Article

Putting value of information theory into practice: a methodology for building sequential decision support systems

Silvano Mussi

CILEA (Interuniversity Consortium for Information and Communication Technologies), via R. Sanzio 4, 20090 Segrate-Mi, Italy
E-mail: mussi@cilea.it

Abstract: *The paper presents a methodology for building sequential decision support systems based on decision theory using value of information (for short, DT-VOI based SDSSs). DT-VOI based SDSSs support decision-makers in difficult problems of sequential decision-making. In particular we consider the problem of building DT-VOI based SDSSs which are capable of supporting decisions in critical situations where (1) making a decision entails knowing the states of some critical hypotheses, and such knowledge is acquired by performing suitable tests; (2) test outcomes are uncertain; (3) performing a test entails, in general, some drawbacks, so that a trade-off exists between such drawbacks and the value of the information provided by the test; (4) performing a test has the side-effect that it changes the expected benefit from performing other tests; (5) exceptional situations alter probability and utility default values.*

Keywords: sequential decision-making, decision theory, Bayesian networks, sequential diagnosis, knowledge engineering, influence diagrams

1. Introduction

The last decade has seen a growing importance of sequential decision support systems based on decision theory using value of information (DT-VOI based SDSSs). A major stimulus in developing them has come from decision analysis (von Winterfeldt & Edwards, 1986) and Bayesian networks (Pearl, 1988; Jensen, 1996, 2001). Bayesian networks help decision-makers avoid stereotypical deviations (biases) from the axioms of probability (Kahneman *et al.*, 1982) and have been demonstrated to be a powerful modelling tool in many real cases (Kalagnanam & Henrion, 1990; Heckerman *et al.*, 1992), A DT-VOI based SDSS is basically an expert system which, by applying decision theory with value of information to the problem at hand, supports the decision-maker in difficult sequential decision-making problems under conditions of pervasive uncertainty.[1]

The proposal addresses a complex sequential decision scenario such as the following. The decision-maker has to sequentially choose, among a set of alternative target actions (which in the following will be called measures[2]), the best one to perform next. Each measure entails both desired consequences (called target effects), which represent the purpose of the measure, and some undesired side-effects. The current utilities of target effects depend on the current states of some hypotheses (called target hypotheses). In order to establish the current states of the target hypotheses, the decision-maker has at his/her disposal a set of alternative tests.[3] Each test on one hand provides a piece of information having a certain value, and on the other entails some undesired side-effects. Finally, let us add the fact that the scenario is pervaded by uncertainty: both test outcomes and knowledge about the world are uncertain. Given all this, what to do next? Which test to perform next? The proposed methodology shows how to build in practice an SDSS able to answer these questions. In particular, it facilitates the construction of both the knowledge base and the inference engine of a DT-VOI based SDSS, and addresses designers of SDSSs applied to the field of diagnosis (in medicine, electromechanical devices, industrial plants, ecological systems, financial systems etc.).

The work presented in this paper is a continuation of and a complement to the work presented in Mussi (2002) where background concepts of sequential decision theoretic models are presented. In fact, the present paper shows how to put those concepts into practice, in this way providing the reader with a view of the sequential decision-making problem that integrates the abstract level of general concepts with the concrete level concerning the construction of SDSSs. This will contribute to a deeper under-

[1] *A DT-VOI based sequential decision-making model is a particular type of the more general sequential decision-making model presented in Puterman (1994).*
[2] *We use the term 'measure' in the same way as in the sentence 'Measures will be taken to combat crime'.*
[3] *Note that both measures and tests are actions.*

standing of the impact that sequential decision models have on designing expert systems.[4]

The paper is organized in three parts. Section 2 shows how to build an SDSS knowledge base step by step; Section 3 shows how to build an SDSS inference engine; in Section 4 the proposal is discussed in the context of related work.

2. Building the knowledge base

In the following six sections we illustrate how to build an SDSS knowledge base in six steps. Each step will be described at an abstract level; however, for clarity and concreteness, we will build step by step the knowledge base of an illustrative example in the field of diagnosis.[5] For simplicity, the number of variables that have been considered in the example is small[6] with respect to the number of variables that would be necessary to represent the whole problem reality; however, the example is sufficiently rich to illustrate how the methodology works. The example will be referred to in the following as 'the example problem'.[7]

The representation of a real-world sequential decision-making problem involves various kinds of knowledge. Let us represent and integrate all these kinds of knowledge in a unique Bayesian network made up of a set of sub-networks, a sub-network for each type of knowledge. To be more precise, let us identify the following types of knowledge:

(1) knowledge concerning causal relations between target hypotheses and test outcomes and other evidential manifestations. This knowledge is represented in a sub-network that will be called DN (diagnostic network).

(2) knowledge concerning the set, say $B = \{b1, b2, \ldots, nb\}$, of possible tests (where nb is the dummy test which stands for 'no test', i.e. 'do nothing') and, for each test $b \in B$, the set J_b of its possible outcomes and the set of its side-effects. Let us represent the set B by creating a special node B whose states are $\{b1, b2, \ldots, nb\}$. Let us represent the test side-effects in a sub-network that will be called SB (side-effect network for the set of tests B).

(3) knowledge concerning the set, say $A = \{a1, a2, \ldots, na\}$, of possible measures (where na is the dummy measure which stands for 'no measure', i.e. 'do nothing') and, for each measure $a \in A$, the set of its target effects and the set of its side-effects. Let us represent the set A by creating a special node A whose states are $\{a1, a2, \ldots, na\}$. Let us represent measure target effects and side-effects in two sub-networks that will be called TN (target-effect network for the set of measures A) and SA (side-effect network for the set of measures A) respectively.

(4) knowledge concerning prior and conditional probabilities, represented according to the Bayesian network standard.

(5) knowledge concerning the utilities of the consequences of both measures and tests. To represent this knowledge we create a set of special nodes that will be called utility nodes. To each node representing a consequence of an action (be it a test or a measure), we attach, as a child node, a utility node. Utility nodes are binary and their values are defined according to utility theory (as will be explained in the following).

Let us now face the problem of how to build in practice each individual sub-network.

Step 1: Building the DN

The DN includes a set of H nodes representing target hypotheses, a set of Sy nodes representing directing observable evidential manifestations and a set of J nodes for representing test outcomes: each J_i node relates to a test b_i in that the set of possible states of J_i represents the set of possible outcomes of b_i. The structure of the sub-network is defined by a set of causal paths from root nodes (H nodes) to leaf nodes (Sy and J nodes), each path possibly including intermediate nodes. In cases involving large Bayesian networks, the construction of the DN can be carried out with the aid of the similarity network technique (Heckerman, 1991).

Let us start with the example problem. There is a patient for whom the presence of the heart malfunction called high risk ventricular pre-excitation (for short, VP) is suspected because the patient suffers from tachycardia (for short, TA). In order to verify if VP is present or not, the physician has at his/her disposal two alternative tests: the Holter test (HT) and the electrophysiological test (ET). In this case the DN is built as illustrated in Figure 1.

Step 2: Building TN for the set of measures A

Let us consider measures affecting target hypotheses,[8] i.e. measures whose target effects consist in varying the probability distributions of some target hypotheses. For example, let us consider the disease 'tonsillitis' and the therapy 'antibiotic treatment'. If, before performing the therapy, the probability of 'tonsillitis is present' is $p > 0$, after the therapy is performed the probability of 'tonsillitis is present' is $p' < p$. For each target hypothesis H:

[4] *Actually, there are not many papers in the literature presenting both practical methodologies for Bayesian model developments and practical applications of value of information theory.*

[5] *We will re-visit the case example that in Mussi (1993) was represented in a heuristic rule-based approach. In this paper we will represent it in the normative approach, experimenting in this way the enormous advantages stemming from using decision theory (Kalagnanam & Henrion, 1990).*

[6] *Only a single target hypothesis is considered; monetary costs etc. are not considered.*

[7] *The example has been implemented with HUGIN, an environment for building Bayesian networks and influence diagrams.*

[8] *Although in some kinds of problems (see the wildcatter problem) measures do not affect target hypotheses, for simplicity let us refer to the more common case in which measures affect target hypotheses (as in diagnosis).*

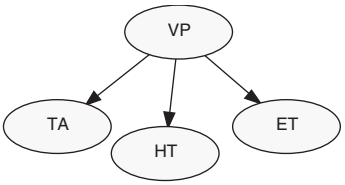

Figure 1: *The DN sub-network represents the causal relations between the disease VP (*H *node) and the symptom TA (Sy node) and the outcomes of the tests HT and ET (*J *nodes). Both HT and ET have two states: pat (i.e. pathologic) and ok. Both VP and TA have two states:* y *(yes, i.e. presence) and* n *(no, i.e. absence). The strengths of links VP → HT and VP → ET respectively represent the diagnostic powers of the two tests.*

(1) let us create an Haf node (where Haf stands for '*H* after the measure has been performed');[9]
(2) let us create the links $H \rightarrow$ Haf, $A \rightarrow$ Haf; Haf $\rightarrow U_{\text{Haf}}$ (where U_{Haf} stands for the utility node of Haf).

Let us turn to the example problem. If VP is present the physician has at his / her disposal two alternative therapies (measures): the pharmacological therapy (for short, Pharm) and the surgical therapy (for short, Surg). We therefore create a node *A* with possible states Surg, Pharm and na, and build the TN sub-network as illustrated in Figure 2.

Step 3: Building SA for the set of measures A

An action (be it a test or a measure) is, in general, more or less costly, more or less risky etc. If we say that an action is risky we intend to say that there is a certain probability that some undesirable anomalous situations (h_is) occur as a consequence of performing the action. So, instead of considering the action attribute 'risk', let us explicitly represent in the network the nodes h_i of the anomalies that might occur (because of performing the action).[10]

Turning to measures, for each anomaly h_i that a measure $a \in A$ can cause we create

(1) a node h_i and the related node $h_i$af ($h_i$af means 'h_i after the measure has been performed');
(2) the links $h_i \rightarrow h_i$af; $A \rightarrow h_i$af; $h_i$af $\rightarrow$ Uh$_i$.

As for the monetary cost, let us represent it by creating a node MC whose states represent intervals of possible monetary costs.[11] Let us note that even *h* nodes are hypothesis nodes. An *h* node (like an *H* node) can have

[9] *The set of states of Haf is obviously equal to that of* H.
[10] *Let us note that in some application domains we might have to do with future anomalies too. For example, in medicine, cancer therapy might produce negative side-effects that appear a certain period of time after the therapeutic treatment. In such a case we might, for example, define* h_i *as 'liver problems after 6 months'.*
[11] *Let us discretize the range of possible costs on a qualitative scale. In general the cost is deterministic, so given an action* a *a single state of MC will have probability 1, and the others will have probability 0.*

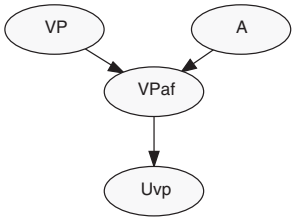

Figure 2: *The TN sub-network represents the effects of the therapies (i.e. the states of* A*: Surg, Pharm, na) upon the disease VP. The conditional probability table of VPaf defines the therapeutic effectiveness of each therapy. Uvp is the utility node related to VPaf.*

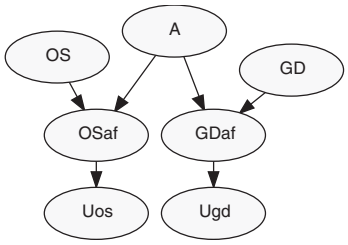

Figure 3: *The SA sub-network represents the side-effects of the therapies (i.e. of* a ∈ A*). Both OS and GD are* h *nodes. They and the related af nodes have two states,* y *and* n.

children nodes of type Sy. However, for simplicity, for an *h* node we do not consider children nodes of type *J*.[12] The sub-networks SA and DN do not have haf nodes in common.[13]

Turning to the example problem, let us represent the fact that Surg, being an operation, entails typical operation side-effects (OS) (e.g. anaesthesia side-effects, post-operation pains, stay in hospital etc.).[14] As for Pharm, its disadvantage consists in the side-effect represented by the possible occurrence of some gastric disturbances (GD).

The SA sub-network is therefore built as illustrated in Figure 3.

Step 4: Building SB for the set of tests B

For each J_i node acquired during the construction of the sub-network DN, a test b_i is added to the set of states *B*. Then, as was done above for building the SA sub-network, we build the SB sub-network by creating, for each anomaly h_i that a test $b \in B$ can cause,

(1) an h_i node and the related $h_i$af node;
(2) the links $h_i \rightarrow h_i$af; $B \rightarrow h_i$af; $h_i$af $\rightarrow$ Uh$_i$.

[12] *In other words, we do not consider service tests, i.e. tests devoted to establish if an* h *node is true or false. So, even if a piece of information is obtained by performing a service test, we consider it as a directly observable manifestation.*
[13] *For simplicity, we do not consider the case in which an intermediate node of DN is also an* h *node in SA.*
[14] *For simplicity, we do not consider the risk due to the possible occurrence of complications typical of that kind of operation.*

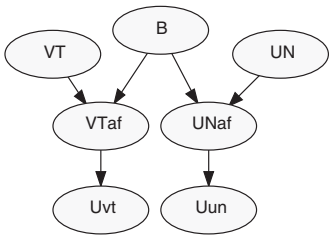

Figure 4: *The SB sub-network represents the side-effects of the tests (i.e. of* b ∈ B*).* B *has three states: ET, HT, nb. Both VT and UN are* h *nodes. They and the related af nodes have two states,* y *and* n.

Let us note that the sub-networks SA and SB can have some haf nodes in common (i.e. both a measure and a test can cause the same anomaly). The sub-networks SB and DN do not have haf nodes in common.[15]

Turning to the example problem, the construction of the sub-network DN has identified the tests HT and ET. We therefore create a node *B* with possible states ET, HT, nb. ET is invasive and is not free of risk. In fact it can cause an anomaly called ventricular tachycardia (VT). As for HT, the only (small) disadvantage it entails is a sort of uneasiness (for short, UN) due to the fact that for 24 hours the patient has to wear a special device that monitors blood pressure, cardiac rhythm etc. On the basis of this knowledge let us build the SB sub-network as illustrated in Figure 4.

Step 5: Defining probability distributions and eliciting exceptional situations

A great deal of literature has been produced about the problem of acquiring probability values. The most common sources of probabilistic information are statistical data, literature and human experts. If a comprehensive data collection is available, probability values can be automatically learned; otherwise the knowledge engineer is prompted to use the third type of source: the human expert. Various techniques have been proposed for supporting the knowledge engineer in the task of probability elicitation, including similarity networks (Heckerman, 1991), sensitivity analysis (Laskey, 1995; Castillo *et al.*, 1997) and other approaches (for more information see Jensen (2001), Morgan and Henrion (1990) and in *IEEE Transactions on Knowledge and Data Engineering* (2000) see the special section 'Building probabilistic networks: where do the numbers come from?'). What is important to note here is that during probability elicitation it can happen that the expert does not feel completely confident in assessing the probability of an event *X*. Usually this fact is a flag that additional information gathering might be valuable (Heckerman & Jimison, 1988). In these cases, the expert is asked to identify other (conditioning) events (i.e. exceptional situations (for short, Excs)) that have outcomes that influence the probability of *X*. Both prior and conditional probabilities can be affected by possible exceptional situations. An exceptional situation Exc, conditioning the probability distribution of a node *X*, is represented by creating both the Exc node and the link Exc → *X*.

[15] *Otherwise a test would perturb the world whose status should be revealed by the test itself.*

Table 1: *The CPT of the node VPaf*

VP	*y*			*n*		
A	na	Surg	Pharm	na	Surg	Pharm
VPaf = *y*	1	**0**	**0.3**	0	0	0

Note: The CPT shows that Surg is 100% effective, whereas Pharm is 70% effective.

Let us turn to the example problem and let us focus on possible exceptional situations. Let us consider Figure 1. When eliciting the prior probability distribution of VP, the physician states that VP is more likely if in the family history of the patient there are cases of cardiac arrhythmia (for short, CAinFH).[16] As a consequence, let us enrich the sub-network of Figure 1 by adding the node CAinFH (with two states: *y*, *n*) and the link CAinFH → VP. Similarly (let us consider Figure 3), when eliciting the prior probability distribution of GD, the physician states that if a patient is a smoker (for short, SM), he / she has a higher probability of suffering from GD. So, let us enrich the sub-network of Figure 3 by adding the node SM (with two states: *y*, *n*) and the link SM → GD. Finally (let us consider Figure 4), when eliciting the conditional probability distribution of VTaf (recall that VT stands for ventricular tachycardia), the physician states that the risk associated with performing ET increases if the age of the patient is over 25. So, let us enrich the sub-network of Figure 4 by adding the node Age (with two states: ≤ 25, > 25) and the link Age → VTaf. Two examples of conditional probability tables (CPTs) are shown in Tables 1 and 2.

Step 6: Defining utilities and eliciting exceptional situations

Let us consider af nodes representing consequences of actions (be they measures or tests), i.e. haf and Haf nodes. Let us assign proper utility values to the states of each af node. In some kinds of applications it is possible to quantify the utility value by means of a unique measure-unit (as, for example, money). When this is not possible, utilities are elicited from the expert. There are several utility-elicitation methods (von Winterfeldt & Edwards, 1986). A commonly used method is the so-called variable probability method (von Winterfeldt & Edwards, 1986; Pearl, 1988).

Let us now consider the concept of overall utility and let us instantiate the general formulae of utility theory (Keeney & Raiffa, 1976; von Winterfeldt & Edwards, 1986) in our context. The expected utility from performing an action is the overall expected utility coming from the

[16] *In other words, CAinFH plays the role of anamnesis for VP.*

Table 2: *The CPT of the node VTaf*

Age	$\leqslant 25$						>25					
VT	*y*			*n*			*y*			*n*		
B	nb	HT	ET	nb	HT	ET	nb	HT	ET	nb	HT	ET
VTaf $= y$	1	1	1	0	0	**0.2**	1	1	1	0	0	**0.6**

Notes: The CPT shows that ET is the only test raising the probability of the presence of VT. Note that, if the patient is over 25, the risk due to ET increases from 0.2 to 0.6.

status of the network consequent to the action. The overall expected utility coming from a certain status of the network is calculated as the weighted sum of the expected utilities coming from the single af nodes.

More formally, let $\mathrm{Xaf}_1, \mathrm{Xaf}_2, \ldots, \mathrm{Xaf}_m$ be the set of af nodes, and for each Xaf_i let $\{\mathrm{xaf}_{i1}, \mathrm{xaf}_{i2}, \ldots\}$ be the set of the states of Xaf_i (for short, $\mathrm{xaf}_{ij} \in \mathrm{Xaf}_i$). For each single Xaf_i let us define (e.g. using the variable probability method) the single utility function $u_i(\mathrm{xaf}_{ij})$. If xaf_{ij*} and xaf_{ij}^* are respectively the worst and the best states, we have that $u_i(\mathrm{xaf}_{ij*}) = 0$, and $u_i(\mathrm{xaf}_{ij}^*) = 1$. Then (using, for example, the variable probability method) let us define the 'weights' k_i such that $\sum_{i=1}^{m} k_i = 1$. In conclusion, the overall expected utility from an action is given by

$$EU(\text{action}) = \sum_{i=1}^{m} k_i \left[\sum_{\mathrm{xaf}_{ij} \in \mathrm{Xaf}_i} u_i(\mathrm{xaf}_{ij}) P(\mathrm{xaf}_{ij} | \text{action}) \right] \tag{1}$$

Even utilities can be affected by exceptional situations. If an exceptional situation Exc affects a utility function u_i we create a link from Exc to the utility node that is child of Xaf_i, and replace (1) with the following:

$$EU(\text{action}) = \sum_{i=1}^{m} k_i \left\{ \sum_{\mathrm{xaf}_{ij} \in \mathrm{Xaf}_i} \left[u_i(\mathrm{xaf}_{ij}) | \mathrm{Exc} \right] P(\mathrm{xaf}_{ij} | \text{action}) \right\} \tag{2}$$

Conversely, if Exc affects the weight distribution, we create a link from Exc to each utility node and replace (1) with the following:

$$EU(\text{action}) = \sum_{i=1}^{m} (k_i | \mathrm{Exc}) \left[\sum_{\mathrm{xaf}_{ij} \in \mathrm{Xaf}_i} u_i(\mathrm{xaf}_{ij}) P(\mathrm{xaf}_{ij} | \text{action}) \right] \tag{3}$$

How can we represent utilities in practice? Let us suppose we want to represent in the utility node Ux_i the utility coming from Xaf_i. Let us instantiate the trick described in Jensen (1996, p. 141) in our context. Let us assign the node Ux_i two states, say Util and dummy, and let us define the CPT of Ux_i as $P(\mathrm{Ux}_i = \mathrm{Util} | \mathrm{Xaf}_i = \mathrm{xaf}_{ij}) = k_i u_i(\mathrm{xaf}_{ij})$. So, the expected weighted utility coming from Xaf_i is just $P(\mathrm{Ux}_i = \mathrm{Util})$. As for the state 'dummy', its role is only that of assuring the complement to 1, but it has no relevance in the reasoning process (hence its name).

Table 3: *The CPT of the utility node* U_{VP}

Agon	*n*		*y*	
VPaf	*n*	*y*	*n*	*y*
$P(U_{\mathrm{VP}} = \mathrm{Util})$	**0.7**	0	**0.8**	0

Notes: Considering, for example, the condition $\mathrm{Agon} = n$, we have $u(\mathrm{VPaf} = y | \mathrm{Agon} = n) = 0$, $u(\mathrm{VPaf} = n | \mathrm{Agon} = n) = 1$, and as a consequence we have $P(U_{\mathrm{VP}} = \mathrm{Util} \mid \mathrm{Agon} = n) = k_{\mathrm{VP}} = 0.7$.

Let us turn to the example problem. The distribution of the relative importance weights is represented by the set $\{k_{\mathrm{VP}}, k_{\mathrm{VT}}, k_{\mathrm{UN}}, k_{\mathrm{OS}}, k_{\mathrm{GD}}\}$. However, in the example problem the distribution is not unique. In fact the physician says that if the patient practises agonistic activity (for short, Agon), then he (the physician) wants to be fully confident that VP is absent. As a consequence agonistic activity plays the role of exceptional situation for the relative importance weight assigned to the utility of the absence of VP. So we elicit from the expert two relative importance weight distributions:[17] $\{k_{\mathrm{VP}}, k_{\mathrm{VT}}, \ldots\} | \mathrm{Agon} = n = \{\mathbf{0.7}, 0.15, \ldots\}$; $\{k_{\mathrm{VP}}, k_{\mathrm{VT}}, \ldots\} | \mathrm{Agon} = \boldsymbol{y} = \{\mathbf{0.8}, 0.09999, \ldots\}$. Note that since the variables VPaf, VTaf, ..., GDaf have two states, namely *y* (with utility 0) and *n* (with utility 1), for each of them the weighted utility is equal to the related weight, and the expected utility is simply obtained by multiplying it by the probability of the related state *n*. For example, Table 3 shows the CPT of the utility node U_{VP}, and the product $P(U_{\mathrm{VP}} = \mathrm{Util} | \mathrm{Agon} = n) * P(\mathrm{VPaf} = n)$ represents the expected utility coming from $\mathrm{VPaf} = n$ given $\mathrm{Agon} = n$. Let us represent in the network the exceptional situation 'agonistic activity' by creating the Agon node and linking it to each utility node, i.e. $\mathrm{Agon} \rightarrow U_{\mathrm{VP}}$, $\mathrm{Agon} \rightarrow U_{\mathrm{VT}}$, ... Finally, let us put all the sub-networks together (Figure 5).

3. Building the inference engine

In the following sections we will describe how to effectively use the knowledge base in order to support the user in sequential decision-making. Even in this part we will give concreteness to the abstract general description by using the example problem.

[17] *Note that, given the distribution with* $\mathrm{Agon} = \mathrm{n}$ *and given the difference between* $\mathrm{k}_{VPaf} | \mathrm{Agon} = \mathrm{y}$ *and* $\mathrm{k}_{VPaf} | \mathrm{Agon} = \mathrm{n}$, *the remaining elements of the distribution with* $\mathrm{Agon} = \mathrm{y}$ *can be calculated automatically.*

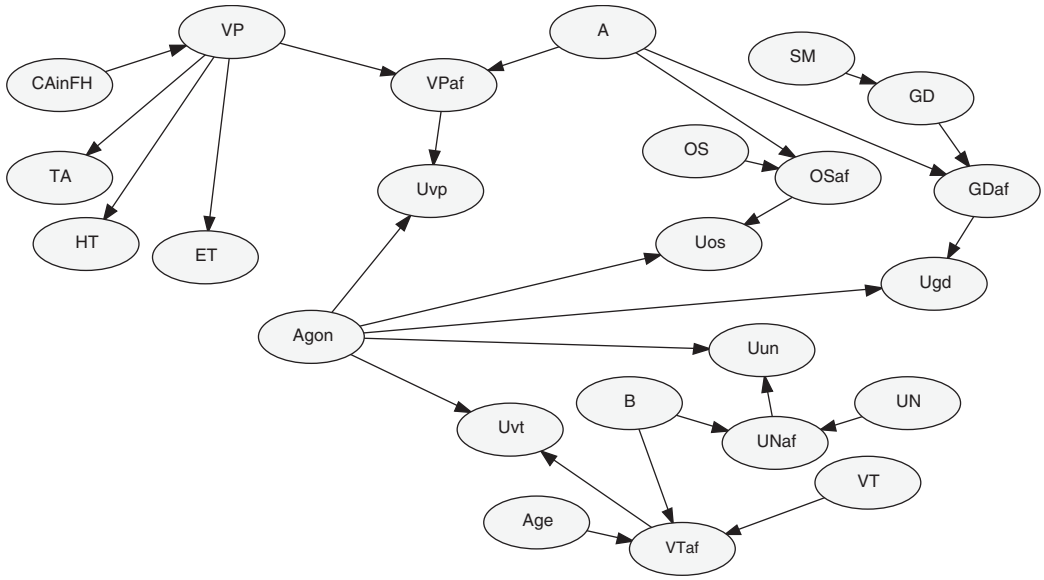

Figure 5: *The whole network of the example problem. The network integrates the four previous sub-networks and includes the Exc nodes: CAinFH, SM, Agon, Age. Let us recognize* J *nodes (HT, ET), Sy nodes (TA),* H *nodes (VP),* h *nodes (OS, GD, UN, VT), af nodes (VPaf, OSaf, GDaf, UNaf, VTaf), the special nodes* A, B *and the utility nodes Uvp, Uos, Ugd, Uun, Uvt. Note that after the first roll-up the nodes CAinFH and SM are no longer present.*

3.1. Rolling-up the network

A sequential decision-making process is made up of a sequence of decision steps. At each decision step the calculation of the expected utility from performing an action should depend on all the pieces of information collected up to that time (even in case of retesting) and should take into account the fact that side-effects of actions can be cumulative in time. To this end let us 'roll up' the network just before each decision step (after the first one). The roll-up procedure encompasses two kinds of operations: topology modification and probability modification. It involves both *h* nodes (see steps 3 and 4) and *H* nodes (see step 1).

The operation of topology modification consists in making *h* nodes and *H* nodes root nodes if they are not root nodes. In other words, if one of these nodes has parent nodes, they (the parents) have to be eliminated.

The operation of probability modification basically consists in assigning a proper prior distribution probability to the *H* and *h* nodes. In particular, for an *H* node, this operation consists in assigning it a prior probability distribution equal to the probability distribution it has in the current status of the network.[18] For an *h* node, the operation consists in assigning it a prior probability distribution

(i) equal to the probability distribution the related haf node has in the current status of the network, or
(ii) equal to the probability distribution it has in the current status of the network, or
(iii) calculated on the basis of both the probability distribution the related haf node has in the current status of the network and the time interval between the last decision step and the next one.

Which alternative among these should be adopted? It depends on which specific side-effect the *h* node represents. The first alternative is adopted if the side-effect keeps constant in time, and the second alternative if the side-effect immediately vanishes. If we want to represent the case of a side-effect varying (e.g. gradually vanishing) in time we choose the third alternative.

More formally, let *K* denote an *H* node or an *h* node. A roll-up consists in executing the following steps.

(1) If *K* is an *H* node then save $P(H)$ in M.[19]
If K is an *h* node then (according to the specific side-effect *h* represents)
  (a) save P(haf) in M, or
  (b) save $P(h)$ in M, or
  (c) determine the prior probability distribution to assign to *h* and then save it in M.
(2) If *K* is not a root node then eliminate its parents.
(3) For each *K* child which is set in a certain state,[20] retract the setting.
(4) Let M be the prior probability distribution of *K* (i.e. assign M to $P(K)$).

[18] *We implicitly assume that the distribution probability of an* H *node does not vary during the time interval between two adjacent decision steps.*

[19] *Let* M *be a variable used to temporarily save data.*

[20] *This corresponds to the fact that evidence (related to that node) has been entered by the user.*

Let us turn back to the example problem and let us consider the network of Figure 5. The set of H nodes is represented by {VP}, and the set of h nodes by {UN, VT, GD, OS}. VP and GD have parents. They will be eliminated in the first roll-up. VP has three children: HT, ET and TA. For each of them, which are set to a certain state, let us retract the setting.

Finally, recalling the above classification (i), (ii), (iii), let us note that UN is a case (ii) (i.e. we assume that uneasiness disappears just after the test), whereas GD, OS and VT are case (iii).

Let us consider, for example, the case of VT (a possible side-effect of ET). The side-effect of ET consists in a sort of 'trauma' (produced by its invasiveness). As a consequence the physician is prompted to retain that, immediately after performing ET, the probability that VT is present rises over its default value (i.e. $P(\mathrm{VTaf}=y) > P(\mathrm{VT}=y)$). However, this sort of trauma is gradually adsorbed in time[21] (see Figure 6).

Let us suppose, for example, that ET is performed and that its outcome is entered after $\Delta t = 3$ days. Then, the statement 'determine the prior probability distribution to assign to h and save it in M' (see the above point 1(c)) becomes

(a) generate the current VT curve (starting from the value of $P(\mathrm{VTaf}=y)$ in the current network);
(b) determine the value of $P(\mathrm{VT}=y)(3 \text{ days})$ and save it in M.

Figure 7 illustrates an example of how the network changes after the first roll-up.[22]

3.2. Supporting sequential decision-making

In this section we present an algorithm for supporting sequential decision-making. The basic fact that performing an action has the side-effect that it changes the expected benefit of performing other actions affects the structure of the algorithm. The algorithm supports decisions concerning both tests (which test to perform next) and measures (which measure to perform next). The general strategy consists in first gathering information until every test is not worth being performed, and then performing a measure. The algorithm stops when each action is no longer worth being performed. Basically the algorithm is characterized by the following cycle.

(1) Calculate and show the user the expected profits[23] of the tests.

[21] *Another typical case of side-effects vanishing in time occurs in the case of radiographies. If a patient undergoes a certain number of radiographies in a short period of time, the risk associated with the last one is greater than the risk associated with the first. However, radiation is gradually absorbed as time goes on.*
[22] *The Exec nodes Agon and Age are not involved in the roll-up.*
[23] *The concept of profit will be defined in the next section.*

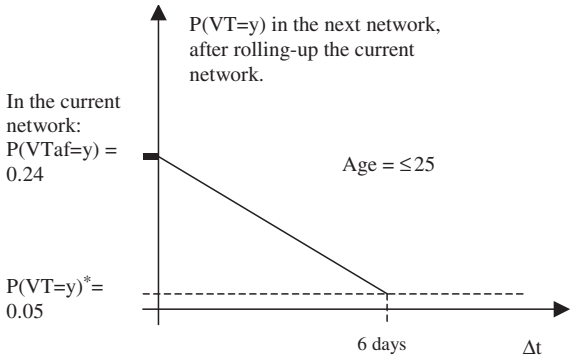

Figure 6: *The basic curve of $P(\mathrm{VT}=y)(\Delta t)$ related to the trauma caused by performing ET (linearity is an approximation) on a patient who is under 25. $P(\mathrm{VT}=y)^*$ stands for the default value of $P(\mathrm{VT}=y)$. The figure shows that, if ET is performed, $P(\mathrm{VTaf}=y)$ rises up to 0.24 and the trauma caused by ET is gradually absorbed in 6 days. Given the slope of this basic curve, a current curve depending on the value of $P(\mathrm{VTaf}=y)$ in the current network can be generated dynamically.*

(2) If there are some tests that are worth being performed, then ask the user to perform the test with the maximum expected profit. After the test has been performed update the network and go back to point (1).
(3) If no test is worth being performed then calculate and show the user the expected utilities of the measures.
(4) If there are some measures that are worth being performed, then ask the user to perform the measure with the maximum expected utility. After the measure has been performed update the network and go back to point (1).
(5) If no measure is worth being performed, then stop.

The algorithm is shown in Figure 8. Referring to the figure let us comment on some lines of the algorithm. Let us start with the case in which there are tests with positive expected profit. We assume that the test is performed at the same time at which the list of tests (LST) is printed. At line 3 real time elapsing is considered. Such time elapse refers to the temporal interval Δt_b between the time at which the test b is performed and the time at which its outcome j is entered.

Line 4 shows that the two facts $B = b$ (occurring at the time at which b is performed) and $J_b = j$ (occurring at the time at which j is available, i.e. Δt_b later) are entered together, and then a unique propagation is performed. This is possible because of the separation between the sub-networks SB and DN.[24]

[24] *If the two sub-networks were not separated, the propagation of the outcome and the side-effect together would produce an incorrect situation. In fact at the time at which the outcome is available it is not guaranteed that the side-effect of the test is still as strong as it was when the test was performed.*

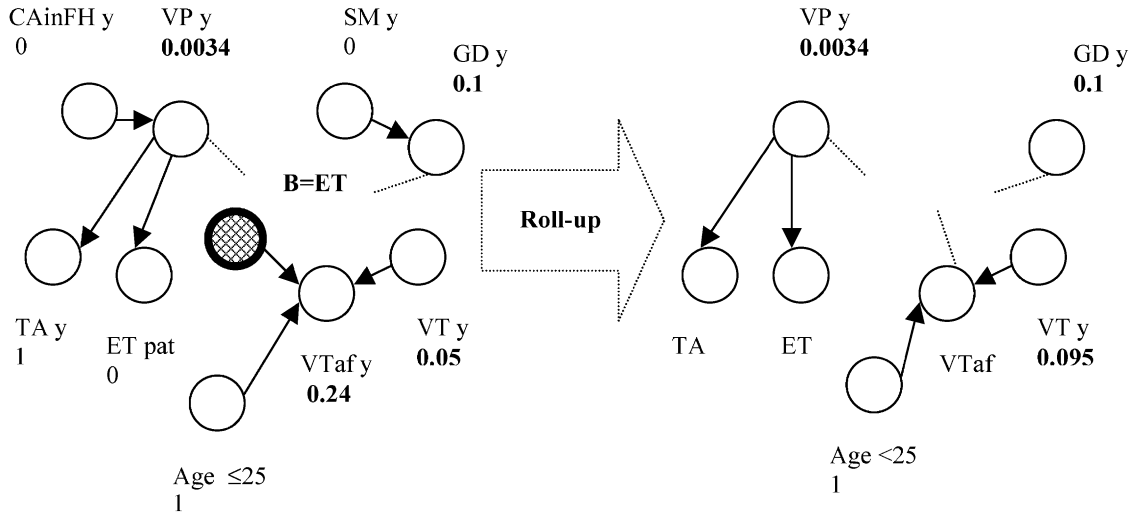

Figure 7: *The effects of the first roll-up of the network of Figure 5, starting from a current network status like the one represented, in a fragment, on the left. For each node, only one value of the binary distribution is represented. The network fragment on the left shows that (1) we have to do with the case of a patient who suffers from tachycardia, does not smoke, is under 25 and has no case of cardiac arrhythmia in his / her family history; (2) ET has been performed, producing the outcome 'ok' and the increase of* $P(\mathrm{VTaf} = y)$ *from 0.05 to 0.24. The fragment on the right shows that (1) the nodes CAinFH and SM have been eliminated leaving VP and GD with the probability distributions they had in the network on the left; (2) the node VT has been given a probability distribution derived from* $P(\mathrm{VTaf} = y)$ *in the network on the left, the curve of Figure 6, and* $\Delta t = 3$ days*; (3) the settings of the nodes TA and ET have been retracted.*

```
0) Set the states of Sy and Exc nodes
While true do
  BEGIN
1) ORDER-TESTS
2) Print LST
  If there are tests with positive expected profit
    then BEGIN
        Print: "perform a test b∈B and enter its outcome j"
3)      Pause (wait for the test outcome, Δt_b ≥ 0)
4)      Set the states: B = b , A = na , J_b = j
5)      Propagate and then Roll-up the network
6)        Possibly set the states of some Sy and Exc nodes
        END
  If the expected profit of each test is ≤ 0
    then BEGIN
        Print "no test has a positive expected profit"
        ORDER-MEASURES
        Print LST
        If the expected utility of each measure is ≤ 0
          then EXIT
          else  BEGIN
              Print: "perform a measure a∈A"
              Pause (wait for the measure outcome, Δt_a ≥ 0)
              Set the state: A=a
              Propagate and then Roll-up the network
              Set the states of Sy and Exc nodes
              END
        END
  ENDof while
```

Figure 8: *The algorithm for supporting sequential decision-making. The expected profits of tests and expected utilities of measures are calculated by the procedures ORDER-TESTS and ORDER-MEASURES respectively. The ORDER- procedures return LST, the list of the tests/measures ordered (in decreasing sense) according to their expected profits/utilities.*

The roll-up statement of line 5 prepares the network for supporting the next decision step (i.e. for the next execution of ORDER-TESTS).

Finally, let us comment on line 6. It can happen that during the time interval Δt_b the real world provides new knowledge. For example, an exceptional situation or a manifestation that was present at the beginning of Δt_b is no longer present at the end of it, or vice versa. In such a case the related node must be set to the proper state. In particular, let us note that if the exceptional situation concerns an Exc node that has been eliminated by some previous roll-up, then the Exc node is first re-created and then set to the proper state.

In the case in which no test has positive expected profit, the expected utilities of measures are calculated and then shown to the user. The following statements are similar to the ones we have commented on.[25]

3.3. Supporting single decision-making

The calculus of the expected profits of tests is based on decision theory using value of information (Jensen, 1996). The outcome j of a test $b \in B$ (i.e. $j \in J_b$) is information

[25] *In diagnosis, after a measure has been performed, it is important to set (if possible) Sy nodes that are children of* H *nodes, even if they have to be set to the same states they had before the measure was performed. For example, after antibiotic therapy, it is important to know if fever is still present.*

whose value $V(j)$ is given by (the symbol ξ stands for 'all the remaining pieces of information so far acquired')

$$V(j) = \max_{a \in A} EU(a|b,\ j, \xi) \tag{4}$$

where $EU(a)$ is calculated by (1) (step 6) and A stands for the set of measures. As a consequence, the expected value from performing b is

$$EV(b) = \sum_{j \in J_b} V(j) P(j|b, \xi) \tag{5}$$

Considering the dummy test nb we have

$$EV(\text{nb}) = \max_{a \in A} EU(a|\text{nb}, \xi) \tag{6}$$

So the expected benefit of a test b is given by

$$EB(b) = EV(b) - EV(\text{nb}) \tag{7}$$

Similarly, the expected cost of b is given by

$$EC(b) = EU(b) - EU(\text{nb}) \tag{8}$$

where EU is calculated by (1) (step 6). Finally, taking into account both the value of each possible outcome ($j \in J_b$) and the expected cost from performing b, we have that the expected profit of b is given by

$$EP(b) = EB(b) + EC(b) \tag{9}$$

The procedures ORDER-TESTS and ORDER-MEASURES that respectively calculate the expected profits of tests[26] and measures are illustrated in Figures 9 and 10.

Let us turn to the example problem and let us experiment with the application of the algorithm of Figure 8 to the network of Figure 5. Let us suppose that the patient we are dealing with suffers from tachycardia; in his/her family history there are no cases of cardiac arrhythmia; he/she does not smoke. Let us represent these facts in the network of Figure 5 by setting the states TA = y, CAinFH = n, SM = n. It is now interesting to know how much the expected profits of the two tests HT and ET are affected by the age of the patient and by the fact that he/she practises agonistic activity or not. Let us therefore execute four times (each time with a different combination of the states of Age and Agon) the statements of lines 0, 1, 2 (see Figure 11). Given the prior probability distributions and the CPTs (elicited by the physician) documented in the Appendix, the result of the executions is shown in Table 4. Let us note that the maximum expected profit for ET is reached if the patient is under 25 and practises agonistic activity, whereas the maximum expected profit for HT is reached if the patient is over 25 and does not practise agonistic activity. These results are in accordance with medical common sense. In fact, although ET is more risky than HT, it is worth performing if the patient is young (the risk is more acceptable) and practises athletics (the utility of VP = n is higher).

[26] *It is worth noting that the expected profit of a test causing a side-effect gradually vanishing in time might be negative immediately after the test has been performed and positive after a certain time.*

```
Procedure ORDER-TESTS

let LST be the empty list.
Set the states: B=nb, A=na
Propagate
Calculate EU(nb)
For each a∈ A do:
    BEGIN
    Set the state: A=a
    Propagate
    Calculate EU(a|nb,ξ)
    END
Calculate EV(nb)
For each b∈ B, b ≠ nb do:
    BEGIN
    Set the states: B=b, A=na
    Propagate
    Calculate EU(b), EC(b)
    For each j∈ J_b do:
        BEGIN
        Set the state: J_b = j
        For each a∈ A do:
            BEGIN
            Set the state: A=a
            Propagate
            Calculate EU(a|b,j,ξ)
            END
        Calculate V(j)
        END
    Calculate EV(b), EB(b), EP(b)
    Put in LST the couple (b , EP(b))
        in decreasing order
    END
```

Figure 9: *The procedure ORDER-TESTS, providing the user with the list of tests (LST) ordered according to their expected profits, supports his / her next decision-making.*

```
Procedure ORDER-MEASURES

Let LST be the empty list.
Set the state: B=nb
For each a  A do:
    BEGIN
    Set the state: A=a
    Propagate
    Calculate EU(a|nb,ξ)
    END
Put in LST the couple (a , EU(a))
    in decreasing order
```

Figure 10: *The procedure ORDER-MEASURES, providing the user with the list of measures (LST) ordered according to their expected profits, supports his / her next decision-making.*

Let us now make a second experiment. Keeping the above background setting (TA = y, CAinFH = n, SM = n), let us consider a specific case, e.g. the case of a patient who is under 25 and practises agonistic activity. The first execution of ORDER-TESTS provides the user with the list shown in the third column in Table 4. Let us now suppose the patient

P(VP=y|CAinFH) = 0.2|y, 0.1| n; P(TA=y|VP) = 0.6|y, 0.2|n;
P(HT=pat|VP) = 0.7|y, 0.2|n; P(CAinFH=y) = 0.1;
P(ET=pat|VP) = 0.99|y, 0.01|n ; P(GD=y|SM) = 0.3|y, 0.1|n;
P(SM=y) = 0.4; P(OS=y) = 0; P(VT=y) = 0.05; P(UN=y) = 0;
P(VPaf=y|VP=y,A) = 1|na, 0|surg, 0.3|pharm; P(VPaf=y|VP=n,A) = 0|na, 0|surg, 0|pharm;
P(OSaf=y|OS=y,A) = 1|na, 1|surg, 1|pharm; P(OSaf=y|OS=n,A) = 0|na, 1|surg, 0|pharm;
P(GDaf=y|GD=y,A) = 1|na, 1|surg, 1|pharm; P(GDaf=y|GD=n,A) = 0|na, 0|surg, 0.3|pharm;
P(UNaf=y|UN=y,A) = 1|nb, 1|HT, 1|ET; P(UNaf=y|UN=n,A) = 0|nb, 1|HT, 0|ET;
P(VTaf=y|Age≤25,VT=y,A) = 1|nb, 1|HT, 1|ET; P(VTaf=y| Age≤25,VT=n,A) = 0|nb, 0|HT, 0.2|ET;
P(VTaf=y|Age>25,VT=y,A) = 1|nb, 1|HT, 1|ET; P(VTaf=y| Age>25,VT=n,A) = 0|nb, 0|HT, 0.6|ET;
$\{k_{VP}, k_{VT}, k_{UN}, k_{OS}, k_{GD}\}$|Agon=n = 0.7, 0.15, 0.001, 0.1, 0.049;
$\{k_{VP}, k_{VT}, k_{UN}, k_{OS}, k_{GD}\}$|Agon=y = 0.8, 0.09999, 0.0006666, 0.6666, 0.0326634;
Probability distributions of A, B, Agon, Age are even. As for the utility nodes, as explained in step 6, the probability of the state Util is 0 if the state of the related af node is y, is equal to the related k if the state of the related af node is n. For example:
P(Uvp=Util|Agon=n,VPaf) = 0|y, 0.7|n; P(Uvp=Util|Agon=y,VPaf) = 0|y, 0.8|n.

Figure 11: *The set of probabilities and utilities related to the network of Figure 5.*

Table 4: *The preference order of ET, HT (quantified in terms of expected utility) according to the states of Agon and Age*

Agon	*n*		*y*	
Age	⩽25	>25	⩽25	>25
	ET (0.0099)	HT (0.0075)	ET (0.0286)	HT (0.0203)
	HT (0.0075)	ET (–0.0470)	HT (0.0203)	ET (–0.0093)

Note: Age does not affect HT, while Agon, affecting the utility weight distribution, affects both ET and HT.

undergoes ET and then, after $\Delta t = 1$ hour, the ET outcome, say 'ok', is entered. The network is then updated according to the statements of line 5 in Figure 8. Suppose we do not enter any state change for Sy and Exc nodes (line 6). The procedure ORDER-TESTS is then executed for the second time and provides the user with the list

HT (–0.0006) ET(–0.0151)

The list shows that it is not worth performing any further test. Moreover let us also note that even the order of the list is different from the one produced by the first execution of ORDER-TESTS (third column in Table 4). In fact, given that $\Delta t = 1$ hour is practically a null time interval if related to the curve of Figure 6, we have that P(VT) in the network after the roll-up is equal to P(VTaf) in the network before the roll-up. So, in the second execution of ORDER-TESTS the value of EU_c(ET) turns out to be less than it was in the first execution. This fact says that a repetition of ET after a short time interval from the previous ET is discouraged. Even this is in accordance with medical common sense: since ET is diagnostically very significant, there is no point in performing ET again, in addition to the fact that ET is risky and has just been performed.

4. Discussion and conclusion

4.1. Discussion and related work

Bayesian networks, and the related extensions called influence diagrams, represent a major knowledge source for sequential decision models (Mookerjee & Mannino, 1997). The scientific literature shows a continuous thread of work in this field, from the seminal work by Tatman and Shacter (1990) to the recent work by Nilsson and Lauritzen (2000). A classical case of sequential decision-making is represented by sequential diagnosis (Gorry & Barnett, 1985) and troubleshooting (Heckerman *et al.*, 1994). In Breese and Heckerman (1996) and Skaanning *et al.* (2000) Bayesian networks are used for defining a sequential decision model for troubleshooting.

Our methodology is based on the use of Bayesian networks that encompass, according to a well-known trick (Cooper, 1988; Jensen, 1996), special nodes for representing actions and utilities. The first part of the methodology shows how to build in six steps a whole integrated network; however, it does not enter the specific field of the various techniques and tools for eliciting probabilities and causal networks. For large diagnostic Bayesian networks, elicitation can be supported by the technique of similarity networks (Heckerman, 1991). A domain-specific knowledge acquisition tool for Bayesian networks is presented in Skaanning (2000). The interested reader can find further information about probability elicitation techniques in Morgan and Henrion (1990).

Let us now discuss some representation issues, showing similarities and differences between the proposal and related work. In Heckerman *et al.* (1992) and Heckerman and Nathwani (1992), a single target hypothesis node is considered: each state of the node represents a disease. Similarly, a single target hypothesis node is considered in Breese and Heckerman (1996): each state of the node represents a fault. Moreover, the costs of observation and

repair do not depend on previous repair or observation actions. The single fault assumption has been adopted even in Skaanning *et al.* (2000) and Jensen *et al.* (2001).[27] Actions are represented as children of the target hypothesis (i.e. the cause variable), and their costs are represented as linear combinations of cost factors (money, risk etc.). The 'risk of breaking something else' is specified on a scale of 0–4.

In our methodology, actions are represented as states of two distinct nodes (node *A* for measures and node *B* for tests). Action costs are explicitly represented in terms of side-effect nodes with related utility nodes. This explicit approach leads us to build two Bayesian sub-networks SA and SB, for the side-effects of measures and tests respectively, so that we can explicitly represent how and by how much each single exceptional situation and piece of evidence affects the overall cost of an action. Moreover, the roll-up procedure described in Section 3.1 allows us to easily model the fact that the current cost of an action depends on previous actions. In fact, on one hand dealing with side-effect sub-networks we can easily represent the possibility that some actions have side-effects in common, and on the other hand by rolling-up the network (as described in Section 3.1) we model the cumulative aspect of side-effects and explicitly act on the single specific components of the overall action cost, considering both the different types of side-effects and the elapsed time since the last action performance. In particular, given that the sub-networks SA and SB may have *h* nodes in common, we can easily represent how and by how much (even taking time into account) tests affect measure costs and vice versa. Finally, the proposal also defines an explicit representation for measure effectiveness, by defining a target-effect sub-network (TN) where possible exceptional situations affecting effectiveness of a measure can be simply represented as network nodes.

Both the work presented in Breese and Heckerman (1996) and the work presented in Skaanning *et al.* (2000) and Jensen *et al.* (2001) concern troubleshooting: a process consisting in a sequence of actions which encompasses not only tests (as in traditional diagnosis) but repairs too. Our methodology takes into account the fact that even measures, not only tests, can be sequentially performed. Moreover, the methodology considers sequential decision scenarios in which measures may or may not affect target hypotheses. The case of diagnosis is a typical case in which measures affect target hypotheses. So in diagnostic applications the roll-up involves, besides side-effects, target effects too.

Finally, let us consider the question of myopic/non-myopic approach. Our methodology uses the myopic value of information approach, based on the assumption that the decision-maker will immediately act after seeing the result of a single test. As is well known, the analysis of all possible sequences of tests is intractable, because the number of sequences grows exponentially with the number of tests. In practice, for simplicity, many real-world applications use the myopic value-of-information approach, which is anyhow considered a good heuristic (Gorry & Barnett, 1985). Heckerman *et al.* (1991) present an approximate non-myopic analysis. However, this approach is limited to information acquisition decisions for problems involving binary action variables and one binary hypothesis variable.

In conclusion, in the light of the above discussion the methodology proposed in the present paper turns out to be characterized by the following sources of power:

- *modelling power*: the capability of reasoning with multiple hypotheses, the capability of taking into account past information and cumulative side-effects varying in time, the capability of representing the pervasive aspect of exceptional situations applied to both probabilities and utilities, and the capability of explicitly representing, through side-effect sub-networks, action costs and their mutual effect make the methodology suitable for modelling complex real-world systems;
- *user-friendliness*: the methodology on one hand guides the designer of a normative knowledge base through six steps each of which involves a specific focus of mind, and on the other hand proposes an inference engine that is easy to implement. As a consequence the proposed methodology provides builders of SDSSs with an easy way of applying decision theory with value of information to difficult sequential decision-making problems.

4.2. Conclusions

The paper has presented a methodology for building DT-VOI based SDSSs. The methodology guides the SDSS designer in building step by step an SDSS knowledge base and the related inference engine. The methodology is characterized by being domain independent, user-friendly, and powerful enough to model sophisticated real-world cases characterized by multiple hypotheses, exceptional situations and mutual influences between actions whose side-effects vary in time. The proposal puts into practice the general concept of value of information, and so, in this sense, the work presented in this paper complements the previous work presented in Mussi (2002), providing the reader with a practical methodology and a deeper understanding of the impact that sequential decision models have on designing expert systems.

Appendix

For completeness we list in Figure 11 the CPTs and the weight distributions elicited by the physician and then used in the experiments described in Section 3.3.

[27] *'When troubleshooting printing systems, it is more natural to assume a single fault than to assume independent faults' (Jensen* et al.*, 2001).*

Acknowledgements

Since most of the paper was written in my holiday time, I would like to thank my wife Maria Donata and my daughter Alessandra Libera for their patience.

References

BREESE, J.S. and D. HECKERMAN (1996) Decision-theoretic troubleshooting: a framework for repair and experiment, Technical Report MSR-TR-96-06, Microsoft Research.

CASTILLO, E., J.M. GUTIERREZ and A.S. HADI (1997) Sensitivity analysis in discrete Bayesian networks, *IEEE Transactions on Systems, Man and Cybernetics – Part A: Systems and Humans*, **27** (4), 412–423.

COOPER, G.F. (1988) A method for using belief networks as influence diagrams, in *Proceedings of the Fourth Workshop on Uncertainty in Artificial Intelligence*, Minneapolis, MN, 55–63.

GORRY, G.A. and G.O. BARNETT (1985) Experience with a model of sequential diagnosis, in *Computer-assisted Medical Decision Making*, J.A. Reggia and S. Turhim (eds), Berlin: Springer, Vol. 1, pp. 206–222.

HECKERMAN, D.E. (1991) *Probabilistic Similarity Networks*, Cambridge, MA: MIT Press.

HECKERMAN, D. and H. JIMISON (1988) A Bayesian perspective on confidence, in *Proceedings of the Third Conference on Uncertainty in Artificial Intelligence*, New York: Elsevier.

HECKERMAN, D.E. and B.N. NATHWANI (1992) Toward normative expert systems: Part II Probability-based representations for efficient knowledge acquisition and inference, *Methods of Information in Medicine*, **31** (2), 106–116.

HECKERMAN, D., E. HORVITZ and B. MIDDLETON (1991) An approximate nonmyopic computation for value of information, in *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, San Mateo, CA: Morgan Kaufmann, 135–141.

HECKERMAN, D.E., E.J. HORVITZ and B.N. NATHWANI (1992) Toward normative expert systems: Part I The Pathfinder project, *Methods of Information in Medicine*, **3** (2), 90–105.

HECKERMAN, D., J. BREESE and K. ROMMELSE (1994) Sequential troubleshooting under uncertainty, in *Proceedings of the Fifth International Workshop on Principles of Diagnosis*, New Paltz, NY, 121–130.

IEEE Transactions on Knowledge and Data Engineering, **12** (4), 2000, 481–528.

JENSEN, F.V. (1996) *An Introduction to Bayesian Networks*, London: UCL Press.

JENSEN, F.V. (2001) *Bayesian Networks and Decision Graphs*, Berlin: Springer.

JENSEN, F., U. KJAERULFF, B. KRISTIANSEN, H. LANGSETH, C. SKAANNING, J. VOMLEL and M. VOMLELOVA (2001) The SACSO methodology for troubleshooting complex systems, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing Archive,* **15** (4), 321–333.

KAHNEMAN, D., P. SLOVIC and A. TVERSKY (eds) (1982) *Judgement Under Uncertainty: Heuristics and Biases*, New York: Cambridge University Press.

KALAGNANAM, J. and M. HENRION (1990) A comparison of decision analysis and expert rules for sequential diagnosis, in *Uncertainty in Artificial Intelligence*, R.D. Shachter, T.S. Levitt, L.N. Kanal and J.F. Lemmer (eds), Amesterdam: North-Holland.

KEENEY, L.R. and H. RAIFFA (1976) *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*, New York: Wiley.

LASKEY, K.B. (1995) Sensitivity analysis for probability assessments in Bayesian networks, *IEEE Transactions on Systems, Man and Cybernetics*, **25** (6), 901–909.

MOOKERJEE, V.S. and M.V. MANNINO (1997) Sequential decision models for expert system optimization, *IEEE Transactions on Knowledge and Data Engineering*, **9** (5), 675–687.

MORGAN, M.G. and M. HENRION (1990) *Uncertainty – A Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis*, Cambridge: Cambridge University Press.

MUSSI, S. (1993) A method for putting strategic common sense into expert systems, *IEEE Transactions on Knowledge and Data Engineering*, **5** (3), 369–385.

MUSSI, S. (2002) Sequential decision-theoretic models and expert systems, *Expert Systems*, **19** (2), 99–108.

NILSSON, D. and S.L. LAURITZEN (2000) Evaluating influence diagrams using LIMIDs, in *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, San Mateo, CA: Morgan Kaufmann.

PEARL, J. (1988) *Probabilistic Reasoning in Intelligent Systems*, San Mateo, CA: Morgan Kaufmann.

PUTERMAN, M.L. (1994) *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, New York: Wiley.

SKAANNING, C. (2000) A knowledge acquisition tool for Bayesian-network troubleshooters, in *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, San Mateo, CA: Morgan Kaufmann.

SKAANNING, C., F. JENSEN and U. KJAERULFF (2000) Printer troubleshooting using Bayesian networks, in *Intelligent Problem Solving, Methodologies and Approaches. Proceedings of the Thirteenth International Conference on Industrial and Engineering Applications of AI and Expert Systems* (*IEA/AIE-2000*), R. Logananharaj and G. Palm (eds), Lecture Notes in Computer Science 1821, Berlin: Springer, 367–379.

TATMAN, J.A. and R.D. SHACHTER (1990) Dynamic programming and influence diagrams, *IEEE Transactions on Systems, Man and Cybernetics*, **20** (2), 365–379.

VON WINTERFELDT, D. and W. EDWARDS (1986) *Decision Analysis and Behavioral Research*, Cambridge: Cambridge University Press.

The author

Silvano Mussi

Silvano Mussi graduated in physics in 1975 from the University of Milan, Italy. He worked at ITALTEL for 10 years in the fields of software engineering and functional discrete simulations of real-time systems. Since 1981 he has been with CILEA. He cooperated in research activities with Milan Polytechnic and Brescia University where for three academic years he was contract-professor of artificial intelligence. For 10 years he has done research in the fields of knowledge engineering and expert systems. His current research interests address methods for providing Websites with capabilities of decision-making and reasoning under conditions pervaded with uncertainty.