Model-based Approach to Automated Calculation of Key Performance Indicators for Industrial Turbines

Gulnar Mehdi[1], Davood Naderi[2], Giuseppe Ceschini[3], Alexey Fishkin[4], Sebastian Brandt[5] Stuart Watson[6], and Mikhail Roshchin[7]

[1,4,5,7]Siemens AG, 81739, Germany

fname.lname@siemens.com
gulnar.mehdi.ext@siemens.com

[2]Siemens Industrial Turbomachinery AB, 61241, Sweden

fname.lname@siemens.com

[3]Siemens AG, 90001, Germany

fname.lname@siemens.com

[6]Siemens AG, LN63AD, United Kingdom

fname.lname@siemens.com

ABSTRACT

In recent years, the service business of the global turbo-machinery industry has undergone important changes. Many of these changes have been motivated by an increased demand for dedicated and systematic approaches to process safety, reliability, asset integrity and the overall health of the system. This has strengthened the role of key performance indicators (KPIs) as a means of providing guidance for the system's health state and improve risk management. In order to provide trustable and accurate calculations of these performance indicators in an automated fashion, we argue for a model-based solution that deals with the complexity of diverse configurations and interdependences between system components. This paper presents a solution for calculating KPIs by a semi-automated process based on post-data processing from the site and specific system models. The models consist of a combination of system descriptions in terms of ontologies and complex event processing models. By virtue of our models, state indicator rules for KPI calculations can be formulated at different levels, identifying performance gaps and indicating precisely where action should be taken by the service engineers. With the adopted solution, we discuss the practical implementation and present results of our success story at Siemens AG for the Industrial Gas Turbines. Finally, we provide an evaluation and future developments.

Gulnar Mehdi et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. INTRODUCTION

In recent years, the turbo-machinery industry has provided a wide range of products and comprehensive services to their customers. The industry has evolved in terms of increasing product standardization and continues to adopted strategies to enhance their value-added services. As part of that industry, Siemens AG aims to expand their service business to mobilize the additional potential of sustainable growth. Keeping up with the technological advances, Siemens Corporate Technology (CT) and the turbo-machinery portfolio is laying the foundations for next-generation smart and efficient solutions in the energy sector. Their focus is to enable improved plant operations, lower maintenance costs, increased plant lifetime, safety, reliability, asset integrity, and mitigation of risks. In general, these objectives can be achieved by the adoption of appropriate monitoring, diagnosis and maintenance tools that support effective decision-making and customer service.

KPI-based approaches are among the most practical and popular ways to describe the state and efficiency of the plant. Ceschini (2002) states that KPIs also provide guidance for monitoring, availability, maintenance and review of the system's health and help to derive sound statistics directly from the operational data. Recently automated calculations for machine performance indices have been reported by Ding et al (2013) and Odgaard et al (2013) which significantly focus on developing engineering models of the machine components and drive results using statistical methods. Whereas Márquez et al (2012) describes various state-of-the-art techniques including qualitative fault tree analysis for performance monitoring of established thermodynamic models. These reported techniques require greater engineering expertise to build a system model, is less transparent and lacks usability. Nevertheless, the application of KPIs for industrial turbines still has its challenges. Some of the prominent features that introduce substantial complexity to the computation of KPIs are as follows:

1. Forsthoffer (2011) shows that industrial turbines may have many different sets of configurations and topologies depending on design and applications. For example, twin-shaft turbines versus single-shaft and applications for mechanical drive versus power generation. As an example Fig. 1 shows a sample list of various configurations occurring in the industry.

Turbine Model	Application	Design
SGT-100 to 300	Power Generation	Single-Shaft
	Mechanical Drive	Twin-Shaft
SGT-400	Power Generation	Twin-Shaft
	Mechanical Drive	Twin-Shaft
SGT-500	Power Generation	Three-Shaft
	Mechanical Drive	Three-Shaft
SGT-800	Power Generation	Single-Shaft

Figure 1. Samples of different designs and applications of industrial gas turbines.

2. In addition to the complexity of diverse plant descriptions, there is also another dimension of the "level" in the plant model. The plant model gives an overview of the main components of the plant in a hierarchical fashion and comprises many levels. Each level consists of number of individual components and supports level-specific information. Within one level, each component contains its physical parameters relevant to computations. Fig. 2 gives an overview of a generic plant model at site level, plant level, system level and so on.

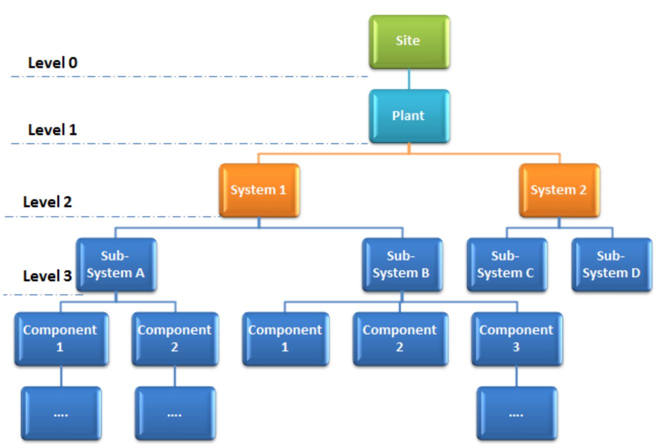

Figure 2. Hierarchical structure of a plant model.

3. It is important to note that the interaction and dependences of components within one level as well as between levels may be quite complex and hence creating the model requires greater expertise.
4. Available off-the-shelf statistical approaches as discussed by Ceschini (2002) and Márquez et al (2012) are based entirely on manual data gathering and manual assessment of scenarios for asset downtime. Such data is often contaminated by human factors and potentially by forced business incentives. Even today, service engineers still need to spend considerable time and effort calculating KPIs for a single site.

Considering all the challenges described above in terms of complexity, diverse configurations, interdependences of the plant model and data acquisition, the key idea is to simplify the computation of KPIs in two steps. Firstly, rather than addressing the KPIs of a plant at each level of its hierarchy in isolation, we introduce dedicated level-oriented rules that re-use KPIs already computed on one level for the computation of related KPIs on another. Secondly, in order to avoid re-phrasing KPI computation rules for each of the numerous different turbine configurations, we introduce an abstraction layer hiding the different configurations and define our KPI computation rules against the abstraction layer rather than the actual machine configurations. The abstraction layer will be based on a domain ontology describing turbines, their components and functions. The level-based KPI computation rules mentioned above will equally make use of the ontology providing the abstraction layer but will be encoded as Complex Event Processing (CEP) rules.

For a given specific plant the computation of actual KPIs does not utilize the abstract CEP rules expressed in terms of the ontology-based abstraction layer but rather depends on an instantiation step in which the abstract rule-base is instantiated for the specific plant and its configuration. This instantiation step is based on mappings between the concrete plants and the abstraction layer. The key observation here is that maintaining these mappings for a variety of different plants and configurations is a small task in comparison to maintaining the entire rule base for each plant and configuration. The paper follows with Section 2 describing the basic standards and KPI definitions used in the model for Industrial Turbines. Section 3 presents our case study and the proposed model-based solution architecture Section 4 introduces the basic concepts and application of ontology and complex event processing technology used for KPI computations. Section 5 presents results and serves for the evaluation and future developments. Finally, we conclude in Section 6.

2. KEY PERFORMANCE INDICATOR STANDARDS

Performance measurement is important to the management of industrial turbines. It identifies performance gaps between the desired and actual state and provides indications of the progress to meet those gaps. While KPIs are common tools for the measurement of system performance, the choice and definition of specific KPIs for a given system is not trivial.

For our KPI solution framework, we have revised and adopted definitions from the IEEE (2006) and ISO (1999) standards. Since we will rely on historic data in our computations, we introduce an additional parameter, "NoData", that deals with possible data gaps. The following is a list of basic KPIs used in our solution.

Period Hours (PH) – Time, in hours, in the period under consideration.

No Data Hours (NoData) – Time, in hours, where not all required data is available, here we use the term **PH*** for Period hours excluding the no data hours.

Available Hours (AH) – Time, in hours, during which the unit was capable of providing service, whether or not it was actually in-service, regardless of the capacity level that it can provide.

Service Hours (SH) – Time, in hours, during which the unit was in-service, i.e., it is electrically connected to the system and performing generation function. For gas turbines, this covers from main flame ignition through to flame extinction.

Reserve Shutdown/Service Hours (RSH) – Time, in hours, during which the unit was available, but not in service (Number of hours when the gas turbine is available but there is no demand).

Unavailable Hours (UH) – Time, in hours, during which the unit was not capable of operation. The unavailable state persists until the unit is made available for operation, either by being synchronized to the system (in-service state) or by being placed in the reserve shutdown state.

Planned Outage Hours (POH) – Time, in hours, during which the unit (or a major item of equipment) was originally scheduled for a planned outage with a pre-determined duration plus the extension of planned work beyond this pre-determined duration. Note that the extension due to either a condition discovered during a planned outage or a startup failure would result in a forced (unplanned) outage.

Forced Outage Hours (FOH) – Time, in hours, during which the unit was unavailable due to a component failure or another condition that requires the unit to be removed from service immediately or before the next planned outage. Fig. 3 shows a hierarchical overview of these definitions that forms the basis of the solution framework.

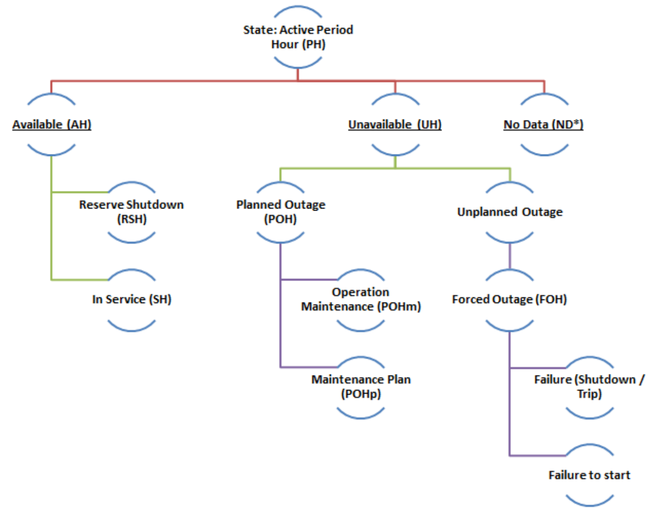

Figure 3. Overview of adopted KPI definitions

Using the four KPIs defined above, we can compute the following factor KPIs:

Availability Factor (AF) – Probability that the unit will be usable at a point in time based on past experience:

$$AF = [AH / (PH^*)] \times 100\%$$

Unavailability Factor (UF) – Probability that the unit will be unusable at a point in time based on past experience:

$$UF = [UH / (PH^*)] \times 100\%$$

Reliability Factor (RF) – Probability that the unit will not be in a forced outage condition based on past experience:

$$RF = [(PH^* - FOH) / (PH^*)] \times 100\%$$

Service Factor (SF) - Probability that the unit will be in an

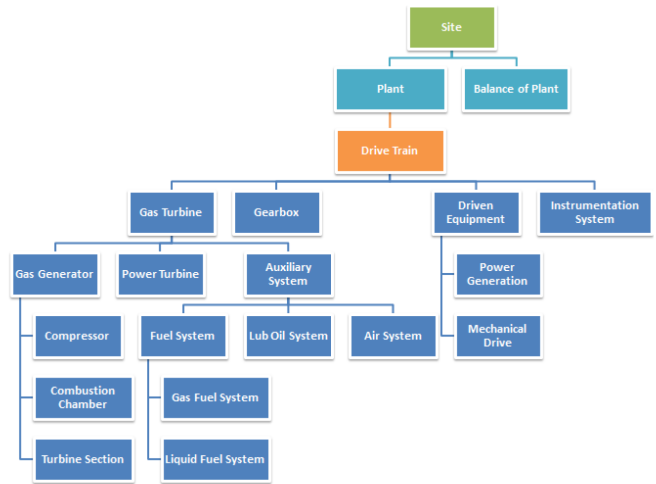

Figure 4. Use-case of an industrial plant model

operating condition based on the past experience:

$$SF = [SH / (PH^*)] \times 100\%$$

Forced Outage Factor (FOF) – Probability that the unit will be in a forced outage condition based on past experience:

$$FOF = [FOH / (PH^*)] \times 100\%$$

Mean Time Between Failures (MTBF) – Average time between failures initiating a forced outage based on the past experience. Here, FON is the number of forced outages:

$$MTBF = SH / FON$$

For simplicity, we also use N/A for indicating the case where the KPI value cannot be correctly computed, e.g., PH == 0, PH == NoData or FON == 0.

3. A NOVEL APPROACH FOR KPI WITH APPLICATION TO INDUSTRIAL GAS TURBINES

The proposed approach has been applied to a fleet of Siemens industrial gas turbines located at different sites around the globe.

3.1. Case Description

At any given site, see Fig. 4, the plant system consist of two subsystems, namely drive train and balance of plant. Based on the configuration, each drive train subsystem comprises i) a *driver package* (for example, gas turbines or steam turbines), ii) *driven equipment* (for example, a compressor or pump), and iii) *gearbox*. Furthermore, within a driver package, a turbine component may consist of a *gas generator*, *power turbine and auxiliary system*. Each functional component includes physical parameters and threshold values, for example, speed, load, temperature etc. Each of this physical parameter needs to be configured. This configuration is a mapping of parameters to one or several sensors (for example, Two-out-of-three) and a setting threshold values.

The following section describes the solution architecture. Details on the models used for our approach will be introduced in Sections 4 and 5.

3.2. Solution Architecture

Modeling a plant system is a critical step for constructing KPIs that accurately reflect the impact of actions taken to manage the plant. The proposed approach uses two well-established paradigms from AI, namely ontology and complex event processing, to be discussed separately in the following sections.

In Fig. 5 we present the overall solution architecture: a domain ontology is used to represent turbine configurations and the relationships between different physical components in the plant and their function and performance variables (such as speed, main flame, active power, etc.). The performance parameters describe the primary behavior of the plant at different levels. We store these configurations in a separate database (turbine configuration database) for easy access. In the next step, we model complex events and formulate abstract state indicators rules and update rules for each node. These rules are abstract in that they are defined w.r.t. the ontology-based vocabulary from the configuration database rather than data specific to any individual plant coming from the remote monitoring service database.

In order to actually compute KPIs for a specific plant however, i.e., apply the rules, we instantiate the abstract set of CEP rules with the concrete plant information using a semantic mapping mechanism. Once the instantiation is completed, we proceed with the KPI computation procedure.

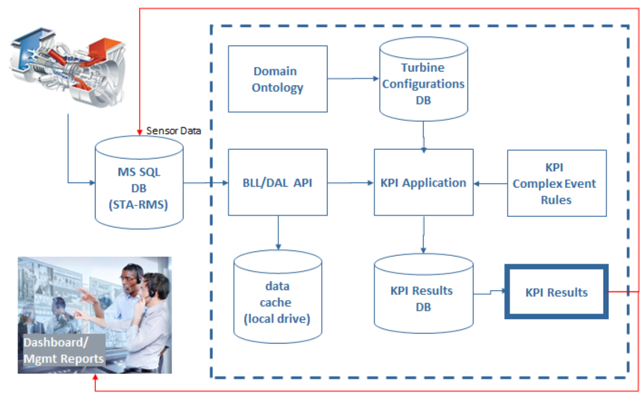

Figure 5. KPI System Architecture

The computation framework uses operational sensor data as well as event streams for its state evaluation. Because of the large amount of data, we use a data cache for storing and post-processing.

4. ONTOLOGY

As described above, our approach to KPI computation rests on an abstraction layer by means of which a comparatively small set of abstract rules can be instantiated to match a very large set of turbines and their various concrete configurations. At the core of this abstraction layer lies a domain ontology that represents basic knowledge about the compositional structure of plants, types of its components, and their function.

Ontologies are logic-based knowledge representation (KR) formalisms that evolved from frame-systems, see Baader (2003). Chandrasekaran and others. (1999) characterize ontologies as “a formal, explicit specification of a shared conceptualization". They usually represent the core notions of a domain of discourse and the relations existing among them. A key advantage of ontologies over many other KR formalisms is their formally well-defined semantics. These enable so-called reasoners to derive implicit knowledge from the explicit ontology statements, detect redundancies and inconsistencies, and discover relationships that may not have been clear to the author of the ontology in the first place. Currently, the most commonly used ontology formalism is OWL[1] and its sublanguages.

Some additional characteristics of Ontology, which address the key challenges in the turbo-machinery domain, are as follows as stated by Ming and Jie (2002):

- Ontologies clarify the structure of knowledge and devices for an effective KR system.
- They separate factual knowledge about the domain from problem-solving knowledge.
- They facilitate sharing and re-using knowledge as well as interoperability of information resources between humans and software agents.

For the purpose of computing KPIs we have developed a domain ontology of turbines, their components and functions. Note that multiple kinds of relationships different from 'is-a' can easily be expressed in OWL. Fig. 6 illustrates a basic example with Driver and Driven equipment as classes, Gas turbine as a subclass and SGT-800 as an object, called 'individual' in OWL. Object properties, such as 'provides power' in the example, establish links between classes or individuals. The combination of object properties and subclass relationships now give rise to additional implicit relationships, for example: "If SGT-800 provides power" then this implies "Generator requires power". A key advantage of OWL is that all implicit knowledge is fully automatically taken into account by the reasoner. Hence, redundancies and inherent contradictions are detected automatically, leading ultimately to smaller and more easily maintainable models.

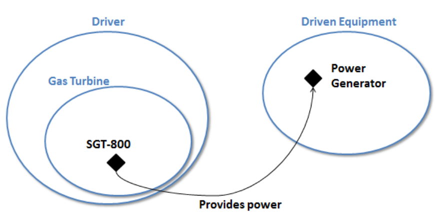

Figure 6. Ontology example to Turbines

4.1. Domain Ontology Design

Our domain ontology comprises several ontology modules of which the largest two are the following:

- *Train Ontology*: This describes the internal structure of the plant i.e. its components and sub-components. For example: Burner is a component of a combustor in a Gas Turbine. In addition to this, the ontology also specifies the functional purpose of each component. For example: Main flame is in hot gas path. The ontology is expressed in OWL 2 DL.
- *Sensor Ontology*: This ontology lists the sensor information, its measured values, sensor type and its location. For example: GT speed sensor measures the shaft rotor speed of the turbine. It also accompanies the observational characteristics (such as measurement range etc.) and measurement characteristics (such as measurement unit etc.) of each sensor. Sensor ontology is also expressed in OWL 2 DL.

In this way, we developed a comprehensive model of the domain by combining the above mentioned ontologies. Fig. 7 depicts the consolidated ontology used for accessing data based solely on the domain model and use them in the rule based component as a knowledge-base.

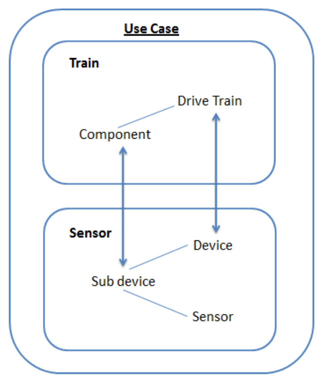

Figure 7. Train ontology design

5. COMPLEX EVENT PROCESSING

Complex Event Processing (CEP) is a paradigm of choice for many monitoring and reactive applications. It supports decentralized information sources by deploying tagging and sensing technology along with integration to real-world objects. CEP helps to build highly scalable and dynamic systems by decoupling the provider and receiver of the information and mediates in form of *events*. Temporal relations can also be specified by using correlation rules (often called *Event Patterns*) as mentioned by Robins (2010). CEP also benefits the scalability of the system by reducing the massive event load through stepwise correlation of events.

In general, CEP is used to generate new set of complex events by aggregation and composition. Its processing promotes detection of a plant-significant situation, which typically involves a collection of evaluation conditions and constraints over an event set as founded by Wasserkrug, S., Gal, A., Etzion, O., & Turchin, Y. (2008). Another

[1] www.w3.org/2004/**OWL**

characteristic of CEP is event transformation, filtering, enrichment, pattern recognition, routing, validation etc. Figure 5. serves as an example of constructing new signal event processing rules for speed and load of turbines by using sensor data and events.

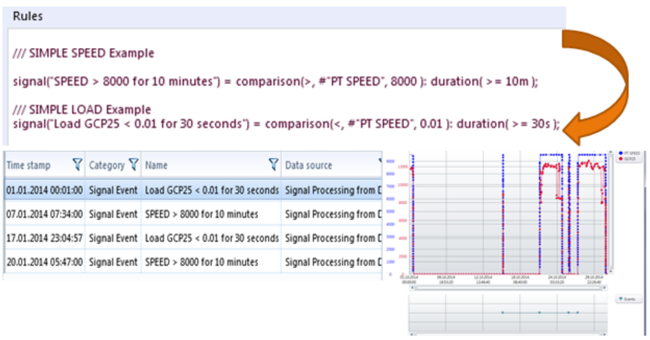

Figure 8. Alarms for high speed and low load using CEP Rules

For our solution, we have devised two set of abstract rules that are encoded as CEP rules to identify the state of a plant at different levels. The following sections briefly discuss the implementation and purpose of these rules.

5.1. State Indicator Rules

The state indicator rules define how a given state of a plant is acquired that is useful for our computation. These states are determined using physical parameters, such speed, load, temperature etc. Every plant has its own set of threshold values and specific events from the control system to indicate its performance. These features in our case study is encoded in the abstraction layer i.e. domain ontology. By using expert knowledge, here we formulate abstract set of state rules that can incorporate all configurations of turbines. The three important state indicators are; i) State of Service Hours (SH), ii) State of Outage Hours (OH), and iii) State of Start Attempt (SA) / Start Failure (SF) / Start Success (SS).

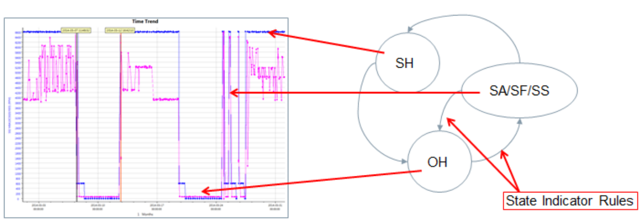

Figure 9. State Indicator Rules

In Figure 9, we have a KPI state machine with State indicator rules on edges, for example. a drive train can move from "start-success" state to "service hours" state if the rotor speed is greater than *#value1* RPM and generator load is greater than *#value2* MW. Here the tags *value1*and *value2* will be replaced upon instantiation.

Figure 10 gives an overview of the states required for computation. For outage hours (OH), we can define more specifiers. For example: reserve shut down (RSH), forced outage (FOH), and planned outage (POH). For our implementation, we do not go into the details of the outage hours at the moment. Though the solution is flexible enough to identify these states based on the manual entries by the service engineers.

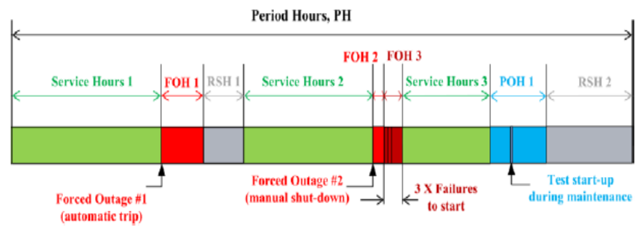

Figure 10. Overview of State Specifiers

5.2. Level Update Rules

The level update rules are formulated to capture the state dependencies at one level of the plant to the other. For example, any entry of outage specifier interval on one system level will lead to the respective "updates" of the outage specifier intervals on the other system levels. One concrete case would be entering a forced outage interval in the gear box. This will lead to a forced outage interval in the drive train, but will be treated as a reserve shutdown interval in the driver package.

This indicates that as soon as an outage specifier, e.g., RSH or FOH, is added to one component, we have to perform so-called Level Update Rules. Figure 11 shows the update mechanism for a drive train at level 1 and gas turbine and generator at level 2. The rules can be:

- If Drive Train is in reserve shutdown state, then gas turbine and generator at level 2 are updated to reserve shutdown state as well.
- If Generator is in forced outage state, then driver train at level 1 is updated to the same state whereas the gas turbine at the same level is updated as reserve shutdown.

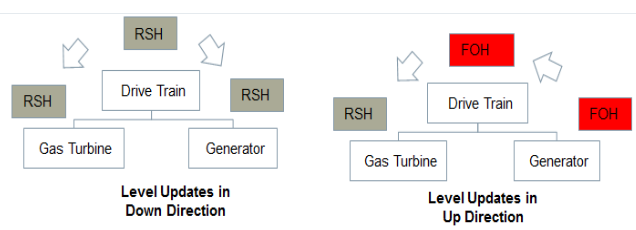

Figure 11. Example of level update rules (Part I)

Figure 12. gives an another view of the above mentioned example. This is a visualization of the state for every level

and unit as identified by state CEP rules and its updated version as specified manually by the engineer.

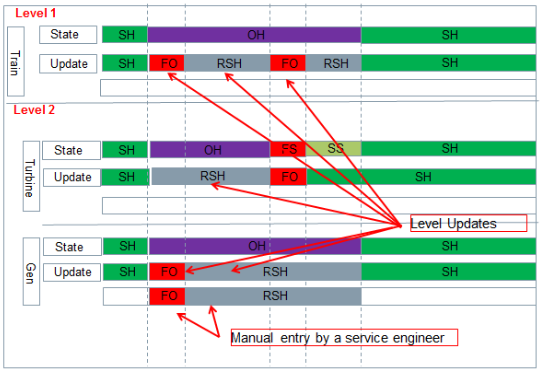

Figure 12. Example of level update rules (Part II)

6. RESULTS

The first set of results using our KPI application provide an availability and reliability comparisons between three design model of gas turbine by year. These indicators play an important role in decision making and put a real challenge when the system model is complex and involves large set of engineering rules. In comparison to the manual calculations, our results are more reliable and accurate because of the adoption of ontology based configurations and reusable rule production system.

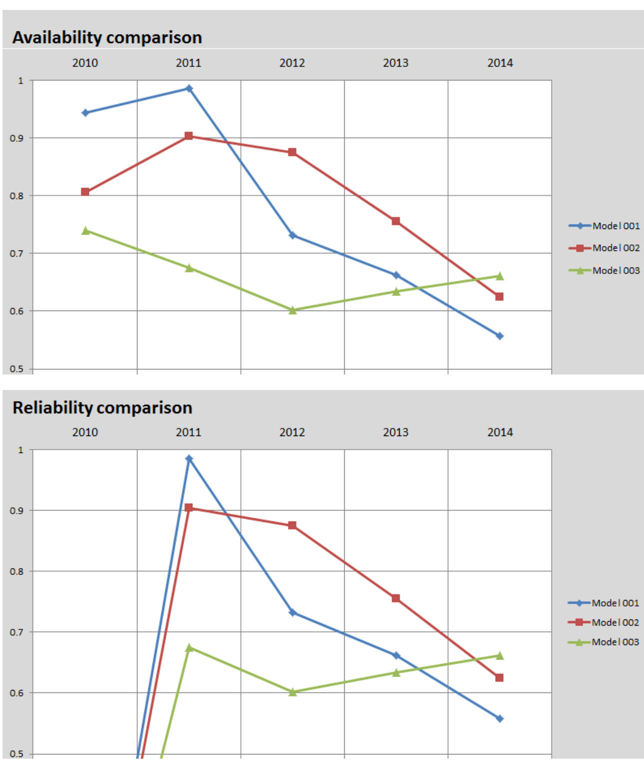

Figure 13. Availability and Reliability Comparison by turbine type and year

Another visualization of results is with respect to a specific drive train and its respective units within the hierarchy. Most of the recent methodologies do not consider the component and system level setup. Whereas our approach facilitates the engineers and managers to look up for indices at any given hierarchy and package level. Another highlight is the use of sensor data and events together to detect the state of the machine. Therefore, our results are more accurate, reliable and justifiable than any other traditional approaches.

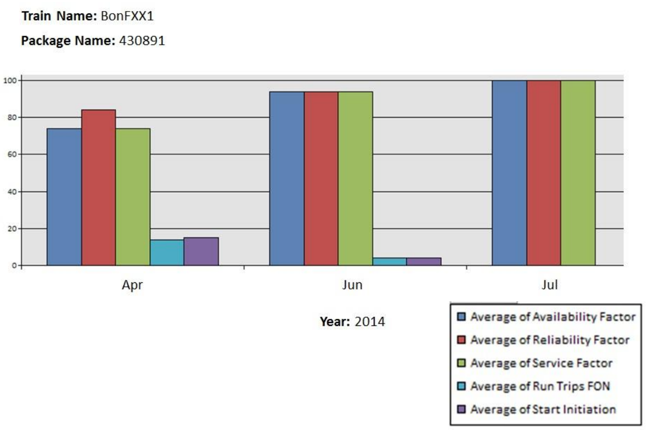

Figure 14. KPIs per drive train and its units.

Here we incorporate the high level performance indices at the train level where we specifically visualize for the unavailability, availability and no data states for a specific unit. Such kind of visualization is readily available at the dashboard for high level managers and is also helpful to detect malfunctions of the data collectors on site.

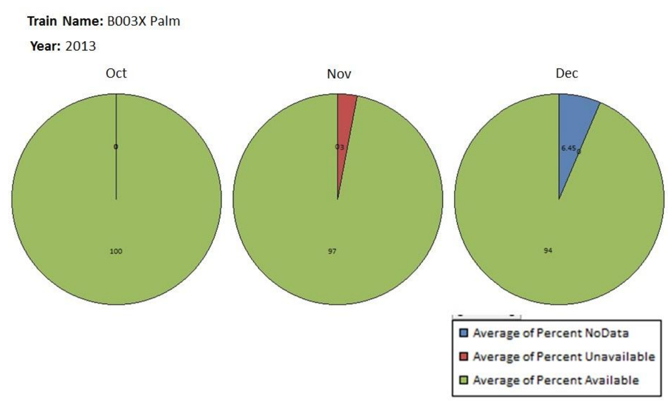

Figure 15. KPI per unit

Similarly, using our approach and generated KPI result database, we can provide different views based on site region or country, customer, driver, driven unit, etc. We claim that our approach is unique and fits best for calculating KPIs in different fashions and provides customized visualization of results that could be integrated as a part of monitoring dashboard services. Fig. 16 shows another view of KPI results filtered for a service region.

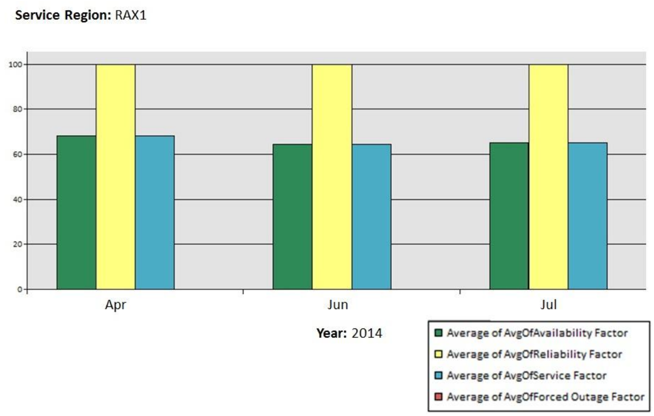

Figure 16. KPI results based on ServiceRegion

7. CONCLUSION

We demonstrated a KPI systems approach using an abstraction layer based on a domain ontology and complex event processing technology. This allows us to adopt our KPI computations for different turbine types, different control system types and incorporate additional information available from the external systems. We extended the standard definitions from IEEE and ISO to be used for our case-study. The solution makes use of the sensor data and events from the control system to identify turbine states and perform the computations. The solution also provides different visualization of the results. The presented architecture is distributed, extensible and scalable. The computations are automated and have minimum dependency on user-interaction. Hence, they provide reliable and trustable results for decision-making. By including the maintenance calendar, we can also automate the computation for the reserve shutdown and planned outage hours. Also the inclusion of events from the control system that specify for the internal and external outage would add value to the application. For the future, the KPI application can be integrated with the remote diagnostic solution framework to evaluate its potential.

ACKNOWLEDGEMENT

We would like to acknowledge Mr. Michal Skubacz for the support and encouragement. Their valuable feedback sessions were important in development of rule-base and software system.

REFERENCES

Ceschini, G. F., & Saccardi, D. (2002). Availability centered maintenance (ACM), an integrated approach. In *Reliability and Maintainability Symposium, 2002. Proceedings. Annual* (pp. 26-31). IEEE.

Ding, S. X., Yin, S., Peng, K., Hao, H., & Shen, B. (2013). A novel scheme for key performance indicator prediction and diagnosis with application to an industrial hot strip mill. Industrial Informatics, IEEE Transactions on, 9(4), 2239-2247.

Odgaard, P. F., Stoustrup, J., & Kinnaert, M. (2013). Fault-tolerant control of wind turbines: A benchmark model. Control Systems Technology, IEEE Transactions on, 21(4), 1168-1182.

Márquez, F. P. G., Tobias, A. M., Pérez, J. M. P., & Papaelias, M. (2012). Condition monitoring of wind turbines: Techniques and methods. Renewable Energy, 46, 169-178.

Forsthoffer, W. E. (2011). Forsthoffer's Best Practice Handbook for Rotating Machinery. Elsevier.

Ceschini, G. F., & Carlevaro, F. (2002, January). Gas turbine maintenance policy: a statistical methodology to prove interdependency between number of starts and running hours. In *ASME Turbo Expo 2002: Power for Land, Sea, and Air* (pp. 1137-1142). American Society of Mechanical Engineers.

IEEE Standard Definitions for Use in Reporting Electric Generating Unit Reliability, Availability, and Productivity. IEEE Std 762™-2006. IEEE Power Engineering Soc.

Gas turbines - Procurement - Part 9: Reliability, availability, maintainability and safety. BS ISO 3977-9:1999. British Standards.

Baader, F, & Calvanese, D., & McGuinness, D., &. Nardi, D., & Patel-Schneider, P., (2003) The Description Logic Handbook. Cambridge University Press.

Chandrasekaran, B., Josephson, J. R., & Benjamins, V. R. (1999). What are ontologies, and why do we need them?. IEEE Intelligent systems, 14(1), 20-26.

Ming, D. Z. T. S. Z., & Jie, Y. D. C. (2002). Overview of Ontology. Acta Scicentiarum Naturalum Universitis Pekinesis, 38(9), 728-730.

Robins, D. (2010, February). Complex event processing. In Second International Workshop on Education Technology and Computer Science. Wuhan.

Wasserkrug, S., Gal, A., Etzion, O., & Turchin, Y. (2008, July). Complex event processing over uncertain data. In Proceedings of the second international conference on Distributed event-based systems (pp. 253-264). ACM.

Luckham, D. (2002). *The power of events* (Vol. 204). Reading: Addison-Wesley.