Reasoning With Different Time Granularities in Industrial Applications: A Case Study Using CP-logic

Johan Kwisthout and Peter J.F. Lucas

Radboud University Nijmegen, Institute for Computing and Information Sciences,
P.O. Box 9010, 6500GL Nijmegen, The Netherlands
$\{$*johank,peterl*$\}$*@cs.ru.nl*

Abstract

Bayesian networks more and more become the framework of choice for modeling uncertainty in industrial applications, in particular dynamic variants that model events over time. These networks, however, cannot deal well with systems in which events take place on different time granularities, e.g., when the time scale of such processes varies from seconds to months. We investigate the possible use of probabilistic logics, in particular CP-logic, to reason with uncertainty under different time scales.

1 Introduction

In many industrial applications an increasing emphasis is put on models of the system or artifact that is to be constructed, e.g., for simulation purposes, in model predictive control, or to generate what-if scenarios. Given the often overwhelming complexity of the system under consideration, these models should be able to deal with uncertainty, both stemming from the environment and the exact dynamics of the physical processes that take place within the system. From an architectural point of view it is desirable to have models that are adaptable and maintainable: it should not be necessary to construct an entire new model for every small change in the system. Also it is desirable to avoid redundancy: it is in general not preferable to have multiple overlapping models, each modeling a small subpart of the system's behavior or a slightly different variant of the system. This is in particular relevant when modeling processes that are related, but occur on different timescales.

In this paper we investigate the possible use of probabilistic logic to construct dynamical models that are capable of reasoning with uncertainty, are maintainable and adaptable, and are able to reason with different time granularities. Our case study is the design of heavy-duty industrial printers. In these printers, many uncertain factors play a role: the ambient temperature and humidity, the exact chemical characteristics of the paper and the toner, the efficiency and precision of the components that are used, and so on. Some processes take seconds - like transferring heat from a heating lamp to a rubber belt - while other are measured in months or years, like the effect of dust accumulation or deterioration of components.

In the last decades, Bayesian networks have become one of the dominant means to model systems that need to deal with uncertainty, in domains as diverse as medical applications [11, 17, 19], traffic [21], environmental [7] or meteorological [10] predictions, but also increasingly in industrial applications [14, 5, 8]. Bayesian networks intuitively model the (in-)dependencies between stochastic variables, yet are limited in their expressive power as they are based on propositional variables and the algebras spanning them. In problem domains where we want to reason with predicates or quantifiers (as in first-order logic), these constraints limit the usage of Bayesian networks.

Probabilistic languages can be seen as a natural generalization of both predicate logic and Bayesian networks. In the context of this paper we work with CP-logic [18] as our language of choice. Using a case study that stems from the use and modeling of large-scale printers, we show how a probabilistic language (as CP-logic) can be used in industrial applications in situations where the expressive power of Bayesian networks is too limited. In particular, we focus on situations where we want to reason about processes that have vastly different time granularities such that the use of Dynamic or Temporal Bayesian networks is impractical.

In the remainder of this paper, we will introduce notation and give an introduction on Bayesian networks and CP-logic in Section 2 and discuss alternative approaches to reasoning with different time granularities in Section 3. We introduce our case study of the use of probabilistic logic in heavy-duty printers in Section 4. We conclude the paper in Section 5.

2 Preliminaries

In this section we briefly introduce Bayesian networks and CP-logic as one of the possible generalizations of the concepts used in Bayesian networks.

2.1 Bayesian networks

A Bayesian network [13, 9] $\mathcal{B}$ is a graphical structure that models a set of stochastic variables, the (in-) dependencies among these variables, and a joint probability distribution over these variables. $\mathcal{B}$ includes a directed acyclic graph $\mathcal{G} = (\mathbf{V}, \mathbf{A})$, modeling the variables and (in-) dependencies in the network, and a set of parameter probabilities Γ in the form of conditional probability tables (CPTs), capturing the strengths of the relationships between the variables. The network models a joint probability distribution $\Pr(\mathbf{V}) = \prod_{i=1}^{n} \Pr(v_i \mid \pi(V_i))$ over its variables, where $\pi(V)$ denotes the parents of V in $\mathcal{G}$.

We will use upper case letters to denote individual nodes in the network, upper case bold letters to denote sets of nodes, lower case letters to denote value assignments to nodes, and lower case bold letters to denote joint value assignments to sets of nodes. We will use $\mathbf{E}$ to denote a set of evidence nodes, i.e., a set of nodes for which a particular joint value assignment is observed, and $\mathbf{e}$ to denote a particular observation. Every (posterior) probability of interest in Bayesian networks can be computed using well known lemmas in probability theory, like Bayes' theorem ($\Pr(H \mid E) = \frac{\Pr(E \mid H)\Pr(H)}{\Pr(E)}$), marginalization ($\Pr(H) = \sum_{g_i} \Pr(H \wedge G = g_i)$), and the property that $\Pr(\mathbf{V}) = \prod_{i=1}^{n} \Pr(v_i \mid \pi(V_i))$.

2.2 CP-logic

CP-logic [18] relates causal probabilistic events with logic programming, and can be seen as a generalization of Bayesian networks in order to allow them to reason with predicates, or, alternatively, as an extension to predicate logic in order to allow for stochastic rules and facts. Its building block is a so-called *CP-event*: a statement of the form $(p_1 : \alpha_1) \vee \ldots \vee (p_n : \alpha_n) \leftarrow \phi$, where p_i are ground atoms, α_i are non-zero probabilities such that $\sum \alpha_i \leq 1$, and ϕ is a first-order sentence in predicate logic. Intuitively, ϕ causes an event whose effect that at most one of the properties p_i becomes true, with probability α_i. If ϕ has a deterministic (rather than stochastic) effect, we simply write $p \leftarrow \phi$ as in traditional predicate logic.

The following example (taken from [18]) describes that pneumonia and angina may cause each other (with varying probabilities) and a bacterial infection can cause either pneumonia and angina.

Example 2.1.

$$(\mathrm{Angina} : 0.2) \leftarrow \mathrm{Pneumonia}. \tag{1}$$

$$(\mathrm{Pneumonia} : 0.3) \leftarrow \mathrm{Angina}. \tag{2}$$

$$(\mathrm{Pneumonia} : 0.4) \vee (\mathrm{Angina} : 0.1) \leftarrow \mathrm{Infection}. \tag{3}$$

Universal and existential quantifiers can be used to represent sets of events for all ground terms in the vocabulary. For example, when the constants $\{\mathrm{John}, \mathrm{Mary}\}$ are part of the vocabulary, the *non-ground* rule

$$\forall x (\mathrm{Angina}(x) : 0.2) \leftarrow \mathrm{Pneumonia}(x). \tag{4}$$

abbreviates the two CP-events

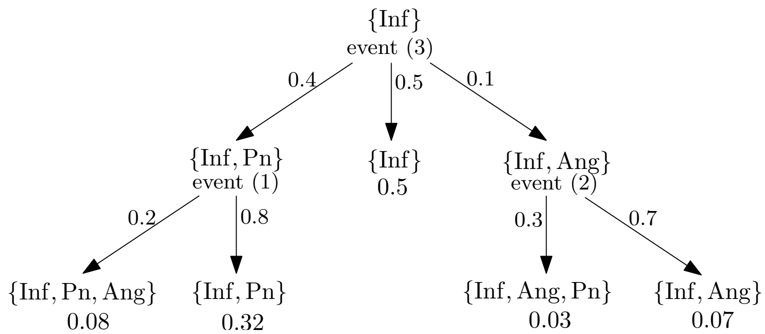

Figure 1: A process tree $\mathcal{T}$ corresponding to the events in Example 2.1 when an infection is encountered. The infection may or may not cause pneumonia or angina; pneumonia may cause angina and vice versa. The numbered events correspond to the events in Example 2.1

$$(\mathrm{Angina(John)} : 0.2) \quad \leftarrow \quad \mathrm{Pneumonia(John)}. \tag{5}$$
$$(\mathrm{Angina(Mary)} : 0.2) \quad \leftarrow \quad \mathrm{Pneumonia(Mary)}. \tag{6}$$

As a semantics, CP-logic uses tree structures whose edges are labeled with probabilities; each node in the tree corresponds to a state in the domain, i.e., each node V is mapped to a Herband interpretation $\mathcal{I}(V)$ using some interpretation function $\mathcal{I}$. The combination of such a tree and the corresponding interpretation (called a *probabilistic* Σ*-process* $\mathcal{T}$) now defines a probability distribution π over the final states that can be inferred. As an example we construct in Figure 1 a process $\mathcal{T}$ for the events in Example 2.1. Note that a problem may have different process trees defining the same probability distribution, for example if there are multiple events than can happen at a particular time; different orders of the events occurring will lead to different trees. Using this process tree it can be computed that the distribution π equals $(\{\mathrm{Inf}, \mathrm{Pn}, \mathrm{Ang}\} = 0.11, \{\mathrm{Inf}, \mathrm{Pn}\} = 0.32, \{\mathrm{Inf}, \mathrm{Ang}\} = 0.07, \{\mathrm{Inf}\} = 0.5)$.

3 Alternative Approaches

Combining reasoning with uncertainty with temporal relations has been investigated from different angles. Here we review some alternative approaches to the problem sketched in the introduction, in particular models of dynamic and/or temporal reasoning in statistics in general and Bayesian networks in particular.

Event history analysis[1, 20] is a technique often used to model the effects of significant events on the subject or topic of study. In psychology (when studying a human subject), events may be e.g. marriage, becoming a parent, or job entry and exit. In management sciences (when studying a company), events may be the appointment of a new CEO, a reorganization, or the merging of two companies. Event history analysis offers a statistical means of computing probability distributions of some event happening over time; however, event history analysis is focused on particular events and is less suitable when there are gradual changes and different time granularities as in our heavy-duty printer case study.

Temporal and Dynamic Bayesian networks [4, 3, 6, 2] use, like event history analysis, survival functions to model the probability that Q is in state q at time $t+1$, given that it was in that state at time t; in addition, exogenous and endogenous dynamic dependences can be modeled. Typically a Dynamic Bayesian network consists of two time slices, the first one depicting the prior distribution at t_0 and the second one depicting the dependences (both exogenous and endogenous) of t_n on t_{n-1} and within t_n. A major drawback of this approach is that the finest time granularity on which events occur determines the time scale of the slices; this is also present in other approaches like [15] that do not use multiple time slices to model dynamic effects. Recent approaches that combine probabilistic logic with dynamics, like [16] also suffer from this problem.

Continuous time Bayesian networks [12] are based on conditional Markov processes. Rather than conditional probabilities a *conditional intensity matrix* (CIM) is associated with each variable, that models the probability of transition of a particular state q_i^x to a state q_{ij}^x given a joint value assignment to its parents.

Using CIMs one is able to use continuous (rather than discrete) time, model endogenous changes (independent of particular events in the problem domain), and reason on different time scales. However, the endogenous changes are limited to exponential decay and while it is possible to specify processes on different timescales, one still needs to choose a uniform time unit to predict the transition time. For example, if one wants to model two independent effects, one happening on a second scale and the other one a month scale, the time unit is in seconds; if we expect that a transition from q_1 to q_2 in six months, we have a probability of remaining in q_1 in the next second of approximately 0.99999993675. These probabilities are both hard to interpret and prone to rounding errors in the inference stage.

4 Case study: using CP-logic to model time granularity

In the processes that play a role in printing using a heavy-duty laser printer, effects can be measured on different time scales. For example, the toner belt that carries the image must be heated in order for the toner to fuse with the paper. This is done using the so-called Heating-On-Demand reflector (HOD). It will take some time (in the order of tens of seconds) before applying power to the HOD will have an effect on the temperature of the HOD. It will then take some time, in the order of seconds, before the toner belt gets heated by the HOD. The effect of accumulating dust on the heater lamp, thus decreasing its productivity, takes months. Typically, if we are interested in events that happen on the second-scale, we assume that the HOD is heated immediately after applying power (i.e., in the same time slice) and that the effectivity of the HOD is constant.

Of course, we can model these effects using different Bayesian networks and use the appropriate network whenever we are interested in particular timescales. This has the obvious disadvantage that the consistency between these models is lost, and if the properties of the machine change, multiple models need to be adjusted. This increases complexity and decreases maintainability of the models. If we choose to combine the knowledge in one dynamic network, we need to choose the smallest common time scale for these processes, i.e., the second range. Observe that there are $3.15 \cdot 10^7$ seconds per year; if we want to measure the effects of dust accumulation in a year we would need to calculate data over $3.15 \cdot 10^7$ time slices. Furthermore, we would suffer from both loss of precision and loss of meaning if we'd need to describe loss of effectivity due to dust accumulation per second.

We suggest to use a different approach, namely to use CP-logic to encode the processes, including the timescales on which they happen, instantiate the timescale on which we want to reason, and use the CP-logic inference mechanism to compute the probabilities of interest[1]. In that way, we can use inference on the time scale that is required, yet keep all the knowledge consistent. One approach could be to use timescale-dependent influences between the variables in the network. This approach can be illustrated using the following example, where the influence between the power applied to the HOD ($\mathrm{P_{HOD}}$), its efficiency ($\mathrm{E_{HOD}}$), the resulting temperature of the HOD ($\mathrm{T_{HOD}}$), and the temperature of the toner belt ($\mathrm{T_{BELT}}$) is modeled (Figure 2). Note that this figure shows static relations. Depending on the time scale we are interested in (seconds or months), we can make the *dynamic* Bayesian networks in Figures 3 and 4.

In Figure 3 we illustrate the situation where we are interested in the effects of a temperature change of the HOD on the temperature change on the toner belt. In T_2, the temperature of the toner belt is based on the temperature of the toner belt in T_1 and on the HOD temperature in T_1. At this time scale, we assume that the HOD temperature immediately increases when power is admitted and that the effectivity of the lamp is constant.

In Figure 4 we illustrate the situation where we are interested in long-term changes in the process due to dust accumulation on the HOD lamp. We assume that the belt heating process is static at this time scale, and we only model the effects of time on the HOD efficiency.

Of course we can describe the static behavior from Figure 2 in CP-logic as follows, using p_X to denote the (unspecified) probability of the CP-event X:

$$\begin{array}{lll}
(\mathrm{P_{HOD}} : p_{\mathrm{P_{HOD}}}). & & \\
(\mathrm{E_{HOD}} : p_{\mathrm{E_{HOD}}}). & & \\
(\mathrm{T_{HOD}} : p_{\mathrm{P_{HOD},E_{HOD}-T_{HOD}}}) & \leftarrow & \mathrm{P_{HOD}}, \mathrm{E_{HOD}}. \\
(\mathrm{T_{BELT}} : p_{\mathrm{T_{HOD}-T_{BELT}}}) & \leftarrow & \mathrm{T_{HOD}}.
\end{array} \tag{7}$$

[1] Alternatively, one could *compile* this partially instantiated CP-logic program into a dynamic Bayesian network, and use appropriate inference engines.

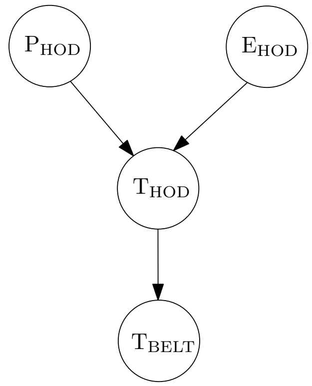

Figure 2: A Bayesian network depicting the relation between $\mathrm{P_{HOD}}$, $\mathrm{E_{HOD}}$, $\mathrm{T_{HOD}}$, and $\mathrm{T_{BELT}}$

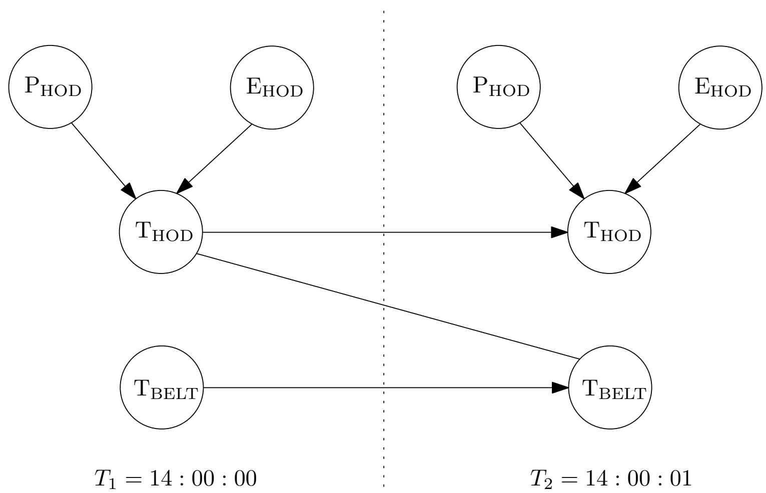

Figure 3: When reasoning on the *second* scale, we assume that the effectivity of the HOD does not change

The dynamic network from Figure 3 can be described as:

$$
\begin{array}{lcl}
(\mathrm{P_{HOD,t}} : p_{\mathrm{P_{HOD,t}}}). & & \\
(\mathrm{P_{HOD,t+1}} : p_{\mathrm{P_{HOD,t+1}}}). & & \\
(\mathrm{E_{HOD,t}} : p_{\mathrm{E_{HOD,t}}}). & & \\
(\mathrm{E_{HOD,t+1}} : p_{\mathrm{E_{HOD,t+1}}}). & & \\
(\mathrm{T_{HOD,t}} : p_{\mathrm{P_{HOD,t}},\mathrm{E_{HOD,t}}-\mathrm{T_{HOD,t}}}) & \leftarrow & \mathrm{P_{HOD,t}}, \mathrm{E_{HOD,t}}. \\
(\mathrm{T_{HOD,t+1}} : p_{\mathrm{T_{HOD,t}},\mathrm{P_{HOD,t+1}},\mathrm{E_{HOD,t+1}}-\mathrm{T_{HOD,t+1}}}) & \leftarrow & \mathrm{T_{HOD,t}}, \mathrm{P_{HOD,t+1}}, \mathrm{E_{HOD,t+1}}. \\
(\mathrm{T_{BELT,t}} : p_{\mathrm{T_{BELT,t}}}). & & \\
(\mathrm{T_{BELT,t+1}} : p_{\mathrm{T_{BELT,t}},\mathrm{T_{HOD,t}}-\mathrm{T_{BELT,t+1}}}) & \leftarrow & \mathrm{T_{BELT,t}}, \mathrm{T_{HOD,t}}.
\end{array}
\tag{8}
$$

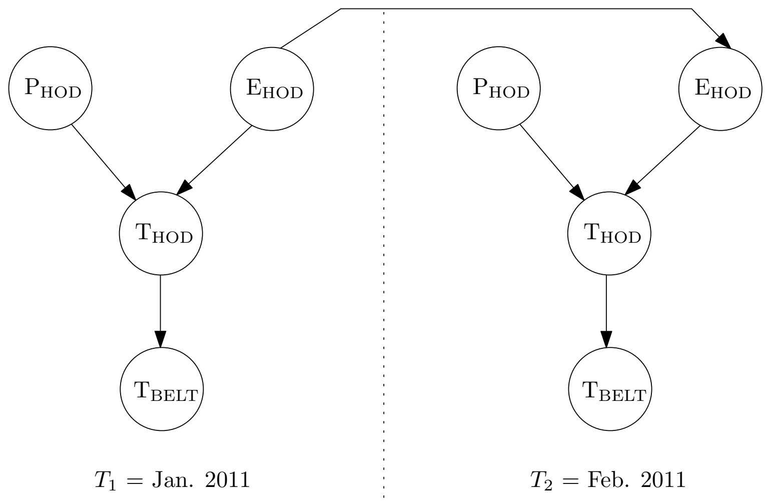

Figure 4: When reasoning on the *month* scale, we assume no temporal effect of temperature changes in the HOD on the belt.

And the dynamic network from Figure 4 as:

$$\begin{array}{lll}
(\mathrm{P_{HOD,t}} : p_{\mathrm{P_{HOD,t}}}). & & \\
(\mathrm{P_{HOD,t+1}} : p_{\mathrm{P_{HOD,t+1}}}). & & \\
(\mathrm{E_{HOD,t}} : p_{\mathrm{E_{HOD,t}}}). & & \\
(\mathrm{E_{HOD,t+1}} : p_{\mathrm{E_{HOD,t}}-\mathrm{E_{HOD,t+1}}}) & \leftarrow & \mathrm{E_{HOD,t}}. \\
(\mathrm{T_{HOD,t}} : p_{\mathrm{P_{HOD,t}},\mathrm{E_{HOD,t}}-\mathrm{T_{HOD,t}}}) & \leftarrow & \mathrm{P_{HOD,t}}, \mathrm{E_{HOD,t}}. \\
(\mathrm{T_{HOD,t+1}} : p_{\mathrm{P_{HOD,t+1}},\mathrm{E_{HOD,t+1}}-\mathrm{T_{HOD,t+1}}}) & \leftarrow & \mathrm{P_{HOD,t+1}}, \mathrm{E_{HOD,t+1}}. \\
(\mathrm{T_{BELT,t}} : p_{\mathrm{T_{HOD,t}}-\mathrm{T_{BELT,t}}}) & \leftarrow & \mathrm{T_{HOD,t}}. \\
(\mathrm{T_{BELT,t+1}} : p_{\mathrm{T_{HOD,t+1}}-\mathrm{T_{BELT,t+1}}}) & \leftarrow & \mathrm{T_{HOD,t+1}}.
\end{array} \tag{9}$$

But now, we are reusing a lot of information and we construct two models where we should have only one. Using additional variables $\mathrm{G_{SEC}}$ and $\mathrm{G_{MONTH}}$ we can instantiate (i.e., add as a fact to the knowledge base) whenever we want to reason on the second or month level. We can then describe *both* networks in one logic program as follows:

$$\begin{array}{lll}
(\mathrm{P_{HOD,t}} : p_{\mathrm{P_{HOD,t}}}). & & \\
(\mathrm{P_{HOD,t+1}} : p_{\mathrm{P_{HOD,t+1}}}). & & \\
(\mathrm{E_{HOD,t}} : p_{\mathrm{E_{HOD,t}}}). & & \\
(\mathrm{E_{HOD,t+1}} : p_{\mathrm{E_{HOD,t+1}}}) & \leftarrow & \mathrm{G_{SEC}}. \\
(\mathrm{E_{HOD,t+1}} : p_{\mathrm{E_{HOD,t}}-\mathrm{E_{HOD,t+1}}}) & \leftarrow & \mathrm{G_{MONTH}}, \mathrm{E_{HOD,t}}. \\
(\mathrm{T_{HOD,t}} : p_{\mathrm{P_{HOD,t}},\mathrm{E_{HOD,t}}-\mathrm{T_{HOD,t}}}) & \leftarrow & \mathrm{P_{HOD,t}}, \mathrm{E_{HOD,t}}. \\
(\mathrm{T_{HOD,t+1}} : p_{\mathrm{T_{HOD,t}},\mathrm{P_{HOD,t+1}},\mathrm{E_{HOD,t+1}}-\mathrm{T_{HOD,t+1}}}) & \leftarrow & \mathrm{G_{SEC}}, \mathrm{T_{HOD,t}}, \mathrm{P_{HOD,t+1}}, \mathrm{E_{HOD,t+1}}. \\
(\mathrm{T_{HOD,t+1}} : p_{\mathrm{P_{HOD,t+1}},\mathrm{E_{HOD,t+1}}-\mathrm{T_{HOD,t+1}}}) & \leftarrow & \mathrm{G_{MONTH}}, \mathrm{P_{HOD,t+1}}, \mathrm{E_{HOD,t+1}}. \\
(\mathrm{T_{BELT,t}} : p_{\mathrm{T_{HOD,t}}-\mathrm{T_{BELT,t}}}) & \leftarrow & \mathrm{T_{HOD,t}}. \\
(\mathrm{T_{BELT,t+1}} : p_{\mathrm{T_{BELT,t}},\mathrm{T_{HOD,t}}-\mathrm{T_{BELT,t+1}}}) & \leftarrow & \mathrm{G_{SEC}}, \mathrm{T_{BELT,t}}, \mathrm{T_{HOD,t}}. \\
(\mathrm{T_{BELT,t+1}} : p_{\mathrm{T_{HOD,t+1}}-\mathrm{T_{BELT,t+1}}}) & \leftarrow & \mathrm{G_{MONTH}}, \mathrm{T_{HOD,t+1}}.
\end{array} \tag{10}$$

Using this approach, we do not need to maintain separate models for both timescales, while still being able to reason on different time scales. Note that it might be possible, at least in this small example, to construct a combined dynamic Bayesian network with an additional variable TimeScale and adjust the conditional probabilities of each variable that is affected by a change in timescale. However, this has several disadvantages: we increase the number of incoming arcs of a number of variables, thus increasing the conditional probability table by an order of magnitude (probably also introducing problems when learning the network from data) and we clutter the networks with meta-information that is not part of the domain knowledge.

5 Conclusion

We investigated, using a case study relevant to heavy-duty printing, to what extent probabilistic logics as CP-logic are able to model uncertain dynamic processes with vastly different time scales. CP-logic is well suited to combine several related dynamic Bayesian networks, reasoning on the desired timescale by adding a temporal variable to the knowledge base. This approach allows to maintain a single model, thus aiding to the maintainability of the system. The resulting logic program can either be run using a CP-logic implementation, or translated to (multiple) dynamic Bayesian networks.

An extension to this idea may be to use probabilistic logics to model slightly different instances of similar artifacts, e.g., printers with different finisher configurations. While these finishers (like staplers and binders) typically have their own power supply, they do influence the printer behavior, e.g., by altering the prior probabilities in the network. For example, when a book binder is attached to the printer, typically the user will be printing relatively more on glossy and heavier paper. Using probabilistic logic may be used to maintain a single model for many configurations, and aid to query such model with respect to properties that hold for *some* or *all* configurations, using existential and universal quantifiers.

Acknowledgements

JK is supported by the OCTOPUS project under the responsibility of the Embedded Systems Institute. This project is partially supported by the Netherlands Ministry of Economic Affairs under the Embedded Systems Institute program. The authors wish to thank the project members for valuable discussion regarding various ideas put forward in this paper. In particular they wish to thank Jochem Brok and René Waarsing from Océ for providing them with the necessary background knowledge.

References

[1] H. P. Blossfeld and G. Rohwer. *Event History Analysis*. Lawrence Erlbaum Associates, 1988.

[2] D. Colbry, B. Peinter, and M. E. Pollack. Execution monitoring with quantitative temporal Bayesian networks. In *Proceedings of the Sixth International Conference on AI Planning and Scheduling*, 2002.

[3] P. Dagum, A. Galper, and E. Horvitz. Dynamic network models for forecasting. In *Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence*, pages 41–48, 1992.

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

[5] S. Dey and J. A. Stori. A Bayesian network approach to root cause diagnosis of process variations. *International Journal of Machine Tools and Manufacture*, 45(1):75–91, 2005.

[6] S. Hanks, D. Madigan, and J. Gavrin. Temporal reasoning in probabilistic knowledge-based systems. In *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence*, pages 111–119, 1995.

[7] H. J. Henriksen, P. Rasmussen, G. Brandt, D. von Bülow, and F. V. Jensen. Public participation modelling using Bayesian networks in management of groundwater contamination. *Environmental Modelling and Software*, 22(8):1101–1113, 2007.

[8] A. J. Hommersom and P. J. F. Lucas. Using Bayesian networks in an industrial setting: Making printing systems adaptive. In *Nineteenth European Conference on Artificial Intelligence*, pages 401–406, 2010.

[9] F. V. Jensen and T. D Nielsen. *Bayesian Networks and Decision Graphs*. Springer Verlag, New York, NY, second edition, 2007.

[10] R. J. Kennett, K. B. Korb, and A. E. Nicholson. Seabreeze prediction using Bayesian networks. In D. Wai-Lok Cheung, G. J. Williams, and Q. Li, editors, *Proceedings of the Fifth Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, volume LNCS 2035, pages 148–153. Springer Verlag, Berlin, 2001.

[11] P. J. F. Lucas, N. de Bruijn, K. Schurink, and A. Hoepelman. A probabilistic and decision-theoretic approach to the management of infectious disease at the ICU. *Artificial Intelligence in Medicine*, 3(19):251–279, 2000.

[12] U. Nodelman, C. R. Shelton, and D. Koller. Continuous time Bayesian networks. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, pages 378–387, 2002.

[13] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, Palo Alto, CA, 1988.

[14] J. C. Ramírez, G., Muñoz, L., and Gutierrez. Fault diagnosis in an industrial process using Bayesian networks: Application of the junction tree algorithm. In *The 2009 Electronics, Robotics and Automotive Mechanics Conference (CERMA 2009)*, pages 301–306, 2009.

[15] A. Y. Tawfik and E. M. Neufeld. Temporal reasoning and Bayesian networks. *Computational Intelligence*, 16:349–377, 2000.

[16] I. Thon, N. Landwehr, and L. De Raedt. Stochastic relational processes: Efficient inference and applications. *Machine Learning*, 82(2):239–272, 2011.

[17] L. C. van der Gaag, S. Renooij, C. L. M. Witteman, B. M. P. Aleman, and B. G. Taal. Probabilities for a probabilistic network: a case study in oesophageal cancer. *Artificial Intelligence in Medicine*, 25:123–148, 2002.

[18] J. Vennekens, M. Denecker, and M. Bruynooghe. CP-logic: A language of causal probabilistic events and its relation to logic programming. *Theory and Practice of Logic Programming*, 9(3):245–308, 2009.

[19] H. Wasyluk, A. Onisko, and M. J. Druzdzel. Support of diagnosis of liver disorders based on a causal Bayesian network model. *Medical Science Monitor*, 7:327–332, 2001.

[20] K. Yamaguchi. *Event history analysis*, volume 28. Thousand Oaks, CA: Sage Publications, 1999.

[21] C. Zhang, S. Sun, and G. Yu. A Bayesian networks approach to time series forecasting of short-term traffic flows. In *Seventh International IEEE Conference on Intelligent Transportation Systems*, pages 216–221, 2004.