

# Assistance System to Support Troubleshooting of Complex Industrial Systems

Dorota Lang, Paul Wunderlich,  
Mario Heinz, Lukasz Wisniewski  
inIT - Institute Industrial IT  
OWL University of Applied Science  
32657 Lemgo, Germany  
{dorota.lang, paul.wunderlich, mario.heinz,  
lukasz.wisniewski}@hs-owl.de

Jürgen Jasperneite,  
Oliver Niggemann and Carsten Röcker  
Fraunhofer IOSB-INA  
32657 Lemgo, Germany  
{juergen.jasperneite, oliver.niggemann,  
carsten.roecker}@iosb-ina.fraunhofer.de

**Abstract**—In ever changing world, the industrial systems become more and more complex. Machine feedback in the form of alarms and notifications, due to its growing volume, becomes overwhelming for the operator. In addition, expectations in relation to system availability are growing as well. Therefore, there exists strong need for new solutions guaranteeing fast troubleshooting of problems that arise during system operation. The approach proposed in this study uses advantages of the Asset Administration Shell, machine learning, and human-machine interaction in order to create the assistance system which holistically addresses the issue of troubleshooting complex industrial systems.

## I. INTRODUCTION

In highly automated plants, availability of machines plays an increasingly important role. The reason for this is, inter alia, the increase in production, in competitiveness among producers, and in dependence between systems and components. Often downtime of one machine results in a halt or delay of the whole value adding process. Hence, the high financial losses. Therefore, in order to maximize the availability, new maintenance concepts are necessary to prevent or minimize unplanned operation interruptions by introducing appropriate maintenance measures or by reducing the repairing time in case of failure.

Despite wishful thinking, it is normal for an industrial system to face certain problems or failures which may result e.g. from wear of components or improper use. They are often indicated by the alarms or notifications. In most cases, the machine operator can deal with the issues by himself. However, some cases require a professional technician to take action which entails additional costs (price of the service) and downtime (time between failure and technician arrival). Moreover, the challenge grows with a higher level of system complexity. In more complex systems, the number of alarms and notifications grows and the time between their occurrences decreases. When the rate of alarms is higher than the response capability of an operator, it is regarded to be an alarm flood. The EN-IEC 62682 [1] standard defines an alarm flood as a 10-minute period with more than 10 alarms per operator. During the alarm flood it often happens that alarms influence each other, which means that one alarm can trigger the next

ones. In such cases, it is crucial to be able to track the root cause of multiple alarms and then eliminate them by resolving the root of the problem.

The goal of this research is to reduce the downtime and minimize (in some cases eliminate) the need for intervention of a professional technician by developing an assistance system that guides a machine operator through all the steps in the problem solving process. In case of a failure, the assistance system should retrieve all the necessary information about the problem and solution to the given problem from the Asset Administration Shell (AAS), which is a digital representation of an asset able to store and share data over the network [2]. In case of many alarms occurring within the short period of time (alarm flood), the assistance system should analyze the runtime data using machine learning (ML) algorithms, and based on previously learnt causal model deduce the possible root cause of the malfunctioning. Subsequently, proposed Uniform Software (US) uses the information provided by the AAS as well as by ML algorithms and presents the necessary corrective steps in a graphical understandable and comprehensive way. The US provides the information on many different devices simultaneously which can be used in accordance with the current needs of the system operator.

The rest of the paper is structured as follows. A description of the architecture of the assistance system is given in Section II. Section III presents the example of the use case scenario taken from the SmartFactoryOWL. While conclusions and future work are contained in Section IV.

## II. ASSISTANCE SYSTEM ARCHITECTURE

The assistance system consists of three tightly cooperating modules, namely Asset Administration Shell, Machine Learning, and Human-Machine Interaction. Each of them tackles different aspect of the system functionality.

### A. Asset Administration Shell

According to the vision of Industry 4.0 (I4.0), every tangible or intangible object that has value for an organization, e.g. machine, sensor or software, should be regarded as an Asset [2]. Moreover, every Asset in a company may have its own AAS,

which is an interface between a physical and a digital world. The combination of the Asset and the AAS constitutes an I4.0 Component [3]. The AAS creates a digital representation, so called digital twin [4], of the physical Asset by collecting and providing data about the Asset through its whole life cycle. These data can be dynamic e.g. a number of running hours, or static e.g. a date of production. One system can possess multiple AAS, for example one shell describing a type of the Asset, another one for a specific instance of the Asset [3]. On the other hand, one AAS can represent a whole system, and the others each administrated component of the system. Furthermore, the AAS can be stored directly at the administrated component or in a higher-level IT system. The location depends on computing and communication capabilities of the administrated Asset.

The structure and functioning of the AAS has been described in the German standard DIN SPEC 91345 [3]. According to it, the AAS consists of the Header and Body, which follow the standard IEC/TS 62832-1 [5]. The content of the AAS is managed by the Manifest and Component Manager. On the one hand, the Manifest acts as a clearly locatable table of contents of all data and functions. Moreover, it comprises globally unique IDs of the Asset and of the AAS, which enable unambiguous identification in the Internet of Things. On the other hand, the Component Manager manages the content of the Body and provides the access to this content on the basis of a service-oriented architecture. The Body itself is divided into separated submodels which contain information regarding different aspects of the Asset's functionality. These submodels can be a data model, a group of functions, or a reference to a nested I4.0 Component. They can be responsible e.g. for maintenance, condition monitoring, or troubleshooting. The maintenance submodel should store maintenance relevant information e.g. manuals provided by the manufacturer or URLs to a store with spare parts. The condition monitoring submodel should provide the information about e.g. maximum temperature recorded, number of running hours, etc. Finally, the troubleshooting submodel should contain all the relevant data required for overcoming problems in case of failure.

However, the current state-of-the-art does not provide any standardized definition of submodels. Therefore, one of the challenges of this research is to develop the AAS containing the troubleshooting submodel regardless of lack of standardization. Every time, when there will be an alarm pointing to a problem in a system, the information about the problem, all the steps leading to solving the problem, as well as further relevant information useful in such a situation, should be provided by the AAS.

## B. Machine Learning

In order to reduce downtime of machines it is necessary to support the operator in fast identifying the root cause of the disruption, which created the accumulation of alarms and notifications. Therefore, a method that assists the operator in identifying the root cause quickly and effectively has been

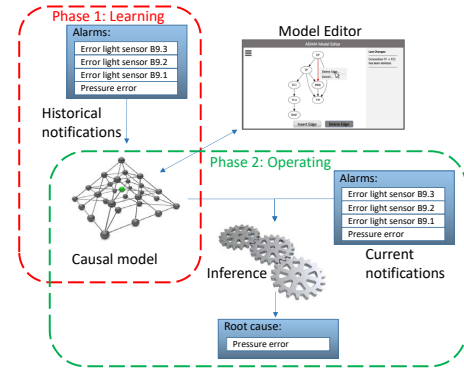


Fig. 1. The ML method for identifying the root cause of an alarm flood.

developed. It consists of two phases, namely the learning phase and the operating phase, see Fig. 1.

In the learning phase, the historical data about the system is used to learn a causal model of the alarms and their probabilities. The causal model is an abstract representation of the plant, which contains information about relations and dependencies of all alarms. Probabilistic graphical models are particularly suitable for mapping causal relationships. Therefore, Bayesian Networks were selected for representing a causal model. A Bayesian Network is a directed acyclic graph (DAG), whose nodes are associated with a set of variables  $X = \{X_1, \dots, X_n\}$  [6] which in this case represent the alarms. These alarms (nodes) are connected by edges, which represent dependencies. In order to be able to learn a precise causal model the historical data should consist of an adequate amount of observations. Moreover, the learning phase needs to be performed before the first use of the assistance system and after any major changes of the system, e.g. reconfiguration of implemented alarms. Depending on the quality and quantity of historical data, this process may take between few minutes to several hours. The learning of the causal model and of the associated probabilities is based on the previous work presented in [7].

Once the learning of the causal model and associated probabilities has been completed, the second so-called operating phase, can begin. The run time alarm data are used together with the causal model to conclude the root cause of the current alarm sequence. It is done by inferring in the Bayesian Network with the Likelihood Weighting (LW) algorithm [8], [9] given the currently active alarms as evidence. The inference with LW method allows the number of alarms to be reduced to the root causes and thus to only display the root causes to the operator. Therefore, the operator can focus on correcting the fault quickly and purposefully.

The learning phase provides the causal model in the form of XML-based format called GraphML [10] while the operational phase gives a table containing the root causes with their probabilities. Due to the fact that the current state of the art in ML does not allow for automatic construction of fully reliable causal model, the concept of the assistance system includes



Fig. 2. Structure of the assistance system

the model editor. The model editor is for the expert to be able to enter corrections of the causal model learnt with the ML method. The improvements are stored in the GraphML format and as a feedback integrated in the causal model for the inference.

### C. Human Machine Interaction

The proposed concept for the human-computer-interaction system consists of two parts, a model editor to manually adapt models learned by the machine learning system and a multi-device assistance system to support an operator during repair and maintenance processes.

The model editor (see Fig. 1 Model Editor) is a software tool which allows experts from a machine manufacturer to optimize models trained by the machine learning system presented in previous section. Therefore, it visualizes a learned model in the form of a two-dimensional graph with nodes representing the alarms and directed edges representing a learned relationship between two alarms. The software allows experts to insert or delete edges between two alarms based on their knowledge about the relations between certain parts in a specific plant.

The assistance system is intended to support operators during the execution of repair processes by providing potential root causes of an error and further information like repair instructions and runtime data. In recent years, various assistance systems for industrial applications have been developed and evaluated to support workers in the processing of various activities, but primarily for assembly tasks. However, these systems are generally based on single devices such as tablet-PCs, in-situ projectors or head-mounted displays [11], [12], [13]. Our system, on the other hand, relies on a combination of different devices to provide operators with an ideal opportunity to access relevant information for each situation without limiting the performance of their activities. In this way, the user can switch between several devices as needed to perform the repair process in an optimal way. Therefore, it will be necessary to synchronize the displayed content on the various devices to automatically get access to the same information. The required content to support a user, such as repair instructions, runtime data, information about previous repairs and a potential root cause is retrieved from the AAS of the respective machine and the ML system which can be accessed via a wireless connection when the service engineer

enters the surrounding of a plant. In addition, the exact position of the user can be determined via external sensors and markers in the environment in order to further improve the support. The collection and processing of information and the transmission to the different devices is done via a US which runs on a stationary PC at the respective machine. We are using a REST server, which keeps the data provided by the AAS and the ML system, as well as additional media content such as images and texts or positioning information from external sensors. The communication with the devices is carried out via a TCP/IP socket connection in a local Wi-Fi network.

### III. APPLICATION

This section presents a use case scenario which illustrates the benefits for the operator which come from use of the assistance system.

#### A. Set up description

For the use case scenario, whose structure is presented in Fig. 2, a folding machine placed in the SmartFactoryOWL [14] is used. The machine's task is to length fold, cross fold, sort, and stack different towels, which are fed by the operator. The sensor data used by the PLC is mirrored and provided through a bus coupler to the Raspberry Pi on which a CoDeSys environment is running. This solution allows for access to the data without disturbing the proper operation of the PLC. Additionally, the Raspberry Pi retrieves alarms from the SQL Server running at the machine's operating unit. The Raspberry Pi hosts the AAS which gathers the data from the machine and makes them available over the OPC UA to other I4.0 participants. OPC UA has been chosen as a communication protocol due to its service-oriented architecture required by the of AAS as well due to its platform and vendor independence. The other part of proposed assistance system is hosted by a PC. It consists of ML model generation by means of Bayesian Network, ML inference by means of LW algorithm, and HMI US which is responsible for providing and synchronizing the operator-relevant information on different devices such as tablet, mobile phone, and HoloLens.

#### B. Use Case Scenario

The use case scenario presents one of the possible problems appearing during the operation of the folding machine and the desired behavior of the assistance system.



Fig. 3. Human-Machine Interaction, Top: an operator using the assistance system with the HoloLens, Bottom: view of the operator with the HoloLens containing the repair instruction.

When the operator presses the start/stop switch the machine starts its operation and then stops. The assistance system receives the following alarms: 1. Pressure error, 2. Error light sensor B9.1, 3. Error light sensor B9.2, 4. Error light sensor B9.3, 5. Error light sensor B9.4. From the expert point of view, the machine starts its operation by carrying out a self-test and conveys out all items possibly present in the machine. However, due to lack of pressure the stackers are unable to move. Hence, the light sensors controlling their positions do not recognize any movement. Therefore, the alarms are raised for lack of pressure and for each of the stackers' light sensor. From the point of view of an inexperienced operator there are five problems which have to be solved and instead of fixing the root problem, the operator might remedy the last displayed error by cleaning or replacing the sensor.

With the assistance system it is possible to use the previously trained with Bayesian Network causal model in order to determine with the Likelihood Weighting inference method that the lack of pressure is the root cause of the alarm flood. This information is passed to the US, which presents the root cause to the operator. Moreover, it retrieves from the AAS additional information required for solving the problem. These are e.g. the current status of all sensors, secondary alarms which by default are hidden to the operator, information about the exact position of the pressure valve, pressure valve and sensor manuals, the frequency of occurrence of the problem, optionally the Internet link to the shop with broken parts and what is the most important the repairing procedure. The US uses the retrieved information for guiding the operator through the troubleshooting process. The necessary steps are presented on various devices (see Fig. 3) in a synchronized way and it's up to the operator which device is best to use in the current situation.

#### IV. CONCLUSION

In this study, a new concept for troubleshooting industrial systems has been proposed that combines concepts such as AAS, machine learning and HMI, in order to maximize the system availability. While the AAS collects and provides all

relevant data over the OPC UA, the ML reduces the alarm floods to the root cause and all the necessary information including the repairing steps are presented by the Uniform Software on appropriate to the situation devices in a synchronized manner. The main benefits of this solution is increase in system availability and decrease in need for a professional technician who solves problems. This entails an increase in revenue for the company using the assistance system.

For the future work, it is necessary to implement and test the concept on more complex systems and measure the benefits in a numerical way which allow for comparison with other solutions.

#### ACKNOWLEDGMENT

This work is funded by the German Federal Ministry of Education and Research (BMBF) for project ADIMA under grant number 13FH019PX5.

#### REFERENCES

- [1] International Electrotechnical Commission, "EN-IEC 62682:2014 Management of alarm systems for the process industries," International Electrotechnical Commission, Tech. Rep., 2014.
- [2] P. Adolphs *et al.*, "Structure of the administration shell. continuation of the development of the reference model for the industrie 4.0 component," German Federal Ministry for Economic Affairs and Energy (BMWi), Tech. Rep., April 2016.
- [3] "DIN SPEC 91345: Reference Architecture Model Industrie 4.0 (RAMI4.0)," DIN Std. DIN SPEC 91345, Standard, April 2016.
- [4] C. Wagner *et al.*, "The role of the industry 4.0 asset administration shell and the digital twin during the life cycle of a plant," in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sept 2017, pp. 1–8.
- [5] "IEC/TS 62832-1: Industrial-process measurement, control and automation. Digital factory framework. General principles," British Standards Institution, Standard, Jan 2017.
- [6] J. Pearl, "Bayesian networks: A model of self-activated memory for evidential reasoning," in *Proceedings of the 7th Conference of the Cognitive Science Society, 1985*, 1985, pp. 329–334.
- [7] P. Wunderlich and O. Niggemann, "Structure learning methods for Bayesian networks to reduce alarm floods by identifying the root cause," *22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2017)*, Sep 2017.
- [8] R. Fung and K.-C. Chang, "Weighing and integrating evidence for stochastic simulation in Bayesian networks," in *Machine Intelligence and Pattern Recognition*. Elsevier, 1990, vol. 10, pp. 209–219.
- [9] R. D. Shachter and M. A. Peot, "Simulation approaches to general probabilistic inference on belief networks," in *Machine Intelligence and Pattern Recognition*. Elsevier, 1990, vol. 10, pp. 221–231.
- [10] U. Brandes, M. Eiglsperger, I. Herman, M. Himsolt, and M. S. Marshall, "Graphml progress report structural layer proposal," in *International Symposium on Graph Drawing*. Springer, 2001, pp. 501–512.
- [11] H. Flatt, N. Koch, C. Röcker, A. Günter, and J. Jasperneite, "A context-aware assistance system for maintenance applications in smart factories based on augmented reality and indoor localization," in *Emerging Technologies & Factory Automation (ETFA), 2015 IEEE 20th Conference on*. IEEE, 2015, pp. 1–4.
- [12] S. Büttner, O. Sand, and C. Röcker, "Extending the design space in industrial manufacturing through mobile projection," in *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, ser. MobileHCI '15. New York, NY, USA: ACM, 2015, pp. 1130–1133.
- [13] S. Büttner, M. Funk, O. Sand, and C. Röcker, "Using head-mounted displays and in-situ projection for assistive systems: A comparison," in *Proceedings of the 9th ACM international conference on pervasive technologies related to assistive environments*. ACM, 2016, p. 44.
- [14] "SmartFactoryOWL." [Online]. Available: <https://www.smartfactory-owl.de>