

# Learning and Diagnosis in Manufacturing Processes Through an Executable Bayesian Network

M.A. Rodrigues<sup>1</sup>, Y. Liu<sup>2</sup>, L. Bottaci<sup>2</sup>, D.I. Rigas<sup>2</sup>

<sup>1</sup>School of Computing & Management, Sheffield Hallam University, Sheffield, S1 1WB, UK

<sup>2</sup>Department of Computer Science, The University of Hull, Hull, HU6 7RX, UK

**Abstract.** In this paper we present a novel approach to modelling a manufacturing process that allows one to learn about causal mechanisms of manufacturing defects through a Process Modelling and Executable Bayesian Network (PMEBN). The method combines probabilistic reasoning with time dependent parameters which are of crucial interest to quality control in manufacturing environments. We demonstrate the concept through a case study of a caravan manufacturing line using inspection data.

## 1 Introduction

This paper describes a method for on-line monitoring and diagnosis of manufacturing defects in a labour intensive manufacturing environment. A case-study of a caravan production line is used where management requires on-line information such as frequency of defects, causal relationships, defect targets, work force skill levels, and so on. Our approach to the problem is based on Bayesian analysis. In the last two decades, Bayesian inference and Bayesian networks have been extensively used to simulate and learn about causal mechanisms that operate in the environment [1], [2], [3], [4], [5], [9], [16], [17]. Bayesian networks are graphical models based on probability theory used to gain insights into system behaviour, or to forecast a system response to specific actions. Early Bayesian networks used message-passing and were limited to singly connected networks or trees [6], [7]. Refinements to the tree-propagation method have been proposed such as node elimination [14], clique-tree propagation and loop-cut conditioning [8], [15]. Our novel approach combines systems modelling, simulation, [10], [11], [12], [13] and Bayesian inference together with graphical executable models into a single framework which, at the same time that it is tuned to manufacturing environments, it is also sufficiently generic to be applied to automated or non-automated production lines. Section 2 describes how manufacturing processes are represented and modelled within a Bayesian framework, Section 3 highlights experimental results and finally in Section 4 some conclusions are drawn.

## 2 Representation and Modelling of Manufacturing Processes

We have devised a common representation framework that allows us to reason about aspects of manufacturing data and also to reason about the manufacturing process itself as described in this section. All data and processes are hierarchically represented as

a set of data defined as: [ Unit Operation Component Location Fault Team ]. The assumptions built into such representation are highlighted as follows. Suppose that a given production batch (Unit) is divided into a number of operations (Operation). We wish to predict the probabilities of faults for each operation given our initial understanding of the manufacturing process which should then be updated as new sample information is available. In order to do so, we denote the operation variables by  $\Theta_i$  ( $i = 1, 2, \dots, n$  where  $n$  is the number of operations), and  $\theta_i$  as the state of the variable  $\Theta_i$ , and  $R$  is a set of observations. The convention used here is that a variable is denoted by an upper-case letter while the state or value of the variable by the same letter in lower case. We use Bayes' rule to obtain the probability distribution for  $\Theta_i$  given  $R$  and background knowledge  $\beta$ :

$$P(\theta_i|R, \beta) = \frac{P(\theta_i|\beta) P(R|\theta_i, \beta)}{P(R|\beta)} \quad (1)$$

where

$$P(R|\beta) = \int_{i=1}^n P(R|\theta_i, \beta) P(\theta_i|\beta) d\theta_i \quad (2)$$

The term  $P(R|\theta_i, \beta)$  is the likelihood function while  $P(\theta_i|\beta)$  is the marginal probability for a given operation  $\Theta_i$ . Both terms require estimation. The marginal probabilities are the prior probabilities  $P(\theta_i|\beta)$  and can be estimated through our belief or knowledge of the process. Posterior probabilities are then used to update our knowledge of the process and thus, our prior assessment. The likelihood function is more complex and can be estimated from observation of physical probability distributions. For instance, assuming a random set of variable  $X$ ,

$$P(x|\theta, \beta) = f(x, \theta)$$

where  $f(x, \theta)$  is the likelihood function with parameters  $\theta$ , some knowledge or assumptions are required to solve this problem. A solution can be found by assuming a finite number of parameters, then the variable under interest  $X$  whose  $x$  states maybe be continuous and have a Gaussian physical probability distribution with mean  $\mu$  and variance  $\nu$ :

$$P(x|\theta, \beta) = (2\pi\nu)^{-1/2} e^{-(x-\mu)^2/2\nu}$$

where  $\theta = \{\mu, \nu\}$ . However, for the case of manufacturing defects it is not always possible to make assumptions about the mean and variance of samples. It is more convenient to assume that events happen over a continuum, such as time, and are described by a *Poisson* distribution. In a Poisson distribution, the observed variable  $X$  has a number of occurrences  $r$  within a time interval  $t$  and  $\lambda$  is defined as the intensity of the process. Thus, we can express the likelihood as:

$$P(x|\theta_i, \beta) = P(\tilde{r}|t, \lambda) = \frac{e^{-\lambda t} (\lambda t)^r}{r!}$$

where  $\tilde{r}$  is the observed value of occurrences, and the mean and variance of the distribution are assumed as:

$$E(\tilde{r}|\lambda, t) = \lambda t, \quad V(\tilde{r}|\lambda, t) = \lambda t.$$

Thus, for such a time-dependent system the Bayes rule is expressed as:

$$P(\lambda_i|x, t, \beta) = \frac{P(x|\lambda_i, t, \beta) P(\lambda_i|t, \beta)}{\sum_i P(x|\lambda_i, t, \beta) P(\lambda_i|t, \beta)} \quad (3)$$

where independency of events and a constant (stationary) intensity  $\lambda$  are assumed. For the `Operation` example, we thus assume that each operation is independent and has its own local probability distribution. Obviously, the joint probability distribution is dependent on each constituent operation but it also implies that each operation is independent even when joint probabilities are known.

The above form the basis for the Bayesian analysis of manufacturing processes through the representation framework [`Unit Operation Component Location Fault Team`] which has been implemented by a *Process Modelling and Executable Bayesian Network* (PMEBN) as follows. For a set of variables  $\Theta = \{\theta_1, \dots, \theta_n\}$ , a Bayesian network structure  $N$  encodes independent assertions about the variables in  $\Theta$  and a set of local probability distributions associated with each variable. Together, those components define the joint probability distributions for  $\Theta$ . Using the example for `Unit` and `Operation` as in the previous section, if we use  $\theta_i$  to represent both the variable and its corresponding node for an operation, and  $\Pi_i$  to denote the parent unit of node  $\theta_i$  in  $N$ , the joint probability distribution for  $\Theta$  is given by

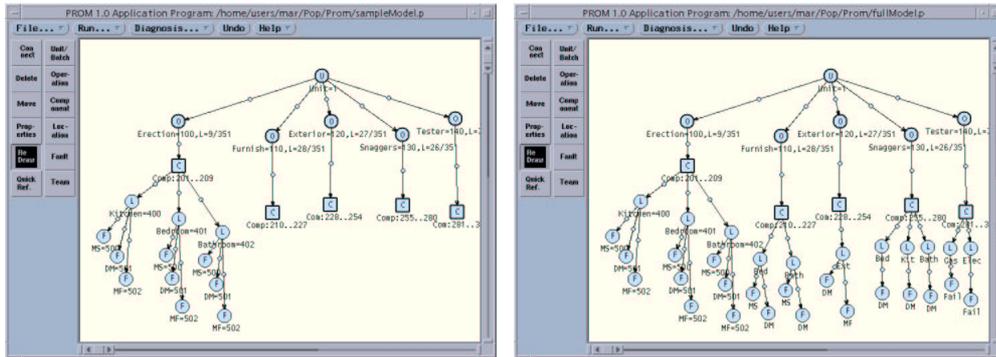
$$P(\Theta) = \prod_{i=1}^n P(\theta_i|\pi_i) \quad (4)$$

Given the semantics of causal relationships between the variables that can be readily asserted by production managers, these relationships normally correspond to assertions of conditional dependence. Thus, in order to build a Bayesian network, one would simply connect cause variables to their immediate effects. The next step would be to assess local probability distributions for the parent node  $P(\theta_i|\pi_i)$ , and this process is repeated for `Location` and `Team`.

The implemented PMEBN model, which doubles as a modelling and programming environment, is easily understood within the context of quality inspection. The system building blocks (`Unit`, `Operation`, `Component`, `Location`, `Fault`, `Team`) are selected from a menu buttons and dragged into the modelling window and connected together according to their hierarchical definition. The hierarchy is strict, but it can be loosened in the sense that missing layers of building blocks are allowed. As soon as any building block is created, the system automatically declares predefined functions whose parameters are then provided through dialogue boxes. Once the model (interconnected picture) is built, it can be run in two distinct modes: (1) in Bayesian inference mode, in which posterior probabilities are evaluated providing insights into the process parameters under study and also providing facts on various parameters; (2) in diagnosis mode, in which data analysis is performed and probabilities for given faults are estimated based on past history and new sample information. An example of a realistic model developed with our industrial partner is described in the next section.

### 3 Experimental Results

We would like to stress that, in model building with our proposed method, only a number of simple steps and decisions are required: *(i.)* divide a manufacturing process into one or more operations (this obviously need not correspond to actual physical processes if a simulation of what-if scenario is desired); *(ii.)* assign assembly components for each operation; *(iii.)* assign optional locations for each component; *(iv.)* select fault codes for which statistical analyses are required; *(v.)* assign component codes to teams for which statistical analyses are required.

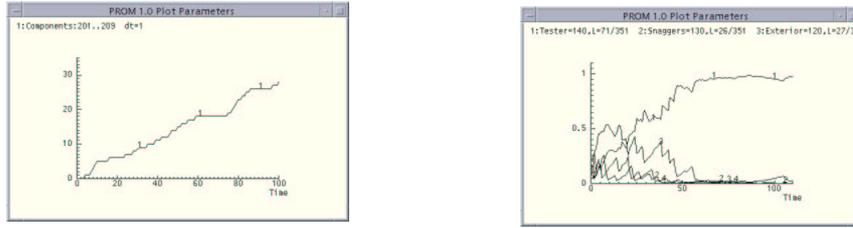


**Fig. 1.** Left, an incomplete model; right, a full model.

We have modelled the manufacturing process of a particular model of a caravan as defined by the production manager for one of the production lines. The model is depicted in Figure 1 above. The model and the modelling process highlights important aspects of our method as follows. First, batches can be modelled independently and saved together with their respective inspection data. Second, a number of arbitrary operations can be defined for each process under investigation. In the same way, any arbitrary combination of components, locations, and faults can be investigated. Third, if it is desired to focus on only a number of operations at a time, the model can be incomplete, as depicted in Figure 1 left. This does not cause any run-time errors and the data are fully analysed.

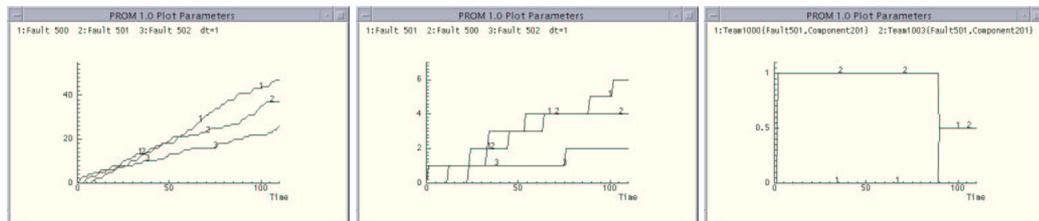
For experimental validation of the system, we have used a sub-set of data from real inspection sheets so that the problem is constrained to a manageable complexity. While the full set of component codes were used, the fault codes were categorised into 3 groups only. Similarly, teams were limited to 5 to match the number of operations (in reality, there are 17 teams in the production line). Similar simplification was also made for location, which only assumes 3 different values. Even with such simplifications, the number of possible combinations for operation/component/location/fault is very large.

The plots depicted in Figure 2 are outputs of the model running in Bayesian inference mode. Since each caravan takes 15 minutes to be manufactured, delta time is thus set to  $dt = 15\text{min}$ . This means that each unit of time represents data for 4 caravans and that one full day of production equates to 32 units of time. On the left of Figure 2 above,



**Fig. 2.** Running the model in Bayesian inference mode: left, simple facts such as number of faults per component. Right, posterior probabilities for assumed rate of faults per operation.

simple statistics can be shown such as number of faults per component. On the right, a number of operations have been defined for the manufacturing process. Given background knowledge from the production and quality managers, expected rate of defects ( $\lambda$ ) and corresponding probabilities were defined for each operation. Such rates can be seen as manufacturing targets for a particular operation whose probabilities are updated with new sample information. The curves show that only one of the given operations has a high probability of staying within its target for manufacturing defects and this represents vital information previously unavailable to the quality manager. In the same way, the model allows reasoning about levels of skills, as these can be defined in terms of desired rate of manufacturing defects with prior probabilities assessed by the production manager. The desired rate of defects may represent different levels of skills such as experienced, trained, and trainee. As new sample information is available, probabilities are updated and this may indicate to the quality manager the need for training or for redesign a process. Because the model is defined as time based, its outputs also give a clear indication of the time of day when performance was at its peak or otherwise. In this way, patterns and trends can be learned and actions taken accordingly.



**Fig. 3.** Performing diagnosis: left, major faults. centre: unattributed faults; right: probabilities for teams responsible for unattributed faults.

Figure 3 depicts results in diagnosis mode. On the left, faults with high frequency are displayed. Middle, unattributed or unknown faults. These faults are thus subject to causal analysis based on past history. On the right, probabilities for teams as evaluated by the system for unattributed faults. In addition to diagnosis, the system also performs interdependency analysis which is to determine which teams are introducing faults on components that have been assembled by other teams. This is only possible due to the adopted representation framework which allows the separate definition of components

that are part of an operation and components that fall into a team's responsibility which are not necessarily the same.

## 4 Summary and Conclusion

The advantages of the method implemented by the Process Modelling and Executable Bayesian Network (PMEBN) are summarised as follows. (1.) Defects often display a time dependency which quality managers are keen to capture and understand. This is built into the model in a very intuitive way through intensity of process, Bayesian probabilities, and dynamical modelling; (2.) although joint probability distributions are obviously dependent on various manufacturing parameters, these parameters are explicitly represented as independent and remain so. Quality managers can then learn their relative influence so that proper actions can be taken; (3.) the causal information encoded into the PMEBN method helps the analysis of sequence information and their interdependencies, such as the effects of sequence of assembly operations on the attribution of defects to teams; (4.) the hierarchically based representation framework for manufacturing parameters as Unit Operation Component Location Fault Team proved to be a powerful tool for inference and diagnosis, even for incomplete models; (5.) the graphical model is executable, which means that when the picture model is completed, so is the programming. This makes it particularly attractive to quality managers with no programming or advanced statistical knowledge, who can learn about the process and build what-if scenarios literally in a few minutes; (6.) similarly, the mapping between the topology of a PMEBN model and the manufacturing environment allows easy reconfiguration of the network in response to changing conditions resulting in process learning by the manager and automatic diagnostic reasoning on novel situations by the network.

## Acknowledgments

The work described in this paper is funded by the UK Engineering and Physical Sciences Research Council through grant EPSRC GR/L19508.

## References

1. J. Bernardo and A. Smith (1994). *Bayesian Theory*, John Wiley and Sons, New York.
2. W. Buntine (1996). A Guide to the literature on learning graphical models. *IEEE Transactions on Knowledge and Data Engineering*, 8:195-210.
3. D. Chickering and D. Heckerman (1996). Efficient approximation for the marginal likelihood of incomplete data given a Bayesian network. *Technical Report MSR-TR-96-08*, Microsoft Research, Redmond, WA.
4. D. Heckerman (1996). A Tutorial on Learning with Bayesian Networks. *Technical Report MSR-TR-95-06*, Microsoft Research, Redmond, WA.
5. D. Heckerman and R Schachter (1995). Decision-Theoretic Foundations for Causal Reasoning, *Journal of Artificial Intelligence Research*, 3:405-430.

6. J.H. Kim and J. Pearl (1983). A computational model for combined causal and diagnostic reasoning in inference systems. *3rd Int Conf KR&R*, CA, 661-672.
7. J. Pearl (1982). Reverend Bayes on inference engines: a distributed hierarchical approach. *AAAI Nat Conf on AI*, PA, 133-136.
8. J. Pearl (1988). *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann, San Mateo, CA.
9. J. Pearl (1994). Bayesian Networks, *UCLA Technical Report R-216*.
10. M.A. Rodrigues and Y.F. Li (1996). Dynamic Systems Modeling and Development of Robotic Applications in a Visual Programming Environment. *Int J Robotics and Computer Integrated Manufacturing*, 12(4):311-320.
11. MA Rodrigues and L Bottaci (1998). Modelling and Simulation of Customer-Manufacturer Interactions in a Labour Intensive Manufacturing Environment. *17th Int Conf Modelling, Identification and Control*, Grindelwald, 50-53.
12. MA Rodrigues and L Bottaci (1998). An AI Modelling Approach to Understanding a Complex Manufacturing Process. *Lecture Notes in AI 1415: Methodology and Tools in Knowledge-Based Systems*, Springer-Verlag, 814-821.
13. MA Rodrigues and L Bottaci (1998). Use of Simulation to Evaluate the Benefits of Timely Defect Information. *IEE Int Conf on Simulation: Innovation Through Simulation*, York, 235-239.
14. R.D. Shachter (ed.) (1990). Special Issue on Influence Diagrams. *Networks: An International Journal*, 20(5).
15. G. Shaffer and J. Pearl (ed.) (1990). *Readings in uncertain reasoning*, Morgan Kaufmann, San Mateo, CA.
16. M. West and J. Harrison (1989). *Bayesian Forecasting and Dynamic Models*, Springer-Verlag, New York.
17. R.L. Winkler (1972). *Introduction to Bayesian Inference and Decision*, Holt, Rinehart and Winston, Inc.