Information and Software Technology 45 (2003) 51–60

INFORMATION AND SOFTWARE TECHNOLOGY

www.elsevier.com/locate/infsof

Short Communication

On the use of Bayesian belief networks for the prediction of software productivity

I. Stamelos[a,*], L. Angelis[a], P. Dimou[a], E. Sakellaris[b]

[a]*Department of Informatics, Aristotle University of Thessaloniki, 54006 Thessaloniki, Greece*
[b]*Singular International, 10A Dodekanisou Str., 54626 Thessaloniki, Greece*

Received 12 April 2001; revised 22 July 2002; accepted 22 July 2002

Abstract

In spite of numerous methods proposed, software cost estimation remains an open issue and in most situations expert judgment is still being used. In this paper, we propose the use of Bayesian belief networks (BBNs), already applied in other software engineering areas, to support expert judgment in software cost estimation. We briefly present BBNs and their advantages for expert opinion support and we propose their use for productivity estimation. We illustrate our approach by giving two examples, one based on the COCOMO81 cost factors and a second one, dealing with productivity in ERP system localization.

Keywords: Cost estimation; Bayesian belief network; Expert judgment; COCOMO

1. Introduction and related work

Software cost estimation has been an open problem in software engineering for many years. A number of estimation models have been proposed, based on expert judgment, estimation by analogy or algorithmic cost models. These methods are not always successful, due to the diversity of cost factors, their unclear contribution to productivity, the high degree of uncertainty and the lack of information in the early stages of software development.

Although it is a widely held belief that expert judgment is an unreliable approach for estimation, this method is one of the most commonly used today. In certain cases (new technology, no historical project data, etc.) expert judgment is the only feasible estimation approach. Additionally, there is research work that suggests that expert judgment should be used in parallel with other estimation models for better results [6].

Several tools and methods have been proposed in order to support expert judgment. In the last years, Bayesian belief networks (BBNs) have appeared among them. Briefly speaking, BBNs are cause–effect graphs based on Bayesian inference, capable of modeling uncertainty. BBNs, as a tool for decision support, have been proposed and deployed in various scientific domains, such as in medicine [9]. In the field of software engineering, BBNs have been introduced in modeling uncertainties in software testing, in defect density prediction and other areas.

More specifically, in Ref. [15] the uncertainties in the testing of software projects are discussed and the use of BBNs to model these uncertainties is proposed. In Ref. [4] a thorough critique of the existing software defect estimation methods is presented and the use of BBNs is proposed as an alternative method for the estimation of resident defects in software products. In another work by the same authors [5], a more sophisticated BBN model is proposed for the prediction of the post-release failure density of software products.

The theoretical foundation of BBNs is Bayesian analysis and inference, as a formal means for dealing with imprecise data and expert judgment. Bayesian analysis has been already applied in Ref. [3] for the calibration of COCOMO II parametric effort estimation model [2]. More specifically, sample project data and expert estimations are combined in order to provide more accurate rating values for the scale factors in the effort multipliers of the model. In Ref. [10], it is shown that Bayesian inference can be used to produce error distributions on subjective estimates of the cohesion classification of software modules.

In this paper, we propose the use of BBNs as a way of estimating the productivity of software projects in the early

[*] Corresponding author. Tel.: +30-31-998227; fax: +30-31-998419.
E-mail address: stamelos@csd.auth.gr (I. Stamelos).

0950-5849/03/$ - see front matter © 2002 Elsevier Science B.V. All rights reserved.
PII: S0950-5849(02)00163-5

stages of the development process. We argue that BBNs can support an expert in estimating productivity, by explicitly modeling the uncertainties concerning the various factors influencing the productivity. In Section 2, we present a short overview of BBNs and their advantages, and in Section 3 we explain how they can be used in productivity estimation. In Section 4, we demonstrate an example of use of BBNs in the productivity estimation, based on the COCOMO81 cost factors [1]. We also present a tentative BBN model for estimating productivity in ERP system localization. Finally, in Section 5 we conclude the paper and present ideas for future work.

For the purposes of the examples of the BBNs in Sections 2 and 4, i.e. for their graphical representations and various probability calculations, we used the SERENE 1.0 tool [14].

2. Bayesian belief networks

BBNs, as a way of modeling uncertainty, are described in Ref. [13]. The BBNs are directed acyclic graphs (DAGs) expressing probabilistic cause–effect relations among the linked nodes. Each node represents a random variable that can take discrete or continuous values according to a probability distribution, which can be different for each node. Each influence relationship is described by an arc starting from the influencing variable (parent node) and terminating on the influenced variable (child node). The absence of an arc connecting two nodes is an indication of conditional independence between the corresponding variables, i.e. there are no situations in which the probabilities of one of the variables depend directly upon the values of the other.

Formally, the relation between the two nodes is based on Bayes' rule:

$$P(X|Y) = P(Y|X)P(X)/P(Y)$$

Each discrete variable node has an $N \times M$ node probability table (NPT), where N is the number of node states and M is the product of its cause-nodes states. In this table, each column represents a conditional probability distribution and, consequently, its values sum up to 1. Further information about the definition, usage, updating algorithms and complexity analysis of BBNs can be found in Refs. [12,13].

In Fig. 1, an example of a simple BBN is illustrated. The BBN has three nodes A, B and C, whose variables take discrete values: (T)rue and (F)alse for A, LOW, MED, HIGH for B and ON, OFF for C. The NPTs of each node are shown in Table 1.

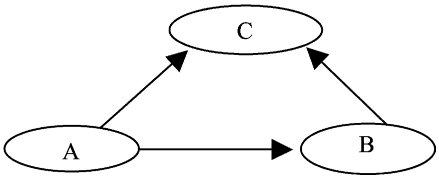


Fig. 1. A simple BBN.

Table 1
The NPT table of the nodes of the BBN in Fig. 1; (a) node A, (b) node B, and (c) node C

(a)
A-node

T	F
0.6	0.4

(b)

B-node	A-node	
	T	F
LOW	0.3	0.6
MED	0.5	0.25
HIGH	0.2	0.15

(c)

	A-node					
	T			F		
B-node C-node	LOW	MED	HIGH	LOW	MED	HIGH
ON	0.7	0.65	0.4	0.45	0.23	0.07
OFF	0.3	0.35	0.6	0.55	0.77	0.93

In Appendix A, we give the explicit explanation of the probabilistic relations between the nodes of the specific BBN. We also give the calculations, which lead to the probability distribution of each node (variable). So, for the three nodes, we have the following distributions:

A-node	
T	0.6
F	0.4
B-node	
LOW	0.42
MED	0.4
HIGH	0.18
C-node	
ON	0.5042
OFF	0.4958

In a BBN, it is possible to have 'observed' values in some nodes. These are the values, which the nodes are known to have in a given instance. These values represent a posteriori probabilities, and, by applying the Bayes rule in each affected node, they can influence other BBN nodes, modifying the probability distributions in their NPTs. For example, in the BBN of Fig. 1 if we have the observed value T(rue) in node A (in other words $P(A = T) = 1$, $P(A = F) = 0$), then the probabilities of values in B and C nodes become: $P(B = \text{LOW}) = 0.3$, $P(B = \text{MED}) = 0.5$, $P(B = \text{HIGH}) = 0.2$, and $P(C = \text{ON}) = 0.615$, $P(C = \text{OFF}) = 0.385$, by applying Bayes' rule.

It is possible to have observed values in input nodes (the ones with no cause-nodes), as well as in intermediate or output nodes (the ones having cause-nodes). It is also possible to combine two or more observed values in the same node. Their probabilities are then recalculated using Bayes rule. For example, in node B, we could have observed values LOW and MED with probabilities $P(B = \text{LOW}) = 0.512$, $P(B = \text{MED}) = 0.488$. The probabilities of the values of the other two nodes become $P(A = T) = 0.585$, $P(A = F) = 0.415$ and $P(C = \text{ON}) = 0.551$, $P(C = \text{OFF}) = 0.449$. Refer to Appendix A for a detailed explanation of the calculations that underlie the above examples of BBN behavior.

Regarding the advantages of BBNs we can mention briefly that they are highly suitable for supporting expert judgment in cost estimation for a number of reasons:

- BBNs can combine information from past statistical data, where available, with expert opinion. Expert opinion may be used when the data are missing or are considered unsuitable to be used, e.g. data taken from projects developed in different domains, using different development processes, etc. On the other hand, parts (subnetworks) of a BBN may be derived from automated data analysis.
- BBNs can be constructed fairly easily. However, they should be constructed carefully because the size of the probability matrices of each node grows exponentially, as the values and the cause-nodes increase.
- BBNs can integrate partial knowledge and data concerning a project in the form of observed values of some nodes. As a consequence, they can be also used as backward reasoning tools in post-mortem analysis, in order to generate hypotheses about project factors affecting productivity.
- BBNs can be used for 'what-if' analysis to explore the impact of changes in some nodes to other nodes.
- BBNs have strong theoretical background (Bayes theory, Pearl's polytree algorithm, Jensen's junction trees).
- A number of software tools for building and using BBNs is already available (see Ref. [11] for a list of such tools).

3. Use of BBNs in productivity estimation

As mentioned earlier, BBNs can be used to support experts in modeling the uncertainties in the software development process and eventually in providing a better estimation of the expected productivity. In this paper, by the term 'experts' we mean individuals with a certain degree of experience in the field of software engineering and, in particular, in software productivity estimation.

For the task of providing an estimate, a BBN specific to the problem domain and development context must be devised. In its construction, one or more experts need to identify the significant factors, which are likely to influence the productivity of a software project lying in the specific domain and development context. Of course, already proposed and defined factors from other cost methods may be used too. The cause–effect relations among these factors need also to be identified. The set of factors will be the set of nodes of the BBN, and the cause–effect relations will be represented by its edges. In addition, for each node an appropriate measurement scale must be defined. The output of the BBN can be a probability distribution of interval estimates of productivity. This is in line with the widely accepted opinion that it is safer to produce interval estimates, along with a probability distribution over the estimate interval (for example, see Ref. [8]).

Since the NPTs of the BBN nodes need to be completed by humans, care must be taken to keep their size as manageable as possible. There is a trade-off between the granularity of the estimation output and the manageability of the BBN: the smaller the size of the NPTs, the more manageable the BBNs become. On the other hand, the larger will be the final interval estimates of the output of the BBN. In order to reduce the size and mental complexity of the NPTs, the values of each node must be as few as possible, preferably no more than eight according to our experience. More importantly, the cause-nodes of each effect-node must also be as few as possible, preferably no more than four. If it is not possible to respect this rule of thumb, then probably additional nodes may be necessary to be inserted between the cause-nodes and the effect node to keep the NPT size manageable. See for instance, the additional nodes 'technical factors' and 'human and process factors' in the example in Section 4.

After constructing the BBN, the experts need to provide values and associated probabilities for the input nodes, in order to produce an estimate for a specific new project. Typically, the values in the NPTs of the intermediate nodes (those having cause-nodes) will not vary for each project, since they reflect the expert opinion about the way the nodes affect each other in general. Partial knowledge for a project can be included in the estimation in the form of observed values of some nodes as explained earlier. This is done only when making forecasts, not when preparing the model.

As said before, the output of the BBN may be an estimation of the productivity, expressed as a probability distribution of productivity intervals. If the probability of an interval is not adequately high, the concatenation of two (or more) neighboring intervals with the highest probability can be taken. In this case, there will be more uncertainty in the estimate, reflecting the uncertainty of the expert in assigning probabilities to the node values. Alternatively, if a point estimation is required, a location statistic of the probability distribution can be taken.

Occasionally, the expert(s) may need to maintain the BBN constructed in order to improve estimation. In this maintenance process, the expert(s) may need to add or exclude some nodes and edges in the BBN, obtaining

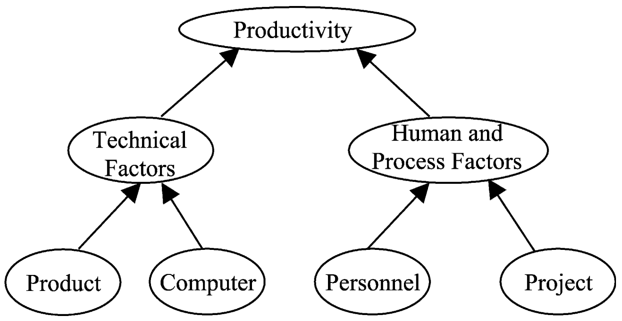


Fig. 2. The COCOMO81-based BBN.

a different set of productivity factors and cause–effect relations among them. Additionally, they may revise the NPTs of the nodes as well, by monitoring the results of the initial BBN and taking into consideration new project productivity figures as they become available.

4. Productivity estimation example

We demonstrate our approach using an example based on the popular COCOMO81 intermediate cost model [1]. Our choice was motivated by the wide spread of the model, the availability of the ratings for the cost drivers and the availability of the COCOMO81 project data.

Productivity P is defined as the ratio between the size S and the effort E of a project, i.e. $P = S/E$. In our example, we are estimating productivity directly. As an alternative, productivity may be calculated by the equation $P = P_{\mathrm{NOM}}F$, where P_{NOM} is the nominal productivity and F is a multiplicative factor. F could be estimated by a suitable BBN, while P_{NOM} might be given by the expert or calculated from large, multi-organizational cost data sets (e.g. the ISBSG data set [7]).

We constructed the BBN shown in Fig. 2, based on Boehm's informal classification of cost factors. The product related factors—RELY, DATA and CPLX—are aggregated to the Product node. In the same sense, TIME, STOR, VIRT, TURN factors are aggregated to the Computer node, ACAP, AEXP, PCAP, VEXP, and LEXP are aggregated to the personnel node and MODP, TOOL, SCED, and RVOL are aggregated to the project node. As another alternative, in a more sophisticated BBN beyond the scope of our example, size and development mode of a project may be taken into account as well, since these variables are also quantities subject to estimation.

The variables of each node are discrete. In each node, except of the productivity node, the variables take five values, namely Very Low, Low, Average, High, and Very High.

In the productivity node, the variable takes eight values which are numeric intervals given by an expert. They express intervals of the productivity measured in delivered source instructions per man-month (DSI/MM). These intervals are shown in Table 2. Narrower estimate intervals may be obtained through the elaboration of a more sophisticated BBN.

Unequal intervals were chosen due to the log-normal distribution of productivity in the COCOMO81 data base, as shown by statistical analysis. In order to fill in the NPTs of each node with probability values we relied entirely on the cost driver ratings for each of the 63 projects in COCOMO81 project database as they are given in Ref. [1], pp. 496–497. The NPT values were decided empirically by the authors, by observing the conditional probabilities of the cost factor ratings. Productivity probability distributions for each of the 63 projects selecting observed values for the nodes of the first level (product, computer, personnel and project) were also computed. These figures were used to choose semi-automatically the NPT values and were further refined based on the authors' intuition.

The reader is reminded that cost factor ratings and values were derived in the first place by Boehm, by consulting the managers of the projects in the COCOMO81 data base. The publicly available information about COCOMO81 provided us with a situation in which part of the expert opinion we needed to construct a BBN was already partially modeled. Overall, we tried to emulate the way an expert might work for constructing a BBN, based on available information and productivity data. In general, in a real case, we would have consulted one or more experts and/or combine their opinions, expressed as probability values in the NPTs, with existing statistical data if available. Nevertheless, we believe that this issue deserves further investigation.

The NPTs for COCOMO81, filled with probability values, are shown in Appendix B. We must remember that these tables were derived artificially. An expert under different conditions might have expressed different probability values and might have filled more table cells with values. However, this fact might increase significantly the uncertainty in the estimates.

After having determined the BBN, we conducted a trial study in which we took again into account the information available in the COCOMO81 project data set. We derived values for the BBN nodes by using simple aggregation rules for the corresponding leaf COCOMO81 cost drivers.

Table 2
The eight interval values of the Productivity node. The productivity is measured in DSI/MM

Interval number	1	2	3	4	5	6	7	8
DSI/MM	<30	30–60	60–100	100–160	160–270	270–450	450–750	>750

Table 3
The output probability distribution for COCOMO81 project no. 48 (observed values PRODUCT = High, Computer = Very High, PROJECT = Average and PERSONNEL = Average)

Productivity (in DSI/MM) distribution		
From	To	Probability (%)
	<30	0
30	60	0
60	100	4.3
100	160	4.3
160	270	69.9
270	450	17.2
450	750	4.3
>750		0

For example, in project no. 48, the reported values for the cost drivers affecting personnel are ACAP = Low, AEXP = High, PCAP = Low, VEXP = High, and LEXP = High. These values have been aggregated to PERSONNEL = Average for the purposes of our tentative BBN. Eventually, for project no. 48 we used the observed values: PRODUCT = High, COMPUTER = Very High, PROJECT = Average, and PERSONNEL = Average, and produced the output productivity distribution shown in Table 3. In this project, the productivity will range between 160 and 270 DSI/MM with a probability of 70%. An estimation based on the interval of 160–270 DSI/MM would be successful, since the actual productivity of this project was 244.4 DSI/MM.

Comparing the results of this trial study with the actual productivity values we found that the interval estimated contained the actual value in roughly 52–67% of the projects, depending on whether we chose one or two neighboring intervals, respectively. The above figures indicate the potential of providing reasonable estimates, which can be further improved through suitable refinement.

We are currently in the process of studying the construction of a BBN for the estimation of the effort in the localization process (i.e. translation and adaptation to local requirements) of ERP systems. For this purpose, we have built a BBN with the collaboration of experts from a software group specializing in software localization. The group possesses no historical data for estimation purposes. A tentative version of this BBN is given in Appendix C.

5. Conclusions and future work

In this paper, we proposed a way for estimating the productivity in software projects using BBNs. BBNs are suitable for modeling uncertainty, and in our opinion can be used to support expert judgment in the production of reasonable and safe estimates.

We demonstrated our approach using an example of a BBN based on the COCOMO81 cost factors. In the construction of the BBN used in the example above, we were based on the COCOMO81 intermediate model for the construction of the network and also for the determination of the probability tables of the nodes. The use of COCOMO81 was intended for demonstration purposes of the BBN method on productivity estimation. Nevertheless, it is obvious that for a real problem of productivity estimation one needs to take into consideration additional factors, such as the important factor of software reuse, and explore their interdependencies. As a result, he/she needs to devise different BBNs to reflect different problem domains and development environments.

We believe that BBNs can be proved particularly useful to start-up companies where historical data are totally lacking or in estimation situations where the first steps in new problem domains or development contexts are taken. In such situations, the expert judgment method is often the only applicable method for productivity estimation.

As all estimation approaches, BBNs will have to be used in real estimation situations and validated against actual data. In the future, we plan to refine the localization BBN, use it for producing effort estimates and eventually validate it against actual project effort data. A research issue that has emerged form the trial study with COCOMO81 is the need for a formal approach to help human experts produce the NPT values. Another interesting idea is to study a systematic and perhaps automated approach to derive probability values in the NPTs from existing project cost databases.

Acknowledgements

The authors would like to thank the reviewers for their helpful comments.

Appendix A

The entries of the NPT for the A node, which is not a child of any other node, show the probabilities assigned to the values of the variable A:

$$P(A = T) = 0.6 \quad \text{and} \quad P(A = F) = 0.4$$

The entries of the NPT for the B node, which is a child of node A, show the conditional probabilities assigned to the values of variable B given the values of A (throughout Appendix A L stands for LOW, M stands for MED and H stands for HIGH):

$$P(B = L/A = T) = 0.3, \qquad P(B = L/A = F) = 0.6,$$

$$P(B = M/A = T) = 0.5, \qquad P(B = M/A = F) = 0.25,$$

$$P(B = H/A = T) = 0.2, \qquad P(B = H/A = F) = 0.15$$

These probabilities can be used to compute the probability distribution for variable B:

$$\begin{aligned}P(B=L) &= P(B=L/A=T)P(A=T)\\&\quad+P(B=L/A=F)P(A=F)\\&=(0.3)(0.6)+(0.6)(0.4)=0.42,\\P(B=M)&=P(B=M/A=T)P(A=T)\\&\quad+P(B=M/A=F)P(A=F)\\&=(0.5)(0.6)+(0.25)(0.4)=0.4,\\P(B=H)&=P(B=H/A=T)P(A=T)\\&\quad+P(B=H/A=F)P(A=F)\\&=(0.2)(0.6)+(0.15)(0.4)=0.18\end{aligned}$$

From the conditional probabilities of B given A, we can easily calculate the joint probability distribution of the variables A and B. For example,

$$\begin{aligned}P(A=T,B=L)&=P(B=L/A=T)P(A=T)\\&=(0.3)(0.6)=0.18,\\P(A=F,B=L)&=P(B=L/A=F)P(A=F)\\&=(0.6)(0.4)=0.24,\ \text{etc.}\end{aligned}$$

So the joint probability distribution of A and B is:

		A	
		T	*F*
	L	0.18	0.24
B	*M*	0.3	0.1
	H	0.12	0.06

Since now the node C is a child of A and B, the entries of its NPT are the conditional probabilities of the values of variable C given the joint values of A and B. That is:

$$\begin{aligned}&P(C=\mathrm{ON}/A=T,B=L)\\&\quad=0.7,\ldots,P(C=\mathrm{OFF}/A=F,B=H)=0.93\end{aligned}$$

We can now compute the probability distribution of variable C:

$$\begin{aligned}&P(C=\mathrm{ON})\\&\quad=\sum_{\substack{x\in\{T,F\}\\y\in\{L,M,H\}}}P(C=\mathrm{ON}/A=x,B=y)P(A=x,B=y)\\&\quad=0.5042\end{aligned}$$

and

$$\begin{aligned}&P(C=\mathrm{OFF})\\&\quad=\sum_{\substack{x\in\{T,F\}\\y\in\{L,M,H\}}}P(C=\mathrm{OFF}/A=x,B=y)P(A=x,B=y)\\&\quad=0.4958\end{aligned}$$

Note that the above notation of the calculations can be simplified by using matrix multiplication:

$$P(B)=\mathrm{NPT}(B)\mathrm{NPT}(A)=\begin{pmatrix}0.3 & 0.6\\0.5 & 0.25\\0.2 & 0.15\end{pmatrix}\begin{pmatrix}0.6\\0.4\end{pmatrix}=\begin{pmatrix}0.42\\0.4\\0.18\end{pmatrix}$$

$$\begin{aligned}P(A,B)=\mathrm{NPT}(B)\mathrm{diag}(\mathrm{NPT}(A))&=\begin{pmatrix}0.3 & 0.6\\0.5 & 0.25\\0.2 & 0.15\end{pmatrix}\begin{pmatrix}0.6 & 0\\0 & 0.4\end{pmatrix}\\&=\begin{pmatrix}0.18 & 0.24\\0.3 & 0.1\\0.12 & 0.06\end{pmatrix}\end{aligned}$$

$$\begin{aligned}P(C)&=\mathrm{NPT}(C)\mathrm{vector}(P(A,B))\\&=\begin{pmatrix}0.7 & 0.65 & 0.4 & 0.45 & 0.23 & 0.07\\0.3 & 0.35 & 0.6 & 0.55 & 0.77 & 0.93\end{pmatrix}\begin{pmatrix}0.18\\0.3\\0.12\\0.24\\0.1\\0.06\end{pmatrix}\\&=\begin{pmatrix}0.5042\\0.4958\end{pmatrix}\end{aligned}$$

If we have observed value T for variable A (i.e. $P(A=T)=1$), then the probability distributions of variables B and C can be computed as before:

$$P(B)=\begin{pmatrix}0.3 & 0.6\\0.5 & 0.25\\0.2 & 0.15\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix}=\begin{pmatrix}0.3\\0.5\\0.2\end{pmatrix}$$

$$P(A,B)=\begin{pmatrix}0.3 & 0.6\\0.5 & 0.25\\0.2 & 0.15\end{pmatrix}\begin{pmatrix}1 & 0\\0 & 0\end{pmatrix}=\begin{pmatrix}0.3 & 0\\0.5 & 0\\0.2 & 0\end{pmatrix}$$

$$P(C)=\begin{pmatrix}0.7 & 0.65 & 0.4 & 0.45 & 0.23 & 0.07\\0.3 & 0.35 & 0.6 & 0.55 & 0.77 & 0.93\end{pmatrix}\begin{pmatrix}0.3\\0.5\\0.2\\0\\0\\0\end{pmatrix}=\begin{pmatrix}0.615\\0.385\end{pmatrix}$$

When in node B we have some evidence that the values LOW and MED are observed with probabilities $P(B=L)=0.512$, $P(B=M)=0.488$ and $P(B=H)=0$, then the calculation of the probability distributions of the variables A and C is obtained by the following steps.

Using the Bayes rule, we calculate the conditional probabilities of A given B:

$$\begin{aligned}
P(A = T/B = L) &= P(B = L/A = T)P(A = T)/P(B = L) \\
&= (0.3)(0.6)/(0.42) = 0.429, \\
P(A = T/B = M) &= P(B = M/A = T)P(A = T)/P(B = M) \\
&= (0.5)(0.6)/(0.4) = 0.75, \\
P(A = T/B = H) &= P(B = H/A = T)P(A = T)/P(B = H) \\
&= (0.2)(0.6)/(0.18) = 0.667, \\
P(A = F/B = L) &= P(B = L/A = F)P(A = F)/P(B = L) \\
&= (0.6)(0.4)/(0.42) = 0.571, \\
P(A = F/B = M) &= P(B = M/A = F)P(A = F)/P(B = M) \\
&= (0.25)(0.4)/(0.4) = 0.25, \\
P(A = F/B = H) &= P(B = H/A = F)P(A = F)/P(B = H) \\
&= (0.15)(0.4)/(0.18) = 0.333
\end{aligned}$$

Then, using the new observed probabilities we have:

$$\begin{aligned}
P(A=T) = {} & P(A=T/B=L)P(B=L) \\
& + P(A=T/B=M)P(B=M) \\
& + P(A=T/B=H)P(B=H) = (0.429)(0.512) \\
& + (0.75)(0.488) + (0.667)(0) = 0.585
\end{aligned}$$

and

$$\begin{aligned}
P(A=F) = {} & P(A=F/B=L)P(B=L) \\
& + P(A=F/B=M)P(B=M) \\
& + P(A=F/B=H)P(B=H) = (0.571)(0.512) \\
& + (0.25)(0.488) + (0.333)(0) = 0.415
\end{aligned}$$

The joint probability distribution of the variables A and B is now:

$$\begin{aligned}
P(A = T, B = L) &= P(A = T/B = L)P(B = L) \\
&= (0.429)(0.512) = 0.219, \\
P(A = F, B = L) &= P(A = F/B = L)P(B = L) \\
&= (0.571)(0.512) = 0.292, \text{ etc.}
\end{aligned}$$

So the joint probability distribution of A and B is:

		A	
		T	F
	L	0.219	0.292
B	M	0.365	0.122
	H	0	0

Then, we have for node C:

$$P(C) = \begin{pmatrix} 0.7 & 0.65 & 0.4 & 0.45 & 0.23 & 0.07 \\ 0.3 & 0.35 & 0.6 & 0.55 & 0.77 & 0.93 \end{pmatrix} \begin{pmatrix} 0.219 \\ 0.365 \\ 0 \\ 0.292 \\ 0.122 \\ 0 \end{pmatrix} = \begin{pmatrix} 0.551 \\ 0.449 \end{pmatrix}$$

Appendix B

Tables (B1)–(B3).

Table B1
The NPT of technical factors node

Computer	VERY HIGH					HIGH					AVERAGE					LOW					VERY LOW				
Product	VH	H	AV	LO	VL	VH	H	AV	LO	VL	VH	H	AV	LO	VL	VH	H	AV	LO	VL	VH	H	AV	LO	VL
Technical factors																									
VERY HIGH	1	0.57	0	0	0	0.5	0	0	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HIGH	0	0.43	1	1	0	0.5	1	0.75	0	0	0.99	0.5	0	0	0	0.42	0	0	0	0	0	0	0	0	0
AVERAGE	0	0	0	0	0.67	0	0	0.25	1	0.24	0	0.5	1	0.25	0	0.58	1	1	0	0	0.66	0.17	0	0	0
LOW	0	0	0	0	0.33	0	0	0	0	0.76	0	0	0	0.75	1	0	0	0	1	1	0.34	0.83	1	1	0
VERY LOW	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Table B2
The NPT of human and process factors

Project	VERY HIGH					HIGH					AVERAGE				
Personnel	VH	H	AV	LO	VL	VH	H	AV	LO	VL	VH	H	AV	LO	VL
Technical factors															
VERY HIGH	1	0.17	0	0	0	0.33	0	0	0	0	0	0	0	0	0
HIGH	0	0.83	0	0	0	0.67	1	0.4	0	0	1	0.61	0	0	0
AVERAGE	0	0	1	1	0.55	0	0	0.6	0.67	0.14	0	0.39	1	0	0
LOW	0	0	0	0	0.45	0	0	0	0.33	0.86	0	0	0	1	1
VERY LOW	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Computer	LOW					VERY LOW									
Product	VH	H	AV	LO	VL	VH	H	AV	LO	VL					
Technical factors															
VERY HIGH	0	0	0	0	0	0	0	0	0	0					
HIGH	0.57	0	0	0	0	0.11	0	0	0	0					
AVERAGE	0.43	1	1	0	0	0.89	0.73	0.08	0	0					
LOW	0	0	0	1	0.47	0	0.27	0.92	0.62	0					
VERY LOW	0	0	0	0	0.53	0	0	0	0.38	1					

Table B3
The NPT of productivity

Technical factors	VERY HIGH					HIGH					AVERAGE				
Human and process factors	VH	H	AV	LO	VL	VH	H	AV	LO	VL	VH	H	AV	LO	VL
Productivity															
<30	0	0	0	0	0.57	0	0	0	0	0.732	0	0	0	0.66	0.9
30–60	0	0	0	0	0.19	0	0	0	0.33	0.184	0.155	0	0	0.17	0.1
60–100	0	0	0	0	0.14	0.14	0	0.1	0.67	0.084	0.14	0	0.33	0.12	0
100–160	0.09	0	0	0	0.1	0.125	0.133	0.1	0	0	0.13	0.43	0.67	0.05	0
160–270	0.15	0	1	1	0	0.14	0	0.3	0	0	0.145	0.285	0	0	0
270–450	0.14	0.25	0	0	0	0.14	0.333	0.4	0	0	0.14	0.285	0	0	0
450–750	0.14	0.25	0	0	0	0.14	0.4	0.1	0	0	0.14	0	0	0	0
>750	0.48	0.5	0	0	0	0.315	0.133	0	0	0	0.14	0	0	0	0
Technical factors	LOW					VERY LOW									
Human and process factors	VH	H	AV	LO	VL	VH	H	AV	LO	VL					
<30	0	0	0.375	0.806	1	0.375	0.56	0.778	1	1					
30–60	0	0.33	0.25	0.156	0	0.185	0.17	0.167	0	0					
60–100	0	0	0.25	0.038	0	0.135	0.12	0.055	0	0					
100–160	0	0.17	0.125	0	0	0.125	0.11	0	0	0					
160–270	1	0	0	0	0	0.138	0.4	0	0	0					
270–450	0	0	0	0	0	0.042	0	0	0	0					
450–750	0	0	0	0	0	0	0	0	0	0					
>750	0	0	0	0	0	0	0	0	0	0					

Appendix C

The tentative BBN shown represents the experts' opinion about cause–effect relationships of the localization process productivity. In the following, the description of each input node is given, along with levels for all nodes.

Text expansion (text swell). Ability to handle expanded translated text in dialog boxes, pop-up windows, menus, etc. Software designers may have anticipated this. (Levels: Yes (handles), partially, No)

Message concatenation. A frequently used method that should be avoided because it creates significant problems in the localization process due to different grammatical rules each language may have. All sentences should be self-contained and complete. (Levels: Yes (uses), No)

Software architecture.Localizable elements should be in separate resources and not hardcoded in the source files. (Levels: Yes (separate), No (hardcoded))

Localizable images. Use of images or graphics that contain text or in any way should be localized. (Levels: Many, Few, None)

Text flow. Different text flow required by the target language (left-to-right, right-to-left, vertical, etc.) may have significant impact in the localization process and productivity. (Levels: Same, Different)

Regional settings. Ability to handle different settings for date and number format. (Level: Yes, partially, No)

Product translatability. Level of difficulty for the product to be translated. (Levels: Low, Med, High)

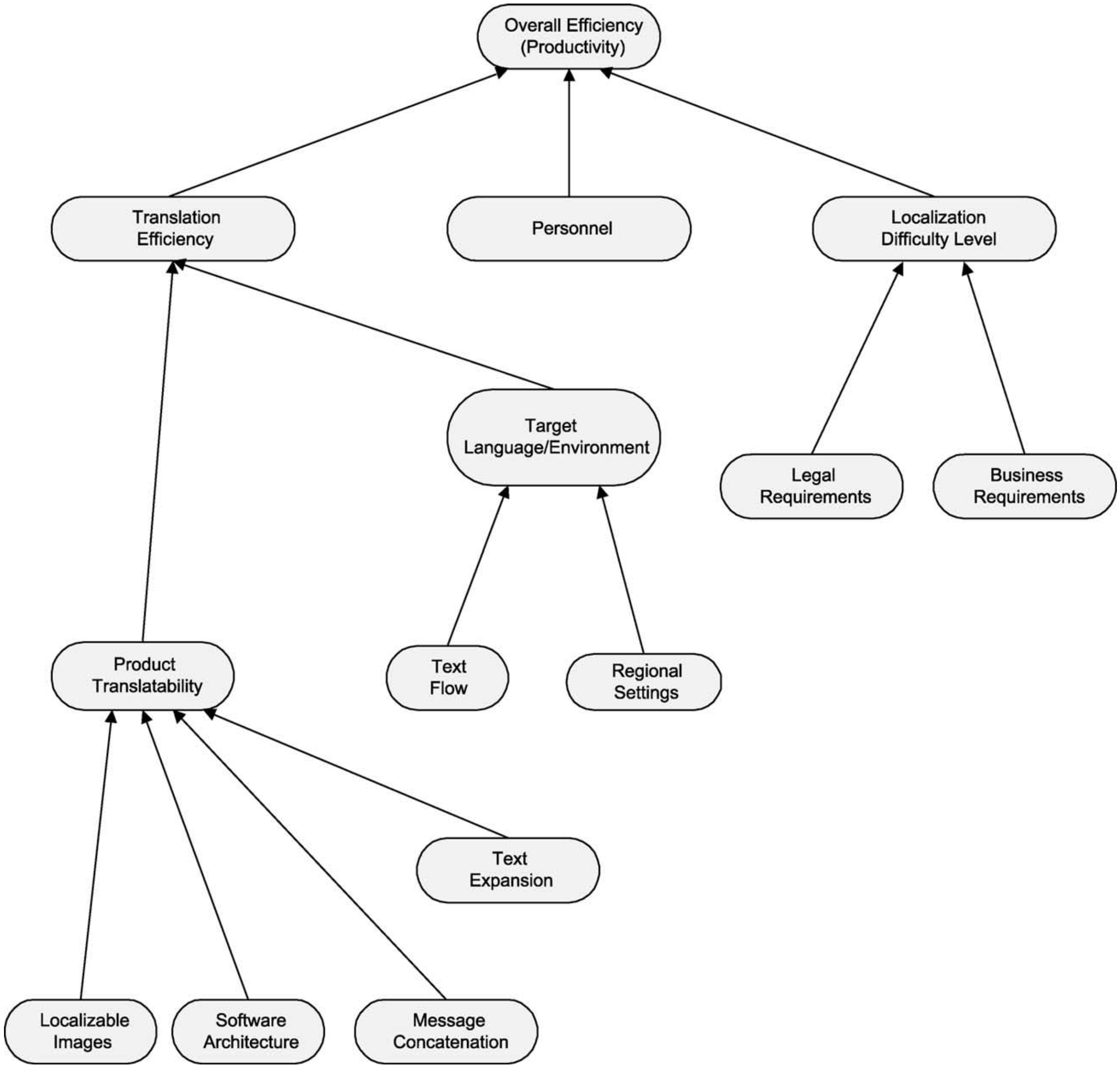
Target language/environment. Levels: Similar, Different

Translation efficiency. Very Low, Low, Med, High

Personnel. Personnel experience (Levels: Low, Med, High)

Localization difficulty level: Low, Med, High

Legal/business requirements. Number and difficulty level combined (Levels: Low, High)



References

[1] B. Boehm, Software Engineering Economics, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[2] Center for Software Engineering, COCOMO II Model Definition Manual (Computer Science Department, USC Center for Software Engineering, 1997).

[3] S. Chulani, B. Boehm, B. Steece, Bayesian analysis of empirical software engineering cost models, IEEE Transactions on Software Engineering 25 (4) (1999) 573–583.

[4] N. Fenton, M. Neil, A critique of software defect prediction models, IEEE Transactions on Software Engineering 25 (5) (1999) 675–689.

[5] N. Fenton, M. Neil, Software metrics: successes, failures and new directions, The Journal of Systems and Software 47 (1999) 149–157.

[6] R. Hughes, Expert judgement as an estimating method, Information and Software Technology 38 (1996) 67–75.

[7] ISBSG Software Project Dataset, http://www.isbsg.org.au/datadisk.htm.

[8] B. Kitchenham, S. Linkman, Estimates, uncertainty and risk, IEEE Software 14 (3) (1997) 69–74.

[9] L. Leibovici, M. Fishman, H. Schønheyder, C. Riekehr, B. Kristensen, I. Shraga, S. Andreassen, A causal probabilistic network for optimal treatment of bacterial infections, IEEE Transactions on Knowledge and Data Engineering 12 (4) (2000) 517–528.

[10] J. Moses, Bayesian probability distributions for assessing measurement of subjective software attributes, Information and Software Technology 42 (2000) 533–546.

[11] K. Murphy, Bayesian Network Software Packages, http://www.cs.berkeley.edu/~murphyk/Bayes/bnt.html, http://groups.yahoo.com/group/BayesNetToolbox/.

[12] R. Neapolitan, Probabilistic Reasoning in Expert Systems: Theory and Algorithms, Willey, New York, 1990.

[13] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann Publishers, San Mateo, CA, 1988.

[14] SERENE Home, http://www.hugin.dk/serene/.

[15] H. Ziv, D. Richardson, Constructing Bayesian-network Models of Software Testing and Maintenance Uncertainties, Proceedings of the 1997 International Conference on Software Maintenance (ICSM'97), 1997.