SelSus: Towards A Reference Architecture for Diagnostics and Predictive Maintenance Using Smart Manufacturing Devices

Mohamed S. Sayed, Niels Lohse
Wolfson School of Mechanical and Manufacturing Engineering
Loughborough University
Leicestershire, UK
[M.Sayed|N.Lohse]@Lboro.ac.uk

Nicolaj Søndberg–Jeppesen and Anders L. Madsen
HUGIN EXPERT A/S
Gasværksvej 5
9000 Aalborg, Denmark
[nsj|alm]@hugin.com

Abstract*—We propose a reference architecture, SelSus (SELf-SUStaining Manufacturing Systems) that aims to enable the provisioning of diagnostic and prognostic capabilities in manufacturing systems that utilize the notions of "smart" automation devices.*

Keywords—Diagnostics; Prognostics; Maintenance; Manufacturing; Component-based design

I. INTRODUCTION

Manufacturing organizations today are progressively pushing to reduce system downtime and unplanned stoppages in response to the increasingly competitive and changing markets. This need for more predictable and quick-to-recover manufacturing systems coupled with the increasing availability of cheaper, more powerful computational resources and sensory data on the shop-floor has resulted in an ever increasing interest in the provisioning of condition-based monitoring and smart diagnostic capabilities in manufacturing systems across all levels of the manufacturing enterprise.

On the other hand a long term direction in the design and provisioning of automation devices has been moving towards modular, component-based systems where component suppliers aim to provide smart components with their own control capabilities that can easily be integrated into a wider system.

Within this context, the SelSus project (SELf-SUStaining Manufacturing Systems) aims to enable component-based diagnosis and prognosis both on the device level and the system level through utilizing the notion of smart components and sensors accompanied with localized data processing and reasoning capabilities. In this paper we describe a novel architecture for diagnostics and predictive maintenance in manufacturing systems utilizing the concept of smart automation devices, or SelComps. In the next sections we give a brief introduction about some of the related work and then introduce the main concepts in the proposed architecture.

II. RELATED WORK

Maintenace and diagnostics have been extensively studied in the context of manufacturing systems. Some studies tried to establish an understanding of the relationship between maintenance strategies and the various perforamce metrics of manufacturing organisations. For example in [1] and [2] the positive impact of following total productive maintenance practices on manufacturing performance, specicifcally on cost reduction and enhanced production quality is established. Moreover, the changing role of maintenance as a central factor in lifecycle management is discussed in [3] then a maintenance framework for management cycles of maintenance activities is suggested in order to enable the idetification of the main technical challenges associated with the changing role of maintenance along with the technological advances that can potentially support this change.

Various approaches have been reported in literature aiming to realize the vision for preictive or condition-based maintenance. For example in [4] a predictive maintenance policy is proposed that is based on trying to achieve more accurate predictions out of equipment degradation and reliability models through utilising realtime operational degradation sensory signals. Other work has been focused on the optimization of maintenance activities scheduling based on predictive degradation models combined with various optimization methods. In [5] discrete event simulations are used to are used to estimate the effects of maintenace scheudles on the systems performance. Grey theory and evaluation diagnosis were used in [6] in order to construct preventive maintenace forecasting models for semicoductor manufacturing machine failures. A predictive maintenance decision support tool is presented in [7] that through the use of genetic algorithms tries to minimise the optimize system availability, and cost with regard to system-maintenance constraints. Similarly in [8] genetic algorithms are used in combination with Monte Carlo simulation-based degradation models in order to optimize both profit and system availability.

It is beyond the scope of this paper to give a comprehensive review for the state of the art of predictive and condition-based maintenance. However, interested readers are referred to review studies such as [9],[10] and [11].

The reasoning approach that is suggested to be used for diagnostic and prognostic reasoning purposes in this architecture is Bayesian networks [12],[13]. The diagnostic process in Bayesian Networks involves inferring the probability of the occurrence of an unobservable fault hypothesis based on the measured (observed) evidence (Symptoms). In other words inferring the probability of X_i being in a certain state, given the observed evidence *E*, which is expressed as $P_r(X_i = x|E)$ where $E = (y_1, \dots, y_m)$, y_j being the observed state of the variable Y_j and $(Y_1, \dots, Y_m) \subset (X_1, \dots, X_n)$. An important feature of Bayesian Networks that makes their use in engineering diagnostic problems particularly attractive is the possibilities for models construction. Bayesian models can either be constructed from expert domain knowledge or from operational data or a combination of both.

The modelling approach that is used in constructing the Bayesian models is Object-oriented Bayesian networks (OOBN)[14]. This approach enables component-based encapsulated modelling on the component level, while enabling the individual component models to be integrated into a wider system-level model that can be used for reasoning on the system level.

III. Overall Architecture Approach

The overall strategy of the architecture is based on the concept of independent smart automation components which can be integrated bottom-up into a wider automation system. The underlying intention is to enable the providers of automation components to offer more functionality as part of their devices but also to allow these devices to be better and faster integrated into the specific context of the wider automation system they will be used in. Hence, the overall concept behind the solution strategy is based on smart components or SelComps which represent the automation components, smart sensor nodes to allow wider data gathering, and mobile input and output devices in the base layer which get integrated into a wider system using a common communication infrastructure.
Furthermore, SelSus architecture supports three levels of communication which can be roughly categorized into data, information and knowledge as shown in Figure 1.

A. SelComp Concept

In order to be able to provide prognostic and diagnostic functionalities on the device level, the automation components or devices in SelSus need to be encapsulated using the concept of SelComp. The aim is to encapsulate the automation component with its own control and sensory capabilities including some of its own local data processing, diagnostic and prognostic capabilities. The overall system will be an integration of a number of SelComps with the added system-level functionalities. This should enable diagnostic and prognostic reasoning to be conducted on two levels: locally on the SelComp level where the only issues within the scope of the SelComp itself are possible to be detected and analyzed; and globally on the System-level where various SelComps make some or all of their observations available for system-level reasoning. This will enable system-level visibility in capturing and analyzing the state of the system as a whole, covering the various influences between individual SelComps. Figure 1 shows an overview of the generic internal architecture of the SelComp and how it relates to the system-level functionalities within SelSus.

Typically a smart automation device (SelComp) will have its own actuators, sensors and possibly a dedicated Human-Machin Interface (HMI) which are controlled and monitored through the device's control environment. As shown in Figure 1 the SelComp concept is based on the four layers of content abstraction. The first layer, the Control Layer is the real-time control environment of the device which controls the various actuators in the device, as well as capturing raw sensory data from the device sensors and human inputs from operators through the HMI. All of the content on this layer is control-related and represents raw content that cannot directly be used for meaningful maintenance-related decision-making.

Therefore, within the SelComp these raw control signals are then annotated to give them contextual meaning within the device. This annotation should enable the identification of the context the signal was generated in. This can be in the form of providing information such as device, process, timing, etc. The annotated signals form what is called the data layer and can then be consumed inside the SelComp based on their contextual information for further reasoning up the content abstraction hierarchy.

This raw annotated data should then be transformed into meaningful events that convey "information" about critical and periodic events. To do that the SelComp will need to have its own internal statistical data processing functionalities that can analyze trends and detect abnormalities in the various data streams. In the information layer these events are annotated to give them their real-world context using the reference data model enriching them with (Product, Process and Equipment) information of which a localized version is retained in the SelComp (Device Self-Description). The SelComp can choose to make some of these events available to the System level functionalities through advertising for topics and services that interested data consumers can subscribe to. Although events generated in this information layer of content abstraction are expected to convey significant meaning about the health status of the device, however such events are unlikely to be useful on their own for diagnostic and prognostic reasoning.

In the knowledge layer the events passed from the information layer are expected to be reasoned with in order to establish estimations about the state of the system (diagnostic and prognostic reasoning). The diagnostic and prognostic reasoning capabilities cannot be derived only from generated

events. There is a need to incorporate engineering domain knowledge and the designers' expertise in constructing the models that can then be used to carry out diagnostic and prognostic reasoning. This gives rise to the highest level of content abstraction, the knowledge layer. The knowledge layer is expected to incorporate the design knowledge about the system's behavior in the form of diagnostic and degradation models. Within this architecture the diagnostic and prognostic modelling approach of choice is Bayesian Networks [12],[13]. The events generated in the information layer are then passed into the Bayesian reasoning engine to enable the diagnostic and prognostic probabilistic reasoning that leads into estimating the state of the device based on these observed events. The structure of the model is expected to be derived from engineering domain knowledge but the parameters are expected to be continuously fine-tuned with incoming observations in the form of events. This Bayesian model and the associated reasoning and learning processes constitute the knowledge layer. The SelComp is expected to share some of the results from the diagnostic and prognostic reasoning with

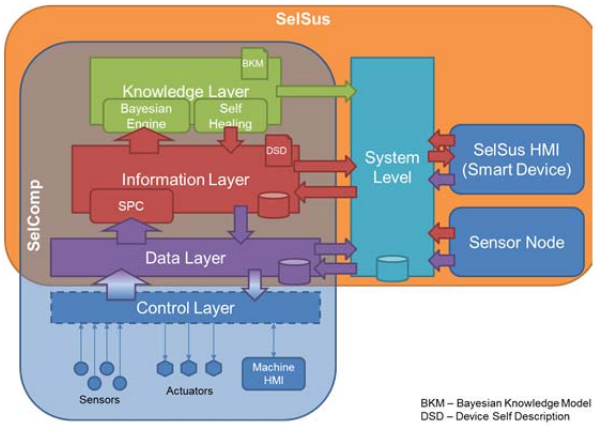

Figure 1 SelComp Concept

the wider system-level functionalities. Similarly the local SelComp Bayesian model (Knowledge Model) itself is also expected to be shared with the wider system in order to enable the formation of component-based, system-level Bayesian knowledge models.

B. SelComp Components

In this section the main building blocks or the internal software components that are required to provide the functionalities described in the previous section will be briefly discussed. The SelComp concept is defined through the ability to provide the aforementioned four content abstraction layers. Therefore; for the SelComp to be able to provide the intended functionalities it will need to have internal components that handle and transform each of the content layers. Firstly to handle the interface with the control environment on the data layer, there need to be an interfacing component that handles all the interactions with the control layer and provides meaningful annotated data items to the consumption of the higher abstraction levels. The component that will realize this functionality is the control Interface. Once the data is generated in the data layer it will then need to be turned into information or events in the information layer. This transformation functionality will need to be carried out by a statistical analysis engine. On the knowledge layer there will be a need for a reasoning engine that handles the diagnostic and prognostic reasoning with the Bayesian models.

The aforementioned three components are the fundamental building blocks for the SelComp concept. However, there are other supporting functionalities that need to be present in order to enable the SelComp concept to be realized. Figure 2 shows the internal structure of the SelComp. In the next subsections these components will be briefly explained.

1) Control Interface

To capture relevant process information for diagnostic purposes, specific variables or signals within the PLC or Industrial Control system will be observed and time stamped. This component has to act like a passive observer that cannot affect or intervene in the control behavior of the core control system. On the other communication direction, i.e from SelSus to the control environment (Self-Healing scenarios) recipe adjustments can be deployed only for pre-defined parameters which need to be present within the control program. The interface will have to synchronize with the Industrial Controller to ensure recipe adjustments are deployed in a timely and well controlled manner. In most cases this will mean that recipe adjustments can only be deployed at a particular point during the control cycle of the Industrial Control Program.

2) Statistical Analysis Engine

The statistical analysis engine imports a well-structured data set and applies various statistical models to analyze whether the data items or wider processes behave normally, are changing or exhibit abnormal behavior.

The specific mathematical models required will be largely dependent on the specific nature of the SelComp and process it is responsible for. It is expected that component developers will embed different statistical tools as part of their components. This is seen as one approach to add value to the SelComp. The statistical functions which could be embedded into the SelComp are not limited but need to cover at the very least the required discretization to prepare the data points for use as part of the Bayesian Diagnosis as this forms the basic requirement for the data-to-information transformation.

3) Reasoning Engine

The reasoning engine is used to update Bayesian network knowledge models for different kinds of questions. This could include root cause analysis as part of diagnostics, prediction of likely future faults as part of prognosis as well as identification of most promising actions. Updating of the knowledge model will take historical information and update the conditional probabilities of the Bayesian knowledge model based on these past known observations.

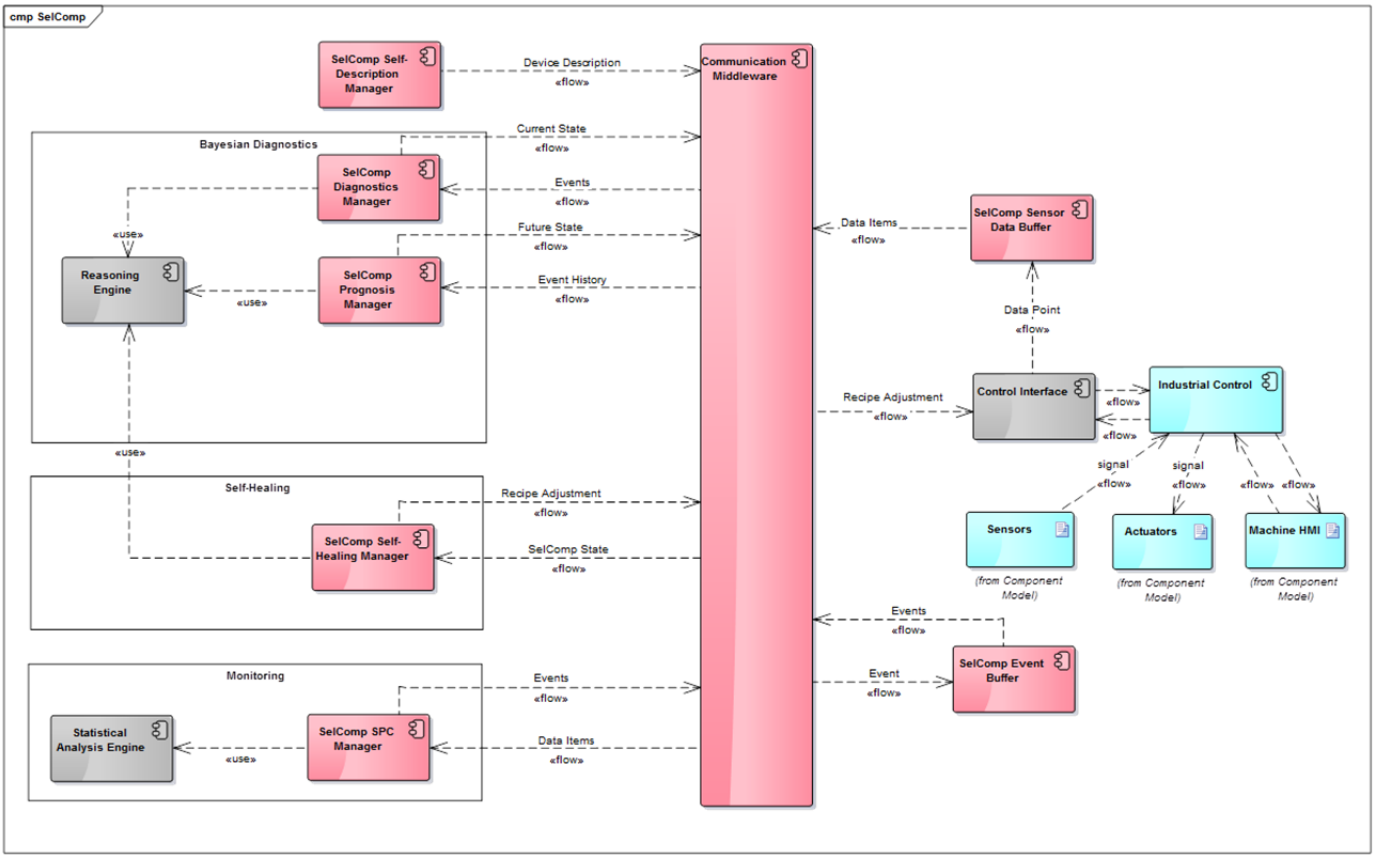

Figure 2 SelComp Internal Structure

4) Communication Middleware

The purpose of the communication middleware is to enable SelSus-compatible communication within and between various SelComp devices, allowing different software components to request and send data, information and knowledge between each other. The main aim of the communication middleware however, is to enable the SelComps to communicate events and data items with the overall SelSus system functionality. A publish and subscribe communication protocol is expected to be used where data providers (SelComps in this case) publish events into predefined topics while interested data consumers (on the system level) can subscribe to the topics that are relevant to their functionality.

a) SelComp Self-Description Manager

The Device Self-Description Manager is responsible for maintaining and communicating the self-description file of the device to the higher level system and other participants in the same network. The device declares itself and all the services it offers to the other participants in the network. The device can also supply its self-description on request.

b) SelComp Knowledge Model Manager

The purpose of this manager is to manage the release and update of the Bayesian knowledge model for the SelComp. The Object Oriented Bayesian Network (OOBN) model of the component can be requested from external applications (e.g. the system level diagnostic engine). The update of the model could either on request, on event or on a regular basis. Example for 'on-request' would be to update the model just before a Bayesian inference task is being carried out. 'On-event' could be in cases that a new maintenance case is submitted which may affect the probability tables of the hidden states. 'Cyclic' behavior could be required to regularly update the status probabilities of the observable states in the Bayesian model.

The manager will have a number of different updating functions and will request the required event history information from the SelComp Event Buffer. The manager will then call the model learning function of the Bayesian diagnostic engine to update the conditional probabilities of the model. The result will be stored as the new Object Oriented Bayesian Model of the SelComp and released to external source when requested.

c) SelComp Event Buffer

Stores SelComp internal condition information for a finite amount of time and makes it accessible for internal and external use. The component will store all condition information about itself for a predefined amount of time which could be dependent on the amount of available storage space. It will provide a number of services to retrieve and send data to internal and external consumers. The need for this event

buffer arises from the fact that most events will tend to be high frequency events due to the high frequency of signal generation on the industrial control level. However, for most analysis purposes there will be a need to analyze these events over a long period of time in order to establish trends and abnormalities.

C. System Integration Concept

The manufacturing system being monitored by SelSus system is expected to be formed of a number of SelComps that are integrated to form the SelSus system level functionality. Individual SelComps will be designed in a way that would allow the local SelSus capabilities to function even when the overall SelSus system becomes unavailable. The system integration will be triggered either during the original system integration (e.g. after system build) or when a change to the system (re-configuration, replacement, refurbishment) is being carried out. SelComps make some of their internal information available to the wider system through advertising for the available services using a discovery service that enables newly introduced SelComps to advertise their presence in the system along with the information they choose to make available externally. The data communication service enables the synchronization and persistence of data between the local and the system level storages.

Following the notions of component-based modelling, the aim here is to enable modular, component-based Bayesian models to be constructed on the SelComp level while accommodating for the possibility of integrating these local models into a wider system-level model that covers the overall system. The component models should ideally be self-sufficient and only require information that is available from the SelComp itself. Instances of the component models are expected to be communicated from the SelComp level to the system level whenever there is a change in the SelComp model structure or parameters. This separation should allow individual SelComps to independently reason about their own states if needed without necessarily having to be plugged in as part of a wider SelSus system.

The main aim behind this is to enable independent SelComps to operate normally even when they are not connected to an overall SelSus system. This can only be true if the diagnostic and prognostic models along with the reasoning process are entirely self-contained within each individual SelComp and do not depend in their operation on external resources, either in the form of observations or reasoning and computation.

D. System-level Components

The system-level functionality requirements are driven following the same approach in specifying the requirements of the Slecomp. That is the separation between the three layers of content abstraction: data, information and knowledge. The system-level functionality results from the integration of SelComps and sensor nodes. Figure 3 shows the main building blocks on the system level. The sensor nodes can stream data which requires that the system level should have data processing capability to transform such data streams into events. This gives rise to a statistical analysis engine similar to the one on the SelComp level. The SelComps are expected to

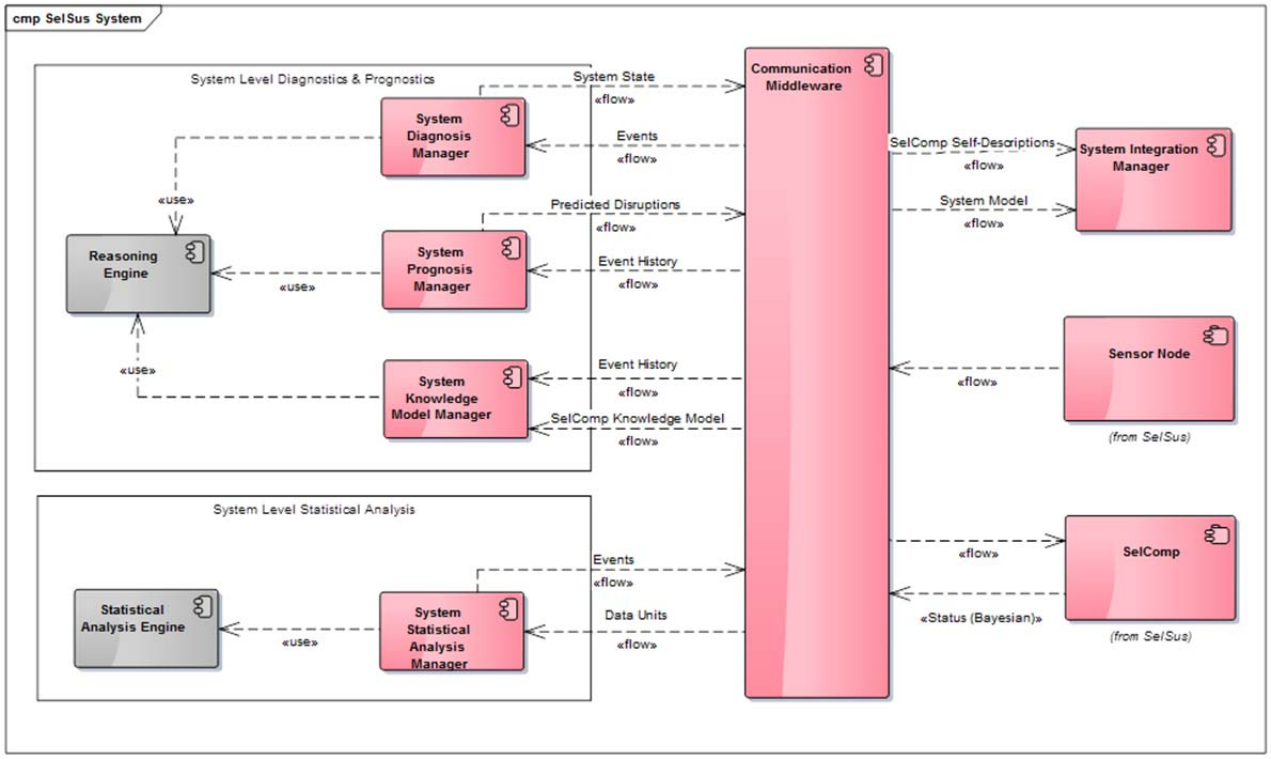

Figure 3 System-level Components

communicate events (information). Therefore, similar to the SelComp, on the highest content abstraction level, the knowledge level Bayesian models will be used to carry out diagnostic and prognostic reasoning. The only difference is that on the system level, the system knowledge models will be a result of the integration of the individual SelComp models that constitute the system. This is made possible through the use of Object Oriented Bayesian Networks (OOBN). This will necessitate the need for a system knowledge manager that ensures the integration of all the SelComp local knowledge models into the system-level model. Another important requirement will be enabling and triggering the integration of the system during the configuration and reconfiguration process. This gives rise to a system integration manager.

CONCLUSIONS

The main concept for a reference architecture to enable the provisioning of diagnostic and predictive maintenance capabilities in manufacturing systems utilizing the localized computational and sensory capabilities of smart automation devices was proposed. The proposed architecture aims to enable Bayesian Networks-based diagnosis and prognosis both on the automation device level as well as on the system level through decoupling the modelling and reasoning processes on the two levels. The main building blocks and the internal structure of the smart automation device within the context of this architecture were motivated and described. The proposed architecture is currently being deployed and tested as part of an ongoing European collaborative research project. Future work will include further testing of the performance of the resultant systems and benchmarking against similar existing standards.

ACKNOWLEDGMENT

This work is part of the project ”Health Monitoring and Life-Long Capability Management for SELf-SUStaining Manufacturing Systems (SelSus)” which is funded by the Commission of the European Communities under the 7th Framework Programme, Grant agreement no: 609382. We thank the project partners for their valuable comments and discussions.

REFERENCES

[1] K. E. McKone, R. G. Schroeder, and K. O. Cua, “The impact of total productive maintenance practices on manufacturing performance,” J. Oper. Manag., vol. 19, no. 1, pp. 39–58, Jan. 2001.

[2] L. Swanson, “Linking maintenance strategies to performance,” Int. J. Prod. Econ., vol. 70, no. 3, pp. 237–244, Apr. 2001.

[3] S. Takata, F. Kirnura, F. J. A. M. van Houten, E. Westkamper, M. Shpitalni, D. Ceglarek, and J. Lee, “Maintenance: Changing Role in Life Cycle Management,” CIRP Ann. - Manuf. Technol., vol. 53, no. 2, pp. 643–655, Jan. 2004.

[4] K. A. Kaiser and N. Z. Gebraeel, “Predictive Maintenance Management Using Sensor-Based Degradation Models,” IEEE Trans. Syst. Man, Cybern. - Part A Syst. Humans, vol. 39, no. 4, pp. 840–849, Jul. 2009.

[5] Z. (Max) Yang, D. Djurdjanovic, and J. Ni, “Maintenance scheduling in manufacturing systems based on predicted machine degradation,” J. Intell. Manuf., vol. 19, no. 1, pp. 87–98, Jul. 2007.

[6] D. D. Sheu and J. Y. Kuo, “A model for preventive maintenance operations and forecasting,” J. Intell. Manuf., vol. 17, no. 4, pp. 441–451, Aug. 2006.

[7] F. Camci, “System Maintenance Scheduling With Prognostics Information Using Genetic Algorithm,” IEEE Trans. Reliab., vol. 58, no. 3, pp. 539–552, Sep. 2009.

[8] M. Marseguerra, E. Zio, and L. Podofillini, “Condition-based maintenance optimization by means of genetic algorithms and Monte Carlo simulation,” Reliab. Eng. Syst. Saf., vol. 77, no. 2, pp. 151–165, Aug. 2002.

[9] J. LEE, “Machine performance monitoring and proactive maintenance in computer-integrated manufacturing: review and perspective,” Int. J. Comput. Integr. Manuf., Apr. 2007.

[10] A. Muller, A. Crespo Marquez, and B. Iung, “On the concept of e-maintenance: Review and current research,” Reliab. Eng. Syst. Saf., vol. 93, no. 8, pp. 1165–1187, Aug. 2008.

[11] J. Lee, J. Ni, D. Djurdjanovic, H. Qiu, and H. Liao, “Intelligent prognostics tools and e-maintenance,” Comput. Ind., vol. 57, no. 6, pp. 476–489, Aug. 2006.

[12] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Elsevier Science, 2014, p. 552.

[13] U. B. Kjærulff and A. L. Madsen, Bayesian Networks and Influence Diagrams. New York, NY: Springer New York, 2008.

[14] D. Koller and A. Pfeffer, “Object-oriented Bayesian networks,” pp. 302–313, Aug. 1997.