Probabilistic System Summaries for Behavior Architecting

Michael Borth

Embedded Systems Innovation by TNO,
PO Box 6235, 5600 HE Eindhoven, The Netherlands
Michael.Borth@tno.nl

Abstract. Smart system of systems adapt to their context, current situation, and configuration. To engineer such systems' behavior, we need to design and evaluate system-level control strategies and the intelligent management of key scenarios. We propose a model-based approach called *probabilistic system summaries* to explore related design choices, e.g., where to put the 'smarts' of a smart building. Our approach uses Bayesian inference to calculate effects of strategies and implementations, offering causal analysis of the costs and benefits of decision strategies in key scenarios. As the modeling is light-weight and suitable for various abstraction levels, probabilistic system summaries are appropriate for early but sound architecture decisions based on computational science. Next to its use within this analysis, the product of this engineering step, i.e., a Bayes net summarizing systems plus their environment, may form the core of decision making within the system of system..

1 Engineering of Smart Buildings

Smart systems of systems are set to recognize and adapt to their situation, to their operational context, and to their configuration, and to enact smart strategies to reach their goals in the best possible way. One example of such systems are smart buildings, where building automation is established to detect environmental circumstances, faults, utilization, or emergency situations, and then to act accordingly. They can, e.g., adapt building operations to energy saving demands while tuning services to the number of people present in a room and furthermore compensating for services that are not available due to failures. Such buildings are systems of systems: lighting, HVAC (heat, ventilation, air conditioning), security, etc. are individually developed and commissioned systems that operate independently, fulfil different and partially conflicting goals of various stakeholders, but benefit from cooperation. Especially an exchange of information is beneficial, e.g., on the presence of people while balancing comfort with energy savings. Complexity and size of building automation systems and high costs of commissioning make it desirable to realize building automation in a self-organizing, cooperative, and robust way.

These advanced functionalities require among other things I) the means to monitor the building and its environment, including the ability to detect faults and special events, II) system-level control strategies and scenario-management that allow the handling of and adaptation to foreseen circumstances, and III) solutions to handle unforeseen dynamics, e.g., fault adaptive behavior or an adaptation to requirement changes. In this article, we define our system-of-systems engineering method for the design, analysis, and architecture of behavior w.r.t. II), i.e., the model-based approach *probabilistic system summaries* that we use to investigate design alternatives and their trade-offs within the engineering of the 'smarts' of a smart system of systems.

1.1 Behavior Architecting within System of Systems Engineering

Following the argument that system of systems engineering (SoSE) focuses on choosing the right systems and their interactions to satisfy the requirements and goals of various stakeholders, we consider the architecting of behavior as a core task of SoSE. The focus on architecting, and not the mere design of functionality, acknowledges that the 'where and when' of a decision taken by a system matters - especially for systems of systems (SoS). Imagine, e.g., two alternate system control strategies in smart buildings: smart rooms that take local, if cooperative, decisions versus a smart building with centralized control. This architectural choice impacts among other things the need for data exchange and thus communication means (affecting costs and privacy), robustness (reliability of many cheap components or single point of failure), but also which data is available to adapt behavior and thus functionality itself.

An investigation of architectural alternatives that targets smart behavior requires methods and tools that enable a lightweight modeling of the solution candidates linked to computational science. It should allow the calculation of key performance indicators with regard to both the SoS' functional and non-functional aspects, e.g., the energy savings achieved by a decision strategy for efficiency or costs of operations. In this, we require a lightweight method, as a detailed process for each alternative incurs undue costs; computational science, i.e., modeling with quantitative analysis, is needed to discriminate decision strategies and architectures with regard to their expected business value, e.g., total costs-of-ownership.

1.2 Use Case: Where to Put the Smarts of a Building?

We support the introduction of the *probabilistic system summaries* method with a simplified case study, the investigation of the two architecture alternatives mentioned above, which addresses a central architectural question on decision making – where to locate it: locally, close to sensors and actuators, thus in many places with a limited range of impact but with direct communications, or in a central instance? Worded differently within the example, this is a choice between a smart building in which rooms without intelligence enact the building's strategy, or smart rooms that take local data to produce local decisions, while occasionally taking global information into account, which is provided by the building.

Most sensor data is local, about an individual area, e.g., a room, but global data always plays a role as well, e.g., outside temperature or available power. Actuators are often local, e.g., lights or air conditioning within a room, but shutting down the heating system is a global action. Strategies and goals might be individual and local, e.g., about the intended level of comfort in a room, or global, e.g., a reduction of energy consumption. The latter aspects, comfort and consumption, will serve as performance indicators of our use case. They represent conflicting goals of different stakeholders, i.e., building managers seek energy efficiency and thus minimal consumption (best achieved by turning everything off), while inhabitants seek comfort, possibly with very in-efficient ideas how to achieve that.

While this over-simplifies the architectural task and ignores performance indicators like costs or robustness of a realization, we believe that the reduction to two opposing indicators and two distinct architectural concepts for decision making and control works well for the demonstration of our design and analysis method. System architects working with our method will be able to define and add further performance indicators to suit their engineering tasks. In any way, we need to point out that the question which architectural solution to choose has no answer that holds for all possible buildings and possibly not even one that holds within one building for all situations. It must be investigated for each SoS – as any such decision impacts costs for operations and the infrastructure, and furthermore depends on the goals, situation, configuration, and many factors more, including non-functional aspects.

In our work, we also look into different AI techniques for smart self-organizing building automation, described in documents and papers listed at [1], with details on the aspects elaborated in this article in [2]. Such choices are not covered here. Instead, we assume that a technique is available and investigate its effects. Still, it is notable that certain architectures favor certain AI techniques to implement the needed smarts, e.g., local decisions with loose coupling work well with agent-based technologies.

2 Foundations: Bayesian Modeling

The decision between architectural alternatives warrants an investigation of each of them with regard to their costs, benefits, and effects. To avoid costly experimentation, we base such investigations on model-based computational science, with simulation, calculation, and probabilistic assessment as possible techniques. We opt for Bayesian networks [3] as probabilistic models. These graph-based representations of the joint probability distribution over all modeled variables offer causal probabilistic modeling [4], optimal to investigate cause-effect relationships, e.g., what impact a strategy has on energy consumption, given that probabilistic factors, like weather or the building usage, affect the outcome. As their calculus allows for the inference of probability distributions over variables given evidence or assumed circumstances, it becomes feasible to investigate the range and likelihood of possible outcomes. Furthermore, Bayes nets allow sensitivity analyses to determine the possible impact of factors [5], e.g., to investigate the dependency of a system's performance on the environment.

While the literature listed above provides extensive knowledge on Bayesian networks, the remaining article only assumes familiarity with the core concepts: Bayes nets are graphs with (random) variables as nodes. Directed edges between nodes show their relationships, i.e., probabilistic or causal dependencies, encoded as conditional probability distribution of a variable given all its parents. Causal relations are often functional: Given the cause, the effect is determined in a functional manner, with the conditional probabilities that encode the likelihood of a variable's states given other variables' states either at 0 or 1 – *impossible or always* true under the specified conditions. Probabilistic dependencies encode influence that is more random and flexible, e.g., a different likelihood for room occupancy given that the building is nearly full in comparison to situations where the building is mostly empty. Bayesian networks may either be modeled manually via knowledge engineering, or learned from data. Together with their real-time capabilities, this makes them suitable for many domains.

Several modeling techniques exist that enable the efficient use of Bayes nets for system modeling, e.g., by supporting re-use with object-orientation [6], and semi-automated construction of networks from knowledge bases [7] or system descriptions [8]. All this ensures a reduction of efforts as well as consistency: As we set out to compare strategies for system-level control for a given building that has a given environment, we use identical network building blocks, called *network fragments*, for the fixed points of our analysis, adding individually developed fragments for the variable parts. As all these fragments are constructed according to the same principles, they can be merged easily into a full Bayesian network.

3 Probabilistic System Summaries

In this section, we detail our probabilistic modeling that renders investigations of architectural alternatives for smart behavior. Essentially, the approach consists of a methodology to construct, in a comparable fashion, a Bayes net for each control strategy under investigation together with its context, i.e., a summary of the system and environment. Special nodes in the networks allow the computation of utility values of the control strategy, e.g., for energy consumption or comfort, given the assumptions encoded in the networks, since the Bayes nets represent the joint probability distribution over all modeled variables. The modeled strategies thus allow for experiments, e.g., to compare the utility values of various strategies given different assumptions.

3.1 System Summaries for Efficient Investigations

Our goal is to encode a global view on the system, in our domain a smart building, so that the impact (cause and effect) of strategies may be investigated. This global view requires insights in the distribution over possible values of key variables in the sense of a summary: There is, e.g., one variable for room occupancy and not one for each room. A distribution over this variable together with dependent variables is sufficient; for example, one needs to know that 70% of the rooms are occupied within a given scenario and that this level of occupancy results in a certain energy need.

Such a causal and probabilistic modeling that encodes a global view at the building allows for small models. This limits modeling efforts, especially compared to simulations that include individual rooms. However, even small networks hold a great number of parameters: With discrete states as possible values of the modeled variables, a node in the graph holds one parameter per state for each possible combination of the states of all parent variables. This leads to thousands of parameters in bigger networks. However, parameters may be computed automatically for functional dependencies, greatly reducing efforts. For non-functional dependencies, parameters are determined via a knowledge engineering process: estimates of the parameters stem from statistics, e.g., on room occupancy, or from an evaluation by experts. The object-oriented approach to model allows doing so in a constructive manner: a fragment encapsulates a piece of the domain, so that its modeling provides no challenge, as its relationships are easily understood. The subsequent construction of the Bayesian network from fragments follows a strict methodology, which may be used manually or even semi-automated, e.g., from a knowledge base (see references above and Chapter 7 of [9] for details of fundamental modeling techniques).

3.2 Use Case: Network Fragments for Smart Buildings

For an easily understood illustration of our work, we start with the following set of probabilistic variables (*in italics*) and the dependencies between them:

a) *Usage of the building*, a measure of utilization that impacts the likelihood of *room occupancy*.
b) *Condition of room*, a measure how outside factors set a room status, e.g., the temperature due to direct sunlight. It impacts, together with *room occupancy*, the *need of the room* with regard to services that consume energy, e.g., to cool the room down.
c) *Services provided* for a room, a measure of service-level provided for a room, which depends on *room occupancy* and the *need of the room*. (Furthermore the policy, not shown, which is not probabilistic, but user determined, as it describes a decision, e.g., to save energy. See below.)
d) The *consumption of the room* follows from the *services provided* to the room.
e) The energy *consumption of the room* determines the energy *consumption of the building*. The latter is a utility variable, i.e., a variable that shows a decision's utility value, in this case the aforementioned policy.
f) *Discomfort* within the building, another utility variable, which results from the discrepancy between the needs and services provided in occupied rooms.

Fig. 1 depicts the network fragments for these variables. It becomes visible that the fragments can be fused into a network that describes a building: conditions and occupancy determine need, services respond to need and consume energy, and a delta between need and services results in discomfort in occupied rooms. The probabilities are easy to determine: with statistic for variables without parents, with distributions for probabilistic relations, with causality for functional relations.

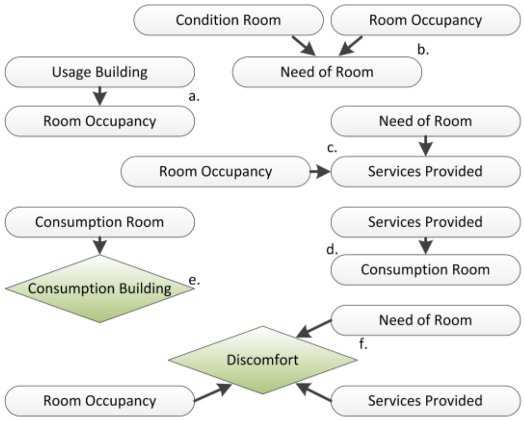

Fig. 1. Network fragments of simplified smart building use case

More elaborate models might mirror reality with less abstraction, e.g., by distinguishing between offices and meeting rooms instead of summarizing all rooms. This is, however, for the system architect to consider: a probabilistic system summary may be generated from known details as well as from more general insights. The network's causal nature ensures that different abstractions work together.

3.3 System-level Control Strategies

The decision process, i.e., the smarts of the building, is not explicit in the network fragments we introduced above. While it could be encoded within fragment *c*, which models the services provided to a room, this would hide the reasoning which is the goal of our investigation. Instead, we compose fragments that mirror the decision strategy and thus complement the other fragments that mirror the environment and processes. These fragments encode the causal effects of the strategy given the data available to the decision taker, incl. the current energy saving policy, summarizing both the decisions based on the available data and its effects on the system.

To analyze different strategies comparatively, we require a model for each strategy. The network fragments forming these models have two distinct aspects: the reasoning process, encoded in the network structure, and probabilities that define the strategy's parameters. The structure is engineered from the understanding which information is processed how within the system for decision takings. This might include complex steps, e.g., to account for missing data where the building automation has to use estimates. The strategy's parameters, on the other hand, are typically computed or determined with experiments, e.g., to find a policy set-point for services given a certain context so that a required energy saving is ensured. The modeling of the control strategies follows the same probabilistic summary techniques as the modeling of the environment and system interactions. It is, e.g., sufficient to know a distribution over alternatives for services set-points, disregarding where exactly a set-point is in effect.

3.4 Use case: Network Fragments for Control

In continuation of the simplified example, we show two distinct ways of building automation. Fig. 2 depicts the Bayesian network for smart rooms that take local decisions while considering global requests: Given local data on *room occupancy* and the *condition of the room*, the *need of the room* is determined. Given that and a *policy on energy savings*, the room sets the *services provided*. *Usage of the building* sets the distribution of the *room occupancy*. Utility nodes on *discomfort* and *consumption* follow functionally. This network holds close to 400 parameters. Fig. 3 shows a smart building that enacts a global strategy, and thus altogether a very different information flow and computational model. Here, information on the *needs of the rooms* and thus the *needs of the building* is collected, the outcome is compared to the constraints set by the *energy saving policy*, which leads to insights into *required savings*, that are then used to set an appropriate *strategy* that fulfils the savings while minimizing *discomfort*. This latter step exploits the information available through the collection of the room data: If, e.g., the calculation shows that the use of an aggressive saving strategy in empty rooms is sufficient to meet the requirements, discomfort in other rooms may be avoided altogether – a fact that an individual smart room could not have taken into account. The network holds over 1000 parameters. Fig. 4. shows a set of parameters that define set-points on how an energy saving policy for smart rooms changes the services provided to a room given its needs. It is key to understand that the complete models handle summaries in the form of distributions: if such conditions are given, this mixture of decisions will be taken, resulting altogether in these effects.

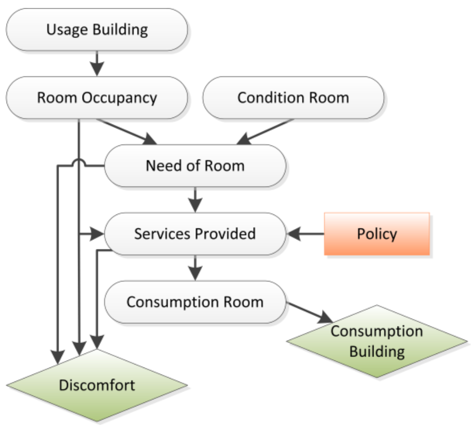

Fig. 2. Simplified Bayesian network for smart room building control

3.5 Network Formation

The final step in the modeling process is the formation of the network structure, i.e., the construction of the graph from the network fragments and other necessary nodes. This process, which may be partially automated [8], mirrors the information flows of the strategy and its realization, i.e., functional aspects as laid out above and other architectural aspects. Fig. 2 and Fig. 3 show the out-come of this process. In these networks, we thus see a local decision process in smart rooms that deduct the services provided from locally available information while taking the global energy policy into account, while the smart building decides its strategy after global budget calculations.

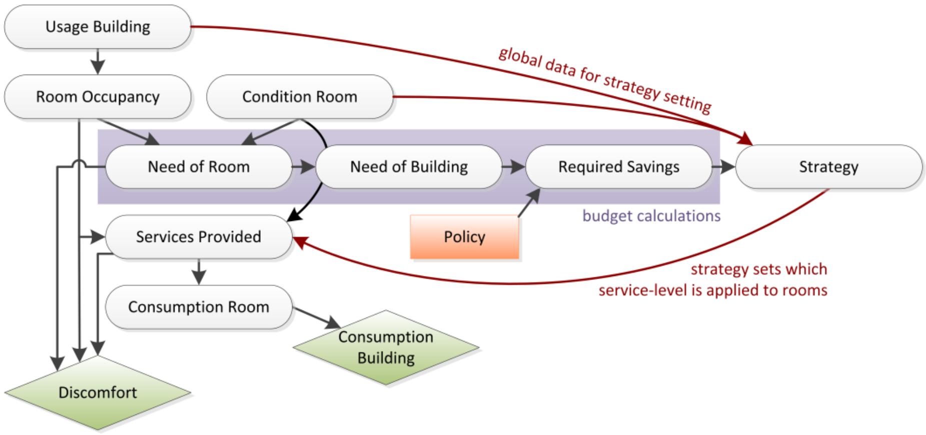

Fig. 3. Simplified Bayesian network for smart central decisions building control

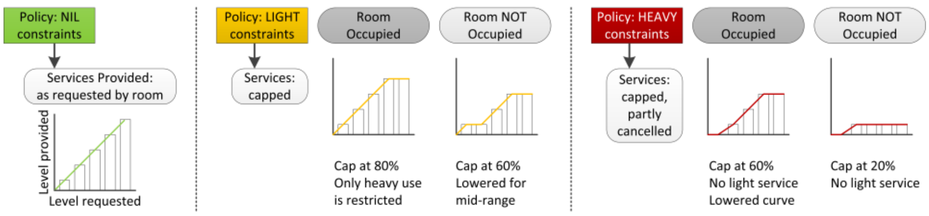

Fig. 4. Strategy for smart rooms: *services provided* given *occupancy*, need of room (*level requested*) for three energy saving policies (NIL, LIGHT, and HEAVY constraints)

If we encounter non-functional concerns, especially w.r.t. the realization architecture, we approach the modeling of their impact on the reasoning in the same way. Imagine, e.g., a masking of occupancy information for a building section due to privacy concerns, which results in incomplete information in the respective node. If the building's control is realized without mechanisms to compensate for this, we would soften the distributions of the room occupancy to allow for a wider range of values within our calculations. If the building's control has a setup to estimate these figures, we would include this flow of information, resulting e.g., in a room occupancy that follows from a combination of observations and an estimate model, which might be based on date and time, or car park observations. Fig. 5 summarizes workflow and considerations of the network generation process: The *Reasoning Network* encodes the information flows and functional aspects. If the realization architecture warrants an adaptation, the additional steps result in the final Bayesian network.

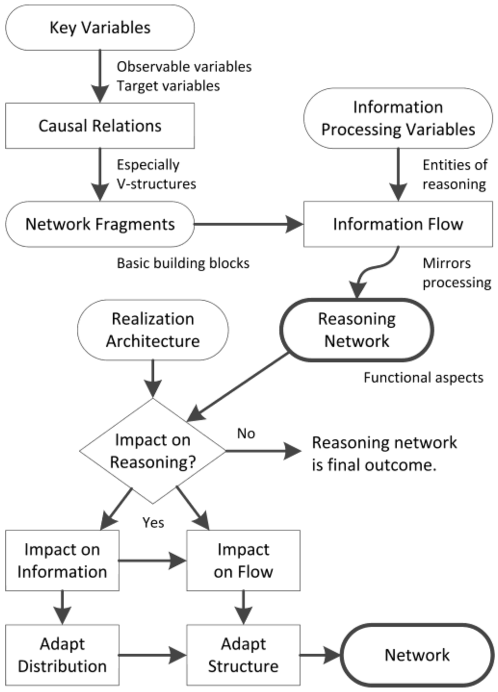

Fig. 5. Process of network generation

4 Experiments to Investigate Strategies

To compare the impact of the modeled decision strategies, e.g., the one for smart building with rooms without decision power and the one with smart rooms but no decision power for the building, we conduct a set of experiments. In each of those, applicable environmental settings, like *building usage*, are set (entered as evidence) in the respective Bayesian network (one per strategy). The effects of all possible energy saving policies or other circumstances on target variables like *discomfort* and *consumption* are computed individually via the probabilistic inference of the network, i.e., the circumstance is set and the Bayes network infers the distribution of all target variables given the evidence entered for the environmental settings.

This setup allows for experiments on the sum of all environmental circumstances, but also investigations that are specific, e.g., for high occupancy under extreme weather conditions and demanding policies. While a presentation of our actual results is pointless within this presentation of the approach due to the simplification of the networks, we present some details that we consider useful to gauge the approach. With regard to the efforts, we note first that an in-depth comparison of two strategies is possible within a few hours, while a high level estimation is merely a matter of minutes once the modeling is done. Second, we advise to run additional experiments with the purpose to test the models, establishing a standard engineering practice to safeguard results and thus subsequent business decisions. Mostly, such test will look into fragments and their links, but also wider aspects are sensible. In our example, we checked that the target variables *consumption* and *discomfort* show identical values for all comparable experiments if no energy saving policy is in effect, but we also required that a smart building can leverage its information advantage, i.e., mere smart rooms should not outperform it on these indicators.

In addition to these investigations, it is possible to reason backwards, from effects to possible causes, e.g., to check the probability of circumstances that lead to levels of discomfort that are unacceptable. This is relevant for the dimensioning of systems and the setting of strategies, as it mirrors service-level agreements. Another type of experiments is the investigation of so-called counter-factuals that describe an alternative reality, e.g., a different building in which the condition of the room leads to different needs of those rooms, e.g., due to stronger insulation. This is useful to initiate change and improvements in a sensible manner. In [2], we detail the workflow for all types of experiments and provide insights into the stability of the results inferred from probabilistic summaries w.r.t. the precision of the network parameters.

5 Operational Use

Once a suitable decision strategy is found with the analysis methods described here, it is possible to implement it with live reasoning that uses a Bayesian network which takes observational data into account. The Bayesian network used for analysis forms a direct input for this; it is sufficient to adapt the level of details to the one observable for the building automation, keeping the structure of the reasoning intact. This works due to the construction of the network as system summary – analysis and operational control use identical reasoning.

6 Discussion

We introduced a method to model decision architectures of smart systems-of-systems that take their context, configuration, and current situation into account to change their behavior according to a dynamic or pre-set strategy. Our modeling summarizes the system, its behavior, and the impact of control decisions. This summarization is encoded in Bayesian networks; it uses probabilistic distributions over all key variables and their relationships. Using such a probabilistic summary allows to investigate architectural alternatives regarding system-level control strategies and the intelligent management of scenarios. This can be done with little efforts in experiments in which effects – and thus costs and benefits – of alternatives are computed and compared. Given the light-weight and modular manner of modeling, this sequence of modeling, experimental analysis, and investigation of cause-and-effect relations grants the benefits of an exploration of the design-space of the system's control architecture and behavior that is based on computational science. This allows system architects to take sound decisions, e.g., on where to put the 'smarts' of a smart building.

6.1 Relation to Engineering and AI

Our work addresses the architecting of smart behavior and control structures of smart systems-of-systems. It is therefore at the interface of system-of-systems engineering to artificial intelligence; two communities that only recently started to interact. It is our observation that existing work of these domains often take the contribution from the other domain out of their considerations: AI puts forward algorithms for smart behavior given a system layout – system engineering investigates designs assuming fixed behavior patterns (which are often close to traditional engineering). We believe that these more isolated points-of-view disregard the impact of a system's architecture to the possibility, quality, and efficiency of computations for behavior and control, and, vice versa, the demands of such computations on architecture. Realizing that the 'when' and 'where' of computations affects the 'what', we identified the need for advances w.r.t. the architecting of behavior and system-level-control, for which we propose our *probabilistic system summaries* as one modeling and analysis technique.

While we see our main contribution in this focus on a smart system-of-systems' architecture, we advise a strong link to operational aspects of decision strategies for future work. Medina-Oliva et al., e.g., propose to support the assessment of maintenance strategies of industrial systems using probabilistic relational models (PRM) in [10]. Their use of key performance indicators as optimization goals within a probabilistic framework is similar to our use of utility variables, pointing to a feasible integration of this work with sustainable operations management [11].

6.2 Implementation and Feasibility

The efforts to conduct an architecture investigation with *probabilistic system summaries* is very low, especially in comparison to alternatives like a detailed simulation where all individual objects are modeled. This assessment is based on experience, as we cannot pursue alternative approaches for various techniques in detail. We can, however, pinpoint various additional advantages: First, the models are Bayesian networks, for which both commercial and open-source software tools exist that offer well-established algorithms and suitable human-computer-interfaces. There is no need for additional engineering tools. Second, the modeling is dual purpose in the sense that models for analysis may be used for the operation of the smart system as well – thus reducing engineering efforts. Given the capabilities of smart systems, building automation systems in our domain, and the efficiency as well as real-time suitability of Bayesian networks, it is feasible to realize this with little to no extra costs.

However, we must re-consider the modelling efforts together with the efforts to fuse network fragments when we extend our approach to include more aspects into the architectural summary, e.g., for operations management as proposed above. Work by Koller et al. on probabilistic relational models [12] together with the foundations on probabilistic frame-based systems in [13] covers the modelling of large complex domains with the coherent probabilistic representation of Bayesian networks. As this work has many application domains, we expect further advances regarding tool support, further guaranteeing the feasibility of industrial use of modelling system summaries.

Acknowledgements

This work was partly funded via the EU FP7 ICT project SCUBA – Self-organising, Co-operative and robust Building Automation (www.ict-scuba.eu).

References

1. Scuba FP7 Project, Scientific papers. www.aws.cit.ie/scuba/index.php/documents/scientific-papers/
2. Borth, M.: Reasoning for smart control strategies. In Scuba FP7 Project, Deliverable 4.2: Scenario management, control and self-organization strategies, initial adaptation strategies, in press (2014)
3. Jensen, F.V.: Bayesian networks and decision graphs. Springer (2001)
4. Pearl, J.: Causality. Models, reasoning, and inference. Cambridge University Press (2000)
5. Jensen, F.V., Aldenryd, S.H., Jensen, K.B.: Sensitivity analysis in Bayesian networks. In: Symbolic and Quantitative Approaches to Reasoning and Uncertainty Vol. 946, Carbonell, J.G., Siekmann, J., Goos, G., Hartmanis, J., Leeuwen, J., Froidevaux, C., Kohlas, J. (eds.) Springer Lecture Notes in Computer Science, pp. 243–250 (1995)
6. Koller, D., Pfeffer, A.: Object-oriented Bayesian networks. In: Geiger, D., Shenoy, P.P. (eds.) Proceedings of the 13th conference on uncertainty in artificial intelligence (UAI'97). Morgan Kaufmann Publishers Inc., pp. 302–313 (1997)
7. Laskey, K.B., Mahoney, S.M.: Network fragments: representing knowledge for constructing probabilistic models. In: Geiger, D., Shenoy, P.P. (eds.) Proceedings of the 13th conference on uncertainty in artificial intelligence (UAI'97). Morgan Kaufmann Publishers Inc., pp. 334–341 (1997)
8. Borth, M., von Hasseln, H.: Systematic generation of Bayesian networks from systems specifications. In Musen, M.A., Neumann, B., Studer, R. (eds.) Intelligent information processing. Kluver, pp. 155–166 (2002)
9. Kjærulff, U.B., Madsen, A.L.: Bayesian networks and influence diagrams: a guide to construction and analysis. Springer (2013)
10. Medina-Oliva, G., Weber, P., Iung, B.: PRM-based patterns for knowledge formalisation of industrial systems to support maintenance strategies assessment. Reliability Engineering & System Safety, 116, 38–56. doi:10.1016/j.ress.2013.02.026 (2013)
11. Kleindorfer, P.R., Singhal, K., Van Wassenhove, L.N.: Sustainable operations management. Production and Operations Management, vol. 14 (4) pp. 482-492 (2005)
12. Getoor, L., Friedman, N., Koller, D., Pfeffer, A., Taskar, B.: Probabilistic relational models. In: Getoor, L., Taskar, B. (eds.) An introduction to statistical relational learning, MIT Press (2007)
13. Koller, D., Pfeffer, A.: Probabilistic frame-based systems. In: Mostow, J., Rich, C., Buchanan, B. (eds.) Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence (AAAI '98/IAAI '98). American Association for Artificial Intelligence, pp. 580-587 (1998)