Agent Based Framework to Support Manufacturing Problem Solving Integrating Product Lifecycle Management and Case-Based Reasoning

Alvaro Camarillo[1,2(✉)], José Ríos[2], and Klaus-Dieter Althoff[3,4]

[1] Exide Technologies GmbH, Odertal 35, 37431 Bad Lauterberg, Germany
alvaro.camarillo@eu.exide.com
[2] Mechanical Engineering Department, Universidad Politécnica de Madrid, Jose Gutierrez Abascal 2, 2800 Madrid, Spain
a.camarillo@alumnos.upm.es, jose.rios@upm.es
[3] German Research Center for Artificial Intelligence (DFKI), Trippstadter Straße 122, 67663 Kaiserslautern, Germany
klaus-dieter.althoff@dfki.de
[4] University of Hildesheim, Universitätsplatz 1, 31141 Hildesheim, Germany

Abstract. During the execution of manufacturing processes, problems arise and they have to be solved systematically to reach and exceed production targets. Normally, a production team analyzes and solves these problems, with the support of different methodologies and working directly on the shop floor. This paper presents an ontology-based approach to easily capture and reuse the knowledge generated in such a process of Manufacturing Problem Solving (MPS). The proposed ontology is used as basis in an ad-hoc MPS software system. The architecture of the MPS system is based on the integration of three technologies: PLM (Product Lifecycle Management), CBR (Case-Based Reasoning) and software agents. The PLM system is used as an automatic source of the problem context information. The CBR system is used as repository of cases and artificial intelligence tool to support the efficient reuse of knowledge during the resolution of new problems. A software agent platform allows developing an integrated prototype of an ad-hoc software system. This paper shows the architecture of the MPS system prototype.

Keywords: Ontology · Product Lifecycle Management (PLM)
Case-Based Reasoning (CBR)
Process Failure Mode and Effect Analysis (PFMEA)
Manufacturing Problem Solving (MPS)

1 Introduction

Analytical methods are applied to prevent failures during the design phase of manufacturing processes and facilities, but the reality shows that the defined production targets are often not reached due to the arise of problems. Aiming to solve them in a systematic way, different methods have been developed. Continuous Improvement Process (CIP)

J. Ríos et al. (Eds.): PLM 2017, IFIP AICT 517, pp. 116–128, 2017.
https://doi.org/10.1007/978-3-319-72905-3_11

and Manufacturing Problem Solving (MPS) embrace these methods [1]. A relevant example of them is the 8D methodology, which is quite spread and known in manufacturing since nowadays it is the standard method used to analyze and present quality claims in important industrial sectors such as the automotive one [2]. The 8D method, similarly to other CIP or MPS methods, provides a structured process to facilitate finding and improving solutions, but it only brings results when actors with enough experience and knowledge drive it [3]. This paper proposes a software system to compensate the possible lack of experience and knowledge of some team members. One of the key requirements for such a system is the capability to collect and reuse knowledge created during the application of the 8D method in the resolution of production problems at the shop floor level. In that way, the system would be linked to the daily MPS activity of the manufacturing plant. To address this global aim, the proposed approach is structured into:

1. Manufacturing problems have to be described in a consistent and systematic way to allow a common understanding by the MPS personnel and their processing by means of a software system. The definition of an ontology is the alternative to address this aspect [4]. This requires reviewing two main methods. Firstly, how the 8D method allows defining a manufacturing problem. Secondly, how Process Failure Mode and Effect Analysis (PFMEA) [2], as preventive method, allows defining manufacturing problems, effects and solutions and how this method could be used to create an initial set of cases of potential problems.
2. The ontological approach is the basis to create two repositories of manufacturing-related knowledge. Manufacturing is an extremely wide context. Therefore, it is necessary to distinguish among different problems and to connect them with previously found solutions; this can be achieved by means of context data. A PLM (Product Lifecycle Management) application [5] can be used as a logical source of extended information of Product-Process-Resource (PPR) related to each problem, and therefore, to describe the context of each manufacturing problem. This will allow enriching automatically the initial problem description given by the MPS personnel. A CBR system [6] can be used as the logical source of manufacturing problems and solutions, where the initial PFMEA case base of potential problems is stored and extended continuously with the resolution of new problems.
3. To assist the MPS personnel, it is needed a systematic reasoning method to connect each problem with similar cases already solved. Case-Based Reasoning (CBR) [6] is proposed as an artificial intelligence tool to support the efficient reuse of collected knowledge in new problems and to assist in the problem solving process.

The paper starts with a general introduction to the main topics and a review of existing similar research works. Section 3 shows the developed ontology to represent a manufacturing problem. Section 4 presents the ontological model implementation in a MPS process flow and the MPS system architecture of the prototype application. The prototype application is implemented as case study in the company Exide Technologies, a multinational company that produces stored electrical energy solutions. The paper ends with some results, conclusions and future work proposals.

2 State of the Art

2.1 Knowledge Representation Based on Ontology

A fundamental aspect of this work is the representation of manufacturing knowledge and problems. One way to address it is by means of an ontology. In the literature, there are examples of ontologies applied to the context of MPS. Foguem et al. [7] create an ontology based on conceptual graphs for Feedback Experience Systems (FES). This approach relates to the topic MPS, however its structure it is not compatible with the PFMEA method, which is considered as the main initial source of solved cases. Dittmann et al. [8] and Ebrahimipour et al. [9] present two examples of ontologies for the FMEA environment developed to ensure the easy reuse of the information stored in the FMEA analyses. The work of Dittmann et al. is selected as basis for the knowledge representation model of this work due to its focus on FMEA and the clarity and simplicity of its model. It proposes a ROOT_CONCEPT class that has as subclasses FMEA, Component, Function, Failure_mode, Control_method, Risk_priority_number, and Containment_action. It also defines relationships among the different concepts, for example "fulfills_a_function" that relates Component to Function, or "has_failure_mode" that relates Function to Failure_mode. In parallel, the classes Component and Function have associated taxonomies. The associated taxonomies allow limiting the possible values to be used when instantiating the model classes, in that sense, it could be seen as a taxonomic approach to natural language processing [10]. An approach based on free text would be an alternative, but in that case a domain specific dictionary and free-text natural language processing techniques would be required to search for relevant text. That line of research is out of the scope of this work.

Having introduced the way of representing knowledge, the next subsection presents PLM systems as the logical source of extended PPR information related to the manufacturing problem under analysis.

2.2 Product Lifecycle Management Systems

As explained in the introduction, the application domain of manufacturing is extremely wide. The ontology selected above allows representing a problem following the PFMEA method. However, it has to be taken into account that the document resulting from applying the PFMEA method is a detailed analysis of a specific process, therefore the specified information is restricted to the scope of the components of that process. In the approach of this work, that limited scope must be cut, the proposed framework should allow dealing with problems coming from any type of manufacturing processes at any place. To do so, it is needed an additional input of information that sets clear differences among problems, and that is the context of the problem. A PLM system [5] arises here as the natural and logical source of this context, since it aims facilitating the storage of all the PPR information of a company.

The work of Bertin et al. [11, 12] is very relevant for this work because they also proposes the use of a PLM system as central repository of data for a Lessons Learned System (LLS). They focus on the capture and reuse of knowledge along the Engineering

Change Request (ECR) process of the company. This approach sets the focus on the technical staff of the company, which is the typical group of users leading ECR. By contrast, the framework presented in this paper sets the focus on the operators working directly on the line, being this group much less used to computer system, and with less technical background. This puts some additional requirements for developing intuitive and simple interfaces. Additionally, it is proposed the use of the PFMEA method as an initial set of possible problems, so the system should be able to provide the users with some solutions from the very beginning.

Having defined the way of representing knowledge, and the system to retrieve structured information needed to describe the context of the manufacturing problem, the next section introduces CBR as the artificial intelligence tool to support the storage of cases and the finding of solutions during the MPS activity.

2.3 Case-Based Reasoning

CBR is useful when the reality under analysis is too complex and difficult to be represented in a model [6]. This is the case of manufacturing, the application domain of this work, with a huge variety of production processes being influenced by many types of machines, environments, materials, methods or persons. CBR is particularly applicable to problems where earlier cases are available, even when the domain is not understood well enough to create a deep domain model. The approach adopted in this work focuses on CBR as the artificial intelligence tool to support the storage of cases and the reasoning to find solutions during the MPS activity.

Manufacturing, as application domain, represents a big challenge for CBR due to its vast extension and complexity. A single case base, containing information related to failures from hundreds of different types of processes, coming from many different manufacturing lines and plants would create serious problems of retrieval speed, and maintainability. Multi-case base reasoning systems were created years ago to tackle this type of issues. Among the different research paths available in the literature related to this type of systems, and due to its flexible approach to knowledge modularization, this work focuses on SEASALT (Shared Experience using an Agent-based System Architecture LayouT). SEASALT is a domain-independent architecture for extracting, analyzing, sharing, and providing experiences [13]. The work of Reuss et al. [14] is an application example of SEASALT linked to Problem Solving in aircraft diagnosis and maintenance, where the system provides maintenance solutions to known failures.

Based on the introduced theoretical background, the next section presents a concrete proposal of ontology to represent manufacturing problems in a generic way.

3 Manufacturing Problem Solving Ontology

This section presents a proposal of ontology to represent any kind of manufacturing problem from any process at any manufacturing place. This ontology has also the additional requirement of being compatible with the information structure of the PFMEA

method. As it was previously mentioned, PFMEA can provide the first set of possible problems to a computer system based on this ontology.

As mentioned in Sect. 2, this ontology uses the model of Dittmann et al. [8] as basis. The classes Component, Function, and Failure with all its interrelationships are taken from that model (see Fig. 1), but additional classes were added to create the proposed ontology. These three initial classes allow defining the core description of the problem with the same structure of a PFMEA analysis. In a PFMEA analysis, each component has different functions, and each function may have different failure modes, aiming to find all possible failures in the process. It should keep in mind that when aiming to find the resolution of a problem, a single and concrete issue is addressed. This creates the need for extending the initial set of classes with the class Problem (Fig. 1) that will have the link to the specific and unique threefold structure: component-function-failure. This threefold structure contains the core definition of the problem under analysis.

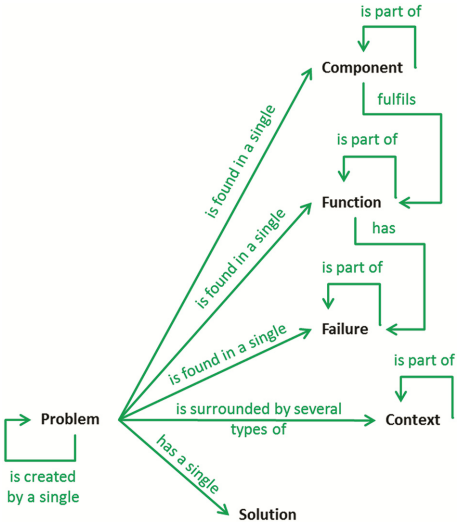


Fig. 1. Manufacturing Problem Solving main concepts ontology

Figure 1 shows that the three classes (Component, Function, and Failure) have a relationship of type "is part of" pointing to themselves. This means that these classes have their own subclasses forming a taxonomy, which could be used by a CBR system to calculate similarity among cases.

As previously stated, the proposed ontology defines a problem as a specific threefold structure of component-function-failure. However, the same exact threefold structure can be found in different scenarios or contexts. This issue is not found in the PFMEA method because each analysis is linked with a specific process in a specific machine, so the context is clearly defined. Therefore, the ontology needs an additional class to add information related to the surroundings of the problem: the class Context. This class has also a relationship of type "is part of" pointing to itself. Therefore, as Component,

Function and Failure it has a taxonomy that has been developed under similarity criteria to allow the later finding of similar problems by a CBR system.

Finally, a last class is needed, to contain the information of the solution to the manufacturing problem: the class Solution. This class will comprise the information related to containment actions, corrective actions, and preventing actions that should be applied to solve a problem.

Each defined class has different associated attributes, and these attributes have sets of allowed values and restrictions. The attributes of the class Problem must be defined by the MPS personnel and they create a basic definition of the problem following the recommendations of the MPS method Kepner-Tregoe [15] (i.e. to answer to the questions "How often?", "What?", "When?", "Where?", "Who?", and "Why is a problem?"). The attributes of the classes Component, Function and Failure are their corresponding type within the defined taxonomies, and they should be also defined by the MPS personnel. The class Context has the attribute of its position within the taxonomy, and all its associated subclasses have a big variety of attributes attending to their nature (e.g. max. casting temperature, min. pressure, or brand of a machine component). These attributes should be defined with data stored in the PLM system.

In order to perform a preliminary validation of the proposed ontology, several problems from different manufacturing environments (machining, stamping, casting and assembly) were collected and represented with this ontology. Table 1 shows an example.

Table 1. Example of problem description with the proposed ontology

Class	Attribute	Allowed values	Value
Problem	Problem ID	ID Number from 0 to 9999	253
	What Problem?	String	Plate with burrs
	Where Product?	Section from set of values	Product A
	Where Machine?	Section from set of values	Line 1 - Station 40
	Who?	Section from set of values	Mr. Smith
	When?	Date from 01.01.00 on	28.05.2016
	How often?	Section from set of values	Every minute
	Why Problem?	String	Product NOK, risk of shortcircuit
	Problem created by	ID Number from 0 to 9999	254
Component	Taxonomy position	Value from Taxonomy	Material
Function	Taxonomy position	Value from Taxonomy	Be within specification/Shape
Failure	Taxonomy position	Value from Taxonomy	50% fulfilment
Context	Taxonomy position	Value from Taxonomy	Process/Casting
	Temperature	From 300 to 400 °C	350
	Taxonomy position	Value from Taxonomy	Man/Operator
	Experience	From 0 to 99 years	3
Solution	Containment	String	Stop production and control produced units since last inspection
	Corrective	String	Not defined
	Preventive	String	Not defined

4 Ontological Model Implementation: MPS Process Flow and MPS System Architecture

The presented ontology is used as basis to define the main data structure to be managed by a prototype MPS software system, with the aim of providing production operators with an MPS tool based on the 8D method to be used during their daily MPS activities. The system must allow capturing and reusing knowledge directly at the shop floor level. A first prototype of the MPS system was developed to support an MPS process flow based on the 8D method (Fig. 2). The system is currently under implementation as case study in the company Exide Technologies, a global provider of stored electrical energy solutions (i.e. batteries and associated equipment and services) for transportation and industrial markets, with several production plants in Europe and USA running similar processes, which could benefit from this work.

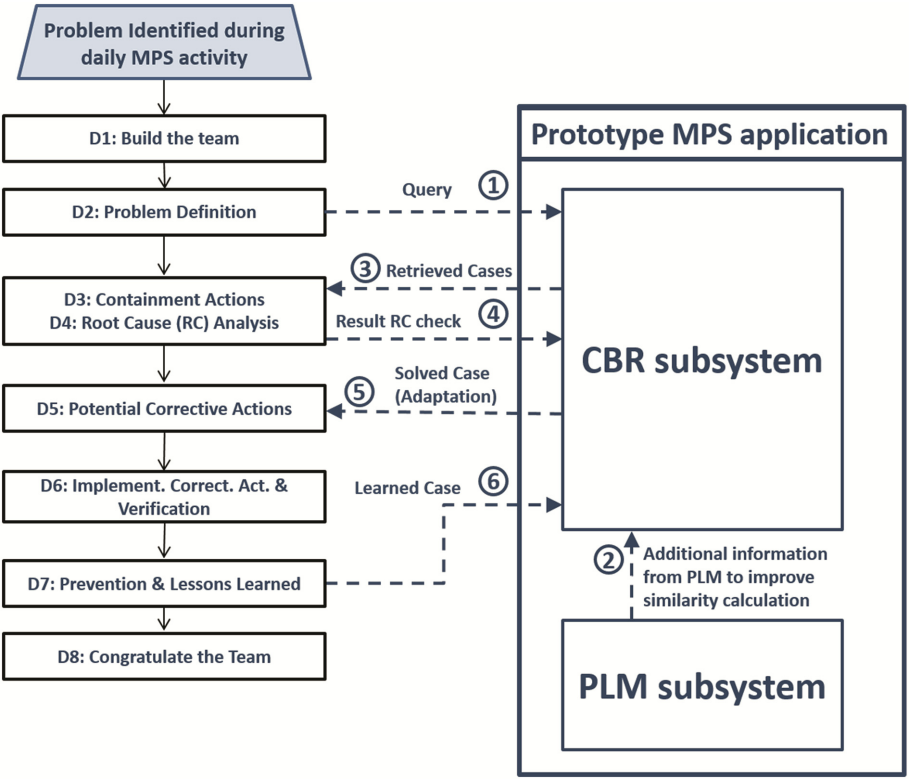


Fig. 2. MPS process flow with the proposed prototype MPS system

The case study considers that during the MPS activity, the MPS personnel must identify a problem that prevents from reaching the defined production targets. The MPS personnel start the analysis following the eight steps of the 8D method. The objective

is, instead of carrying out the method in a paper-based way, the MPS personnel must use the prototype MPS system.

The main characteristic of the targeted users for this software system (i.e. operators) is their lack of deep knowledge about the processes and products. In some cases, the operators are temporary workers, due to seasonal increases of production volumes, and they have no knowledge at all about the production process and/or resources. This profile of the future users of the prototype MPS system means that the application must be able to provide automatically as much information as possible to compensate their lack of knowledge. The user must be able to create a problem query, input the data associated to it (i.e. component-function-failure plus some additional basic information such as date, line, or product, where the problem is found), and the system should connect with the PLM system to extract automatically all existing context information of the problem.

The PLM solution selected for the development of the prototype was Aras Innovator. This PLM application provides an Application Programming Interface named AML (Aras Markup Language) that allows extracting data from the PLM database.

The PLM system has to be customized to fit into the defined ontological model, and it has to be configured to retrieve the requested information in a format understandable by the CBR system. According to the PFMEA methodology [2], used as basis in the proposed ontology, there are six types of components: process, machine, material, man environment and method. The initial configuration of Aras Innovator provides an item type called "Parts" to contain any mechanical design element, so the components of type material and machine can be of that type. The PLM main types has to be extended with four additional items called "Manufacturing Process", "Manufacturing Man", "Manufacturing Method", and "Manufacturing Environment" to contain the rest of component types. Both the part type and the new types need to be created or extended with the following general attributes:

- Name.
- Component number. It represents the reference number in the PLM system.
- Revision number.
- State. It has two alternatives 'released' or 'not released'.
- Effective date. From which date on the item has the state released.
- Classification. Represents the position of the component in the Context taxonomy.
- List of Methods. It contains links to items of type Manufacturing Method.
- Item Nature. It is an attribute to indicate if the component where the problem occur is real or abstract. The 'abstract' alternative allows indicating a general family of products or production lines instead of a specific one. Therefore, specific elements will be tagged with "Real", and general ones with "Abstract". This attribute is used when searching for the problem context data in the PLM application.

The item Manufacturing Method is associated to the PFMEA component Method, which represents the defined procedures or standards. In the proposed model, Manufacturing Method contains the technical specification associated to an item (e.g. Part or Manufacturing Process). Part of this technical information can be common for a whole family of components (e.g. a family of hex bolts with a specific diameter and thread where each one distinguishes from the others in the length). Since the Manufacturing

Method is the container of technical data, each of the other five items (component, machine, process, man and environment) have the attribute called "List of Methods". This attribute allows specifying the links to several Manufacturing Methods containing technical information from multiple levels within the family structure of the component (e.g. Manufacturing Method for all hex bolts made of stainless steel, Manufacturing Method for all hex bolts with standard thread and diameter 5/8″-11, and Manufacturing Method for the hex bolt with length 6″).

The definition of a Manufacturing Method is based on the attribute "List of Manufacturing Parameters". A "Manufacturing Parameter" is the smallest data unit. It contains the type of a single attribute (e.g. pressure, temperature, or experience years), its limit type (i.e. max, min, nominal, or not applicable), its value (either numerical or a selection from a set of possible values), and its measurement unit. The attribute "List of Manufacturing Parameters" contains an open list where an unlimited number of items of Manufacturing Parameter type can be indexed. For example, an instance of the item Manufacturing Process "Casting" will have a link to an instance of an item Manufacturing Method "Casting method", and "Casting method" will contain a list of instances of the item type Manufacturing Parameter, such as "Pressure/10/bar", or "Temperature/300/°C".

Back to MPS process flow (Fig. 2), based on the information introduced by the user, the MPS system must search for the involved components in the PLM subsystem. Once the components are identified, the value of attributes to be used as context information of the problem should be extracted. For example, in a problem related to a component of type Man (i.e. operator) the MPS system should get, from the PLM subsystem, the experience of the operator, but also the type of process and machine where the operator works. In the case of a problem related to a component of type Process, the MPS system should get, from the PLM subsystem, data contained in the associated "Manufacturing Method", for instance the parameters that define the nominal pressure or temperatures at which the process has to run, and but also the type of material that is processed.

The input for the CBR subsystem is the problem description introduced by the MPS system user together with the context information extracted from the PLM subsystem. The CBR subsystem is responsible for providing to the MPS personnel with a list of similar solved problems identified in the case base. The similarity calculation makes use of the problem description and the context information. The open source software myCBR was selected for the CBR subsystem.

As it was previously mentioned, the initial set of manufacturing problems to fill the case base of the CBR subsystem is derived from the PFMEA analysis of the production lines. The PFMEA analysis is conducted during the development phase of the manufacturing processes, and therefore it is available prior to the start of the production. This represents a quite significant set of possible failures that allows having similarity results from the start. As soon as the production starts and the MPS personnel start reporting problems by using the MPS system, the case base will increase. The use of proposed MPS system requires not only the report of problems, but also, the report of the solution to such problems (i.e. the feedback step of the 8D method).

Once the MPS process flow and the main subsystems of the proposed MPS system are discussed, it is then time to introduce the MPS system architecture. As it was

previously mentioned (Sect. 2.3), this work takes as reference the SEASALT architecture. Figure 3 shows the architecture of the proposed MPS system.

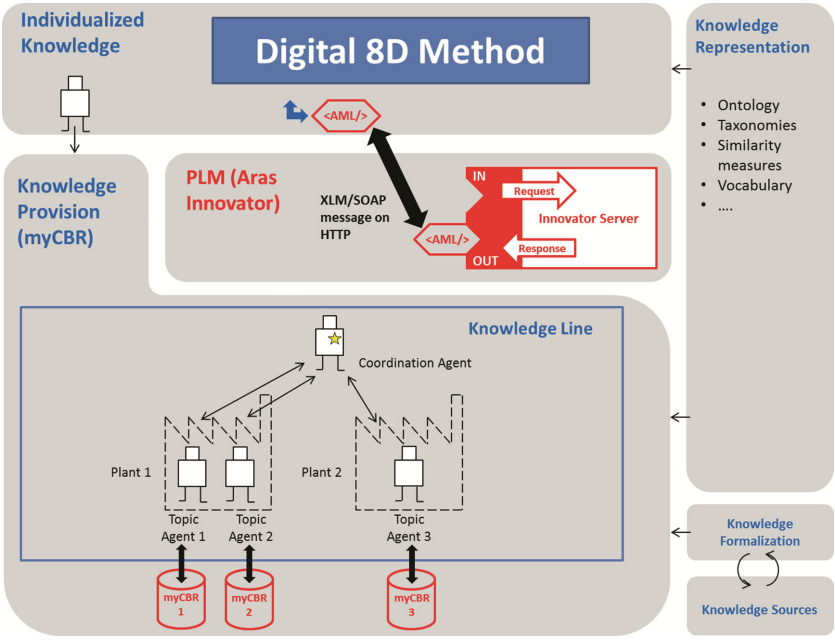


Fig. 3. SEASALT architecture adapted to the proposed MPS system

In this work, the SEASALT architecture was simplified. The SEASALT architecture includes also the modules Knowledge Formalization and Knowledge sources, which are not included as such in this work. The Knowledge Representation module corresponds with the ontological model (Sect. 3). The implementation is made of three different types of agents [16] (see Fig. 3):

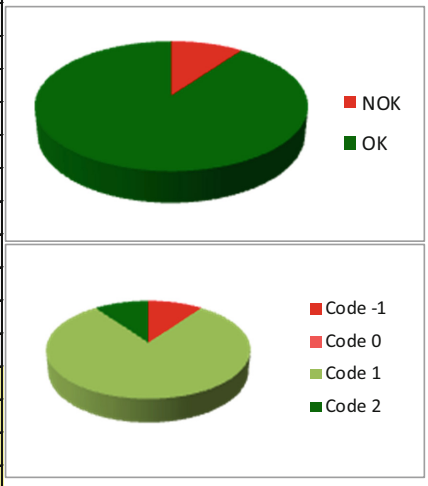
- Individualized Knowledge Agent. It is located in the SEASALT Individualized Knowledge module. It is responsible for capturing and showing information to the MPS system user through the Graphical User Interface (GUI). In the prototype, the GUI is developed to represent a digital form of the 8D method. It also includes the needed interface to collect the problem context information from the PLM subsystem. The number of Individualized Knowledge Agents corresponds with the number of the MPS system users. Such agents are hosted in the devices located directly at the production lines where the users are located (e.g. PC, smart touchscreen, or even a smart phone).
- Topic Agent. It is located in the SEASALT Knowledge Provision module. It is responsible for calculating similarities through the CBR subsystem, and proposing the best solution from its specific case base. The number of Topic Agents correspond with the number of production units. Each Topic Agent is hosted in a central device of its corresponding production unit (e.g. PC or Server).

- Coordination Agent. It is located in the SEASALT Knowledge Provision module. It is responsible for the communication coordination among agents, and the selection of the best solution among the ones proposed by the Topic Agents.

In the implementation of the MPS system, the different agents are deployed across different manufacturing plants of the company (locations), and inside each location, across the areas with different manufacturing processes (production units). In this way, each agent hosted in a specific production unit of a specific location will be able to communicate and to interchange information with all the other agents hosted in different production units and locations through the intranet of the company by sending HTTP-based MTP (Message Transport Protocol) messages [14].

Table 2. Results of initial validation

Initial validation: German Plant - Wet Filling		
	Code	Similarity %
A01	1	85,2
A02	1	83,8
A03	1	82,2
A04	1	85,2
A05	1	77,1
A06	-1	0
A07	1	70,1
A08	1	67,2
A09	2	84
A10	1	72,8
Summary		
NOK	1	10%
OK	9	90%
Code -1	1	10%
Code 0	0	0%
Code 1	8	80%
Code 2	1	10%



Meaning of codes

-1 = No solution useful out of system proposals but no similar case in case base

0 = No solution useful out of system proposals even though there are similar cases in case base

1 = One solution directly useful out of system proposals

2 = One solution useful out of system proposals with adaptation by user

5 Results, Conclusions and Future Work

The proposed ontological model was applied in the development of a prototype MPS system, which integrates a PLM subsystem and a CBR subsystem. The system follows the 8D method and takes results from PFMEA analyses as initial case base.

The developed prototype has been tested in the company Exide Technologies with an initial case base containing 72 cases. A single process and a single production plant have been selected for this first validation step, which represents the easiest level of complexity for the system, since the cases in the prototype were all collected in the same process and plant. 10 problems found on the shop floor have been analyzed in parallel by the system and experts. As it is showed in the Table 2, 80% of the results obtained for the queries were similar to the answer provided by experts, and an additional 10% could guide to the solution by adapting the proposal of the system to the context of the problem under analysis. The results obtained showed the importance of the problem context information, defined in the PLM subsystem, in the case similarity calculation. A second version of the prototype MPS system is currently under development to be tested in different processes and production plants in parallel.

An advantage of the proposed MPS system, which has been realized during the execution of the tests, is the access and reuse by operators of solutions to problems already identified in the PFMEA analyses of their lines. This is significant, because without the support of the prototype MPS system, such reuse rarely happens, mainly because of the difficulties in finding and analyzing information in the complex document of a PFMEA analysis. The population and maintenance of the system is conducted by a role named Knowledge Engineer. The current version of the application allows uploading cases by means of a csv format file.

Two possible barriers were identified during the implementation of the first prototype of the MPS system. The first one is the requirement of having a PLM system, where information of the products, their processes and resources (PPR) of the company should be stored. The second one is the need of having access to the MPS system at the shop floor level. Nevertheless, the current trend of digitalization in the industry (e.g. Industry 4.0 initiative) [17] should help to overcome these possible barriers.

A possible future work is the development of the Knowledge Source and the Knowledge Formalization modules, defined in the SEASALT architecture, to extract automatically knowledge from PFMEA analyses. In that direction, some research works propose the use of SysML, to create a system model, where artifacts contain FMEA information, and the use of Prolog engine to query the created model to derive FMEA results [18].

References

1. Bhuiyan, N., Baghel, A.: An overview of continuous improvement: from the past to the present. Manage. Decis. **43**(5), 761–771 (2005). https://doi.org/10.1108/00251740510597761
2. VDA: Qualitätsmanagement in der Automobilindustrie – Qualitätsmanagement-Methoden Assessments. Verband der Autoindustrie (VDA) (2015)
3. Liu, D.R., Ke, C.K.: Knowledge support for problem-solving in a production process: a hybrid of knowledge discovery and case-based reasoning. Expert Syst. Appl. **33**, 147–161 (2007). https://doi.org/10.1016/j.eswa.2006.04.026
4. Sowa, J.F.: Knowledge Representation - Logical, Philosophical and Computational Foundations. Brooks Cole Publishing Co., Pacific Grove (2000). ISBN 0 534-94965-7
5. Stark, J.: Product Lifecycle Management. Springer International Publishing, Switzerland (2015). https://doi.org/10.1007/978-3-319-17440-2. ISBN 978-3-319-17440-2

6. Richter, M.M., Weber, R.: Case-Based Reasoning: A Textbook. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40167-1. ISBN 978-3-642-40167-1
7. Foguem, B.K., Coudert, T., Béler, C., Geneste, L.: Knowledge formalization in experience feedback processes: an ontology-based approach. Comput. Ind. **59**(7), 694–710 (2008). https://doi.org/10.1016/j.compind.2007.12.014
8. Dittmann, L., Rademacher, T., Zelewski, S.: Performing FMEA using ontologies. In: 18th International Workshop on Qualitative Reasoning, Evanston, USA, pp. 209–216 (2004)
9. Ebrahimipour, V., Rezaie, K., Shokravi, S.: An ontology approach to support FMEA studies. Expert Syst. Appl. **37**(1), 671–677 (2010). https://doi.org/10.1109/RAMS.2009.4914711
10. Cambria, E., White, B.: Jumping NLP curves: a review of natural language processing research. IEEE Comput. Intell. Magaz. **9**(2), 48–57 (2014)
11. Bertin, A.: Intégration d'un système de Retour d'Expériences à un PLM. Doctoral dissertation - Laboratoire Génie Production (LGP) - Ecole Nationale d'Ingénieurs deTarbes - Université de Toulouse (2012)
12. Bertin, A., Noyes, D., Clermont, P.: Problem solving methods as lessons learned system instrumentation into a PLM tool. In: 14th IFAC Symposium on Information Control Problems in Manufacturing, Bucarest, Roumanie, 23–25 May 2012 (2012). ISBN: hal-01005542
13. Bach, K.: Knowledge acquisition for case-based reasoning systems. Ph.D. thesis, University of Hildesheim (2012)
14. Reuss, P., Althoff, K.D., Hundt, A., Henkel, W., Pfeiffer, M.: Multi-agent case-based diagnosis in the aircraft domain. In: Workshop Proceedings - 23rd International Conference on Case-Based Reasoning (ICCBR 2015), pp. 43–52 (2015)
15. Kepner, C.H., Tregoe, B.B.: The New Rational Manager - An Updated Edition for a New World. Princeton Research Press, Princeton (2008). ISBN 978-0-9715627-1-4
16. Bellifemine, F.L., Caire, G., Greenwood, D.: Developing multi-agent systems with JADE. Wiley Series in Agent Technology. Wiley, England (2007)
17. Cearley, D.W., Walker, M.J., Burke, B.: Top 10 Strategic Technology Trends for 2016: At a Glance. Gartner (2015)
18. Scippacercola, F., Pietrantuono, R., Russo, S., Silva, N.P.: SysML-based and prolog-supported FMEA. In: 2015 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), pp. 174–181. IEEE (2015)