

Availability of enterprise IT systems: an expert-based Bayesian framework

Ulrik Franke · Pontus Johnson · Johan König ·
Liv Marcks von Würtemberg

Published online: 13 May 2011
© Springer Science+Business Media, LLC 2011

Abstract Ensuring the availability of enterprise IT systems is a challenging task. The factors that can bring systems down are numerous, and their impact on various system architectures is difficult to predict. At the same time, maintaining high availability is crucial in many applications, ranging from control systems in the electric power grid, over electronic trading systems on the stock market to specialized command and control systems for military and civilian purposes. This paper describes a Bayesian decision support model, designed to help enterprise IT system decision-makers evaluate the consequences of their decisions by analyzing various scenarios. The model is based on expert elicitation from 50 experts on IT systems availability, obtained through an electronic survey. The Bayesian model uses a leaky Noisy-OR method to weigh together the expert opinions on 16 factors affecting systems availability. Using this model, the effect of changes to a system can be estimated beforehand, providing decision support for improvement of enterprise IT systems availability. The Bayesian model thus obtained is then integrated within a standard, reliability block diagram-style, mathematical model for assessing availability on the architecture level. In this model, the IT systems play the role of building blocks. The overall assessment framework thus addresses measures to ensure high availability both on the level of individual systems and on the level of the entire enterprise architecture. Examples are presented to illustrate how the framework can be used by practitioners aiming to ensure high availability.

Keywords Systems availability · High availability · Downtime · Bayesian networks · Noisy-OR · Expert elicitation

U. Franke (✉) · P. Johnson · J. König · L. Marcks von Würtemberg
Industrial Information and Control Systems, Royal Institute of Technology, Stockholm, Sweden
e-mail: ulrikf@ics.kth.se

P. Johnson
e-mail: pj101@ics.kth.se

J. König
e-mail: johank@ics.kth.se

L. Marcks von Würtemberg
e-mail: livm@ics.kth.se

1 Introduction

Maintaining high enterprise IT systems availability is a high priority throughout many industries. In a frequently cited report, IBM Global Services (1998) report that unavailable systems cost American businesses \$4.54 billion in 1996, due to lost productivity and revenues. The report goes on to list average costs per hour of downtime ranging from airline reservations at \$89.5 thousand to brokerage operations at \$6.5 million (all in 1998 dollars). A vivid reminder of the financial sector's sensitivity occurred when the Nordic and Baltic stock markets were forced to close down for 5.5 h on June 4 2008 due to the trading system Saxess being down. This outage prevented transactions worth approximately 20 billion SEK (ca €2 billion) (Askåker and Kulle 2008).

While useful in some contexts, cost estimates do not always accurately reflect the criticality of systems availability. This is often the case for IT systems supporting emergency response, police and military operations, operation of the power grid, etc. In a recent Gartner report, it is therefore recommended that investments to ensure high availability in such systems are justified using qualitative measures of the impact on the population affected (Malik 2009). The same line of reasoning applies to information and control systems serving critical infrastructure, such as the electric power grid, railway transportation, water supply, etc.

Citing the data on average downtime costs referred to above, Marcus and Stern (2003) observe that not all losses are easy to quantify, in particular when they are partly composed of opportunity costs, as in the case of brokerage services. However, this is not to denigrate the importance of availability, as they go on to list some indirect costs that can be brought about by system outages: (i) poor customer satisfaction, (ii) bad publicity, (iii) plummeting stock price [while Campbell et al. (2003) suggest that this effect is actually small], (iv) legal liabilities, (v) worsened employee morale, and (vi) an impact to external reputation.

Yet another measure of the importance of systems availability is stakeholder polling. In a recent survey, 178 enterprise IT system executives and practitioners from Sweden and the German-speaking countries were asked to assess future prioritization of various system qualities (in the sense of ISO-9126 (International Organization for Standardization 2003)) in their companies. On a five-point Likert scale, 48.9% of respondents gave availability the highest mark, making it the most highly prioritized system quality in the survey (Franke et al. 2010). A Gartner report, based on surveys conducted in 2007 and 2008, notes consistent findings and concludes that “[t]he overall proportion of mission-critical IT services continues to increase, along with the cost of business downtime” (Scott 2009).

1.1 Outline

The remainder of the paper is structured as follows: Section 2 contrasts the present contribution with some related work. Section 3 introduces some basic concepts from availability theory and elaborates on the scope of the paper in that context. Section 4 introduces Bayesian networks, survey methodology in general, and the particular methods employed for building a Bayesian model from expert assessments. Section 5 is the locus of the main contribution. Here, the results of the expert survey are described, and the resulting Bayesian model for assessment of enterprise IT systems availability is built. An applied example of the usage of the model is presented in Sect. 6. The example also considers availability in the wider architectural context of an enterprise application landscape and

explains how to reconcile the Bayesian model of Sect. 5 with more traditional mathematical reliability analysis. A discussion of the strengths and weaknesses of the contribution then ensues in Sect. 7, followed by some concluding remarks in Sect. 8.

2 Related work

A general and widely cited description of IT systems availability is found in Marcus and Stern (2003), where the authors present an "availability index" describing the relationship between various availability-increasing measures and their costs. The presented availability index gives guidance on improving systems availability, but it is not empirically validated in a structured way. The present contribution partially aims to address this by taking Marcus and Stern as the basis for the survey questions, as discussed in Sect. 4.2.

In Malek et al. (2008), the authors present an approach for analytical service availability assessment, mapping dependencies between low-level ICT infrastructure and high-level services. The mapping, however, does not give a detailed description of the supporting ICT infrastructural elements, nor any weighting of how each element impacts the service availability. In Milanovic et al. (2008), a similar mapping is presented, but here the focus is the impact of ICT infrastructure availability upon business processes, rather than upon availability assessment as such.

An effort to identify factors impacting software reliability is presented in Zhang and Pham (2000). The article includes the identification of 32 factors involved in the software development process, all of which impact software reliability. A ranking based on empirical research from 13 companies working with software development is presented, highlighting the most important factors influencing the software reliability. However, only the software development phase is addressed—how to ensure availability once systems have been taken into service is not mentioned.

The application of Bayesian networks for information system quality analysis is proposed and applied in Lagerström et al. (2009). In this paper, an enterprise architecture evaluation framework for the analysis of information systems modifiability is presented. An expert survey was conducted in order to create a Bayesian model, the details of which are found in Lagerström et al. (2009). The present paper is similar in method, but focuses on availability rather than modifiability.

An interesting application of Bayesian networks for IT service availability modeling is found in Zhang et al. (2009), where system logs are used to build Bayesian networks for availability prediction. This is an important complementary approach to the expert assessments used in the present paper. The main drawback of the system logs approach is the limited availability of such logs—indeed, the results of Zhang et al. are based on a case study of a single enterprise system. A reasonable course for future work is to combine the two approaches by validating the expert assessments of the present paper with case studies in the spirit of Zhang et al.

3 Availability theory

This section starts by briefly outlining some basic concepts from availability theory. Against this background, the main contribution of the paper is then elaborated.

3.1 Availability

The availability of an item, whether a single component or a larger system, is often defined as

$$A = \frac{\text{MTTF}}{\text{MTTF} + \text{MTTR}} \quad (1)$$

where MTTF denotes “Mean Time To Failure” and MTTR “Mean Time To Repair”, respectively. The quotient is easy to interpret as the time that a system is available as a fraction of all time (Rausand and Høyland 2004). A more cautious availability estimate is found by instead using the “Maximum Time To Repair”, corresponding to a worst-case scenario (Marcus and Stern 2003).

The exponential distribution is central in most reliability work and is the distribution most commonly used in applied reliability analysis. The reason for this is its mathematical simplicity and the fact that it gives realistic lifetime models for certain types of items, at least as a first approximation (Rausand and Høyland 2004).

If the time to failure T of an item is exponentially distributed, it has the following well-known probability density function:

$$f(t) = \begin{cases} \lambda \cdot e^{-\lambda t} & \text{for } t > 0, \lambda > 0 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

The corresponding MTTF is simply the reciprocal of the parameter λ :

$$\text{MTTF} = \frac{1}{\lambda} \quad (3)$$

The assumption of exponentially distributed lifetime has two important implications (Rausand and Høyland 2004):

1. A used item is stochastically as good as a new, i.e., there is no reason to replace a working item.
2. When estimating MTTF etc., it is sufficient to collect data on the observed time in operation, and the number of failures. There is no need to keep track of the age of items.

Making the same assumptions about the MTTR, this distribution can similarly be described by a parameter μ :

$$\text{MTTR} = \frac{1}{\mu} \quad (4)$$

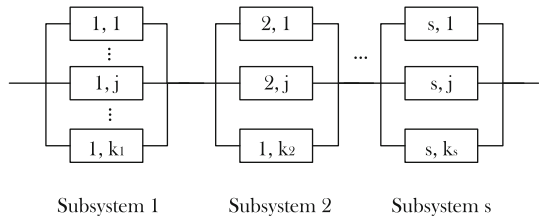
The average availability A_{avg} of a component can now be computed from (1):

$$A_{\text{avg}} = \frac{\mu}{\mu + \lambda} \quad (5)$$

Systems rarely consist of a single component, and oftentimes, these components are connected in parallel. The case where we have s subsystems, where each subsystem i consists of k_i parallel components, and where the subsystems are connected in series is depicted in Fig. 1.

Assuming exponentially distributed MTTF and MTTR, the average availability for this general case can be obtained as follows (Sallak et al. 2006),

Fig. 1 A block diagram showing the general case of parallel series systems, from Sallak et al. (2006)



$$A_{\text{avg}} = \prod_{i=1}^s \left(1 - \left(\frac{\lambda_i}{\lambda_i + \mu_i} \right)^{k_i} \right) = \prod_{i=1}^s \left(1 - (1 - A_i)^{k_i} \right) \tag{6}$$

As seen in (6), this assumes that the failure and repair rates are the same (λ_i and μ_i) for all components in a subsystem i .

The assumption of exponential distributions is more thoroughly discussed in Sect. 7.

3.2 Inter- and intracomponent availability analysis

In Fig. 1, as in the whole theory section so far, the intrinsic availability of each component is taken as a brute fact. In a sense, the components are regarded as black boxes, and all analyses are conducted on a systems level. However, this is clearly a simplification. In reality, the components are not black boxes, but do have internal characteristics—including characteristics that can be affected so as to improve the resulting availability of the system as a whole.

The theory outlined above might be called *intercomponent* availability analysis—calculations based on a number of components being interconnected in a certain fashion. The main contribution of this paper, however, is in the field of what might be called *intracomponent* availability analysis—an attempt to find the internal characteristics of a single component needed to calculate its intrinsic availability. More specifically, the “components” under scrutiny are enterprise IT systems—large components, as it were, but parts of even larger systems, viz., entire enterprise architectures.

Needless to say, a complete availability model for an enterprise architecture—the architecture made up of all the IT systems in an enterprise—needs to account for both inter- and intracomponent availability. Therefore, while the main contribution of the paper is an expert-based Bayesian model for predicting the (intracomponent) availability of enterprise IT systems, Sect. 6 also contains a detailed example of how this contribution can be put to use in the larger context of architectural (intercomponent) availability.

4 Method

This section outlines the methods used in the article. First, a short introduction to Bayesian networks is given, followed by a longer section on methods for eliciting expert knowledge. The method section is then concluded with a synthesis, outlining how to build Bayesian networks based upon expert elicitation.

4.1 Bayesian networks

Friedman et al. (2000) describe a Bayesian network, $B = (G, P)$, as a representation of a joint probability distribution. The first component G is a directed acyclic graph consisting

of vertices, V , and edges, E , i.e., $G = (V, E)$. The vertices denote a domain of random variables X_1, \dots, X_n , also called chance nodes. Each chance node, X_i , may assume a value x_i from the finite domain $\text{Val}(X_i)$. The edges denote causal dependencies between the nodes, i.e., the causal relations between the nodes. Whenever an edge goes from a node X_i to a node X_j , X_i is said to be a causal parent of X_j . The second component P of the network B describes a conditional probability distribution for each chance node, $P(X_i)$, given the set of its causal parents $\text{Pa}(X_i)$ in G . It is now possible to write the joint probability distribution of the domain X_1, \dots, X_n using the chain rule of probability, in the product form:

$$P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i | \text{Pa}(X_i)) \quad (7)$$

In order to specify the joint distribution, the respective conditional probabilities that appear in the product form must be defined. The component P describes the distribution for each possible value x_i of X_i , and $\text{pa}(X_i)$ of $\text{Pa}(X_i)$, where $\text{pa}(X_i)$ is the set of values of $\text{Pa}(x_i)$. These conditional probabilities are represented in matrices, henceforth called Conditional Probability Distributions (CPDs). Using a Bayesian network, it is possible to answer questions such as what is the probability of variable X being in state x_1 given that $Y = y_2$ and $Z = z_1$.

In the general case, the relations between variables described by the conditional probability matrices can be arbitrarily complicated conditional probabilities. The model presented in this paper uses only a single rather simple relation, leaky Noisy-OR, described in Sect. 4.3.

More comprehensive treatments on Bayesian networks can be found in, e.g., Neapolitan (2003), Jensen (2001), Shachter (1988), and Pearl (1988).

4.2 Expert elicitation

Expert elicitation is the process where a person's knowledge and beliefs about one or more uncertain quantities are formulated into a joint probability distribution (Garthwaite et al. 2005), i.e., the act of parameter estimation through the use of domain experts. This approach is generally used when available datasets are sparse in comparison with the number of nodes that need to be parameterized (Johansson and Falkman 2006). Using a well-structured process for expert elicitation is important in order to minimize the bias of the domain expert. A rough outline of such an elicitation process is given by Renooij (2001):

1. Select and motivate the expert
2. Train the expert
3. Structure the questions
4. Elicit and document the expert judgments
5. Verify the results

In the following, we detail how each of these steps was carried out in the present study.

4.2.1 Select and motivate the expert

The selection of respondents in the present survey was based upon academic publications. To identify respondents, searches were performed in major publishing databases (Springer and Elsevier), in professional societies databases such as the IEEE, and in pure indexing

databases such as SCOPUS. The search criteria involved combinations of topic words such as “availability”, “reliability” and “dependability” with research area delimitations such as “information system”, “IT system”, and “corporate IT”. The resulting selections of articles were then manually screened, based on title and abstract (if sufficient) or full content (if necessary) to determine whether the authors should be invited to participate or not. Whenever several co-authors to a single paper were encountered, no distinction was made between them (all or none were invited). The searches were limited in time to the past decade, i.e., only publications from 1999 and onward were selected. In all, 154 authors of journal articles, 298 authors of conference articles, and 11 authors of edited volumes were invited to participate, i.e., a grand total of 463 experts.

As the experts consulted in this study were widely geographically spread, a mail survey was used (Mangione 1995). Another reason to use a mail survey is that the nonrespond bias of mail surveys tends to be directly related to the subject, i.e., chiefly respondents particularly interested in the subject return the questionnaire (Fowler 2002). This effect will be further discussed in Sect. 5. The Internet-based application SurveyMonkey hosted the survey, which was open for 2 weeks, from January 4 to January 15, 2010. As recommended by Blaxter et al. (2006), a reminder was sent to non-responding participants in the middle of the second week to increase the response rate.

As noted in Renooij (2001), it is important to convince the experts that there is no straightforward way to tell a “right” from a “wrong” answer, but that their assessments should rather represent their own knowledge and experience as faithfully as possible. Indeed, the very rationale for selecting this particular research approach is that the subject is difficult to investigate in other ways. In the introduction to the present survey, it was therefore clearly stated that “your particular piece of experience and your corresponding answers are very important to us as we try to build a general model”. Furthermore, each question in the present survey included a self-assessment on the credibility of the answer, enabling anyone feeling uncertain to communicate this. As will be discussed in Sect. 3, this self-assessment also plays an important role in the construction of the Bayesian model.

4.2.2 *Train the expert*

The validity of the study is highly dependent on the respondents’ comprehension of the questions. Therefore, it is often advisable to spend part of the survey to train the expert (Czaja and Blair, 2005), so that she will not only be a subject matter expert but also an expert on giving probability estimates. In the present survey, this was accomplished by the use of an initial tutorial question, where the scope and aim of the question was explained at some length using text and figures. Only after reading this text was the respondent asked to answer the first question, selected for being relatively uncomplicated.

During the training phase, feedback on answers with known correct answers can help experts calibrate their responses (Baecher 1988). However, in the present study, this was not feasible due to lack of indisputable data of sufficient generality.

4.2.3 *Structure the questions*

There are some different approaches to elicitation, direct elicitation being the most obvious and straightforward one. Here, questions are asked along the lines of “What is the probability that variable *A* takes this state given these parent values?”. However, these questions can be hard for domain experts to relate to (Garthwaite et al. 2005), forcing the use of alternative approaches as described for example in Woodberry et al. (2005). In the present

survey, a behaviorally anchored scale (Mangione 1995) was used. The experts were asked to answer “How large a share of currently unavailable enterprise IT systems would you guess would be available if a best practice *factor X* had been present?” (Mutatis mutandis, depending on the appropriate grammar of each factor.) The factors themselves were derived from the availability index presented in Marcus and Stern (2003). Their completeness is thoroughly discussed in Sect. 7.

Including subjective wordings such as “best practice” in a survey has both advantages and disadvantages. On the one hand, the question can be interpreted in several ways, making it more difficult to compare the answers with each other. On the other hand, the answers are to a lower extent limited to assumptions specified in the question (Mangione 1995). In this particular case, the authors of this article do not claim to know the technical details of “best practice” better than the respondents, which is the reason why a subjective wording was used.

A separate question was written for each probability to be assessed. In those cases where potentially ambiguous or unclear terms were used, a short explanatory note was appended to the question to clarify the intended use. To provide the respondent with a birds-eye-view of the survey, a figure illustrating all question categories in order of appearance was continuously displayed throughout the survey.

As noted by Cooke (1991), experts dislike writing numbers for subjective probabilities and prefer to check scales, place an ‘X’ in a box, etc. In the present survey, this was accommodated by using predefined scales in drop-down lists for the alternatives.

To manage respondent uncertainty, each probability estimate was accompanied by the question “What do you think is the probability that your guess is correct?”. The respondent was presented with four fixed alternatives: “12.5% (I have no idea. I just picked a random interval.)”, “50% (I think so.)”, “90% (I am quite sure.)”, and “99% (I am almost completely certain.)”. Wording the alternatives like this, presenting both a qualitative description and a quantitative number to the respondents, minimizes the unwanted implications of converting qualitative factors into quantitative ones *ex post*.

To make the survey questions as clear and lucid as possible, a few test surveys were tried out iteratively, as recommended in Czaja and Blair (2005). The test respondents included both non-domain experts (for general advice on structure and readability) and two of the actual respondents (for more topic-related advice).

4.2.4 Elicit and document the expert judgments

Since elicitation is taxing for the expert, Cooke (1991) recommends that sessions should not exceed 1 h. The present survey being web based, with the possibility for the respondent to take a break or withdraw at his or her discretion, this problem can be considered of marginal importance. However, if a survey is too long or too complex, the response rate of the questionnaire decreases (Blaxter et al. 2006). The level of detail in this study was therefore limited by the expected response rate considered acceptable. A survey of the responses *ex post* indicates that a typical full response required about 20–30 min.

4.2.5 Verify the results

This aspect is addressed as part of the general validity discussion in Sect. 7 below.

Having thus outlined the expert elicitation method, the next section addresses how to build the Bayesian model based upon these data.

4.3 Building Bayesian networks

Bayesian networks are a powerful formalism, but their use requires the specification of conditional probability distributions (CPDs). As the number of variables X_1, \dots, X_n causally affecting a target variable Y grows, fully specifying these distributions becomes increasingly cumbersome. As noted by Onisko et al. (2001), a binary variable with n causal parents requires 2^n independent parameters to exhaustively describe the conditional probabilities. As n grows, 2^n parameters quickly becomes a prohibitive number. Often, however, canonical parameter-based distributions can be used to decrease the modeling effort, yet still give a sufficiently good approximation of the true distribution (Onisko et al. 2001).

The solution described by Onisko et al. is the use of a Noisy-OR gate. Using this formalism, the number of parameters required from expert estimation becomes only n , a significant gain. The underlying assumption is that instead of investigating every combinatorial interaction among the $X_1 \dots X_n$ causal parent variables, their interactions are modeled by a Noisy-OR gate. Furthermore, since Noisy-OR distributions approximate CPDs using fewer parameters, the resulting distributions are in general more reliable, being less susceptible to overfitting (Friedman and Goldszmidt 1999).

The Noisy-OR gate (Pearl 1988; Onisko et al. 2001) is typically used to describe the interaction of n causes X_1, \dots, X_n to an effect Y . In the present article, of course, this effect Y is the unavailability of enterprise IT systems. Two assumptions are made, viz., (i) that each of the causes has a probability p_i of being sufficient for producing Y and (ii) that the ability of each cause X_i , to bring about Y is independent. Mathematically, the following holds:

$$p_i = P(y|\bar{x}_1, \bar{x}_2, \dots, x_i, \dots, \bar{x}_n) \tag{8}$$

where x_i denotes the presence of causal factor X_i and \bar{x}_i its absence. It follows that the probability of y given that a subset $\mathbf{X}_p \subseteq \{X_1, \dots, X_n\}$ of antecedent causes are present can be expressed as:

$$P(y|\mathbf{X}_p) = 1 - \prod_{i: X_i \in \mathbf{X}_p} (1 - p_i) \tag{9}$$

This is a compact specification of the CPD.

A natural extension proposed by Henrion (1989) is the so-called *leaky* Noisy-OR gate. The rationale for the leakage is that models typically do not capture all causes of Y . If some potential X_i have been left out, as is the case in “almost all situations encountered in practice” (Onisko et al. 2001), this shortcoming can be reflected by adding an additional parameter p_0 , the leak probability, such that

$$p_0 = P(y|\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) \tag{10}$$

In words, this reflects the probability that Y will occur spontaneously, in the absence of all the explicitly modeled causes X_1, \dots, X_n . In a leaky Noisy-OR gate, the CPD then becomes

$$P(y|\mathbf{X}_p) = 1 - (1 - p_0) \prod_{i: X_i \in \mathbf{X}_p} \frac{(1 - p_i)}{(1 - p_0)} \tag{11}$$

The aim of the expert elicitation survey can now be stated more explicitly. For each of the factors X_1, \dots, X_n identified in the survey, a probability p_i of X_i being a cause of

enterprise IT system unavailability can be estimated. Depending on the respondent comments regarding causes not listed in the survey, an approximate value of p_0 can also be found, as discussed in Sect. 5

5 Results

5.1 The respondents

Figure 2 displays some data about the respondents who chose to start taking the survey, viz., their affiliation, professional relation to enterprise IT systems availability, their number of years of experience of enterprise IT systems availability, and the correlation between this experience and the confidence expressed in their survey answers.

The data illustrated pertain to two groups: the 96 respondents who began the survey and the 50 who completed it. For obvious reasons, no similar data exist for the grand total group of the 463 experts invited. As seen in Fig. 2, there is no obvious change in the proportion of affiliations from the group that began the survey to the group that completed it. Academia is the most common line of work (67 who began, 33 who completed the

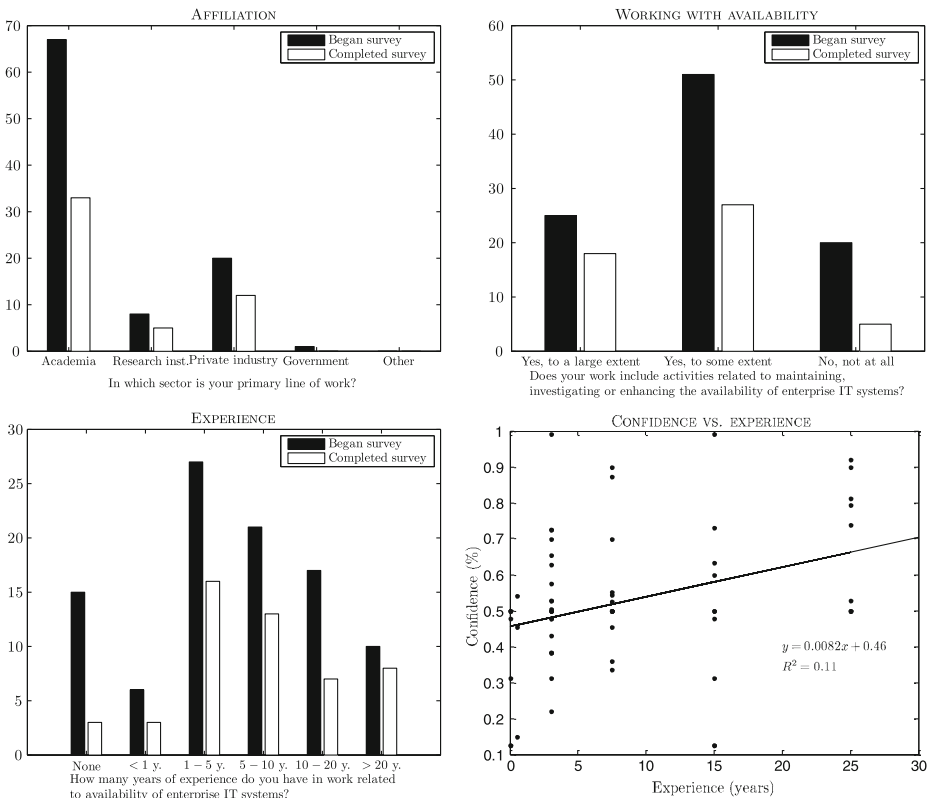


Fig. 2 Data on respondents’ affiliations, working experience with enterprise IT systems availability, years of such experience, and the correlation between years of experience and the average confidence level expressed in the survey answers

survey) followed by private industry (20 who began, 12 who completed the survey) and research institutes (8 who began, 5 who completed the survey). A single government employee began but did not complete.

As for working with questions related to enterprise IT systems availability, the reassuring trend is that those not involved in the field to a large extent dropped out from the survey. Thus, while 72% (18 out of 25) of those who identified themselves as “to a large extent” working with enterprise IT systems availability completed the whole survey and 53% (27 out of 51) of those who identified themselves as “to some extent” working with enterprise IT systems availability completed the whole survey, a mere 25% (5 out of 20) of those “not at all” working with enterprise IT systems availability did so. Based on these figures, it seems reasonable to assume that the quality of the responses collected was improved by this self-selection. A similar trend can be seen when it comes to the number of years of experience. While 80% (8 out of 10) of the respondents with more than 20 years of experience completed the survey, only 20% (3 out of 15) of those with no experience did so. Again, these are reassuring figures, indicating that the quality of the dataset was improved by the self-selection mechanisms described in Fowler (2002).

Furthermore, as is to be expected, experience translates into somewhat greater confidence. As seen in Fig. 2, the respondents with a greater number of years of experience tend to express greater confidence in their answers, even though the variance is large. As explained in the next section, respondent confidence is taken into account when building the Bayesian model, meaning that the opinions of the most experienced respondents will be weighted a bit higher, on average, than those of the less experienced.

5.2 The Bayesian model

The main results of the survey are presented in Table 1. As described in Sect. 4, the respondents not only answered each question but also stated their certainty. In the table, the N specified excludes respondents who “just picked a random interval” and retains only those who were 50, 90, or 99% certain. This corresponds to the answers actually used in building the Bayesian model.

As can be seen, the number of useful answers vary from 36 to 54 over the causal factors, with a mean of 44. While Table 1 does give a good overview of the results, it does not show the self-assessed uncertainties associated with each answer by the respondents. These, however, play a vital role in determining the Noisy-OR probabilities p_1, \dots, p_n associated with each of the causes X_1, \dots, X_n listed in the table.

Table 2 illustrates the distribution of answers over the intervals with certainty gradings for the first causal factor, physical environment. As before, respondents who “just picked a random interval” are excluded. As can be seen, most feel comfortable with the 50% level, “I think so”.

To weight these judgments into a single probability p_i for use in the Noisy-OR model, the number of respondents in an interval j has been multiplied with the certainty $q \in \{0.5, 0.9, 0.99\}$ of their responses (these figures were the alternatives used in the survey). The weighted voting score w_j of interval j is thus defined as

$$w_j = \sum_{k \in K_j} q_k = 0.5 \cdot n_{0.5} + 0.9 \cdot n_{0.9} + 0.99 \cdot n_{0.99} \quad (12)$$

where K_j designates all the respondents who selected interval j , q_k the certainty level of respondent k , and $n_{0.5}$, $n_{0.9}$, and $n_{0.99}$ are the number of respondents within K_j answering

Table 1 Causal factors (based on Marcus and Stern 2003) and strengths as per the respondents' answers

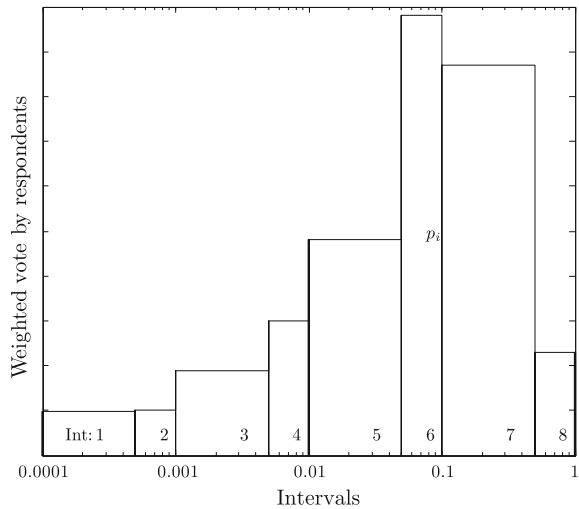
Causal factor <i>X</i>	How large a share of currently unavailable enterprise IT systems would you <i>N</i> guess would be available if a best practice <i>factor X</i> had been present?								
	<0.05%	0.05–0.1%	0.1–0.5%	0.5–1%	1–5%	5–10%	10–50%	>50%	
1. Physical environment	1	2	3	5	8	18	14	3	54
2. Requirements and procurement	1	1	1	2	10	11	16	5	47
3. Operations	1	0	0	1	6	16	19	5	48
4. Change control	0	1	1	2	7	13	17	8	49
5. Technical solution of backup	2	3	1	6	9	15	5	3	44
6. Process solution of backup	0	2	4	6	12	10	9	0	43
7. Data redundancy	0	2	4	6	8	11	9	4	44
8. Storage architecture redundancy	0	3	3	8	15	6	4	1	40
9. Infrastructure redundancy	1	3	3	8	12	7	5	3	42
10. Avoidance of internal application failures	0	4	3	2	7	12	15	2	45
11. Avoidance of external services that fail	2	1	0	5	4	15	12	4	43
12. Network redundancy	0	4	1	4	11	13	9	2	44
13. Avoidance of network failures	0	5	1	5	9	9	11	2	42
14. Physical location	2	0	7	7	11	8	3	2	40
15. Resilient client/server solutions	1	1	2	5	9	9	6	3	36
16. Monitoring of the relevant components	0	2	1	4	13	6	14	3	43

Table 2 The 54 useful respondent answers regarding the physical environment factor, displayed by certainty

Physical environment	50% (I think so.)	90% (I am quite sure.)	99% (I am almost completely certain.)	Weighted vote <i>w</i>
<0.05%	0	0	1	0.99
0.05–0.1%	2	0	0	1
0.1–0.5%	2	1	0	1.9
0.5–1%	4	0	1	2.99
1–5%	6	2	0	4.8
5–10%	16	2	0	9.8
10–50%	10	3	1	8.69
>50%	1	2	0	2.3

Fig. 3 Weighted voting scores for physical environment, logarithmic version. $p_i \approx 0.082$ represents the share of currently unavailable enterprise IT systems that would, in the experts’ opinion, be available if the physical environment had been managed according to best practice

WEIGHTED VOTED DISTRIBUTION FOR PHYSICAL ENVIRONMENT (LOGARITHMIC)



with the different certainty levels. For example, the physical environment weighted voting score for the interval 0.1–0.5% has been calculated simply as $w_j = 2 \cdot 0.5 + 1 \cdot 0.9 + 0 \cdot 0.99 = 1.9$. The weighted voting scores for physical environment are displayed graphically in Fig. 3.

As a consequence of the distribution of the intervals, a linear plot is difficult to read. A logarithmic version is therefore given in Fig. 3. To determine the probability p_i of a less than best practice physical environment factor to cause unavailability in enterprise IT systems, the interval with the highest weighted voting score is selected. To determine the exact location within the interval j , the weighted voting scores of the two adjacent intervals $j - 1$ and $j + 1$ are used, so that

$$p_i = j_{\min} + \frac{w_{j+1}}{w_{j+1} + w_{j-1}}(j_{\max} - j_{\min}) \tag{13}$$

where j_{\min} and j_{\max} designate the start and end points of interval j . In this case, as illustrated in Fig. 3, the interval with the highest weighted voting score is number 6, 5–10%, with $w_6 = 9.8$. The adjacent intervals have the weighted voting scores $w_5 = 8.69$ and $w_7 = 8.69$. These relative scores indicate that the probability p_1 should be located slightly above the midpoint of the 5–10% interval. The calculation yields

$$p_1 = 0.05 + \frac{8.69}{8.69 + 4.8}(0.1 - 0.05) \approx 0.0822$$

The procedure is iterated for each and every causal factor, resulting in probabilities p_1, \dots, p_n as illustrated in Table 3 (rounded to one decimal). Each p_i reflects the share of currently unavailable enterprise IT systems that would, in the experts’ opinion, be available if the *factor* X_i had been managed according to best practice. It might seem counterintuitive that $\sum p_i > 100\%$, but consider a system that went down because of an internal application error, and then did not come up because proper backups did not exist. At an appropriate time after the mishap, it is true that the system would have been available if the application

Table 3 Causal factors with probabilities for Noisy-OR model

Causal factor X_i	p_i (%)
Lack of best practice ...	
1. ... physical environment	8.2
2. ... requirements and procurement	25.2
3. ... operations	23.0
4. ... change control	28.1
5. ... technical solution of backup	7.0
6. ... process solution of backup	3.6
7. ... data redundancy	7.8
8. ... storage architecture redundancy	2.8
9. ... infrastructure redundancy	2.9
10. ... avoidance of internal application failures	16.9
11. ... avoidance of external services that fail	8.7
12. ... network redundancy	7.6
13. ... avoidance of network failures	18.3
14. ... physical location	3.3
15. ... resilient client/server solutions	3.6
16. ... monitoring of the relevant components	26.1

error had been avoided, and also true that the system would have been available if the backups had been better. Thus, the factors need not be mutually exclusive.

As can be seen from Table 3, judging from the respondents' answers, best practice change control is the factor most prone to increase availability of enterprise IT systems, closely followed by best practice component monitoring, and best practice requirements and procurement.

One factor is still missing in order to obtain a complete leaky Noisy-OR model, viz., the leakage p_0 . To obtain an estimate of the leakage, the experts consulted in this survey were asked to comment whether they believed that any important factors contributing to unavailability were left out in the survey. This is discussed in further detail in Sect. 7 Suffice to note here that since no single proposed missing factor was mentioned by more than two experts (out of the 50 respondents) it seems safe to assume that the leakage should be less important than the least important factor considered in the survey. As seen in Table 3, the smallest p_i belongs to storage architecture redundancy at 2.8%. A leakage $p_0 = 1\%$ therefore seems reasonable and will be used throughout the remainder of the article.

5.3 Rescaling for case-based assessment

The questions answered by the respondents were explicitly concerned with increasing availability of unavailable systems (“How large a share of currently unavailable enterprise IT systems would you guess would be available if a best practice *factor* X had been present?”). The leaky Noisy-OR model therefore explains enterprise IT systems *un*-availability (Y), employing the parameters X_1, \dots, X_n describing the lack of best practices, and the model is built in the domain of unavailable enterprise IT systems. However, a more practical typical concern is the availability of an entire park of systems, with a known prior availability baseline. The Bayesian model therefore needs to be

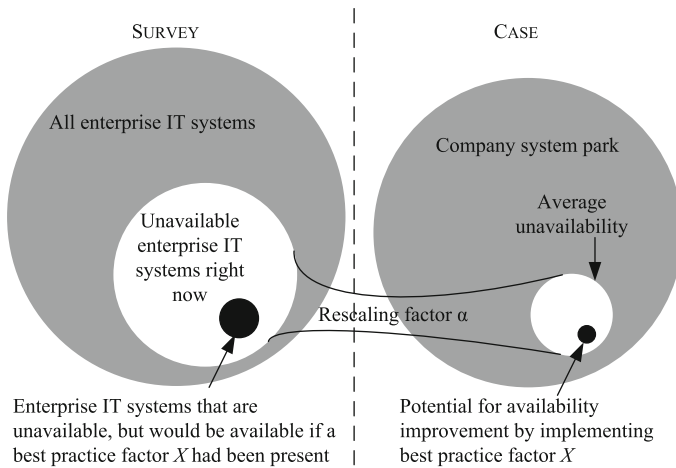


Fig. 4 Venn diagrams schematically depicting the relation between the survey and an application case illustrating the need for the rescaling factor α

rescaled from the set of unavailable enterprise IT systems to the whole set—available and unavailable alike—of enterprise IT systems. Figure 4 (slightly adapted from the survey) illustrates the issue.

Another way to express the issue is that the unscaled Noisy-OR model reflects the *potential for improvement*, by addressing only unavailability.

The most straightforward way to rescale the model, in order to answer how a system’s availability can be improved by applying best practice solutions, is to apply a rescaling factor α to all p_i , including the leakage p_0 . It could be argued that a single α should not be applied to all factors alike, but in the absence of good reasons to treat them separately, this is surely the simplest and best warranted solution. It follows from (11) that

$$A(\mathbf{X}_p) = 1 - P(y|\mathbf{X}_p) = (1 - \alpha p_0) \prod_{i: X_i \in \mathbf{X}_p} \frac{(1 - \alpha p_i)}{(1 - \alpha p_0)} \tag{14}$$

where $A(\mathbf{X}_p)$ is the availability of a given system lacking the best practice factors listed in the vector \mathbf{X}_p .

5.4 Survey of research interests

The concluding part of the survey asked the respondents to select some research areas of future interest. Figure 5 summarizes the results. It was possible to select several alternatives and also to add additional areas. One respondent added a new area, viz., automatic computing.

6 Availability analysis examples

This section is tripartite. First, an illustration is given of how the leaky Noisy-OR model presented in the previous section can be used for actual assessment of an enterprise IT

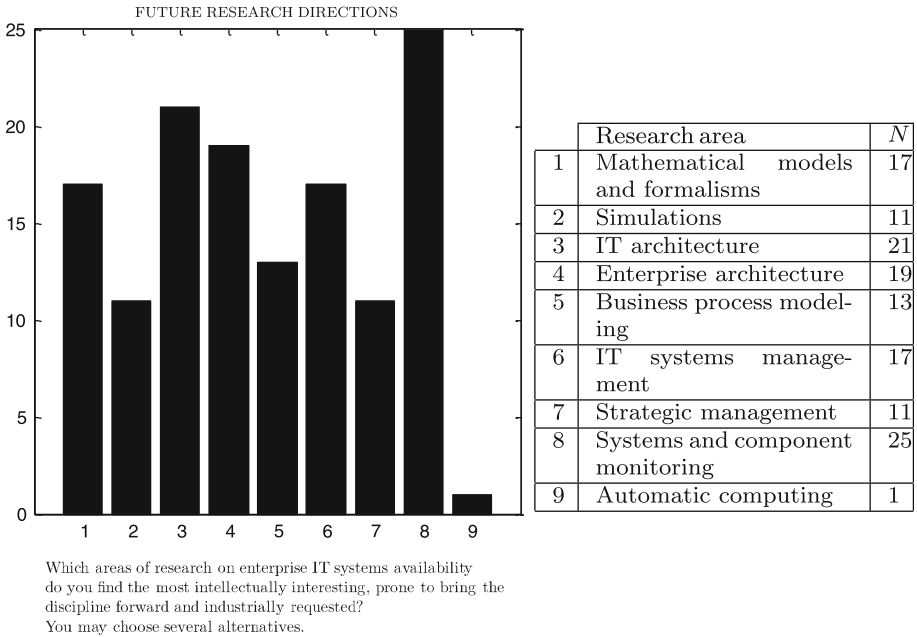


Fig. 5 Data on future research on enterprise IT systems availability

system, and how it can thus guide decision-making with an impact on availability. This corresponds to availability analysis on the level of a single enterprise IT system, or the intracomponent availability of enterprise IT systems, to use the phrase coined in Sect. 3. Second, this is contrasted with the traditional architecture level, intercomponent, analysis. Third, a synthesis example is given, illustrating how to address the combination of intra- and intercomponent measures to ensure high overall availability on the enterprise architecture level in a comprehensive manner.

6.1 Enterprise IT system level (intracomponent) availability analysis

To give an example, assuming that the fictitious system *Saurischia* has a current availability of 99.8% and that best practice was only applied in the cases of data redundancy (X_7), storage architecture redundancy (X_8), and infrastructure redundancy (X_9), (14) becomes

$$99.8\% = (1 - \alpha p_0) \prod_{i=1}^6 \frac{(1 - \alpha p_i)}{(1 - \alpha p_0)} \prod_{i=10}^{16} \frac{(1 - \alpha p_i)}{(1 - \alpha p_0)}$$

Solving for α (analytically this is cumbersome due to all the binomial coefficients, but numerically it is easy) yields a rescaling factor $\alpha \approx 0.00117223$.

Continuing the example of the fictitious *Saurischia* system, it is natural to ask how to improve its availability. Here, the model can give precious guidance. Assuming that the 13 factors not currently reaching best practice are at the decision-maker’s disposal, their respective impacts can easily be analyzed and compared to the prior baseline of 99.8%.

Figure 6 illustrates the predicted impact of each of these 13 factors taken by themselves. As can be seen, factor 4 (change control), factor 16 (monitoring of the relevant components), and factor 2 (requirements and procurement) are the most promising candidates for availability improvement of the *Saurischia* system. It might be objected that this could have been read straight off Table 3—finding the most promising candidates requires only an ordinal ranking. However, a key strength of the Bayesian method is the possibility to investigate the impact of getting several factors up to best practice level at the same time [as seen from (14)]. To evaluate these interactions of several factors, Table 3 is not sufficient by itself, but the full leaky Noisy-OR model is needed. Another strength of the full model is that the expected cost of unavailability (e.g., from IBM Global Services (1998) or using a method like the one proposed by Scott (2009)) can be compared to the estimated costs for getting the various factors up to best practice level.

Extending Bayesian networks in this way with costs and utilities amounts to creating an influence diagram (Shachter 1988). A good graphical tool that can be employed to handle influence diagrams, and standard Bayesian networks, is GeNIe (Druzdzel 1999).

6.2 Enterprise architecture level (intercomponent) availability analysis

In a general availability assurance situation, an obvious alternative to improving the factors so far discussed is to simply duplicate systems, introducing redundancy on a system level in the architecture. For example, in the case of the *Saurischia* system above, Fig. 6 should be complemented with Table 4.

It might be objected that this is a naïve interpretation of redundancy. Only if system failures are independent will availability be increased—for software with identical failure causes, availability might not be increased at all. In practice, though, redundancy is used to increase the availability of enterprise information systems. One method to bring about the required statistical independence of software failures upon each other is N-version programming (Chen and Avizienis 1978), which mimics N-modular redundancy on the hardware side. In the general case, both hardware and software (e.g., N-versioning)

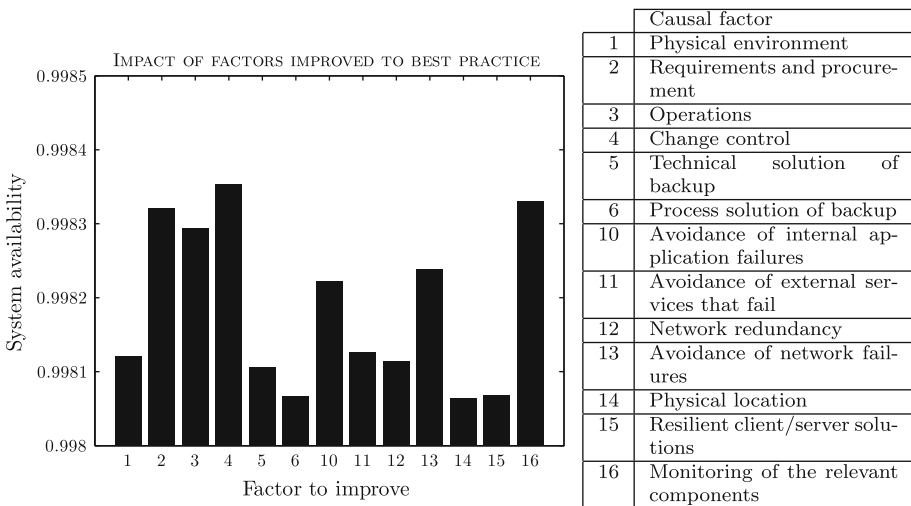


Fig. 6 Prediction of how improvements of factors to the best practice-level would impact the availability of the example *Saurischia* system

Table 4 Availability improvements from multiple redundant *Saurischia* systems with perfect switching

Redundant systems	1	2	3
Availability	99.8%	99.9996%	99.999992%
Average annual downtime	17.52 h	2.1024 min	0.252288 s

redundancy is required to approach the upper bound on availability improvements through redundancy illustrated in Table 4. The optimal use of N-version programming is a research area of its own (Ashrafi et al, 2002), and outside the scope of the present paper. An additional complication is the implicit assumption of perfect switching between redundant systems. A more thorough discussion of models assuming perfect and imperfect switching, respectively, can be found in Rausand and Høyland (2004).

Bearing these complications in mind, Table 4 nevertheless hints that running two redundant *Saurischia* systems has the potential (in the upper bound-sense) to deliver better availability than can be achieved by adjusting any single causal factor listed in Table 4. Indeed, even if *all* the causal factors are set to best practice level, the 99.9988278% availability thus predicted by the Bayesian expert model (corresponding to just above 6 min of annual downtime) is slightly below the upper bound for two redundant systems. Of course, these are ballpark numbers with not so many significant figures.

This example, though simplified, illustrates a general trade-off situation when it comes to ensuring availability. Given a fixed budget, it has to be divided between (i) the *intracomponent* factors of Table 3 and (ii) the *intercomponent* (architectural) redundancy measures that can be introduced when looking at the components as fixed, black box-like, building blocks. The resulting availability for each choice can be approximated by first assessing the impact of the intracomponent factors according to (14), then assessing the overall architectural intercomponent impact according to (6).

6.3 Combined analysis of intra- and intercomponent factors

To explore the notion of a trade-off further, consider the simple case of a fictitious enterprise architecture where three IT systems are connected in series. First, information is processed in the *Astrodon* system, then in the *Baryonyx* system, and finally in the *Cardiodon* system. Each of the systems has an initial availability (97, 99.5 and 98%, respectively) leading to an overall systems availability of 94.585% in the basic, serially connected, case. To increase this availability, a decision-maker can either improve the intracomponent availability (as predicted by the leaky Noisy-OR model) or improve the intercomponent, architectural availability [as predicted by (6)], or combine these two approaches as she sees fit. Figure 7 conceptually illustrates this trade-off. The costs used in the example have been arbitrarily picked to illustrate the general principle.

The limiting factor is the budget available. In Fig. 7, the budget is set to the unity cost c , and the constraint is illustrated as the straight line defined by $x + y = c$. Any feasible solution must be placed below this budget line. The basic case is illustrated by the a dot in the figure. It is situated on the y -axis since no money have been spent on intracomponent improvements (i.e., the component cost on the x -axis is 0). The vertical location on the y -axis reflects the intercomponent costs of the bare-bones, three component basic architecture, nothing else.

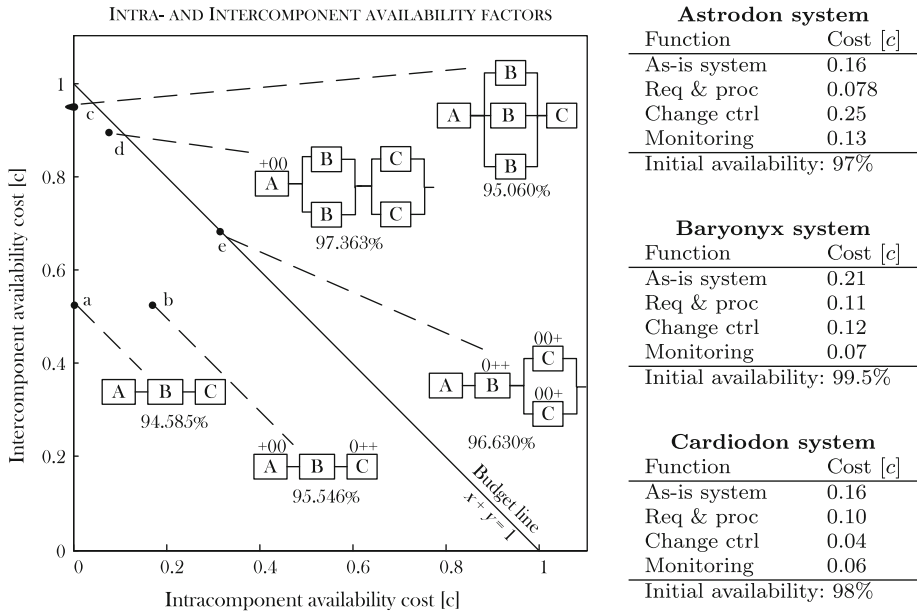


Fig. 7 The trade-off between spending on intra- and intercomponent measures for high availability, with cases labeled *a–e* illustrated

An improved case is illustrated by the *b* dot in the figure. Here, intracomponent spending is nonzero, as system A has been improved with regard to requirements and procurement and system C has been improved with regard to change control and component monitoring. (For simplicity, these are the only three component factors considered in the example. In the figure, a three three-place code is used to denote whether these factors reach best practice level (+) or not (0). Hence, system A is labeled +00.) As a result, the systems availability has increased to 95.546%.

The *b* case can be contrasted with *c*, where intracomponent spending is still zero, but significant resources have been spent on the intercomponent axis to achieve redundancy (e.g., through N-version programming). As a result of three redundant B systems, availability is 95.060%—up from the basic case, but not as good as *b*.

d is the first case where both intra- and intercomponent improvements have been introduced. Here, systems B and C both run with two redundant systems, whereas A has been improved to reach the requirements and procurement best practice level. As a result, availability is 97.363%.

The *e* case is another example of using both intra- and intercomponent improvements. This solution leaves system A as-is, improves the change control and component monitoring of system B, and introduces two redundant C systems with improved component monitoring. While this solution manages to spend exactly the allotted *c* amount of money (the solution is situated on the budget line), it achieves no more than 96.630% availability—up from the basic case, but not as good as *d*.

To summarize, Fig. 7 illustrates (i) that the combination of intracomponent and intercomponent (architecture level) measures to ensure high availability is not a trivial matter and (ii) how this matter can be addressed in a comprehensive manner. Together, these two

aspects of enterprise architecture availability—epitomized by the leaky Noisy-OR model and (6)—constitute a complete framework, albeit approximate, for enterprise architecture level availability assessment.

7 Discussion

This section aims to critically discuss some strengths and weaknesses of the contribution. First, some underlying assumptions are scrutinized, followed by a more general discussion on validity.

7.1 The Noisy-OR assumptions

As noted in Sect. 4, proper use of the Noisy-OR gate makes two assumptions regarding the structure of the interaction of causes (X_1, \dots, X_n) and effect (Y) (Onisko et al. 2001; Pearl, 1988). These are (i) that each of the causes has a probability p_i of being sufficient for producing Y and (ii) that the ability of each cause X_1, \dots, X_n to bring about Y is independent. In the present study, Y is unavailability of enterprise IT systems, and X_1, \dots, X_n are causes of such unavailability.

Arguing for (i) in this case is straightforward. Indeed, it is almost always assumed that failing non-redundant components of complex systems can cause malfunctions by themselves. However, these faults are not always deterministic—e.g., a non-best practice requirements and procurement process will not infallibly lead to unavailability, but will do so with a certain probability p . Arguing for (ii) is harder. In many cases, it is reasonable to assume that factors are independent, but this is not always the case. Backup systems, for example, only come into play when a system has failed and it is time to restore it. Therefore, the impact of factors such as *technical solution of backup* and the *process solution of backup* depends to some extent on other factors. In general terms, the distinction between proactive and reactive factors indicates that (ii) is an approximation that does not hold in all circumstances. In the full model, such dependencies could be modeled by rescaling different factors p_i with different factors α_i accounting for interactions. However, to accurately reflect these phenomena, more empirical data are needed. To conclude, while the assumptions required for the Noisy-OR model are reasonable as a first approximation, the model should certainly be open to further refinement. By and large, such refinement is a matter of empirical investigation, where availability data from enterprise IT systems can be analyzed and statistically checked for independence. It should be noted, however, that there is no need to refashion the entire Bayesian model should some cause variables X_1, \dots, X_n turn out to be dependent. The bulk of the causes can remain modeled in a Noisy-OR relation to each other, while a select few can be modeled using different CPDs.

7.2 The assumption of exponential distributions

The architectural part of the analysis framework, described in Sect. 2, depends on the assumption of exponentially distributed mean times to failure (MTTF) and repair (MTTR). Two arguments speak in favor of these assumptions:

First, the exponential distribution is central in most reliability work and is the most commonly used distribution in applied reliability analysis (Rausand and Høyland 2004). Many works on software reliability take the exponential distribution as a starting point or a

first approximation, before describing more elaborate models such as the Jelinski-Moranda, Musa, or Littlewood models (Pham 2000; Fenton and Pfleeger 1997; Musa 1999; Laird and Brennan 2006).

Second, the exponential distribution is mathematically simple. This is not only an argument from convenience. The fact that the exponential distribution has only a single parameter (that can be estimated by collecting data on the observed time in operation, and the number of failures) makes it suitable for ballpark calculations about largely unknown systems. As illustrated in Sect. 6, the method presented in the present paper is intended to be used at an early planning stage to evaluate different to-be architecture scenarios. If used for this purpose and at this stage, the simple single-parameter exponential distribution is surely more warranted than any distribution that requires additional assumptions (parameters). In this sense, the use of the exponential distribution is an instance of Occam's razor. (An analogy to significant figures is illuminating. It is not meaningful to perform calculations with a greater accuracy than that of the original data. Similarly, it is not meaningful to perform—or prescribe as part of a method—calculations that require multiparameter distributions that cannot be reliably parametrized in practice.)

We conclude that while the assumption of exponential distributions cannot a priori be proven correct, no other assumption is *better* justified. In the absence of such justification, the assumption of exponential distributions is the least controversial modeling choice.

7.3 Contrasting academia and private industry

Even though all the respondents were selected based upon academic publications as described in Sect. 5, it is interesting to see whether there are differences between the academics (respondents affiliated with academia and research institutes) and practitioners (respondents affiliated with private industry). Table 5 extends Table 3 by also listing the

Table 5 Causal factors with probabilities for Noisy-OR model for different groups of respondents

Causal factor X_i	p_i All (%)	p_i Academics (%)	p_i Practitioners (%)
Lack of best practice ...			
1. ... physical environment	8.2	7.9	10.0
2. ... requirements and procurement	25.2	24.1	30.0
3. ... operations	23.0	20.7	33.3
4. ... change control	28.1	26.5	32.4
5. ... technical solution of backup	7.0	7.2	5.0
6. ... process solution of backup	3.6	3.5	10.0
7. ... data redundancy	7.8	7.5	30.0
8. ... storage architecture redundancy	2.8	3.5	0.9
9. ... infrastructure redundancy	2.9	3.0	0.8
10. ... avoidance of internal application failures	16.9	15.9	20.0
11. ... avoidance of external services that fail	8.7	8.6	8.8
12. ... network redundancy	7.6	7.8	1.0
13. ... avoidance of network failures	18.3	19.6	3.7
14. ... physical location	3.3	3.7	0.8
15. ... resilient client/server solutions	3.6	3.8	10.0
16. ... monitoring of the relevant components	26.1	25.3	30.0

causal factor probabilities as calculated based exclusively upon academic and practitioner responses, respectively.

Overall, the results from the different respondent groups are rather similar. A few exceptions can be seen, most notably the factors Operations and Data redundancy that are given more importance by practitioners and Avoidance of network failures that is given less. While these discrepancies might be taken as clues for further investigation, they are not in themselves substantial enough to warrant modification of the Bayesian model developed in Sect. 5. As illustrated in Fig. 2, the number of practitioners is small (20 who started, 12 who finished the survey) compared to the number of academics. This alone is an important source of variability—such a small dataset is much more prone to exhibit outliers.

7.4 Validity of the model

So far, the validity discussion has mainly focused on the numbers. As discussed in Sect. 2, the respondents were carefully selected based on scientific merit, the uncertainty of their answers was taken into account, and self-selection ensured that the 50 final respondents were among the most qualified. However, the discussion of leakage also leads to a discussion of the *completeness* of the model. First and foremost, a strong argument for the completeness of the model is that it is based on the widely cited work by Marcus and Stern (2003). However, converting the qualitative theory of Marcus and Stern into questions suitable for building a quantitative Bayesian model unavoidably introduces distortions. Two questions thus need to be addressed: (i) are there causes missing that should be added? and (ii) are there superfluous causes that should be removed? Together, these questions determine whether the model contains all relevant causal factors.

(i) was explicitly addressed in the survey. The question “Do you believe that important aspects of enterprise IT systems availability have been left out in the survey? If so, please describe the areas missing.” received 18 answers, i.e., 32 of the respondents did not find any aspects missing important enough to warrant an answer. Out of these 18 answers, two were in the negative, i.e., confirming the completeness of the model. Another two addressed methodology issues that will be discussed below, but did not constitute

Table 6 Missing factors identified by 14 respondents

Causal factor	<i>N</i>
Security attacks	1
Automatic computing	1
Software quality	1
Disaster management	1
Human factors	2
Methodologies such as ISO-standards	1
Evolution/maintenance	1
Data availability and data model	1
Performance monitoring	1
Cost/benefit issues	2
Virtualization	1
Load management	1
Infrastructure should be more specific	1
Enterprise organization	1
Relationship between SLA and desired uptime	1

Some respondents identified more than one missing factor

suggestions for additional causal factors relevant to enterprise IT systems availability. The remaining 14 replies are summarized in Table 6.

The two methodology questions raised were (a) that subjective perception of availability may differ from objective measures and (b) that practitioners rather than academics should have been selected as respondents. These are both relevant points, but they are also complementary in an interesting way. Asking practitioners to give estimates would run a higher risk of being influenced by subjective perceptions (since a user or administrator dealing with systems on an everyday basis has the opportunity to develop a subjective perception, as opposed to a scientist collecting data or building models in a fashion more disconnected from daily system usage). Conversely, asking published scientists limits the risk of subjective perceptions based only on ones own systems (since scientific publication requires a certain generality, and a careful discussion of validity), but at the same time runs the risk of missing valuable “down-to-earth” insights from the practitioner community. There seems, thus, to exist an inherent methodological trade-off between (a) and (b), and in light of this, receiving one comment on each is not a bad result. The details on the selection of survey participants were more thoroughly discussed in Sect. 2.

As seen in Table 6, no single potentially missing causal factor was identified by more than 4% of survey respondents (2 out of 50). Most were identified by just a single respondent. Since there is no strong agreement among the 50 experts on which causal factors are missing, we conclude that the model contains an appropriate set of causal factors causing enterprise IT system unavailability.

As for concern (ii)—superfluous causes of unavailability in the model that should be removed—it was not explicitly addressed in the survey in the sense that any particular question was devoted to it. However, every question on causes implicitly addresses the issue, as the respondents could always say that a very minute fraction ($<0.05\%$) of currently unavailable enterprise IT systems would be available if a best practice *factor X* had been present. It should be noted, of course, that this is not an unambiguous measure of the superfluousness of a cause. A causal factor that is both very important to availability and very well managed in the real world does not offer the kind of potential for improvement that the question looks for. However, as discussed in Sect. 5, it does offer a measure of the practical relevance of the causal factor. A causal factor with a large potential for improvement is, *ceteris paribus*, more relevant to a practitioner than a causal factor with a small potential for improvement.

We conclude that the model, by its very nature, appropriately deals with very weak and potentially superfluous causal factors.

8 Conclusions

The contribution of the present paper is threefold. First, the results from an academic expert survey on the causes of unavailability of enterprise IT systems are presented. Second, these results are used to build a Bayesian decision support model for assessment of enterprise IT systems availability. Third, this model is integrated with a standard model for assessing availability on an architecture level, thus forming an assessment framework addressing both component and architecture level measures to ensure high availability. Examples are presented to illustrate how the framework can be used by practitioners aiming to ensure systems availability.

A natural continuation of the present line of research is to validate the results with case studies of actual enterprise IT systems. Empirical data from such investigations could be

used both quantitatively to calibrate the numbers in the Bayesian model and qualitatively to restructure the Bayesian network if the leaky Noisy-OR assumptions prove unsuitable for some variables.

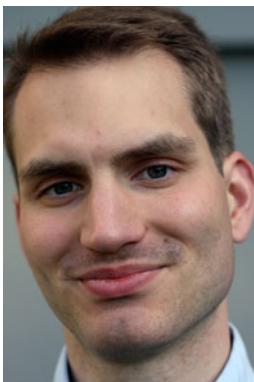
Acknowledgments The authors are grateful to those colleagues and respondents who helped improving the survey before it was sent out, as well as to all the researchers who responded when invited. A special thanks is due to Lars Nordström for his contribution and support throughout the project. Three anonymous reviewers provided useful comments upon the paper. A previous version of this paper was presented at the Workshop on Software Quality and Maintainability (SQM 2010)—Franke et al. (2010)—and the authors would also like to thank the anonymous reviewers of that version for their comments.

References

- Ashrafi, N., Berman, O., & Cutler, M. (2002). Optimal design of large software-systems using N-version programming. *IEEE Transactions on Reliability*, 43(2), 344–350.
- Askåker, J., & Kulle, M. (2008, June 4). Miljardaffärer gick förlorade. *Dagens Industri*, pp. 6–7 (in Swedish).
- Baecher, G. (1988). *Judgemental probability in geotechnical risk assessment*. Technical report, The Office of the Chief, US Army Corps of Engineers.
- Blaxter, L., Hughes, C., & Tight, M. (2006). *How to research*. Buckingham: Open University Press.
- Campbell, K., Gordon, L. A., Loeb, M. P., & Zhou, L. (2003). The economic cost of publicly announced information security breaches: Empirical evidence from the stock market. *Journal of Computer Security*, 11(3), 431–448.
- Chen, L., & Avizienis A. (1978). N-version programming: A fault-tolerance approach to reliability of software operation. In *Proceedings of the 8th IEEE international symposium on fault-tolerant computing (FTCS-8)*, pp. 3–9.
- Cooke, R. M. (1991). *Experts in uncertainty. Opinion and subjective probability in science*. Oxford: Oxford University Press.
- Czaja, R., & Blair, J. (2005). *Designing surveys*. Beverly Hills: Sage Publications Inc.
- Druzdel, M. J. (1999). Genie: A development environment for graphical decision-analytic models. In *Proceedings of the 1999 annual symposium of the American medical informatics association (AMIA-1999)*, p. 1206, more information available on <http://genie.sis.pitt.edu/about.htm>.
- Fenton, N. E., Pfeeger, S. L. (1997). *Software Metrics* (2nd ed.). Boston, MA: PWS Publishing Company.
- Fowler, F. J. J. (2002). *Survey research methods*. Beverly Hills: Sage Publications Inc.
- Franke, U., Johnson, P., König, J., & von Würtemberg, L. M. (2010a). Availability of enterprise IT systems—an expert-based Bayesian model. In *Proceedings of the fourth international workshop on software quality and maintainability (SQM 2010)*, Madrid.
- Franke, U., Lagerström, R., Ekstedt, M., Saat, J., & Winter, R. (2010b). Trends in enterprise architecture practice—A survey. In *Proceedings of the 5th trends in enterprise architecture research (TEAR2010) workshop*.
- Friedman, N., & Goldszmidt, M. (1999). Learning Bayesian networks with local structure. In M. I. Jordan (Eds.), *Graphical models* (pp. 421–459). Cambridge, MA: MIT Press.
- Friedman, N., Linial, M., & Nachman, I. (2000). Using bayesian networks to analyze expression data. *Journal of Computational Biology*, 7, 601–620.
- Garthwaite, P. H., Kadane, J. B., & O’Hagan, A. (2005). Statistical methods for eliciting probability distributions. *Journal of the American Statistical Association* 100, 680–701. <http://ideas.repec.org/a/bes/jnlasa/v100y2005p680-701.htm>.
- Henion, M. (1989). Some practical issues in constructing belief networks. In L. Kanal, T. Levitt, & J. Lemmer (Eds.), *Uncertainty in artificial intelligence 3* (pp. 161–173). North Holland: Elsevier Science Publishers B.V.
- IBM Global Services. (1998). *Improving systems availability*. Technical report, IBM Global Services.
- International Organization for Standardization. (2003). Software engineering—Product quality—Part 2: External metrics. International standard ISO/IEC TR 9126-2:2003(E), International Organization for Standardization.
- Jensen, F. V. (2001). *Bayesian networks and decision graphs*. Secaucus, NJ: Springer.
- Johansson, F., & Falkman, G. (2006). Implementation and integration of a bayesian network for prediction of tactical intention into a ground target simulator. In *2006 9th international conference on information fusion*, pp. 1–7. doi:10.1109/ICIF.2006.301605.

- Lagerström, R., Franke, U., Johnson, P., & Ullberg, J. (2009a). A method for creating enterprise architecture metamodels—Applied to systems modifiability analysis. *International Journal of Computer Science & Applications*, *VI*, 89–120.
- Lagerström, R., Johnson, P., Höök, D., & König, J. (2009b). Software change project cost estimation—A Bayesian network and a method for expert elicitation. In *Third international workshop on software quality and maintainability (SQM 2009)*.
- Laird, L., & Brennan C. (2006). *Software measurement and estimation: A practical approach*. New York: IEEE Computer Society/Wiley.
- Malek, M., Milic, B., & Milanovic, N. (2008). Analytical availability assessment of it services. In *ISAS*, pp. 207–224.
- Malik, B. (2009). *Q&A: How much does an hour of downtime cost?*. Technical report, Gartner, Inc.
- Mangione, T. W. (1995). *Mail surveys improving the quality*. Beverly Hills: Sage Publications Inc.
- Marcus, E., & Stern, H. (2003). *Blueprints for high availability* (2nd ed). Indianapolis, IN: Wiley.
- Milanovic, N., Milic, B., & Malek, M. (2008). Modeling business process availability. In *SERVICES '08: Proceedings of the 2008 IEEE congress on services—Part I* (pp. 315–321). Washington, DC: IEEE Computer Society. doi:[10.1109/SERVICES-1.2008](https://doi.org/10.1109/SERVICES-1.2008).
- Musa, J. (1999). *Software reliability engineering: More reliable software, faster development and testing*. New York: McGraw-Hill.
- Neapolitan, R. E. (2003). *Learning Bayesian networks*. Upper Saddle River, NJ: Prentice-Hall Inc.
- Onisko, A., Druzdzel, M. J., & Wasyluk, H. (2001). Learning Bayesian network parameters from small data sets: Application of noisy-or gates. *International Journal of Approximate Reasoning*, *27*(2), 165–182. doi:[10.1016/S0888-613X\(01\)00039-1](https://doi.org/10.1016/S0888-613X(01)00039-1).
- Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. San Francisco, CA: Morgan Kaufmann Publishers Inc.
- Pham, H. (2000). *Software reliability*. Singapore: Springer.
- Rausand, M., & Høyland, A. (2004). *System reliability theory: Models, statistical methods, and applications* (2nd ed.). Hoboken, NJ: Wiley. <http://www.ntnu.no/ross/srt>.
- Renooij, S. (2001). *Qualitative approaches to quantifying probabilistic networks*. PhD thesis, Utrecht University, Utrecht, The Netherlands.
- Sallak, M., Simon, C., & Aubry J. F. (2006). Optimal design of safety instrumented systems. In *Workshop on advanced control and diagnosis, ACD*, Nancy.
- Scott, D. (2009). *How to assess your IT service availability levels*. Technical report, Gartner, Inc.
- Shachter, R. D. (1988). Probabilistic inference and influence diagrams. *Operations Research*, *36*(4), 589–604.
- Woodberry, O., Nicholson, A. E., Korb, K. B., & Pollino, C. (2005). Parameterising bayesian networks. In *AI 2004: Advances in artificial intelligence* (pp. 1101–1107). Berlin: Springer.
- Zhang, R., Cope, E., Heusler, L., & Cheng, F. (2009). A Bayesian network approach to modeling IT service availability using system logs. In *Workshop on the analysis of system logs*.
- Zhang, X., & Pham, H. (2000) An analysis of factors affecting software reliability. *Journal of Systems and Software* *50*(1), 43–56. [10.1016/S0164-1212\(99\)00075-8](https://doi.org/10.1016/S0164-1212(99)00075-8).

Author Biographies



Ulrik Franke received his MSc in Engineering Physics from the Royal Institute of Technology (KTH) in Stockholm in 2007, and is now pursuing a PhD in Industrial Information and Control Systems. His primary research interest is Enterprise Architecture (EA), a method for rational decision making in the realm of enterprise information systems, focusing on modeling for decision support and the incorporation of formal methods into the EA paradigm. His thesis work revolves around the availability of enterprise information systems—how to predict it and improve it.



Pontus Johnson is Professor and Head of the Department of Industrial Information and Control Systems at the Royal Institute of Technology (KTH) in Stockholm, Sweden. At the department, research focuses on the analysis of architectural models of information systems in their context. He is secretary of the IFIP Working Group 5.8 on Enterprise Interoperability, technical coordinator of the FP7 Viking project, organizer, program committee member and associate editor of several international conferences, workshops and journals. He received his MSc from the Lund Institute of Technology in 1997 and his PhD and Docent titles from the Royal Institute of Technology in 2002 and 2007. He was appointed professor in 2009.



Johan König received his MSc in Electrical Engineering from the Royal Institute of Technology (KTH) in Stockholm in 2008. He is currently a PhD student at the department of Industrial Information and Control systems at KTH. His research interests are quality analysis of active distribution grids from an ICT perspective.



Liv Mareks von Würtemberg holds a MSc in Engineering and of Education from the Royal Institute of Technology (KTH) and Stockholm University. She joined the department of Industrial Information and Control Systems on the Royal Institute of Technology for PhD studies in November 2009. Her research interests are project management, Lean Product Development, Design for Six Sigma and the implementation of development frameworks in the industry.