Mazandaran University of Science and Technology

Available online at www.iiec2017.com

13th International Conference on Industrial Engineering

(IIEC 2017)

Iran Institute of Industrial Engineering

An approach to develop an intelligent distributed Dependability and Security supervision and control for industry 4.0 systems

Hermann Kühnle[a] and Hessamedin Bayanifar[b]

[a] Institut für Arbeitswissenschaft, Fabrikautomatisierung und Fabrikbetrieb (IAF)
Otto-von-Guericke-University of Magdeburg, Germany
Tel: +49 391 671 85 17, E-mail: hermann.kuehnle@ovgu.de

[b] Institut für Arbeitswissenschaft, Fabrikautomatisierung und Fabrikbetrieb (IAF)
Otto-von-Guericke-University of Magdeburg, Germany
Tel: +49 391 675 10 49, E-mail: hessamedin.bayanifar@ovgu.de

Abstract

Despite all its potentials, new industrial revolution enabled by cyber-physical systems (CPS), on its way to be fully appreciated still has major concerns and obstacles with regards to dependability and security. This study targets these concerns by proposing a generic model for intelligent distributed dependability and security supervision and control mechanism, that enables components to autonomously meet their own security and dependability objectives, through real-time distributed supervision and control. In addition, a multi-agent system (MAS) based implementation approach is proposed to enable full exploitation of the model's capabilities.

Keywords:
Industry 4.0, Cyber-Physical Production Systems, Dependability and Security, Multi-Agent Systems

Introduction

Smart distributed manufacturing systems, consist of a large number of widely dispersed loosely-coupled yet collaborating heterogeneous components, that are vastly connected to and communicating with cyber space. To enhance their capabilities, these systems try to exploit smart properties through enhancing their own intelligence and processing power, or via accessing the internet and its glass options to enhance these properties. On the one hand, using these properties and enhancing capabilities can afford manufacturing enterprises a plethora of opportunities and strategic advantages, on the other hand however, such vast dispersity and exposure to cyber space, as well as versatility of processes and systems' structures, raise major vulnerabilities as dependability and data security issues that may diminish the tendency to rely on such enormous capabilities. Hence, to harness all the capabilities, not just through implementation, but also in real-time dependability and security must be assured. Otherwise, due to some security failures and/or partial breakdowns in system, the enterprise may undergo heavy and disastrous losses. Nevertheless, there is still a need for a generic reliable model, and an adaptable, learning tool to analyse and steadily improve all conditions that accommodate satisfactorily the dependability and security needs, would be an important contribution to make smart manufacturing units' application more attractive.

Smart Distributed Manufacturing systems, enabled by Cyber-Physical Systems, have major structural similarities, as they typically have three main layers: physical layer, cyber layer, and data communication and integration layer. Each of these layers has its own concerns with regards to dependability and security. Accordingly, many studies tried to point out these issues or suggest countermeasures for them [1-5]. Considering these studies and a recommendation released by federal office for information security, on industrial control system security [6], some of the main issues are namely: Distributed Denial of Service (DDoS) attack, social engineering and phishing, control system compromising (masquerading, repudiation, manipulation, etc.), Man-In-The-Middle attack, malware infection, intrusion, etc., and stability and controllability issues such as loss of connectivity, observability, breakdown or failure, Quality of Service (QoS), to be seen among the major possible risks. In [7], the author tries to design and implement a robust cyber physical system focusing on security, stability, and systematicness, and [8] attempts to model ontology-based dependability in CPSs using FMEA techniques. Authors in [9] used systems' context awareness to increase security in information access, by asking the questions *who* wants the information, *how*, *what* information, from *where*, and *when* and developing an information system through which accesses are verified. This study on the other hand, tries to propose a generic model, and an agent-based implementation structure to develop an intelligent and autonomous distributed dependability and security supervision and control that is broken down throughout the system, and is to be carried out by smart components themselves, in cyber physical production systems (CPPS).

To this aim, it tries to adopt and harness smart manufacturing systems' capabilities and properties that are elaborated in [10], (i.e. interoperability, autonomy, scalability, modularity, heterogeneity, reconfigurability, and context-awareness), for maximising its performance and versatility. The rest of the paper will be outlined as the introduction of the model and its relationship with smart systems' properties, then presenting multi-agent systems as a toolset to deliver the model, and some examples. And the paper ends with conclusions and related future works.

Methods

Dependability and security model and architecture

To assure maximum dependabilities throughout an enterprise, the adopted approach must be able to deal with all incorporated components, information flows among them as well as the cyber areas, networks, databases and servers. To this goal, a distributed Dependability and Security Model (Fig. 1) is introduced for covering the entire system, every units and components down to all levels of detail (LoD). It aims at guaranteeing smooth and resilient performance by putting its main focus on security and stability. The model includes a core model, a control loop, and a connection to the virtual world.

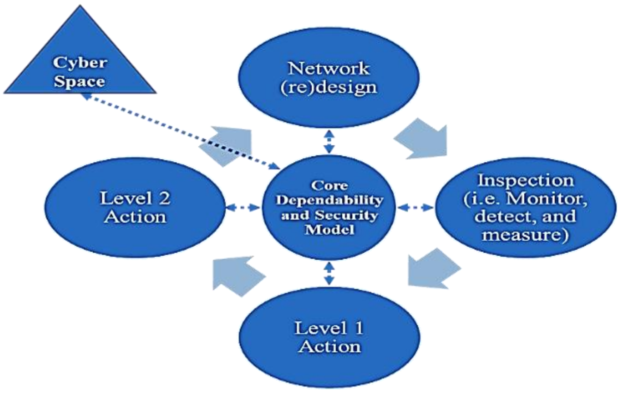
Cyber Space
Network (re)design
Level 2 Action
Core Dependability and Security Model
Inspection (i.e. Monitor, detect, and measure)
Level 1 Action

Figure 1. Smart Dependability and Security architecture

The Model Suggest Autonomous Dependability and Security control for Cyber-Physical Systems/ Cyber-Physical Production Systems (CPS/CPPS). That means each component or element that carry the model have the ability of autonomously carrying on dependability and security analysis and measures when needed by having self-controllability through self-monitoring, self-reconfigurability, along with self- and context-awareness in collaboration with other components. This model aims at improving the dependability and security of the system and all the components in a distributed and semi-/fully autonomous way. The model consists of a control loop, a core model, and connection to virtual world. The core model is comprised of two parts (Object Description, and Risk Model), and the control loop has two main parts: inspection (i.e. Monitoring, Detecting, and identifying and measuring), and Reaction (i.e. giving alarms, taking action, and doing the reconfiguration afterwards). In overall it is to make sure that the Dependability and Security objectives and challenges defined, are met. Some of these objectives could be as follow:

Integrity: which simply means to assure secure transmission of data between elements in the system. In other word, no unauthorized entity should be able to make changes or alterations to the data being sent or received by components. Given the fact that data integration and transmission is the backbone of smart DMs and CPPSs, data integrity comes high on the agenda in making a system dependable. If adversaries by any mean can have access to the data and be able to change them, the system may face heavy consequences.

Confidentiality: suggests that data must not be observed by unauthorized entities. As one of main components of the security triad (CIA), Confidentiality is necessary to avoid adversaries and wrong individuals from eavesdropping sensitive information, or data leakage.

Quality of Service is also playing a significant role in having a dependable system by guaranteeing timely and accurate delivery of data to where it is aimed. That means the right data, will go to the right place at the right time. ITU telecommunication standard "Y.2221" [11], recommends that based on service requirements, quality of services and data prioritization can defer. That means information of higher importance should have priority over less important ones so that to make sure that transmission of essential data will in any case not be at risk.

Stability: is the ability of the system to run flawless when a part or parts of it is compromised. That means if due to any reason a failure (whether intentionally or unintentionally) occurred, the system should not break down.

The **core model** consists of two main parts: *object description*, and *risk model*, where the former focuses more on objects' context and self-awareness, and imports data about object's environment, collaborations, functions and modules, objectives, application and task description, etc., and the latter covers accordingly all Dependability and Security parameters, vulnerabilities and risks, and the ways of measuring and dealing with them. The core model, in other words, feeds the improvement process.

Within the Object Description section, the model provides for overall objectives of the component, the tasks to be performed, and the operations involved. The interacting modules, its structure, its environmental parameters and its position, the components in the group or in other layers it is collaborating with, and other required data furthermore contribute to developing an accurate risk model. The relevant data can be imported from the cloud or sensed as a part of the object's self-/context-awareness. The risk model, in collaboration with the descriptions provided, together with the dependability and security objectives, deals with vulnerabilities and risks that the object is susceptible to. It also contains a feature for assessing risks and threats and their possible effects on the objects or the system in total (e.g. Failure Mode, Effect and Criticality Analysis (FMECA)/Fault Tree Analysis (FTA). The model is to be designed in a modular way, so its parts can be imported or used in other similar or related objects. It is self-optimizing through sharing knowledge with all other smart units, and updating its own structure and database through continuous feedbacks (control loops).

Table 1. Sections of the core model

Objects Description	Risk Model questions
▪ Objectives ▪ Task description ▪ Structure and modules ▪ Environment and position ▪ Collaborations ▪ …	▪ What are the Dependability and security objectives? ▪ What are the vulnerabilities and occurrence probability? ▪ How can they be detected? ▪ What are the effects and their severity? ▪ Who must be alarmed at occurrence ▪ How can they be terminated/ prevented in the future?

The Control Loop invokes the process of Inspection (i.e. *Monitoring*, *Detecting*, and Identifying and *Measuring*), and Reaction (i.e. giving *Alarms*, taking *Action*, and doing the *Reconfiguration* afterwards) in real-time. All steps can be carried out fully- or semi-autonomously by smart objects through this attached core model. As shown in the fig. 1, all the steps are in communication with the core model, which is located in the cyber space and is in collaboration with all other models. This gives the components all abilities to collaborate with the common objective of raising and maintaining the dependability and security of the total system. The following discusses the steps further:

Monitoring: The first step in intelligent dependability and control process is monitoring, which is being done in a semi/fully autonomous way, through the resources and the model the components are equipped with. Smart autonomous components constantly and in real-time are monitoring the condition to make sure that everything matches the approved given and known parameters. These approved parameters are defined and being updated based on the context in which the component is performing. Intelligent components update their knowledge of the context in real-time. Also through their model, they are aware of the tasks they should fulfil and the expected throughputs and the demanded specifications. They are aware of their collaborations and the environment in which they are performing. That means they know other components they are working with, data flow parameters, righteous entities to access data and their parameters, the ambient properties (heat, vibration, moisture etc.), and so forth. With that being said, components have the capability of knowing what is approved and accordingly look for anomalies, that can occur in every layer of the system, i.e. the physical layer, cyber layer, or integration and data communication layer. Given the vast heterogeneity of system's components (real and virtual objects, communication and data transition mechanism, type of service, level of details, etc.), models for monitoring can differ significantly. A simple example can be that a sink sensor node and a rolling machine (though can be on the same shop floor), a CRM datacentre in the cloud, a casting line, and an inventory of dairy products, have different parameters to supervise and monitor. Components can also interoperate in performing the act of monitoring. They can collaborate in monitoring subsystems and subcomponents, or groups of sensors and actuators that together constitute a WSAN. Also in a group of components, on entity's monitoring agent can also find anomalies in its collaborative component, e.g. by receiving no, or a series of broken data from it. In addition, due modular nature of the system, resources and methods (models, sensors, etc.) used in monitoring a component or system (e.g. actuator, machine, unit, etc.) can be used in monitoring other systems of related more or less complex components, or in various setups. That gives rise to the need for the scalability of the monitoring mechanism. The dynamic nature of the distributed manufacturing systems provokes changes in structure, setups, addition or removal of resources, etc. that requires the monitoring mechanism to be accordingly scalable.

Detecting: After an anomaly or an unknown or undesirable change occurred, it shall be detected by each autonomous unit, while it is making a constant and real-time comparison between current parameters, and their approved values perceived from the context and the model attached to them giving them the required artificial intelligence. Here, given the heterogeneity of components, types of threats, risks, and dependability issues also varies. That implies various methods detection and comparison to be applied for each object as a part of its dependability and security model. The dependability model can impose specification limits in which changes are accepted and are not considered as anomalies. Statistical methods can also be applied to gain more precision and accuracy in finding changes.
Components can be constituted from, or themselves be a constituent of other components. They also collaborate in order to deliver various services. This fact suggests components to have high interoperability to collaborate in detecting risks, should one or more arise. Moreover, detection resources and databases, can work and be used in a modular way. For instance, databases for various detection methods for various risks can be put together to be applied in new setups and situations or for difference components.

Measuring: The process of measuring is done autonomously buy the components' model. In their dependability and security model, components have identification and measuring mechanisms (risk assessment methods like FMEA/FTA, and a database of possible risks and their parameters, etc.), to find the type, severity, and possible side-effects of risks. This model is to be developed and updated based on the real-time context in which given component is

working. Taking this into account, given the vast variety of contexts and emerged by inherent heterogeneity of the system, the measuring criteria, parameters, risk models etc. inevitably have to be designed and carried on accordingly.
Measuring mechanism has to take into account the possible side-effects of various risks in a given system of collaborating entities. Hence, risks of one entity can cause damages not only to that very component, but rather to some other parts of the system as well. Here, components need to have interoperable models to figure out the global effects of a threat or risk. Identification and measuring model and resources of several components can be mixed into one setup to perform the identification and measuring process of a more complex entity. The reverse is also true to use some parts of the models of a more complex component to deliver measuring process of a less complex entity, which is demonstrative of a part of modular characteristic of the Dependability and Security model. Likewise, resources and databases are scalable to suit dynamic and varying situations and meet corresponding requirements.

Alarming: When the risk is identified and its severity and possible side-effects are estimated, it is time to provide alarms accordingly. Compromised components, through their digital twins, or physically by any defined means, provide alarms to the system and to right entities. that firstly requires the component to be aware of the context in which it is working to know what entities need to be alarmed, and how the alarming process is to be carried out. Moreover, alarming mechanism can also get more complex when many components and agents are working together in a group or as sets of groups in a system as for example may appear in a sensor/actuator networks, demanding a group of agents cooperate in providing alarms to right entities (of the same or different level). This as a result would require interoperability of components and agents of possibly even heterogeneous types to perform the alarming process flawlessly. Alarming mechanism can be different based on the context and changes in the context may require additional resources or extensions to alarm system, that suggest the need for adequate level of scalability.

Taking action: As a part of their artificial intelligence, through their models, components can make decision about the right action to be taken in defence and clearing out the confronted issue. Here, knowing the context and its properties, the component can make more accurate decisions and perform them more effectively and efficiently. Also, interoperability can be manifested through cooperation of various components (more likely in a group), to perform right actions. For example, in some cases, components need to cooperate to solve an issue. e.g. to compensate a package loss or to guess based on each other’s data what the delayed or missing data of one sensor is. Components in their model, carry data about various possible types of risks and their countermeasures. The can also be equipped with real/virtual resource to apply required actions. These, resources and models can be extended, reduced, or rearranged to provide best fitting performance in various scales or combinations of resources in different occasions.

Reconfiguration: Following the “taking action” phase, takes place the reconfiguration phase, which deals with reconfiguring the compromised or broken-down component to prepare them for performing tasks again. For example, in some cases some nodes in CPSs may be loaded with ill data and codes by an attacker in order to eavesdrop, or to enter misleading data into the system. In such cases the compromised nodes can be disinfected and reloaded with proper codes. This phase is also to provide feedback to the system which elevates the knowledge of the system in forecasting and dealing with similar risks. That means, this information must be translated into a meaningful data, to be understood by other components, which requires the definition of share semantics, and is a part of components and models interoperability. Reconfiguration/ self-reconfiguration is to be done based on the real-time context. Required data can be reloaded and updated through component’s contest-awareness. It also suggests reusing the component in other ways. For instance, the component can be a processing agent that is going to be allocated to several other tasks, or to join other processing agents to perform one more complex task. while the dependability model and its databases themselves are made of modules, they also can be reconfigured and scaled to fit given conditions. The following table summarizes the interrelations between control loop and the properties of smart systems.

Table 2. Control loop steps and the relationship with smart manufacturing systems' properties

	Contribution	Requirements / method
Monitoring:		
Context-awareness	Parameters are derived from the context	Via sensors/ cloud
Interoperability	Hierarchical/heterarchical collaboration	Shared semantic/ ontology
Autonomy	Autonomously done by Intelligent Agents	Core model/ sensors
Modularity	Resources to be used in various setups	Modular resources
Scalability	Resources to be added or removed	Registering mechanism
Heterogeneity	Mechanism differs based on object type	Via core model
Detecting:		
Context-awareness	Current vs Approved context comparison	Via sensors/ core model
Interoperability	Hierarchical/heterarchical collaboration	Shared semantic/ ontology
Autonomy	Done by agents, and aided by core model	Agents/ data base
Modularity	Data-base and model to be re-useable	Modular risk Data-base

Scalability	To be seen data-base / detection methods	Updateable core model
Heterogeneity	Methods differs based on object type	Set in core model
Measuring:		
Context-awareness	Finding global measure based on context	Updating context data
Interoperability	Objects negotiating to find global measure	Via agents/ model
Autonomy	Done by agents and by using risk model	Risk model in the core
Modularity	Measurement resources to be shared reused in various setups	Risk categories to be modularly saved
Scalability	Criteria/ data-base to be changed/ updated	Scalable risk model
Heterogeneity	Criteria and risk model differ object-wise	Risk model definition
Alarm:		
Context-awareness	Right entities are known through context	Context update in model
Interoperability	In carrying alarm to various entities	Semantic definition
Autonomy	Done by agents after measuring risks	Agent collaboration
Modularity	Agents/functions to be used in new setups	Modular alarm resource
Scalability	methods/agents to be added or removed	Scalable alarm resource
Heterogeneity	Mechanism tries to stay the same for all	Semantic definition
Action:		
Context-awareness	Optimum decision/reaction context-wise	Via sensors/ models
Interoperability	Sharing resources/ information for taking optimum decision and action	Via semantic and ontology definition
Autonomy	To be done autonomously by agents	Agent collaboration
Modularity	Resources to be mixed in various setups	Modular agents/actuators
Scalability	Agents/actuators/ models to be scalable	Registering mechanism
Heterogeneity	Actions/resources differ by object type	Via model/ resources
Reconfiguration:		
Context-awareness	To be done based on context requirements	Via the model
Interoperability	Providing understandable feedbacks to others/ receiving required data	Via model and semantic definition
Autonomy	Semi/fully autonomous and done by agents	Model/ Agents
Modularity	Components may be reused in new setups	Modular components
Scalability	Extended/reduced structure in new setup	Scalable model/object
Heterogeneity	Done based on object type. Same feedback mechanism	Via model Same agent functionality

Multi-Agent Systems (MAS) and Model Implementation

The dependability and security model as described, with the properties shown, requires a toolset capable of backing such properties. Accordingly, Multi-Agents Systems (MAS) can be a decent candidate, since intrinsically, intelligent agents (IA) demonstrate responsiveness, proactiveness, goal-orientation, social-ability, scalability, flexibility, robustness, self-configuration, adaptability/ re-configurability, along with their decentralized architecture and learning capabilities [12]. CIRP Encyclopedia of Production Engineering defines an agent as a computational system that is situated in an unpredictable, dynamic environment where it is capable of exhibiting autonomous and intelligent behavior, and a multi-agent system is simply the community of interacting agents that together are able to solve complex problems that are beyond the capabilities of individual agents [13]. After determining the tool, the model is to be translated into an architecture composed of interacting agents. The first step would split the model task-wise onto single agents. Doing so, the following lists the agents that are defined to enable our model, and their task description. Subsequently, figure 2 shows their overall structure and collaborations' relations.

Agents for the core model:

- *Status manager*: Updates the status of components. The main part of context-awareness. Knows the approved context, authorities, topography of the system, etc.
- *Database*: Stores components models and risk data. Data are stored here in modules for each type of risk to be accessed by analysers and assessment agent. Other components when authorized, can have access to some of the data during negotiation or when they are new to the system, to get updated with vulnerabilities and measures, etc.
- *Assessment Agent*: Receives data from analyser and assesses the risk through negotiating with other components' assessors, receiving context data from TS manager, and having access to the database and the object model.
- *Interface*: For negotiations between agents of other components. Updating the topography and context information, more accurate and global risk assessment, providing access to databases of other components.

Agents related to control loop

- *Data filter*: Filtering out redundancies. Looking for useful data among loads of data.
- *Monitoring*: Looking for anomalies and risks, by comparing the current-state sensed data with current-state approved context. Then sends the detected cases to level one analyser.
- *Data-analyser level 1*: For simpler problems/quicker

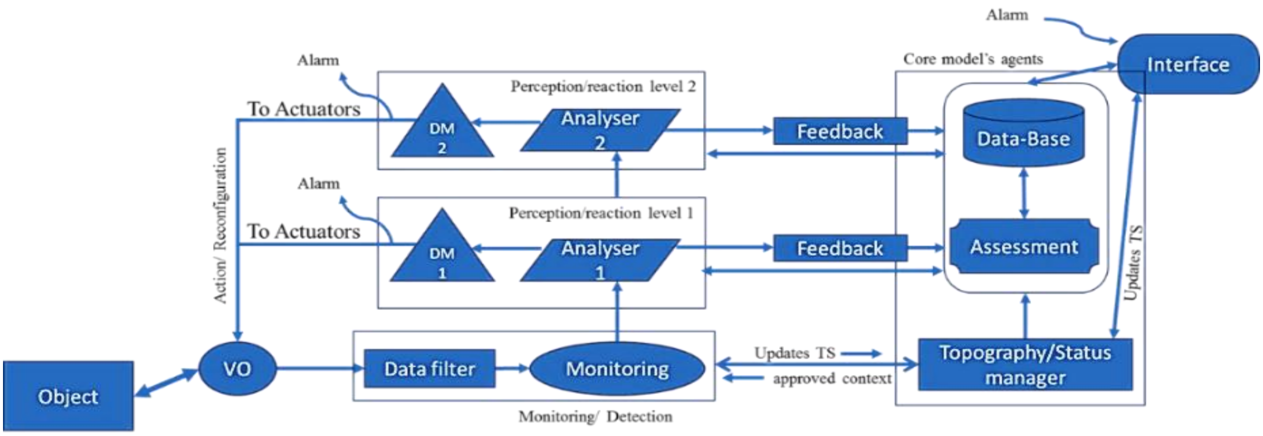

Figure 2. Proposed Agent Architecture for components' intelligent dependability and security

responses. Data analyser does the identification and measurement of risks. Lvl 1 analyser does the simple analysis and communicates with assessment agent to provide proper input for DM level one. It then provides feedback to the core model.

- *Data analyser level 2*: For more complicated problems, analyser level 1 sends the case to analyser level two with more abilities. If needed this analyser practises negotiations with other components agents to provide best global data of the risk to feed the DM level 2.
- *Decision-maker level 1*: For simpler reactions/ quicker responses. After supplied with proper risk information, releases alarm to right entities and send proper commands to actuators to apply right corrective actions, and reconfiguration when needed. Feedback is then being provided to the core model.
- *Decision-maker level 2*: Provides higher lever reactions, and reconfigurations, more advanced alarming system. If needed asks for collaboration of other agents and components resources to solve a problem. Feedback is then being provided to the core model.

As can be seen, data captured by sensors are sent to the VO (virtual object), which is the cyber representative or digital twin of the component (e.g. industry 4.0 component). Then, these data are filtered, monitored, and at the same time this stage is being fed into a status/topology manager to gain the approved context for comparing the filtered sensed data to detect anomalies. When detected, info is sent to the level one analyser, where in collaboration with assessment agent and database, the risk will be identified and measured. Then, risk data will be sent to the decision-making agent (DM level one) for decisions on the appropriate actions, e.g. alarms, and reconfigurations, if necessary and send the command actuators. However, if the problem is not trivial, and requires more advanced analysis, it will be sent to the analyser level two, where harder problems can be analysed and negotiations with other agents might be necessary to come up with the right measurements and analyses to provide accurate data for DM level 2. In decision making level two, more complex actions, and if required, negotiations with agents of other components (e.g. for sharing resources in fixing an issue), take place. After the actions are carried out, and issues are confirmed to be solved, the result is fed into the core model to update the risk assessor.

A simplified example to show the functionality of the architecture can be seen in a wireless sensor/actuator network. It may happen that one sink node for instance due to its higher data traffic be detected by adversaries who could intrude the network. Attacker may compromise the component and replace it with a malicious one to send ill data on its behalf. A system equipped with dependability and security model consisting of agents with roles and activities as defined in the table above, can monitor the node's activities and data traffic in real time, detect malicious activities or anomalies, identify it from the risk categories in its data base and estimates the impacts of the risk on the node and the network, or even other components or units in collaboration with this network, provides alarms to entities in collaboration with the compromised component to terminate their connection with it, if possible try to negotiate with other node of enough resources and capabilities to perform the task instead, and to reconfigure the compromised node after disinfection for bringing it back into the system.

Another simplified example can consider a job-shop unit with several automated machines and conveyors, using multi-agent systems to control their production system. Simultaneously, along with data collected through sensors, machines and controllers' interactions can be checked via "monitoring agent", receiving information being send and received by controller units and machines. Two of the possible risks can be either one of the controller agents itself be compromise by an adversary, or something unintentional occurs to one of the machines. Some cyber security risks associated with the former (i.e. controller agents of the example) have been shortly described previously in this section (e.g. cloning, repudiation, MITM attack, DDoS, eavesdropping, etc.), and some risks concerning the machines can be partial or full breakdowns, connectivity loss, etc. Taking breakdown of one of the machines as an

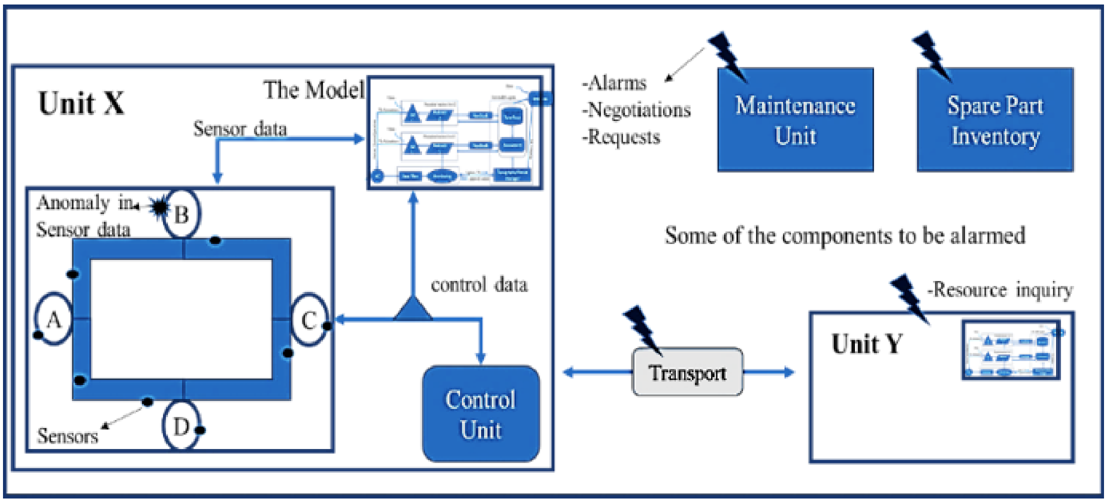

Figure 3. A sketch of the experimentation set up and actions in a smart manufacturing unit breaks down

example, should it happens, the monitoring agent will notice the change in the system in real-time, analyser will identify the issue (i.e. in this case breakdown of a machine) and will measure the impact on the system and components with which it collaborates and will send the data to decision maker agent. Alarms will be published to right entities (e.g. controllers, maintenance centre, spare part inventory, etc.). The machine will be terminated and called unavailable. The request will be sent to other agents for availability of another machine to do the task instead of the broken-down machine and after locating the alternative machine, ways (e.g. conveyors, AVGs, etc.), will be found to send the parts to the new alternative machine. And finally, a feedback will be sent to the core model for updating the data base and assessment model, and generating reports. Figure 3, shows a sketch demonstration of the second sample case explained above.

Conclusion

The use of cyber-physical systems in industries has gained a tremendous attention in recent years. But due to yet to be addressed dependability and security aspects of these systems, they still are not being widely used. In this paper, we introduce a theory based generic model to enable autonomous treatment of various possible security and dependability risks that can compromise smart manufacturing and cyber-physical systems. Moreover, based on available technologies, a structure based on multi-agent systems was developed and proposed to implement the dependability and security model. The next step of the study would be testing the model in various cases and extending its performance and capabilities. One case would be one explained in the example in the previous section. Another experiment will focus on the data security risks (e.g. intrusion attack, DDoS attack), on one component and test the models performance on detecting and blocking it, and after disinfection reconfiguring the component to be used again in the system. The experiment is to be done by simulating DDoS attack by overloading and increasing data traffic, and assessing the models reaction in handling the risk.

References

[1] S. Huang, C.-J. Zhou, S.-H. Yang, and Y.-Q. Qin, "Cyber-physical system security for networked industrial processes," *International Journal of Automation and Computing,* vol. 12, pp. 567-578, 2015.

[2] L. J. Wells, J. A. Camelio, C. B. Williams, and J. White, "Cyber-physical security challenges in manufacturing systems," *Manufacturing Letters,* vol. 2, pp. 74-77, 4// 2014.

[3] G. Wu, J. Sun, and J. Chen, "A survey on the security of cyber-physical systems," *Control Theory and Technology,* vol. 14, pp. 2-10, 2016.

[4] L. Zhang, Q. Wang, and B. Tian, "Security threats and measures for the cyber-physical systems," *The Journal of China Universities of Posts and Telecommunications,* vol. 20, Supplement 1, pp. 25-29, 8// 2013.

[5] S. Karnouskos, "Chapter 6 - Industrial Agents Cybersecurity," in *Industrial Agents*, ed Boston: Morgan Kaufmann, 2015, pp. 109-120.

[6] F. O. f. I. Security, "Industrial Control System Security: Top 10 Threats and Countermeasures 2016," 2016.

[7] F. Hu, Y. Lu, A. V. Vasilakos, Q. Hao, R. Ma, Y. Patil, *et al.*, "Robust Cyber–Physical Systems: Concept, models, and implementation," *Future Generation Computer Systems,* vol. 56, pp. 449-475, 2016.

[8] T. Sanislav, G. Mois, and L. Miclea, "An approach to model dependability of cyber-physical systems," *Microprocessors and Microsystems,* vol. 41, pp. 67-76, 2016.

[9] K. Wan and V. Alagar, "Context-Aware Security Solutions for Cyber-Physical Systems," *Mobile Networks and Applications,* vol. 19, pp. 212-226, 2014.

[10] H. Kühnle and G. Bitsch, "Smart Manufacturing Units," pp. 55-70, 2015.

[11] T. S. S. O. ITU, "Requirements for support of ubiquitous sensor network (USN) applications and

services in the NGN environment," ed, 2010.

[12] R. Unland, "Chapter 1 - Software Agent Systems A2 - Leitão, Paulo," in *Industrial Agents*, S. Karnouskos, Ed., ed Boston: Morgan Kaufmann, 2015, pp. 3-22.

[13] "Agent Systems," in *CIRP Encyclopedia of Production Engineering*, L. Laperrière and G. Reinhart, Eds., ed Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 33-33.