, while gathering these tools, we quickly discovered that as well as the subject, the tools are complicated to gather. A lot of different kind of tools can be considered as a DT simulator, and doing a whole state of the art on it could be the subject of a Master Thesis on its own. We also had difficulties to find tools according to the criteria given in the paragraph above. Indeed, we found numerous DT simulator that are not free or that only run on Windows for example. Thus the tool list we are going to present is far from a comprehensive one.

Simulation tools for specific topics can sometimes meet with the Digital Twins definition. In our case, the topic can be anything that can be called a CPS.

Veins (Vehicles in network simulation) [29], the open source vehicular network simulation framework.

Virtual Mote Network (VMNet) [30] is a Wireless Sensor Network emulator. It aims at being the more realistic possible for performance evaluation in Wireless Sensor Network (WSN) applications.

The Automatic Network Guardian for ELectrical systems (ANGEL) [31]. The ANGEL Digital Twin is a novel approach for improving the security of CPS. It is a framework that aims at adapting the Digital Twin to the application of microgrid security.

2.4 Conclusion

Overall, MASs and DTs can be used in combination to provide a comprehensive approach to CPSs security, by providing the ability to monitor the system in real-time, identify vulnerabilities, and respond quickly to potential security threats. This makes it possible to protect CPSs from cyber attacks, and to ensure the Confidentiality, Integrity, and Availability of the system's data and control.

In the Master Thesis, we are going to present our contribution based on the information gathered in this Pre-Study. More precisely, we are going to give specifications on our system and attack model and explain how the chosen tools will be used to answer these specifications. The full process of thinking from the model creation to the development of a simulator detecting few CPS routing attacks. We will try to write down all the choices we made for this contribution and explain them.

Bibliography

[1] N. Navet, Y. Song, F. Simonot-Lion, and C. Wilwert, “Trends in automotive communication systems,” *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1204–1223, 2005.

[2] CPS Steering Group, “Cyber-Physical Systems Executive Summary,” Chicago, IL, USA., Mar. 2008.

[3] E. K. Wang, Y. Ye, X. Xu, S. M. Yiu, L. C. K. Hui, and K. P. Chow, “Security Issues and Challenges for Cyber Physical System,” in *2010 IEEE/ACM Int’l Conference on Green Computing and Communications & Int’l Conference on Cyber, Physical and Social Computing*, Dec. 2010, pp. 733–738.

[4] S. Singh, N. Yadav, and P. K. Chuarasia, “A Review on Cyber Physical System Attacks: Issues and Challenges,” in *2020 International Conference on Communication and Signal Processing (ICCSP)*, Jul. 2020, pp. 1133–1138.

[5] A. Saad, S. Faddel, T. Youssef, and O. A. Mohammed, “On the Implementation of IoT-Based Digital Twin for Networked Microgrids Resiliency Against Cyber Attacks,” *IEEE Transactions on Smart Grid*, vol. 11, no. 6, pp. 5138–5150, Nov. 2020.

[6] V. Gorodetsky, I. Kotenko, and O. Karsaev, “Multi-agent technologies for computer network security: Attack simulation, intrusion detection and intrusion detection learning,” *Comput. Syst. Sci. Eng.*, vol. 18, pp. 191–200, 07 2003.

[7] “Zotero,” https://www.zotero.org/, accessed: 2023-01-22.

[8] L. Chen, F. Hu, S. Wang, and J. Chen, “Cyber-physical system fusion modeling and robustness evaluation,” *Electric Power Systems Research*, vol. 213, p. 108654, Dec. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378779622007246

[9] C.-U. Lei, K. Wan, and K. L. Man, "Developing a Smart Learning Environment in Universities Via Cyber-Physical Systems," *Procedia Computer Science*, vol. 17, pp. 583–585, Jan. 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050913002081

[10] Y. P. Tsang, T. Yang, Z. S. Chen, C. H. Wu, and K. H. Tan, "How is extended reality bridging human and cyber-physical systems in the IoT-empowered logistics and supply chain management?" *Internet of Things*, vol. 20, p. 100623, Nov. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2542660522001044

[11] M. Wazid, A. K. Das, S. Kumari, and M. K. Khan, "Design of sinkhole node detection mechanism for hierarchical wireless sensor networks," *Security and Communication Networks*, vol. 9, no. 17, pp. 4596–4614, 2016. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/sec.1652

[12] C. Thule, K. Lausdahl, C. Gomes, G. Meisl, and P. G. Larsen, "Maestro: The into-cps co-simulation framework," *Simulation Modelling Practice and Theory*, vol. 92, pp. 45–61, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1569190X1830193X

[13] M. Wooldridge, *An Introduction to MultiAgent Systems*. Wiley, 2009. [Online]. Available: https://books.google.fr/books?id=X3ZQ7yeDn2IC

[14] J. Ferber, O. Gutknecht, and F. Michel, "From agents to organizations: An organizational view of multi-agent systems," in *Agent-Oriented Software Engineering IV*, P. Giorgini, J. P. Müller, and J. Odell, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 214–230.

[15] Y. J. Cai, X. Y. Cheng, and Y. Pan, "Solutions of Single Point of Failure in Intrusion Detection System," *Applied Mechanics and Materials*, vol. 128-129, pp. 285–288, 2012. [Online]. Available: https://www.scientific.net/AMM.128-129.285

[16] Bonnet Adrian, "Sécurisation des communications dans un système multi-agents embarqués," LCIS, Tech. Rep., 2021.

[17] Derdour Najoua, "Conception et réalisation d'un protocole de gestion de confiance dans les systèmes multi agents," LCIS, Tech. Rep., 2020.

[18] P. Taillandier, B. Gaudou, A. Grignard, Q.-N. Huynh, N. Marilleau, P. Caillou, D. Philippon, and A. Drogoul, "Building, composing and experimenting complex spatial models with the GAMA platform," *GeoInformatica*,

vol. 23, no. 2, pp. 299–322, Apr. 2019. [Online]. Available: https://doi.org/10.1007/s10707-018-00339-6

[19] P. Busetta, R. Rönnquist, A. Hodgson, and A. Lucas, "JACK Intelligent Agents - components for Intelligent Agents in Java," *www.agentlink.org*, Jan. 1999. [Online]. Available: http://www.agent-software.com.au/media/research/jack/busetta99jack.pdf

[20] F. Bellifemine, A. Poggi, and G. Rimassa, "Developing multi-agent systems with a fipa-compliant agent framework," *Software: Practice and Experience*, vol. 31, no. 2, pp. 103–128, 2001. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/1097-024X%28200102%2931%3A2%3C103%3A%3AAID-SPE358%3E3.0.CO%3B2-O

[21] S. Rodriguez, N. Gaud, and S. Galland, "SARL: a general-purpose agent-oriented programming language," in *the 2014 IEEE/WIC/ACM International Conference on Intelligent Agent Technology.* Warsaw, Poland: IEEE Computer Society Press, 2014.

[22] O. Gutknecht and J. Ferber, "MadKit: a generic multi-agent platform," in *Proceedings of the fourth international conference on Autonomous agents*, ser. AGENTS '00. New York, NY, USA: Association for Computing Machinery, 2000, pp. 78–79. [Online]. Available: https://doi.org/10.1145/336595.337048

[23] J. Kazil, D. Masad, and A. Crooks, "Utilizing python for agent-based modeling: The mesa framework," in *Social, Cultural, and Behavioral Modeling*, R. Thomson, H. Bisgin, C. Dancy, A. Hyder, and M. Hussain, Eds. Cham: Springer International Publishing, 2020, pp. 308–317.

[24] U. Wilensky, "Netlogo," 1999. [Online]. Available: http://ccl.northwestern.edu/netlogo/

[25] L. S. Melo, R. F. Sampaio, R. P. S. Leão, G. C. Barroso, and J. R. Bezerra, "Python-based multi-agent platform for application on power grids," *International Transactions on Electrical Energy Systems*, vol. 29, no. 6, p. e12012, 2019, e12012 ITEES-18-0867.R2. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/2050-7038.12012

[26] M. E. Gregori, J. P. Cámara, and G. A. Bada, "A jabber-based multi-agent system platform," in *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, ser. AAMAS '06. New York,

NY, USA: Association for Computing Machinery, 2006, p. 1282–1284. [Online]. Available: https://doi.org/10.1145/1160633.1160866

[27] M. Mrissa, L. Médini, J.-P. Jamont, N. Le Sommer, and J. Laplace, "An Avatar Architecture for the Web of Things," *IEEE Internet Computing*, vol. 19, no. 2, pp. 30–38, Mar. 2015, conference Name: IEEE Internet Computing.

[28] W. Danilczyk, Y. Sun, and H. He, "Angel: An intelligent digital twin framework for microgrid security," in *2019 North American Power Symposium (NAPS)*, 2019, pp. 1–6.

[29] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing (TMC)*, vol. 10, no. 1, pp. 3–15, January 2011.

[30] H. Wu, Q. Luo, P. Zheng, and L. M. Ni, "VMNet: Realistic Emulation of Wireless Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 2, pp. 277–288, Feb. 2007.

[31] W. Danilczyk, Y. Sun, and H. He, "Angel: An intelligent digital twin framework for microgrid security," in *2019 North American Power Symposium (NAPS)*, 2019, pp. 1–6.