ELSEVIER

Knowledge-Based Systems 14 (2001) 307–325

Knowledge-Based SYSTEMS

www.elsevier.com/locate/knosys

Making decisions: using Bayesian nets and MCDA

N. Fenton[a,*], M. Neil[b]

[a]*Computer Science Department, Queen Mary (University of London), London, E1 4NS, UK*
[b]*Agena Ltd, 11 Main Street, Caldecote, Cambridge, CB3 7NU, UK*

Received 17 August 1999; revised 7 April 2000; accepted 20 April 2000

Abstract

Bayesian belief nets (BBNs) have proven to be an extremely powerful technique for reasoning under uncertainty. We have used them in a range of real applications concerned with predicting properties of critical systems. In most of these applications we are interested in a single attribute of the system such as safety or reliability. Although such BBNs provide important support for decision making, in many circumstances we need to make decisions based on *multiple* criteria. For example, a BBN for predicting the safety of a critical system cannot be used to make a decision about whether or not the system should be deployed. This is because such a decision must be based on criteria other than just safety (cost, politics, and environmental factors being obvious examples). In such situations the BBN must be complemented by other decision making techniques such as those of multi-criteria decision aid (MCDA). In this article we explain the role of BBNs in such decision-making and describe a generic decision-making procedure that uses BBNs and MCDA in a complementary way. The procedure consists of identifying the *objective* and *perspective* for the decision problem, as well as the *stakeholders.*This in turn leads to a set of possible *actions,*a set of *criteria* and *constraints.*We distinguish between, *uncertain* and *certain* criteria. The BBN links all the criteria and enables us to calculate a value (within some probability distribution in the case of the uncertain criteria) for each criterion for a given action. This means that we can apply traditional MCDA techniques to combine the values for a given action and then to rank the set of actions. The techniques described are demonstrated by real examples, including a safety assessment example that is being used by a major transportation organisation.

Keywords: Bayesian belief networks; Multi-criteria decision aid; Analytical hierarchy process

1. Introduction

Bayesian belief networks (BBNs) model problems that involve uncertainty. A BBN is a directed graph, such as the one shown in Fig. 1, which is one of the main examples we will explain and use in Section 6. The nodes of a BBN represent uncertain variables and the arcs are the causal or influential links between the variables. Associated with each node is a set of conditional probability functions that model the uncertain relationship between the node and its parents. The benefits of using BBNs to model uncertain domains are well known [1,2], especially because the breakthroughs in algorithms [3,4] and tools to implement them [5].

BBNs have proven to be an extremely powerful technique for reasoning under uncertainty. We have used them in a range of real applications concerned with predicting properties of critical systems. For example, in recent collaborative projects we have used BBNs to:

- provide safety or reliability arguments for critical computer systems (the DATUM, SHIP, DeVa and SERENE projects have all addressed this problem from different industrial perspectives [5–8]);
- provide improved reliability predictions of prototype military vehicles (the TRACS project [9]]);
- predict general software quality attributes such as defect-density and cost (the IMPRESS project [10,11]).

In consultancy projects we have used BBNs to

- assess safety of PES components in the railway industry;
- provide predictions of insurance risk and operational risk;
- predict defect density of software in consumer electronics products;
- assess risks in changing the architecture of air traffic management systems.

In most of these applications the clients are interested in a

* Corresponding author. Tel.: +44-020-7882-7860; fax: +44-020-8980-6533.
E-mail address: norman@agena.co.uk (N. Fenton).

0950-7051/01/$ - see front matter © 2001 Elsevier Science B.V. All rights reserved.
PII: S0950-7051(00)00071-X

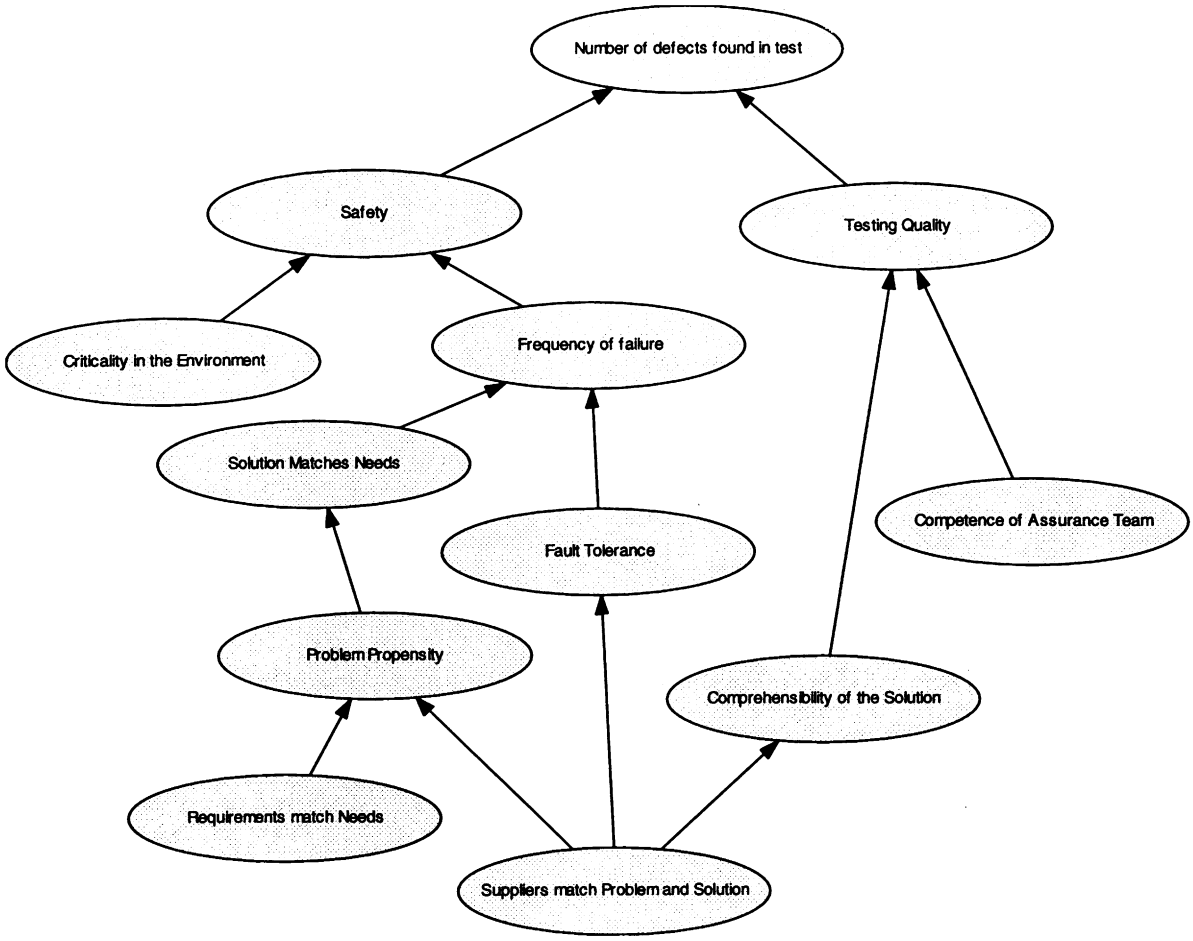

Fig. 1. A BBN for predicting system safety.

single attribute of the system such as safety or reliability. The BBNs capture causal factors that affect the attribute of interest. In such circumstances the BBN provides a powerful decision-support tool, as it can be used to predict the effect that changes to the causal factors have on the attribute of interest. For example, in the BBN of Fig. 1 we can see the likely effects on safety of different quality suppliers and testing processes. However, in many circumstances we need to make decisions based on *multiple* criteria and BBNs do not allow us to incorporate the notion of *preference* which is necessary in such cases. Because of this BBNs cannot, alone, provide a complete solution for the kind of wider decision problems in which a system safety assessment exercise inevitably fits. For example, suppose we wish to determine whether a proposed software-controlled protection system should be deployed in a particular reactor. We could use BBNs (notably the SERENE toolset) [18] to construct a safety argument of the system, but the result of this exercise is merely one component of the information that a regulator will use before reaching a decision about deployment. The regulator will be interested in other criteria like cost (economic, environmental, and political) and functionality. These criteria may have heavier weighting than the predicted safety level when it comes to making a decision about the nature of deployment. In other words, the safety assessment problem is attempting to predict just a single criterion in what is a multi-criteria decision problem.

In this article we provide a model of reasoning about the broader context in which safety assessment is performed. In doing this we clarify misunderstandings about important concepts in dependability argumentation and show how to avoid the confusion and ambiguity associated with much work in this area.

This article covers the following areas, using two examples that are described in Section 2 (one safety critical decision problem, and one everyday decision problem) to illustrate the concepts:

- Identifying objectives from a given perspective (Section 3).
- The notion of a decision problem and its constituent parts; we use the accepted terminology of multi-criteria decision aid (MCDA). Thus, we introduce the key notions of *criteria*, *constraints* and *actions*. (Section 4).
- Defining and measuring criteria (Section 5).

Table 1
Two example problems

Safety critical problem	'Everyday' problem
I am a Government-appointed Regulator for Nuclear Power. It is my job to license computerised equipment for nuclear power plants. It is proposed to deploy a new software controlled protection system in an existing reactor to replace the existing mechanical controlled system. I have the authority to inspect every aspect of the new system's design and test as well as the company that produced it. The problem is to decide whether to: ● Deploy the new system ● Deploy with specified minor changes ● Deploy only after specified major changes ● Do not deploy, but retain existing mechanical system ● Decommission plant Obviously I have to be assured that the new system is sufficiently safe. However, I also have to take account of the cost of the new system and any proposed changes to it, and I have to take account of political requirements and the cost of maintenance (which is expected to be lower with the new system)	I have to get to Heathrow Airport in time for an 8.45 AM flight to Rome. The problem is to choose both the main mode of transport and the departure time. The main modes of transport are: ● Car (i.e. drive myself and park) ● Taxi ● Train For simplicity we take the departure times to be discrete one hour intervals between 4 am and 9am Obviously I want the journey to be as comfortable and cheap as possible (within certain constraints) and I would seek to minimise both the journey time and the waiting time at the airport (again within certain constraints). But I have to take account of certain conflicts and also factors like how much luggage I am carrying as well as the weather, roadworks, train delays, and the rush-hour traffic building up after 6 AM

- The key notion of uncertain criteria and inference (Section 6).
- An example of safety assessment that is a sanitised version of a BBN from a major industrial project (Section 7).
- Combining BBNs and MCDA to provide a more complete solution for decision support under uncertainty (Section 8).

2. The example problems

We will structure this article around two examples. One is a safety critical decision problem, and one is an everyday decision problem to which everyone can relate easily. It is important that the everyday example is included in juxtaposition to the safety-critical one because the concepts are much more widely understood and accepted in such a concrete example. The two problem examples are presented in Table 1.

The key concepts to be defined are shown in summary form in Table 2 for each of the two examples. In the rest of the paper we explain these concepts in more detail.

3. Identifying objective and perspective

The *objective* of a decision problem is the ultimate reason you are interested in solving the problem. The objective is always with respect to a particular *perspective*, the most important component of which is the person or party making the decision. The regulator will have a different perspective of the problem of ensuring that a safe protection system is installed in a nuclear plant compared to the system supplier. Similarly, the traveller will have a different perspective of the problem of getting to Heathrow compared to a London taxi driver.

Example. The objective (from the perspective of the traveller) of the travel problem is get to Heathrow in good time to catch the Rome flight; the objective is *not* to have a comfortable journey, even though this is one of the factors we consider in our final choice. The objective of the safety assessment problem (from the perspective of the Regulator) is to ensure that a safe protection system is installed; it is *not* to deploy the new system, even though this is one of the options available.

The perspective of the problem incorporates not only the decision-maker, but also the *stakeholders*. These are the parties most affected by the chosen outcome *and* whose viewpoints will need to be considered in arriving at a decision. Different stakeholders may have quite different interests, which in turn may be quite different from those of the decision-maker.

Example. In our travel problem let us assume that the traveller is attending an important business meeting in Rome. The most important stakeholders are: (a) the other people (that is, the Romans) who will be at the meeting; and (b) the traveller's boss. The Romans are really only interested in the traveller getting to the airport in time for the flight; if it was their choice alone they would insist on the traveller leaving as early as possible by the quickest mode of transport. The traveller's boss on the other hand is interested in cost, while the traveller is interested in the comfort of the journey and not having to wait too long.

Example. In the safety assessment problem (from the perspective of the regulator) the Government and local community are the main stakeholders; they have similar overall objectives, but they may have radically different ways of judging the best outcome. For example, the local community probably does not consider cost as an

Table 2
The key concepts summarised

	Safety assessment example	Travel example
Objective	To ensure that a safe protection system is installed in a nuclear plant at reasonable cost	To get to Heathrow in good time to catch the Rome flight with reasonable cost/comfort
Perspective	*Decision maker*: The regulator *Key stakeholders*: the Government and the local community	*Decision maker*: The traveller *Key stakeholders*: The people meeting the traveller in Rome
Decision problem	To decide if the proposed computer protection system is appropriate for deployment	To determine the most suitable mode of transport and start time
The set of possible actions	Deploy; Deploy with specified minor changes; Deploy only after specified major changes; Do not deploy, but retain existing mechanical system; Decommission plant	The set of pairs of the form (A,B) where A is the transport type (car, taxi, train) and B is a start-time (4–5, 5–6, 6–7, 7–8, 8–9).
Criteria (functions defined on actions)	*Safety*, *functionality*, *cost* (*financial*), *cost* (*political*). For example, *safety* might be defined as the probability of failure on demand (pfd); *functionality* might be defined as either 'satisfactory' or 'unsatisfactory'; *financial cost* might be the cost in pounds including life-cycle maintenance costs. Note that the value of some criteria for some actions may never been known with certainty.	*Journey time, wait time, cost, comfort.* For example: *journey time* might be defined as the elapsed time in minutes between leaving home and arriving at the check-in desk; *comfort* might be defined as one of 'low', 'medium', or 'high'.
Constraints (properties of criteria that you specify as desirable)	Examples: *safety* $<10^{-3}$ pfd *Cost* <£10 Million	Examples: *wait time* >15 min (otherwise we miss the flight) *Cost* < £50
External factors (variables you cannot control, but which can influence the value of criteria for a given action)	Test results Test effort Experience of development team Quality of methods used	Roadworks Train problems
Internal factors (variables you may be able to control and which can influence the value of criteria for a given action)	Examples: *system load* (you could insist that the system be deployed providing that it is subject to a maximum number of hours of continuous use) *System environment* (you could specify that it can be used for reactor A but not reactor B).	Examples: *start time* (if there are bad roadworks you could leave earlier). *Amount of luggage* to take (given information about roadworks you might be able to cut down on luggage sufficiently for you to be able to go by train)

important factor at all, but the Government certainly will. The Regulator's challenge is to take account of these different considerations as well as those regulations that are his responsibility to enforce.

It is just as important to ensure that we know who are *not* considered to be stakeholders, as this is a crucial step in scoping and simplifying the problem. Generally a party which is affected by the decision should be excluded from being considered a stakeholder if either

1. Their viewpoints/needs are not relevant; or
2. Their viewpoints are fundamentally inconsistent with that of the decision maker or an accepted stakeholder (there is no point in attempting to solve a decision problem when there is *no* solution that could be accepted by *all* the stakeholders).

Example. In the travel problem it is reasonable to exclude the London taxi-drivers from the set of stakeholders. Although they *may* be affected by the outcome (in the sense that one of them may benefit from a high-paying job) there is no need for the traveller to consider the needs of the taxi drivers. The traveller certainly does not owe them a living. On the other hand, the viewpoint of the traveller's wife certainly is relevant. But, if his wife is fundamentally opposed to him travelling abroad on business then there is little point in including her as one of the stakeholders; her only interest is in stopping the trip completely and this is incompatible with the interests of the traveller and other

stakeholders. It would be impossible to arrive at a decision that satisfied them all.

Example. In the safety problem it is reasonable to exclude the system developers from the set of stakeholders. They will be affected by the outcome but it is not necessary to consider their needs, (which in this context is simply to sell the system). The Regulator does not owe the developers a living.

The above examples confirm the importance of identifying a clear and appropriate objective from a clearly defined perspective. Many real-life decision problems fail on this first hurdle. If not done properly the entire safety assessment process could be a costly waste of time. We believe that in many cases where safety assessment is being performed the motive and perspective is not at all clear.

4. The decision problem and its constituent parts

Having identified the objective and perspective our next task is to define the decision problem that we need to solve to meet the objective from the given perspective. Although it is useful to express the decision problem in the kind of summary prose shown in the third row of Table 2, the decision problem is only truly well-defined once we identify the following, using the standard terminology of MCDA [12]:

- the set of possible (mutually exclusive) *actions* we can take (these are the alternatives);
- a set of *criteria*, which are functions defined on actions;
- a set of *constraints* which are properties of the criteria– these can also be thought of as *preferences*.

In both of the examples we have a finite set of actions, but generally the set of actions could be infinite and even continuous. For example, in the travel example, the departure time could be the (continuous) *real* time in the interval between 4.00 and 9.00 AM.

If there were only a single criterion with which to judge the actions it would not be difficult to solve the decision problem—we would just choose the action that returned the 'best' value for the criterion. In the safety example, if safety (defined as the probability of critical failure on demand) really *were* the only criterion on which we had to choose our actions then we would simply choose the action with the highest value of safety. Unfortunately, we also have to consider other criteria like cost (both economic and political) and functionality. Inevitably some of these will be conflicting; the safest system may not be either the cheapest or the one with the most functionality. Generally, we wish to optimise a number of possibly conflicting criteria. We may be guided in our decision choice by a number of constraints. These are properties of the attributes that we regard as necessary (from the chosen perspective)—any action that fails to satisfy a constraint for any criterion is automatically rejected. The more constraints there are the narrower will be our choice of actions. Ideally, we would like the constraints to leave just a single action to choose (the optimal action). In general, this is rare. Hence we have to look at methods that help us to choose between actions.

The extensive body of work on MCDA [12] does provide concrete help for solving such decision problems. MCDA includes such well known techniques as linear programming (only relevant when the criteria all have equal weighting and can be measured on a ratio scale) and other more recent techniques which help us to solve problems in more general cases when we do not have such ideal circumstances. For example, the analytical hierarchy process (AHP) [13] is a popular (albeit crude) technique that includes a means of weighting criteria against each other at one level and actions with respect to particular criteria at a lower level. More rigorous approaches, such as *outranking methods*, avoid the theoretical limitations of AHP, but do not guarantee a linear ordering of the actions; in other words you may still end up with having to find some other method of choosing between 'equally acceptable' best actions.

MCDA has limitations that we must (and fortunately can) take account of by using BBNs in a complementary way. Specifically, the vast body of MCDA techniques makes three critical assumptions:

1. That the relevant criteria are *well defined* (and hence for a given action a it is obvious how you can compute $g(a)$ for a given criteria g).
2. That the relevant criteria are *certain* (and hence for a given action a and criteria g the value $g(a)$ is deterministic rather than stochastic).
3. That the relevant criteria are *independent* of each other.

Example. The following is a classical MCDA problem: Choose one from a set of cars to buy based on the criteria: age, price, engine size, petrol consumption at 30 mph, maximum speed. For a given car x each of the values age(x), price(x), engine_size(x), petrol_consumption(x), maximum_speed(x) are both well defined and certain. Thus, for each 'action' (in other words each car) we can construct a well-defined vector of values corresponding to the various criteria. However, even in this case, the assumption that the criteria are independent of each other is not valid; petrol consumption will depend on age, engine_size etc.

BBNs provide precisely the ammunition for dealing with the important cases when these assumptions are not valid. In the next two sections we explain how.

5. Defining criteria

The theory of MCDA assumes that criteria are always

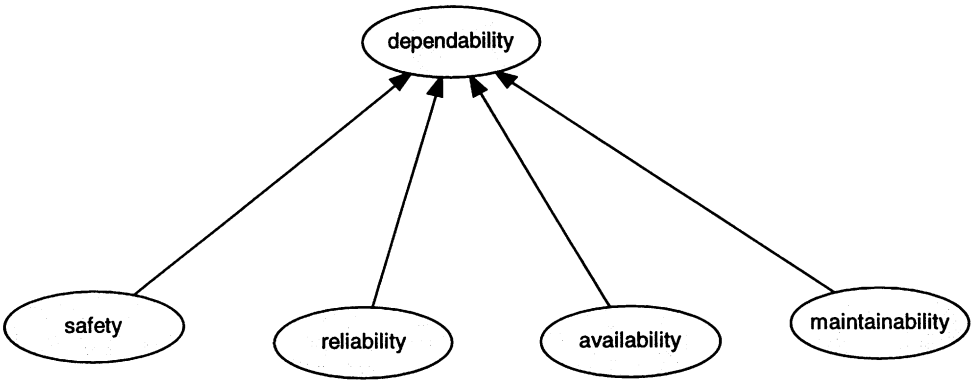

Fig. 2. Definition of system dependability (can be viewed as an instantiation of the definitional/synthesis BBN idiom).

well defined. Our examples confirm that this is not true for real-life problems.

Example. In the travel example the criteria are *start time*, *journey time*, *wait time*, *cost* and *comfort*. Recall that, formally, a criterion is a function from the set of actions into some ordered set. Thus, for a well-defined criterion we first need to define the ordered set (which is the range of the function and which may also be thought of as the 'measurement scale'). For *journey time* and *wait time* this might be the set of positive real numbers or we might be content with a simple ordinal scale like {short, medium, high}. However, in either case the value of the function for each action must be unambiguous. For example, we might define *journey time* as the elapsed time in minutes from leaving our front door until arriving at the airport check-in desk. For a given action, such as $<$taxi, 6.00$>$ we could then compute the value *journey time* $<$taxi, 6.00$>$ as the actual time in minutes taken by taxi when we leave at 6.00AM. The criterion *wait time* does not present too many problems, but even an apparently well-understood criterion like *cost* must be very carefully defined. For example, does *cost* mean just price paid on the day or does it also include overheads (such as a sum for 'wear and tear' when the transport type is your own car)? When it comes to the criterion *comfort* it is not at all clear how the criterion should be defined even if we could agree on an appropriate measurement scale.

Example. In the safety example none of the criteria *safety*, *functionality*, *cost (financial)*, *cost (political)* are easily defined. For example, we have suggested that *safety* might be defined as the probability of failure on demand (pfd). However, this is clearly a function that cannot be properly computed for all of the possible actions—in fact it is only known for certain for one of the possible actions—'Decommission' where safety is perfect (pfd $= 0$) simply because there are no demands.

In summary the problem is that, in most real-world decision problems we will be interested in criteria which are not necessarily well defined (in the sense of MCDA). Let us call such criteria 'synthetic'. It is beyond the scope of this document to provide a set of guidelines on how to define and hence measure 'synthetic' criteria (readers should consult ([14]) for a good general account and ([15]) for an account in the context of software engineering). However, the following points are especially relevant for BBNs:

1. Synthetic criteria are often decomposed into lower level attributes that are assumed to be well defined. For example, according to [16], *system dependability*, is decomposed into *safety*, *reliability*, *availability*, and *maintainability*. In the SERENE approach such decompositions are part of a class of BBNs that we refer to as definitional/synthesis idioms, as shown in Fig. 2. It is important to note that the decomposition alone is not *sufficient* to define the higher level criterion (for example, there may be many ways to define system dependability as a combined measure of the lower level attributes). What you must also *not* do is confuse the decomposition (and any subsequent refined definition) with the notion of *causal dependence*. Dependability is not *caused* by safety, reliability, etc. but is merely *defined* in terms of these attributes.
2. Defining synthetic criteria is the same thing as defining measures for attributes. The rules of measurement theory (notably the representation condition [15]) govern when we have truly defined a measure for an attribute and what the appropriate scale type is. Often a simple ordinal scale may be sufficient for our purposes.
3. A *measure* for an attribute should never be seen as *defining* an attribute (this is one of the most important lessons of measurement theory). Thus, for example, the notions of 'comfort' and 'dependability' exist independently of any means of measuring them. While it may be sufficient for our purposes to measure comfort on the simple scale {low, medium, high} this measurement does not replace all existing intuition about comfort and therefore does not re-define it.
4. In some situations we may need to define a very crude

measure of a 'synthetic' attribute. For example, rather than defining an indirect measure of *dependability* using the Laprie decomposition above, it maybe be sufficient to provide a crude direct ordinal scale measure such as {low, moderate, average, high, very high}.

5. Synthetic attributes are ones of whose *definition* we are uncertain (*ambiguous*, or *vague* in fuzzy set terms). This must not be confused with uncertain inference about the attribute (we deal with this key notion of uncertainty in the next section). For example, there is no uncertainty about how to define and measure a person's weight, but if we wanted to predict a person's weight in two years time then that value is uncertain. Conversely, although we may be unsure how to define and measure system safety, there is no uncertainty about the safety of a system that has been built, used, and decommissioned (it is just that we may not agree on how to measure it). What we need to be careful about is to distinguish between the different types of 'uncertainty' that arise in decision problems:
   - uncertainty in meaning—where we have a 'synthetic' criteria that we do not properly define;
   - imprecision—where our measurement process is inaccurate even though it may be well defined;
   - uncertainty in inference.

6. Uncertain criteria and inference

In classical MCDA, once a criterion g is defined (even if it is synthetic in the sense discussed in the previous section) it is assumed that for a given action a the value of $g(a)$ is certain. For example, it is reasonable to assume that the age, price, and engine size, of each of a set of cars that we may wish to buy, are defined with certainty because we have no control over them (unless we are car manufactures). However, in general many key criteria cannot be computed with any kind of certainty. Rather, they require some kind of uncertain inference. Even in our car example a criteria like petrol consumption will be uncertain, being dependent on (among other factors) the speed and road conditions.

Example. In our travel example we had four criteria on which to base our decision about which action we should choose from all pairs $<$*transport type*, *start time*$>$. These criteria were *journey time*, *wait time*, *cost*, and *comfort*. For simplicity, we can assume that the values of *cost* and *comfort* are certain for each possible action. However, *journey time*, and *wait time* are uncertain. For example, *journey time* for a specific choice of action will vary according to whether or not there are *train problems* or *roadworks*, and there is also uncertainty arising from the variability of delays that occur due to rush hour traffic.

Whereas traditional MCDA assumes that all criteria can be measured with certainty, it is clear that any interesting problem will involve key criteria that are inherently uncertain. Having a specific method for handling this uncertainty is where, of course, the BBNs come in.

The BBN should include not just the uncertain criteria but also other factors that can influence the value of a criterion for a given action. Such factors can be thought of as *risk factors*—they often cannot be controlled by the decision-maker. These factors, along with the uncertain criteria themselves, will form the set of nodes in a BBN for predicting the values of the uncertain criteria.

Example. In our travel example the two uncertain criteria *journey time* and *waiting time* are affected by risk factors *train problems*, *roadworks,* and *delay* (due to rush hour traffic). In Fig. 3 we have produced a single BBN that incorporates the uncertain criteria with the factors that impact on them. Notice that the BBN does not include the 'certain' criteria *cost* and *comfort.* The probability tables of the BBN in this example are relevant for travel between one area of London and Heathrow airport; most are derived from empirical data plus a small amount of expert judgement, while some (like *waiting time* and *adjusted journey time*) are simply deterministic functions of their parents. The initialised probability values are shown in Fig. 4. Thus, for example, the probability that there are major roadworks on any day is 0.125, while the probability that the journey time is between 60 and 90 min is 0.33. Table 3 shows part of the probability table that was elicited for the node "nominal delay".

In Fig. 5 we use the BBN of Fig. 3 to calculate values of the uncertain criteria. In this scenario, for a flight departure time of 8.30 we decide to go by car and leave between 6 and 7 AM (the values of these known variables are shown by a dark bar indicating probability = 1). In this scenario we do not have any information about *roadworks* (and hence we use the prior probabilities). The BBN calculates that the probability the journey time is less than 60 min is 0.35. Moreover, the probability that the waiting time is between 30–60 min (which is regarded as 'ideal') is only 0.2895. In fact there is a 0.197 probability that we will miss the flight (waiting time less than 15 min).

In Fig. 6 we specify that, for a flight departure of 8.30 we wish to leave between 7 and 8 and have a waiting time of between 30 and 60 min. In this case the BBN calculations show that, by far, the most probable way to achieve our objective is to travel by train. Suppose, however, that we discover before leaving that there are major train disruptions and so decide to travel by car. Fig. 7 shows what happens in this scenario: with a probability of 0.8573 the waiting time will be too low (less than 15 min) -and we will miss the flight. In Fig. 8 we use the same scenario as Fig. 7, but now we wish to *ensure* a waiting time of 30–60 min. The resulting BBN calculations show that, with probability 0.7041 we need to leave between 6 and 7.

Table 3
Part of the probability table for the node "nominal delay"

Transport type	Car								
train problems	Negligible			Minor			Major		
roadworks	Negligible	Minor	Major	Negligible	Minor	Major	Negligible	Minor	Major
0–2	1	0.2	0	1	0.2	0	1	0.2	0
2–10	0	0.8	0.04	0	0.8	0.04	0	0.8	0.04
10–20	0	0	0.44	0	0	0.44	0	0	0.44
20–30	0	0	0.36	0	0	0.36	0	0	0.36
30–40	0	0	0.16	0	0	0.16	0	0	0.16
40–50	0	0	0	0	0	0	0	0	0
50–60	0	0	0	0	0	0	0	0	0
60–70	0	0	0	0	0	0	0	0	0
70–infinity	0	0	0	0	0	0	0	0	0

7. A real example: making decisions about component safety

To confirm that the approach described above is both practically possible and practically usable in real applications, we return to the safety assessment example shown in Fig. 1. The BBN shown there is a sanitized and simplified version of a BBN that was developed during 1998–2000 by the authors in a commercial project for a major transportation company. The development involved extensive elicitation with domain experts and was subject to rigorous internal validation. The purpose of the BBN is to

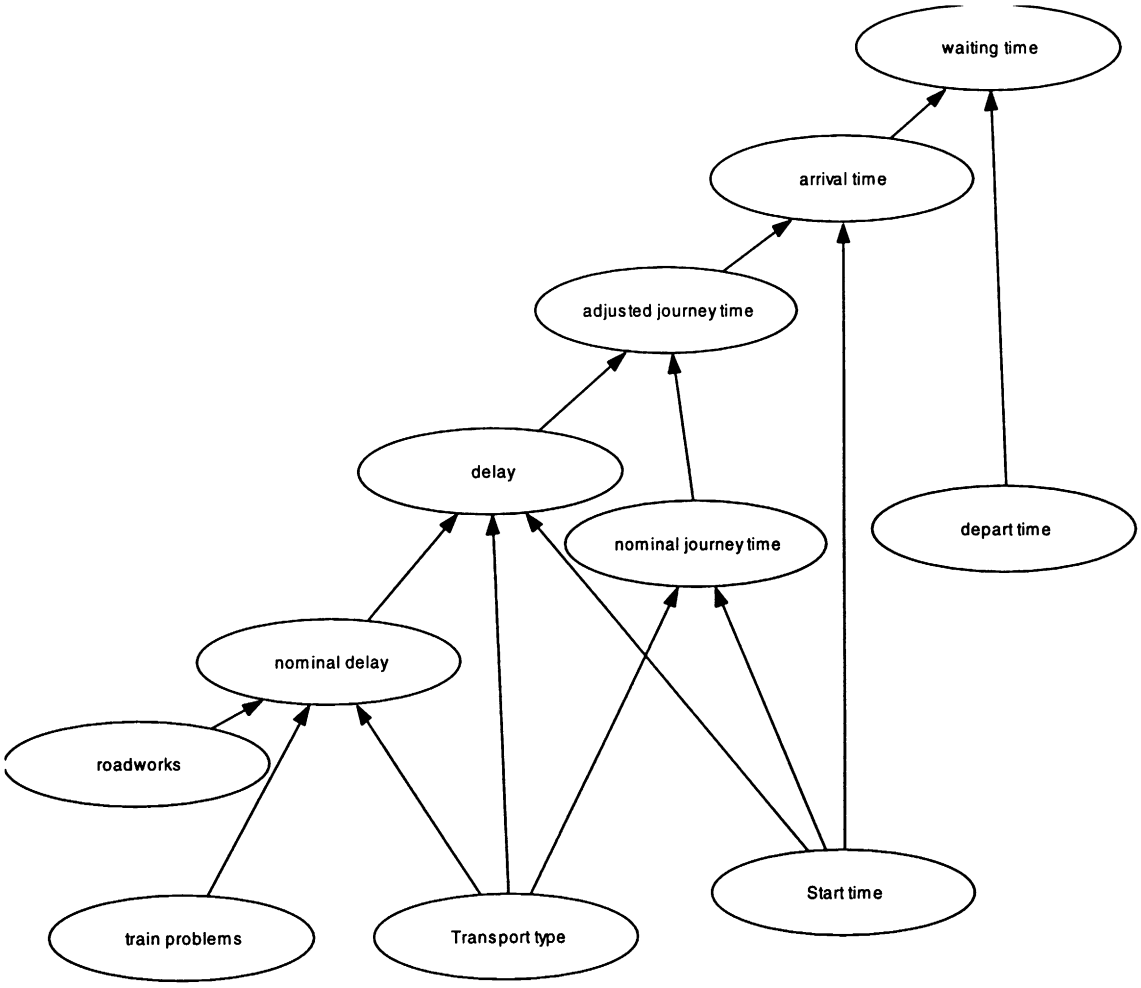

Fig. 3. BBN for predicting the uncertain criteria in the travel example.

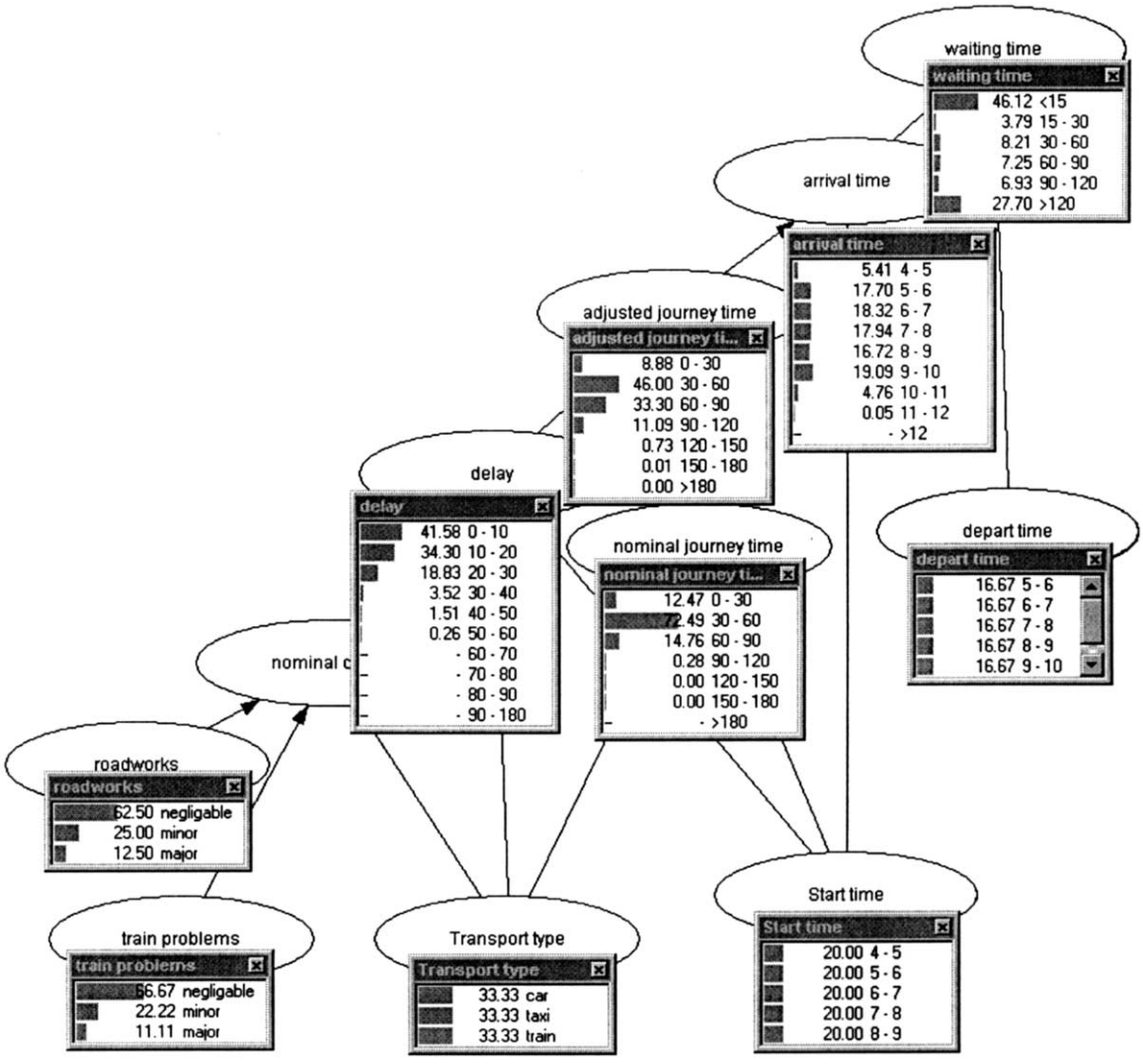

Fig. 4. BBN in initialised state.

determine whether electronic components available from outside contractors are sufficiently safe to deploy in the transport system. The node 'safety' was actually defined in terms of financial loss. For example, a failure resulting in loss of life obviously incurs a much higher financial loss than a failure that can only ever cause a short delay. A component is considered 'safe' if the probability of a high financial loss is below a certain threshold. For obvious reasons we have simplified and sanitized the node "safety", replacing its range of state values with the set of values "very high", "high", "medium", "low", and "very low". The BBN shown here is also massively simplified—several of the nodes are actually abstract nodes in the sense that in the full model they were themselves decomposed into lower level subnets (the full model contained 125 nodes in total).

The main uncertain criterion in this decision problem is *safety*. In this context *safety* is a synthetic node in the sense of Section 5, as it is defined by the *criticality in the environment* of the component and the *frequency of failure*. For a given frequency of failure the greater the criticality, in terms of financial or human loss, the greater the overall risk and vice versa. A fragment of the probability table for the 'safety' node is shown in Table 4.

The 'frequency of failure' is influenced by the 'degree to which the solution matches requirements' and the degree of 'fault tolerance'. Here we can think of 'degree to which solution matches requirements' as the extent to which faults exist in the software which could, if activated cause failures. Of course, the extent to which these will cause failures will depend on the depth of fault tolerance and error recovery designed into the system. In this way we can think of 'fault tolerance' reducing the effects of faults in a way that enhances safety. However, this may have a detrimental affect on reliability as fault tolerance may impinge on overall reliability (if one version from an n-version system fails the overall reliability falls by a commensurate amount; also if the errors are encountered the system may fail safe resulting in a loss of service).

Table 4
Fragment of probability table for 'safety' node

Criticality in the environment	Very low					Low				
Frequency of failure	Very low	Low	Medium	High	Very high	Very low	Low	Medium	High	Very high
Very high	1	0.6944	0.4064	0.2864	0.2224	0.6944	0	0	0	
High	0	0.3056	0.5936	0.7136	0.6688	0.3056	1	0.6224	0.1504	
Medium	0	0	0	0	0.1088	0	0	0.3776	0.8496	0.94
Low	0	0	0	0	0	0	0	0	0	0.06
Very low	0	0	0	0	0	0	0	0	0	0

The extent to which a correct solution can be constructed depends on the correctness and appropriateness of the requirement specification. If this is wrong then the resulting solution will be wrong (unless the supplier corrects the mistakes during development). Hence the 'degree to which solution matches requirements' depends on whether 'requirements match true needs' and whether the 'supplier is matched to the solution and problem'. The quality of the supplier could not reasonably be expected to make up for shortfalls in the requirements. Likewise, a poor supplier may not correctly implement a perfect requirement specification.

The degree of 'fault tolerance' will depend on whether the 'supplier is matched to the solution and problem'. Good quality suppliers, knowledgeable in the domain will tend to build in fault tolerance and other defensive mechanisms. Also good quality suppliers will produce solution which are

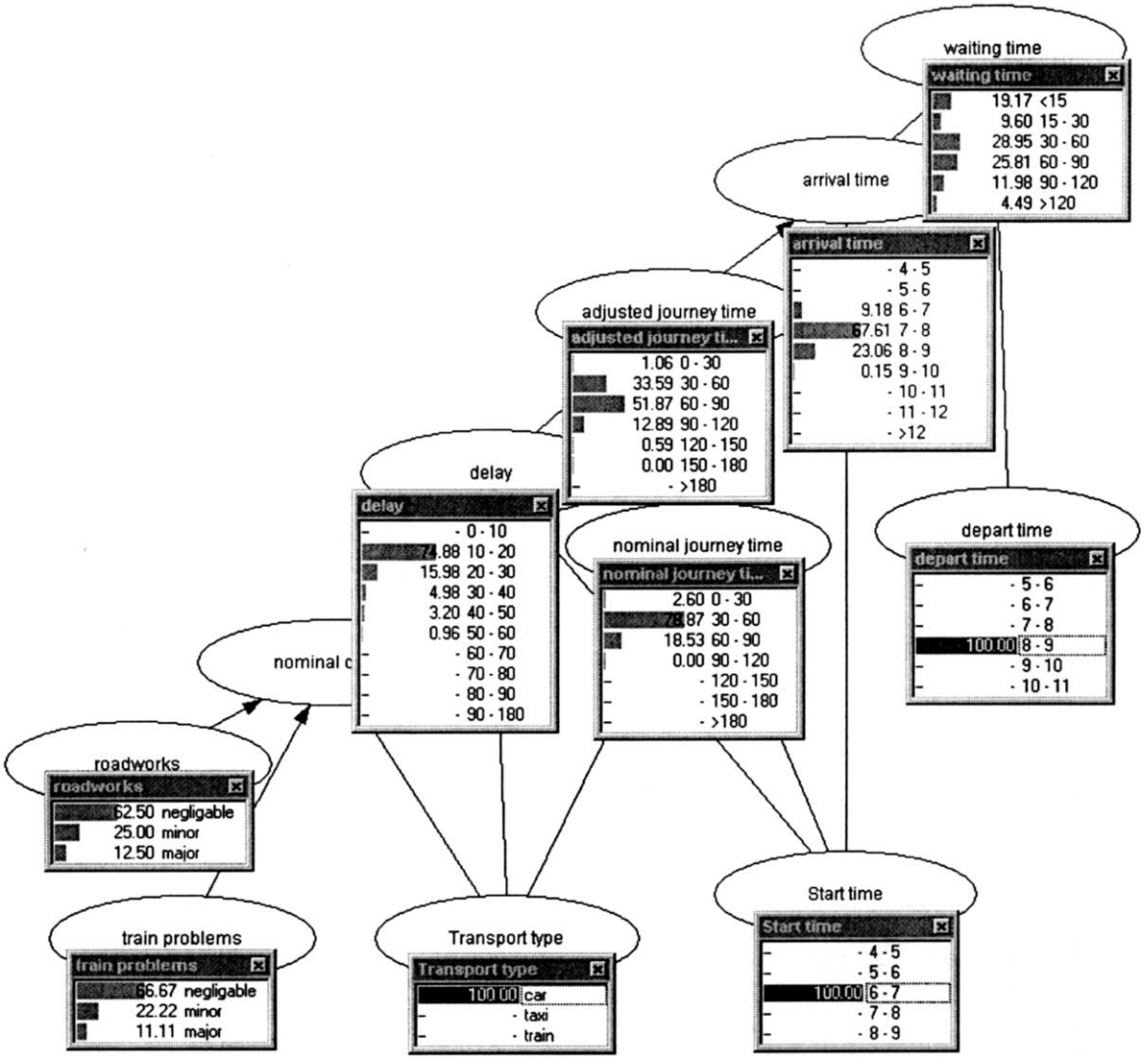

Fig. 5. Calculating values of uncertain criteria when we travel by car for 8.30 flight leaving between 6 and 7.

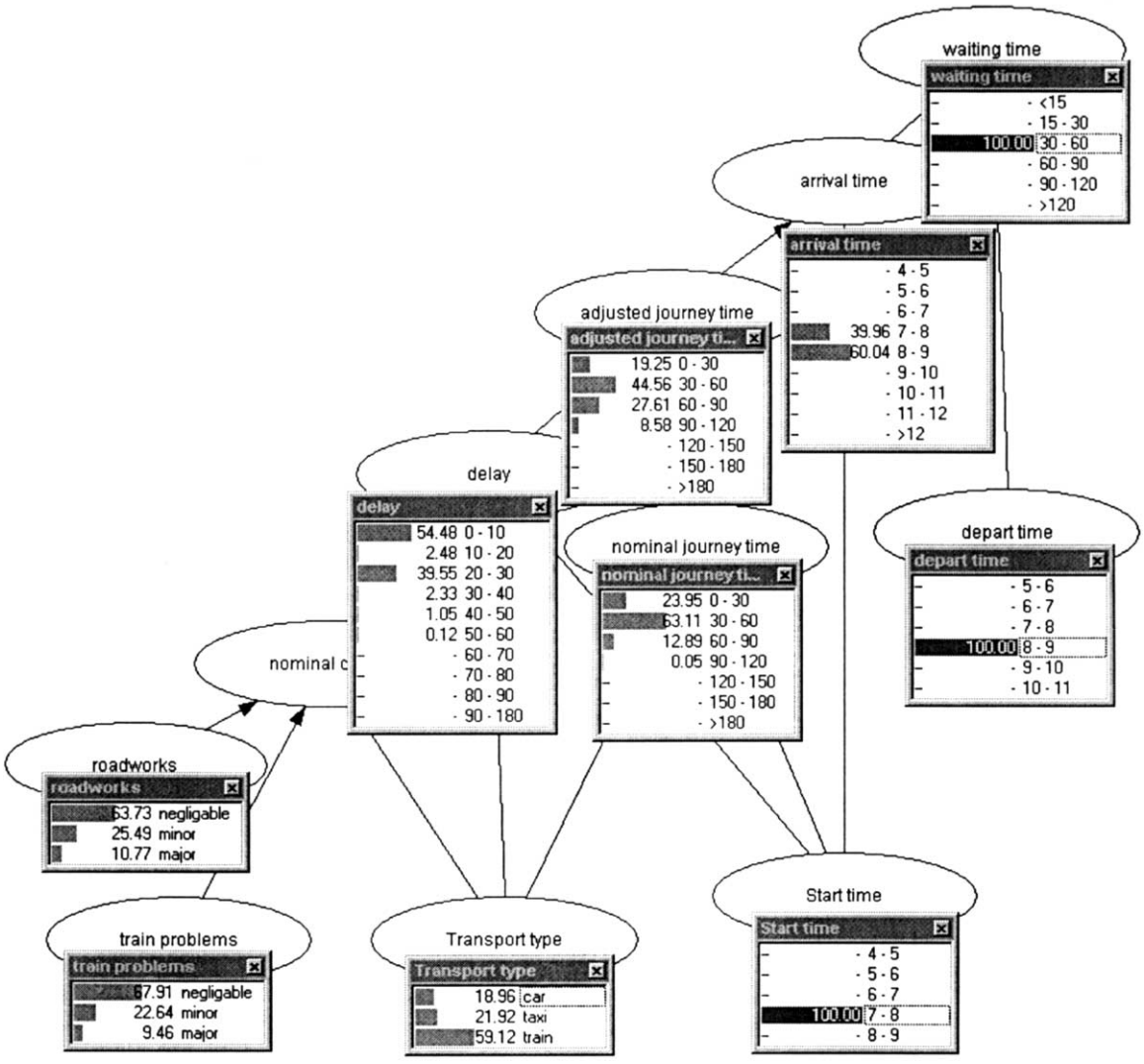

Fig. 6. Now we specify that we wish to wait between 30-6- min and start out between 7 and 8 for our 8.30 flight.

verifiable. The degree of verifiability is represented by 'comprehensibility of the solution'.

Incomprehensible solutions will be difficult to inspect, review and test throughout the life-cycle no matter how good the testing team is. The 'quality of testing' (a synthetic criterion) will also depend on the 'competence of the assurance team'. Incompetent inspectors and testers will be less likely to find faults should they exist.

The 'number of defects found in testing' will depend on 'testing quality' (poor testing will uncover few defects) and 'safety' (the number of defects found is limited by the number of residual defects to find). All testing is imperfect so the test results will always be an inaccurate estimate of the actual 'safety'.

In addition to the synthetic nodes 'safety', and 'testing quality' the BBN has an additional synthetic node 'problem propensity'. These synthetic nodes help combine parent node values into a 'combined score' according to some rule. The rule chosen represents the expected relationship between the parent nodes. Using these rules helps reduce the number of probability values that need to be estimated for the child nodes. For example, by using the 'problem propensity' node we reduce the number of probability numbers to be estimated for 'fault tolerance' from 125 down to 25—a much more manageable figure).

To give some idea of how this BBN is used in decision analysis, Fig. 9 shows a typical scenario in which some specific observations (dark bars showing 100%) have been made. For example, here we know that the 'criticality in the environment' is *medium*, the 'requirements matches needs' is *high*, and the *supplier* is medium.

With these observations the BBN computes that the probability that the safety is very high is 0.127. What this means in real terms is that, based on the previous data and expert judgement the probability that a component with similar observations has very high safety is 0.127.

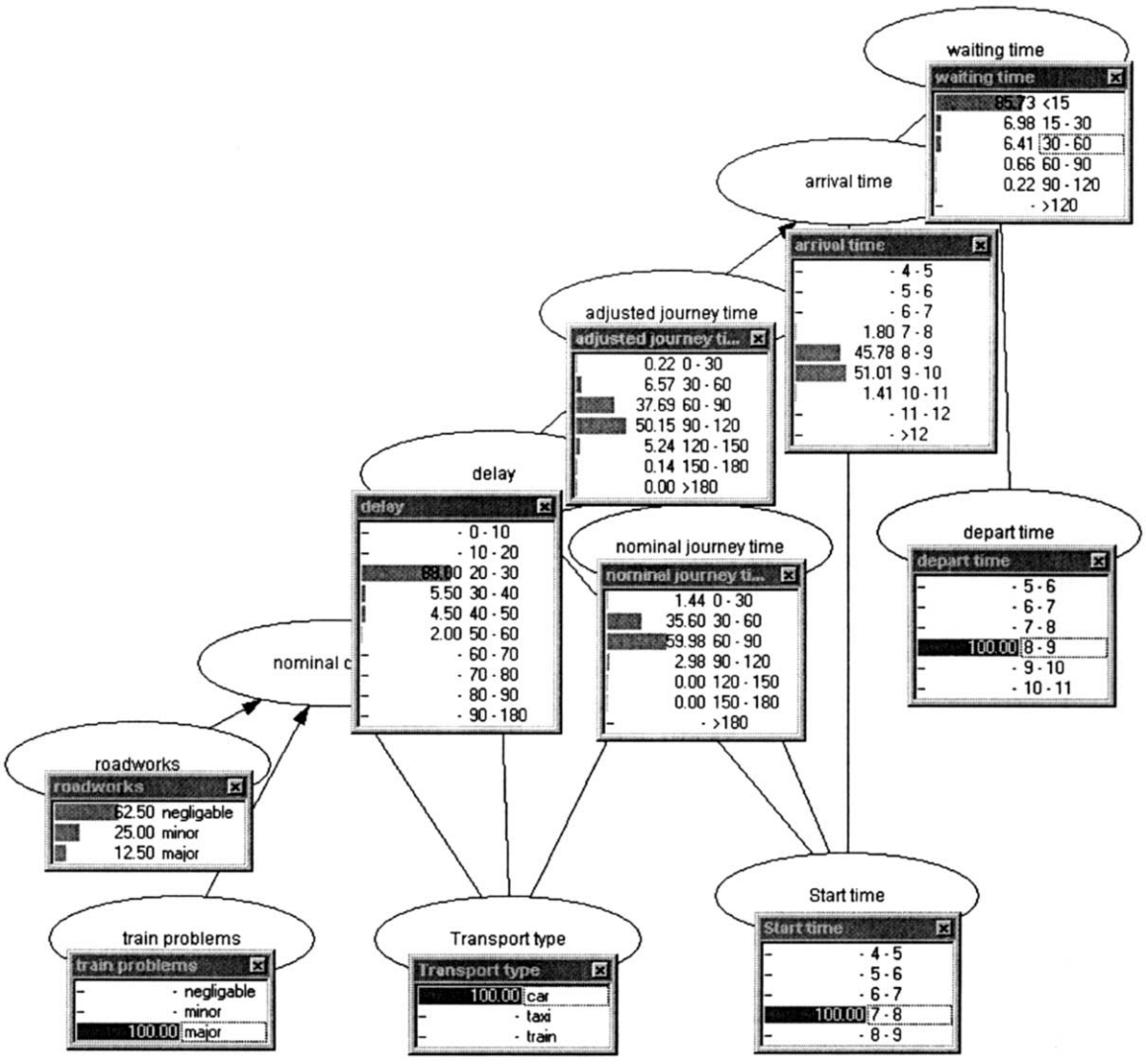

Fig. 7. Knowing there are major train problems we decide to go by car. Unfortunately, the waiting time will probably be too low - less than 15 min (with probability 0.8573) means we miss the flight.

Let us suppose that a component can only be accepted if the probability of safety being at least 'high' is at least 0.99. With the information above this probability is only 0.547, so the component cannot be accepted. What the company does in such circumstances is to perform independent testing on the component.

Suppose that such testing reveals a 'very low' number of defects (which is the best answer we could hope for). Fig. 10 shows the computed BBN when we enter this observation. We can now see that the belief in the safety being at least 'high' has increased to 0.68; this is still well short of the safety requirement, and on this information alone we would therefore have to reject the component. The reason why the very good evidence about defects found in testing has not increased our belief in safety sufficiently is that at this stage we know nothing about the testing quality.

Suppose, for example, that we discover the testing quality is very low. Fig. 11 shows the results of entering this information. Our belief about safety actually reverts back almost exactly to our belief before we saw any testing information. In other words the very low number of defects has been *explained away* by the very low quality of testing (bad testing will reveal no defects). However, suppose we find out that the testing quality is very high as shown in Fig. 12. In this case the probability that safety is at least 'high' is now 99.96 and we can accept the component.

8. The combined approach

The approach to solving decision problems that we are proposing has a close analogy with GQM (goal question metric) [17]. You start by asking what are your goals—that is, the objective for your decision. Next you have to consider the perspective (for example, the

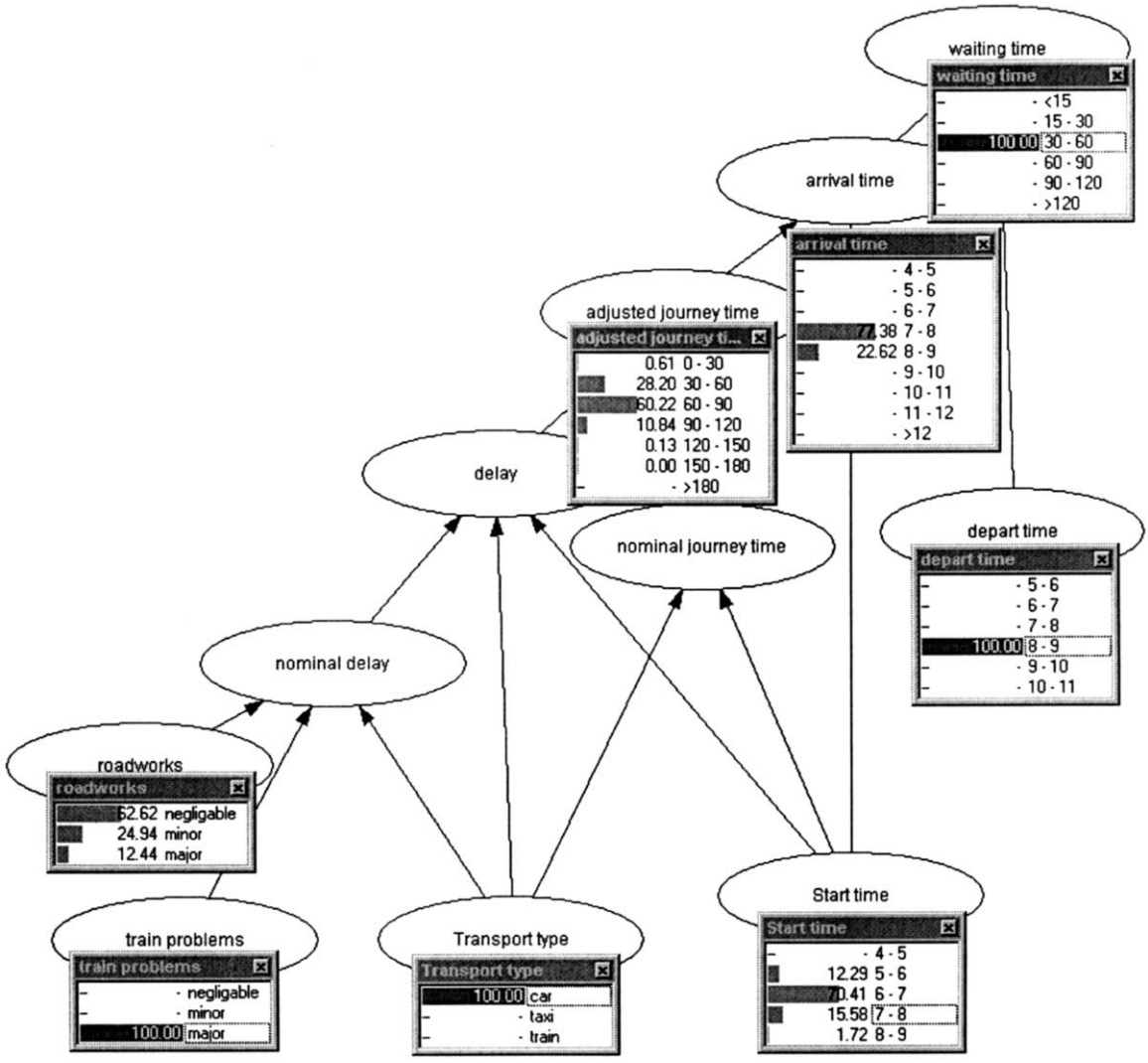

Fig. 8. To ensure a waiting time of between 30-60 min when travelling by car, the most probable option is to leave between 6 and 7.

Regulator as opposed to the Developer). Next you ask 'questions', which we think of as identifying the set of possible actions and then the set of criteria that distinguish these actions. At this point traditional GQM would simply insist that you define the underlying measures for your chosen criteria and traditional MCDA would then provide a means of combining the resulting measures for each action and provide a means of ranking the actions as a result. The key difference we have is that while some criteria may be certain, and hence depend on a traditional approach to measurement, many key criteria will require uncertain inference. These criteria will depend on various factors that we have to identify. Having identified them we use them to make predictions of the values of the uncertain criteria for the different actions. We do this by using a BBN. This enables us to compute values for each criterion for a given action and we can then apply traditional MCDA techniques to combine the values and rank the actions. The process is shown schematically in Fig. 13.

The combined approach works with any MCDA method. The book [12] provides an overview of such methods. In the example that follows (which builds on the travel example) we can use (for simplicity) a crude multi-attribute utility approach. In such an approach each criteria g_i is assumed to be measurable on a ratio scale, and hence each can be mapped into a common interval, say [0,1] where 0 represents the 'worst' value for the criteria and 1 represents the 'best'.

Example. In Table 5 the five criteria in our travel example (*comfort*, *cost*, *start time*, *journey time*, and *waiting time*) are given values that are mapped into [0,1]. Thus, for example, the 'best' value of *comfort* that we can achieve for the three specified transport types is 0.8 (for taxi); the best value for journey time we can achieve is 1 (when journey

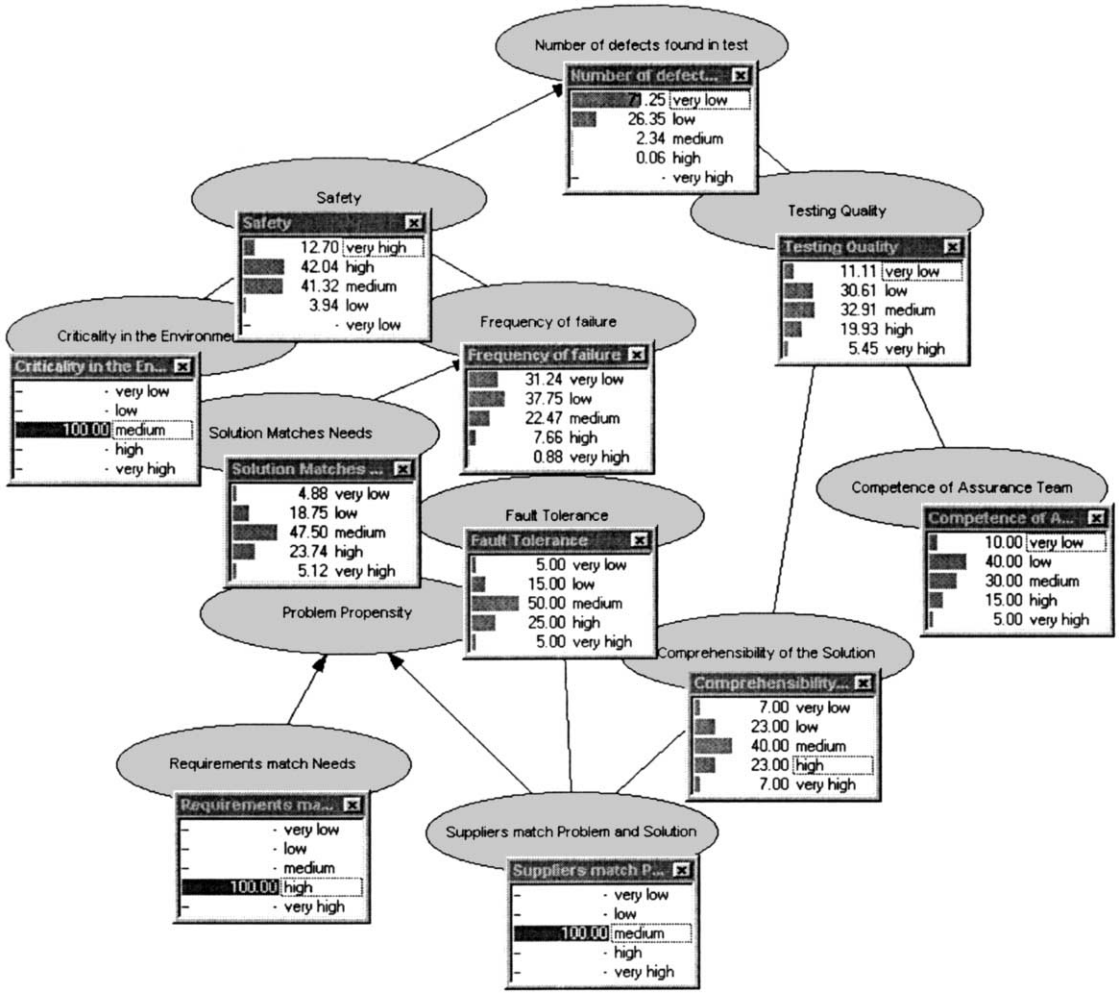

Fig. 9. A particular scenario for the safety BBN.

time is less than 60 min). The table for waiting time also includes the reject value for waiting time <15 min; this is a constraint on the problem (since <15 min means we miss the flight).

Each criteria g_i is then given a utility weighting u_i that represents the relative importance of each attribute for the given decision problem. The overall 'utility' $U(a)$ of an action a is then simply the weighted sum $\sum u_i g_i(a)$.

Example. In Table 6 we have specified utility weightings to the criteria in the travel example. For example, *start time* is the most important criteria (weighted 2.5 times greater than *comfort*). In the table the rows represent the possible actions we can take—recall that an action here is a pair ⟨*transport type*, *start time*⟩. The values of the certain criteria (*comfort* and *cost*) are taken straight from Table 6. To obtain the values for the uncertain criteria (*journey time, start time, waiting*

Table 5
Criteria mapped to values in [0,1] interval

	comfort	cost
car	0.7	0.5
taxi	0.8	0.2
train	0.5	0.7

start time	
4-5	0.2
5-6	0.3
6-7	0.5
7-8	0.9
8-9	1

waiting time	
<15 minutes	**reject**
15-30	0.1
30-60	1
60-90	0.7
90-120	0.5
>120	0.3

journey time	
0-60	1
60-90	0.8
90-120	0.5
120-150	0.2
>150	0.1

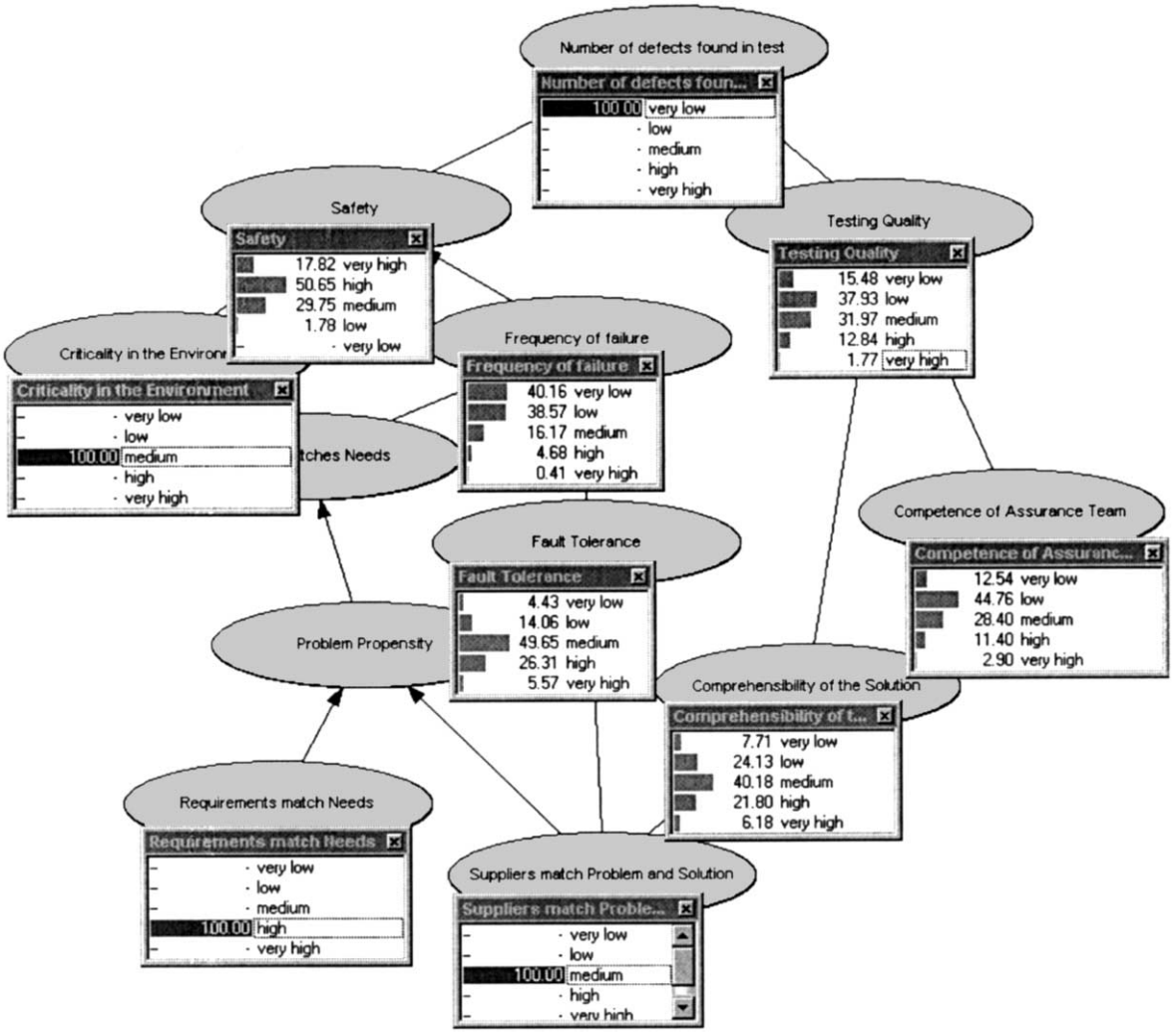

Fig. 10. Results of independent testing are entered.

Table 6
Weighted utilities of different actions

	Comfort (1)	Cost (1.5)	Journey time (2)	Start time (2.5)	Waiting time (2)	Total
(Car, 4–5)	0.6	0.5	1	0.2	0.3	4.45
(Taxi, 4–5)	0.8	0.2	1	0.2	0.3	4.2
(Train, 4–5)	0.2	0.7	1	0.2	0.3	4.35
(Car, 5–6)	0.6	0.5	0.8	0.3	0.3	4.3
(Taxi, 5–6)	0.8	0.2	0.8	0.3	0.3	4.05
(Train, 5–6)	0.2	0.7	1	0.3	0.3	4.6
(Car, 6–7)	0.6	0.5	0.8	0.5	1	6.2
(Taxi, 6–7)	0.8	0.2	0.8	0.5	1	5.95
(Train, 6–7)	0.2	0.7	1	0.5	0.7	5.9
(Car, 7–8)	0.6	0.5	0.5	0.9	0	Reject
(Taxi, 7–-8)	0.8	0.2	0.5	0.9	0	Reject
(Train, 7–8)	0.2	0.7	1	0.9	0.1	5.7

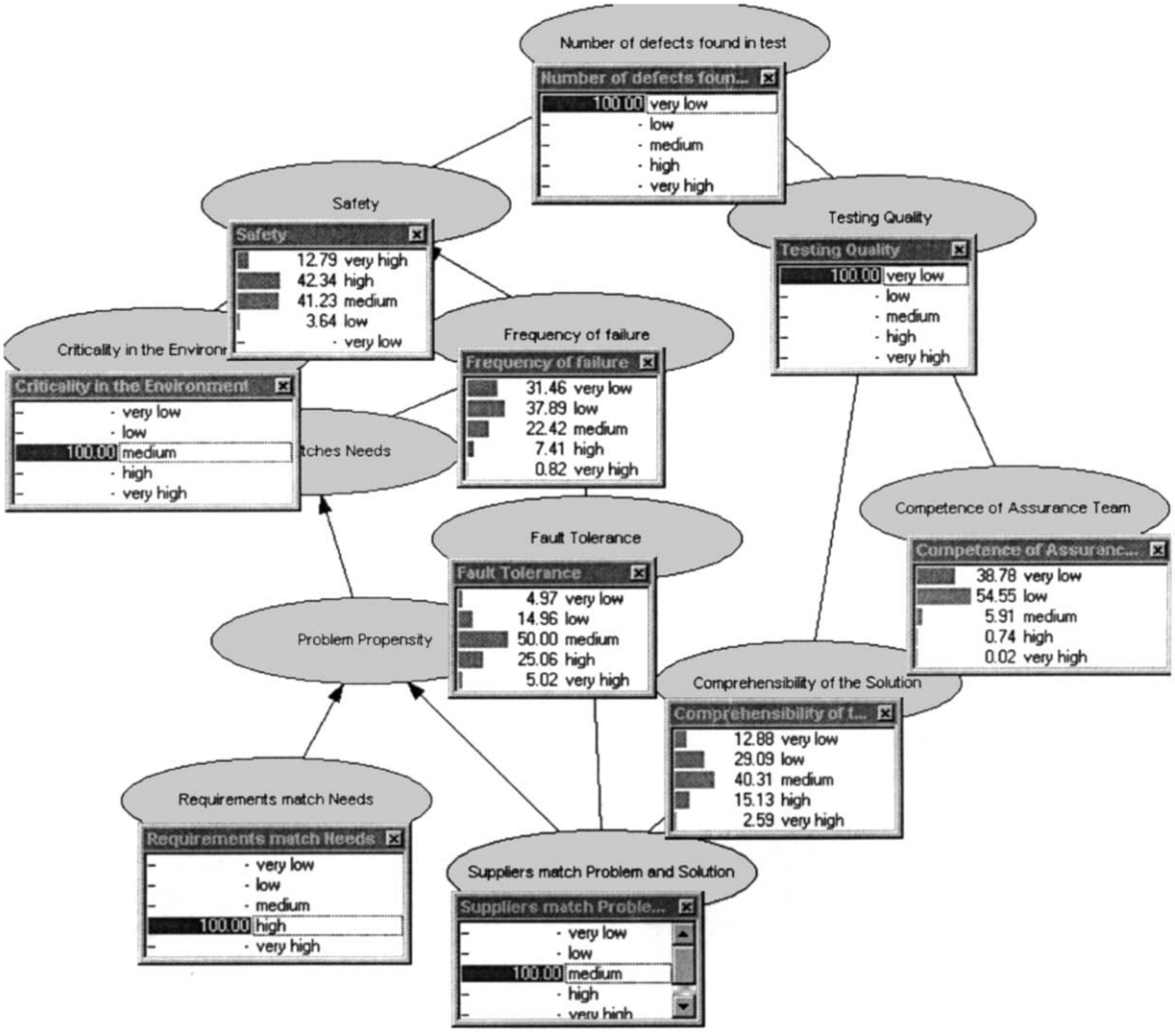

Fig. 11. Now we enter information about testing quality.

time) we get the *mean* values from the BBN when we enter the specified ⟨*transport type, start time*⟩ and then transform these using the values in Table 3. The final column is then the weighted sum. In this example, the rational decision should be to travel by car leaving between 6 and 7. Obviously if we discover in advance that there are roadworks then the BBN would provide different values for the uncertain criteria and a different preferred outcome.

Because our method (unlike usual MCDA methods) provides a complete description of uncertain criteria in terms of their probability distribution we have great flexibility over how to use this in calculating $U(a)$. In Table 4 we simply used the mean of each uncertain criteria $g_i(a)$ to arrive at a point value for $U(a)$. We could also use the distributions of uncertain criteria as additional criteria. For example, a risk-averse person might feel that the *variability* of the waiting time is a crucial criterion. If so, we could calculate the variance of the node *waiting time* from the BBN and include it as an additional citerion.

9. Conclusions

BBNs help us to make predictions about uncertain factors like safety of a proposed system. While this is extremely important it is only one component of a broader decision making process when there are multiple 'success' factors to consider. In this article we have described the broader decision making context and have provided a rigorous method for tackling it. In summary this method consists of the following:

1. Agree on the *objective* for your decision problem.
2. Make sure you know *from whose perspective* the problem must be solved. Thus, identify carefully both the *decision maker* and the *stakeholders*.
3. Identify the set of possible *actions* that will form the set of alternatives available to you.

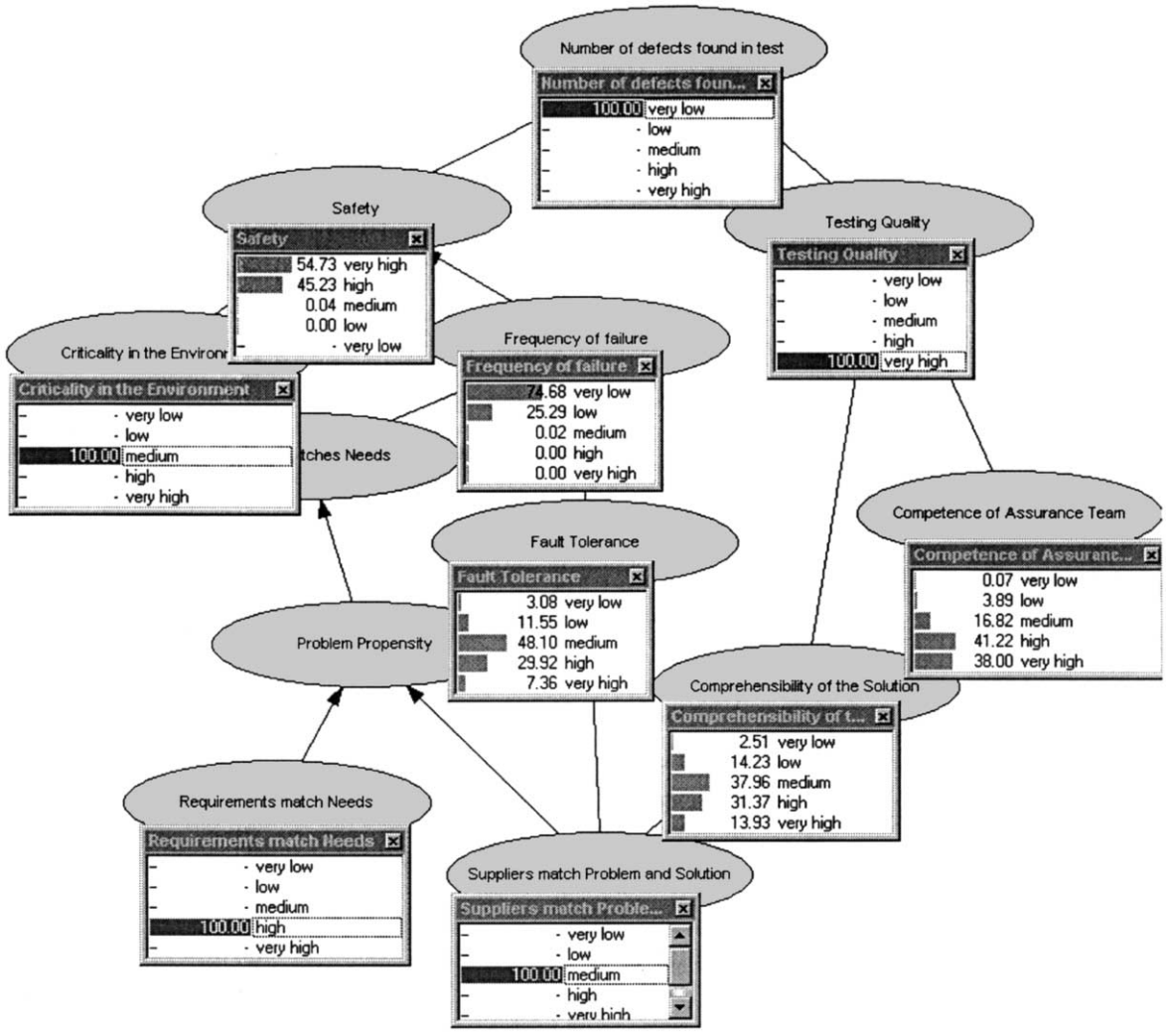

Fig. 12. We find out the testing quality is very high.

4. Identify the set of *criteria*, that is the attributes of actions, which will determine your choice.
5. Identify any fixed constraints, that is properties of criteria that must be satisfied for any chosen action.
6. Determine which criteria are uncertain (that is, can only be calculated for a given action using uncertain inference) and which criteria can be calculated with certainty.
7. For the certain criteria ensure that you have appropriate definitions that enable an unambiguous mapping of actions into a totally ordered set. There is no harm if the ordered set is a simple ordinal scale as long as clear rules are defined for the mapping. If a criterion is vague or complex, it may be necessary to decompose it into lower level attributes. However, all definitions of the certain criteria (including any decomposition) must be done separately from the BBN.
8. For the uncertain criteria, identify the factors that will affect them. There will generally be external factors that you cannot control and some internal ones that you can control. Having identified them construct one or more BBNs for the various factors and uncertain criteria.
9. As a result of steps 7 and 8 you will be able calculate a value (within some probability bounds in the case of the uncertain criteria) for each criterion for a given action. This means that you can apply traditional MCDA techniques to combine the values for a given action and then to rank the set of actions. In the case of the uncertain criteria you could, for example, apply values for 'most likely' as well as the upper and lower bounds. If the result of the MCDA analysis produces a unique 'best' action which satisfies all of the defined constraints then you are done. If not you will have to relax various constraints or introduce new actions (MCDA deals with these issues and it is beyond the scope of this paper).

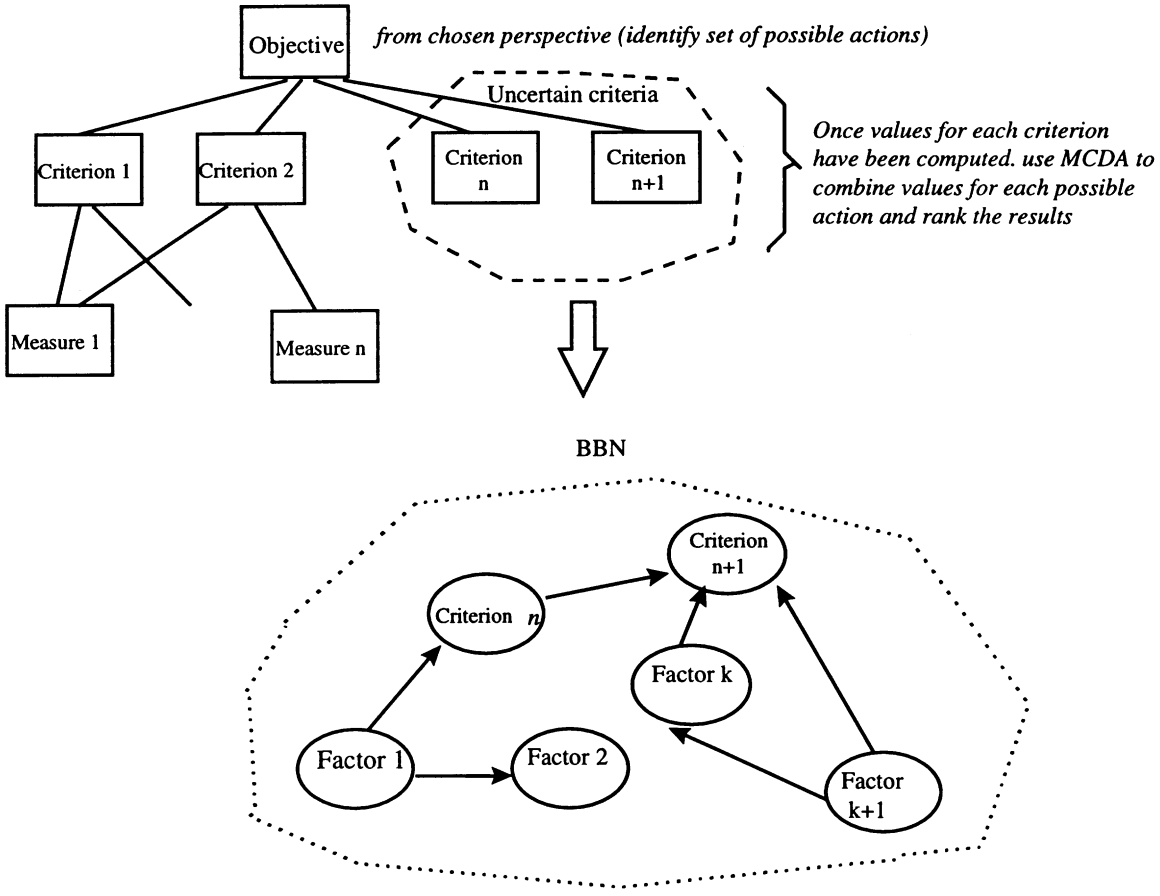

Fig. 13. How the BBN approach fits in with GQM and MCDA.

Acknowledgements

The work in this article was done in part supported by the ESPRIT project 22187 SERENE and the EPSRC project IMPRESS (grant GR/L06683). We are indebted to our partners in these projects for their support and advice. The case study and related material were undertaken as part of a commercial project by Agena, and we acknowledge the valuable input of Agena's client for this. Finally we acknowledge the helpful comments of the anonymous referees that have led to a significantly improved version of the paper.

References

[1] D. Heckerman, A. Mamdani, M. Wellman, Real-world applications of Bayesian networks, Communications of ACM 38 (3) (1995) 25–26.
[2] F.V. Jensen, An Introduction to Bayesian Networks, UCL Press, 1996.
[3] S.L. Lauritzen, D.J. Spiegelhalter, Local computations with probabilities on graphical structures and their application to expert systems (with discussion), Journal of the Royal Statistical Society B 50 (2) (1988) 157–224.
[4] J. Pearl, Probabilistic reasoning in intelligent systems, Morgan Kaufmann, Palo Alto, CA, 1988.
[5] SERENE consortium, SERENE (SafEty and Risk Evaluation using bayesian Nets): Method Manual, ESPRIT Project 22187, http://www.csr.city.ac.uk/people/norman.fenton/serene.htm, 1999.
[6] K.A. Delic, F. Mazzanti, L. Strigini, Formalising a software safety case via belief networks, Proceedings DCCA-6, Sixth IFIP International Working Conference on Dependable Computing for critical Appliations, Garmisch-Partenkirchen, Germany, March 1997.
[7] B. Littlewood, L. Strigini, D. Wright, N.E. Fenton, M. Neil, 'Bayesian Belief Networks for Safety Assessment of Computer-based Systems', in System Performance Evaluation Methodologies and Applications (Ed: Gelenbe E), CRC Press, Boca Raton ISBN 0-8493-2357-6, pp. 349–364, 2000.
[8] N.E. Fenton, B. Littlewood, M. Neil, L. Strigini, A. Sutcliffe, D. Wright, Assessing dependability of safety critical systems using diverse evidence, IEE Proceedings Software Engineering 145 (1) (1998) 35–39.
[9] TRACS (Transport Reliability Assessment & Calculation System): Overview, DERA project E20262, http://www.agena.co.uk/tracs/index.html, 1999
[10] N.D.C. Lewis, N. Fenton, M. Neil, Uncertainty, software quality and statistical process control, submitted for publication.
[11] N.E. Fenton, M. Neil, A critique of software defect prediction models, IEEE Transactions on Software Engineering 25 (5) (1999) 675–689.

[12] P. Vincke, Multicriteria Decision Aid, Wiley, New York, 1992.
[13] T. Saaty, The Analytic Hierarchy Process, McGraw Hill, New York, 1980.
[14] F.S. Roberts, Measurement Theory with Applications to Decision Making, Utility, and the Social Sciences, Addison-Wesley, Reading, MA, 1979.
[15] N.E. Fenton, S.L. Pfleeger, Software Metrics: A Rigorous and Practical Approach, 2nd ed, International Thomson Computer Press, 1996.
[16] J.-C. Laprie (Ed.), Dependability: basic concepts and terminology Springer, Berlin, 1992.
[17] V.R. Basili, H.D. Rombach, The TAME project: towards improvement-oriented software environments, IEEE Transactions on Software Engineering 14 (6) (1988) 758–773.
[18] M. Neil, N.E. Fenton, L. Nielsen, Building large-scale Bayesian networks, accepted for publication Knowledge Engineering Review, 2000.