This regular paper was presented as part of the main technical program at IEEE ETFA'2011

Risk Minimization in Modernization Projects of Plant Automation – a Knowledge-Based Approach by means of Semantic Web Technologies

Martin Strube[1], Stefan Runde[2], Helmut Figalist[3], Alexander Fay[1]

Helmut Schmidt University, Hamburg[1]
Institute for Automation
Holstenhofweg 85
22043 Hamburg, Germany
martin.strube@hsu-hh.de
alexander.fay@hsu-hh.de

Siemens AG[2]
I IA IA&DT-ATS 3 1
Oestliche Rheinbrueckenstr. 60
76187 Karlsruhe, Germany
stefan.runde@siemens.com

Siemens AG[3]
I IS MT PEP PA A
Schuhstr. 60
91052 Erlangen, Germany
helmut.figalist@siemens.com

Abstract

In high-wage countries the number of Greenfield projects for plant automation is decreasing. In contrast to this, plant modernization becomes more and more important. The estimation of the costs for a re-engineering of the existing plant automation is an error-prone task which has to be done in the bidding phase of a modernization project.
This article describes a knowledge-based approach to reduce the risk potential in the bidding phase of plant modernization projects. Based on a concept for rough plant modeling in CAEX and technologies of the semantic web a concept for a software assistance system is presented.

1. Introduction

In North America and Central Europe the Greenfield business for plant projects is decreasing. In contrast to this development, plant modernization is a major necessity for producers, especially in high-wage countries, to enable them to meet the continuously rising level of productivity. “During its entire economic life of 40–50 years, a plant will be repeatedly upgraded, rationalized, renewed; and adapted and optimized with increasing frequency to keep up with an ever-faster change of plant requirements.” [1]. As a consequence the market for plant automation modernization becomes more and more important in high-wage countries.

As a special challenge in plant modernization, engineers already need to identify the impact of changes to the overall functionality of the plant in the bidding phase. Based on this knowledge the effort required for necessary hardware and software adjustments needs to be calculated to estimate the costs, schedule and risks of the modernization project. Because of higher costs for engineering and installation in comparison to costs for the system, this is the most important step within the bidding phase.

Nowadays, the proceedings in the bidding phase of modernization projects can be characterized as follows: experienced engineers review a large amount of heterogeneous planning documents like layout plans, signal lists and circuit diagrams to create a structured plant overview. For summarizing the results of this review, Microsoft Office is the tool most often used. On the basis of Excel spreadsheets, Word documents and automation overview figures the engineers try to predict the modernization costs, risks and schedule as exactly as possible.

Thus, engineers have to do this work predominantly manually by means of a suboptimal software support, which requires huge manual effort. Moreover, this fact results in an increased risk potential for the modernization planning.

Each plant modernization project has impact on the installed plant automation. That means plant automation adjustments are part of plant modernization projects, e.g. to renew obsolete hardware, to implement new automation functions or to install high quality controls for a better system performance. The more complex existing plant automation is, the more serious mistakes occur in the previously described procedure. Even highly experienced engineers are not able to identify all functional relations between the elements of the existing plant automation by reviewing the large amount of heterogeneous planning documents without any methodology. This risk potential in the bidding phase causes problems for automation suppliers, end customers, system integrators etc. If they overlook some functional relations between the elements of the existing plant automation, their calculation could be too low. As a consequence, the modernization project

978-1-4577-0018-7/11/$26.00 ©2011 IEEE

could be a loss-making operation which could equalize the profit of some successful modernization projects.

Due to the increasing importance of plant modernization, new methods, models and tools are required to reduce the risk potential in the bidding phase. The issues that have to be solved for an improved modernization planning are briefly described in the next Section.

2. Need for action

The key prerequisites for risk minimization in modernization projects for plant automation are knowledge about the "as build" status of the plant and domain-specific (human) expert knowledge of engineering in plant automation modernization. Due to the complexity of plant automation modernization, the planning engineers should already be assisted by software in the bidding phase. To achieve this target it is necessary fulfill the following requirements:

1. The "as build" status of the plant to be modernized needs to be captured systematically in a formalized way.
2. It has to be defined how to structure and specify the expert knowledge with regard to engineering in plant automation modernization planning.
3. It needs to be shown how the results obtained with regard to the previously described requirements can be combined for a software assistance system.

Recent research projects of different domains like process or manufacturing automation show some improvements. Firstly, plants can be described by using a neutral description language. Secondly, these languages arise the possibility for a more efficient engineering (cf. [2], [3], and [4]). An example for such a neutral description languages has already been established for the data exchange between different engineering domains is CAEX (cf. [2], [3]). By use of CAEX for describing the "as build" status of the plant, which is the first step in a plant automation modernization project, the advantages listed above can be transferred to the modernization business.

If it is possible to describe the functional dependencies between different elements of the plant automation in a machine-interpretable way, software assistance systems can be used to reduce the risk in the bidding phase. Therefore, such software assistance systems need to include the human expert knowledge about plant automation modernization and algorithms to apply these knowledge to the description of the plant "as build" status.

Typically, the human expert knowledge (about plant automation modernization) to work on plant descriptions is strongly embedded as source code within these software assistance system. It is almost impossible to change this already described expert knowledge and add new knowledge respectively. A knowledge-based system (KBS) explicitly allows the separate description of human expert knowledge as a part of the so-called knowledge-base. With this, it is possible to change the already described expert knowledge and add new knowledge, respectively.

KBSs are already used in process, factory and building automation for different use cases such as software code generation and safety analysis. Thereby, they have shown their suitability as software assistance system and engineering assistance to counteract the rising engineering effort (cf. [4], [5]).

The combination of both aspects, i.e. capturing the "as build" status by the use of CAEX and the software assistance by means of a KBS, can help to reduce the risk in the bidding phase.

In the following, the specification of neutral functional plant description for the modernization business based on CAEX (cf. Section 3) and a KBS by means of semantic web technologies (cf. Section 4) is described. Furthermore, it is shown how both approaches can be combined to reduce the risk potential in the bidding phase of plant automation modernization projects (cf. Section 5).

3. CAEX-based functional plant description

Plant documentation is rising up from the idea of the plant up to the planning, realization, commissioning and maintenance. This can only be achieved by the cooperation of different domains, like e.g. electrical engineering, mechanical engineering and plant automation. Each engineering discipline uses specific tools in its area of responsibility. As a result, the plant documentation is an extremely heterogeneous information package with a growing number of inconsistencies during plant life time.

Based on a concept for functional plant description, which addresses to the special needs of modernization business (cf. [6]), the following section of this paper shows how to capture the "as build" status of the plant by the use of CAEX.

3.1. Elements of a functional plant description

A process optimization is the main reason for plant modernizations (often combined with mechanical changes). Additionally, the substitution of old automation systems can be mentioned. Reflecting that, the technological process should be the common base for all engineering domains involved in a modernization project. Therefore, the first element of a functional plant description is a formalized description of the process. For further information about the Formalized Process Description see [7]. The plant layout, which shows the technical resources like crank shears, rolling mill stands, presses and furnaces, is the second element of the functional plant description. Next to these elements the automation functions and the automation devices, like programmable logic controllers (PLC) or control cubicles, are important for a functional plant description. In this context, automation function means a software module which controls a sub-process. In the concept of functional plant description sensors and actors are seen as a part of the technical resources.

3.2. Dependencies between these elements

The most important aspect of the functional plant description is the modeling of the functional links between the different elements described before. These functional links, in form of dependencies and assurances, are created by two types of relations between plant components. On the one hand, there are communication relations between devices which interchange signals. On the other hand, there are so-called causal relations in complex systems like industrial plants. The main difference between both relations is that communication relations can be taken almost directly from parts of the plant documentation (like the circuit diagrams), while causal relations need to be identified by human experts. The following two examples will illustrate the causal relations. A typical causal relation can be identified between an automation function and the PLC on which this function runs, because the performance of the PLC affects the performance of the automation function. Another causal relation can be defined between an automation function and the technical resource which is controlled by this automation function because changes in the automation function affect the functionality of the technical resource.

3.3. The 4-layer plant model

The 4-layer plant model which is presented in [6] shows how the elements of a functional plant description can be connected to each other by the use of causal relations and communication relations (cf. Figure 1).

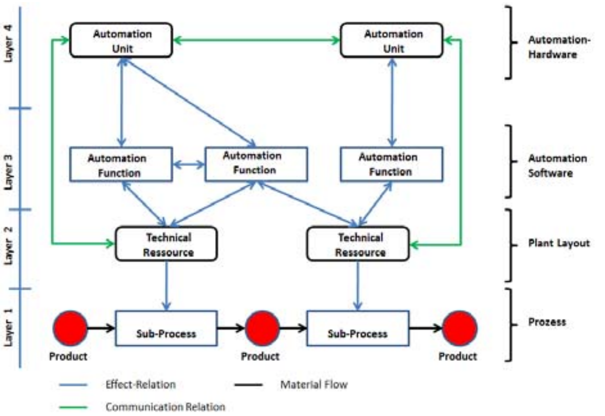

Fig. 1. 4-layer plant model.

The 4-layer plant model presents the factual knowledge about the plant. This knowledge is necessary to identify the impact of changes to the overall functionality of the plant. To identify this impact the planning engineer can remove all elements from the plant model which should be modernized. Thereby, some communication or causal relations will be interrupted as shown in Figure 2.

All elements of the plant model that are affected by this modernization operation can be identified through interrupted communication or causal relations.

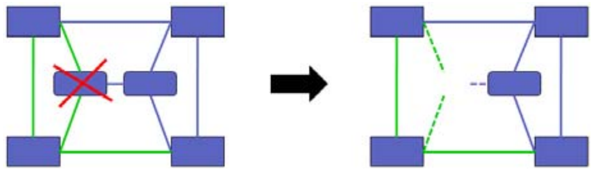

Fig. 2. Interrupted relations.

3.4. Capturing the “as build” status

Capturing the “as build” status of the plant begins with an abstract process description (layer 1). It can be refined later on. Next, each process step of the process description needs to be connected to the technical resources for realizing this sub-process, which must be at least one. The technical resources represent the plant layout (layer 2). In the next step, the following two questions for each technical resource need to be answered:

1. Which sensors and actuators, mounted into the technical resource, not only serve for local display or manual operations?
2. Which parts of the technical resource must be controlled by an automation function?

With regard to the answers to both questions the associated interfaces (communication interfaces and causal interfaces) are modeled at the technical resource.

When that part has been finished, modeling of automation functions (layer 3) begins. This should be done in a process-oriented way, starting with the technical resources connected to the first process step. For each causal relation interface at a process step, an associated automation function must be identified, modeled and connected. After that, for each technical resource the automation functions need to be reviewed in detail. Human experts have to analyze, which automation functions are related to each other. These automation functions are connected by a causal relation. In the next step, the associated automation devices (layer 4) must be identified, modeled and connected for each automation function. This should also be done in a process-oriented way. In the last step, all communication interfaces from the technical resources are connected to the associated automation device interfaces.

By following the modeling methodology described before, the systematic analysis of heterogeneous plant documentation is possible.

3.5. Functional plant description by the use of CAEX

CAEX, standardized in the IEC 62424 standard [8], enables a systematic life-cycle accompanying data exchange between different CAE (Computer Aided Engineering) systems. Capturing the plant "as build" status by the use of CAEX makes it easier to interchange this knowledge about the plant with different domain specific engineering tools in later planning processes.

3.5.1. CAEX basics

CAEX is a meta model for vendor- and tool-independent structuring and categorization of CAE data. Nevertheless, it provides mechanisms to refer to vendor-specific information. Furthermore, CAEX does not define domain-specific semantics.

The four fundamental elements of CAEX are described in the following (for further information, see e.g. [9]).

- Within the *Interface Library*, possible interfaces for e.g. material flow, energy flow or information flow can be defined.
- By means of the *Role Library*, plant entities and functions can be described by symbolic placeholders (roles). Interfaces can be assigned to roles by referencing these to the interface library.
- Connected systems (units) consisting of functions and/or plant entities can be defined in the *System Unit Library* by referencing roles to the role library and using interfaces from *the Interface Library*.
- In the *Instance Hierarchy* the structure of a plant with its plant entities and functions (including the automation systems) can be described. Therefore, roles of the role library and/or units of the unit library can be referenced.

In principle, CAEX does not depend on a special representation form such as XML (E**x**tensible **M**arkup **L**anguage). Nevertheless, due to its well readability and exchangeability, XML has been selected as the representation language for modeling with CAEX.

3.5.2. Transformation of plant description to CAEX

All elements of the 4-layer plant model can be modeled in CAEX the same way. Each element, like e.g. a sub-process, a product, an automation function or a technical resource, is described by an *Internal Element* in CAEX. These *Internal Elements* are defined through their attributes (e.g. real-time requirements for an automation function) and interfaces (cf. Section 3.2). Interfaces can also have attributes. The interpretability for computers is achieved by the allocation of role classes to each element, e.g. *Automation Function* or *Programmable Logic Controller*. Each element can be decomposed by adding child *Internal Elements* to it, which allows a rough modeling in the early project phases. As an example for decomposing a technical resource, Figure 3 shows a "Finishing Mill Stand" modeled with the AutomationML Editor (AML Editor) (cf. [10]). The tool AutomationML Editor can be used for CAEX based modeling.

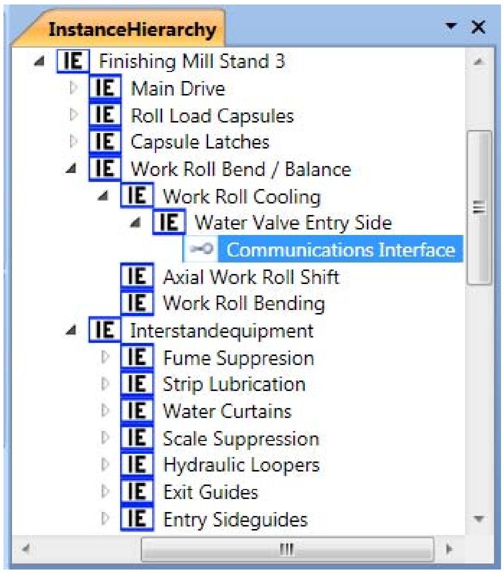

Fig. 3. A CAEX based Finishing Mill Stand model.

To describe the functional dependencies between these elements they need to be connected by causal relations and communication relations as it is explained in Section 3.2. For modeling these relations in CAEX it is necessary to define the corresponding interfaces in the *Interface Library*. Figure 4 shows an example of different types of communication interfaces defined in an *Interface Library* created with the AML Editor. As an example, the frame in the lower right of Figure 4 shows attributes of the highlighted communication interface *OPC Tag*.

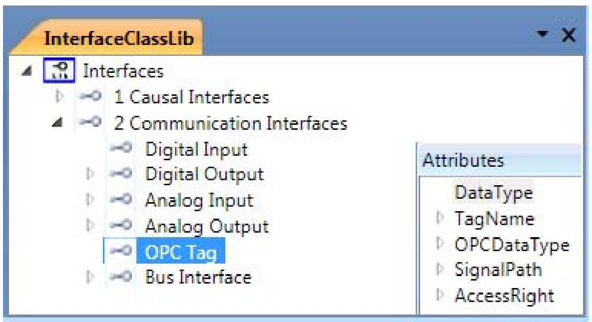

Fig. 4. Interface library.

A connection between two interfaces of the same type represents a causal relation or a communication relation. Using the AML Editor it is possible to create such relations via drag & drop from one interface to another.

4. Knowledge-based system by means of Semantic Web technologies

The proposed KBS for engineering (of automation systems) is based on the well-proven concepts of [11], [12]. The KBS has been successfully used for engineering in the building automation domain (cf. [13]). This KBS has the potential to fulfill the second requirement mentioned in Section 2. For further information we refer [14].

Basics of the KBS concept (cf. Section 4.1) and execution (cf. Section 4.2) are described in the two following sections. Details and examples for the implementation of a software assistance system for risk minimization in the bidding phase by the use of this engineering KBS are described in Section 5.

4.1. Knowledge-based system concept

Generally, KBS are subdivided into the two main modules: control system and knowledge base. This is shown in Figure 5 for the case of KBS for engineering.

The *Class Knowledge* (domain-specific, declarative knowledge) can be described by the use of elements (semantic terms) and functional dependencies (correlations between the elements). The *Factual Knowledge* is inferred from the Class Knowledge (declarative knowledge). By means of the Factual Knowledge (case-specific, declarative knowledge) facts of a plant with its automation system can be described. Moreover, the *Rule Knowledge* (domain-specific, procedural knowledge) allows the "production" of new knowledge based on the Factual and Class Knowledge.

The *control system* of the KBS, not the control system and the automation station respectively of the automation system, contains program code for the program execution, problem-solving strategies, and interfaces to the user and expert. It consists of the parts *interviewer module*, *explanation module*, *knowledge acquisition module*, *control module*, *interface module*, as well as the *inference engine* (cf. Figure 5). The software code of the inference machine and the interface module are independent of the knowledge represented within the several parts, i.e. classes, factual and rule knowledge. The inference machine applies rule knowledge to the class and factual knowledge and allows to (re)write new factual knowledge or to search on the factual knowledge. The various engineering data (cf. Section 3) can be imported and exported with the interface module via the appropriate data exchange format.

Because the focus is on the application of the described KBS, the other parts, i.e. interviewer, explanation, and knowledge component, are not further described within this paper.

4.2. Combining the Knowledge-Based System Concept and Semantic Web Technologies

Challenges in the engineering of automation systems and modernization planning, respectively, correlate with the challenges in the Semantic Initiative; an initiative of the W3C (World Wide Web Consortium). As an example it is in both cases a challenge to expand the existing descriptions (internet and plant information, respectively) for better enabling computers and people to work in cooperation [15]. Therefore, it is an obvious approach to investigate the developed technologies within the Semantic Web Initiative. Many technologies are developed not only for use in the internet but also in industrial applications [16].

In the following, the main technologies and tools of the Semantic Web Initiative are described with regard to the KBS for engineering. Further information on the Semantic Web and its technologies can be retrieved from [15].

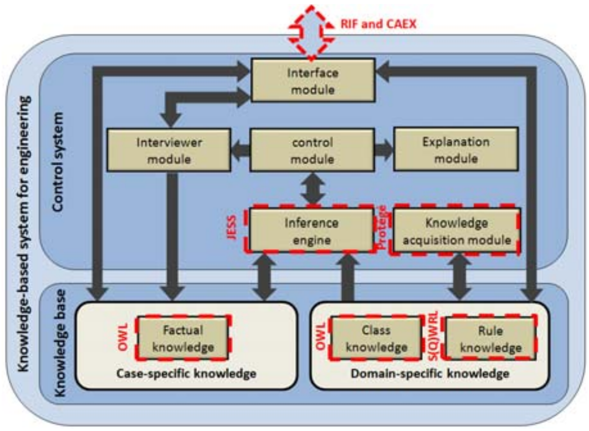

Fig. 5. Knowledge-Based System by means of Semantic Web Technologies.

The major Semantic Web technology OWL (Web Ontology Language) has been developed to represent machine-interpretable semantic content [17]. It enables a formal, semantic specification of the Class Knowledge and Factual Knowledge of the KBS. OWL distinguishes between the terminological level and the assertion level. The class knowledge can be described at the terminological level of OWL. The assertion level of OWL ontologies yields the possibility to represent the factual knowledge. Therefore, the class knowledge can be instantiated at this level.

By means of *SWRL* Rules (Semantic Web Rule Language) [18], the rule knowledge is represented. The rules or queries for a search are declared by *SQWRL* (Semantic Query-Enhanced Web Rule Language) [19].

SQWRL is a SWRL-based query language. Both rule languages are used explicitly for OWL ontologies and are part of the RuleML (Rule Markup Language), which is currently employed and tested in engineering [4]. Based on the RuleML initiative, RIF (Rule Interchange Format) [20] is considered as a future standard of the W3C.

Using formal description languages of the semantic web, such as OWL, SWRL und SQWRL, allows the application of existing algorithms and software tools for inferencing and queries. In contrast to this, proprietary algorithms have to be implemented for inferencing and queries based on CAEX. This approach results in a huge effort with regard of the complex challenge of knowledge representation [11].

After a detailed analysis of different inference machines, which support OWL and S(Q)WRL, the JAVA-based *Jess* [21] has been selected as the inference machine for the engineering KBS. Based on the OWL ontology, Jess inferences new knowledge by use of the SWRL Rules and adds it to the ontology. Jess also allows queries in SQWRL[1].

Semantic Web technologies can therefore be used to implement the core of the KBS. However, the Semantic Web technologies are less suitable for the other modules, which are implemented in C#. Therefore, the Jess inference for the JAVA platform has been integrated in the C# platform. The software code of the interface module has been implemented independently of special KBS applications, as shown in [14], where CAEX acts as interface for class and factual knowledge, and RIF as interface for the rule knowledge. The interview module, explanation module, and knowledge acquisition module are dependently implemented to the application "software assistance system for risk minimization".

5. Approach of a software assistance system for risk minimization in the bidding phase

The combination of both approaches – previously described in Section 3 as plant description by use of CAEX and Section 4 as the Semantic Web KBS – makes it possible to use the advantages of data exchange resulting from CAEX to transfer the information from the bidding phase to the engineering phase of a modernization project. Furthermore, it allows using the benefits from developments of the Semantic Web technologies for an efficient development of a software assistance system.

This Section addresses the third requirement listed in Section 2 and describes how to combine both previously described technologies for a software assistance system. For a better understanding, the following section illustrates how knowledge-based software can assist the planning engineers from capturing the "as build" status of the plant, to developing a new automation solution. For both aspects, use cases are presented.

5.1. Use cases for software-assisted bidding

In a first step, the planning engineers need to capture the "as build" status of the plant as described in Section 3.4. Due to the heterogeneous and inconsistent plant documentation, this task is very prone to errors. That is why, the captured plant model should be validated – *use case 1*. One example is that each sensor or actuator needs to be connected to an automation device by a communication relation and each technological resource needs to be connected to an automation function by a causal relation. Another task is the consistency check of each modeled relation by comparing their interface attributes, e.g. the attribute bus protocol needs to be identical at two connected bus interfaces.

In a second step the planning engineers identify elements that need to be modernized to meet the customer requirements. These elements have to be removed from the plant model and new elements have to be implemented into the model. By removing them, some communication relations and/or causal relations will be interrupted. After the implementation of the new elements this interrupted relations need to be reconstructed. This should be validated by the use of consistency checks – *use case 2*.

The actions to be taken within these two use cases are routine tasks, which lead to failures and increased engineering costs in modernization planning of plant automation. Automated mechanisms based on the KBS minimize such failures and with that engineering costs.

Some details of the implemented "Semantic Web KBS" (cf. Section 4) for plant automation modernization planning consisting of *Knowledge Base* (cf. Section 5.2) and *Control System* (cf. Section 5.3) are described in the following sections.

5.2. Knowledge Base

The *Class Knowledge* is domain-specific for each industry branch (cf. Section 3). The class knowledge of the KBS discussed in this paper is described with regard to the functional plant description (cf. Section 3).

Therefore, roles of the Role Library in CAEX, e.g. Sensor, PLC and Automation Function are so-called OWL Concepts [17]. Interfaces of the CAEX Interface Library are modeled by means of OWL DataType Properties such as Digital Input, Analog Output and Bus Interface. These DataType Properties are also used

[1] Furthermore, this Semantic Web KBS allows the use of SPARQL (SPARQL Protocol and RDF Query Language) as a second possibility for an efficient search in OWL ontologies. As a query language for the Semantic Web, SPARQL fulfils the same role as SQL in the field of relational databases. Semantic Web technologies can therefore be used to implement the core of the KBS.

to describe the elementary attributes for an automation device (e.g. `the device ingress protection for a temperature sensor`) and possible attribute values `(54, 66, etc.)`, for example, which are predefined as RoleRequirements of the associated role class in CAEX. In OWL the causal dependencies (causal relations) between OWL concepts (e.g. between the above-mentioned classes `Rolling Stand` and `Control Cubicle Common Control PLC`) are modeled by use of object properties. The communication relations, that are defined in CEAX by the use of Internal Links, are defined by means of OWL Object Properties (e.g. `hasConnected_PLC` or `isConnected_PLC_of`), which are to be considered by the instantiation as factual knowledge.

The *Factual Knowledge* representing the "as build" status of the plant (cf. Section 3.3) is derived from class knowledge. For example, `Rolling Stand 1` (OWL Individual) `hasconnected_PLC` (OWL Object Property) `Control Cubicle Common Control. PLC 18` from type `SIMATIC TDC, which is_mounted` (OWL Object Property) in `Control Cubicle Common Control` (OWL Individual). In CAEX the appropriate knowledge, which has been automatically transformed to OWL, is described in the Instance Hierarchy by the use of e.g. Internal Elements, Internal Links and Attributes (cf. Section 3.5.1).

The rule knowledge described within the KBS for modernization planning is used to assist the planning engineers in verifying their plant model after capturing the "as build" status – regarding the previously described use case 1 (cf. Section 5.1). The following rule illustrates this approach: **IF** two automation functions (e.g. a function for gauge control gets input from an associated automation function for set point settings), which are connected by a causal relation, are running on different PLCs, **THEN** there must be a communication relation between these PLCs. In SWRL the rule is described as follows:

(PLC (?a) ^ AutomationFunctions(?b,?a,) ^ PLC (?c) ^ AutomationFunctions(?d, ?c,) CausalRelation(?b,?d)
→ *CommunicationRelation(a,c)*

Such a rule and SWRL rule is subdivided into premise – **IF** - and conclusion – **THEN**. The premise and conclusion consist of single atoms. Atoms are e.g. *C(?x)* (here: PLC and AutomationFunctions) and *P(?x,?y)* (here: CausalRelation and CommunicationRelation). C represents an OWL concept or an OWL instance, P an OWL property and ?x,?y are variables.

On the one hand, these rules help to verify the plant model, on the other hand, the rule knowledge can assist the planning engineers also in checking the consistency of the new projected automation solution – regarding the previously described use case 2 (cf. Section 5.1). For example: **IF** a Digital Output is connected to an Analog Input, **THEN** this communication relation is inconsistent; **IF** two Bus Interfaces are connected to each other, **THEN** their attributes bus protocols need to have the same value. These rules are not shown in the SWRL notation in this paper because of the limited space available.

5.3. Control System

Besides the function of coordinating the different KBS modules (cf. Section 4), the control module provides the correct procedure for the modernization planning. The first step is to import the project-specific factual knowledge (plant information) from CAEX plant description (cf. Section 3) into the KBS for engineering. This factual knowledge is based on the class knowledge and can be imported automatically with the CAEX interface and interface module. Furthermore, the rule knowledge can be imported by means of the RIF interface. Of course it is possible to adjust the class, factual and rule knowledge by means of the knowledge acquisition module.

After starting the inference engine, this engine "fires" the rule knowledge to the factual knowledge with regard to the class knowledge. Thus, the software automatically determines inconsistencies and redundancies within the plant description based on the rule knowledge by means of the described rules (cf. Section 5.2).

Example 1: Rule-based attribute check

As a simple example: The assistance software recognizes that a sensor, which is assigned to a rolling mill stand, must at least meet ingress protection class 54. The software sets the value of this attribute in the respective dialogue with the interviewer module and explanation module, but nevertheless gives the planner the possibility to change it.

Example 2: Rule-based logical consistency check

A more complex example: if an automation function is connected to a PLC by a causal relation, then the value of PLC interface attribute real-time performance has to be even to or higher than the Automation Function interface attribute real-time requirement. If an automation function is connected by a causal relation to another automation function which does not run on the same PLC, then there must be a communication relation between the two PLCs. Furthermore, the attributes communication protocols of two connected communication interfaces must be identical.

The assistance software recognizes whether the appropriate connection between the automation function and the PLC is given and both communication interfaces are identical by means of rule knowledge. The software informs the planner of the facts required to solve this problem (interviewer and explanation module), e.g. which connections are given and which communication interfaces are still in use.

The overall results are available in OWL. In addition, these results can be written back to the CAEX plant description by use of the appropriate CAEX interface and the interface module [21], making them available for further processing.

6. Conclusions

Within this article a knowledge-based approach for risk minimization in modernization projects for plant automation is described. It has been shown that plants which have to be modernized can be modeled roughly with focus on the functional dependencies between the plant elements through using CAEX. Furthermore, it has been described that human expert knowledge about plant automation modernization can be structured and formalized. Based on this it was shown that both concepts can be combined to a software assistance system for risk minimization in the bidding phase of plant automation modernization projects.

References

[1] N. Schröder.: Process Plant Markets 2010. Development of the global market for plant-related products and services in the process industries until 2010. Intechno Consulting, Basel, 2003. Available: http://www.intechnoconsulting.com/pdfs/D_Press_PP2010.pdf (last access 01.03.2011).

[2] M. Schleipen, R. Drath, O. Sauer: The system-independent data exchange format CAEX for supporting an automatic configuration of a production monitoring and control system. In Proceedings of IEEE International Symposium on Industrial Electronics (ISIE 2008), Cambridge, 2008.

[3] R. Drath, A. Lüder, J. Peschke, L. Hundt: AutomationML – The glue for seamless Automation Engineering, Proceedings of the IEEE Conference on Emerging Technologies & Factory Automation - ETFA 2008, Hamburg, 2008.

[4] R. Drath, A. Fay, T. Schmidberger: Computer aided design and implementation of interlocking control code. In: Proceedings of the IEEE International Symposium on Computer-Aided Control Systems Design (CACSD'06), Munich, 2006.

[5] K. Güttel, P. Weber, A. Fay: Automatic generation of PLC code beyond the nominal sequence. In: Proceedings of the 13th IEEE Emerging Technologies and Factory Automation (ETFA'08), Hamburg, 2008.

[6] M. Strube, A. Fay, S. Truchat, H. Figalist: Funktionale Anlagenbeschreibung als Basis der Modernisierungsplanung [Functional Plant Description as a basis for modernization planning]. In: Proceedings of „Automation", Baden-Baden, 2010.

[7] The Association of German Engineers – Association for Electrical, Electronic & Information Technologies Guideline VDI/VDE 3682: Formalized process descriptions. Beuth, Berlin, 2005.

[8] International Electrotechnical Commission: IEC 62424 – Representation of process control engineering – Requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools, 2008.

[9] M. Fedai, R. Draht: "CAEX - a neutral data exchange format for engineering data", *atp international*, 2005.

[10] The AutomationML association. Available: http://www.automationml.org/

[11] Feigenbaum, E.A.: A Personal View of Expert Systems. Looking Back and Looking Ahead, Stanford University, Knowledge Systems Laboratory, Department of Computer Science, 1992.

[12] F. Puppe: Systematic introduction to expert systems – Knowledge representations and problem-solving methods, Springer, Berlin, 1993.

[13] S. Runde, A. Heidemann, A. Fay, P. Schmidt: Engineering of Building Automation Systems - State-of-the-Art, Deficits, Approaches. In: Proceedings of the 15th IEEE Emerging Technologies and Factory Automation (ETFA 2010), Bilbao, Spain, 2010.

[14] S. Runde, A. Fay, W.-O. Wutzke: Knowledge-Based Requirement Engineering of Building Automation Systems by means of Semantic Web Technologies. In: Proceedings of the 6th IEEE International Conference on Industrial Informatics (INDIN'09), Cardiff, 2009.

[15] T. Berners-Lee, J. Hendler, O. Lassila: „The Semantic Web", in: Scientific American, 2001.

[16] World Wide Web Consortium (W3C): W3C Semantic Web Activity, 2008. Available: http://www.w3.org/2001/sw/

[17] World Wide Web Consortium (W3C): OWL Ontology Web Language, 2004. Available: http://www.w3.org/TR/owl-features/

[18] World Wide Web Consortium (W3C): SWRL: A Semantic Web Rule Language - Combining OWL and RuleML, 2008. Available: http://www.w3.org/Submission/SWRL/

[19] Protégé: SQWRL, 2009. Available: http://protege.cim3.net/cgi-bin/wiki.pl?SQWRL

[20] World Wide Web Consortium (W3C): RIF Working Group, 2009. Available: http://www.w3.org/2005/rules/wiki/RIF_Working_Group

[21] Jess, the Rule Engine: Jess 2008. Available: http://www.jessrules.com/jess/index.shtml