Plant Asset Management Functions driven by Property Models

Martin Mertens, Ulrich Epple
Chair of Process Control Engineering, RWTH Aachen University
Turmstraße 46, 52064 Aachen, Germany
m.mertens@plt.rwth-aachen.de

Abstract

During the plant's life cycle, plenty of information is generated manually and automatically, not only in the stage of plant operation. Such information concerns all entities which can be described by property models, for example field devices, functional roles from the P&I-Diagram or even software units for process control.

This article presents an approach to model these distributed properties in an integrated way. Property information is modeled by statements, which contain important semantics to support diagnosis and asset management related functions. In addition, the model may be used to automatically gain property values of a plant element which does not offer this information a priori. Typically, we address non-intelligent plant elements like pumps or other devices, that do not offer their own measurements. The presented algorithm is designed in a modular way which minimizes the engineering effort and scales with the plant complexity.

The approach is already implemented within an industrial environment, realizing a fully automated monitoring application for pumps in a German BP refinery.

1. Introduction and Overview

The need to model and manage the information about the plant in an integrated way has become more and more important. Current technical equipment that is controlled via complex field bus protocols often offers a large amount of information about its functionality, properties and condition, while non-intelligent devices may not offer detailed information a priori. In general, most plant elements – both intelligent and non-intelligent – require many different sources of information, starting with planning the process, engineering the plant, continuing in the operational stage and ending in the shut-down. The question arises how to respond to these aspects within the plant's whole life cycle. In our opinion, an answer to this question requires generic models, views and methods instead of new specific technological solutions or standards.

Fig. 1 gives some examples of information sources that contribute essentially to the knowledge about a plant within its life cycle. Additionally, the prior step of standardization is denoted. This step is important, since it produces several standards, that are (both directly and indirectly) related to property models. Two aspects can be identified as main threads of all these information sources:

- the explicit description of structure and
- the explicit description of properties.

These two kinds of description are often mixed within one modeling method. For example, the process to be implemented in a projected plant leads to several functional requirements which often are represented in the form of Piping and Instrumentation Diagrams (P&IDs) and complementary planning documentation [7, 5]. They already contain requirements in the form of properties that have to be fulfilled by physical devices that will be installed later. But the P&I Diagrams primarily contain a description of the (desired) plant structure.

In this article, we focus on the aspect of properties. The goal is to use property information in an integrated way, considering a minimal semantic "core" which all sources of properties have in common. This semantic core has the feature to build up simply structured statements, which may be used to apply asset management functions. In fact, it is suitable for several functions which require plant properties from several sources and stages within the life cycle.

In consequence, to realize asset management related functionality like monitoring the condition of simple or complex plant elements, different applications and formats like engineering tools, (electronic) plant documentation or business administration systems are required, as postulated in [12]. For example, to determine if a temperature sensor is operated within the correct temperature limits, a comparison to the technical documentation of the installed sensor device is required. This simple piece of information, which might already be interesting from the point of view of asset management, already requires sensor properties distributed over the plant's life cycle. The task is ambiguous, since the requirement to be operated

978-1-4244-2728-4/09/$25.00 ©2009 IEEE

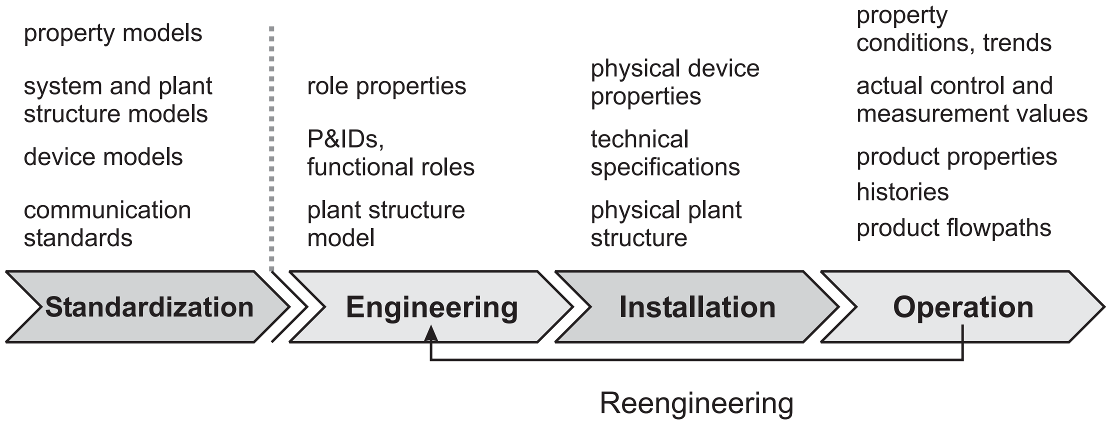

Figure 1. Several information sources within the plant's life cycle which have semantic dependencies and may serve as information clients to asset management functions

within the correct temperature limits may address limits from the technical device specification or from the functional role that has to fulfill the demanded process. So on the one hand, the condition of the hardware device is checked, and on the other hand, the process trajectory is supervised. Here, it is obvious that we have to consider different objects that are carrier of properties: functional roles from the planning stage and the real hardware devices which are already installed or are considered to be installed during procurement. Both sources of information should be provided during runtime, since a removal of a device and the installation of new equipment instead may lead to new device properties, while the plant function that has to be fulfilled remains the same.

To detect whether a physical device is able to fulfill (in general or at the moment) a functional role which is defined in the P&ID diagram could also be a relevant task. Again, properties coming from a CAE-tool and from the P&ID must be checked against device properties or potentially actual values. These examples already reveal the need to have a (minimal) and generic foundation for plant information, giving the ability to consider heterogeneously distributed properties. An approach for this foundation – containing just a minimal model core – will be presented in the following section.

2. Modeling and Managing Plant Properties

As illustrated in the last section, an important foundation for asset management related functionality is a model of the available properties. In the area of planning and procurement, several standards and recommendations already exist to describe components and devices. The NAMUR Recommendation 100 [13] defines lists of properties (LOP's) to describe devices in the engineering workflow. DIN EN 61360 [3] defines data element types and the corresponding classification schemes for electric components, also introducing a domain specific description language for that purpose [4]. Further standards are DIN 4002 [2] or ISO 13584 [8].

Our purpose is not to adopt these data models that partially contain complex and in-depth constructs, but to build up a description core which offers a kind of basic semantics to describe devices, functional roles or other assets within the plant's life cycle. This description model must not be contradictory to the existing models, but can be seen as a common semantics core which focuses on the possibility of forming statements about the properties of an object within the plant's life cycle. For example, we want to express that the functional role $P32$ of a pump requires a volume flow of at least 80 m^3/h, while an existing device offers a rated value of 100 m^3/h. These two statements contain different semantics: A requirement and an assertion, probably originated at differing moments. The functional requirement of 80 m^3/h can be generated at a point in time when no physical device is installed yet. Furthermore, both statements (or their respective values) may change within the life cycle (during reengineering, for example) and may be checked for different reasons. One purpose could be to verify that the physical device fulfills the functional requirement, additionally taking an adequate tolerance into account.

Considering these aspects, we propose an information model based on three main generic object classes, marked as A1, A2 and A3 in Fig. 2. We use the UML notation [14] for this model. Class A1 addresses all objects that may be characterized by (formalized) properties, the so called *property carrier objects*. As mentioned, these objects may have been created at different life cycle stages, coming from heterogeneous information sources. In general, we address arbitrary "assets" to be property carriers, but in the context of this article, we want to focus on the following two special classes of property carriers:

- functional roles which are typically defined in P&ID

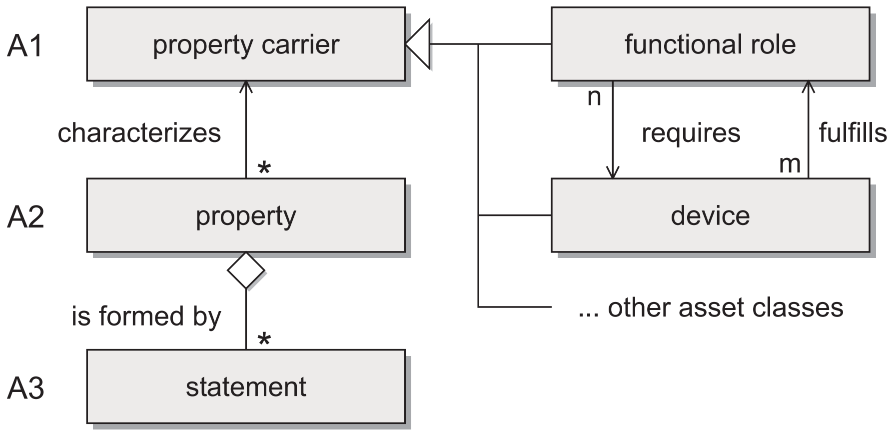

Figure 2. Generic property model that comprises the plant's life cycle

diagrams and the corresponding requirements specifications and

- devices, that means the real hardware objects that are installed in the plant to fulfill the requirements defined by functional roles.

Selecting these two types of described objects makes it possible to connect planning, engineering, procurement and operating phase semantically. The two associations ”requires” and ”fulfills” in Fig. 2 represent this aspect. So the model is able to keep track of functional requirements and the corresponding hardware to implement these requirements.

Up to now, we have introduced the objects that may be characterized by properties. As illustrated, there are several standards and models of properties for products, components or devices, representing different views (planning, engineering, process operation, spatial, electrical, ...) on the carrier objects. A second class (A2 in Fig. 2) is a generic representation of *properties*. We do not focus on the detailed data model of properties, for example description, possible values and units, since there are already models which represent these aspects. Beside the standards already mentioned, [6] gives a comprehensive overview of property models.

A third class A3 adds an important part of semantics to the information model. The basic idea is that all occurrences of a property value are formulated in the form of *statements*, which imply a certain intention background. For example, functional roles are built up in the engineering stage, representing requirements which have to be fulfilled by devices in the operational stage to realize the projected process. In the planning stage, assertions about later devices may also be made by specifying certain environmental conditions, for example. The technical documentation of devices, for example given by a Device List of Properties (DLOP) [13], contains assertions made by the manufacturer, concerning functional abilities, geometric properties or control information of the delivered hardware, amongst others.

Thus, a property *statement* is a third important class of the presented information model (A3 in Fig. 2). Statements types like actual values, assertions, requirements, definition of lower or upper bounds/alarm limits or assumptions about the value of properties are conceivable. Tab. 1 gives three examples of such statements.

The use of statements in place of direct property values has several advantages. In our opinion, the set of possible and reasonable statement types is finite and can be standardized. After defining possible statement types, logic operations can be defined, opening the door for automated checks on statements. For example, requirements and assertions may be processed, or actual values may be compared to technical specifications. By providing a semantic connection between functional roles and the devices currently implementing this function, statements from one property carrier may also be compared to statements from the other property carrier.

To use an analogy to natural language, the property carrying objects represent the subjects of the statements we want to form. The set of predicates is given by the (pre)defined properties, which is task of the standardization process. The predefined statement types define the set of valid sentences which may be formulated. The left column of Tab. 1 shows the formally modeled statements represented by a sentence in natural language. So the property model of Fig. 2 also offers an intuitive understanding. Nevertheless, semantics and dependendies between different statements of varying information sources may also be recognized by automatons, since we only allow a finite set of statement types to be formed.

The classes presented to manage the plant property information represent a continuation and an application of a property model we presented in 2007 [10]. Here, we only focused on the model details which are required to understand the remaining sections. We now use this property model as a foundation for an algorithmic approach to determine new property information during the plant operation automatically.

formulation in natural language	property carrier	property	statement type	value, unit	stage of origin
"The functional role $P32$ requires a volume flow of at least 80 m^3/h."	functional role	volume flow	requirement, lower bound	80 m^3/h	engineering
"The pump device with the serial 38705 offers a rated volume flow of 100 m^3/h."	pump device	volume flow	assertion, rated value	100 m^3/h	procurement
"The current volume flow of measurement $F33$ is 97 m^3/h."	flow sensor device	volume flow	actual value	97 m^3/h	operation

Table 1. Examples of property statements

3. Solving Static and Dynamic Assignment Problems

The examples of statements given in Tab. 1 refer to the property "volume flow", which implies a clear understanding and semantics to the reader. Especially conclusions can be drawn from those statements, using the fact that they are connected semantically. For example, potential semantic dependencies between those statements and resulting logic conclusions could be the following ones:

- The pump device 38705 is able to implement the functional role $P32$ *in general*, since 100 $m^3/h \geq$ 80 m^3/h holds.
- The flow measurement $F33$ also represents the *current* volume flow of the pump device 38705.
- Thus, *at the moment*, both device 38705 and functional role $P32$ are implemented and operated correctly (within tolerable limits).

However, these statements are representative of two general assignment problems, especially if these conclusions have to be drawn automatically:

1. **Static assignment:** Functional roles and devices are usually not modeled and constructed by the same tools and formats. Instead of "volume flow", the properties in Tab. 1 could have different names and result from different models. This leads to a semantic gap, since the fact that those three statements have a strong dependence, is not recognizable automatically anymore. Since this gap is caused by the models which have to be established before the plant's life cycle, we call this assignment problem *static*.
2. **Dynamic assignment:** To gain actual values of properties in the operational stage, further information is necessary. For example, to map the property "volume flow" of the measurement $F33$ to the property "volume flow" of device 38705, further structural plant information is required. The amount and availability of such derived property information depends strongly on the current plant situation. Thus we call this assignment problem *dynamic*.

We believe that the first, static assignment problem can only be solved by standardization. CAE tools, P&IDs, device models and technical specifications must refer to determined properties using a consistent nomenclature. This will close the semantic gap between planing, engineering and procurement.

The second, dynamic assignment problem, i.e. gaining property values during plant operation, may have different solutions, which depend on the plant structure and complexity. The probably simplest solution to gain a required property value could be to assign it directly, for example by installing a flow sensor close to the pump or by identifying an appropriate measurement that (always!) mirrors the pump's volume flow. This can be accomplished by automatically executed (knowledge based) rules at engineering time [16]. Fig. 3 shows an example.

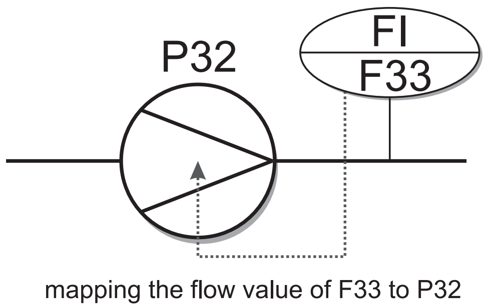

Figure 3. Plant example of a direct and static 1:1 assignment of properties

In general, plant situation and structures may occur that do not allow such a direct assignment. For example, flexible pipe networks with multiple valves make such direct assignments impossible, since a device may dynamically be connected to the other plant elements by different product flow paths. Thus, we will now present a generic solution to this assignment problem.

3.1 Algorithm Overview

A usual case is that a plant element under consideration (or even an aggregation of several plant elements) does not offer desired property values of its own and the direct surroundings do not allow such a direct assignment as sketched in Fig. 3. A pump, for example, might not offer the pressure difference to determine the delivery head and thus the pump performance, which in turn is important to rate the device condition. Instead of installing new measurements for every property that might potentially be interesting, we present an approach to gain these properties automatically, if possible. To do so, first the overall available plant properties must be modeled consistently (layer A in Fig. 4). We assume that the property model presented in the last section 2 serves as information base, containing property statements for further use. Starting with this assumption, we present an algorithm consisting of three important steps (see Fig. 4):

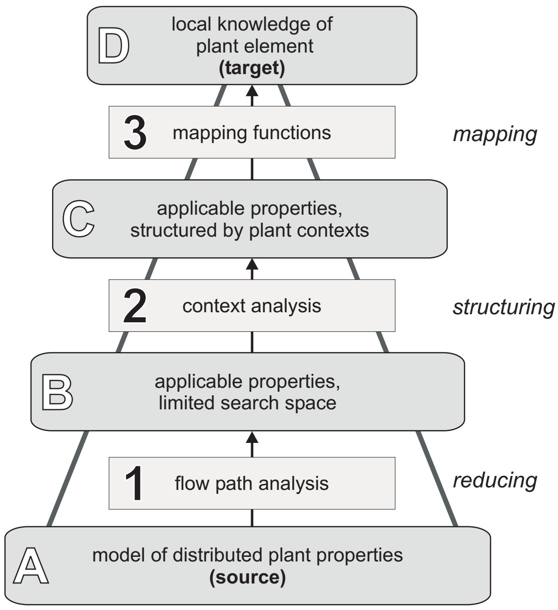

Figure 4. Algorithm outline: the profile of a pyramid indicates the process of information compression from global to local information

- **Step 1: Identifying Applicable Properties – The Flow Path Analysis:** Only a subset of all modeled plant properties might help to gain information about an element or device. Therefore, the search-space must be reduced in a way that only (potentially) applicable properties remain (layer B in Fig. 4). This is performed by the *flow path analysis*, which determines the sequence of plant elements which are connected to the considered element by the product flow. Within this sequence, information in the form of actual values or engineered properties can be used further. This step is described in Sec. 3.2.
- **Step 2: Identifying Typical Plant Situations – The Context Analysis:** The *context analysis* analyzes the given flow path and matches it to predefined plant situations, the so called contexts. Contexts may change with the current flow path, which is often the case in flexible pipe systems. This step is illustrated in Sec. 3.3.
- **Step 3: Gaining new Property Information – Mapping Functions:** The two previous steps have helped to identify and structure the plant properties (layer C in 4), which can now be used to infer local properties of a plant element under consideration (layer D in Fig. 4). This last step is described in Sec. 3.4.

The following sections present details to the three algorithm steps.

3.2 Identifying Applicable Properties: The Flow Path Analysis

To keep at the statement examples from Tab. 1, the current volume flow of pump $P32$ may be required, but not available, since no local measurement does exist and the measurement $F33$ might not be usable in general. In terms of diagnosis or asset management, a comparison of the current flow to the rated value in the technical specifications could be helpful, detecting potentially problems.

However, to use the distributed information within the plant, we need to determine the search space of properties that might help to gain the required volume flow *at runtime and depending on the current plant situation*. Here, the use of both plant structure and (already) available property statements is required. This can be shown by abstracting the plant topology to a graph based structure. Fig. 5 gives an example of such a graph. Assuming that some of the nodes are valves, the plant control enables different flow paths. The number of possible flow paths scales with the possible paths in the graph, resulting in a combinatoric explosion.

So if some properties of the pump $P32$ are of interest, we first have to determine which other plant properties may potentially help to gain information about $P32$ at the current moment. The search space of all available properties can be reduced by determining the current product flow path. An example of such a flow path is marked gray in Fig. 5. So the set of plant elements and corresponding properties is reduced to the marked elements. If a flow measurement is found on the flow path, it helps to determine the pump's volume flow for the duration of the flow path. If the flow path is decomposed, for example by closing a valve, this assignment is not valid anymore. So the

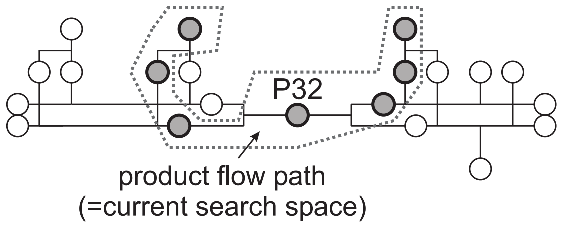

Figure 5. Graph based plant model with one of multiple possible flow paths

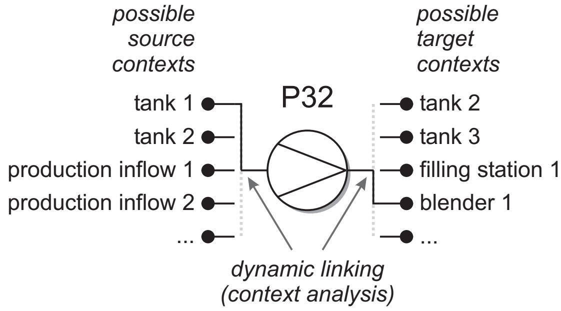

Figure 6. example of different context situations for a pump

goal is to have an assignment of properties to other properties, but in a dynamic way, since the flow path may change in the time span of regarding a plant element.

There are several methods to detect the current product flow path. An analytic solution requires a model of the plant topology at runtime. For example, a global search or a local, component based approach can be used. In the latter, local communicating function blocks are connected identically to the plant structure, realizing a distributed search of the current flow path [15]. The required valve positions can be obtained in the same way as other properties, by using statements about actual values from the statement component in the property model (Fig. 2). This demonstrates the "cooperation" of structural description and property information. As a plant model, CAEX [1] or RIVA [9] may be used, amongst others. Another, non analytic solution is the use of control information from the process control system. For example, the activation of a group control and its parameters can be used to detect the flow path. In this case, a predetermined sequence of plant elements is assumed to be current flow path.

With the product flow path analysis, step 1 in Fig. 4 is realized, leading to a reduced search space of properties (layer B). Instead of working on the flow path directly to gain local information about a plant element under consideration, we propose an intermediate step, which often can ease the plant situations. This step is presented in the next section.

3.3 Identifying Typical Plant Situations: The Context Analysis

If, for example, a measurement is identified on the flow path by iterating over its elements, the direct assignment (Fig. 3) to required properties may temporarily be possible. The situation becomes more complex if there is no measurement and thus no direct assignment possible. Nevertheless, a calculation of the required properties may be still possible. For example, if the intake pressure of a pump is required and no direct measurement exists, a tank model using the Bernoulli equation could be used, assuming that the flow path starts with a tank. But this tank model is more complex than a direct measurement, since it requires several combined properties like a level measurement, tank geometry and the geodetic height of tank and pump. In general, plant situations may occur that require a combination of several properties and need specialized computation models to determine local properties like the intake pressure. We call these situations contexts, which can be linked dynamically to a plant element under consideration by the so called context analysis. Fig. 6 gives examples of different contexts as source or target of the pump. Contexts can be found by iterating over the elements of the current flow path and check some necessary conditions which have to be fulfilled.

Contexts can be very plant-specific, depending on specific physical plant properties. But every type of context, which usually may occur more than once (see Fig. 6), has to be specified only once, which leads to a reduction of engineering effort.

Furthermore, the intermediate step of the context analysis (step 2 in Fig. 4) makes it possible to simplify the real plant situation: Although the flow path might be a complex sequence of different plant elements going through several valves, the context analysis is able to map the current plant situation to a very simple structure (layer C in Fig. 4), just consisting of source, pump and target (in general context, element, context). The context analysis also scales with complex and flexible plant structures, since the combinatoric explosion of possible flow paths and the many possible assignments are bypassed by analyzing the flow paths for contexts.

3.4 Gaining new Property Information: Mapping Functions

In the last step, local property values of the plant element under consideration can now be determined, if the amount of information distributed over the current flow path suffices. If no direct assignment is possible (the flow measurement $F33$ could be contained in the current flow path), the context analysis delivers the name and type of the currently connected contexts, for example tank 2 as source context for the pump $P32$ (Fig. 6). These contexts may now be used to map the properties which are distributed over the flow path to local properties.

Fig. 8 illustrates some examples of operations on a

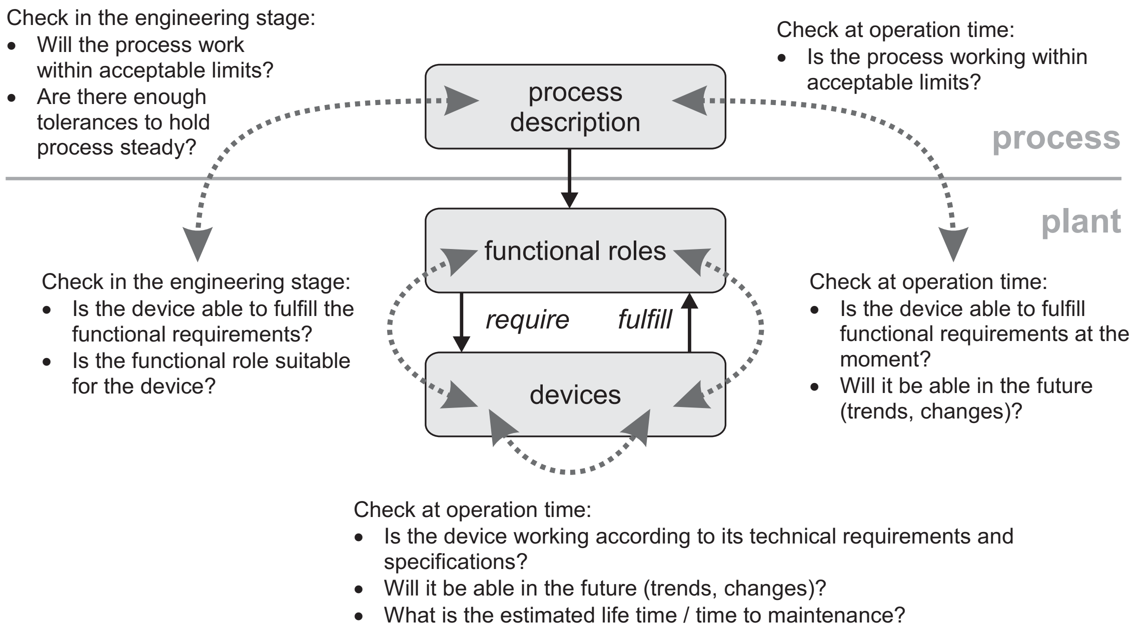

Figure 7. By describing different types of property carriers with the same statement model, several planning, diagnosis and asset management functions can be applied automatically

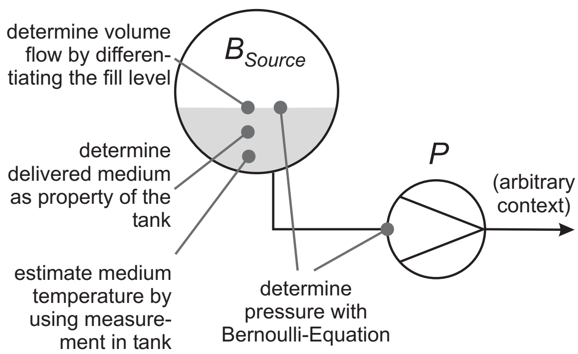

Figure 8. example of gaining local pump properties from a tank context at source side [11]

tank context at the source side of a pump. The pumps volume flow – if not available directly – can be determined by differentiating the fill level of the tank. Usually, the product contained in the tank is a property of the tank, thus if the tank is part of the current product flow path, the same medium is delivered by the pump. So the tank context provides the possibility to infer several local pump property values. Once a context type and the corresponding mapping operations are specified, this process of gaining new properties runs fully automatically. For more details on the algorithmic aspects, we refer to [11].

Although the mapping functions have a generic character (a level differentiation always leads to a volume flow), very plant specific properties may also influence these functions. For example, the pipe lengths and diameters connecting the different contexts to the pumps may influence the pressure calculation. Such plant specific dependencies can be taken into consideration when modeling the different classes of contexts which may occur in the plant.

4. Building up Generic Diagnosis and Asset Management Functions

Up to now, we have presented an approach to model and gain property statements over the plant's life cycle. Fig. 2 illustrated the small property model core, Fig. 4 gives an overview of a generic algorithm to gain new property values at the plant's operation time. Several types of statements about properties, for example requirements from the engineering stage, assertions from device specifications and actual values from the operational stage are available.

These statements provide important information to applications from the area of diagnosis or asset management. Property statements about a plant element under consideration can be collected within its life cycle. If some properties do not exist, they may be gained automatically by the presented flow path approach. If, for example, the pressure difference of a pump and information about the delivered medium can be gained, the pump performance is computable and the pumps condition may be estimated. Too low intake pressures in combination with a "wrong" medium may lead to cavitation and thus to damages in the pump. In general, statements about actual values of device properties that help to predicate the device condition may purposefully looked for.

The model provides the information about functional roles and the corresponding devices fulfilling these roles. So beside checking that a device is operated in order to its technical specifications, it can also be checked that the device is currently able to fulfill its required function. A heat exchanger, for example, may have a reduced heat transfer efficiency, which can lead to unwanted changes in the process, even if the device is still working according to its technical specificationsand limits.

All in all, the presented model may help to check several conditions automatically in several stages of the plant's life cycle. Fig. 7 sketches some more examples of operations that help during engineering, procurement and operation, in particular for diagnosis and asset management related problems.

5. Summary

We presented an approach to represent and gain plant information in the form of property statements within the plant's life cycle. This permits an integrated treatment of property information resulting from planning, engineering, procurement and operation. All in all, a large background of information for possible diagnosis or asset management functions is provided.

For the case of missing property values in the operational stage, a fully automated algorithm was presented to gain these values. This algorithm requires information about the plant structure, using adjacent properties of a plant element under consideration to infer local information. To reduce the (potentially) large search space, it analyzes the current product flow path. By additionally identifying standard plant situations in the form of contexts, uniform mapping operations from properties on the flow path to local properties can be applied.

The algorithm presented in Fig. 4 and the information model in Fig. 2 are already implemented and running in a German BP refinery. The application collects properties to monitor the condition of pumps and runs fully automated.

References

[1] Deutsches Institut für Normung. *DIN V 44366: Specification for Representation of process control engineering requests in P&I Diagrams and for data exchange between P&ID tools and PCE-CAE tools*, Dec. 2004.

[2] Deutsches Institut für Normung. *DIN 4002:Properties and their scopes for product data exchange – Part 2: Terms and conceptual information model*, 2007.

[3] Deutsches Institut für Normung. *DIN EN 61360 (draft): Standard data element types with associated classification scheme for electric components – Part 1: Definitions*, 2008.

[4] Deutsches Institut für Normung. *DIN EN 61360 (draft): Standard data element types with associated classification scheme for electric components – Part 2: EXPRESS dictionary schema*, 2008.

[5] Deutsches Institut für Normung / International Organization for Standardization. *DIN EN ISO 10628: Flow diagrams for process plants – General rules*, Mar. 2001.

[6] M. Heeg. *A Contribution to the Modelling of Properties in the Area of Process Control Engineering*. PhD thesis, Chair of Process Control Engineering, RWTH Aachen University, 2005.

[7] 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*, Aug. 2008.

[8] International Organisation for Standardization. *ISO 13584: Industrial automation systems and integration - parts library, Part 1: Overview and fundamental principles*, 2001.

[9] R. Jorewitz, U. Epple, A. Münnemann, R. Böckler, W. Wille, and R. Schmitz. Automation of performance monitoring in an industrial environment. In *PCIC Europe 2005, 2nd European Conference on Electrical and Instrumentation Applications in the Petroleum and Chemical Industry*, pages 116–120, Basel, Switzerland, 2005.

[10] M. Mertens and U. Epple. Online Plant Asset Management Based on a Characteristic-Processing Structure Model. In *VDI Berichte 1980, GMA-Kongress 2007: Automation im gesamten Lebenszyklus*, pages 467–479. VDI/VDE Gesellschaft Mess- und Automatisierungstechnik, June 2007.

[11] M. Mertens, Q. Quirós, and U. Epple. Using Flowpath Information for Diagnosis Applications. In *Automation 2008: Lösungen für die Zukunft*, pages 23–26. VDI/VDE Gesellschaft Mess- und Automatisierungstechnik, June 2008.

[12] NAMUR: User Association for Automation in Process Industries. *NE 91: Requirements for Online Plant Asset Management Systems*, Nov. 2001.

[13] NAMUR: User Association for Automation in Process Industries. *NE 100, Version 3.1: Use of Lists of Properties in Process Control Engineering Workflows*, Nov. 2007.

[14] Object Management Group. Introduction To OMG's Unified Modeling Language (UML), http://www.omg.org, July 2005.

[15] G. Quirós, M. Mertens, and U. Epple. Function blocks for decentralised analysis of product flow paths. In *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Apr. 2008.

[16] T. Schmidberger, A. Fay, and R. Drath. Automated engineering of process automation systems. *atp – Automatisierungstechnische Praxis*, 47(2):45–51, Feb. 2005.