# Using Explicit and Machine-Understandable Engineering Knowledge for Defect Detection in Automation Systems Engineering

Olga Kovalenko and Thomas Moser
Christian Doppler Laboratory for Software Engineering Integration for Flexible Automation Systems
Vienna University of Technology
Favoritenstrasse 9-11/E188, A-1040 Vienna
{kovalenko, moser}@ifs.tuwien.ac.at

***Abstract*—Today the costs of a failure in operation of huge industrial complexes are very high. Traditional approaches for defect detection in automation systems engineering in principle work, but generally don't take into account the semantic heterogeneity of tools and data models which are used within the engineering of industrial automation systems. Thus, some defects can remain undetected. Also, such systems have to be implemented anew for each concrete case. In this paper we present our ongoing and planned research aimed to improve the defect detection processes. Our approach is based on using explicit knowledge about industrial system stored in a set of ontologies which integrate information from different heterogeneous data sources and present it in machine-understandable form. Another important part of the approach is rules describing system's logic. Such rules can, through the use of integrated engineering knowledge stored in ontologies, detect faults which otherwise are hard to identify using traditional methods. Major expected results are the more efficient and effective defect detection and the potential reuse of the created ontologies in other projects.**

## I. Introduction

Nowadays, industrial automation systems have tendencies to become more and more complex and large-scale. At this rate, system failures can lead to considerable material losses and even to serious dangers for environment or people. Thus, there exists the vital need for effective techniques for defect detection (DD) for such kinds of systems.

Let's clarify the meaning of word "defect" within this paper. We define "defect" not only as equipment defects during runtime system operation but also as inconsistencies between different design-time artifacts, e.g. electrical plans and P&ID (Piping and Instrumentation) models; as well as inconsistencies between design-time and runtime data models. Accordingly, "defect detection" is a process, which addresses all these types of defects.

To be able to identify above-listed defects it is required to use all available information, i.e. historical data about system parameters values, information from control system and monitoring devices, data from different analytical tools, in conjunction for effective system analysis and defect diagnosis. But the problem is that such information is usually dispersed through various heterogeneous data sources and it is difficult to obtain it in explicit and machine-understandable form.

Heterogeneity comes from different origins. The first type of heterogeneity is the semantic gap between design-time and runtime. Usually different data formats and data models are used during design-time and runtime which makes it difficult to obtain the overall integrated view on the industrial system [1]. The second type of heterogeneity comes from the fact that multiple disciplines are involved in the engineering of industrial automation systems, e.g. mechanical engineering, electrical engineering and software engineering. Experts from different disciplines work on the same tasks but consider them from different points of view. Therefore they use various tools which are good in supporting their concrete discipline but are not adapted to cooperate with each other [2].

Another limitation of traditional approach is the necessity to implement DD systems anew for each industrial system. The development and deployment of such system require lots of efforts from different types of experts, e.g. software engineers, technical engineers and domain experts, and the impossibility to reuse it at least partially is objectionable.

The goal of this paper is to propose a knowledge-based approach for industrial DD which could help addressing the above-mentioned problems. The cornerstone of the approach is an ontology-based engineering knowledge base (EKB). By using ontologies we can integrate both design-time and runtime data from heterogeneous data sources and data from different disciplines and present it in machine-understandable form. Thus, different applications can share the engineering knowledge and use it for earlier and easier detection of defects in system operation. In addition to the EKB and based on domain experts knowledge, we can define specifications describing the system behavior and restrictions for system parameters and translate them into simple "if-then" rules. Depending on structure of rules and the information which is available in EKB, such systems could help to identify inconsistencies between design-time and runtime, inconsistencies in data models between different disciplines or facilitate diagnosing of runtime defects in system operation.

Another benefit of such approach will be the possibility to partially reuse the created ontologies in next projects and thereby an effort reduction to deploy DD systems in new projects compared to start from the scratch again.

The remainder of the paper has following structure: Section 2 summarizes related work on industrial DD using ontologies. Section 3 introduces the solution approach and section 4 presents the planned validation method. Section 5 summarizes current research status and identifies further research work.

## II. Related Work

While there are a lot of works which describe approaches for DD using ontologies, most of them concentrate on some specific type of defects, i.e. on runtime equipment defects or merely on finding inconsistencies between various design-time models. Although these contributions provide important solutions for implementing DD systems in various domains they don't provide a possibility to address a wide range of defects which could be done by our approach. There is also lack of work on semantic integration of data across heterogeneous data sources, which, we strongly believe, could improve the quality and efficiency of industrial DD processes.

Ontologies for DD are used primarily as engineering knowledge bases. In the majority of scientific works on this subject, ontologies are used to explicitly describe knowledge in some specific domain and then to share this knowledge between different applications or agents. For example, Guiyang et al. [3] introduce a general ontology to describe the semantics of and the condition monitoring and maintenance domain. Semantic enhancement is achieved by mapping the sensor data and legacy system data into the monitoring and maintenance domain ontology. Thus, various applications for maintenance management or maintenance technicians can utilize this knowledge base to make decisions in a consistent and systematic way.

Papadopoulos and Cipcigan [4] use an ontology describing the wind turbine components, possible observations and corresponding types of faults. Using Protege queries on data from sensor measurements, the proposed system allows identifying limit violations and informs user about fault.

Some authors include in ontology concepts for evaluating riskiness level of faults. The ontology model of Lewis and Roberts [5] represents system concepts and associated faults in the railway domain. All detected faults are divided into 2 groups, according to their criticality and there are different reactions on "critical" and "incipient" type of faults. Such technique could be also useful in industrial automated systems domain to produce different responses depending of potential hazardous of detected fault and is considered as one of possible directions for further work.

Some other approaches are based on distributed network of maintenance or monitoring devices with embedded semantics which use ontology as a common knowledge base. Terziyan et al. [6] present a framework for industrial semantics-enabled maintenance services organized in peer-to-peer network of services platforms embedded into maintained devices and specific maintenance centre nodes. The proposed framework - OntoServ.Net, provides a solution for building large-scale industrial maintenance networks. D'Elia et al. [7] present a practical approach to smart context-aware applications for the maintenance of large buildings, where ontology-based interoperability is exploited to enable the easy integration of multi-vendor multi-platform devices with existing applications. Designed to interoperate, these applications support automatic fault detection; assignment of interventions to maintenance operators; and on-the-field support (such as aided identification of fault locations).

## III. Proposed Solution

In this section we describe a knowledge-based approach for industrial DD which combines integration of data from heterogeneous data sources with flexible technique for rules definition, which allows diagnosing various types of defects within the industrial system. This approach is part of a large project in the CD Lab "Software Engineering Integration for Flexible Automation Systems Engineering". The goal of this research is to develop a tool-supported method that supports effective detection of wide range of defects in complex heterogeneous environments.

### A. Engineering Knowledge Base

For resolving the problem of semantic heterogeneity we use the Engineering Knowledge Base (EKB) concept, which was introduced by Moser and Biffl [2]. EKB is layered semantic model which allows integration of knowledge from heterogeneous data sources and is well suited for supporting the data exchange between heterogeneous engineering tools [2].

The EKB presents a set of local specific area ontologies together with common domain ontology and a set of assertions which describe industrial system behavior.

Each local ontology contains knowledge about some specific area within the engineering process, e.g. electricity engineering knowledge, technical specialist knowledge, software engineering knowledge, and thus, dividing whole industrial system into several semantic layers and representing it from different points of view. Experts from each specific area can use different terminology and different design-time data models. For example, "circuit" concept could have diverse meaning for software engineer and electrical engineer. For software engineer "circuit" could mean a closed directed path, with repeated vertices allowed, while for electrical engineer it will be an electrical circuit. Local ontologies reflect these differences through using area-specific vocabulary to describe entities and relationships between them.

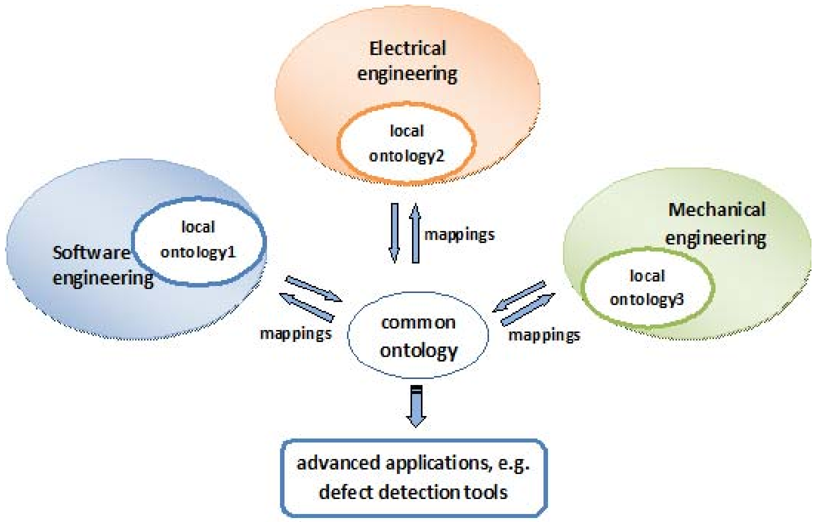


Figure 1. Engineering Knowledge Base: integration of area-specific knowledge in common domain ontology.

The common domain ontology stores the more general domain-level knowledge than local area-specific ontologies. It ensures the semantic integration of concepts and data models from heterogeneous data sources within the engineering process. A set of mappings is determined between each local ontology and common ontology. Each concept from a local ontology is mapped to either corresponding concept in common domain ontology or an attribute of some domain ontology concept. The granularity of the mapped elements does not need to be similar, e.g., a concept can be mapped to the attribute of another concept, or vice versa [2]. Such mappings allow domain experts to use familiar data model and, simultaneously, through the common ontology they can share the engineering knowledge with other participants of engineering process. Figure 1 shows the high-level EKB structure on example of three area-specific ontologies, which corresponds to different disciplines within the whole industrial system.

The detailed process of EKB creation for multidisciplinary engineering project and corresponding difficulties are considered in [8].

Using ontologies as knowledge repositories has some important advantages. The first one is the ability to explicitly model the engineering concepts using machine-understandable syntax. Secondly, it becomes possible to derive the new facts using ontology-based reasoning for DD [9]. In cooperation with domain experts a set of assertions describing restrictions of system parameters and explicitly declaring design-time preconditions of system behavior can be specified. Then, for instance, applications can query the EKB at runtime with semantic web query languages like SPARQL (SPARQL Protocol and RDF Query Language) or SWRL (SemanticWeb Rule Language) to infer about the assertions based on measured runtime data.

Figure 2 shows the process of industrial DD with marked contributions of our research. Integrated knowledge about industrial system from EKB, runtime and historical data from monitoring devices and a set of predefined assertions - all serve as an input for a DD tool, which is capable to identify various types of defects (i.e., equipment defects during runtime system operation, inconsistencies between different design-time artifacts and inconsistencies between design-time and runtime data models and tools). Then, through human-machine interface (HMI), information about detected defects is presented to industrial users, who are responsible for making decisions. It could be control personnel in case of equipment defects or system analyst if some inconsistencies between data models were identified. Our contributions consist in a) configuring the EKB and analyzing which initial data should be available in ontologies to identify each specific type of defects (marked as (1)); b) defining the structure of assertions for each type of defects and providing a clear and understandable mechanism to insert and modify assertions (marked as (2)); c) providing a DD tool which will be able to identify all relevant types of defects and has appropriate mechanism for presenting the results to industrial users (marked as (3)).

### *B. RuleBench tool*

Although we can use existing tools, as SPARQL queries in Protege or reasoners to check predefined assertions it will be difficult for industrial experts or control personnel to work with such tools without assistance of software engineers. Therefore, we provide a simplified middle-layer tool for assertions definition and diagnosing of faults based on historical and runtime data from monitoring devices - RuleBench. It has simple and clear user interface. The assertions can be entered as simple ”if-then” rules in Java

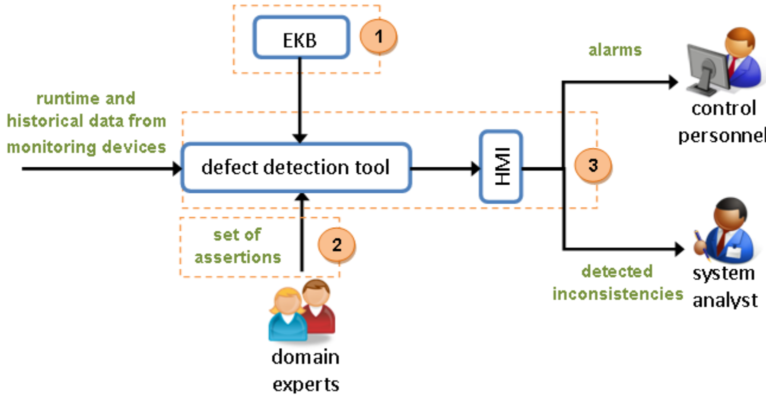


Figure 2. Process of industrial DD with marked contributions.

syntax. A set of rules then can be saved in file on hard disk.

For example, if considering a sector of plant consisting of a tank, a pump, which pumping liquid in this tank and a several valves controlling the liquid movement to the tank, then a rule for identifying liquid leakage could sound like following in natural language: ”if the pump is inactive and all valves closed, but the liquid level in the tank changes, then there could be a leakage in the tank or some valves could be broken”. Such rule can be easily transformed in RuleBench syntax and used for DD in industrial system.

RuleBench aids to simplify the process of rules-based DD in comparing with working directly in ontology development environments, e.g. Protege, so the industrial plant staff could independently input the rules and carry out the checking, without help of software experts.

RuleBench integrates in one several functions: a) it works as a DD tool; b) it provides HMI for presenting information about identified defects to industrial users; and c) it provides the tool for creating and modifying of assertions. Thus, it covers the contribution (3) and partially contribution (2) from Figure 2.

## IV. Plan for Validation

We plan to validate our approach in two steps. First step will consist in using our approach on the educational prototype of an industrial process plant - the so called tank model of the Odo Struger Laboratory of the Automation and Control Institute at the Vienna University of Technology, which, despite its simplicity, provides the possibility to simulate typical industrial production processes. As second step, we intend to apply our approach to diagnose defects to real-life data from our industrial partners. After both steps we will analyze the results and discuss them with industrial users to assure the correct evaluation and relevance for real word industrial needs.

## V. Current Research Status and Next Steps

Currently we have worked on analyzing which types of defects could be relevant for our research. Based on information about relevant defects we can derive configurations of EKB and structure of rules which are necessary to diagnose these types of defects. We also investigate which metrics could be used to validate the effectiveness of our approach comparing to other approaches.

Further important research issues are a) the development of RuleBench tool, e.g. make it capable to work with ontologies with more complex classes structure and expanding the number of supported property types; b) implementation of abilities to determine the danger level of detected failure and, according to this level, give a recommendation on how to react for plant personnel; c) conformation of ontologies to existing international industrial standards, such as ISA-88, ISA-95, OPC UA; d) investigation of real industrial use cases in cooperation with industrial experts to make the proposed system more useful for real-world needs.

## Acknowledgment

This work has been supported by the Christian Doppler Forschungsgesellschaft and by the Erasmus Mundus Action 2 Programme of the European Union.

## References

[1] M. Melik-Merkumians, T. Moser, A. Schatten, A. Zoitl, and S. Biffl, “Knowledge-based runtime failure detection for industrial automation systems,” in *13th International Conference on Model Driven Engineering Languages and Systems (MODELS 2010)*, 2010.

[2] T. Moser and S. Biffl, “Semantic tool interoperability for engineering manufacturing systems,” in *Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on*, sept. 2010, pp. 1–8.

[3] G. Jin, Z. Xiang, and F. Lv, “Semantic integrated condition monitoring and maintenance of complex system,” in *Industrial Engineering and Engineering Management, 2009. IE EM ’09. 16th International Conference on*, oct. 2009, pp. 670 –674.

[4] P. Papadopoulos and L. Cipcigan, “Wind turbines’ condition monitoring: an ontology model,” in *Sustainable Power Generation and Supply, 2009. SUPERGEN ’09. International Conference on*, april 2009, pp. 1–4.

[5] R. Lewis and C. Roberts, “Using non-monotonic reasoning to manage uncertainty in railway asset diagnostics,” *Expert Systems with Applications*, vol. 37, no. 5, pp. 3616–3623, 2010.

[6] V. Terziyan and O. Kononenko, “Semantic web enabled web services: State-of-art and industrial challenges,” in *Web Services - ICWS-Europe 2003*, ser. Lecture Notes in Computer Science, M. Jeckle and L.-J. Zhang, Eds. Springer Berlin / Heidelberg, 2003, vol. 2853, pp. 183–197.

[7] A. D’Elia, L. Roffia, G. Zamagni, F. Vergari, A. Toninelli, and P. Bellavista, “Smart applications for the maintenance of large buildings: How to achieve ontology-based interoperability at the information level,” in *Computers and Communications (ISCC), 2010 IEEE Symposium on*, june 2010, pp. 1–6.

[8] T. Moser and S. Biffl, “Semantic integration of software and systems engineering environments,” *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. PP, no. 99, pp. 1–13, 2011.

[9] M. Melik-Merkumians, A. Zoitl, and T. Moser, “Ontology-based fault diagnosis for industrial control applications,” in *Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on*, sept. 2010, pp. 1–4.