# A Methodology for Developing Local Smart Diagnostic Models Using Expert Knowledge

Anders L. Madsen
Aalborg University, and
HUGIN EXPERT A/S
Gasværksvej 5
9000 Aalborg, Denmark
Email: alm@hugin.com

Nicolaj Søndberg–Jeppesen
HUGIN EXPERT A/S
Gasværksvej 5
9000 Aalborg, Denmark
Email: nsj@hugin.com

Niels Lohse
and Mohamed S. Sayed
Loughborough University
Leicestershire, UK, LE11 3TU.
Email: [N.Lohse|m.sayed]@lboro.ac.uk

***Abstract*—This paper describes an innovative modular component-based modelling approach for diagnostics and condition-monitoring of manufacturing equipment. The approach is based on the use of object-oriented Bayesian networks, which supports a natural decomposition of a large and complex system into a set of less complex components. The methodology consists of six steps supporting the development process: *Begin*, *Design*, *Implement*, *Test*, *Analyse*, and *Deploy*. The process is iterative and the steps should be repeated until a satisfactory model has been achieved. The paper describes the details of the methodology as well as illustrates the use of the component-based modelling approach on a linear axis used in manufacturing. This application demonstrates the power and flexibility of the approach for diagnostics and condition-monitoring and shows a significant potential of the approach for modular component-based modelling in manufacturing and other domains.**

## I. Introduction

The need for diagnostic and health monitoring capabilities in manufacturing systems is becoming increasingly important as manufacturing organisations continuously aim to reduce system downtime and unpredicted disturbances to production. This crucial need accompanied with the availability of increasing amounts of sensory data and decreasing costs of computation on the shop-floor level have opened new opportunities for component suppliers and system integrators to provide more competitive functionalities that go beyond traditional control and process monitoring capabilities. However, the modelling effort required for enabling such capabilities in complex systems such as manufacturing systems has proven to be a major barrier for industry adoption. Therefore, approaches for component-based modelling that enable component suppliers to construct diagnostic models on the component level that also lend themselves to later integration into emergent wider system-level models seem to be a more suitable approach. Such component-level diagnostic capabilities should first enable component-level embedded diagnostic reasoning on the device level, as well as enabling wider system integration into system-level diagnostic models as part of the system integration phase.

Various approaches have been reported in literature to enable diagnostic and health monitoring in manufacturing systems. Some of these include Discrete Event Systems (DES) in [1] and [2]. Despite their theoretical soundness, however, the construction of viable DES models for complex domains such as manufacturing systems has proven to be very difficult due to state explosion and the need for accurate detailed modelling which is not always feasible for manufacturing systems. On the other hand, data driven approaches with little modelling requirements have also been investigated. These include artificial neural networks [3], fuzzy logic [4] and Bayesian networks [5], [6] among others. The models here do not necessarily capture precise nominal system behaviour as in model-based approaches, but rather a general understanding of how the system works. Another important approach that has been extensively used for diagnosis in manufacturing systems is Stream of Variation (SoV) theory [7] which aims to integrate multivariate statistics with control theory and design knowledge to provide root cause diagnosis, mostly for dimensional quality variation problems in multi-stage automotive assembly systems [8] and multi-stage machining processes [7]. It is worth noting that SoV works under the assumption of process linearity, however in reality this ceases to exist as various assembly processes are in fact non-linear such as welding thermal deformation, compliant deformation and contact interaction [9]. These non-linearities render estimation-based approaches such as SoV mostly inaccurate. Moreover, other influencing factors that characterise many manufacturing systems such as measurement inaccuracies, measurement noise, and difficulty of obtaining complete data sets especially during the first phases of production, these factors make such manufacturing processes highly characterised by uncertainty [10]. These uncertainty criteria motivate the need for employing modelling and reasoning approaches that are capable of handling and reasoning under conditions of uncertainty.

This paper describes an innovative modular component-based modelling approach for diagnostics and condition-monitoring of manufacturing equipment. The approach is based on the use of Object-Oriented Bayesian networks (OOBNs), which supports a natural decomposition of a large and complex system into a set of less complex components.

## II. Preliminaries and Notation

### A. Bayesian Networks

A Bayesian network (BN) [11], [12], [13] is a probabilistic graphical model that simplify a probabilistic representation by exploiting the marginal and conditional independencies in the domain. Simply speaking, a BN is a pair $\langle \mathcal{G}, \mathcal{P} \rangle$, where $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ is an acyclic directed graph (DAG) over a set of

random variables $\mathbf{V}$ and a set of directed edges $\mathbf{E}$ that represent probabilistic relationships between variables $\mathbf{V}$. $\mathcal{P}$ is a set of conditional probability distributions (CPDs) that quantify the strength of the relations induced by $\mathbf{E}$. Specifically, $\mathcal{P}$ contains for each $V \in \mathbf{V}$, the CPDs $P(V|pa(V))$, where $pa(V)$ is the set of parent variables of $V$ in $\mathcal{G}$.

A BN supports diagnostics by computing the posterior probability $P(H|e)$ of an unobservable fault hypothesis $H$ given observed evidence $e = \{e_1, \ldots, e_m\}$, where each $e_j$ is the observed state of the variables $\{E_1, \ldots, E_m\} \subset \mathbf{V}$. The calculations are based on the chain rule [14]:

$$P(\mathbf{V}) = P(V_1, \ldots, V_n) = \prod_{V_i \in \mathbf{V}} P(V_i|pa(V_i)). \quad (1)$$

Figure 1 displays an example of a BN representing a simple electro mechanical component. Assume that a power supply ($PS$) supplies a motor ($M$) and a sensor ($S$) with electricity. The motor pulls some unit (e.g. a wagon on a track) while the sensor measures when it is time to release the unit and let it slip back to its origin. We are interested in whether this unit moves or not ($UM$). Finally, a thermometer ($T$) measures the temperature of the power supply. The model shows the relationship between the different parts. Assume for simplicity that all the variables are boolean. For a boolean variable $V$ we shall use $v$ and $\neg v$ to indicate error and no error, respectively.

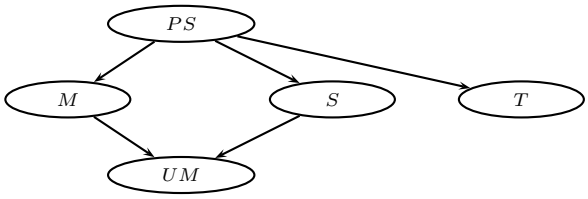


Fig. 1: An example of a Bayesian Network.

Notice how the directed edges between the nodes indicate causal relationships between the variables. The variables $T$, $M$ and $S$ are influenced by the state of $PS$ while $UM$ is influenced by both $M$ and $S$. For this network we need to specify the CPDs that quantify the strengths of the causal relationships. According to the definition given above the model must have the distributions: $P(PS)$, $P(S|PS)$, $P(T|PS)$, $P(UM|M,S)$ and $P(T|PS)$, and we have $P(\mathbf{V}) = P(PS)P(S|PS)P(T|PS)P(UM|M,S)P(T|PS)$.

The BN can be used for both diagnosis and prognosis. Diagnosis is when we wish to identify the most likely cause of some observed event while prognosis is the process of predicting the consequences of a particular error. It is an example of diagnosis, if we, for instance, know that the unit does not move and we have observed abnormal temperatures (i.e., $e = \{um, t\}$). We can calculate the probability that a power supply fallout is the cause as $P(ps|um, t)$. Software packages exists that can perform these calculations [15].

### B. Nested structures in Bayesian Networks

A BN model often contains parts that can logically be joined together into coherent groups because they are part of the same subcomponent of the domain and sometimes such groups are repeated multiple times in the model. OOBNs [16] and Multiply Sectioned Bayesian Networks (MSBNs) [17], [18] are different paradigms that add the ability to perform such grouping within a BN model. In MSBNs the entire model is divided into sections, which each have their own computational unit that acts autonomously. Probability calculations is then performed inside sections and via communication between sections. OOBNs, on the other hand, are BN representations that embed other BNs into the models in order to encapsulate details of a particular section of a domain. Such embedded models are referred to as instances. We want to point out that the methodology proposed in this paper is independent of the choice of modelling paradigm, but we have chosen to use OOBNs since they support multi-level hierarchies and they are more widely applied.

### C. Knowledge Elicitation

Knowledge elicitation is the process of establishing a BN model. Methods have been developed that guide or even automate this process. The process is often performed in two steps that can be iterated a number of times, where the first step is to establish the structure of the BN while the second step is to quantify the model with probability distributions [19].

Methods that can guide the establishment of a BN structure are, for instance, [20] who construct models in a bottom-up fashion by first identifying building blocks and then applying a set of combination rules while [21] have developed a framework that automatically generates a BN structure from a knowledge base expressing generic probabilistic relations. Furthermore, [22] proposes a top-down strategy that starts out with an overall abstract target problem which is then decomposed into a number of less abstract sub-problems until eventually the overall target problem has been decomposed into its possible root causes.

When the structure of a BN has been established, the challenge is to assessed the CPDs. If sufficient data is available the quantification of the model amounts to estimating the probabilities from the available data. Unfortunately however, in many cases sufficient data is not available and the quantification depends on assistance from domain experts. This can be a considerable task if a large number of probabilities have to be elicited [19]. Fortunately, techniques have been developed that can simplify this process, for instance the noisy-OR gate and its generalizations [23], [24], [11]. The number of probabilities to be assessed with such a technique is linear rather than exponential in the number of parents. But no matter how the quantification is established imprecise probabilities can deteriorate final model [19]. It is common to use sensitivity analysis [25], [26], [27] to analyse the possible impact of imprecise probabilities. Recent studies has been focused on establishing error bounds for the possible errors introduced by the application of noisy-OR gates and its generalizations [28].

## III. Methodology

We apply the model development cycle proposed by [13] illustrated in Figure 2 (taken from [13]) to the development of local smart diagnostic models in the domain of manufacturing. The main steps of this approach are: 1) Begin, 2) Design, 3) Implement, 4) Test, 5) Analyse, and 6) Deploy. The individual steps of the methodology are described in more detail in the

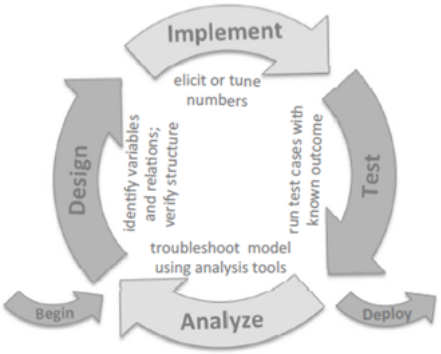


Fig. 2: Model development cycle [13].

following sections. These steps should be iterated as indicated in the figure in order to find the best workable model.

### A. Begin

The objective of the *Begin*-step is to prepare the development process by identifying and understanding the objective of the model development process. This also includes the task of determining the appropriate model type, i.e., static BN or dynamic Bayesian network [29], [30] as well as whether an object-oriented approach should be applied.

### B. Design

The *Design*-step is the first step in the iterative part of the process. The objective of this step is to identify the variables $\mathbf{V}$ and relations $\mathbf{E}$ of the DAG $\mathcal{G} = (\mathbf{V}, \mathbf{E})$. A useful approach to identify the variables of the model is to divide variables into *hypothesis*, *information* and *mediating* variables [13]. In the case of smart diagnostic models, hypothesis variables are the root causes to be considered while information variables are sensor readings, observations from a potential user of the equipment, et cetera. Mediating variables are non-hypothesis and non-information variables included in the model to obtain the correct dependence and independence properties as defined by $\mathcal{G}$. Since we consider diagnostic models, there will be at least one *problem* defining variable corresponding to the knowledge that the system has failed.

In the *Design*-step it is important to make sure that variables are defined sufficiently precise. If not, then a variable is said to fail the *clarity test* [31]. This will often produce problems at later steps in the model development process. It is often useful to define the subtype of a random variable in order to be able to put a semantic interpretation on its states in the *Implement*-step.

In an OOBN, $\mathbf{E}$ includes direct probabilistic dependence relations and binding links. This is referred to as the qualitative part of the model. In this process the network classes corresponding to subcomponents should be identified along with interface of each class, i.e., input and output variables used to link an instance node to its encapsulating class.

The *Design*-step includes verification of the qualitative knowledge using the notion of $d$-separation [11], [32]. The $d$-separation criterion is used to verify dependence and independence properties of $P(\mathbf{V})$ as specified in $\mathcal{G}$. The principle of *Occam's razor* should be applied in order to achieve as simple a model as possible with the desired properties.

### C. Implement

The objective of the *Implement*-step is to quantify the model. This means to specify the strength of the relations defined by $\mathcal{G}$ using the CPDs of $\mathcal{P}$. For each variable $v \in \mathbf{V}$, we specify the CPD $P(V|pa(V))$ of $V$ conditional on $pa(V) \subseteq \mathbf{V}$. When sufficient hard data to perform parameter estimation is not available, it is necessary to assess the probability values in a different way. There are a number of techniques that can be used to elicit probability values from domain experts [33] including direct, e.g., [34] and indirect methods, e.g., [35]. An common alternative to elicitation from experts, is to use literature, mathematical formulas to specify the relationship between a variable $V$ and its direct parents $pa(V)$ or methods as described above. This can be achieved by assuming certain classical distributions (e.g., Normal or Weibull) in the case of variables of continuous subtype, logical relations or classical distributions (e.g., binomial) in the case of variables of Boolean subtype. This decreases the elicitation burden of the *Implement*-step significantly.

### D. Test

The objective of the *Test*-step is to evaluate the performance of the model by either domain expert using the model or more formally by running a number of test cases with known outcomes against the model. In a diagnostic model, the objective of the *Test*-step is to evaluate the performance of the model against a set of hypothetical or real world scenarios in order to validate the behaviour of the model. This amounts to determining if the model is able to identify the most likely root causes. An important property of BNs and OOBNs, in particular, is the possibility of testing the model as it is being developed. This includes unit tests of OOBN classes and even testing the relationship defined by a CPD $P(V|pa(V))$.

The *Test*-step could include deployment of the model either on a web site or as a web service for experts to be able to test and evaluate the model at their convenience using a simplified user interface without having to use an advanced model development interface [36].

### E. Analyse

The objective of the *Analyse*-step is to determine the behaviour of the model from different perspectives. This include identifying and correcting undesired behaviours. Possible analysis tools to consider include parameter sensitivity analysis, e.g., [25], [26], [27], but also value of information analysis (VOI), e.g., [37], [38], conflict analysis [39], [40] and evidence sensitivity analysis, e.g., [41], [13].

### F. Deploy

The objective of the *Deploy*-step is to integrate the model into the component control software. For a diagnostic model the integration into control software involves linking the information variables of the model, e.g., sensor readings, to the corresponding data sources which could be a data base with performance data, values obtained directly from the sensor, or

processed sensor data. It also involves linking the root cause probabilities to, for instance, a human machine interface (HMI) such that the operator sees the $n$ most likely root causes after a diagnostic process is launched either automatically by the system when a failure is observed or by the operator. The *Deploy*-step can, in principle, be initiated once the information and root causes of the model have been settled.

As part of the *Deploy*-step, the documentation of the developed model and its integration should be finalized. It is important to document the elements of the model as it is being developed, i.e., documentation should be a running process and most model development tools allow the user to associate notes or attributes with specific elements of a model.

## IV. Linear Axis Model

This section describes how the proposed methodology has been applied to develop a component-level diagnostic model for a Linear Axis produced by IEF-Werner GmbH[1]. We do not describe all steps of the method, but focus on *Begin*, *Design*, *Implement* and *Test* while *Analyse* and *Deploy* are only considered briefly.

The Linear Axis as a self-sustainable handling system that is designed to be a high performance machine with a demand to work 24h / day seven days a week. Therefore, there is little or no time for maintenance and repair. This means that there is a need for system condition monitoring to prevent failures and for system failure diagnosis. We focus on the latter case, but the methodology has been applied for both the case of condition monitoring using Dynamic BNs [29], [30] and fault diagnosis. The diagnosis model is used under the assumption that a problem is observed and we need to identify the most likely root cause.

### A. Begin

The diagnostics model is developed as an OOBN as it is to be applied for self diagnostics at the component level as well as to be integrated into a larger system-wide model for diagnostics at a higher level of abstraction, i.e., shop-floor or even factory level. The model is developed using the HUGIN software package[2][15].

### B. Design

The OOBN in Figure 3 depicts the OOBN model that was the output of the *Design*-step. The OOBN reflects root causes, possible observations and whether a problem has been reported or not. Additionally, the model contains five instance nodes which represent OOBN models of nested components.

For this discussion it suffices to know about these five instance nodes that they in turn contain root cause variables, possible observations and whether or not a problem has been reported within that particular component. The instance nodes are labelled GuideCarriage, ToothedBelt, MotorSystem, LimitSwitchNeg and LimitSwitchPos and they each contain two, five, eight, seven and seven variables, respectively. The root cause variables are ServoAmplifier, DriveUnit, DeflectionUnit, ControlUnit, Solder and Gear while the possible sensor readings are noise $\eta$ (Boolean), motor current $I_M$ ($none$, $normal$, $high$), guide carriage motor temperature $T_{GCM}$, amplifier temperature $T_A$, negative limit switch sensor $S_-$ ($normal$ or $inactivated$) and positive limit switch sensor $S_+$ ($normal$ or $inactivated$). Some of these sensor readings are hidden by the instance nodes. The variable that indicates whether a problem has been reported is labelled Problem.

Notice how the directed edges between the nodes indicate causal influence. For instance the state of ServoAmplifier influences $T_A$ reflecting the fact that if there is a problem with the amplifier is likely to cause an elevated temperature level.

Likewise the amplifier has outgoing edges going into MotorSystem, LimitSwitchNeg and LimitSwitchPos indicating that neither of these components will work if the amplifier has failed. UnitDoesNotMove has a similar reasoning, in that, it will fail if any of its parents fail.

Notice the resemblance with the model in Figure 1 between the part of the model consisting of the ServoAmplifier, MotorSystem, LimitSwitchNeg and UnitDoesNotMove. The most noticeable difference is the use of instance nodes in Figure 3. Instead of just seeing a problem in, for instance, the motor sub system the use of instance nodes allow us to direct the diagnostics towards the exact part of the motor sub system that failed. This allows us to perform a much more fine grained analysis without packing the top level model with details.
The ability to perform this fine grained analysis is used in the link between MotorSystem and IncreasedNoise. The MotorSystem instance has two variables that are parents of IncreasedNoise, one variable indicating that the Motor Bearings are worn and one indicating a problem with the brakes. Thus, the observation $\eta$ becomes an indicator of how the MotorSystem has failed.

### C. Implement

In the *Implement*-step the model from Section IV-B was quantified with CPDs. First, the marginal probability distributions were set for the root cause variables. These probability distributions were set according to assessments by the experts and according to hardware data sheets. The probabilities should reflect the marginal probability for each component to fail. For instance, the DeflectionUnit is among the components with the highest marginal probability of failing with $P(DeflectionUnit) = 0.002$ whereas, for instance, ControlUnit is among the components with the lowest marginal probabilities of failing with $P(ControlUnit) = 0.0001$.

The CPDs on UnitDoesNotMove, IncreasedNoise, Noise and Problem were configured such that the probability of a failure is 1 if at least one of the parent variables fail, and 0 otherwise. The CPD associated with $T_A$ reflects the fact that a Problem in ServoAmplifier is likely to cause elevated temperature levels. The CPD representing $P(T_A|ServoAmplfier)$ is shown in Table I.

### D. Test

Once the initial model structure has been identified and verified in the *Design*-step and an initial quantification of the

[1]http://www.ief-werner.de
[2]http://www.hugin.com

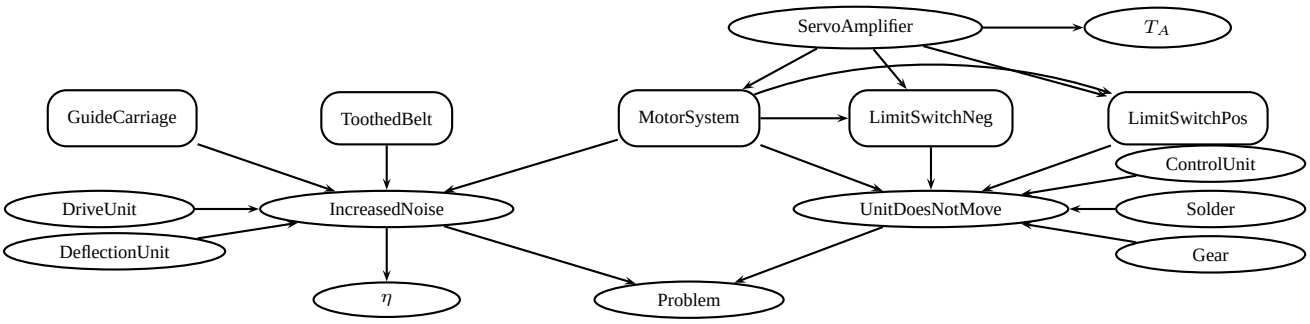


Fig. 3: The top level class of the Linear Axis Model.

TABLE I: The CPD associated with $T_A$.

| | 0-30 | 30-40 | 40-50 | 50-60 | 60-70 | 70-inf |
|---|---|---|---|---|---|---|
| OK | 0.2 | 0.3 | 0.3 | 0.2 | 0 | 0 |
| NO OK | 0.01 | 0.02 | 0.05 | 0.12 | 0.3 | 0.5 |

relations has been obtained, the performance of the model should be evaluated. The Linear Axis model was continuously tested during the model development process by domain experts interrogating the model with the help of knowledge engineers as part of model development workshops. This included evaluation of the model at individual OOBN class level with respect to both quantification and qualitative structure. In the *Deploy*-step, a web interface to the model was created to support the domain experts in the model test and evaluation.

As part of the evaluation a number of scenarios have been developed to evaluate the behaviour of the model on a set of predefined evidence sets with known root causes. Tables II and III show three different scenarios $e_1$, $e_2$, and $e_3$ that have been used to test and evaluate the model (we limit the presentation to three scenarios due to space restrictions).

TABLE II: The sensor values for three scenarios.

| $e$ | $\eta$ | $T_A$ | $T_{GCM}$ | $I_M$ | $S_-$ | $S_+$ |
|---|---|---|---|---|---|---|
| $e_1$ | $false$ | 28 | 55 | $none$ | $normal$ | $normal$ |
| $e_2$ | $false$ | 28 | 55 | $high$ | $normal$ | $normal$ |
| $e_3$ | $false$ | 28 | 55 | $normal$ | $normal$ | $inactivated$ |

The three evidence sets $e_1$, $e_2$, and $e_3$ define three different, but very similar scenarios as $e_1$ and $e_2$ differ only on the observed value of $I_M$ while $e_3$ differs from $e_1$ and $e_2$ on the value of $I_M$ and $S_+$. The three most likely root causes under each evidence set are shown in Table III where $MS$ is short for $MotorSystem$, $NSLE$ is $NominalServiceLifeExceeded$.

TABLE III: Most likely root causes in each scenario.

| Id. | $R_1$ | $P(R_1)$ | $R_2$ | $P(R_2)$ | $R_3$ | $P(R_3)$ |
|---|---|---|---|---|---|---|
| $e_1$ | $MS.Motor$ | 0.45 | $MS.Cable$ | 0.27 | $MS.Connection$ | 0.27 |
| $e_2$ | $MS.Motor$ | 1 | $GC.NSLE$ | 0.0005 | $GC.GuideRails$ | 0.0002 |
| $e_3$ | $Gear$ | 0.3 | $TB.TB$ | 0.15 | $GC.GuideRails$ | 0.15 |

If the domain experts do not agree with the results produced by the model, then the model should be tuned by adjusting the value of parameters. Here parameter sensitivity analysis plays an important role as it supports the identification of the most influential parameters on a probability $P(x|e)$ and supports solving constraints on probabilities.

### E. Analysis and Deploy

Different analysis tools were applied to analyse the performance of the model. This included parameter sensitivity analysis and VOI analysis. For instance, myopic hypothesis driven VOI analysis was used to identify the most informational next questions or observation requests to present to the operator on the HMI. That is, VOI is not only used for model performance analysis, it can also be used to guide the diagnosis process. Once the most likely root cause has been identified, VOI analysis is used to identify additional observations that have the most information on the root cause, e.g., under $e_1$ the most informational next observation with respect to $MS.Motor$ is to check if there is a problem with $MS.Electrics$. VOI may suggest the operator to collect additional information by visual inspection where the operator is guided on what to look for to confirm or not the proposed root cause.

The model was integrated into the control software using an Application Programming Interface. It can be used both as a predictive model for predicting failure (in the short term) or as a diagnostic model once the failure as been observed. A special-purpose web interface [36] for the Linear Axis local-component model was developed to support the *Test*-step of the development process [3]. This has served as an important tool in the *Test* and *Analysis*-steps as the component supplier have been able to interact with the model at her own convenience.

## V. Conclusions and Future Work

A methodology for building component-based Bayesian diagnostic and predictive models to enable the realisation of smart manufacturing devices is presented. The derived diagnostic models can both be used separately on the device level or as part of a wider system-level model. Crucially the localisation of the models on the device level enables better modelling quality from expert knowledge due to the fact that component suppliers are likely to have retained detailed knowledge about the behaviour of the device. This overcomes the problem of knowledge retention when building system-level models from experts, as system integrators might not

[3] http://selsus.hugin.com

always have access to detailed knowledge about the detailed behaviour of all the components within the system.

Although the modelling methodology is primarily expert-driven, future work will focus on enabling the derivation of the models from existing engineering design information such as CAD models and FMEA data [42].

## 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 IEF-Werner GmbH for allowing us to use the Linear Axis model.

## REFERENCES

[1] N. Viswanadham and T. Johnson, "Fault detection and diagnosis of automated manufacturing systems. in decision and control," in *Proceedings of the 27th IEEE Conference*, 1988.

[2] L. Holloway and S. Chand, "Time templates for discrete event fault monitoring in manufacturing systems," in *In Proceedings of the 1994 American Control Conference*, 1994, pp. 701–706.

[3] Y. Bae, S.-H. Lee, H.-C. Kim, B.-R. Lee, J. Jang, and J. Lee, "A real-time intelligent multiple fault diagnostic system," *The International Journal of Advanced Manufacturing Technology*, vol. 29, no. 5, pp. 590–597, 2006.

[4] S. Tzafestas, "Concerning automated assembly: knowledge-based issues and a fuzzy system for assembly under uncertainty," *Computer Integrated Manufacturing Systems*, vol. 10, no. 3, pp. 183–192, 1997.

[5] K. McNaught and A. Chan, "Bayesian networks in manufacturing," *Journal of Manufacturing Technology Management*, no. 6, pp. 734–747, 2011.

[6] M. S. Sayed and N. Lohse, "Distributed bayesian diagnosis for modular assembly systemsa case study," *Journal of Manufacturing Systems*, vol. 32, no. 3, pp. 480 – 488, 2013.

[7] J. Shi, *Stream of variation modeling and analysis for multistage manufacturing processes*. CRC Press, 2006.

[8] S. Hu and Y. Koren, "Stream-of-variation theory for automotive body assembly," *CIRP Annals-Manufacturing Technology*, vol. 46, no. 1, pp. 1–6, 1997.

[9] K. Xie, L. Wells, J. Camelio, and B. Youn, "Variation propagation analysis on compliant assemblies considering contact interaction," *Journal of Manufacturing Science and Engineering,*, vol. 129, p. 934, 2007.

[10] S. Jin, Y. Liu, and Z. Lin, "A bayesian network approach for fixture fault diagnosis in launch of the assembly process," *International Journal of Production Research*, vol. 50, no. 23, 2012.

[11] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, ser. Series in Representation and Reasoning. San Mateo, CA: Morgan Kaufmann Publishers, 1988.

[12] R. Cowell, A. Dawid, S. Lauritzen, and D. Spiegelhalter, *Probabilistic Networks and Expert Systems*. Springer, 1999.

[13] U. Kjærulff and A. Madsen, *Bayesian Networks and Influence Diagrams: A Guide to Construction and Analysis*, 2nd ed. Springer, 2013.

[14] F. Jensen and T. Nielsen, *Bayesian Networks and Decision Graphs*, 2nd ed. Springer, 2007.

[15] A. L. Madsen, F. Jensen, U. Kjærulff, and M. Lang, "The HUGIN Tool for Probabilistic Graphical Models," *International Journal of Artificial Intelligence Tools*, vol. 14, no. 3, pp. 507–543, 2005.

[16] D. Koller and A. Pfeffer, "Object-oriented Bayesian networks," in *Proc. of UAI*, 1997, pp. 302–313.

[17] Y. Xiang, D. Poole, and M. P. Beddoes, "Multiply sectioned bayesian networks and junction forests for large knowledge based systems," *Computational Intelligence*, vol. 9, pp. 171–220, 1993.

[18] Y. Xiang, *Probabilistic Reasoning in Multiagent System - A graphical models approach*. Cambridge University Press, 2002.

[19] M. Druzdzel and L. van der Gaag, "Building probabilistic networks: "where do the numbers come from?" guest editors' introduction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, no. 4, pp. 481–486, 2000.

[20] M. Neil, N. Fenton, and L. M. Nielsen, "Building large-scale Bayesian networks," *The Knowledge Engineering Review*, vol. 15, no. 3, pp. 257–284, 2000.

[21] K. B. Laskey and S. M. Mahoney, "Network fragments: Representing knowledge for constructing probabilistic models," in *Proc. of UAI*, 1997, pp. 334–341.

[22] U. Kjærulff and A. Madsen, "A methodology for acquiring qualitative knowledge for probabilistic graphical models," in *Proc. of IPMU*, 2004, pp. 143–150.

[23] F. J. Diez, "Parameter adjustment in Bayes networks, the generalized noisy OR-gate," in *Proc. of UAI*, 1993, pp. 99–105.

[24] M. Henrion, "Some practical issues in constructing belief networks," in *Proc. of UAI*, 1989, pp. 161–173.

[25] K. B. Laskey, "Sensitivity analysis for probability assessments in Bayesian networks," in *Proc. of UAI*, 1993, pp. 136–142.

[26] E. Castillo, J. M. Gutiérrez, and A. S. Hadi, "Sensitivity analysis in discrete Bayesian Networks," *IEEE Transactions on Systems, Man and Cybernetics, Part A,*, pp. 412–423, 1997.

[27] V. M. Coupé and L. C. van der Gaag, "Practicable sensitivity analysis of Bayesian belief networks," in *Proc. of Prague Stochastics '98*, 1998, pp. 81–86.

[28] P. Woudenberg and L. van der Gaag, "Using the noisy-or model can be harmful ... but it often is not," in *Proc. of ECSQARU*, J.Lemmer, T. Levitt, and L. Kanal, Eds., 2011.

[29] T. Dean and K. Kanazawa, "A model for reasoning about persistence and causation," *Computational Intelligence*, vol. 5, pp. 142–150, 1989.

[30] U. B. Kjærulff, "dHugin: a computational system for dynamic time-sliced Bayesian networks," *International Journal of Forecasting*, vol. 11, pp. 89–111, 1995.

[31] R. A. Howard and J. E. Matheson, "Influence diagrams," in *Principles and Application of Decision Analysis*. Strategic Decisions Group, Menlo Park, California, 1984.

[32] T. Verma and J. Pearl, "An algorithm for deciding if a set of observed independencies has a causal explanation," in *Proc. of UAI*, 1992, pp. 323–330.

[33] L. C. van der Gaag, S. Renooij, C. Witteman, B. M. P. Aleman, and B. G. Taal, "How to elicit many probabilities," *CoRR*, vol. abs/1301.6745, 2013.

[34] L. van der Gaag, S. Renooij, C. Witteveen, B. Aleman, and B. Taal, "Probabilities for a probabilistic network: a case study in oesophageal cancer," *Artificial Intelligence in Medicine*, vol. 25, no. 2, pp. 123–148, 2002.

[35] N. Fenton, M. Neil, and J. Caballero, "Using ranked nodes to model qualitative judgments in bayesian networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 10, pp. 1420–1432, 2007.

[36] A. Madsen, M. Karlsen, G. Barker, A. Garcia, J. Hoorfar, F. Jensen, and H. Vigre, "A Software Package for Web Deployment of Probabilistic Graphical Models," in *Proc. of SCAI*, 2013, pp. 175–184.

[37] H. Raiffa, *Decision Analysis*. Addison-Wesley, Reading, MA, 1968.

[38] S. L. Dittmer and F. V. Jensen, "Myopic value of information in influence diagrams," in *Proc. of UAI*, D. Geiger and P. Shenoy, Eds. San Francisco, California: Morgan Kaufmann Publishers, 1997.

[39] K. B. Laskey, "Conflict or surprise: Heuristics for model revision," in *Proc. of UAI*, 1991, pp. 197–204.

[40] F. V. Jensen, B. Chamberlain, T. Nordahl, and F. Jensen, "Analysis in HUGIN of data conflict," in *Proc. of UAI*, 1991, pp. 519–528.

[41] F. V. Jensen, *An Introduction to Bayesian Networks*. UCL Press, London, 1996.

[42] M. Sayed and N. Lohse, "Ontology-driven generation of bayesian diagnostic models for assembly systems," *The International Journal of Advanced Manufacturing Technology*, vol. 74, no. 5-8, pp. 1033–1052, 2014.