

# Reference Model Concept for Structuring and Representing Performance Indicators in Manufacturing

Stefan Hesse<sup>1</sup>, Bernhard Wolf<sup>1</sup>, Martin Rosjat<sup>1</sup>, Dražen Nadoveza<sup>2</sup>,  
and George Pintzos<sup>3</sup>

<sup>1</sup> SAP AG, SAP Research Dresden, Chemnitzer Str. 48, 01187 Dresden, Germany  
s.hesse@sap.com, b.wolf@sap.com, martin.rosjat@sap.com

<sup>2</sup> École polytechnique fédérale de Lausanne EPFL, 1015 Lausanne, Switzerland  
drazen.nadoveza@epfl.ch

<sup>3</sup> University of Patras, Rio, Patras 26500, Greece  
pintzos@lms.mech.upatras.gr

**Abstract.** Performance indicators (PIs) are used to monitor and assess production systems. There are thousands of PIs described in standards or in commercial PI collections; however, the PIs implemented in the factories may differ enormously due to use-case-specific requirements. In this work a reference model is proposed to support the process from a generic description to a use case specific PI implementation. There are two exemplary implementations described utilizing data stream processing and database technologies.

**Keywords:** performance indicators, KPI modeling, manufacturing.

## 1 Introduction

In modern manufacturing, performance indicators (PIs) are used to express certain conditions in the production, e.g., efficiency of machines and processes, quality of products, probability of failures, or resource consumption of certain machines or production facilities. There are PIs that are common to particular industry domains or to the whole manufacturing industry, such as OEE (Overall Equipment Effectiveness), but also very specialized PIs that are used by only one manufacturer.

An analysis of available PIs – including standards, public PI collections, and industrial use cases – showed that there are performance indicators, which share the same name and descriptive information but have a different semantic, even in the same factory. Furthermore, the implementation of PIs often distinguishes from the common description since PIs are adapted to use cases and individual requirements. This situation makes it difficult to compare performance indicators in industry domains or even between departments inside one enterprise.

To solve this problem a reference model is proposed, which consists of five layers to describe performance indicators and their application in a structured manner, beginning with a very generic PI description and adding more use case specific details

in the following layers. The model also supports references to hierarchical plant or product descriptions, as well as relations to other performance indicators.

## 2 Motivation

Monitoring and analyzing production processes is usually done from different perspectives. PIs can be related to different physical or logical entities, e.g., energy consumption can be relevant for a machine or a whole plant, but also for single products or a product series. While these use cases are different on the first view, both formulate same requirements to performance indicators:

- *Extensibility*: Attributes are used to describe PIs. There are common attributes that are shared by all PIs, e.g., name, but also attributes that are specific to domains, companies etc. To describe PIs appropriately, standard attributes as well as a possibility to add further attributes are needed.
- *Flexibility*: Common PI descriptions sometimes include alternative attribute values, e.g., for units. However in a certain scenario only a subset of the units is relevant. PIs should support refinements of attributes.
- *Dependencies*: PIs can depend on other PIs, i.e., one PI is calculated from other PIs. However, PI dependencies do not necessarily have to be modeled bi-directional. For simple models it is sufficient to list only input PIs.
- *Relations*: PIs can be related to different physical or logical entities, even at the same time. For instance, a performance indicator may be calculated for machines or product instances as well as for work centers or product series.
- *Aggregation*: PIs are sometimes aggregations of other PIs, e.g., when particular plant levels are considered.
- *Comparability*: PIs are used to compare for instance machines, resources, departments, or products inside and across enterprises.

## 3 Related Work

Currently there is an increasing usage and importance of PIs across several industries and there are attempts to collect and even standardize PIs and their descriptions, e.g., ISO 22400 [1] and VDMA 66412 [2]. The ISO 22400 contains the definitions and descriptions of 35 manufacturing PIs, including two PIs related to energy, while the VDMA 66412 is an industrial standard that includes 21 PIs related to manufacturing provided by the German association for engineers (VDMA). Furthermore, a number of (commercial) collections of PIs exist in the internet such as KPI Library [3], SAP Community Network [4] and SmartKPIs.com [5].

Although these PI standards and collections use a number of characteristics to describe a PI they are lacking in three areas: specific descriptions of the relations between the different indicators, descriptions of the use of indicators in specific systems or use cases, and a detailed description of how the indicators are calculated. In current literature, specific architectures have been proposed for all manufacturing systems

together with specific hierarchy levels, which can be used to describe in which type of system or on which hierarchy level an indicator is applied. In [6] [7] [8] [9] similar hierarchy levels have been proposed that can lead to a more specific description of an indicators application.

Furthermore, the indicators can be described through their relevance to a process or a product part and therefore additional product compositions can be used such as final product, subassembly, raw materials etc.

### 4 Reference Model

Based on the requirements described in Section 2 we built a five-layer reference model for the definition of performance indicators, depicted in Fig. 1. Beginning at the top layer, every layer can be seen as a refinement of PIs towards a use-case-specific implementation. This means that every layer inherits the attribute values from the layer above and defines more specific information by additional attributes and by substantiating the inherited attribute values. The details of the layers are described below.

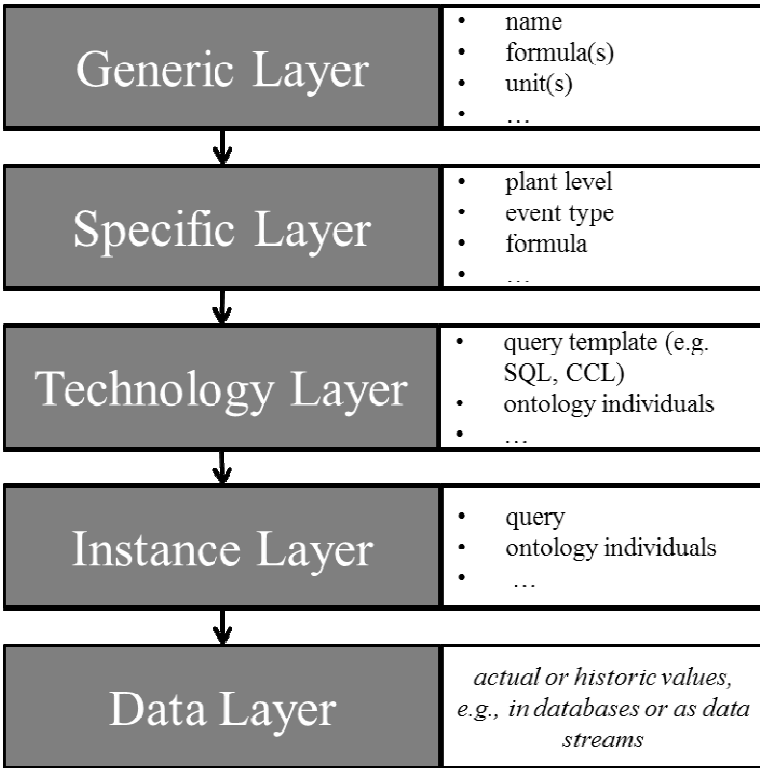


Fig. 1. Performance indicator reference model

## 4.1 Generic Layer

The *Generic Layer* describes PIs in the most common way, i.e., *generic PIs*. It is comparable to known standards; however, the number of attributes used to describe generic PIs might be less compared to PIs in the standards. The attributes to describe generic PIs are name, acronym, formula(s), unit(s), description, as well as relations to other performance indicators. To build a dependency model for PIs, for each PI a list of PI inputs is sufficient. For the exemplary performance indicator Energy Consumption, the formula would contain  $\text{EnergyConsumption} = \text{sum}(\text{Power})$ , where Power is another PI summarized over a specified time. Therefore, Power is also listed in the relations (see Table 1).

The formula and unit attributes can contain none, one or more entries. There are cases where semantically equal PIs are specified with various calculation formulas, e.g., Mean Time Between Failures (MTBF). Especially for units usually a set of potential units is defined, e.g., sec, min, hours, etc. for time-based PIs such as Cycle Time.

**Table 1.** Generic description example

<b>Name</b>	Energy Consumption	<b>Acronym</b>	EnC
<b>Formula</b>	$\text{EnC} = \text{sum}(\text{Power})$	<b>Unit</b>	Wh, kWh, J
<b>Input PIs</b>	Power		
<b>Description</b>	The Energy Consumption ...		

## 4.2 Specific Layer

The *Specific Layer* describes PIs specific to a certain use case. A *specific PI* is based on one generic PI and inherits all attributes from the generic description but is also able to overwrite these entries. The set of generic attributes is extended by other (optional) attributes to describe particular properties of the use case that are related to the specific PI, including references to related machines, the plant level where it is applied, the industrial domain, or production information such as components of a produced part. Thereby, a relation to production hierarchies or product compositions can be created. In addition to that, properties about the calculation of the PI can be determined, usually, what triggers the calculation of a PI, how many data is considered, or how often the calculation of the PI is executed, e.g., every minute or every month.

Continuing with the former example; several specific PIs can be created based on the generic Energy Consumption. Examples are “Energy Consumption (machine)” or “Energy Consumption (work center)” for different entities of the production hierarchy. Product-related PIs are also possible, e.g., “Energy Consumption (per part)” or “Energy Consumption (per series)”, however, the calculation may differ from first-mentioned examples.

Overwriting is not only applicable for the names but could be done for all inherited attributes; however, the semantic of the PI must not be altered. Further examples are the extension of a description with additional information or the restriction to use-case-relevant relations or units, basically subsets of the generic definitions. The most

varying changes are probably assigned to the formula field. In simple cases the formula is not modified or one of the generic formulas is selected. Apart from that, we observed cases in real-world scenarios, where formulas were only defined on the specific layer; i.e., the formula attribute of the generic PI is empty. In the concrete example, formulas were specific for product series and depending on certain features of a product a quality PI is calculated. The results are semantically equal and can be used to compare certain product series.

Another way to calculate PIs is aggregation. Where the Energy Consumption for a machine might be calculated from the Power that is measured at the machine, the Energy Consumption of a work center is the sum of the consumed energy of all machines in the work center over a specified time. In that case, the formula is not relevant, but a definition of an aggregation. To decide which machines belong to the work center, relations to the production hierarchy can be used.

Table 2 extends the generic Energy Consumption example used above. The specific PI refers to a machine in the production facility. The calculation is triggered every hour and done considering power data over the last day, i.e., the last 24 hours. There-with, a hopping window is realized. The set of potential units is reduced to kWh only.

**Table 2.** Specific description example

<b>Name</b>	Energy Consumption (ma- chine)	<b>Acronym</b>	EnC
<b>Formula</b>	EnC=sum(Power)	<b>Unit</b>	kWh
<b>Input PIs</b>	Power (machine)		
<b>Plant level</b>	machine	<b>Event type</b>	hourly
<b>Window Size</b>	1 day	<b>Window type</b>	hopping
...	...		
<b>Description</b>	The Energy Consumption ...		

The specific descriptions can be used to create calculation templates for PIs or models of the overall PI structure in a factory.

### 4.3 Technology Layer

The Technology Layer refers to abstract and technology-specific artifacts or expressions, which describe the specific PIs. Examples are individuals of an ontology and query templates for databases (e.g. SQL, MDX) or for data stream systems (e.g. CCL, Linq) respectively. Introducing technology-specific templates saves time and cost during the configuration and maintenance of the PI processing system. So, query instances can be created quickly without too much repetitive efforts. Query templates contain placeholders for inputs and parameters to be replaced during the runtime by concrete values. An exemplary query template for the CEP (complex event processing) query language CCL [10] for the energy consumption PI above may look like follows. Placeholders for parameters, input streams, and output streams are represented by #par:...#, #in:...#, and #out:...#, respectively.

```
CREATE OUTPUT WINDOW #out:EnCStream# PRIMARY KEY DEDUCED
KEEP #par:windowSize# #par:windowSizeUnit# AS SELECT
sum(inStream.Power) EnC, inStream.ID ID
FROM #in:PowerStream# AS inStream GROUP BY inStream.ID
OUTPUT EVERY #par:stepSize# #par:stepSizeUnit# ;
```

#### 4.4 Instance Layer

The Instance Layer relates to entities of the PI processing system. Based on the PI templates concrete instances of PIs are created. Such queries are instantiated replacing the placeholders in a query template with proper data sources and parameters.

So, one PI query template can be used to create several PI instances. In the Energy Consumption example, the template for “Energy Consumption (machine)” can be instantiated for each machine that has a power measurement, assigning the appropriate input (the power signal of that machine) and setting the parameter for the duration, e.g., one hour. The same can be done for work centers utilizing an “Energy Consumption (work center)” template. Based on the template example and the parameters of the specific description above a query for “machine 1” could be:

```
CREATE OUTPUT WINDOW EnergyConsumption_Machine1 PRIMARY
KEY DEDUCED KEEP 1 DAY AS SELECT sum(inStream.Power) EnC,
inStream.ID ID FROM Power_Machine1 AS inStream
GROUP BY inStream.ID OUTPUT EVERY 1 HOUR ;
```

#### 4.5 Data Layer

The *Data Layer* symbolizes all data that is created by or associated to PI instances. There is a wide range of potential data consumers. In the following, just a few cases from the research projects are mentioned. The PI data can be visualized to monitor and analyze the processes by the plant personnel. In other cases, PIs are calculated to identify critical situations and trigger further activities. PI data can be fed back into the production process to control or improve it or used as input for other systems, e.g., optimization tools, for scheduling, or reporting. Moreover, data can be stored for later analyses or as a reference.

## 5 Implementation

The reference model was implemented in two industrial use cases utilizing different technologies.

The first implementation is a hybrid system consisting of a CEP engine and an in-memory database where PIs are calculated in real-time using continuous data stream queries. Further details of the architecture are described in [11]. For monitoring of a production facility, performance indicators were selected and described as specific PIs. Based on that, data stream query templates for two different CEP engines were

created using the CEP-specific query languages. Depending on the CEP that executes the calculation, the proper query template is selected to create PIs instances in the system. In addition to that, the generic and specific PI descriptions were used to create an ontology. The ontology can be used for root cause analysis, i.e., supporting the user to find out which PIs influence other ones or are influenced by others. Sophisticated scenarios for the ontology usage are planned, e.g., (semi-) automatic query generation or production optimization based on PI data.

The second implementation used the reference model for a PI monitoring application in the automotive industry [12] on top of a database and a data warehouse. Like in other industries various PIs exist with the same or similar semantics. The proposed model was used to find similarities. Thus, a better overview and the ability to compare PIs in a consistent way could be provided to end users. The application provides user-configured views on relevant PIs instances. Besides the monitoring of PI values the application offers a possibility to express dependencies between single PIs. Using the new approach simplifies the configuration of views and dependencies. Traditionally, users define relevant PIs (instances) out of a set of available and authorized PIs. Dependencies between the selected PI instances are individually modeled afterwards. In most of the cases, same or similar dependencies are modeled multiple times and by different users. The introduction of the generic and specific models allowed for a description of PI dependencies already at a higher level of abstraction. Once available, this information can be used during the creation and analysis of single PI instances. Out of the general descriptions the dependencies on instance level can be retrieved automatically. The dependency model is used for root cause analyses, as well.

## 6 Conclusion and Future Work

The application of performance indicators in manufacturing shows the necessity of a consistent model to define and re-use PI descriptions with different granularities. In this paper a reference model for performance indicators is presented. It can be used as a generic approach to describe PIs for specific use cases. Starting with an abstract description, use-case-specific details are added resulting in a PI description ready for implementation. Future work will include a semantic PI model implementing the reference model to allow for a detailed description of the inheritance attributes as well as the extern relationships. As a further aspect, time dependent properties will be included into the reference model, such as absolute and relative time relations (e.g. yesterday, last quarter).

**Acknowledgments.** The projects PLANTCockpit (260018) and KAP (260111) are co-funded by the European Union under the Information and Communication Technologies (ICT) theme of the 7<sup>th</sup> Framework Programme for R&D (FP7).

## References

1. International Organization for Standardization (ISO): ISO/CD 22400-2: Automation systems and integration - Key performance indicators (KPIs) for manufacturing operations management - Part 2: Definitions and descriptions (May 2012)
2. Verband Deutscher Maschinen- und Anlagenbau e.V (VDMA): VDMA-Einheitsblatt 66412-1 – Manufacturing Execution Systems (MES) – Kennzahlen (October 2009)
3. KPI Library B.V.: KPI Library, <http://kpilibrary.com/>
4. SAP AG: SAP Developers Network - KPIs, <http://wiki.sdn.sap.com/wiki/display/KPI>
5. The KPI Institute Pty. Ltd.: SmartKPIs.com, <http://www.smartkpis.com>
6. Chryssolouris, G.: Manufacturing Systems: Theory and Practice, 2nd edn. Springer (2006)
7. Jones, A., McLean, C.: A proposed hierarchical control model for automated manufacturing systems. *Journal of Manufacturing Systems* 5(1), 15–25 (1986)
8. Westkämper, E., Hummel, V.: The Stuttgart Enterprise Model - Integrated Engineering of Strategic & Operational Functions. *CIRP Seminars on Manufacturing Systems* 35(1), 89–93 (2006)
9. Catron, B.A.: A Framework for Semiconductor Manufacturing. In: 3rd DARPASCR CIM-CI Workshop, Stanford, CA (1988)
10. Sybase Inc.: CCL Reference Guide - Sybase CEP Option R4 (2011), [http://infocenter.sybase.com/help/topic/com.sybase.infocenter.dc01031.0400/doc/pdf/CEP\\_CCLReference.pdf](http://infocenter.sybase.com/help/topic/com.sybase.infocenter.dc01031.0400/doc/pdf/CEP_CCLReference.pdf)
11. Schramm, A., Wolf, B., Hartung, R., Preußner, A.: Real-time production monitoring in large heterogeneous environments. In: Emmanouilidis, C., Taisch, M., Kiritsis, D. (eds.) APMS 2012, Part II. IFIP AICT, vol. 398, pp. 72–79. Springer, Heidelberg (2013)
12. Hesse, S., Vasyutynskyy, V., Rosjat, M., Hengstler, C.: Modeling and Presentation of Interdependencies between Key Performance Indicators for Visual Analysis Support. In: Emmanouilidis, C., Taisch, M., Kiritsis, D. (eds.) APMS 2012, Part II. IFIP AICT, vol. 398, pp. 281–288. Springer, Heidelberg (2013)