An Industry 4.0-Ready Visual Analytics Model for Context-Aware Diagnosis in Smart Manufacturing

1st Lukas Kaupp
Department of Computer Science
Darmstadt University of Applied Sciences
Darmstadt, Germany
0000-0001-9411-6781

2nd Kawa Nazemi
Human-Computer Interaction & Visual Analytics
Darmstadt University of Applied Sciences
Darmstadt, Germany
0000-0002-2907-2740

3rd Bernhard Humm
Department of Computer Science
Darmstadt University of Applied Sciences
Darmstadt, Germany
0000-0001-7805-1981

***Abstract*—The integrated cyber-physical systems in Smart Manufacturing generate continuously vast amount of data. These complex data are difficult to assess and gather knowledge about the data. Tasks like fault detection and diagnosis are therewith difficult to solve. Visual Analytics mitigates complexity through the combined use of algorithms and visualization methods that allow to perceive information in a more accurate way. Thereby, reasoning relies more and more on the given situation within a smart manufacturing environment, namely the context. Current general Visual Analytics approaches only provide a vague definition of context. We introduce in this paper a model that specifies the context in Visual Analytics for Smart Manufacturing. Additionally, our model bridges the latest advances in research on Smart Manufacturing and Visual Analytics. We combine and summarize methodologies, algorithms and specifications of both vital research fields with our previous findings and fuse them together. As a result, we propose our novel industry 4.0-ready Visual Analytics model for context-aware diagnosis in Smart Manufacturing.**

***Index Terms*—Visual Analytics, Smart Manufacturing, Cyber-Physical Systems, Reasoning, Outlier Detection, Data Science.**

I. Introduction

Through the digital transformation, smart factories turned into highly complex interconnected environments, where analogous machines and cyber-physical systems exists side-by-side. Each cyber-physical system (CPS) consists of software and hardware modules [1] that produces large amounts of data. Observed outages, failures and misbehavior are hard to detect in the mass of raw data, which makes fault diagnosis or predictive maintenance difficult.

Visual Analytics (VA) can mitigate complexity and speed up problem solving in such complex environments through the interconnection between algorithms, visualizations and cognition. VA is an evolving research field, that combines the visualization with machine learning methods [2, 3]. The goal is to provide insights, that in return lead to a better supervision and deduct better decisions. Especially, big data analysis benefits from VA. Today the general model provided by Keim et al. [2] is broadly adopted, its broad applicability were shown [3] and it was meaningful extended over the recent decade [4, 5]. Furthermore, Sacha et al. [4] extended the general model by clarification of the knowledge generation process and human cognitive activities. To complement the work Andrienko et al. [5] recently concretize the outcome of the visual analytics process. Over the time the general model is well-defined [2–7], but a weak spot persists and it is not directly covered by previous work, the context. In VA the word context is a vague periphrasis, where the domain expert comes into play [4] and try to deduct new insights. The recent attempts towards context-awareness by Zhou et al. [8] and Wu et al. [9] emphasize the need for a generalizable VA-model for Smart Manufacturing (SM). To our best of knowledge, there exist no VA-model that defines the context and the upcoming diagnosis within the smart manufacturing domain. To overcome the circumstances we adhere our industry 4.0-ready VA-model for contex**t**-**a**ware diagn**o**sis **i**n **s**mart **m**anufacturing (TAOISM). Our TAOISM VA-model successfully couples VA and SM. In order to provide a novel model we specify each area of a VA-model. Therefore, we start with a definition of data within SM and list potential data-sources. In addition, we define a first set of models for context-aware diagnosis in SM. Especially, we provide a first draft of a context definition and propose an integration in a VA-model. We also name the requirements for visualizations in SM that arise from our model and provide a first draft of visualizations. To complete our model, we identify and define the main tasks in Smart Manufacturing that are significant for our model. In brief our main contribution is three-fold: (1) We provide an overview about current trends and tendencies for Visual Analytics and Smart Manufacturing; (2) we formalize, construct and compose the context-aware diagnosis in SM through our model and (3) we propose a model that combines VA and SM through the definition of context and the context creation process.

II. Related Work

General VA-models build the foundation of our model. Furthermore, first advances exist in Smart Manufacturing (SM) that try to contextualize SM-processes involving human perception or computational models. At the end, there will be multiple visualizations systems that cover similar steps to achieve similar goals in SM that have traces of an underling VA-model. We cover these systems to connect the traces and develop a concrete VA-model for SM.

978-1-7281-9134-8/20/$31.00 ©2020 IEEE
DOI 10.1109/IV51561.2020.00064

A. General Visual Analytic Models

Visual Analytics is a vibrant area of research and has been the foundation for the creation of multiple models over the past years [10, 11]. Keim et al. [2] with their model provoke one of the first general approaches, which was detailed later [3]. As an extension, Sacha et al. define the knowledge generation process [4] and human cognitive activities. Another well thought-out extension, given by Andrienko et al., who characterize the outcome of Keim's VA-model [5]. We use Keim's Visual Analytics model as a foundation, specify and refine it for the SM-domain. Additionally, we add a first draft of context definition and how the context can be used in the Smart Manufacturing domain. Context is rather vague term, that already was mentioned by Sacha et al. [4], where a domain expert is mandatory. As a consequence, contexts are enclosed by a domain, for our model in the SM-domain. Besides Sacha et al. [4], there are also data-driven approaches to enlighten the context term. Ceneda et al. [7] characterize guidance within a Visual Analytics to complement Keim's model. Munzner [12] provides a framework to specify tasks as a tuple of action and target. We use their framework to characterize our Visual Analytics model.

B. Context-awareness in Smart Manufacturing

A first step towards context-awareness was done by Emmanouilidis et al. [13]. Their conceptual model integrates the knowledge of domain experts as a single entity. Their ideas to contextualize machinery symptoms by integration of human perception are an inspiration and can be seen as an early predecessor of the proposed model in this paper. In addition to context-awareness Zhou et al. [8] define a novel situational awareness model incorporating qualitative (temperature sensor data) and quantitative (temperature zones with boundaries) measures. Their situational awareness model is split into an index part (rules for temperature zones) and a computational model, that utilizes the measurements; in their case temperature. In addition, the computational model deducts a value from multiple measurements, which represents the state of the production line (low, guarded, elevated, high, severe). To the best of our knowledge, that is the first model that formalizes a context in a SM-process and calculates the severity of that context. We adapt the principle to take not only the production line itself as the single source of information, but the surrounding variables as well. In contrast to Zhou et al. [8] that relies only on temperature data and cannot be applied in scenarios with complex multivariate data, we are aware of that situation and employ a data transformation step to even cover complex cases. Additionally, we leverage other production related systems (e.g. MES, ERP) to enhance the scope of context beyond the production line.

C. Visualization for Maintenance and Production

The SM-domain is difficult to assess due to concerns about security, rights of intellectual properties or data sovereignty. Nevertheless, Zhou et al. managed to publish a thorough survey of current visualizations in this domain [14]. They structured the visualizations using the concepts of creation and replacement. Visualizations in the replacement concept free people from dull work through implacement of intelligent devices (e.g. replacing monitoring personnel through online fault diagnosis) or virtualize dangerous work environments where people are able to learn the needed skills [14]. Creation encompasses the design phase (creation of products), the production phase (ideology to physical forms), the testing phase (guarantee established standards), the service phase (insights from usage) [14]. In order to gain insights in manufacturing data Xu et al. [15] combination of extended Marey-graph and station graph to exploit production flow and spatial awareness and provide insights to uncover anomalies. Jo et al. [16] provide an aggregated view of ongoing tasks in the production line with an extended Gantt-chart. Where Post et al. [17] use flow, workload and stacked graph to provide a user-guided visual analysis of a production line. Most impact for the proposed model had the work of Arbesser et al. [18] and Zhou et al. [8]. Arbesser et. al. developed a visual data quality assessment for time series data with integrated plausibility checks. Plausibility checks are simple rules that apply on given meta-information (e.g. sensor type, position) and observe for example out of range values. Thus, these are similar to our foreknown case models, which can be initialized before installation, on bases of manufacturers cyber physical system (CPS) documentation. Their well-thought overview contains information with different levels of granularity (from overview to detail) with employment of the checks to color the severity of the anomaly based on the checks. Wu et al. [9] set up a novel VA-pipeline to manually combine and pick features for the machine learning models and visualize the effectiveness in a training set view to act accordingly if results do not match observations. Additionally, they added a system overview with an extended theme river and a radial layout with a multifaceted pane for details on demand. Where Wu et al. preference is more manually configured, is our model more driven by automation protocols such as the de facto communication standard OPC-UA [19]. OPC-UA comes with machine models in place, which combine sensors to groups for different aspects of the CPS. The manual selection of individual sensor values for the generation of feature vectors is inefficient, our small learning smart factory already has about 17.000 values to choose from [20]. Additionally, the determination on machine learning can be an issues in hybrid scenarios, where the smart factory already has statistical models in place. Our model solves this issues by including also statistical models.

III. TAOISM VA-Model

Our proposed TAOISM VA-model (Figure 1) consists of four main layers (data, models, visualization, knowledge) and a cross-sector meta-layer (trigger). Originating in the established general VA-model by Keim et. al [2], depends each layer on another with multiple cross-sector relationships. All layers are involved to provide a context-aware diagnosis. Starting at the data layer, which acquires and transforms the data and provides information. The models layer observes the

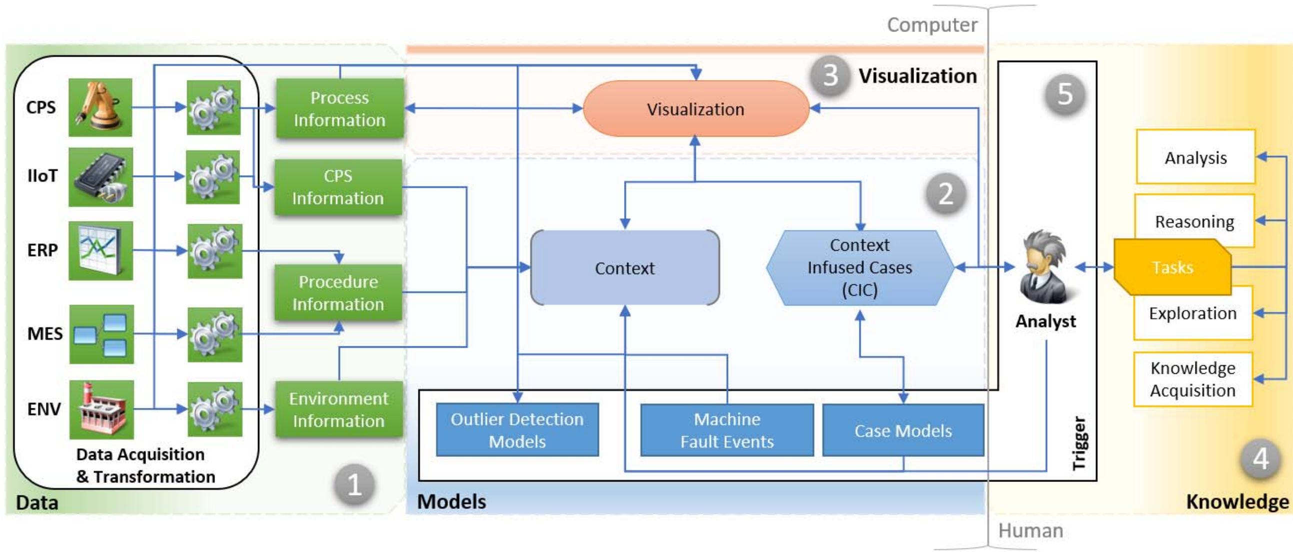

Fig. 1. Proposed contextualized TAOISM VA-model for Smart Manufacturing embedded in the established general VA-model [2]. The four main areas align with the general VA-model, extended with a new meta-layer. (1) Data encompasses all accessible information within a SM-environment. Visualized information sources should be seen as an example and may vary. (2) Models comprise either active models (triggers) or passively build models (context-related). Context Infused Cases (CIC) are generated by human, and stored and used actively again. (3) Visualization assists the users in their tasks and help to narrow down the underlying reason for an event. (4) Knowledge is where the users build hypotheses based on new insights, driven by their daily tasks. We also identified four main tasks within the process of concept creation [14]. (5) Finally, we extend the model with a cross-sector meta-layer (the trigger) used to combine all instances which can enforce a context creation build, such as machine entities (machine fault events, case models), observer models (outlier detection) or the user himself. Given instances are examples and may vary.

available information and triggers a context fetch (snapshot of the current situation) to construct *Context Infused Cases* (CICs). CICs are build with human perception by transforming presented information into a case with a description and a derived error model. Next, the visualization, which plays a crucial role in providing a meaningful representation of the data collected and supporting the tasks performed by the analyst, learning, exploring, analyzing and reasoning. Knowledge is the area, where the insights thrive new hypothesis and vice-versa. We identified four tasks, knowledge acquisition, exploration, analysis and reason as part of the process phase [14]. We use Munzner's Framework [12] to characterize our VA-model. A smart factory is a complex environment that yields vast amounts of data [1]. As a matter of fact, it is impossible to capture all data in real-time concerning e.g. low-bus speeds, where the actual process instructions and safety features has to be delivered as well. For this reason, we employ the asynchronous process of context creation. The process of context creation fetch the data in question one-by-one accordingly to a defined time frame without overloading the system. As a result, the analyst is able to reason over the current adjacent values and the current situation. We introduce the cross-sector meta-layer (trigger), where all instances are combined that are able to start the context creation process. Before we introduce each layer in detail, we name the involved entities, state the triggers, describe the human part in the equation and provide an example use case of error inference with our TAOISM VA-model. Our TAOISM VA-Model is based upon knowledge found through our various preliminary studies [1, 21].

A. Involved Entities

Multiple entities are involved in a SM-environment; tasks as error reasoning or predictive maintenance are complicated to fulfill in such an environment. Through the mass amount of log data [1] it is hard to gain insights and find the underlying fault for an occurring error. Schriegel et al. [22] have analyzed the standard industry 4.0 architecture with its entities on a technical level. We align with their actor definition, but use the bundle of sensors, PLCs and SCADA-Systems as synonym for a CPS. We also add the human as equal-worth entity and add IIoT as synonym for small cheap devices or device bundles within the smart factory. Our definition of actors is as follows:

- **CPS.** Cyber-physical systems are the source of process information. *Process Information* summarizes all information about the working process or the information which each machinery provides about the manufacturing process itself. This information may thrive also information about the health status of the machinery and its components or not (considered here as *CPS information*).
- **IIoT.** The Industrial Internet of Things encompasses devices that enhance current non-smart machines with the capabilities of a CPS. Alternatively, used as pre- and post-processing units to gather additional information about the production process and its CPSs. As well as the CPS, the IIoT devices deliver *Process Information* or *CPS information*.

- **ERP.** Enterprise Resource Planning software contains information about the overall production capacity and provide higher information in terms of planned output, real output, estimated revenue etc. These information become valuable in reasoning, where reasoning is implied to visualize a first estimation about the failure costs. The ERP system provides *Procedure Information*.
- **MES.** Manufacturing Execution System holds information about the overall manufacturing process. Which product is manufactured at which machinery and where the process will be continued, what steps are left for the finished product. These information are useful because of the visible chain of commands (manufacturing protocol) and the possibility of backtracking the relationships. Consequently, the MES also provides *Procedure Information*.
- **Environment.** The environment of the manufacturing process itself is getting not much attention by the research community yet. But in some cases, even in a smart factory, parts of the process chain may be fail. The injection modeling process is vulnerable to environmental influences, and they are recognized as uncontrollable factors [23–25]. Such information can be gathered through environmental sensors, providing *Environment Information*.
- **Human.** The personnel involved in the manufacturing process, such as operation, maintenance, repair or monitoring. The human provides information about certain situations, such as conditions, concerns or upcoming faults. With this situation awareness she ties the loosely coupled ends of information provided through the visualization.

Each of the involved entities provide information that are useful for a context-aware diagnosis. The sources of information may vary from the technical side from factory to factory and should be taken as an extensible list, with a bottom line for discussion.

B. Triggers

Triggers in the cross-sector meta-layer observe the provided information either automatically or manually and start the context creation process if the observations are subject to suspicion. We define four triggers:

- **Outlier Detection.** The unsupervised outlier detection, that consists of statistical and machine learning methods will give a hint on abnormal behavior through an event as previously proposed [1]. These models can be re-trained either by reinforcement learning or by annual, monthly or weekly wither manual or automatic updates to cover the latest developments within the surveyed CPSs.
- **Machine Fault Events.** In critic cases the machine itself will throw an event on faulty components.
- **Case Models.** A case contains known issues enriched with context information and a pre-built error model that activates the trigger for observation.
- **Human.** In rare cases, where none of the automatic triggers will be activated, the human can force a context creation process, if the observations look suspicious from his perspective. A typical situation is that quality assessment reported faulty products, while each system operates normal.

Each context creation process ends in a derived view of the current situation presented to the user. The visualization in multiple views presents the aggregated interconnected blended information in an understandable sustainable manner. Assisting the personnel in the situation.

C. The Human as Valuable Source of Expertise

The role of the human is vital and considered as human-interactive systems within cyber-physical systems [26]. Therefore, the concepts in Smart Manufacturing can be combined with Visual Analytics, where the human is also indispensable [2–7, 27]. The human interprets the presented information and draw a conclusion towards the goal of a task, e.g. take error inference (analysis task). The presented information encompass actual process information, paired with procedure information, CPS information and enriched with higher information coming from ERP- and MES-systems. Furthermore, the presented information is used to create a context infused case (CIC). We define a CIC as a case of abnormal behavior, that contain the annotated context information and where the human impose an error model through e.g. a rule-based approach to trigger the case. It differs from earlier introduced case models. The CIC is based upon a formerly unknown issue, incorporate more information about the state (context of the production process and a human-validated error model). Whereas case models are initialized with known problems of the machinery, that operates context-independent referring maintenance schedules or CPS documentation. These simple cases are supplied at the installation step of the machinery. Therefore, the CICs are the contextual opposite. As a result, CICs can be trained and learned in production compromising information relationships between on-site machinery and interactions to fuel the error model. In addition, these CICs are composable, learnable and transferable to other production systems, enabling a smarter start of similar production lines. As error inference is only one analysis task, we identified also knowledge acquisition, exploration and reasoning as primary tasks within a Smart Manufacturing production setup. For Munzner [12] a task can be separated in an action and target (goal). Countless combinations exists in Munzner's Framework e.g. between the analysis of a current situation and the investigation of all data, attributes, network data or spatial data using a defined search or query. The integration of Munzner's framework enables our model to be extensible for new tasks in the future. We emphasize the current task list as a first draft and may subject to change in further research.

D. Use-Case: The Error Inference Procedure

Error inference is quite difficult in vast amounts of industrial log data, as our previous work shows [1]. For that reason its an ideal use case scenario for our TAOISM VA-model. In a situation where the CPS reports a normal behavior and our employed outlier detection models observe some anomaly within the provided information. The observation can encompass a single source, or multiple sources, as well as multivariate high dimensional data. Next, the observation triggers the asynchronous context creation process. The process gather data from all available sources to provide contextual information around the suspicious observation. After the gathering, the analyst is given access to all the information in multiple views with different visualizations. Now, the analyst can self search, annotate and set boundaries for e.g. thresholds to him presented similarity and correlation analysis and highlight the data that led him to the decision which event may cause the issues. The output will be compressed within a CIC and an error model is created using all the features discovered by the domain expert, e.g. a rule set. Furthermore, the analyst can alter captured CICs, add additional information or rules or add more contextual information. In addition, the CICs can be composed to higher level CICs to react on more complex error situations. In the end, the CIC will be used to train new unsupervised outlier detection models or adjust current once in place. The error model of an CIC also leverage similarity and correlation measurements to also react in more noisy environments, where hard set thresholds are not sufficient. The CICs are also stored within the case models leading to more accuracy in terms of error detection over time.

IV. Data

Many sources for data exist for Smart Factories (see Section III-A). Schriegel et al. [22] structure the entities within an automation pyramid from a few instances such as ERP to many instances such as sensors. Applied to our previous definition, we change the pyramid to ERP, MES, CPS, IIoT and at the bottom sensors. Starting at the bottom line of the pyramid, the following section outlines the available data types in a smart factoring environments. For this reason, we released a first dataset [20] to provide a hint about the complexity of an Industry 4.0 Smart Factory. The data-set of the smart factory already consists of 17.464 columns and 11.455 rows. Each column, besides the first, correspond to an OPC-UA endpoint within one of many OPC-UA models. The data-set contains numeric and categorical values and include data structures like arrays. Simons et al. [28] describe the testbed in detail. The Smart Factory consists of a high-bay storage, a six-axis robot for assembly, a pneumatic press, an inspection unit proofing optical and weight parameters, an electrical inspection unit and everything is interconnected by a shuttle system. That testbed produces complex log data that is hard to analyze and consists only on pure *Process Information*. We conduct a study with more system data included such as data from ERP, MES, Environment and IIoT devices. The following example shows an excerpt of the released log data (see Table I).

The excerpt contains numeric and categorical values and include data structures like arrays. This further increases the complexity of possible values and makes a transformation step mandatory (as shown in Figure 1). An additional layer of complexity is that OPC-UA has 25 data types [29] that can be arranged in arrays, structures and unions, which can be also extended in the future. Stepping the pyramid upwards to CPS level, each CPS is typically bundled with an OPC-UA model, which is shipped with the machinery itself. The model is a list of references, which hold information about the integrated sensors and routines. Each reference can be subscribed to in order to retrieve changing values. A PLC as part of a CPS for example has access to multiple sensors and has software routines, which can also emit messages. In order to handle such variety of data types we already published some transformation steps [1, 21]. In order to access the Information, we follow these steps:

- Parse incoming data into a standardized format
- Use complex event processing (CEP) to infer higher level events
- Transform events and values to specific formats for the used algorithms

Especially, the last step is mandatory in order to utilize multiple algorithms, from rule-based approaches to neural networks. In our study [1] we had to transform incoming data to a numerical format. Furthermore, the challenge was to maintain the data characteristics within the new format. Multiple transformation strategies are necessary for multiple algorithms, that strategies differ based on the input types of the involved algorithms.

All strategies in common is the fact, that a transformation step needs to maintain the characteristics of a data-set. For that reason, our TAOISM VA-model add a transformation step (Figure 1) for each data source. The complexity of the information within the presented smart manufacturing environment add additional requirements towards the visualization, in terms of complexity reduction and information highlighting. Concluding this section, the data has to be transformed per algorithm that usage is planned.

TABLE I
EXCERPT OF THE SMART FACTORY DATA [20]

Columns	**Data**
Timestamp	2019-03-13 14:27:25.277000 - 2019-03-13 14:27:25.964000
[...].ACK_EF	False
[...].ActPosPercCtrlOutp	-0.2084[...]1067
[...].readWriteData	b'\x02\x00 [...]
[...].Betriebszustand	5
[...].RT_DATA .EXEC_BITS	[False, False, True, [...]]
[...].DB 333.textlist	b'Station 60 \x00index.html [...]'

V. Models

Our TAOISM VA-model consists of five models: outlier detection (Equation (4)), machine fault events (Equation (5)), case models (Equation (7)), context models (Equation (3)) and Context Infused Cases (CIC, Equation (8)). As denoted, the named models should be taken as an example and are subject to further research, additional models may be added in the

$$P = \{t \in T_O\}, T_O \subseteq T_A \tag{1}$$
$$W_n = \{t \in T_A \mid p \in P \wedge x \in X \wedge p_n - x \leq t \leq p_n + x\}, W_n \subseteq W, n \in [1, |P|] \subseteq \mathbb{N} \tag{2}$$
$$C_n = \{pi_m, ci_m, pri_m, e_m \mid pi \in PI \wedge ci \in CI \wedge pri \in PRI \wedge e \in Env \wedge \min(W_n) \leq m \leq \max(W_n)\}, C_n \subseteq C \tag{3}$$

future. The following section outlines the formal definition of each of our employed models. In addition, the formal definition helps to differentiate between the individual models and enables models, algorithms and methods that have already been published to be classified, assigned or segregated to our models.

We differentiate two types of models either active models, which cause a context creation, or passive models, which will be build automatically or with user interaction. Active models, that actively trigger a context creation are the outlier detection models, machine fault events and the case models.

The outlier detection models (OD) are a composition of different algorithms (A), classical approaches such as ARIMA [30] or neural networks such as autoencoder [31]. These methods can also be applied in ensembles to cover the weakness of one algorithm with the strength of an another [32, 33].

$$OD = \{a_1, a_2, ..., a_n \mid a \in A\} \tag{4}$$

Nowadays, most machinery have some sort of report system, that automatically reports on faulty components or throw events on incoming issues. Consequently, the machine fault events (MF) consist of multiple events (E) with a mapping function ($F(CO) : (co_1, .., co_n) \rightarrow E$), where components ($CO$) trigger the events directly.

$$MF = \{(e_1, f_1), (e_2, f_2), .., (e_n, f_n)) \mid e \in E \wedge f \in F\} \tag{5}$$

The machine fault event model cover those trivial cases and help the professional by starting the fetch of context related data for the visualization after an event is caught.

The case models (CM) build the bridge between the initial setup and the operation phase of the CPS. Meanwhile, the setup of the CPS standard cases (see Equation (7)) that where known upfront are employed in the case models. A basic case (CA) consists of an Error Model (EM) and a description (D).

$$CA = \{D, EM\} \tag{6}$$
$$CM = \{ca_1, ca_2, .., ca_n \mid ca \in CA\} \tag{7}$$

Those cases can encompass rule-based approaches that incorporate CPS-related logic. We already published a rule-based approach to fuse documentation and incoming machinery events [21].

The context model (C) (see Equation (3)) fuses process information (PI), CPS information (CI), procedure information (PRI) and environment information (Env) together utilizing one or more windows (W). The fuse points (P) are the timestamps of the outliers (T_O), which are a subset of all available timestamps (T_A). A window (W) spans around an outlier timestamp (P) and configurable range (X). Furthermore, the earliest and latest timestamp will than be used to build the context and fetch the data within the interval. As a result, the context can be seen as a current snapshot or joint of the situation, that will provide a fine-grained overview around a timestamp of a suspicious observation.

A CIC is a combination of both the context (C) and the case (CA), with additional historical information (H):

$$CIC = \{C, CA, H\} \tag{8}$$

The process information is part of the context (CO), the analyst (Figure 1) employs domain knowledge within the exploration task to find patterns that can be connected to cover the case and build an error model. Additionally, it is possible to add historical data to strengthen the error model and refine the trigger. After everything is in place the CIC is saved to the case models as an additional extended case. The now annotated information, current and historical data is also used for training of the different outlier detection algorithms, e.g. to also cover noisy cases.

VI. Visualization

We integrate multiple layers in our TAOISM VA-model (Figure 1) that have an effect on the visualization e.g. data or models. Each layer has several implications that result in requirements for the visualizations. The following section outlines the requirements for visualizations that intend to use our TAOISM VA-model. Additionally, we add a first draft of three user interfaces (overview, configuration, analysis), which arose from the given requirements. These drafts depend on each other and are ordered from overview to detail to provide broad first hint about the impact between the relationships of the requirements. The key requirements that emerge from our model are:

- **Unified System Integration (1).** Through the extendable automated transformation process, we provide a way to interconnect and integrate new system components (e.g. sensors). The visualization has to be extendable and universal to adapt to new elements and provide an overview about the involved systems (CPSs, Env, ERP, MES etc.).
- **Configurability (2).** Our model is configurable. We put the analyst in charge to change the transformation process and the models (outlier detection, case, machine events). Additionally, the analyst can limit the information provided by the different visualizations. Consequently, the

visualization has to allow a modular configuration of each component within the model and provide an ability to slice the granularity of information.

- **Surveillance of Mass Information (3).** We do not hide information in our model. We employ ways to provide the analyst with distilled information with details on demand. As a result, the visualization has to implement intelligent aggregation strategies to cope with the vast amount of information.
- **Highlight important Information (4).** Our visualizations highlight outlier and use context information to project and compose different data sources to provide deeper insights. Therefore, the visualization must contain several ways to highlight information without overwhelming and interrupting human perception.
- **Context Creation and User Integration (5).** Upon event notification the build context information are automatically provided and visualized. The analyst is able to compose problem related data, enrich the data by rules and infuse historically data to enhance old cases or create new cases. For this reason, the visualization has to allow the user to mark important information and additionally chain the found information and multiple observations together to generate new error models.

The first overview draft (Figure 2) shows a fusion of different hard- and software models and is part of the unified system integration (1). A smart manufacturing production line consists of different CPSs and data sources (Section IV). We are able to acquire a lot of data through OPC-UA and other manufacturing protocols. OPC-UA machine models provide information about all the available data from process data to a single sensor. Each available model is used to automatically generate an overview about the production line (Figure 2). The draft in Figure 2 shows the output of the generation process. Currently, we are in a transition between loosely coupled interconnected production plants towards the fully-automated smart factory. In this hybrid state there is also the need to add production plants manually to the shown graph in Figure 2. For this reason, the overview is editable.

The generation process of the overview draft is started after one or more OPC-UA endpoints are added through the UI or automatically found due to used standard OPC-UA ports (spared to save space). Each found model is drawn as a orange square, containing their OPC-UA namespace in order to cover its name. An ontology is used to find an icon fitting the name for each model. The icons are interchangeable and are in place in favor of distinctness and perception. An icon together with a name is more recognizable as a single string. In addition, the visual layout supports the perception of the process sequence. The more advanced part of the visualization is the automatic information flow annotation. In order to get the information all models will be subscribed to and through the presence and absence of activity within the models sensors the directed graph is computed and added to the visualization. Every automatism has its flaws, so the user can change the

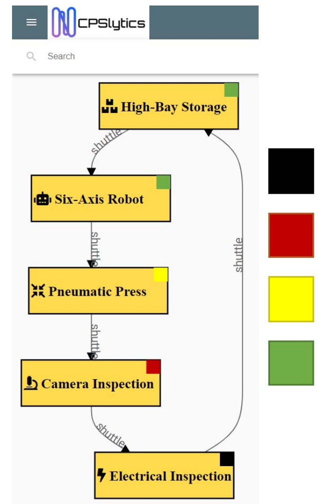

Fig. 2. Generated overview based on the incoming Information, validated using profound OPC-UA models (left). Additionally, the current severity levels (right).

automatically build graph, afterwards. Each visualized node has an additional field in right upper corner. Here, the current health status of the plant is visualized, the health status of the machine aggregates the different information sources to its highest level. That square creates a space for a visual placement of the results through a context algorithm, such as the situation value formula invented by Zhou et al. [8]. In our model, the status contains information about current machine load, process flow, outlier models and prediction models, case models and machine fault events. Each event occur in different severity levels, visualized from green (everything ok) to black (faulty machine).

Figure 2 shows also the different severity levels. Thereby, green indicates a normal running system, yellow stands for most severe situation and red is completely wrong if nothing is changed. The scale ends with black if the production line is forced to pause and the worst case situation has occurred e.g. a machine fault. The elements within the overview are movable to ensure that it is adjustable to visually map the outline of the production line into the dashboard.

Next, we propose a visualization draft for the configurability (2) of the different transformation steps and the configuration of different algorithms. We name our approach visual orchestration of methods, because we align different transformation steps with parametrization of different data models (outlier detection, case, machine fault events). Figure 3 shows the configuration dashboard. The view is split into two areas, the available nodes and the configuration view. Each available node can be used to create a configuration graph. All found sensors or different layers of machine abstraction delivered through OPC-UA can be selected in a source node. The views (Figure 2, Figure 3) depend heavily on each other. Every time a node is added either on the overview page or on the configuration layer the models are updated. After a source

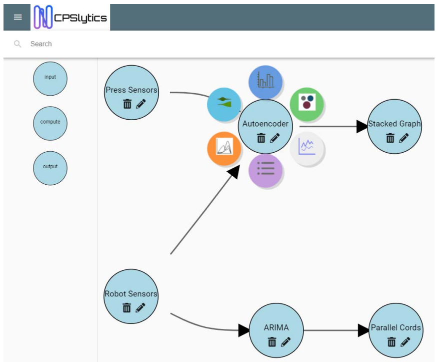

Fig. 3. Visual orchestration of methods: Dashboard configuration view

node is added, one of the information sources Section IV can be selected. Multiple nodes can be joined together within a transformation node. A menu emerges on a node hover (Figure 3) to visualize the output of the given node e.g. the transformation step, to either inspect or explore the transformed data. As shown, different visualization techniques are available to the user (selected node is colored orange). A click on edit, enables the analyst to tweak the parameters for actual transformation steps.

It is shown that e.g. each outlier detection model has its own node, where again the parameters of the given model can be tweaked through the UI and the domain expert. Each node is mapped directly to an output node which in return results in a new visualization in the downstream visualization views of our model. Multiple outputs of different transformation steps or models can be joined again to operate e.g. on a preprocessed data stream. Some used algorithms may process massive amounts of data slower, which would delay the analysis. A reduced data stream resolves such a bottleneck. Each configured pipeline can be stored by a name. That enables the analysis of different user-created pipelines. Clustering between simple and advanced users or domain experts is examined in a future study. The study will encompass if AI models are able to assist the basic user to build an optimal pipeline for a given problem. As a result, an automated prefabrication of a pipeline based on user feedback may be possible in the future.

The last visualization draft is the system performance overview (Figure 4), which serves as gateway to observe mass information (3), highlight important information (4) and be the port for context creation and user interaction (5). Each output element of the previous visualization layer is given its own visualization, where the output is visualized e.g. as a graph. Hereby, the visualization is based on both, rules and prediction.

The estimation of the current trend or value range is visualized as bright blue and for each point a min and max value is shown, leading to an advanced river chart. Reported outliers are represented by circles and the color reflects its severity or its probability e.g. in case of a neural network. Cases that are triggered are visualized with a plus, again colored to their severity. Furthermore, the context is visualized on a click on either a case or an outlier. On the downstream view (ditched in favor of the other views) the analyst can explore the underlying context and mark suspicious observations. For this reason the analyst can mark areas on the graph with a plus for a case or a circle to explicit save the area for the training process of one of the employed supervised outlier detection algorithms. Multiple pluses and circles can be connected to create fuzzy rule chains to trigger the case. In addition, the analyst describes each case and set a proper name. All data, including the context, is stored in a database as an annotated corpus for the unsupervised outlier detection algorithms. The database leads to an adoptable system that becomes better over time on the detection of different quirks of the CPS. As a result of a centralized storage, the database, with stored contexts and cases can be interchanged with other production lines with similar machines. In return, a database migration may provide a way to integrate a new production line in the future.

VII. Knowledge

Knowledge is used ambiguously, therefore we define this term in this section. We use Munzner's work [12] as foundation for formalizing and characterizing tasks. Munzner formalizes tasks as action and target [12]. Accordingly, we classify for our model within SM as an action (A) and a target (Tr):

$$T = (A, Tr) \tag{9}$$

These actions are used to analyze, search or query data. This can be done for all data, attributes, network or spatial data in order to achieve different goals such as trend or outlier detection. The formal definition (Equation (9)) will lead in a future work to a fine granular task definition. In the meanwhile, we use the work of Zhou et al. [8] on visualizations in SM to illustrate the most wide-spreaded tasks and include them in our model. For this, we identified four main tasks for our model:

- **Knowledge Acquisition.** Knowledge acquisition is the task, where the user gets familiar with the SM-environment. It is mandatory to acquire an understanding of the complex CPSs to handle upcoming incidents. The

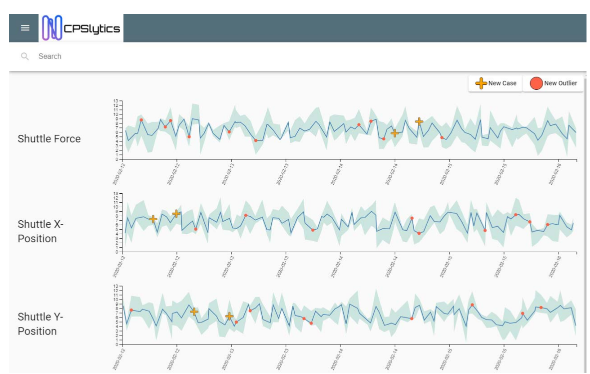

Fig. 4. Distilled system performance zoomed overview backed by predictions with marked outliers and triggered cases.

task is essential for trainee programs, advisory or recovery from a crisis to mitigate losses in domain knowledge through employee exchange. We assist the knowledge acquisition due our prefabricated case models with descriptions and different levels of information granularity to assess the systems and its internals.

- **Exploration.** Exploration is defined as a task after the user gets familiar with the system through knowledge acquisition. The process of exploration leads to new defined insights and correlations without a certain predefined task in the user's mind. Therewith is a self-driven knowledge acquisition process without a main goal. Exploration is part of the day-to-day work in form of surveillance and maintenance of the production line. The exploration process is mainly supported through our interactive visualizations.
- **Analysis.** Analysis is the process of investigating deeper insights in particular to detect a certain pattern or solve a complex analytical task, e.g. detecting and solving an outlier. We enable the analysts to investigate outliers and search the system for possible correlations. Outlier Detection and Predictive Maintenance are part of the analysis task. Already acquired knowledge is used to examine the system to search for possible reasons or causing effects.
- **Reasoning.** Finally, the reasoning task is the inference of a reason for a specific issue. It builds upon the acquired knowledge and vast exploration during an analysis. We support the analysts to reason outliers providing the tooling for recording, annotation and interaction with the data and the process throughout the visualization. The CIC creation and a Fault Recovery is part of the reasoning task. Our proposed VA-model provides the tool chain to put data into a perspective.

The Knowledge Acquisition tasks is a prerequisite to the other task in order. CPSs are complex interconnected systems with networks and process schedules. Furthermore, the knowledge about the systems, system architecture and common events has to be acquired upfront, in order to use exploration, analysis and reasoning. Our model provides the tools to assess the system simplified. In addition, the case models database provides the opportunity to browse through common already known events. A trainee, professional or analyst that want to get familiar with the CPSs starts with browsing the delivered data. They want to get to know the topology of the CPSs and the paths of the information flow. We introduced a general process overview (Figure 2) and the configuration dashboard (Figure 3) to visualize the information flow in the system. In addition, information are delivered in different zoom levels (Figure 4). Figure 4 is also an example, how to compare between sensor values and summarize events according to each case or outlier. In the exploration task, the users are already familiar with the CPSs and the user interfaces. They want to discover more aspects of the production process. Meanwhile, the exploration tasks the analyst wants to discover new insights through exploration. The case models (Figure 1) provide information in a centralized point. The search field (Figure 2) help to query the database of recent or historically known cases. If available, all cases are delivered with more information such as contexts (process information, procedure information etc.). The provided information is visualized in already known views (detail view, Figure 4), in order to keep the user's perception consistent. Now, after a basic understanding of the production process is established, attributes are additionally in focus. The analyst is able to survey different distortions, extremes or similarities (Figure 4). The obtained knowledge is than utilized in the analysis task. An analysis of an outlier or a new fault is a complex task. Our model supports this tasks by providing the visualizations to annotate, record, derive data in order to locate or identify information that are useful in that scenario. In Section VI we describe an analysis scenario. The analysts can review data and annotate outliers or cases to the found suspicious observations. Furthermore, the saved context and its values are transferred to the case models database. Different kinds of algorithms derive the data in order to observe e.g. minima or maxima. Additionally, estimations are also visualized in the performance overview (Figure 4). The analyst is also enabled to configure each algorithm for the analysis task (Figure 3). Mature processes can be edited and altered to get extended and varied to match new conditions of historic events. As a result, the analyst is capable of the final task, the reasoning. Besides the analysis tasks, the reasoning task involves more data. New insights and hypothesis have to be proven, in order to get the underlying reason. Hawkins postulates a definition that an outlier is "an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism" [34, p. 1]. The reasoning task tries to find these different mechanisms and to provide a kind of rule that can be recognized. No matter whether it is just an outlier, a machine event or a faulty component.

VIII. Conclusion

In this paper we presented an industry 4.0-ready VA-model for context-aware diagnosis in Smart Manufacturing which we call TAOISM. We combined and summarized methodologies, algorithms and specification to form a novel model. Additionally, we gave a first draft of a context-definition for Visual Analytics in Smart Manufacturing. Equally important, we specify the process of context creation and gave an example how the context can be handled and used within a Visual Analytics system. Moreover, we specified a list of possible information sources and models for VA. Our model is also industry 4.0-ready, because we utilize the latest smart factory protocol OPC-UA and with our already published transformation strategy [1] we are able to support more protocols in the future. Furthermore, we derived a first set of requirements for visualizations in Smart Manufacturing and provided visualization drafts, which implement the given requirements. Finally, we identified four main tasks and classified them under utilization of Munzner's Framework [12]. Which is to the best of our knowledge a novelty and firstly used for knowledge task classification within a VA-model. Further, our TAOISM VA-model open opportunities for additional research. We plan for further studies on transformation strategies to cover additional data sources for SM. A current limitation is the focus on OPC-UA only for the overview generation process, other protocols will be covered in future studies. A plan is conducted to develop and implement different algorithms for the overview generation process. Multiple studies are planned for our visual orchestration of methods. The visual orchestration of methods carry certain opportunities. A study was conducted to analyze basic, advanced and professional users and their build pipelines to build an AI to assist inexperienced users in the building procedure for given problem. An automatic approach to pipeline building could be possible as well to automatically build a processing pipeline for a new observation based on veteran feedback. A study is planned to examine this opportunity.

To conclude our work, we provided a novel VA-model for context-aware diagnosis in Smart Manufacturing. It should be seen as a living model open to future research. We proposed a first revision of our model based on our previous findings.

Acknowledgments

This work was partially funded by the Hessen State Ministry for Higher Education, Research and the Arts within the program "Forschung für die Praxis" and was conducted within the research group on Human-Computer Interaction and Visual Analytics (https://vis.h-da.de). The presentation of this work was supported by the Research Center for Applied Informatics. We want to thank Prof. Dr. Stephan Simons for the permission to use the learning smart factory of the Darmstadt University.

References

[1] L. Kaupp, U. Beez *et al.*, "Outlier detection in temporal spatial log data using autoencoder for industry 4.0," in *Engineering applications of neural networks*, ser. Communications in Computer and Information Science, J. Macintyre, L. Iliadis *et al.*, Eds. Cham: Springer, 2019, vol. 1000, pp. 55–65.

[2] D. Keim, G. Andrienko *et al.*, "Visual analytics: Definition, process, and challenges," in *Information Visualization*, ser. Lecture Notes in Computer Science, D. Hutchison, J.-D. Fekete *et al.*, Eds. Berlin and Heidelberg: Springer, 2008, vol. 4950, pp. 154–175.

[3] D. Keim, Ed., *Mastering the information age: Solving problems with visual analytics*. Goslar: Eurographics Association, 2010.

[4] D. Sacha, A. Stoffel *et al.*, "Knowledge generation model for visual analytics," *IEEE transactions on visualization and computer graphics*, vol. 20, no. 12, pp. 1604–1613, 2014.

[5] N. Andrienko, T. Lammarsch *et al.*, "Viewing visual analytics as model building," *Computer Graphics Forum*, vol. 37, no. 6, pp. 275–299, 2018.

[6] D. Sacha, H. Senaratne *et al.*, "The role of uncertainty, awareness, and trust in visual analytics," *IEEE transactions on visualization and computer graphics*, vol. 22, no. 1, pp. 240–249, 2016.

[7] D. Ceneda, T. Gschwandtner *et al.*, "Characterizing guidance in visual analytics," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 1, pp. 111–120, 2017.

[8] F. Zhou, X. Lin *et al.*, "Visually enhanced situation awareness for complex manufacturing facility monitoring in smart factories," *Journal of Visual Languages & Computing*, vol. 44, pp. 58–69, 2018.

[9] W. Wu, Y. Zheng *et al.*, "A visual analytics approach for equipment condition monitoring in smart factories of process industry," in *2018 IEEE Pacific Visualization Symposium*. Piscataway, NJ: IEEE, 2018, pp. 140–149.

[10] K. Nazemi, M. Breyer *et al.*, "SemaVis: A new approach for visualizing semantic information," in *Towards the Internet of Services: The THESEUS Research Program*. Springer International Publishing, 2014, pp. 191–202.

[11] K. Nazemi, *Adaptive Semantics Visualization*. Springer International Publishing, 2016.

[12] T. Munzner, *Visualization analysis & design*, ser. AK Peters Visualization series. Boca Raton, FL: CRC Press, 2015.

[13] C. Emmanouilidis, P. Pistofidis *et al.*, "Context-based and human-centred information fusion in diagnostics," *IFAC-PapersOnLine*, vol. 49, no. 28, pp. 220–225, 2016.

[14] F. Zhou, X. Lin *et al.*, "A survey of visualization for smart manufacturing," *Journal of Visualization*, vol. 22, no. 2, pp. 419–435, 2019.

[15] P. Xu, H. Mei *et al.*, "Vidx: Visual diagnostics of assembly line performance in smart factories," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 291–300, 2017.

[16] J. Jo, J. Huh *et al.*, "Livegantt: Interactively visualizing a large manufacturing schedule," *IEEE transactions on visualization and computer graphics*, vol. 20, no. 12, pp. 2329–2338, 2014.

[17] T. Post, R. Ilsen *et al.*, "User-guided visual analysis of cyber-physical production systems," *Journal of Computing and Information Science in Engineering*, vol. 17, no. 2, p. 9, 2017.

[18] C. Arbesser, F. Spechtenhauser *et al.*, "Visplause: Visual data quality assessment of many time series using plausibility checks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 641–650, 2017.

[19] D. Bruckner, M.-P. Stanica *et al.*, "An introduction to opc ua tsn for industrial communication systems," *Proceedings of the IEEE*, vol. 107, no. 6, pp. 1121–1131, 2019.

[20] L. Kaupp, B. G. Humm *et al.*, "Raw opc-ua dataset of a working industry 4.0 smart factory (h-da autfab)," doi: 10.5281/ZENODO.3709619. 2020.

[21] U. Beez, L. Kaupp *et al.*, "Context-aware documentation in the smart factory," in *Semantic applications*, T. Hoppe, B. Humm, and A. Reibold, Eds. Berlin: Springer, 2018, vol. 23, pp. 163–180.

[22] S. Schriegel, T. Kobzan, and J. Jasperneite, "Investigation on a distributed sdn control plane architecture for heterogeneous time sensitive networks," in *WFCS 2018*. Piscataway, NJ: IEEE, 2018, pp. 1–10.

[23] Pledger and M. J, "Utilising measurable uncontrollable factors in parameter design to optimise the response," 1998.

[24] Shu-Guang He, Li Li and Er-Shi Qi, *Study on the Quality Improvement of Injection Molding in LED Packaging Processes based on DOE and Data Mining: WiCom 2007 ; 21 - 25 Sept. 2007, [Shanghai, China*. Piscataway, NJ: IEEE Operations Center, 2007.

[25] V. Modrak and J. Mandulak, "Exploration of impact of technological parameters on surface gloss of plastic parts," *Procedia CIRP*, vol. 12, pp. 504–509, 2013.

[26] NIST, "Foundations for innovation in cyber-physical systems: Workshop report," 2013.

[27] N. Andrienko and G. Andrienko, "A visual analytics framework for spatio-temporal analysis and modelling," *Data Mining and Knowledge Discovery*, vol. 27, no. 1, pp. 55–83, 2013.

[28] S. Simons, P. Abé, and S. Neser, "Learning in the autfab – the fully automated industrie 4.0 learning factory of the university of applied sciences darmstadt," *Procedia Manufacturing*, vol. 9, pp. 81–88, 2017.

[29] OPC Foundation, "Opc-ua datatype xml schema file," 2020. [Online]. Available: https://opcfoundation.org/UA/schemas/1.04/Opc.Ua.Types.bsd

[30] G. E. P. Box and D. A. Pierce, "Distribution of residual autocorrelations in autoregressive-integrated moving average time series models," *Journal of the American Statistical Association*, vol. 65, no. 332, pp. 1509–1526, 1970.

[31] G. E. Hinton and R. S. Zemel, "Autoencoders, minimum description length and helmholtz free energy," in *Proceedings of NIPS'93*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc, 1993, pp. 3–10.

[32] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *Journal of Artificial Intelligence Research*, vol. 11, pp. 169–198, 1999.

[33] J. Chen, S. Sathe *et al.*, "Outlier detection with autoencoder ensembles," in *Proceedings of the 2017 SIAM International Conference on Data Mining*, N. Chawla and W. Wang, Eds. [S.l.]: Society for Industrial and Applied Mathematics, 2017, pp. 90–98.

[34] D. M. Hawkins, "Introduction," in *Identification of Outliers*, ser. Monographs on Applied Probability and Statistics, D. M. Hawkins, Ed. Dordrecht: Springer, 1980, pp. 1–12.