Improving Problem Resolving on the Shop Floor by Context-Aware Decision Information Packages

Eva Hoos[1,3], Pascal Hirmer[2], and Bernhard Mitschang[2,3]

[1] Daimler AG, 71034 Böblingen, Germany
Eva.Hoos@daimler.com
[2] Institute of Parallel and Distributed Systems,
University of Stuttgart, 70569 Stuttgart, Germany
firstname.lastname@ipvs.uni-stuttgart.de
[3] Graduate School of Excellence Advanced Manufacturing Engineering,
University of Stuttgart, 70569 Stuttgart, Germany

Abstract. Industry 4.0 with its accompanying trends, such as cyber physical systems and internet of things, leads to new approaches like self-organization of production environments. However, the human actor still is needed and beneficial for decision-making in the product lifecycle. For the decision-making process, it is crucial to provide situation-specific data. To address this issue, we introduce decision information packages (DIP), which compose relevant engineering data for a specific context. We analyze the requirements and domain-specific challenges for defining and implementing DIPs by investigating a real-world use case scenario at a German car manufacturer. On this basis, we develop a first approach for DIPs which comprises a context-aware engineering data model as basis to compose them.

Keywords: Industry 4.0, Context-awareness, Data Provisioning, Manufacturing, Engineering

1 Introduction

Despite Industrie 4.0 and its accompanying data-driven techniques like cyber physical systems and internet of things [10,14], human interaction and decision making is still mandatory and beneficial [7,15]. This is especially true in a pre-production plant where the first prototypes of a product are manufactured. Since the product and its manufacturing processes are in development, failures can likely occur and the shop floor worker has to decide how to resolve them. For an effective decision making, the context and other information have to be investigated. We call this set of required information a *decision information package* (DIP). As of today, whenever a problem occurs, the corresponding DIP is not automatically available. Instead, the decision maker is mostly concerned with searching and accessing this data distributed onto heterogeneous engineering

X. Franch, J. Ralyté, R. Matulevičius, C. Salinesi, and R. Wieringa (Eds.):
CAiSE 2017 Forum and Doctoral Consortium Papers, pp. 121-128, 2017.

IT systems. However, the decision maker, i.e. the shop floor worker, is trained to solve the problem and not to find the required information, which requires comprehensive IT skills. Our goal is to provide the necessary decision support via automatically composed DIPs in order to relief the decision maker from browsing complex IT systems and to redirect his focus on domain-specific problem-solving.

Decision information packages compose relevant data from heterogeneous engineering data sources specific to the context and the problem that occurred. According to Dey [6], context *"is any information that can be used to characterize the situation of an entity"*. For example, shop floor workers represent the entities and they are dependent on their current task, location, and of the state of the environment. In order to use context to filter *relevant* (i.e., situation-specific) data, context data has to be linked to engineering data. Furthermore, the automated data acquisition from engineering data sources requires knowledge about the IT landscape and their demands. For example, the IT landscape is composed of heterogeneous IT systems that constantly have to be integrated [11–13].

In summary, the challenges of composing and providing DIPs are (i) identification of influences of the engineering domain, (ii) linking engineering data with context to achieve context-aware provisioning of data, and (iii) automated access to and packaging of data. However, existing approaches either focus on filtering relevant data without considering domain-specific engineering challenges [3, 5] or they exclusively focus on the provisioning of engineering data without filtering relevant data specific to the context [11,13]. To address these issues, we develop an approach to define and automatically provision DIPs. The contribution of this paper is: (i) analysis of requirements for context-aware DIPs based on a real-world use case scenario at a German car manufacturer, and (ii) a meta-model to specify DIPs by linking context data and engineering data.

This paper is structured as follows: Section 2 introduces a real-world use case scenario and derives requirements for our approach. Section 3 contains the main contribution of our paper: an approach for the composition and provisioning of DIPs. Finally, Sect. 4 covers related work and Sect. 5 summarizes the paper.

2 Use Case Scenario and Requirements

This section introduces a use case scenario, which is used to derive requirements for our approach for the composition and provisioning of DIPs.

2.1 Engineering Use Case Scenario: Prototype Factory

We conducted a case analysis at a German car manufacturer in the engineering domain. On this basis, we conceive a use case scenario which emphasizes the demand for composition of DIPs. The use case scenario is part of the manufacturing of car prototypes, also known as *pre-production test.* The engineering domain encompasses the development of the product and its manufacturing processes. In the pre-production test, the product design as well as the manufacturing process design are validated. During the production in a prototype factory, the cars pass

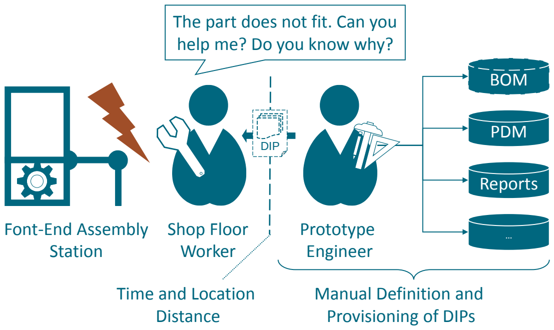

Fig. 1. Use Case Scenario "Problem Resolving in the Prototype Factory"

several assembly stations in a production line. At each station, a shop floor worker assembles multiple parts. Multiple variants can be manufactured at one station. Variants differ, for example, in modality, such as whether it is a limousine or cabriolet, right-handed or left-handed drive, and also in the equipment, such as the seat, the material used, or the functionality that is provided. Since the product and the process are not as well defined as they are in series production, plenty of failures may occur, e.g., parts cannot be assembled because the tolerances do not match or wrong parts are supplied. In order to identify the cause and solve the problem, the shop floor worker notifies the prototype engineer to receive DIPs as shown in Fig. 1.

In the following, we describe a case scenario, in which the part "console" does not fit in the apparatus of the *front-end assembly station*: The shop floor worker at the front-end assembly station recognizes the problem and notifies the prototype engineer to get feedback to resolve it. There could be plenty of causes for this problem. The prototype engineer has to check all causes and consequently defines a process how the error can be eliminated. In the following and due to space reason, we only look into three representative steps to identify an error:

(1) Identify Part Number: Identifying the part number is crucial to gather information about the part. There are multiple ways to determine the part number. One is to search and scan documents where part information is contained or to ask colleagues. Another possibility is that the engineers search the part number and information about the part in the *bill of material* (BOM) system, which stores all parts necessary to manufacture the product.

(2) Geometry Visualization: In order to check the correctness of the part's geometry, the *front-end assembly* 3D geometry has to be visualized. Sometimes the geometry is wrong, because the supplier has manufactured the wrong version or the wrong variant. The visualization requires that the variant of the prototype is known, such as if it is a left-hand drive or right-hand drive, and to find all parts and assemblies with the correct version and belonging to the correct variant. The engineer has to access the *product management system* (PDM), identify the correct parts, and the right version of the 3D geometry file. Since there are so

many possibilities of variants and versions, and the engineer is not an expert of these systems, this is a complex and error-prone task.

(3) Check Tolerances: Parts are constructed using tolerances. However, it may happen that the tolerances do not lie in the defined range. Therefore, measurement reports have to be checked to find out which part does not fit into the tolerance range. Since the engineering domain is dynamic, sometimes the engineer does not know which measurements are performed and where the reports are stored.

Since the shop floor worker is not an expert of these systems, the prototype engineer has to manually acquire and compose the information from the different data sources to a DIP by printing the information. The DIP contains information about the part from the BOM such as name and material thickness. In addition, it contains the 3D Model of the part and related measurement reports. DIPs are highly dependent on the current situation, e.g., the skills of the shop floor worker, the location and the tasks. Printed DIPs are handed to the worker.

Therefore, the vision is to create DIPs automatically and provide them to the shop floor worker by an (e.g., mobile) application. This leads to a significant time and cost reduction, since the shop floor worker gets the information ad-hoc and the prototype engineer is not involved in this task.

2.2 Requirements

Based on the scenario and on workshops with domain experts, we identified several requirements characterizing DIPs:

(R1) Information filtering: DIPs should gather information that are required for their specific situations. For example, only information about assemblies which have the correct version and variant should be provided.

(R2) Information acquisition: DIPs should acquire information from different data sources. For an effective data acquisition, uniform data access across various engineering data sources is required.

(R3) Information discovery: DIPs should gather information from all required data sources. Consequently, a means is required to discover and consider new data sources. Furthermore, related information should be discovered and composed from different data sources, although the relation is not modeled and reflected in the data itself.

In addition to the case scenario, we analyzed the existing engineering IT landscape and performed a literature review [1,11–13]. On this basis, we identified additional domain-specific requirements for our approach:

(R4) Support of legacy IT-systems: In the engineering domain, there are lots of legacy systems. However, they contain a huge part of the enterprise's knowledge, and thus, cannot be replaced [11]. Since these legacy systems oftentimes have interfaces to many other systems, they furthermore cannot be changed easily. Hence, the approach should not require a change of underlying systems.

(R5) Support of dynamic environments: Data sources in manufacturing environments appear and disappear. One reason for this is the ongoing innovation in simulation technologies, which leads to new tools that have to be integrated into the environment. Furthermore, sensors to monitor the manufacturing process

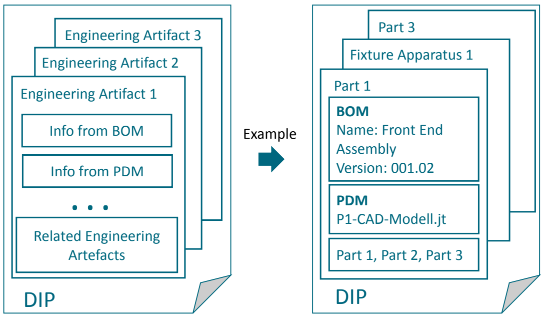

Fig. 2. DIP schema (on the left) and corresponding example (on the right)

are constantly being installed, moved, or removed. Hence, the approach should also be able to deal with dynamically (dis-)appearing data sources.

(R6) Support of domain users: In general, the involved users in the engineering domain, e.g. shop floor workers, do not have extensive knowledge about IT-systems or programming languages. Consequently, an approach characterizing DIPs in the engineering domain needs to support these users through abstraction from technical details and should not force them to adopt new skills.

3 Context-aware Composition of Decision Information Packages

In this section, we introduce the concept of decision information packages and the context-aware engineering data model, which is fundamental to create DIPs.

3.1 Decision Information Packages

The goal of a DIP is to provide the required information in order to solve a problem on the shop floor. DIPs compose data from multiple, heterogeneous data sources with respect to the context of the problem.

Fig. 2 shows an abstract schema of a DIP. A DIP encapsulates data of multiple *engineering artifacts*. Engineering artifacts represent everything required to build a product. This includes components of the product as well as of the production process such as machines and manufacturing tools. The data of an engineering artifact is composed from different data sources and information about related engineering artifacts. Related engineering artifacts of a part can be subcomponents or machines at which the part is manufactured.

In the depicted example, on the right of Fig. 2, the DIP contains information about *part 1*, *part 2* and the *fixture apparatus 1*. Information regarding *part 1* come from the BOM and the PDM-System. Furthermore, the name of the part and the current version number are acquired from the BOM. From the PDM system, the file containing the 3D model is included, called P1-CAD-Modell.jt, which is

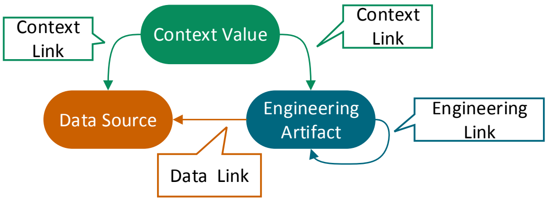

Fig. 3. Context-aware engineering model

necessary to visualize the part. *Part 1* is related to emphpart 2, emphpart 3, and *part 6*, which can be subcomponents.

3.2 Context-Aware Engineering Data Model

In order to compose a DIP, context data has to be linked to engineering data. This enables identifying relevant engineering artifacts to the current context of a problem. Furthermore, it has to be defined, in which data sources data about the engineering artifact is stored and if this data is relevant for a particular context.

To address these issues, we developed the context-aware engineering data model as shown in Fig. 3. In an abstract manner, it models the relation between engineering artifacts, data sources, and context values. Context attribute values characterize the context of the problem, e.g., the station where the problem occurs or the role of the shop floor worker. Data sources represent an abstract description of IT systems or data storage. On the one hand, context links connect context values to engineering artifacts to define which engineering artifacts are relevant for a particular context. On the other hand, they connect context values to data sources to define from which source relevant data originates. Data links define which data source contains information about this engineering artifact. Engineering links define dependencies between engineering artifacts.

In order to construct a DIP with respect to a particular context, the context-aware engineering model can be used as follows. The context of the problem is described by a set of context values. To identify relevant engineering artifacts, the context links have to be used. Outgoing from the source context values, the relevant engineering artifact can be found. Using data links, the required data sources can be identified. Thereby, the context links to data sources have to be considered because they could restrict the relevant data sources. Using the engineering links, the related engineering artifact can be identified.

4 Related Work

There are approaches which provide data acquisition from multiple engineering IT systems. Katzenbach et al. [11] introduce a common engineering client, where data are provisioned by an engineering service bus. Similarly, the authors of [13] suggest a system-level integration using standards and harmonized human interfaces. However, none of them consider filtering relevant data.

There are several works that link context and data on the application level. Bobillo et al. [3] develop a context-domain relevance model for knowledge-based systems. This is based on the domain ontology of the knowledge-based system and on a context ontology. Furthermore, they provide a reasoning algorithm. Barkat et al. [2] define a context ontology to integrate context into the semantic databases, called OntoDB. Therefore, it is just useable for one application based on this database. Bolchini et al. [4] introduce a context-domain ontology based on a self-developed context model to define the portion of the ontology which are relevant. Similar to this, they provide a method to define context-aware views for relational databases [5]. Hence, many related approaches try to achieve similar goals using ontology models. In our approach, we decided to omit the use of ontologies to reduce the complexity. Most advantages of ontologies come with reasoning and linking to other ontologies. However, in our approach, this is not necessary. Consequently, a simple meta-model is sufficient. With enhancements in the future, ontologies could be introduced with reasonable effort due to the interchangeability of the models.

5 Summary and Outlook

In this paper, we introduce a first approach for the composition and provisioning of *decision information packages*, called DIPs, to support problem resolving on the shop floor. We conduct a real-world use case analysis at a German car manufacturer to derive specific requirements and to discover domain-specific issues. To cope with these requirements, our approach introduces decision information packages that are constructed using a meta-model. The meta-model – called Context-Aware Engineering Data Model – links engineering data to context data.

For future work, we plan to design an architecture to automatically compose DIPs, which serves as a foundation for a corresponding proof-of-concept implementation, also comprising the CAEM model. This also comprises the integration of the Resource Management Platform, as introduced in [8, 9], to retrieve distributed data. Based on this implementation, we plan to conduct a thorough evaluation of our approach. Furthermore, we plan to design mobile apps to provide these DIPs in an appropriate and non-intrusive manner.

References

1. Ahlers, D., Mehrpoor, M., Kristensen, K., Krogstie, J.: Challenges for information access in multi-disciplinary product design and engineering settings. In: The Tenth International Conference on Digital Information Management (ICDIM 2015) (2015)
2. Barkat, O., Bellatreche, L.: Linking Context to Ontologies. In: 2015 11th International Conference on Semantics, Knowledge and Grids (SKG) (2015)
3. Bobillo, F., Delgado, M., Gómez-Romero, J.: Representation of context-dependant knowledge in ontologies: A model and an application. Expert Systems with Applications (2008)

4. Bolchini, C., Curino, C., Schreiber, F.A., Tanca, L.: Context Integration for Mobile Data Tailoring. In: 7th International Conference on Mobile Data Management (MDM'06) (2006)
5. Bolchini, C., Quintarelli, E., Schreiber, F.A., Baldassarre, M.T.: Context-Aware Knowledge Querying in a Networked Enterprise. In: Methodologies and technologies for networked enterprises. Springer (2012)
6. Dey, A.K., Abowd, G.D.: Towards a Better Understanding of Context and Context-Awareness. In: Handheld and Ubiquitous Computing. Springer Berlin Heidelberg (1999)
7. Gröger, C., Kassner, L., Hoos, E., Königsberger, J., Kiefer, C., Silcher, S., Mitschang, B.: The Data-driven Factory - Leveraging Big Industrial Data for Agile, Learning and Human-centric Manufacturing. In: ICEIS 2016 (2016)
8. Hirmer, P., Wieland, M., Breitenbücher, U., Mitschang, B.: Automated Sensor Registration, Binding and Sensor Data Provisioning. In: Proceedings of the CAiSE 2016 Forum at the 28th International Conference on Advanced Information Systems Engineering (2016)
9. Hirmer, P., Wieland, M., Breitenbücher, U., Mitschang, B.: Dynamic Ontology-based Sensor Binding. In: Advances in Databases and Information Systems. 20th East European Conference, ADBIS 2016, Prague, Czech Republic, August 28-31, 2016, Proceedings (2016)
10. Jazdi, N.: Cyber physical systems in the context of Industry 4.0. In: Automation, Quality and Testing, Robotics, 2014 IEEE International Conference on (2014)
11. Katzenbach, A.: Automotive. In: Concurrent Engineering in the 21st Century. Springer International Publishing (2015)
12. Stark, R., Hayka, H., Israel, J.H., Kim, M., Müller, P., Völlinger, U.: Virtuelle Produktentstehung in der Automobilindustrie. Informatik-Spektrum (2011)
13. Trippner, D., Rude, S., Schreiber, A.: Challenges to Digital Product and Process Development Systems at BMW. In: Concurrent Engineering in the 21st Century. Springer International Publishing (2015)
14. Vermesan, O., Friess, P.: Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems. River Publishers (2013)
15. Zuehlke, D.: SmartFactory—Towards a factory-of-things. Annual Reviews in Control (2010)