Providing an Access Control layer to Web-Based Applications for the industrial domain

Samuel Olaiya Afolaranmi, Borja Ramis Ferrer, Wael M. Mohammed, Jose L. Martinez Lastra
Tampere University of Technology,
Tampere, Finland
{samuel.afolaranmi, borja.ramisferrer, wael.mohammed, jose.lastra}@tut.fi

Mussawar Ahmad, Robert Harrison
University of Warwick,
Coventry, UK
{mussawar.ahmad, robert.hasrrison}@warwick.ac.uk

***Abstract*— The advances in virtual engineering brought about the creation of models of physical systems. This has been enhanced by the use of relevant standards and web technologies to develop interfaces for monitoring and controlling industrial systems at the shop floor or remotely over the internet. However, owing to the open nature of the internet as well as vulnerabilities associated with internet-based technologies, it becomes imperative to put in place security measures. This will allow to mitigate this particularly as it concerns the management, control, and monitoring of entities making use of the interface. The first step is to ensure an efficient user access control for web-based solutions. This article proposes an approach for developing an access control layer to web applications. In addition, a web-based application, i.e., the FASTory simulator, is used as the test-bed for implementing this approach.**

Keywords—virtual engineering; industrial engineering; web-based applications; access control.

I. INTRODUCTION

The creation of tools that may support engineers to better analyse, simulate and optimize processes is a priority in actual industrial enterprises. This is integrated into virtual engineering environments wherein users can monitor and control systems within models of the real systems. An example of how such virtual ecosystem may support both virtual and physical engineering of industrial automation systems is described in [1].

At present, engineers work towards the creation of flexible, adaptable, and intuitive interfaces for easy monitoring and control of industrial systems [2]–[4]. Internet-based technologies are commonly used for implementing such interfaces in the industrial field due to the emergence of the industrial internet of things or, in a sense, the interconnectivity of field devices through the Web [5]–[7]. Therefore, many developments include the employment of web-based technologies that permit the remote interaction with systems e.g.,[8], [9].

One of the major concerns of tools of this kind is the security and control access. In fact, this is a critical issue in any system connected to large and spanned networks because of their multiple access points and vulnerability to be attacked by malicious entities. Hence, something to consider at the design phase of any development is the manner in which the tool is to be accessed and the rights that users might have once logged in.

This research work proofs the concept within the FASTory simulator, which is a web-based simulator for monitoring and controlling an assembly line [10]. This tool has been developed by the FAST-Lab.[1], at Tampere University of Technology, and is used in European research projects (e.g., eScop[2] and C2NET[3]) and in Master of Science courses about industrial informatics. The simulator is employed as an interface of the real assembly line in order to remotely execute operations throughout REST services. Currently, the FASTory simulator is lacking an access control layer which would help administrators of the system to manage the information about who accesses the line and restrict, if needed, the set of actions that can be performed. Moreover, the ultimate version of such tool should permit the creation of instances of the model in order to execute operations on different replicas of the real system. This would be helpful in order to have a web-based simulator of the real system that can be used by multiple users at the same time. This is now possible only if the tool is run locally in a computer. The first step towards the aforementioned ultimate version is the management of users accessing to the system.

The objective of this research work is to design an access control layer to be implemented in web-based applications for management of different services, according to user credentials. The solution must manage the roles and access rights of users that desire to employ the simulator. Therefore, different types of users together with their access rights must be identified and defined. In particular, the approach is proved within the implementation of an application capable of managing user activities and access to the FASTory simulator.

The rest of the paper is structured as follows: Section II presents a literature review and industrial practices of virtual engineering and web-based applications. In addition, the section includes a short description about access control and security techniques relevant to this research work. Afterwards,

[1] http://www.tut.fi/en/fast/
[2] http://www.escop-project.eu/
[3] http://c2net-project.eu/

Section III presents a set of UML diagrams and main concept to be considered in order to implement the proposed access control layer. Then, Section IV describes i) a web-based simulator requiring access management and ii) how the access control can be implemented in such tool. Finally, Section V concludes the paper, including the further steps of this research work.

II. Literature Review and Industrial Practices

A. From Digital to Cyber-Physical Manufacturing Models

The modelling and simulation of systems within digital engineering environments continue to see more use owing to reduction in the cost of computing power. The key benefits of these modelling efforts are the testing and validation of a system's behaviour prior to physical build. This is beneficial as it facilitates better designs due to undesirable behaviours or design inconsistencies being identified and rectified earlier in the lifecycle reducing engineering time and costs [1]. Chryssolouris et al. highlight the use of digital engineering tools to support product development, production planning, and facility planning through a suite of software solutions [11]. This paradigm, referred to as "Digital Manufacturing", enables the integration of a myriad of models at the development phase and is supported by a range of software vendors including Siemens, Dassault Systèmes, and PTC [11], [12].

However, the digital and physical worlds remain isolated from one another. Digital models on their own are often insufficient for supporting later lifecycle phases e.g. build, commissioning, maintenance, operation, and re-engineering/re-configuration [13]. Unidirectional digital data integration from the physical factory to the digital model results in the generation of a "Virtual Factory" that enables more accurate simulations and thus more relevant optimisation strategies during operation [14], [15]. Westkämper and Jendoubi introduced the concept of the "Digital Factory" which proposed that data should flow bi-directionally between physical systems and digital models at run-time [16]. This enables monitoring of a system and consequently appropriate simulations can be carried out depending on the observed scenario [17].

With the advent of increased computational power that can be encapsulated within smaller volumes and lower mass, high speed communication, and Internet connectivity i.e. IoT, there is a paradigm shift towards Cyber Physical Production Systems (CPPS) that exploit the intersection of the cyber and the physical world [17]–[19]. In this case, models drive and respond to changes in the environment, be they on the shop floor or from stakeholders across the business. The new complex way in which these models are used require a greater degree of control to ensure safety and prevent unintentional changes being made to the system that may impact productivity. For this reason, one of the greatest challenges to address from an industrial standpoint to fully realise CPPS is that of security [17].

B. Web-based applications

Since the early days of the ICT field, the development of computer application tends to exploit available resources till the maximum limit. In the 90's, developers focused on standalone applications. At that time, the evolution of network-based applications was shaded by the lack of technologies [20]. Then, the late 90's marked the advent of web applications, thanks to the huge leaps in the ICT domain. This revolution, allowed programmers to invest more in a centralized approach called web-based applications. A web-based application can be defined as an application that is deployed in a server which hosts some services for several clients [21], [22]. This approach significantly reduced after-sales services. This means the users no more need to update or upgrade their application since they did not install it on their computers [23], [24]. In addition, web-based applications allow the users to benefit from these applications regardless of the computer specification or the operating system. Currently, with more advanced devices such as smart phones, web-based applications have become highly involved as a basic approach to develop programs.

Considering versatility as a feature of web-based applications, many standards are created and improved to accommodate this requirement [25]. For example, HTTP (Hypertext Transfer Protocol) for exchanging information through the network using the concept of request-response, WS (Web Sockets) for long term tunnelling or MQTT (Message Queue Telemetry Transport) for implementing the publish/subscribe message exchange. All these standards are served over the TCP/IP (Transmission Control Protocol and Internet Protocol) standards and provide a structure or media for manipulating the information in both server and client side. On the other hand, several standards are involved in web-based applications for visualization purposes i.e. HTML (Hypertext Markup Language) [26], SVG (Scalable Vector Graphics) [27] and CSS (Cascading Style Sheets) [28]. With the existence of these standards, the users visualize the web-applications via web browsers.

As it appears, the web-based application approach seems to be a robust, efficient and trustable solution. However, the security of data and information is highly necessary since the application services are exposed on the internet. In this manner, dedicated security standards [29] are developed for encrypting the transferred data and controlling the access of the available services of the web-based applications.

C. Security and Access control

In the use of applications, it is vital to ensure effective security in order to know who interacts with the application as well as the activities performed by such persons. Security refers to the provision and enforcement of safeguards and measures aimed at ensuring the protection of an asset: a resource or substance of great value. The value associated with an asset is what mandates the necessity for its protection. Some examples of assets are data, property, industrial applications, information etc. Security is implemented through policies, mechanisms, protocols etc. to offer protection against misuse, abuse, theft, damage, modification and unauthorized access to an asset. Specifically, it can be seen in two folds i.e., to prevent possible attacks and to mitigate the effect of an attack.

Fundamentally, the objectives of security are confidentiality, integrity and availability [30]–[32].

Confidentiality involves the protection against illegal access to information. It involves the use of mechanisms such as encryption, authentication, authorization and access control. Integrity has to do with preventing modification thereby ascertaining information accuracy. Measures such as access control, hashing, and cryptographic checksums are used to ensure integrity. Lastly, availability ensures that access to information at any time can be guaranteed. Redundancy and the use of clusters are typical measures for availability. As it relates to this research, access control is the focus, particularly in web-based applications for industrial domains. This is elaborated in the next paragraphs.

Access control is a security measure or mechanism that is used to manage accessibility to a resource. Its main purpose is to ensure that only authorized persons have access to the resource and are able to perform corresponding authorized operations. With this, interactions with/within the resource can be properly controlled and monitored thereby limiting the possibility of attacks or misuse by unauthorized or external persons. An example of existing technology for implementing access control is Kerberos [33], a client/server authentication protocol based on secret-key cryptography. The first step in the implementation of access control is the specification of security goals, identification of relevant stakeholders and specification of the security approach. These basic requirements can be summarized under three main headings: Policies, Models and Mechanisms [34], [35]. They represent *what should be done* and *how it should be done* in the domain of access control.

An access control policy defines what is permitted and the extent of the permission within the system where security is to be enforced. Simply put, it is a set of acceptable rules and actions by which access control is to be managed, thereby making a clear distinction between an authorized and unauthorized access [34]. For example, an access control policy may include a condition like *User_B has only Read and Write Access.* Primarily, it begins with the identification of the stakeholders. This includes the subjects (i.e. users), objects (i.e. the resource) and the actions (i.e. operations together with the restrictions to be enforced within the system). This becomes necessary in order to ensure that the security requirements are clearly outlined to system stakeholders [36].

An access control model is a formal representation of an access control policy. It specifies the relationship that exists between the stakeholders. Typical examples of an access control model as presented in [37] are mandatory access control (MAC), discretionary access control (DAC), Role-based access control (RBAC) and Rule-based access control (RBAC) etc.

Lastly, access control mechanisms provide a means or function for enforcing the access control policy specified earlier on. These mechanisms are responsible for granting or denying the requests of users as outlined in the access control policy. The mechanisms mainly focus on authentication and authorization.

III. An Access Control Layer to Web-Based Applications

An access control layer to a web-based application is typically responsible for managing all entries and interactions with the web-based application. The main objective of the access control layer is security management of entities using the application particularly in the management of identities, rights and access. The layer simply acts as a gateway, which receives incoming requests to the web-based application and then grants or denies access according to a specific security policy. The main requirements for actualizing the objectives of an access control layer to web-based applications can be grouped under four key actions that are explained in detail in the next section.

A. *Requirements of the access control layer*

Following the specification of the access control policy, model and mechanism, it becomes very clear how identities, rights and access should be managed. An Access control layer is required to provide four main functions, namely identification, authentication, authorization and accountability. These four functions are explained below.

- *Identification:* This is a process whereby a supplicant (an unconfirmed subject) selects an identity (i.e. an ID) or a tag by which that subject would be known by the access control layer. This is required so that each time the web-based application is to be accessed; the access control layer knows who is trying to make access. Identifiers may be codes, username, email etc. and will be unique for every user.
- *Authentication:* This extends the process of identification by verifying the identity of the supplicant. Typically, it involves the following mechanisms as outlined in [38]: something a supplicant knows (e.g. PIN, password, passphrase etc.), something the supplicant has (e.g. ID card, ATM card, token etc.) and something the supplicant can produce (e.g. biometrics). Authentication is required for the access control layer to grant access to valid subjects and deny invalid subjects. A minimum of two of these mechanisms is required for strong authentication.
- *Authorization:* Following the verification of the subject, the next step is authorization. This step involves the access control layer granting an authenticated user its corresponding access rights to the resource. Access rights are specified using access control list (ACL) which contains the rights to be granted to each authenticated user.
- *Accountability:* This ensures that every activity involved in accessing the resource is closely tracked and monitored by the access control layer. This involves tracking all successful and unsuccessful attempts to access the resource. This helps to guide against non-repudiation and can be achieved through system logs.

B. *Design architectural views*

In the design of an access control layer for web-based application, the following diagrams, which represent the flow, order and sequence of interaction between the stakeholders and the resource, have been developed. They are shown and explained below.

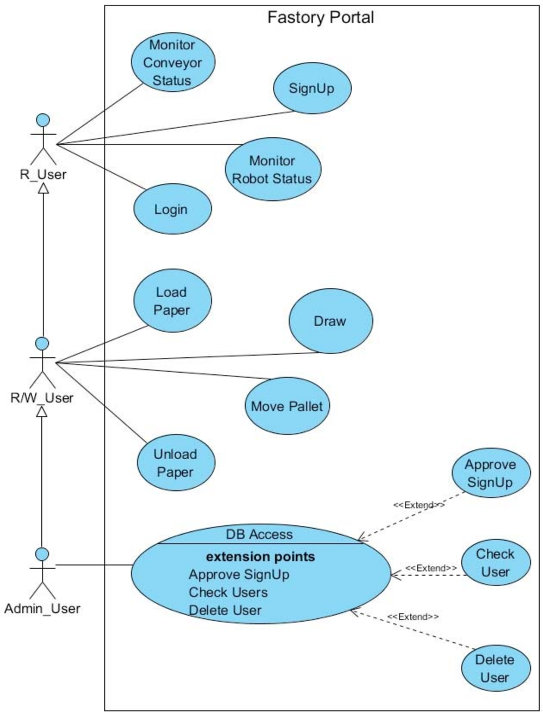

Fig. 1: Use case diagram

The use case diagram presents a model view of the web-based application and the different entities interacting with it. This involves mainly the actors/subjects (i.e. users), the system/object (i.e. the web-based application) and the use cases (i.e. the operation that can be carried out by the different subjects). As illustrated in Fig. 1, three users were defined i.e. *Read_User* (*R_User*), *Read/Write_User* (*R/W_User*) and *Admin_User*. The permissible operations are represented by use cases e.g., *Draw*, *Move Pallet*, *Load Paper.* These use cases are mapped to different actors using connectors i.e., a thin line that connects an actor with a use case e.g. The connection between *R/W_User* and *Draw* means that *R/W_User* can perform *Draw* operation.

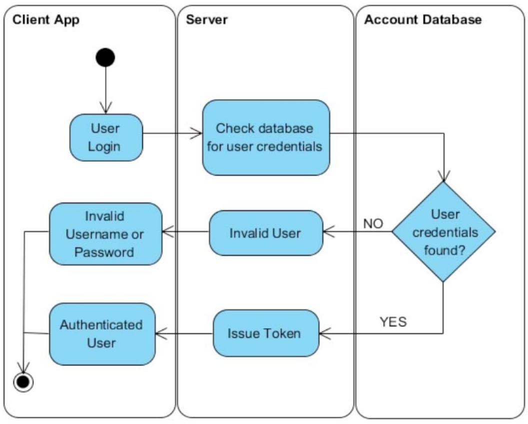

Fig. 2: User Authentication activity diagram

Authentication involves the verification of the identity of the different subjects accessing the web-based application. The authentication process as illustrated in Fig. 2 begins with the subject *logging in* through the client application. This involves the specification of the identity (username and password) of the subject i.e. user credentials. The user credentials are passed to the *Server*, which checks the *Accounts database* to confirm if the user credentials are available. If the user credentials are found, an *access token* is generated and issued by the server and sent to the client application and the user becomes authenticated. An access token is a string that represents the authorization issued to the client. It contains information about the user identity, the signature of the server and the access rights of the user. If the user credentials are not found, the server notifies the client application that the user is invalid. The outcome of this process is that the user is either authenticated (accepted) or invalid (denied).

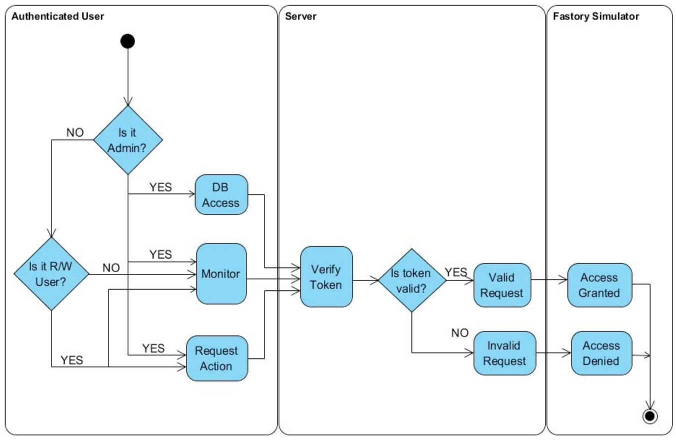

Fig. 3: User Request activity diagram

Following successful user authentication, an authenticated user can now begin to make requests to the web-based application. As already expressed in the Fig. 1, different users can perform different operations. As illustrated in Fig. 3, the first action when a user makes a request is to check the type of user. The user type determines the permissible requests. Once a request is made, the *access token* earlier issued by the *server* is attached to it and forwarded to the server. At the receipt of the request, the server performs *token verification*. In token verification, the server verifies its own signature and checks if the request is valid. The signature is verified to determine if it is genuine and valid e.g. if the token has expired or not based on the timestamp. If the token is found valid, the server grants access to the web-application where the response to the request made is provided. If the token is invalid, access is denied and request is not granted. The outcome of this process is that the authenticated user is either granted or denied access to the web-based application following the request made.

IV. PROVIDING ACCESS CONTROL TO THE FASTORY SIMULATOR

This Section explains how the approach is implemented in a specific use case: the FASTory simulator. Firstly, the web-based application is presented. Then, the selected technologies and their purpose in order to implement the access control are described.

A. *The FASTory simulator*

The FASTory assembly line is used to demonstrate an assembly process of mobile phones by drawing screens, keyboards and frames. As shown in Fig. 4, the ability of drawing three different models with three different colours provides up to 729 different product variants.

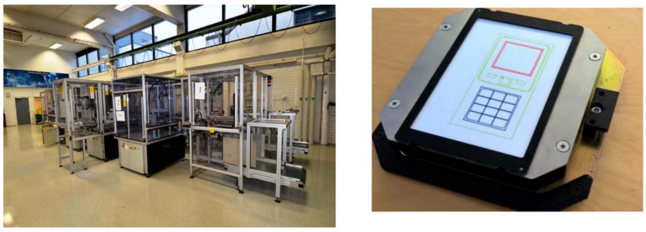

Fig. 4: FASTory Line on the left and final product on the right [10]

During the eScop project, the FASTory assembly line was used as a demonstrator for the project. At that time, a need for accessing the assembly line by other domestic partner was necessary since the project applications were developed by different partners across Europe. This was one of the main reasons why the simulator was developed and deployed on the university servers for allowing the partners and students to access and exploit it services.

As shown in Fig. 5, the FASTory line contains 12 work cells. A manual work cell (WS7) is dedicated to palletizing the assembly line. On the other side, a work cell (WS1) loads and unloads papers from pallets. The remaining work cells draw the mobile parts. This typology is identical to the real assembly line.

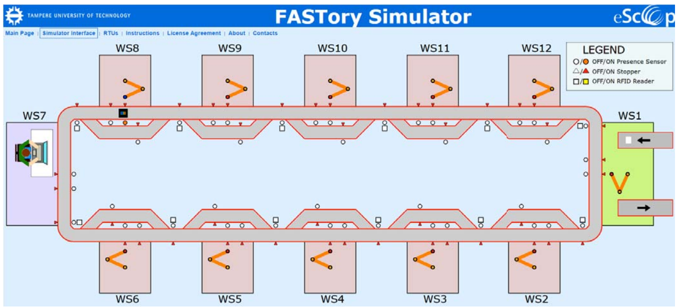

Fig. 5: FASTory Simulator

The main challenge of developing the FASTory simulator[4] was to copy and mimic the real line features. This includes the structure and the services in the IoT devices that installed in the FASTory line. As depicted in Fig. 6, both the line and the simulator host the services via HTTP Restful servers. This allows the user to migrate their implementation from one line to another without any complications. Besides, the simulator provides a description of all services that can be invoked through SWAGGER2.0[5] which helps for seamless and automatic interactions.

The FASTory simulator is intensively involved in both research and educational purposes in FAST-Lab. The advantages of the simulator include a virtual tool as a web-application. This means that users can make mistakes without harming themselves or the university facilities. As well, users might use it without being in the university. In fact, this feature raised a challenge of controlling the access of the users.

[4] http://escop.rd.tut.fi:3000/fmw
[5] http://www.swagger.io

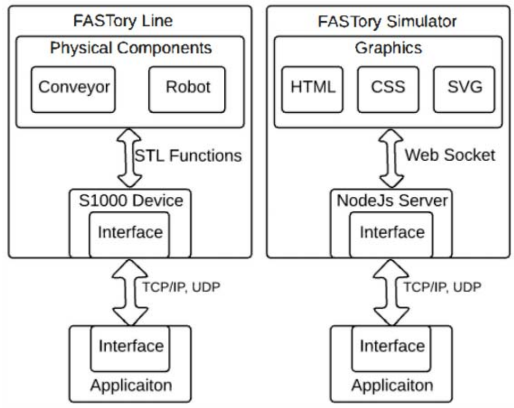

Fig. 6: FASTory assembly line vs. FASTory simulator architectures [10]

B. *Access control implementation*

As a means of addressing the security challenges already identified in the use of the FASTory simulator, an access control layer is required. As outlined in section III, an access control layer should provide *identification*, *authentication*, *authorization* and *accountability* functionalities. In order to achieve this, the proposed access control layer will be composed of three sub-layers that will handle key functionalities. These sub-layers are Graphical User Interface (GUI), Business Logic layer and the Storage layer, as depicted in following Fig. 7. The GUI will provide user identification functionality; the business logic layer will handle authentication and authorization functions while the storage layer will archive user account credentials and system logs. Relevant standards and web technologies such as HTML, CSS, JavaScript and OAuth 2.0 [39] are the selected ones to develop the different sub-layers of the proposed access control layer.

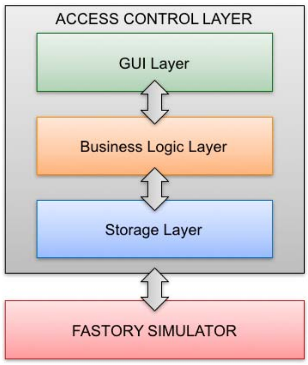

Fig. 7: Proposed Access control layer

The first layer is the GUI. It presents an interface for the user to clearly identify himself i.e. to tell the system who he is while attempting to access the FASTory simulator. This enables the performance of the first function of *Identification*. This involves the specification of user credentials (username and password) that are supplied through *input textboxes*, after which the supplied credentials are forwarded to the business logic layer for further actions. HTML, CSS and JavaScript will be used to develop the Graphical UI, which will provide functionalities for user identity creation (i.e. Sign up) and user access (i.e. Login).

The second layer is the business logic layer. This layer is responsible for *authentication* and *authorization*. Authentication and authorization will be implemented in line with OAuth 2.0 framework; an open standard for granting client applications access to protected resources on behalf of users through the coordination of authorization interactions between the users and the protected resource. This layer will mainly consist of a *web server* that will be developed using JavaScript (NodeJS [40] and ExpressJS [41]). User authentication will be *token-based* and will be handled using a JavaScript authentication middleware called PassportJS [42]. This layer will provide functionalities such as user identity creation, user identity and access rights verification, generation of system logs and provision of access to the storage layer. It will interact constantly with the storage layer to check for the availability and validity of users when a request is made.

The third layer is the storage layer. This layer is responsible for archiving user account credentials generated when a new user is created, updates made on user account information, and system logs generated from user activities while using or attempting to access the FASTory simulator. The storage layer is a typically a database. For this layer, MongoDB NoSQL database [43] will be used.

Following the approach presented in the previous paragraphs detailing the different sub-layers together with their respective functionalities, the proposed access control layer can be developed to effectively control, manage and monitor the role and access rights of FASTory simulator users.

V. Conclusion and Further Work

This manuscript describes an approach for providing an access control layer for web-based applications. The approach consists on the implementation of several components following a set of activity diagrams which are presented in Section III. The information flow of such diagram and main activities shown in the UML use case diagram fulfil the general requirements discussed in the same section. Moreover, the approach adds the authentication of users within specific and temporal tokens which permit to perform different actions depending on the type of user that is logged in the system. Finally, all the user data is stored in a DB, which can only be accessed by users with the administrator role.

In particular, the implementation of presented approach will enable the access management of a web-based assembly line simulator: the FASTory simulator, which is used for both research and education projects at Tampere University of Technology. The technologies and tools that are compatible with the current version of the simulator are described. This includes the employment of i) NodeJS and ExpressJS for server side developments, ii) MongoDB for the Database implementation and management and iii) PassportJS for managing a secure access to the simulator.

This research work is the first step towards the creation of a more secure, flexible and controllable web-based simulator. Once the access control is implemented, the tool might be instantiated and accessed by separate users. Therefore, it will be capable of supporting multiple sessions at the same time. This is needed when the FASTory simulator is used by remote users (i.e., international partners or clients) and students simultaneously in order to provide a personal session. Then, further versions of the simulator will permit the use of simulator instances with different rights, for executing services and monitoring the assembly line behaviour, depending on the user type. Moreover, further research will compare different technologies to implement the access control requirements in order to employ and justify the best one for ensuring a secure interface between any kind of users and the web-based application.

Acknowledgment

The project leading to this paper has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement n° 644429 correspondent to the project shortly entitled MUSA, Multi-cloud Secure Applications.

In addition, the authors gratefully acknowledge the support of the UK EPSRC under the iCASE PhD studentship (grant number: 1377735).

References

[1] R. Harrison, D. Vera, and B. Ahmad, "Engineering Methods and Tools for Cyber-Physical Automation Systems," *Proc. IEEE*, vol. 104, no. 5, pp. 973–985, May 2016.

[2] "PLM−Product Lifecycle Management: Siemens PLM Software." [Online]. Available: http://www.plm.automation.siemens.com/en/. [Accessed: 21-Mar-2017].

[3] "Discrete Event Simulation Software | Manufacturing, Supply Chain & Healthcare Simulation Software | Arena Simulation." [Online]. Available: https://www.arenasimulation.com. [Accessed: 21-Mar-2017].

[4] "Home," *INCONTROL Simulation Solutions*. [Online]. Available: http://www.incontrolsim.com/. [Accessed: 21-Mar-2017].

[5] P. Leitão, S. Karnouskos, L. Ribeiro, J. Lee, T. Strasser, and A. W. Colombo, "Smart Agents in Industrial Cyber #x2013;Physical Systems," *Proc. IEEE*, vol. 104, no. 5, pp. 1086–1101, May 2016.

[6] S. Farné, E. Bassi, F. Benzi, and F. Compagnoni, "IIoT based efficiency monitoring of a Gantry robot," in *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*, 2016, pp. 714–719.

[7] C. Alexakos, C. Anagnostopoulos, and A. P. Kalogeras, "Integrating IoT to manufacturing processes utilizing semantics," in *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*, 2016, pp. 154–159.

[8] B. Ramis *et al.*, "Knowledge-based web service integration for industrial automation," in *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*, 2014, pp. 733–739.

[9] J. Jokinen, M. B. Ambat, and J. L. M. Lastra, "Condition monitoring for distributed systems with reconfigurable user interfaces and data permissions," in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, 2016, pp. 5705–5710.

[10] W. M. Mohammed, A. Lobov, B. R. Ferrer, S. Iarovyi, and J. L. M. Lastra, "A web-based simulator for a discrete manufacturing system," in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, 2016, pp. 6583–6589.

[11] G. Chryssolouris, D. Mavrikios, N. Papakostas, D. Mourtzis, G. Michalos, and K. Georgoulias, "Digital manufacturing: History, perspectives, and outlook," *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.*, vol. 223, no. 5, pp. 451–462, May 2009.

[12] P. Waurzyniak, "Connecting the Digital World with the Factory Floor," *Manufacturing Media Engineering*, vol. 1, p. 34, Apr-2014.

[13] R. Drath, A. Luder, J. Peschke, and L. Hundt, "AutomationML - the glue for seamless automation engineering," in *2008 IEEE International*

Conference on Emerging Technologies and Factory Automation, 2008, pp. 616–623.
[14] W. Kuhn, "Digital Factory - Simulation Enhancing the Product and Production Engineering Process," in *Proceedings of the 2006 Winter Simulation Conference*, 2006, pp. 1899–1906.
[15] W. Terkaj, T. Tolio, and M. Urgo, "A virtual factory approach for in situ simulation to support production and maintenance planning," *CIRP Ann. - Manuf. Technol.*, vol. 64, no. 1, pp. 451–454, 2015.
[16] E. Westkämper and L. Jendoubi, "Smart factories - manufacturing environments and systems of the future," in *Progress in Virtual Manufacturing Systems*, 2003, pp. 13–16.
[17] L. Monostori *et al.*, "Cyber-physical systems in manufacturing," *CIRP Ann. - Manuf. Technol.*, vol. 65, no. 2, pp. 621–641, 2016.
[18] L. Wang, M. Törngren, and M. Onori, "Current status and advancement of cyber-physical systems in manufacturing," *J. Manuf. Syst.*, vol. 37, Part 2, pp. 517–527, Oct. 2015.
[19] "Lee and Seshia, Introduction to Embedded Systems." [Online]. Available: http://leeseshia.org/. [Accessed: 21-Mar-2017].
[20] "The history of the Web - W3C Wiki." [Online]. Available: https://www.w3.org/wiki/The_history_of_the_Web. [Accessed: 21-Mar-2017].
[21] S. Karnouskos, P. G. da Silva, and D. Ilić, "Developing a web application for monitoring and management of Smart Grid neighborhoods," in *2013 11th IEEE International Conference on Industrial Informatics (INDIN)*, 2013, pp. 408–413.
[22] S. Iarovyi, J. Garcia, and J. L. M. Lastra, "An approach for OSGi and DPWS interoperability: Bridging enterprise application with shop-floor," in *2013 11th IEEE International Conference on Industrial Informatics (INDIN)*, 2013, pp. 200–205.
[23] A. W. Colombo, T. Bangemann, and S. Karnouskos, "IMC-AESOP outcomes: Paving the way to collaborative manufacturing systems," in *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*, 2014, pp. 255–260.
[24] P. Balda and M. Štětina, "eScop project physical layer development: INCAS conveyor line pilot case study," in *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*, 2015, pp. 1044–1049.
[25] I. Medeiros, N. F. Neves, and M. Correia, "Securing energy metering software with automatic source code correction," in *2013 11th IEEE International Conference on Industrial Informatics (INDIN)*, 2013, pp. 701–706.
[26] "HTML 5.1." [Online]. Available: https://www.w3.org/TR/html/. [Accessed: 21-Mar-2017].
[27] "Scalable Vector Graphics (SVG) 1.1 (Second Edition)." [Online]. Available: https://www.w3.org/TR/SVG/. [Accessed: 21-Mar-2017].
[28] "Cascading Style Sheets." [Online]. Available: https://www.w3.org/Style/CSS/Overview.en.html. [Accessed: 21-Mar-2017].
[29] A. M. Osman, A. Dafa-Allah, and A. A. M. Elhag, "Proposed security model for web based applications and services," in *2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE)*, 2017, pp. 1–6.
[30] T. Lu, J. Zhao, L. Zhao, Y. Li, and X. Zhang, "Security Objectives of Cyber Physical Systems," in *2014 7th International Conference on Security Technology*, 2014, pp. 30–33.
[31] J. Puttonen, S. O. Afolaranmi, L. G. Moctezuma, A. Lobov, and J. L. M. Lastra, "Security in Cloud-Based Cyber-Physical Systems," in *2015 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)*, 2015, pp. 671–676.
[32] S. O. Afolaranmi, L. E. G. Moctezuma, M. Rak, V. Casola, E. Rios, and J. L. M. Lastra, "Methodology to Obtain the Security Controls in Multi-cloud Applications," presented at the 6th International Conference on Cloud Computing and Services Science, 2017, pp. 327–332.
[33] F. Al-Ayed and H. Liu, "Synopsis of Security: Using Kerberos Method to Secure File Transfer Sessions," in *2016 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2016, pp. 1016–1020.
[34] P. Samarati and S. C. de Vimercati, "Access Control: Policies, Models, and Mechanisms," in *Foundations of Security Analysis and Design*, 2000, pp. 137–196.
[35] S. Murugesan and I. Bojanova, "Distributed Access Control in Cloud Computing Systems," in *Encyclopedia of Cloud Computing*, Wiley-IEEE Press, 2016, p. 744-.
[36] "Access Control Cheat Sheet - OWASP." [Online]. Available: https://www.owasp.org/index.php/Access_Control_Cheat_Sheet. [Accessed: 21-Mar-2017].
[37] Y. Zhang, K. Chen, Y. Liu, and Y. Lian, "A Lattice-Based Access Control Model for Social Networks," in *2016 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, 2016, pp. 54–61.
[38] M. E. Whitman and H. J. Mattord, *Principles of Information Security*. Cengage Learning, 2011.
[39] D. Hardt, "The OAuth 2.0 Authorization Framework." [Online]. Available: https://tools.ietf.org/html/rfc6749. [Accessed: 21-Mar-2017].
[40] "Index | Node.js v7.7.3 Documentation." [Online]. Available: https://nodejs.org/api/. [Accessed: 21-Mar-2017].
[41] "Express 4.x - API Reference." [Online]. Available: https://expressjs.com/en/4x/api.html. [Accessed: 21-Mar-2017].
[42] "Passport." [Online]. Available: http://passportjs.org/docs. [Accessed: 21-Mar-2017].
[43] "The MongoDB 3.4 Manual — MongoDB Manual 3.4," *https://github.com/mongodb/docs/blob/master/source/index.txt*. [Online]. Available: http://docs.mongodb.com/manual/. [Accessed: 21-Mar-2017].