Formalization of Weighted Factors Analysis

Ali Hessami*and Anthony Hunter†‡

March 7, 2002

Keywords: Argumentation; Knowledge representation and reasoning; Logic; Decision support; Scenario analysis; Knowledge management.

Abstract

Weighted Factors Analysis (WeFA) has been proposed as a new approach for elicitation, representation, and manipulation of knowledge about a given problem, generally at a high and strategic level. Central to this proposal is that a group of experts in the area of the problem can identify a hierarchy of factors with positive or negative influences on the problem outcome. The tangible output of WeFA is a directed weighted graph called a WeFA graph. This is a set of nodes denoting factors that can directly or indirectly influence an overall aim of the graph. The aim is also represented by a node. Each directed arc is a direct influence of one factor on another. A chain of directed arcs indicates an indirect influence. The influences may be identified as either positive or negative. For example, `sales` and `costs` are two factors that influence the aim of `profitability` in an organization. `sales` has a positive influence on `profitability` and `costs` has a negative influence on `profitability`. In addition, the relative significance of each influence is represented by a weight. In this paper, we develop Binary WeFA which is a variant of WeFA where the factors in the graph are restricted to being either true or false. Imposing this restriction on a WeFA graph allows us to be more precise about the meaning of the graph and of reasoning in it. Binary WeFA is a new proposal that provides a formal yet sufficiently simple language for logic-based argumentation for use by business people in decision-support and knowledge management. Whilst Binary WeFA is expressively simpler than other logic-based argumentation formalisms, it does incorporate a novel formalization of the notion of significance.

1 Introduction

Weighted Factors Analysis (WeFA) has been proposed as a new approach for elicitation, representation, and manipulation of knowledge about a given problem, generally at a high and strategic level [8]. Central to this proposal is that a group of experts in the area of the problem can identify factors with positive or negative influences on the problem outcome. Furthermore, for each of these

*WSAtkins, 171 High Holburn, London WC1V 7AA.
†Department of Computer Science, University College London, Gower Street, London WC1E 6BT.
‡Corresponding author (a.hunter@cs.ucl.ac.uk)

factors, this breakdown can be repeated by recursion. This gives a hierarchical decomposition of factors according to how directly each factor affects the overall problem being analysed. WeFA harnesses this elicitated knowledge in a structured framework for further analysis.

Once the need for the application of WeFA to a problem has been identified, a meeting of relevant and interested stakeholders and experts can be called to examine the key factors in the problem. The first task of this group is knowledge elicitation and the starting point for this is the identification of a factor that is the overall aim of the analysis. For example it might be `The project is completed on time and within budget` or it might be `The business has increased profitability`. Having agreed on the aim, the group can normally identify factors with positive or negative influences on the outcome of the aim. Furthermore, for each of these factors, subsidiary factors can be identified with positive and negative influences on the factor. This process can be continued, until an appropriate breakdown and analysis of the problem has been achieved.

The tangible output of WeFA is a WeFA graph. This is a set of nodes denoting factors that can directly or indirectly influence an overall aim of the graph. The aim is also represented by a node. Each directed arc is a direct influence of one factor on another. A chain of directed arcs indicates an indirect influence. The influences may be identified as either positive or negative. For example, `sales` and `costs` are two factors that influence the aim of `profitability` in an organization. `sales` has a positive influence on `profitability` and `costs` has a negative influence on `profitability`.

Each node, other than the aim, in a WeFA graph is either a positive or negative influence on the aim. A factor can have a positive influence on the aim via a direct positive influence on the aim or indirectly by having a negative influence on a node that has a negative influence on the aim or indirectly by having a positive influence on a node that has a positive influence on the aim. A factor can have a negative influence on the aim via a direct negative influence on the aim or indirectly by having negative influence on a node that has a positive influence on the aim or indirectly by having a positive influence on a node that has a negative influence on a node.

A node that has a direct positive influence on another node is called a driver and a node that has a direct negative influence on another node is called an inhibitor. Construction of a WeFA graph is usually done by just considering the drivers and inhibitors for each node. This allows for the contruction of the graph by focussing on local issues. Once the graph is constructed, it can be analysed globally.

In addition to identifying the polarity for each influence, it is possible to identify the relative weight of the influence. For a factor in a graph, there may be a number of factors that influence it. These factors may have different weights. For example, consider the factor `passing a new bill in parliament`, with the subsidiary factors `the bill is supported by a backbench MP` and `the bill is supported by the Prime Minister`. Both these subsidiary factors have a positive influence, but the latter factor carries much more weight than the former.

In a given WeFA graph, there are a number of factors represented that can directly or indirectly influence the aim of the graph. So the outcome of the aim of the graph is regarded as contingent on the truth value of the other factors in the graph. To analyse this contingency, a scenario is an assignment of truth values to some of these factors. Given this assignment, the truth values can be propagated through the graph in order to determine the truth value for the aim.

Scenario analysis involves identifying both extreme scenarios, such as best-case and worst-case scenarios, and average-case scenarios, to see the effect on the outcome of the aim of the graph. Other criteria may also be used to generate interesting sets of scenario to test in a WeFA graph.

In this paper, we provide a formalization of WeFA. In particlar, we develop Binary WeFA which

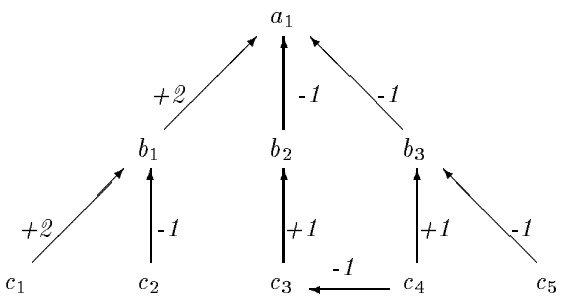

Figure 1: A Binary WeFA graph with the aim a_1. b_1 has a direct positive influence on the aim, and b_2 and b_3 have direct negative influences on the aim. So b_1 is a driver for the aim and b_2 and b_3 are inhibitors. c_1, c_4 and c_5 have an indirect positive influence on the aim, and c_2, c_3 and c_4 have an indirect negative influence on the aim. As an example of a scenario, we could have c_1, c_2 and c_3 true. From this, we would get c_4, c_5 and b_3 are false by default, and b_1 and b_2 are true by inference, and finally a_1 true by inference.

.

is a variant of WeFA where the factors in the graph are restricted to being either true or false. Imposing this restriction on a WeFA graph allows us to be more precise about the meaning of the graph and of reasoning in it. In this formalization, we have sought a balance between having a simple representation that participants in WeFA should feel happy with, and a representation that has sufficient expressibility for worthwhile problem analysis.

In order to do reasoning with a Binary WeFA graph, we assume a scenario which is an assignment of true or false to some of the nodes in the graph. If a node is true, then any arc coming out from that node is fired. From this assignment, we can propagate truth assignments to the other nodes as follows: For each node x, if the sum of the weights on the positive fired arcs is greater than the sum of the weights on the negative fired arcs, then node x is true, otherwise it is false. The gives us a way of representing and reasoning with the relative significance of direct influences on a node.

In the basic definition for Binary WeFA, factors can either be true or false and there is no explicit probability value. However it does incorporate default reasoning: If a node is not assigned true by a scenario, and it is not forced to be true by application of the scenario to the graph, then it is assumed to be false. Uncertainty is captured by the combination of default reasoning and the formalisation of significance of influences.

Binary WeFA is a form of logic-based argumentation system that incorporates a simplification of techniques from Hunter et al [10, 9]. An excellent introduction to the notion of argumentation is by Toulmin [17] and an excellent introduction to the role of logic in argumentation is Fisher [5]. See [16] for a recent reviews of logic-based argumentation.

Whilst there have been a significant number of recent proposals for logic-based argumentation, most seem to implicitly assume that the reasoning is done on a knowledge-base constructed by knowledge engineers. In contrast, we assume that the participants in the WeFA exercise are constructing the knowledge-base (ie. the WeFA graph). This means we have adopted a simpler and more intuitive language, with simpler reasoning, than that normally proposed for logic-based argumentation. However, we believe that there is a significant need for Binary WeFA in problem analysis, decision-support and knowledge management in organizations.

2 Formalization of syntax

A WeFA graph is composed of a set of nodes that represent measurable parameters in the real-world. An arc from a node x to a node y denotes an influence of x on y. In addition, each arc is weighted to indicate the strength of the arc relative to the other arcs with the same destination node.

A WeFA graph is used to represent a general situation. More specifically it is used to represent how some events can influence other events. This can be to model a one-off situation, such as for analysing a strategic decision, or to model a reoccurring situation, such as for analysing a set of operational decisions.

Definition 2.1 *A WeFA graph is a tuple (N, A, F) where N is a set of nodes, A is a set of arcs and F assigns an integer, called the weight, to each arc in A. In addition, we assume there is a node x in N such that (1) there are only inward arcs into N and (2) there is a path (respecting the directionality of the arcs) from every other node to x. We call x the aim of the graph.*

We assume that a WeFA graph is developed for analysing some aspect of an organization. The aim is the central aim or goal of the organization in the situation being modelled and analysed. Normally, we do not know whether the aim is true (or false), and we want to determine whether according to the graph certain combinations of other nodes being true or false makes the aim true (or false). However, an alternative way of using a WeFA graph is as follows: It may be known that the aim is true (or false), but it is unknown which combination of other nodes being true or false made the aim true (or false).

According to the definition of a WeFA graph, the constraints are such that the graph is connected though not strongly connected. Cycles are allowed. Also allowed are bidirectional influences between nodes, i.e. for a pair of nodes x and y, it is possible to have an arc from x to y and an arc from y to x.

Definition 2.2 *Each node in a WeFA graph has an associated statement (normally a phrase or sentence in free text) that denotes an outcome that can be verified or falsified in the real-world.*

Example 2.1 *The following are examples of statements associated with nodes:*

```
The project completion is within-time and under-cost
There is an adequate cost-cutting strategy
Inflation is high
There is understaffing
There are problems with bad weather
Increase the charge to the customer
```

Taking the first statement `The project completion is within-time and under-cost`*, we may associate further text that defines what* `project completion` *is, what* `within-time` *means, and what* `under-cost` *means. This further text may be comprehensive and may refer to other documents, reports, standards, etc.*

Definition 2.3 *Each node is either an environmental variable or an organizational variable. The environmental variables are outside the control of the organization, eg.* `Inflation is high`*. The organizational variables are inside the control of the organization, eg.* `Increase the charge to the customer`*.*

One of the roles of WeFA modelling is to see how decisions by an organization are affected by the wider environment when trying to realise some goal. It can also be used to see how certain negative environmental outcomes can be offset or ameliorated by organizational decisions.

Definition 2.4 *We can represent a WeFA graph using sets, or a figure, or using the following inline representation: The [] symbol is used to represent each arc with a weight so that an arc (x, y) with weight $+3$ is represented by $x[+3]y$. So for any WeFA graph (N, A, F), $F(x, y) = n$ is equivalent to $x[n]y$.*

Example 2.2 *Let the following be a WeFA graph. Here there are four nodes, with all the arcs pointing to node y.*

$$x_1[-1]y \qquad x_2[-1]y \qquad x_3[+2]y$$

Definition 2.5 *The weight associated with each arc denotes the strength of the arc relative to the other arcs entering the same node. The whole number part of the integer gives the significance, and the sign indicates whether it is a positive influence (a driver) or a negative influence (an inhibitor). A + sign indicates a positive influence and a − sign indicates a negative influence.*

Example 2.3 *Returning to Example 2.2, we see that the influence of x_3 on y is twice as significant as either the influence of x_1 or x_2 on y.*

We give the formal semantics for significance in Section 3. However, informally we motivate it in the following examples.

Example 2.4 *Consider a WeFA graph with two arcs converging on a node y. Suppose one of the arcs is a positive influence and the other is a negative influence. In addition, suppose they are both of equal significance. These could be represented by the following arcs:*

$$x_1[-1]y \qquad x_2[+1]y$$

If both x_1 and x_2 are true, then y is false. If x_1 is true and x_2 is false then y is false. Finally if x_1 is false and x_2 is true then y is true.

Example 2.5 *Now consider a different graph with two arcs converging on a node y such that one arc is twice as significant as the other.*

$$x_3[-1]y \qquad x_4[+2]y$$

If both x_3 and x_4 are true then we can determine that y is true. If x_3 is true and x_4 is false then y is false. Finally if x_3 is false and x_4 is true then y is true.

We explain the details of the mechanism behind these examples in this section. For this, we require the definition of a number of subsidiary functions. In order to simplify the presentation we omit qualifying subscripts, in particular for referencing WeFA graphs.

Definition 2.6 *For a WeFA graph (N, A, F), the sets Inputs and Outputs are defined as follows:*

For each $x \in N$, $Inputs(x) = \{y \in N \mid (y, x) \in A\}$

For each $x \in N$, $Outputs(x) = \{y \in N \mid (x, y) \in A\}$

For each $x \in N$, $Feeders(x) = \{y \in N \mid$ there is a path from y to x in $A\}$

Definition 2.7 *For a WeFA graph (N, A, F), the sets PositiveInputs and NegativeInputs are defined as follows:*

$$For\ each\ x \in N, PositiveInputs(x) = \{(y, x) \in A \mid F(y, x) > 0\}$$

$$For\ each\ x \in N, NegativeInputs(x) = \{(y, x) \in A \mid F(y, x) < 0\}$$

For a WeFA graph (N, A, F), each x in N is a variable that can take a value either true or false. In a given situation, we may know the truth values of some of the nodes with the remainder unknown. This is represented by a scenario (defined as follows).

Definition 2.8 *For a WeFA graph (N, A, F), a scenario, denoted S, is a partial function from N to $\{true, false\}$.*

A scenario fixes zero or more variables with a Boolean truth value. However, it does not take the structure of the WeFA graph into consideration. As a result no ramifications of the scenario are identified. We address this in the following.

Definition 2.9 *A primed WeFA graph is a tuple (N, A, F, S) where (N, A, F) is a WeFA graph and S is a scenario.*

Definition 2.10 *For a primed WeFA graph (N, A, F, S), an extension, denoted E, is a total function from N to $\{true, false\}$.*

Given a primed WeFA graph, we want to determine what are the truth values of all the nodes in the graph are. In other words, we want to determine the ramifications of the scenario when applied to the graph. The result of applying a scenario to a WeFA graph is an extension.

Definition 2.11 *For a primed WeFA graph (N, A, F, S), an extension E, and a node $x \in N$, the sets PosFire and NegFire are defined as follows:*

$$PosFire(E, x) = \{(y, x) \in PositveInputs(x) \mid E(y) = true\ and\ F(y, x) > 0\}$$

$$NegFire(E, x) = \{(y, x) \in NegativeInputs(x) \mid E(y) = true\ and\ F(y, x) < 0\}$$

If (y,x) is in PosFire(E,x), then (y,x) is a positive fired arc in E. Similarly, if (y,x) is in NegFire(E,x), then (y,x) is a negative fired arc in E.

Definition 2.12 *For a WeFA graph (N, A, F), where $\{(z_1, x), .., (z_i, x)\}$ is a subset of A, the Value function is defined as follows:*

$$Value(\{(z_1, x), .., (z_i, x)\}) = F(z_1, x) + .. + F(z_i, x)$$

where $Value(\{\}) = 0$.

Definition 2.13 *For a primed WeFA graph (N, A, F, S), an extension E is a consistent extension of (N, A, F, S) iff the following two conditions are satisfied for each node x in N:*

$$E(x) = true\ iff\ S(x) = true\ or\ (S(x) \neq false\ and\ Value(PosFire(E, x)) > Value(NegFire(E, x)))$$

$$E(x) = false\ iff\ S(x) = false\ or\ (S(x) \neq true\ and\ Value(PosFire(E, x)) \leq Value(NegFire(E, x)))$$

A consistent extension is an extension that gives priority to the scenario so that if a scenario says a particular node is true (or respectively false), then it is true (or respectively false) in the consistent extension. In the case when a scenario does not specify whether a particular node is true or false, then the truth value is determined from the aggregation of the input arcs. If the sum of the significance of the positive fired arcs is greater than the sum of the negative fired arcs, then the node is true in the consistent extension, otherwise it is false.

Example 2.6 *The following WeFA graph, with scenario* $\{(x, true)\}$*, gives the consistent extension* $\{(x, true), (y, true), (z, false)\}$.

$$x[+1]y \qquad y[-1]z$$

Example 2.7 *Consider the following WeFA graph*

$$y_1[+1]x \qquad y_2[-1]x$$

and the following scenarios, S_1,S_2,S_3*, and* S_4*, which respectively give the corresponding consistent extensions* E_1*,* E_2*,* E_3*, and* E_4 *on the right.*

$$\begin{array}{ll} S_1 = \{(y_1, false), (y_2, true)\} & E_1 = \{(y_1, false), (y_2, true), (x, false)\} \\ S_2 = \{(y_1, true), (y_2, true)\} & E_2 = \{(y_1, true), (y_2, true), (x, false)\} \\ S_3 = \{(y_1, true), (y_2, false)\} & E_3 = \{(y_1, true), (y_2, false), (x, true)\} \\ S_4 = \{(y_1, false), (y_2, false)\} & E_4 = \{(y_1, false), (y_2, false), (x, false)\} \end{array}$$

Example 2.8 *Consider the following WeFA graph*

$$y_1[+2]x \qquad y_2[-1]x \qquad y_3[-1]x \qquad y_4[-1]x$$

and the following scenarios, S_1 *and* S_2*, which give the corresponding consistent extensions* E_1 *and* E_2 *below.*

$$S_1 = \{(y_1, true), (y_2, true), (y_3, true), (y_4, true)\}$$
$$E_1 = \{(y_1, true), (y_2, true), (y_3, true), (y_4, true), (x, false)\}$$

$$S_2 = \{(y_1, true), (y_2, true)\}$$
$$E_2 = \{(y_1, true), (y_2, true), (y_3, false), (y_4, false), (x, true)\}$$

Example 2.9 *Consider the following WeFA graph*

$$y_1[+2]x \qquad y_2[-1]x$$

and the following scenarios, which respectively give the corresponding consistent extensions on the right.

$$\begin{array}{ll} S_1 = \{(y_1, true)\} & E_1 = \{(y_1, true), (y_2, false), (x, true)\} \\ S_2 = \{(y_1, true), (y_2, true)\} & E_2 = \{(y_1, true), (y_2, true), (x, true)\} \\ S_3 = \{(y_1, false), (y_2, false)\} & E_3 = \{(y_1, false), (y_2, false), (x, false)\} \\ S_4 = \{(y_1, true), (x, false)\} & E_4 = \{(y_1, true), (y_2, false), (x, false)\} \\ S_5 = \{(y_2, true)\} & E_5 = \{(y_1, false), (y_2, true), (x, false)\} \end{array}$$

A primed WeFA graph is not guaranteed to give a unique consistent extension, as illustrated by Example 2.10.

Example 2.10 *Consider the following WeFA graph with the empty set scenario:*

$$x[+1]y \qquad y[+1]x$$

Here the extension E_1 where $E_1(x) = E_1(y) = true$ is consistent but it is only because the nodes are self-supporting. In contrast, the extension E_2 where $E_2(x) = E_2(y) = false$ is also consistent but the nodes are not self supporting.

In order to avoid multiple extensions, we use the following definition of complete extension that gives the minimal consistent extension.

Definition 2.14 *For a primed WeFA graph (N, A, F, S), a consistent extension E_i is a complete extension of (N, A, F, S) iff for all E_j such that E_j is a consistent extension of (N, A, F, S), if $E_i(x) = true$ then $E_j(x) = true$.*

A complete extension is a minimal extension in the sense that a node is only true in the extension if it is forced to be true either by being true in the scenario or by propagation of truth values in the graph.

Definition 2.15 *For a primed WeFA graph (N, A, F, S), the negation-as-default assumption is that each node is assumed false unless found to be true either because assumed true in S or inferred true by the application of S to (N, A, F).*

The negation-as-default assumption is implicit in the definition for a complete extension. In other words, if E is a consistent extension of (N, A, F, S) then the negation-as-default assumption holds for E.

Every primed WeFA graph is guaranteed to have a complete extension.

Example 2.11 *Consider the following WeFA graph with an empty set scenario.*

$$x[+1]y \qquad y[-1]x$$

This has the complete extension $\{(x, false), (y, false)\}$.

We define inference in a primed WeFA graph in terms of an extension.

Definition 2.16 *If E is the complete extension of a primed WeFA graph (N, A, F, S), and $E(x)$ is true, then x is an inference from (N, A, F, S).*

Example 2.12 *Consider the nodes $a1$ to $c3$ below, where $a1$ is the focus of the WeFA graph, the b_i nodes are the second layer nodes, and the c_i nodes are the third layer nodes.*

a_1	`The requirements capture and management is robust`
b_1	`There is a credible process`
b_2	`There is an understanding of the social and legal context`
b_3	`The system is complex`
b_4	`There are appropriate tools and skills`
b_5	`There is a lack of communication`
b_6	`The system context is understood`
b_7	`There is a failure to identify stakeholders needs clearly`
c_1	`There are conflicting stakeholder needs`
c_2	`There is a change in stakeholders`
c_3	`There is no business case`

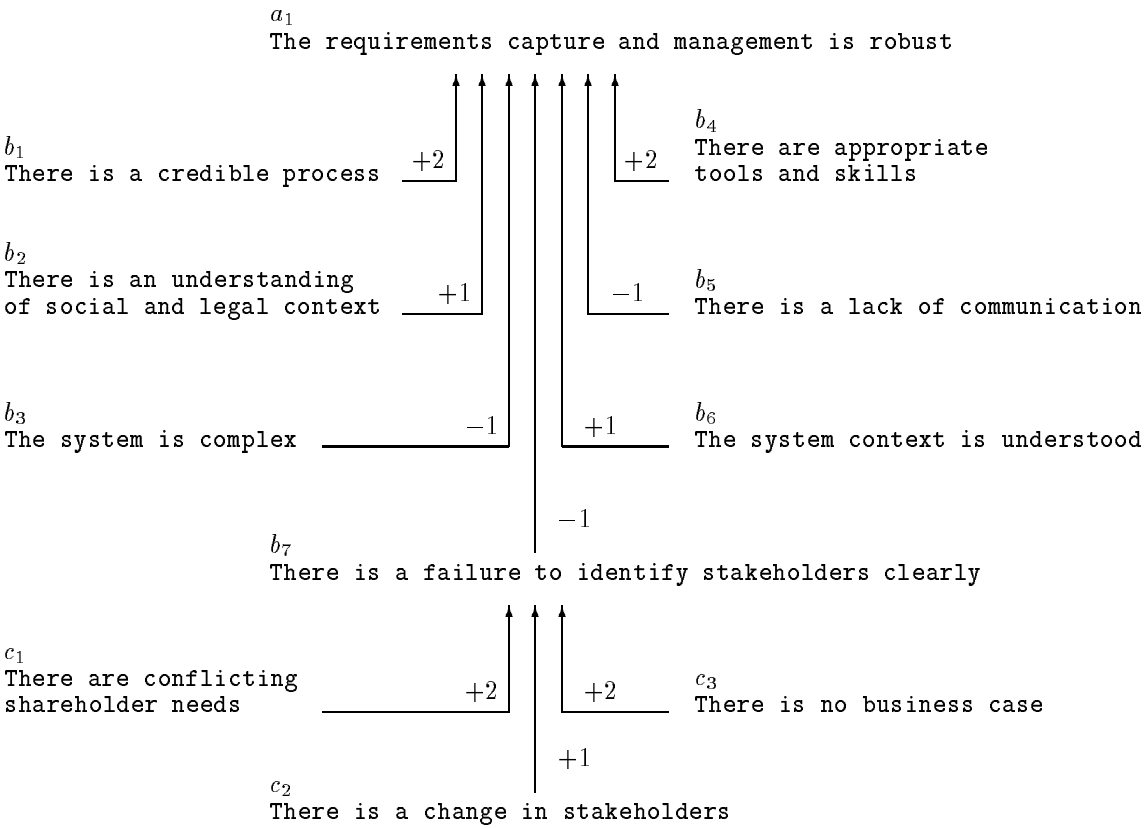

Figure 2: A Binary WeFA graph with the aim `The requirements capture and management is robust` which is discussed in Example 2.12

.

The graph can be represented by the following arcs. This graph is also given in Figure 2.

$$\begin{array}{lllll} b_1[+2]a_1 & b_3[-1]a_1 & b_5[-1]a_1 & b_7[-1]a_1 & c_2[+1]b_7 \\ b_2[+1]a_1 & b_4[+2]a_1 & b_6[+1]a_1 & c_1[+2]b_7 & c_3[+2]b_7 \end{array}$$

So if we have a scenario

$$S = \{(b_1, true), (b_2, true), (c_1, false), (c_2, true), (c_3, false)\}$$

then we get the following complete extension

$$\{(a_1, true), (b_1, true), (b_2, true), (b_3, false), (b_4, false), (b_5, false), \\ (b_6, false), (b_7, true), (c_1, false), (c_2, true), (c_3, false)\}$$

and hence the top-level goal a_1 is true.

The notion of significance is more than just representing a total ordering over the incoming arcs to each node. It provides a powerful notation for representing and reasoning with the uncertainty inherent in argumentation in real-world models. It is less powerful, but more practical, than seeking a separate aggregation function for each node [9].

The use of Boolean truth value for nodes, and for aggregation of truth values at each node avoids the more complex problems found in uncertainty reasoning approaches such as probabilistic reasoning and possibilistic reasoning. Here we are dealing with normality rather than probability: If

we know the truth values for the nodes in $Input(x)$ then we determine what normally we would expect the truth value of x should be using the significance in the positive fired and negative fired arcs. We argue that this is a more lucid representation of uncertainty than an approach based on propagating numerical values.

Implicit in the Binary WeFA approach is that uncertainty is managed in part by the weights on the arcs. In addition to scenario analysis, we can explore shifting weights between a pessimistic and an optimistic bias. If we want a pessimistic bias, we increase the significance of the negative influences, whereas if we want an optimistic bias, then we increase the significance of the positive influences.

3 Formalization of semantics

So far we have a definition for the representation of Binary WeFA graphs and associated concepts. We also have some informal semantics for the notation and concepts introduced. We now consider a formal semantics. For this, we will assume classical propositional logic. We will define WeFA graphs in terms of this logic. Note, however there is a default mechanism in binary WeFA graphs, and so the formalization here extends classical propositional logic.

Definition 3.1 *We assume the usual definition of the classical propositional language. Let* $\neg$ *be the negation operator,* $\vee$ *be the disjunction operator,* $\wedge$ *be the conjunction operator, and* $\rightarrow$ *be the implication operator. If* α *is an atom then* α *is a formula, and if* α *and* β *are formulae, then* $\neg\alpha$, $\alpha \vee \beta$, $\alpha \wedge \beta$ *and* $\alpha \rightarrow \beta$ *are formulae. If* α *is an atom, then* α *is a literal and* $\neg\alpha$ *is a literal. We also assume the usual truth tables for classical propositional logic for interpreting these formulae.*

In order to represent a WeFA graph as a set of classical logic formulae, we require some subsidiary functions.

Definition 3.2 *The functions* $Conjunction$ *and* $NegatedConjunction$ *form a formula from a set of literals as follows:*

$$Conjunction(\{q_1, ..., q_i\}) = q_1 \wedge .. \wedge q_i$$
$$NegatedConjunction(\{q_1, .., q_i\}) = \neg q_1 \wedge .. \wedge \neg q_i$$

Definition 3.3 *For any set* K, $\wp(K)$ *is the power set of* K.

Definition 3.4 *For a WeFA graph* (N, A, F), *the functions* $PosContribution$ *and* $NegContribution$ *are defined as follows. Let* K *be a subset of* N *and let* x *be in* N. *It is not necessary for* x *to be in* K.

$$PosContribution(K, x) = \{(y, x) \in PositiveInputs(x) \mid y \in K\}$$

$$NegContribution(K, x) = \{(y, x) \in NegativeInputs(x) \mid y \in K\}$$

Example 3.1 *Consider the WeFA graph*

$$x_1[+1]y \qquad x_2[-1]y \qquad x_3[+2]y$$

Then

$PosContribution(\{x_1\}, y) = \{(x_1, y)\}$ $\quad NegContribution(\{x_1\}, y) = \{\}$
$PosContribution(\{x_2\}, y) = \{\}$ $\quad NegContribution(\{x_2\}, y) = \{(x_2, y)\}$
$PosContribution(\{x_3\}, y) = \{(x_3, y)\}$ $\quad NegContribution(\{x_3\}, y) = \{\}$
$PosContribution(\{x_1, x_2\}, y) = \{(x_1, y)\}$ $\quad NegContribution(\{x_1, x_2\}, y) = \{(x_2, y)\}$
$PosContribution(\{x_1, x_3\}, y) = \{(x_1, y), (x_3, y)\}$ $\quad NegContribution(\{x_1, x_3\}, y) = \{\}$
$PosContribution(\{x_1, x_2, x_3\}, y) = \{(x_1, y), (x_3, y)\}$ $\quad NegContribution(\{x_1, x_2, x_3\}, y) = \{(x_2, y)\}$

Definition 3.5 *For a WeFA graph (N, A, F), where x is in N, the Choice function is defined as follows, where for all K, $Choice(K, x) = x$ or $Choice(K, x) = \neg x$*

$$Choice(K, x) = x \text{ iff } Value(PosContribution(K, x)) > Value(NegContribution(K, x))$$

Example 3.2 *Consider the graph*

$$x_1[+1]y \qquad x_2[-2]y$$

Hence

$Choice(\{\}, y) = \neg y$
$Choice(\{x_1\}, y) = y$
$Choice(\{x_2\}, y) = \neg y$
$Choice(\{x_1, x_2\}, y) = \neg y$

Definition 3.6 *We define the Formulate function as follows: Let (N, A, F) be a WeFA graph. For each x in N, where $Inputs(x)$ is not the empty set, and for each K in $\wp(Inputs(x))$, we form the following formula, where K^* is $Inputs(x) - K$.*

$$Conjunction(K) \wedge NegatedConjunction(K^*) \rightarrow Choice(K, x)$$

Let the set of all such formulae for (N, A, F) be denoted $Formulate(N, A, F)$.

Example 3.3 *Consider the following WeFA graph*

$$x_1[+1]y \qquad x_2[-2]y$$

Here $Formulate(N, A, F)$ gives the following set of formulae:

$x_1 \wedge x_2 \rightarrow \neg y$
$x_1 \wedge \neg x_2 \rightarrow y$
$\neg x_1 \wedge x_2 \rightarrow \neg y$
$\neg x_1 \wedge \neg x_2 \rightarrow \neg y$

The definition for the set *Formulate* means there is some redundancy in the number of formulae. Using classical logic, it is straightforward to assume an equivalent though simpler and more compressed set of formulae. We can define a set of rewrite rules that give an efficient algorithm for doing this simplification.

Example 3.4 *Consider the following WeFA graph.*

$$a[-2]x \qquad b[+1]x \qquad c[+1]x \qquad d[+1]x$$

This gives the following formulae after simplification using classical logic:

$a \wedge \neg(b \wedge c \wedge d) \rightarrow \neg x$ $\quad \neg a \wedge (a \vee b \vee c) \rightarrow x$
$\neg a \wedge \neg b \wedge \neg c \wedge \neg d \rightarrow \neg x$ $\quad b \wedge c \wedge d \rightarrow x$

We can consider how we use these formulae.

Definition 3.7 *For a primed WeFA graph (N, A, F, S), the function Assumptions is defined as follows:*

$$\begin{aligned} &Assumptions(N, A, F, S) = \\ &\quad \{x | S(x) = true\} \\ &\quad \cup \{\neg x \mid S(x) = false\} \\ &\quad \cup \{\alpha \rightarrow x \in Formulate(N, A, F) \mid S(x) \neq true \text{ and } S(x) \neq false\} \\ &\quad \cup \{\alpha \rightarrow \neg x \in Formulate(N, A, F) \mid S(x) \neq true \text{ and } S(x) \neq false\} \\ &\quad \cup \{\neg x \mid S(x) \neq true \text{ and } S(x) \neq false \\ &\qquad \text{and for all } y \in Feeders(x)\ (S(y) \neq true \text{ and } S(y) \neq false)\} \end{aligned}$$

where α is some conjunction of literals.

The set $Assumptions(N, A, F, S)$ is a set of classical formulae that corresponds to (N, A, F, S) when used with classical logic. Before explaining how we use this set, we motivate the sets listed in Definition 3.7 from which it is composed. The first two sets give priority to the scenario. The next two sets are rules with consquents that represent nodes not assigned by the scenario. The last set captures the default assumptions. Essentially, if the graph has a node x where none of the nodes with a path to x being assigned a truth value by the scenario then it is false by default. To illustrate this, consider a graph with just $x[+1]y$ and an empty scenario. Here, none of the nodes in $Feeders(x)$ have an assignment in the scenario, and so by default we can assume that they are all false, and so x is also false. Similarly, none of the nodes in $Feeders(y)$ have an assignment in the scenario, and so by default we can assume that they are all false, and so y is also false. We look at this example in more detail below.

Example 3.5 *Consider the following WeFA graph*

$$x[+1]y$$

giving the following formulae with the Formulate function

$$x \rightarrow y \qquad \neg x \rightarrow \neg y$$

Hence

$$\begin{aligned} &\text{if } S = \{(y, true)\}, \text{ then } Assumptions(N, A, F, S) = \{\neg x, y\} \\ &\text{if } S = \{(y, false)\}, \text{ then } Assumptions(N, A, F, S) = \{\neg x, \neg y\} \\ &\text{if } S = \{(x, true)\}, \text{ then } Assumptions(N, A, F, S) = \{x, x \rightarrow y, \neg x \rightarrow \neg y\} \\ &\text{if } S = \{\}, \text{ then } Assumptions(N, A, F, S) = \{\neg x, x \rightarrow y, \neg x \rightarrow \neg y, \neg y\} \end{aligned}$$

We now consider a couple of simple examples of WeFA graphs with cycles.

Example 3.6 *Consider the following WeFA graph*

$$x[+1]y \qquad y[-1]x$$

giving the following formulae with the Formulate function

$$x \rightarrow y \qquad \neg x \rightarrow \neg y \qquad y \rightarrow \neg x \qquad \neg y \rightarrow \neg x$$

Hence, if $S = \{\}$, then

$$Assumptions(N, A, F, S) = \{\neg x, x \rightarrow y, \neg x \rightarrow \neg y, y \rightarrow \neg x, \neg y \rightarrow \neg x, \neg y\}$$

Example 3.7 *Consider the following WeFA graph*

$$x[+1]y \qquad y[+1]x$$

giving the following formulae with the Formulate function

$$x \rightarrow y \qquad \neg x \rightarrow \neg y \qquad \neg y \rightarrow \neg x \qquad y \rightarrow x$$

Hence, if $S = \{\}$*, then*

$$Assumptions(N, A, F, S) = \{\neg x, x \rightarrow y, \neg x \rightarrow \neg y, \neg y \rightarrow \neg x, y \rightarrow x, \neg y\}$$

We now consider the logical reasoning with the set $Assumptions(N, A, F, S)$.

Definition 3.8 *Let* $\vdash$ *be the classical consequence relation.*

Definition 3.9 *For a primed WeFA graph* (N, A, F, S)*, the Literals function is defined as follows:*

$$Literals(N, A, F, S) = \{x \in N \mid Assumptions(N, A, F, S) \vdash x \text{ and } x \text{ is a literal } \}$$

Example 3.8 *Consider the following WeFA graph.*

$$x_1[-1]y \qquad x_2[-1]y \qquad x_3[+2]y$$

This is can be represented by the following formulae:

$$x_1 \wedge x_2 \rightarrow \neg y$$
$$\neg x_3 \rightarrow \neg y$$
$$(\neg x_1 \vee \neg x_2) \wedge x_3 \rightarrow y$$

Suppose, we now have a scenario $S = \{(x_1, true), (x_2, false), (x_3, true)\}$*. Here we would get the literals* x_1*,* $\neg x_2$*, and* x_3*, which together with the other formulae, would allow us to derive* y *as an inference using classical logic. So* $Literals(N, A, F, S) = \{x_1, \neg x_2, x_3, y\}$*.*

We can show that the logic-based semantics is equivalent to the definition for Binary WeFA.

Proposition 3.1 *For a primed WeFA graph* (N, A, F, S)*, and a node* $x \in N$*, the following equivalence holds, where* E *is the complete extension of* (N, A, F, S)

$$x \in Literals(N, A, F, S) \text{ iff } E(x) = true$$

$$\neg x \in Literals(N, A, F, S,) \text{ iff } E(x) = false$$

Proof: ($\Rightarrow$) We need to construct the complete extension E from $Literals(N, A, F, S)$ First assume $x \in Literals(N, A, F, S)$. Hence $Assumption(N, A, F, S) \vdash x$, and x is a positive literal. From this, one of the following two cases follow: (Case 1) $S(x) =$ true; and (Case 2) $\alpha \rightarrow x \in Formulate(N, A, F)$ and $Assumptions(N, A, F, S) \vdash \alpha$. If Case 1, then $E(x) =$ true. If case 2, then there is a $K \subseteq Inputs(x)$ such that the following hold,

$$Conjunction(K) \wedge NegatedConjunction(K^*) \rightarrow x$$

$$Assumptions(N, A, F, S) \vdash Conjunction(K) \wedge NegatedConjunction(K^*)$$

and hence, $Value(PosConjunction(K,x)) > Value(NegConjunction(K,x))$. Since there is a K s.t. $Assumptions(N,A,F,S) \vdash Conjunction(K) \wedge NegatedConjunction(K^*)$, we can assume that by recursion, for all $y \in Conjunction(K)$, we have $E(y) =$ true, and for all $y \in NegatedConjunction(K^*)$, we have $E(y) =$ false. From this we can infer $Value(PosFire(E,x)) > Value(NegFire(E,x))$, and hence $E(x) =$ true. Now assume $\neg x \in Literals(N,A,F,S)$. Hence $Assumption(N,A,F,S) \vdash \neg x$, and $\neg x$ is a negative literal. From this, one of the following three cases follow: (Case 3) $S(x)$ = false; (Case 4) $\alpha \rightarrow \neg x \in Formulate(N,A,F)$ and $Assumptions(N,A,F,S) \vdash \alpha$; and (Case 5) $S(x) \neq true$ and $S(x) \neq false$ and for all $y \in Feeders(x)$ $S(y) \neq true$ and $S(x) \neq false$. If Case 3, then $E(x) =$ false. If Case 4, then the proof is equivalent to that for Case 2. If Case 5, then for every node $y \in Feeders(x)$ we have $E(y) = false$ by recursion. Hence, there is no K such that for all $y \in Feeders(x)$ we have $y \notin K$ and $Value(PosContribution(K,y)) > Value(NegContribution(K,y))$. Therefore, we have $Value(PosFire(E,x)) \leq Value(NegFire(E,x))$, and hence $E(x) = false$. From this we have shown that for every postive literal $x \in Literals(N,A,F,S)$ we have $E(x) = true$ and every negative literal $\neg x \in Literals(N,A,F,S)$, we have $E(x) = false$. All that remains is to show that E is a minimal function. Given the priority of the scenario over the rules in composing the set $Assumptions(N,A,F,S)$, it is not possible to add a literal x to $Assumptions(N,A,F,S)$ and a rule $\alpha \rightarrow \neg x \in Assumptions(N,A,F,S)$. Similarly, it is not possible to add a literal $\neg x$ to $Assumptions(N,A,F,S)$ and a rule $\alpha \rightarrow x \in Assumptions(N,A,F,S)$ when $Assumptions(N,A,F,S) \vdash \alpha$. Also by Definition 3.6, it is not possible to have rules $\alpha \rightarrow x$ and $\beta \rightarrow \neg x$ in $Assumptions(N,A,F,S)$ with $Assumptions(N,A,F,S) \vdash \alpha \wedge \beta$. Hence, E is a function. The minimality of E comes from the negative literals added to $Assumptions(N,A,F,S)$ for each node that is not fixed by the scenario and not fixed by the aggregation of the input nodes. ($\Leftarrow$) Follows similarly. □

Whilst the syntactic definition for Binary WeFA is simple and intuitive, and can be informally presented to business users, it is important to have the semantic definition to better understand the logical foundations of the proposal. As a result, we can compare it with other logic-based approaches to argumentation and it also can facilitate the development of tool support.

4 Frameworks issues

Here we here consider some of the options that WeFA gives us, and in particular consider some further concepts that we can use to apply Binary WeFA more fruitfully. These concepts and definitions constitute a framework supporting the syntax and semantics of Binary WeFA.

4.1 Modelling options

Now that we have the basic definitions for syntax and semantics for Binary WeFA, we consider how we can use the approach to model more complex problems.

Definition 4.1 *For a WeFA graph (N,A,F), and a node x in N where $Inputs(x)$ is the empty set, a splitback involves adding further nodes to N, and arcs from these extra nodes to x.*

Whilst the definition of splitback is quite general, it is intended to capture the refinement of a node so that when its truth value is not fixed by a scenario it is desirable to not just resort to the negation-by-default assumption. In other words, by splitback the truth value of a node can be made conditional on the nodes with positive and negative influence on it.

Example 4.1 *Consider the following node with associated statement and assume that $Inputs(a_1)$*

is the empty set.

a_1 `Project completed in-time and under-budget`

For a splitback, we can add the following nodes and arcs.

b_1 `Estimates are accurate`
b_2 `Estimates are approximate`
b_3 `Project management is reliable`
b_4 `Project management could be problematical`

$b_1[+2]a_1$ $b_2[-1]a_1$ $b_3[+2]a_1$ $b_4[-1]a_1$

So now we have provided some of the conditions on a1 being true or false. So if the truth value of a_1 is not established in a scenario, a context sensitive truth value assignment for a_1 can be made on the basis of the truth assignments for b_1 to b_4.

Definition 4.2 *For a WeFA graph (N, A, F), and a node x in N, a splitdown involves replacing x and the arcs involving x with two or more new nodes and associated arcs that influence the nodes in $Output(x)$.*

The splitdown definition is intended to give a finer grained representation of the events that may be observed or predicted.

Example 4.2 *Consider the following nodes and arc.*

a_1 `Price rise in project supplies`
b_1 `Inflation is high`

$b_1[+2]a_1$

Here b_1 may be regarded as too vague. It could be replaced by the following nodes and arcs

b_2 `Inflation is below 2%`
b_3 `Inflation is between 2% and 5%`
b_4 `Inflation is between 5% and 10%`
b_5 `Inflation is above 10%`

$b_2[-1]a_1$ $b_3[+1]a_1$ $b_4[+2]a_1$ $b_5[+3]a_1$

One of the things we see with the above example is that with refinement we can handle quantified information by translating the ranges into a digitized or interval form. In this way, each value or interval in the original quantified range is represented by a truth-valued node.

Example 4.3 *Consider the following nodes with associated statements:*

a_1 `Project completed within budget`
b_1 `Estimates are accurate`
b_2 `Project management is reliable`
b_3 `Inflation rises significantly`
b_4 `Inflation falls significantly`
b_5 `Exchange rate moves significantly in our favour`
b_6 `Exchange rate moves significantly against our favour`

with the following graph

$$b_1[+1]a_1 \qquad b_2[+2]a_1 \qquad b_3[-2]a_1 \qquad b_4[+1]a_1 \qquad b_5[+1]a_1 \qquad b_6[-1]a_1$$

Here there may be uncertainty about node b_3. In particular it may be necessary to consider the conditions under which b_3 is normally true. This could be done by the following splitback involving the following two extra nodes.

c_1 `Unemployment falls sharply`
c_2 `Bank of England is strongly maintaining an anti-inflation policy`

and the following extra arcs:

$$c_1[+1]b_3 \qquad c_2[-2]b_3$$

Alternatively, it may be felt that node b_3 is too coarse-grained. To address this, a splitdown could replace b_3 with the following nodes

b_7 `Inflation is between 5% and 8%`
b_8 `Inflation is between 8% and 12%`
b_9 `Inflation is above 12%`

and add the following arcs

$$b_7[+1]a_1 \qquad b_8[+2]a_1 \qquad b_9[+3]a_1.$$

We can extend the notation used in representing WeFA graphs to indicate a constraint that a set of nodes is mutually exclusive. In other words, if a set of nodes is marked as mutually exclusive then at most one of the nodes in the set can be true in any consistent extension. So in the above example, we could state that $\{b_7, b_8, b_9\}$ are mutually exclusive.

If we record the evolution of refinements in drawing up a WeFA graph as a sequence of splitdown and splitback operations, then we have a relationship between structure and substructure. This can allow for different resolutions of viewing of a WeFA graph.

4.2 Connectivity options

There are a number of kinds of connectivity within WeFA graphs that we need to characterize and consider. These kinds of connectivity are permitted by the definition of WeFA graphs. Some kinds may be regarded as problematical or undesirable. Once we have a formal characterization, we can use tool support to identify them.

Definition 4.3 *Let (N, A, F) be a WeFA graph, where $x, y \in N$ and there is a path from x to y. If there are zero or an even number of negative arcs in the path from x to y, then x has a positive influence on y, otherwise x has a negative influence on y.*

Definition 4.4 *A pair of nodes x and y are independent if the following conditions hold: (1) x is not a positive influence on y; (2) x is not a negative influence on y; (3) y is not a positive influence on x; and (4) y is not a negative influence on x.*

Normally, if y and z are in $Inputs(x)$, then we would want y and z to be independent.

Definition 4.5 *For a WeFA graph (N, A, F), there is a bidirectional link between a pair of nodes x and y in N iff there is an arc $x[n]y$ in A and an arc $y[m]x$ in A where n and m are integers.*

Example 4.4 *As an example of a graph with a bidirectional link, consider the following WeFA graph, with the scenario* $S = \{(x, true)\}$*, which gives the extension* $E = \{(x, true), (y, true), (z, true)\}$*.*

$$x[+1]y \qquad y[+1]z \qquad z[+1]y$$

A bidirectional link is a type of cycle in a WeFA graph. The motivation for including bidirectional links, and cycles generally, is that some variables are not independent.

Definition 4.6 *For a WeFA graph* (N, A, F)*, where* x*,* y*,* a*, and* b *are four different nodes in* N*, if there is a path from* x *to* y *via* a *but not via* b *that respects the directionality of the arcs and if there is a path from* x *to* y *via* b *but not via* a *that respects the directionality of the arcs, then there is a multiple path from* x *to* y*.*

A multiple path emanating from a node x in a graph indicates that the variable x can influence the aim in more than one way. In argumentation terms, it means x is a factor in more than one argument.

Definition 4.7 *For a WeFA graph* (N, A, F)*, where* x *and* y *are in* N*, there is an ambiguous influence of node* x *on node* y *iff there are two arcs* $x[n]y$ *and* $x[m]y$ *in* A *such that* $n > 0$ *and* $m < 0$*.*

An ambiguous influence does not cause any problems in using a Binary WeFA graph. However it is a form of redundancy and it decreases the lucidity of the graph.

4.3 Reasoning options

Using classical proof theory, we can automate reasoning with WeFA graphs. Using the definition of Binary WeFA semantics, we can represent each WeFA graph and scenario as a set of logical formulae and then these formulae can be used directly with a classical logic theorem prover. There are a number of prototype automated reasoning tools that could be used for this. Alternatively, since the formulae are relatively simple, a theorem prover could be implemented based on a semantic tableau algorithm. The advantage of this approach is that extra domain knowledge such as general rules and constraints can be added as extra logical formulae in the logical reasoning. The disadvantage is the increase in complexity in understanding how to use the technology. In general, the disadvantages will greatly outweigh the advantages for the intended target users of Binary WeFA.

For automating the reasoning with WeFA graphs, the recommended option is to implement the application of scenarios using the propagation definition given in the formalization of syntax of Binary WeFA. This definition can be straightforwardly translated into an algorithm for implementation.

If using additional domain knowledge is of particular value, then an alternative approach could be based on Prolog. This is a programming language that supports declarative representation of knowledge such as rules and is ideally suited to implementing simple propositional logic theorem provers.

5 Application issues

In order to harness the formal framework for Binary WeFA, we need to consider strategies for using WeFA for application problems. In particular, we need strategies for knowledge elicitation, scenario analysis, and knowledge management.

5.1 Strategies for knowledge elicitation

Knowledge elicitation is an important activity in WeFA and is central to the process of constructing a WeFA graph. As we indicated in Section 1, we assume that to undertake WeFA we need a meeting of stakeholders or experts.

(1) Domain understanding: The group should have adequate coverage of the problem domain. If key aspects of an organization's activities in the problem area are not represented, then it is likely that a suboptimal analysis will be conducted. During the meeting, participants should communicate the essence of their perspective to the other participants in order to build up a common understanding of the problem domain.

(2) Definition of the WeFA aim: If this has not been specified in advance, the working group should agree on a specification of the aim. Remember that this is captured as a statement that can potentially be true or false.

(3) Collation of suggestions for influences on a factor in the graph (first-time around this will be the aim of the graph): During an open session, or perhaps via break-out sessions, a list of suggestions for influences on the aim should be collated. In addition to the suggestions for factors, there will be the need to suggest the polarity of the influence.

(4) Refinement of suggestions for influences on the aim: The suggestions will inevitably include some overlapping suggestions for factors. Furthermore, suggestions that are not sufficiently clear should be refined. This stage may require some negotiation and conflict resolution between the participants.

(5) Negotiation over the significance of the factors: Given the factors and polarity of influence, the working group needs to decide on the relative significance of the factors.

(6) If the graph requires splitdown or splitback on a factor in the graph, then go through the sequence (3) to (6) again.

(7) At the end of each cycle of (3) to (6), the graph should be checked to avoid ambiguous influences and to identify mutually exclusive sets of factors. In addition, the graph may need to be refined to increase lucidity and reusability.

(8) Once the working group agrees that the WeFA graph is sufficiently developed, scenario analysis can be used to validate the graph for completeness and consistency with expectations.

An alternative to the recursive breakdown strategy where the group builds the graph layer-by-layer, is to first list and agree the majority of factors to be included in the model before considering the polarity, significance, and level of each factor.

In general, there is a difficult trade-off in building a WeFA graph between producing a comprehensive model and producing an easily understandable model. In any case, features to try to avoid are multiple paths, bidirectional arcs, and high connectivity. These features are allowed and can be unavoidable. However they can render a WeFA graph difficult to read and moreover may indicate

that a problem has not been adequately or appropriately decomposed.

5.2 Strategies for scenario analysis

In a WeFA graph, there are a number of factors represented that can directly or indirectly influence the aim of the graph. So the outcome of the aim of the graph is regarded as contingent on the truth value of the other factors in the graph. To analyse this contingency, we use scenario analysis. Different scenarios tell us different things about the problem being analysed. Of particular interest are scenarios that could be described as extreme or normal:

Best-case scenarios scenarios that give the best outcome for the aim of the graph.

Worst-case scenarios scenarios that give the worst outcome for the aim of the graph.

Average-case scenarios scenarios that could be described as normal.

The average-case scenario is a scenario that is one of the most likely scenario whereas the extreme case scenarios are not necessarily likely. Given that it would normally take too long to exhaustively consider all scenarios using key scenarios such as these is important.

We can also work backwards in WeFA graphs: We start with a particular outcome for the aim and see which combinations of factor leads to that scenario outcome. A specialization of this is to also assume the organizational variables have been fixed in order to see which environmental variables need to be true for this aim to be true.

In addition to considering individual scenarios, there is the need to consider sets of scenarios. These can tell us about the sensitivity of the graph to changes in the scenario. If the behaviour of the outcome appears very sensitive to the input, then this may indicate that the deciding factors for the desired outcome of the aim have not been adequately established. In particular if the outcome is sensitive to environmental factors (factors that are not under the control of the organization), then the outcome of the aim may be susceptible to the vagaries of the environment. This kind of scenario analysis is called sensitivity analysis.

A specialization of sensitivity analysis is to order the scenarios according to likelihood. If the truth (or falsity) of a node, including the aim of the graph, is relatively stable in inferred truth value for the more likely scenarios, then there is increased confidence that uncertainty has been minimized in the model.

Another kind of scenario analysis is temporal analysis. Here a sequence of scenarios is meant to indicate how inputs to a problem may evolve over time. For example if we have a WeFA graph that models a project management problem, we may have a sequence of scenarios S_1, S_2, S_3, etc, where S_1 is the scenario for month one, S_2, is the scenario for month two, and S_3 is the scenario for month three. Here, we can see how the outcome of the aim is affected by the sequence of scenarios. We can also do temporal analysis with extreme or normal scenarios and we can combine it with sensitivity analysis.

A specialization of temporal analysis is to only fix environmental variables in the scenarios, and then see if organizational variables can be manipulated to ensure the aim of the graph is true. This kind of analysis tests whether an organization has enough control to respond to problems (or opportunities) arising in the environment.

In addition to looking at different scenarios in scenario analysis, we can manipulate the values given for significance. If we increase the significance for a driver, then we are taking a more

optimistic view on that driver, and if we decrease the significance, then we are taking a more pessimistic view on that driver. In contrast, if we increase the significance for an inhibitor, then we are taking a more pessimistic view on that inhibitor, and if we decrease the significance then we are taking a more optimistic view on that inhibitor. Identifying the ranges of significance for particular outcomes can be helpful in determining the confidence in a model.

5.3 Strategies for knowledge management

WeFA has been proposed for high and strategic problem analysis and in particular for analysing the risks and rewards involved in decision making. Normally, in making big decisions, the focus is forward, we worry about how the decision will be affected by other factors we try to delineate the uncertainties and we try to predict the likely outcomes.

Yet implicit in this process is the role of looking backwards. We try to bring similar decision making situations into consideration to see if they can enable understanding of the current decision and help in predicting the outcomes. Since a WeFA graph is a record of an analysis of a problem, it can be recycled. In other words, it can be used in the future by people with a similar problem.

Recycling may involve adapting the WeFA graph. Recycling may also involve evaluating the actual real-world outcome of the WeFA graph, and adapting the graph to avoid problems reoccurring. This also leads to viewing WeFA graphs as a mechanism for supporting auditing of decision making in an organization, and hence supporting a form of organizational feedback and learning such as argued for by Butler [3].

In order to benefit from recycling, there are some issues that need to be addressed. First, the question of lucidity of the WeFA graph is even more important if the graph is to be used by other people who have not been involved in the original problem analysis. This means that extra effort has to be directed at choosing meaningful labels for nodes, and for providing clear self-contained associated text.

In addition, there is a need to consider appropriate indexing of the documentation. Suppose a repository of WeFA graphs is maintained for use by members of an organization, there is a need to collate and index them for straightforward retrieval in response to future needs.

6 Conclusions

In this report, we have provided a formalization of the syntax and semantics for representing and reasoning with Weighted Factors Analysis (WeFA). The version of WeFA we consider here is Binary WeFA.

Binary WeFA provides an intuitive and apparently simple way of modelling influences of events on other events and goals. The approach uses and adapts key ideas from WeFA and puts them into a logic-based framework with default reasoning. We now have a clear definition of factors, and of polarity and significance of influence, in terms of logic. Using these definitions allows for both a clear presentation of a WeFA graph, and for propagating scenarios in a WeFA graph.

In addition, the formal basis of Binary WeFA can be built upon using further techniques from uncertainty modelling and argumentation for more sophisticated modelling and reasoning. However, if more complexity is introduced into Binary WeFA graphs, such as explicit probabilistic reasoning or temporal reasoning, it may decrease the acceptability and uptake of WeFA for general use in problem solving.

An informal modelling technique, that is related to WeFA, is based on cognitive maps. In a cognitive map, a directed labelled graph is used to capture the structure of a decision-maker's stated beliefs about a particular problem [1]. Some cognitive maps can be used for a form of factors analysis. As with WeFA, the use of cognitive maps has been proposed for modelling high level issues. But the approach does not incorporate a formal logical basis.

WeFA does involve representing aspects of uncertainty and causality. These are issues reflected in a range of other approaches. In Bayesian networks (for a review see [11]), a directed acyclic graph is used to represent causal relations between random variables. These causal relations are used to identify independence assumptions between random variables to facilitate more efficient representation and reasoning with conditional probabilities. Whilst in a sense, Bayesian networks provide a form of factors analysis, the nature of the formalization is fundamentally different from that given by causal mapping. In particular, they provide a means for looking at the ramifications of changes in random variables as dictated by the probability distribution and the axioms of probability theory.

Qualitative probabilistic networks are a qualitative form of probabilistic network [19] (for a review see [13]). Reasoning in qualitative probabilistic networks is qualitatively dictated by a qualitative probability distribution and the axioms of probability theory, and so it is also fundamentally different from reasoning with WeFA graphs. However, in [7] there is a limited formalization of cognitive maps.

Another approach to handling uncertainty and causality is possibilistic networks. In possibilistic networks (for a review see [6]), a directed acyclic graph is used to represent casual relations between possibilistic variables. Possibilistic networks can provide a form of factors analysis, though again the nature of the formalization is fundamentally different from that given by WeFA. In particular, they provide a means for looking at the ramification of changes in possibilistic variables as dictated by the possibility distribution and the axioms of possibility theory.

Finally, Binary WeFA is complementary to a variety of logic-based approaches to argumentation (for examples of formalisms for argumentation see [14, 4, 15, 12, 18, 16, 2]). Whilst Binary WeFA is less expressive than these other proposals, it is more appropriate for non-technical users. In addition, it incorporates the novel formalization of significance.

References

[1] R Axelrod, editor. *Structure of Decisions: The Cognitive Maps of the Political Elites*. Princeton University Press, 1976.

[2] Ph Besnard and A Hunter. Towards a logic-based theory of argumentation. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI'2000)*. MIT Press, 2000.

[3] R Butler. *Designing Organizations*. Routledge, 1991.

[4] M Elvang-Goransson and A Hunter. Argumentative logics: Reasoning from classically inconsistent information. *Data and Knowledge Engineering Journal*, 16:125–145, 1995.

[5] A Fisher. *The Logic of Real Arguments*. Cambridge University Press, 1988.

[6] J Gebhardt and R Kruse. Background and perspectives of possibilistic graphical models. In A Hunter and S Parsons, editors, *Applications of Uncertainty Formalisms*, Lecture Notes in Computer Science. Springer, 1998.

[7] H Geffner. A formal framework for causal modelling and argumentation. In D Gabbay and H-J Ohlbach, editors, *Practical Reasoning*, number 1085 in Lecture Notes in Computer Science. Springer, 1996.

[8] A Hessami. Risk: A missed opportunity. *Risk and Continuity Journal*, 2:17–26, 1999.

[9] A Hunter. Ramification analysis using causal mapping. *Data and Knowledge Engineering*, 32:1–27, 2000.

[10] A Hunter and P McBrien. Default databases: Extending the approach of deductive databases using default logic. *Data and Knowledge Engineering*, 26:135–160, 1998.

[11] F Jenssen. *An Introduction to Bayesian Networks*. UCL Press, 1996.

[12] S Parsons. Defining normative systems for qualitative argumentation. In D Gabbay and H-J Ohlbach, editors, *Practical Reasoning*, volume 1085 of *Lecture Notes in Computer Science*, 1996.

[13] S Parsons. *Qualitative Methods for Reasoning under Uncertainty*. MIT Press, 2001.

[14] H Prakken. An argument framework for default reasoning. In *Annals of Mathematics and Artificial Intelligence*, volume 9, 1993.

[15] H Prakken. *Logical Tools for Modelling Legal Argument*. Kluwer, 1997.

[16] H Prakken and G Vreeswijk. Modelling argumentation in logic. In D Gabbay, editor, *Handbook of Philosophical Logic*. Kluwer, 1999.

[17] S Toulmin. *The Use of Argument*. Cambridge University Press, 1958.

[18] D Vermier, E Laenens, and P Geerts. Defeasible logics. In *Handbook of Defeasible Reasoning and Uncertainty Management*, volume 3. Kluwer, 1998.

[19] M Wellman. Qualitative probabilisitc networks for planning under uncertainty. In J Lemmer and L Kanal, editors, *Uncertainty in Artificial Intelligence 2*. Elsevier, 1988.