Access Control for Smart Manufacturing Systems*

Björn Leander[1,2][0000−0003−2488−5774], Aida Čaušević[1][0000−0001−5293−3804], Hans Hansson[1][0000−0002−7235−6888], and Tomas Lindström[2]

[1] Mälardalen University, Västerås, Sweden {bjorn.leander, aida.causevic, hans.hansson}@mdh.se

[2] ABB Industrial Automation, Process Control Platform, Västerås, Sweden tomas.lindstrom@se.abb.com

Abstract. In the ongoing 4^{th} industrial revolution, a new paradigm of modular and flexible manufacturing factories powered by IoT devices, cloud computing, big data analytics and artificial intelligence is emerging. It promises increased cost efficiency, reduced time-to-market and extreme customization. However, there is a risk that technical assets within such systems will be targeted by cybersecurity attacks. A compromised device in a smart manufacturing system could cause a significant damage, not only economically for the factory owner, but also physically on humans, machinery and the environment.

Strict and granular Access Control is one of the main protective mechanisms against compromised devices in any system. In this paper we discuss the requirements and implications of Access Control within the context of Smart Manufacturing. The contributions of this paper are twofold: first we derive requirements on an Access Control Model in the context of smart manufacturing, and then asses the Attribute Based Access Control model against these requirements in the context of a use case scenario.

Keywords: Access control · Industrial Automation and Control Systems · Smart Manufacturing · Industry 4.0 · Cybersecurity

1 Introduction

Smart manufacturing [1, 2] is a development of traditional manufacturing implying a shift from production of big series of identical units towards a highly dynamic manufacturing environment that is tuned to extreme customization, fluctuating markets, and specific customer needs. The technology to enable this dynamic behavior includes an increasing amount of interconnected sensors, actuators and related services in the manufacturing environment, in combination

* This work is supported by the industrial postgraduate school Automation Region Research Academy (ARRAY), funded by The Knowledge Foundation. The authors would like to acknowledge Andrea Macauda and Axel Haller for valuable discussions and feedback.

with e.g., cloud technologies, data lakes, artificial intelligence, etc., for inference and aid to decision-makers [3].

In the dynamic smart manufacturing environments of today and tomorrow, the traditional view of the manufacturing networks being air-gapped and protected by proprietary technologies no longer holds [4]. Considering that a great number of these devices introduced in a smart manufacturing system have wireless connectivity, are living on the edge of the network, possibly with direct connections to unprotected networks, there is an increasing risk that any of these devices become compromised. This has been illustrated in a number of attacks targeting industrial systems over the last ten years [5]. To protect the manufacturing environment from compromised devices, there is a need to introduce a number of security measures in the form of e.g., Intrusion Detection Systems (IDSs), end-to-end security for sensitive data, malware detection and fine-grained access control.

In this article we focus on *access control*, as one of the basic security functions in any system, enabling access restriction to operations on resources only to legitimate authorized subjects. The models for access control that are currently in use are tailored to authorize human subjects performing operations on digital assets, mainly supporting use-cases for rather static sets of resources and subjects or roles. These traditional models do not provide a high level of flexibility for expressing fine-grained policies [6], as frequently needed in smart manufacturing. Attribute Based Access Control (ABAC) is a relatively new model for policy formulation, potentially useful for machine-to-machine authorization [7, 8]. Our aim in this paper is to derive requirements on access control in smart manufacturing systems, and evaluate ABAC against those requirements.

The remainder of this paper is structured as follows: Background is presented in Section 2. In Section 3 we identify a compilation of requirements on access control. In Section 4 a use cases scenario for smart manufacturing is presented, including suggestions on policy formulations for ABAC in this context. A discussion on how ABAC relates to the requirements are provided in Section 5. Scientific work related to our findings is presented in Section 6. Finally the work is summarized and some remaining challenges and future areas of research are described in Section 7.

2 Background

2.1 Smart Manufacturing concepts

The term *smart manufacturing* is used for describing the 4^{th} industrial revolution from a manufacturing perspective, with origin in a joint work by several agencies in the US [3]. Smart manufacturing is sometimes also referred to as Cyber Physical Production Systems (CPPS) [9] and Intelligent Manufacturing Systems (IMS)[3].

[3] More information available at `http://ims.org`.

In general, smart manufacturing encompasses the whole manufacturing chain, from supply to production and logistics. Data collected from sensors within the process are used for advanced data analytic in order to improve the overall operations. A key aspect of smart manufacturing is to provide flexibility and dynamicity in the manufacturing environment by modularization of process steps, so that process steps can be combined and re-combined based on current production requirements [10]. Integrating modular process steps in the manufacturing system enables Workflow as a Service (WfaaS), where vendors of production equipment could sell pre-fabricated process-steps as a service, allowing the factory owners to easily adapt to fluctuating market demands.

2.2 Cybersecurity Threats to Smart Manufacturing Systems

The increasing amount of connected and interconnected devices required for the data acquisition together with external stakeholders in need to access the data, considerably increases the attack surfaces of a smart manufacturing system. Furthermore, as different modules within the system are dynamically connected to each other, the authorization of privileges between devices and services must be equally dynamic to allow continuous secure operation. According to Tuptuk et al. [4], cybersecurity is rather seen as a characteristic than as a design principle within the development of smart manufacturing systems, a misconception that may lead towards many systems being insufficiently protected.

The CIA-model is often used to describe desired security characteristics of a system (**C**onfidentiality, **I**ntegrity and **A**vailability [11]). In the context of smart manufacturing, a cybersecurity attack may breach any of these characteristics, e.g., leading to possible loss of Intellectual Property (IP), costly errors in production due to unreliable or faulty data, and down-time or potentially safety-related threats to production machinery, workers amd the environment.

2.3 Access Control definitions

There are a number of guiding principles for access control, the most notable ones being [12]:

1. **Least privilege**, requires that a subject should only have the minimum possible privileges needed to perform its tasks.
2. **Complete mediation** requires that any access to a resource must be monitored and verified.

Following these principles in a smart manufacturing system will help minimize the harm an adversary can do after gaining an initial foothold within the system, and even shorten the detection time, since failed access attempts typically are logged and monitored.

Sandhu el al. [13] describe access control as being comprised of models at three different layers, **P**olicy, **E**nforcement and **I**mplementation (PEI). Policy models are used to formalize high level access control requirements, enforcement

level models describe how to enforce these policies from a systems perspective, and the implementation level models show how to implement the components and protocols described by the enforcement model. Following the PEI-model, this work is focusing on the policy-layer models, meaning that we will discuss how rules can be expressed, rather than mechanisms to enforce the rules.

A prerequisite for robust access control is reliable authentication of entities. In this work we assume that a trustworthy solution for authentication is used.

Historically, Mandatory Access Control (MAC) and Discretionary Access Control (DAC) have been the two main paradigms within access control [14]. MAC is based on security classifications of resources, combined with security clearances for subjects, e.g., top-secret content only readable for subjects with the highest security clearance. In DAC on the other hand, the privileges are defined as a relation between the resource and subject, often with the subject allowed to transfer its privileges.

Role-Based Access Control (RBAC) is a model building on principles from both DAC and MAC, where subjects are assigned to one or several roles that may be hierarchically ordered. Privileges are derived from the roles rather than from the subject. In a number of studies it has been shown that the traditional access control schemes are not sufficient for, e.g., cloud-connected cyber physical systems [15] and IIoT [16].

2.4 Attribute Based Access Control (ABAC)

A relatively novel scheme in access control is ABAC. In the work of Yuan and Tong [17], the application is aimed at providing access control in web services. They show that the granularity of the traditional RBAC scheme is not fine enough, in order to formulate certain policies easily expressed in natural language. The following example is extracted from [17], and provided here to introduce ABAC and illustrate that such natural language rules are difficult to express using the traditional Access Control models:

Let us assume we need to grant a user access to movies in an online streaming service. In this example we consider a movie *rating* (R, R-13, G) and $freshness$ (New release, Normal), mapped to the user *age* and subscription *category* (Budget, Premium). The following to rules apply for a user to be allowed to watch a movie:

1. To watch movies with rating R, user must be over 17 years old, and for movies with rating R-13, over 12 years.
2. To watch a New release, the user subscription category must be Premium.

In ABAC, the subject s's right to perform operation o on a resource r in environment e is calculated based on attributes of the subject, resource and environment, A_s, A_r, A_e respectively:

$$allow_o(s, r, e) \leftarrow f(A_s, A_r, A_e)$$

For the movie streaming service, the following policy rules can be expressed, based on the viewer and movie attributes:

$$f_1(s, r, e) = (rating(r) = \text{G}) \vee (age(s) > 12 \wedge rating(r) = \text{R-13}) \vee (age(s) > 17)$$

$$f_2(s, r, e) = (freshness(r) = \text{normal}) \vee (category(s) = \text{premium})$$

allowing for rules to be further combined:

$$allow_{view}(s, r, e) = f_1(s, r, e) \wedge f_2(s, r, e).$$

Several works on ABAC have been conducted, including two major standardization efforts in the area: eXtensible Access Control Markup Language (XACML) by OASIS [18], and Next Generation Access Control (NGAC) by NIST [19]. A comparison between NGAC and XACML is provided by Ferraiollo et al. [20].

Authorization architectures for ABAC typically contain a number of standard components [8, 20, 18]: A subject can only access a resource through the the Policy Enforcement Point (PEP), which acts as a mediator for any privilege request. The PEP queries an authorization decision from the Policy Decision Point (PDP) that reads policy information from the Policy Information Point (PIP), which has access to Policy Data. An administrator maintains Policy Data through a Policy Administration Point (PAP).

3 Access Control Requirements on Smart Manufacturing

In this section we formulate a list of requirements on access control for a smart manufacturing system. To provide such a list we have studied the literature, using an adapted version of the method presented by Kitchenham [21]. We have selected relevant requirements guided by the basic principles for access control. For details regarding the literature review and used protocol we refer the reader to [22].

3.1 Requirements related to a traditional manufacturing system

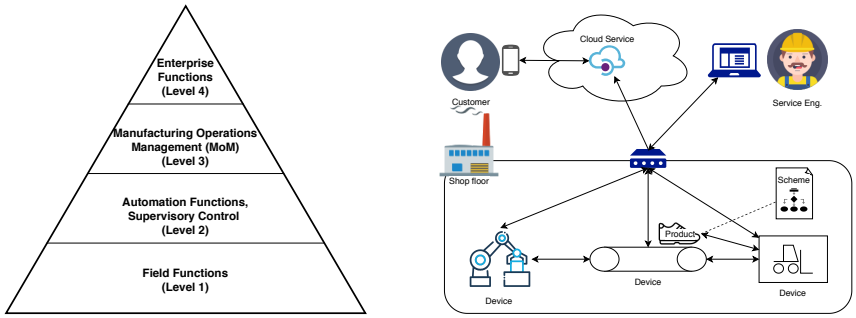
A traditional manufacturing system can be described as an Industrial Automation and Control System (IACS) which typically supports safety- and security critical processes [23]. IACS are used to control and monitor a wide range of different types of physical processes, e.g., in chemical industries, power plants, and discrete manufacturing.

An illustration of a generic traditional manufacturing system architecture can be seen in Figure 1a, inspired by the Purdue Enterprise Reference Architecture (PERA) [24]. These systems contain a number of *essential functions* that cannot be disrupted, and that are required to maintain health, safety and availability of the equipment under control. In principle, a security measure must not result in a state of the system that could lead to Health, Safety or Environmental (HSE) consequences. A number of requirements on the access control arise from the need to support essential functions [23]:

R1 Availability: The manufacturing system should be operable even if some components fail, e.g., a failed server or a disruption in network connectivity between shop floor and cooperate network should not interfere with production.

R2 Security measures must not have a negative impact on essential functions. Specifically, HSE-related incidents shall not happen as a result of loss of control due to lack of privileges.

Non-Repudiation is also an important characteristic of access control that is required by e.g., IEC 62443[4]. We choose not to list it as a requirement in this context, as the focus of this work is on mechanisms for access control at a policy level and non-repudiation refers to logging and auditing of execution of granted privileges.



(a) A traditional manufacturing system architecture based on PERA

(b) A simplistic smart manufacturing scenario

Fig. 1

3.2 Requirements related to smart manufacturing systems

A number of requirements on access control are shared between the smart manufacturing domain and other dynamic systems of interconnected cyber-physical systems. These requirements arise through the evolution of the traditional automation pyramid towards a service oriented and decentralized system [10, 15]:

R3 Diversity: A system should provide support for several different kinds of applications to be integrated throughout the whole life-cycle. This implies that multiple categories of users, usages of services and production related data shall be supported by the system.

R4 Scalability: A system should be scalable with regards to users and policies. Management of a huge amount of devices, services and users must be simple and cost efficient, still providing necessary transparency.

[4] Part 3-3: System security requirements and security levels, Ed 1.0, 2013

R5 Flexibility: The access control mechanism shall provide an easy way of defining new policies.
R6 Efficiency: The computational cost of inferring privileges should not negatively impact the performance of the system as a whole.

From [16, 25–27] we have derived the following requirements specific to the smart manufacturing domain:

R7 Temporal policies: The required privileges to perform a task may shift between each batch, or even between each produced unit. The access control model shall be equally flexible, following the principle of least privilege.
R8 Logical ordering: Production in a manufacturing environment is usually described as a workflow, meaning that the order of the actions, and the number of times an action can be executed could be limited. The access control model shall be able to express such logical ordering at a policy level.

3.3 Generic access control requirements

In the following we describe generic access control requirements not covered in earlier sections. These requirements are the result of discussions with industrial experts:

R9 Transparency: From the perspective of an administrator, it must be easy to deduce current state of granted privileges, and historical changes to privileges. This transparency requirement could also extend to other privileged users.
R10 Delegation: For certain scenarios, it should be possible to transfer privileges from one subject to another through delegation.

4 A Smart manufacturing Scenario

In this section we describe a generic smart manufacturing scenario to be analyzed from an access control perspective. We provide a discussion on how ABAC can be applied to the scenario in Figure 1b. The scenario essentially follows the set-up of a service-driven architecture for manufacturing, described in [10, 26], connected to the IEC 61499 [28].

Let us assume that a product p is to be manufactured. p is associated to a set of devices $\mathbf{D}$ that must perform tasks on p for it to be finalized. In order to perform the actions there is a need for a device $d \in \mathbf{D}$ to share information, and execute operations on one or more other devices in $\mathbf{D}$, according to the manufacturing scheme defined specifically for p.

The customer c wants to read information from the system for data related to product p via a cloud service, e.g., production status and expected delivery time. A 3rd party service organization o is responsible for maintaining some of the devices in $\mathbf{D}$, and must therefore be able to read status and perform

service-related actions on the devices, e.g., reading health records and performing firmware upgrades.

In practice, the rules we describe in the following would be implemented using e.g., XACML [18]. For brevity, we choose to describe only the logical expressions of the policies, following the formalism introduced in [17]. The following attributes will be used in the ABAC policy formulations below:

- $batch_{id}(x)$[5] is the value of the batch attribute, related to a produced entity p or related to the current context of execution for a device d.
- $batches(e)$ is the set of all active batches in the manufacturing environment e.
- $purchases(c)$ is the set of batches that customer c has purchased. In this example we assume a one-to-one connection between customer and batch.
- $contract_{id}(d)$ is the value of the service-contract attribute related to a device d.
- $contracts(o)$ is a set of contracts under which the service organization o is working.
- $idle(d)$ is a Boolean attribute indicating that device d is currently idle if TRUE or busy if FALSE.
- $*$ is used to indicate an unassigned attribute value.

Given the example, we are able to show some interesting characteristics regarding access control in smart manufacturing systems.

C1 Machine to Machine (m2m) cooperation is limited by the current entity/batch attribute.
C2 Customer outside organization read rights are limited by a purchase.
C3 Service organization personnel (possibly 3rd party) having read and e.g., firmware-update rights limited by a contract.

Using ABAC, a policy to satisfy characteristic C1 could be expressed as:

$$allow_{op}(d_1, d_2, e) = (batch_{id}(d_1) = batch_{id}(d_2)) \wedge batch_{id}(d_1) \in batches(e) \quad (1)$$

Stating that the privilege to perform the operation will be granted only if the devices d_1 and d_2 have the attribute $batch_{id}$ assigned with the same id, and that id is among the active batches in the environment. Similarly, the customer could be granted privileges based on a combination of attributes of the data and attributes of the customer, which would allow a very fine-grained model for authorization (i.e., related to characteristic C2). One simple example of an authorization rule could be:

$$allow_{read}(c, p, e) = batch_{id}(p) \in purchases(c) \quad (2)$$

Note that in this specific rule, as well as the following, no environment attributes are used. Entity e will still be used in the declaration of the formula for consistency reasons. The above equation is stating that reading information about

[5] Here x is used as variable representing either an entity p or a device d.

product p is allowed if the $batch_{id}$ for p is present in the set of $purchases$ that the customer c has done. Typically such information is retrieved through filtering, i.e., the privilege is enforced by the application or API implementation, which is a much weaker condition than granting privileges through the access control mechanism. In fact, following the traditional practice, the access control mechanism will grant read-access to any valid customer and rely on the application to perform the correct filtering.

The privileges of personnel from the service organization (i.e., related to characteristics C3) is an interesting issue, since there may be many factors within the manufacturing environment that should prevent interruption or additional load on devices or services related to direct operation. In a classical service operation scheme, privileges to perform maintenance related operations may not be allowed except when the production unit is halted for planned maintenance or similar. However, in a smart manufacturing environment, this may be a common case, especially for WfaaS scenarios, i.e., it is up to the service organization to make sure that the workflows are running as needed. In these cases, an ABAC policy could be used to minimize the risk of disturbing ongoing operations. For example, an attribute indicating that the device is currently in use could inhibit the right to perform disruptive actions, and attributes indicating a need to perform an update or a similar disruptive maintenance action could inhibit the device from being assigned to a batch. The following rule could be set up for intrusive service operations:

$$allow_{op}(o, d, e) = (contract_{id}(d) \in contracts(o)) \wedge idle(d) \tag{3}$$

Stating that the operation is allowed if the service contract for the device d is in the set of contracts the service organization o is working under, and d is currently idle.

5 Fulfillment of requirements

A summary of the requirements and the fulfillment levels with regards to ABAC is provided in Table 1. The fulfillment level *Fulfilled* denotes that ABAC is well suited to fulfill the requirement; *Possible* denotes that fulfillment is possible, but depends on the implementation; and *Unclear* denotes a requirement where the fulfillment level is difficult to assess from available documentation. In the following we discuss the reasoning behind the fulfillment assessment.

ABAC is able to express fine-grained rules due to the use of attributes on subjects, objects and the environment, as well as the possibility to set up policy-rules as functions of these attributes. This granularity and expressiveness will allow a very high level of flexibility, leading to fulfilling **requirement R5**. As illustrated in the Section 4, it seems possible to express rules in ABAC so that the principle of *least privilege* is satisfied, something that would be more challenging using e.g., RBAC. The **requirement R3** on diversity is also fulfilled, provided that policies can be easily added and adapted for different applications and user

ID	Requirement	Description	Fulfillment
R1	Availability	Work in spite of degraded functionality	Possible
R2	Critical Events	No HSE impact	Possible
R3	Diverse	Many user categories and usages of services and data	Fulfilled
R4	Scalable	Management of huge amount of devices, services, users	Unclear
R5	Flexible	AC must allow easy policy creation for new scenarios	Fulfilled
R6	Efficient	Cost of AC cannot impact system performance	Unclear
R7	Temporal policies	Quick shift in policies, due to customization	Fulfilled
R8	Logical Ordering	Workflow based access control	Unclear
R9	Transparency	Administrator to see what privileges are granted and why	Possible
R10	Delegation	Privileges transferable through delegation	Possible

Table 1: Requirements fulfillment for ABAC

categories. Here the enforcement and implementation considerations are of great importance.

The reasoning used for **R5** is also valid for **requirement R7**, as it arises as a result of quick shifts in policies, due to e.g. customization. Hence, it can be fulfilled since it is possible to express very fine-grained rules based on attributes. As demonstrated in the scenario description, it is possible to express policies so that they are meaningful in the context of shifting production schemes.

The management effort of an ABAC-model may not scale well with increasing size and complexity of the system (**requirement R4**). It may be the case that policy rules can be expressed in such a general way, as suggested in Section 4, but there are certainly more complex scenarios including a potentially larger set of rules. Any privilege request needs to evaluate all rules applicable for that specific request, demanding logic for handling combinations of rules. In a system with a complex set of policies, the implications of adding or altering a policy can be difficult to foresee. Attribute provisioning is also a management issue in a dynamic system. There is a need for trusted Attribute Authorities to provide the integrity of claimed attributes.

A low computational cost (**requirement R6**) is not a general property of ABAC. Depending on the implementation and how the policy base is formulated, the operation of granting or denying a privilege request may be computationally expensive. In case of using e.g., XCAML [18] for policy expression, there does not seem to be a bounded cost for inference [20, 29]. The total cost of inference must also include the time for attribute enumeration, which may need additional communication rounds with Attribute Authorities.

Requirement R1 implies that there should be a distributed architecture for access control in smart manufacturing applications, possibly including redundancy for key entities. This characteristic is uncommon in most available access control enforcement models. An ABAC architecture consist of several authorities, which all must be available to provide continuous privilege enforcement. However, it is possible to fulfill the requirement of a functioning access control mechanism during degraded mode using an enforcement architecture with local caches for attributes and policies that can be used in isolation. Another possibility is using a distributed architecture of policy- and attribute-authorities.

Requirement R2 concerns the possibility for a system to stop (e.g., operator lock-out during a critical scenario), and could possibly be met by ABAC using an environment attribute indicating a *system state* within the plant. This would however not be the first option for designing the system to protect it from HSE-related incidents. Instead, secondary control-units are typically used for essential functions, e.g., an emergency stop. Those controls are not dependent on standard user authentication and authorization, and will have a very limited functionality. Therefore, the fulfillment of this requirement can be seen as *possible*, even though it is not directly dependent on the access control model.

Requirement R8 is stating that the access policies should follow the workflow in the process. This is currently not supported by ABAC. There are mechanisms in e.g., NGAC and UCON [7] called obligations, which may alter privileges based on previous policy decisions. However, it is not clear if obligations can be used to describe a state-machine altering attribute assignments to mimic a process workflow.

A generic requirement on an access control model is to provide transparency (**requirement R9**). For ABAC it is unclear if such functionality is available neither with regards to an administrator, nor to a user. A solution on the implementation-model level could possibly be able to answer to the transparency needs of an administrator, but it is not intrinsic to the access control scheme, as in the case with e.g., the access control list (ACL) ability to perform per-resource review, or the RBAC ability to perform a per-subject review.

To be able to transfer privileges between subjects, as stipulated by **requirement R10**, is common during delegation in industrial systems. In the case of ABAC, this would require a subject to be able to transfer a set of attributes to another entity, as the privilege inference is based on attribute values. In principle there is nothing in ABAC that specifically prohibits this. However, it may prove a challenge in practice, as the subject needs to know precisely which attributes to transfer in order to achieve the intended privilege delegation. Detailed knowledge on how the policy-rules are expressed is needed to perform privilege delegation in ABAC. Looking at the examples from our use case scenario in Section 4, it would be quite easy to allow delegation by e.g., transfer the $contract_{id}$ attribute to a service engineer temporarily working with maintenance under a specific contract, but there are more complex scenarios in which several rules concurrently may influence a privilege decision. Furthermore, when transferring attributes there is a need to limit the usage of the attributes to the actual scope of the delegation, otherwise there is an apparent risk that the attribute transfer will grant other privileges than was intended. Our conclusion is that additional mechanisms in the enforcement and implementation layers are needed to make this requirement practically achievable.

6 Related Work

Salonikas et al. discuss the concept of access control requirements in a dynamic industrial system with focus on the wider concept of IIoT [16], while Lopez

et al. target cloud connected cyber physical systems [15]. Both articles discuss different access control models at the policy level, very similar to our work. However, these articles do not consider modular system characteristics specific for a smart manufacturing, as we do.

Watson et al. [6], discuss the use of different access control models in conjunction with OPC UA. They advocate the use of ABAC or a combination of ABAC and RBAC as a good match for protection against privilege escalation for both inside and outside attackers within IACS. Their work can be seen as a suggestion for the enforcement layer, whereas our work provides guidance applicable to the policy layer.

Some of the existing work present variations of ABAC suitable in different domains. Lang et al. [8] suggest a proximity based access control (PBAC), well suited for intelligent transportation systems. It originates from the ABAC model, but uses the mathematical proximity between subject and resource as one of the deciding factors for granting privileges. To derive policy rules, Model Driven Security (MDS) is used. MDS usually relies on a modeling tool in which the policy can be described at a high level of abstraction and the actual enforcement rules are then generated based on that model. Park and Sandhu [7] present the Usage CONtrol (UCON$_{ABC}$)-model, which can also be seen as an extension of the ABAC model with obligations. In this approach, an access-control event could alter attributes or conditions for future access controls. This mutability of attributes, or a variation thereof, could possibly be used to model the behavior of temporal workflows required by smart manufacturing. Next Generation Access Control (NGAC) [20] is the NIST proposal on how ABAC should be described. Compared to the traditional ABAC, in this variant attributes are provided as hierarchical labels (i.e., similar to RBAC group hierarchies), rather than properties with values as described in the initial ABAC-models. All of these approaches have interesting features useful in a smart manufacturing system, e.g., the model driven approach from PBAC and the obligations from UCON$_{ABC}$. As a future work, we aim to investigate the possibility to combine the beneficial concepts from these approaches in a practical smart manufacturing scenario.

7 Conclusions

Smart manufacturing is a technology that has a huge economical potential, transforming manufacturing towards servitization and extreme customization. However, the technologies that such systems are built upon bring new challenges, especially as the increasing attack surface expose the system to additional cybersecurity threats. As we have argued in this paper, one of the largely neglected mechanisms for security within manufacturing systems is access control between devices and services. Since the dynamic properties of smart manufacturing require a similarly dynamic model for access control, additional attention must be directed to this issue.

In this article we have derived a number of requirements on access control within smart manufacturing systems, based on knowledge related to traditional

manufacturing systems, interconnected cyber-physical systems, and industrial expertise. These requirements are considering both the guiding principles for access control and the basic safety principles of an industrial control system.

Illustrated by a use-case scenario we have mapped the requirements to the ABAC model, and shown that the model aligns well with the requirements. However, there are still several open questions to be answered. How to handle scalability with regards to management of policies and attributes in large systems seems to be the most difficult issue to deal with, especially for complex sets of access control policies. The management process must be sufficiently light-weight in order for the model to be adopted in real applications. Transparency and efficiency are other areas where additional efforts are needed to make the ABAC model a feasible alternative for modern industrial manufacturing systems.

As future research we envision conducting a simulation study with use-cases from the smart manufacturing domain, together with e.g., the Policy Machine, which is the reference implementation of NGAC from NIST[6]. The management issue of security policy generation could possibly be handled using model driven security, as discussed by Lang et al. [8].

References

1. S. Mittal, M. A. Khan, and T. Wuest, "Smart manufacturing: Characteristics and technologies," in *Product Lifecycle Management for Digital Transformation of Industries* (R. Harik, L. Rivest, A. Bernard, B. Eynard, and A. Bouras, eds.), (Cham), Springer International Publishing, 2016.
2. J. Davis, T. Edgar, J. Porter, J. Bernaden, and M. Sarli, "Smart manufacturing, manufacturing intelligence and demand-dynamic performance," *Computers and Chemical Engineering*, vol. 47, pp. 145–156, 2012.
3. K.-D. Thoben, S. Wiesner, and T. Wuest, ""Industrie 4.0" and Smart Manufacturing – A Review of Research Issues and Application Examples," *International Journal of Automation Technology*, 2017.
4. N. Tuptuk and S. Hailes, "Security of smart manufacturing systems," *Journal of Manufacturing Systems*, vol. 47, no. April, pp. 93–106, 2018.
5. J. Slowik, "Evolution of ICS Attacks and the Prospects for Future Disruptive Events," tech. rep., 2017.
6. V. Watson, J. Sassmannshausen, and K. Waedt, "Secure granular interoperability with OPC UA," in *INFORMATIK 2019: 50 Jahre Gesellschaft für Informatik – Informatik für Gesellschaft*, 2019.
7. J. Park and R. Sandhu, "The UCON$_{ABC}$ usage control model," *ACM Transactions on Information and System Security*, vol. 7, no. 1, 2004.
8. U. Lang and R. Schreiner, "Proximity-based access control (PBAC) using model-driven security," Springer Fachmedien Wiesbaden, 2015.
9. A.-R. Sadeghi, C. Wachsmann, and M. Waidner, "Security and privacy challenges in industrial internet of things," *The 52nd IEEE Annual Design Automation Conference*, 2015.
10. Y. Lu and F. Ju, "Smart Manufacturing Systems based on Cyber-physical Manufacturing Services (CPMS)," *IFAC-PapersOnLine*, 2017.

[6] `https://csrc.nist.gov/Projects/Policy-Machine`

11. M. Whitman and H. Mattord, *Principles of Information Security*. Cengage Learning, 4th ed., 2012.
12. J. Saltzer and M. Schroeder, "The Protection of Information in Computer Systems," in *IEEE*, no. 9, September 1975.
13. R. Sandhu, K. Ranganathan, and X. Zhang, "Secure information sharing enabled by trusted computing and PEI models," *Proc. of ACM Symposium on Information, Computer and Communications Security*, 2006.
14. R. S. Sandhu and P. Samarati, "Access control: Principles and Practice," *IEEE Communications Magazine*, no. September, 1994.
15. J. Lopez and J. E. Rubio, "Access control for cyber-physical systems interconnected to the cloud," *Computer Networks*, 2018.
16. S. Salonikias, A. Gouglidis, I. Mavridis, and D. Gritzalis, "Access control in the industrial internet of things," in *Security and Privacy Trends in the Industrial Internet of Things*, Springer International Publishing, 2019.
17. E. Yuan and J. Tong, "Attributed Based Access Control for web services," in *Proc. of IEEE Int. Conference on Web Services*, 2005.
18. "eXtensible Access Control Markup Language (XACML) version 3.0 plus errata 01," standard, OASIS, 2017.
19. V. C. Hu, D. Ferraiolo, R. Kuhn, A. Schnitzer, K. Sandlin, R. Miller, and K. Scarfone, "Guide to Attribute Based Access Control (ABAC) Definition and Considerations," tech. rep., NIST, 2014.
20. D. Ferraiolo, R. Chandramouli, R. Kuhn, and V. Hu, "Extensible Access Control Markup Language and Next Generation Access Control," 2016.
21. B. A. Kitchenham, "Procedures for Undertaking Systematic Reviews," tech. rep., Keele University, 2004.
22. B. Leander, "Towards an access control in a smart manufacturing context," tech. rep., Mälardalen Real-Time Research Centre, Mälardalen University, 2020.
23. "IEC 62443 security for ind. automation and control systems," standard, Internation Electrotechnical Commission, Geneva, CH, 2009-2018.
24. T. J. Williams, "The purdue enterprise reference architecture," *Computers in Industry*, vol. 24, no. 2, pp. 141 – 158, 1994.
25. J. Ladiges, A. Fay, T. Holm, U. Hempen, L. Urbas, M. Obst, and T. Albers, "Integration of modular process units into process control systems," *IEEE Transactions on Industry Applications*, vol. 54, 2018.
26. C. Faller and M. Höftmann, "Service-oriented communication model for cyber-physical-production-systems," *Procedia CIRP*, 2018.
27. I. Ayatollahi, J. Brier, B. Mörzinger, M. Heger, and F. Bleicher, "SOA on smart manufacturing utilities for identification, data access and control," *Procedia CIRP*, vol. 67, pp. 162–166, 2018.
28. "IEC 61449 function blocks," standard, Internation Electrotechnical Commission, Geneva, CH, 2012.
29. F. Turkmen and B. Crispo, "Performance evaluation of XACML PDP implementations," *ACM Conference on Computer and Communications Security*, 2008.