Law, Probability and Risk (2016) **15**, 199–221 doi:10.1093/lpr/mgw005
Advance Access publication on July 29, 2016

Extensions and modifications to explanatory coherence

Gerard Vreeswijk[†]
Utrecht University

[Received on 18 January 2016; revised on 3 May 2016; accepted on 12 May 2016]

Thagard's theory of explanatory coherence (TEC) and its implementation ECHO might be considered as the *de facto* calculus of explanatory coherence. It is an elaborate framework to compare competing scientific theories. Recently, it has become apparent that TEC is also useful as a tool for the analysis of different scenarios in so-called sense-making systems. To this end, it is expedient to discuss a number of extensions and modifications to TEC. This article proposes a number of extensions and modifications to TEC in the context of sense-making systems. The following topics are discussed: input format, representation of false formulas, representation languages, relaxation methods, schemes of coherence, meta-explanations, scenarios, leaking hypotheses, knowledge acquisition, and contextual explanation. The discussion is detailed enough to carry through changes in existing sense-making systems.

Keywords: sense-making system; scenario; coherence; theory of explanatory coherence.

1. Introduction

Recently, there is an increasing interest in so-called sense-making systems. A sense-making system is a system in which a certain group collaborates on a typically complex case by inputting and relating data. Sense-making systems help to maintain order, integrity and overview in complex cases. They support the followings tasks: acquiring data from end users; ordering and relating data; drafting scenarios; posing hypotheses; spotting lacunas; putting data in perspective; identifying pertinent consequences. Sense-making systems are especially important in crime analysis (Bex, 2011; Bex *et al.*, 2007; Wagenaar *et al.*, 1993) and more generally in the analysis of so-called *wicked problems* (Ritchey, 2011; Rittel, 1972).[1]

Many sense-making systems are involved in the analysis of different *theories* or *scenarios*. A theory or scenario is a collection of hypotheses of what might be the case, what might happen, or what might *have* happened. Typically the objective is to select 'the best' theory or scenario. There are many different conceptual tools to analyse competing theories or scenarios, ranging from good old tools like mathematical logic and statistics, to more modern tools like Bayesian belief networks, and more controversial tools like diagnostic logics, tools for logical abduction and formal argumentation. A typical disadvantage of many such tools is that they are complex and therefore induce a considerable cognitive load on end users.

[†] Correspondence author. E-mail: gv@uu.nl

[1] Wicked problems are pressing social, economical, environmental or political problems that due to their complexity, vagueness, ambiguity, subjectivity, mixed interests and possibly changing requirements are difficult to manage. It has been claimed that defining a wicked problem is a wicked problem itself. Rittel and Webber formally described the concept of a wicked problem in a 1973 treatise, contrasting wicked problems with relatively 'tame' soluble problems in mathematics and puzzle solving (Rittel and Webber, 1973).

One of the many tools to analyse competing theories is Thagard's theory of explanatory coherence, commonly abbreviated as TEC (Thagard, 1992, 2000). TEC is an elaborate framework to compare competing theories. It is firmly rooted in the epistemological tradition on coherence as advocated by Rescher, Lehrer, Bonjour and others (Audi, 1998; Everitt and Fisher, 1995; O'Brien, 2006). Compared to other approaches, TEC does not bother end users with complex quantitative input but instead accepts simple symbolic input. This input is then processed quantitatively after which again a qualitative result is presented to the end user. Despite of its apparent simplicity, TEC is a mature framework that has proven to be of great value in the analysis of complex cases. Currently, the main sources of TEC are (Thagard, 1992) and (Thagard, 2000).

TEC has always been criticized, especially during its inception, which is roughly through 1985–1995. Indeed, some assumptions behind TEC are controversial. For example, TEC ignores the role of probabilities in scientific reasoning, and TEC does little to illuminate the process of theory evaluation. For an overview of the early criticism, cf. (Thagard, 1992, pp. 89–97). But also recently, in 2005–2015, TEC has been criticized, see e. g. (Olsson, 2005). In Olsson (2005) it is argued that coherentism is untenable as an epistemological theory. Most (but not all) criticism has seriously and often convincingly been dealt with by Thagard and others, most notably in (Thagard, 1992, 2000) and (Thagard and Findlay, 2011). (In Thagard and Findlay (2011), Thagard responds to Olsson (2005).) Since then, science progressed. TEC has been utilized in many other areas, such as argumentation (Pasquier and Draa, 2005) and negotiation (Pasquier and Draa, 2005). Furthermore, alternative formalisms for coherence have been proposed (Danenberg and Marsella, 2010; Mackonis, 2013; Schoch, 2000).

Notwithstanding the fact that TEC has been criticized over the years, its conceptual elegance and simplicity still makes it an attractive tool. Based on this motivation, I decided to re-implement TEC and see on which aspects it could be improved. As a result, a number of extensions and modifications suggested themselves. These extensions and modifications concern input format, representation of false formulas, representation languages, relaxation methods, schemes of coherence, meta-explanations, scenarios, leaking hypotheses, knowledge acquisition, and contextual explanation. The discussion is detailed enough to be carried through in existing sense-making systems.

Most proposed modifications and extensions are implemented in a prototype that matches this article. This prototype can be experimented with and is reachable at

`http://www.projects.science.uu.nl/pipo/TEC++/.`

The rest of this article is organized as follows. Section 2 explains the basic concepts and vocabulary of TEC. Sections 3–7 discuss various modifications. Sections 8–12 discuss various additions. The proposed modifications are discussed first, because most of them seem easier to comprehend. Section 13 discusses related work. Section 14 is a brief conclusion.

2. Basic concepts

This article assumes familiarity with TEC. Let me nevertheless briefly repeat the basics. The description will be rather formal. Especially not much motivation will be given for the different principles of coherence. Also TEC's importance for the area of philosophy of science will not be discussed here. For philosophical and scientific backgrounds of TEC the reader is referred to Thagard's main sources, viz. (Thagard, 1992) and (Thagard, 2000).

Syntactically, TEC's input consists of four types of formulas, namely *facts*, *explanations*, *contradictions*, and *analogies*.

- Facts, also called *evidence*, or *data*, occur in the form 'datum P'.
- Explanations occur in the form '$P_1, \ldots, P_n$ explain Q'.
- Contradictions occur in the form 'P contradicts Q'.
- Analogies occur in the form 'P_1 explains Q_1 is analogous to P_2 explains Q_2'.

The P, Q, P_i, Q_j are atomic propositions. There are no logical connectives, like negation or conjunction. The input of TEC's original implementation, ECHO, looks like

```
// Wegener example from Conceptual Revolutions p.184,
// Table 7.2

// E1 - The shape of the Atlantic coastlines match
// E2 - There are several North-South mountain chains
// E3 - Major folding also occurs along East-West lines
// etc.

// explanations
explain((W8,W9),W11)
explain((W11,W4),E19)
// etc.

// contradictions
contradict(E6,NE6)
contradict(E11,NE11)
// etc.

// data
data(E1,E2,E3,E4,E5,E6,E7,E8,E9,E10,E11,E12,E13,E14,E15)
```

Every atom is considered to be a hypothesis by default, unless it is enlisted as data. This example does not contain analogies. If an analogy would occur, it would look like `analogous((H1,E1),(H2,E2))`. Each of the four slots in this analogy can be occupied by a datum or a hypothesis.

From the input, the first step is to construct a so-called *coherence network*

$$c = (G, w),$$

where $G = (V, E)$ is an undirected graph and $w : E \to \mathbb{R}$ is a weight on edges.[2] V consists of all atomic propositions, plus an extra node representing the truth, conventionally named TRUE. The weight function w indicates to what extent nodes attract or repel each other. For example, if $e = v_1v_2$ is an edge, and $w(e) > 0$, and v_1 is accepted, then this is the reason to accept v_2 as well. Similarly, if $w(e) > 0$, and v_1 is rejected, then this is a reason to reject v_2 as well. Also, if $w(e) < 0$, and v_1 is accepted, then this is reason to reject v_2, etc.

A coherence network is constructed as follows. The algorithm starts with an empty graph. Nodes and edges are created 'on demand': if atoms in a formula occur that do not yet occur as nodes in the

[2] I write $\mathbb{R}$ on the right-hand side, because I could never find out whether weights in TEC's coherence graphs are bounded. I guess not, because it is possible to let two atoms, say A and B, take part in arbitrary many competing explanations, say $\{A, B \to P_i | 1 \leq i \leq n\}$ for some $n > 0$. In a moment it will turn out that arbitrary many competing explanations cause the weights between competing atoms to be negative with an arbitrary large amplitude. A similar construction is possible to generate links of positive weight with an arbitrary large amplitude.

network, then these nodes are created in place. If the weight of a non-existing edge must be increased, then first an edge with weight zero is created, of which its weight is increased afterwards.

- *Data priority.* If P is a fact, then the weight of edge P TRUE is increased with a small amount, in TEC with 0.05.
- *Contradiction.* If P contradicts Q, then the weight of edge PQ is decreased with a small amount, in TEC with 0.06.
- *Explanation.* If $P_1, \ldots, P_n$ explain Q, then the weights of edges P_iP_j, $i \neq j$ and P_iQ are increased with a small amount, in TEC with $0.04/n$.
- *Competition.* If the sequence $R_1, \ldots, R_m$ is a competing explanation of Q, then the weights of all edges P_iR_j are decreased with a small amount, in TEC with $0.06/(n+m)$, provided $P_i \neq R_j$.
- *Analogy.* If 'P_1 explains Q_1' and 'P_2 explains Q_2' are analogous explanations, then the weights of edges P_1P_2 and Q_1Q_2 are increased with a small amount, in TEC with 0.04.

For example, suppose A explains D, and B, C explain D, and A contradicts E, and E is data. Then

$$\begin{array}{rcl} A \text{ explains } D & \Rightarrow & AD \mathrel{+}= 0.04; \\ B, C \text{ explains } D & \Rightarrow & BC \mathrel{+}= 0.02, BD \mathrel{+}= 0.02, \text{ and } CD \mathrel{+}= 0.02; \\ & \Rightarrow & AB \mathrel{-}= 0.01 \text{ and } AC \mathrel{-}= 0.01 \text{ through competition}; \\ A \text{ contradicts } E & \Rightarrow & AE \mathrel{-}= 0.06; \\ E \text{ is data} & \Rightarrow & E \text{ TRUE} \mathrel{+}= 0.05. \end{array}$$

Once a coherence network $c = (G, w)$ is constructed, it is possible to compute c's coherence under different node activations $a : V \to [-1, 1]$ by

$$\text{coherence}(c) = \sum_{ij \in E} a_i \cdot w(ij) \cdot a_j \tag{1}$$

where $a_i \in [-1, 1]$ is the activation of node i, and $w(ij) \in \mathbb{R}$ is the weight of edge ij.

The objective is to maximize c's coherence. This is also known as to 'relax' the network. TEC's method to relax a network is based on a connectionistic method described in McClelland and Rumelhart (McClelland, 2013; McClelland and Rumelhart, 1987), and works as follows. In iterations $t = 1, 2, \ldots$ all nodes are updated simultaneously according to

$$a_j(t+1) = (1-d)a_j(t) + \begin{cases} net_j(t)(\max - a_j(t)) & \text{if } net_j \geq 0, \\ net_j(t)(a_j(t) - \min) & \text{otherwise,} \end{cases} \tag{2}$$

where

$$net_j = \sum_i a_i \cdot w_{ij}. \tag{3}$$

is the *net* input to node j and where the parameter d is a decay factor. Furthermore, the parameters 'min' and 'max' represent minimal and maximal activation. Typically $d = 0.05$, $\min = -1$ and $\max = 1$.

McClelland *et al.*'s relaxation process can be understood as follows. The most important part of (2) is $net_j(t)$ which determines whether the activation of a_j will increase or decrease. If $net_j(t)$ is positive, then the evidence that a_j should be accepted is strong and the activation will increase. Similarly, if

$net_j(t)$ is negative, then a_j's activation will decrease. The 'max $- a_j(t)$'and '$a_j(t) -$ min' terms ensure that the activation remains between 'min' and 'max'. If an activation falls outside [min, max] it is set to either 'min' or 'max', whichever is closest. The $(1 - d)a_j(t)$ part of the equation ensures that activation converges to zero in the absence of external input.

Once activation has converged, atoms with a positive activation are considered accepted (and typically coloured green in many implementations). Atoms with a negative activation are considered to be rejected (and typically coloured red in many implementations). In my implementation, I let the absolute value of the activation determine the colour saturation of nodes.

Here the description of TEC ends. The rest of this article proposes improvements and extensions to especially technical and conceptual parts of TEC.

3. Alternative input format

The input format of TEC's original implementation, ECHO, strongly resembles list processing language (LISP). This is because the first implementations of ECHO were in LISP.[3] Consequently ECHO's input format is sprinkled with parentheses. A disadvantage of this much parentheses is that they do not reflect the conceptual simplicity of the input. Another shortcoming of ECHO's input format is that it is not possible to enter descriptions of atoms. It is of course always possible in ECHO to enter descriptions as comment (see ECHO's input above), but since comment is not parsed, all inputted descriptions will not be seen as official input, hence will get lost eventually.

Figure 1 shows an alternative and slightly extended input format. In contrast with ECHO's original input format, the alternative input format does away with the many parentheses. It also parses descriptions of atomic propositions (which might be hypotheses or evidence), and enables the input and description of so-called *scenarios*. (Scenarios are dealt with in Section 9.) Also case titles (here: 'The Lavoisier example from pages 83–84 of Conceptual Revolutions') are parsed and entered as structural information.

In contrast with ECHO's original input format, the type of a proposition is determined by its position in the input. For example, an atom is considered to be a datum (piece of evidence) if and only if it occurs below a line that matches the regular expression

```
^\s*evidence\s*={10,}
```

Any other atom is considered to be a hypothesis.[4] If a hypothesis occurs under a line that matches

```
^\s*(?:scenario|theory)\s*(.*?)\s*=+\s*(.*?)\s*=+\s*$
```

it is assumed that this hypothesis belongs to the scenario with name first match and description second match. If a hypothesis occurs under no scenario marker at all, it is considered to belong to all scenarios. In my implementation, I used the new input format.

4. Alternative representation of false formulas

If in TEC one wants to express that a certain proposition, say *A*, is considered true, it is entered as a datum. In ECHO that would be `data(A)`. If TEC processes the input, it links *A* to the special

[3] Later in C, and still later in Java as a web-applet.

[4] Actually the parser is more general. It recognises all patterns of the form `^\s*(\w+)\s*={10,}` and subsequently sets the reading mode equal to whatever is captured in the first match, `(\w+)`.

```
================================================================
The Lavoisier example from pages 83-84 of Conceptual Revolutions.
All text below the first bar is parsed and incorporated in the
data structure as a description of the case. Parsing stops at
a reasonable point (currently the first empty line, or the first
line with more than ten non-word characters).
................................................................

theory T1 == Phlogiston Theory ==================================

PH1: Combustible bodies contain phlogiston
PH2: Combustible bodies contain matter of heat
PH3: In combustion, phlogiston is given off
// etc.

theory T2 == Oxygen Theory =======================================

OH1: Pure air contains oxygen principle
OH2: Pure air contains matter of fire and heat
OH3: In combustion, oxygen combines with the burning body
// etc.

explanations =====================================================

PH1, PH2, PH3 explains E1
PH1, PH3, PH4 explains E2
PH5, PH6 explains E5
// etc.

evidence =========================================================

E1: In combustion, heat and light are given off
E2: Inflammability is transmittable from one body to another
E3: Combustion only occurs in the presence of pure air
// etc.
```

FIG. 1. New input format. (Here for the Lavoisier case.)

proposition TRUE by increasing the weight of that link with an amount of 0.05 (as explained above, the 0.05 is specific to ECHO).

Such a manoeuvre is not possible if we wish to express that a certain formula, say *B*, is considered false. The problem is that there is not a special proposition, named FALSE, that *B* can be connected to.

Furthermore, it is forbidden in TEC to specify that the negation of *B* is considered true. (The language of TEC does not know of logical connectives.) It is of course possible to create a negative link between *B* and TRUE indirectly by formulating that *B* contradicts TRUE. In that case TEC decreases the weight on the link between *B* and TRUE with an amount of 0.06. Notice the asymmetry in this situation.

To eliminate this asymmetry, a special proposition, named FALSE, can be introduced, together with the possibility to specify that a certain proposition is considered false. This modification is marginal, therefore I did not implement it in my prototype and did not further experiment with it.

5. Alternative representation languages

An attractive property of TEC's representation language (i.e. the input language modulo syntactic details) is its conceptual simplicity. Explanations, analogies and contradictions are constructed from atomic propositions, and there are no logical connectives. The conceptual simplicity of TEC's representation makes the language manageable and reduces cognitive load.

Nevertheless, one can think of extensions of TEC's representation language that enhance its expressiveness. One extension would be to use the language of propositional logic and focus on the (perhaps partial) satisfiability of propositional scenarios. This idea is further elaborated in Section 7. A further extension would be to introduce variables and work with the language of first-order logic. However, TEC is not driven by general schemes of inference so that incorporating the language of first-order logic does not really seem to meet a certain need. In fact, I think it would only complicate matters.

5.1 The semantic web

A much more fruitful extension is in the direction of semantic nets and the semantic web. From a distance, these two concepts are identical. Languages of the semantic web rest on so-called

subject-predicate-object

triples such as '*H*1 is-a Hypothesis' and '*H*1 is-part-of-scenario *S*2'.[5] An advantage of the semantic web vocabulary is that it is relatively finegrained. Predicates, objects and triples can themselves be objects, and subjects can have multiple object values for the same predicate. Therefore, semantic web languages like RDF and OWL can be used to embed TEC in a dedicated (although yet to be defined) ontology of scenario investigation and explanatory coherence (Bex *et al.*, 2007; van den Braak *et al.*, 2007). A similar drill has been exercised for argumentation, which yielded preliminary versions of the so-called argument interchange format (AIF) (Chesñevar *et al.*, 2006; Rahwan *et al.*, 2007). An advantage of a dedicated ontology is that it can directly be used to express the internal structure of the case. (The two triples above are examples in point.) Semantic web languages can furthermore be used to create the necessary context for a case, for example by annotating input, or to connect cases to other cases, all by means of triples. Simple examples include date of entry, person of entry, etc., like '*H*1 is-entered-by John' and '*H*1 is-entered-at June3-2011-07:00pm'. Giving context to cases is vital in applied professional environments such as sense-making systems for crime investigation, cf. (Bex *et al.*, 2007; van den Braak *et al.*, 2007) and others.

[5] In more complex environments, *namespace*-subject-predicate-object quadruples.

TABLE 1. *Maximal coherence achieved with McClelland and gradient ascent*

	McClelland		Gradient ascent	
Case	iterations	max. coherence	iterations	max. coherence
Lavoisier	105	0.83	174	1.92
Darwin	68	2.05	61	3.48
Wegener	140	3.33	200	5.90
Chambers	88	1.70	89	3.53

6. Alternative relaxation methods

In Section 2 it was described how the coherence of a network may be improved (but not necessary maximized) with McClelland *et al.*'s connectionistic method (McClelland, 2013; McClelland and Rumelhart, 1987). A perhaps more straightforward way to improve coherence is by gradient ascent with restarts.

Gradient ascent with restarts works as follows. Input variables (atoms) are denoted by a node set V. Every node can assume an activation value that ranges from -1 to 1. An activation is a function from V to $[-1, 1]$. The set of all activation functions is denoted by $[-1, 1]^V$. The function to maximize is

$$\text{coherence} : [-1, 1]^V \to \mathbb{R} : (a_1, \ldots, a_n) \mapsto \sum_{ij \in E} a_i \cdot w_{ij} \cdot a_j \tag{4}$$

Due to (4), the influence of the activation a_j of input variable $j \in V$ on the global coherence is given by

$$\frac{\partial}{\partial a_j} \text{coherence}(c) = \frac{\partial}{\partial a_j} \sum_{ij \in E} a_i \cdot w_{ij} \cdot a_j.$$

The latter is a particularly simple expression, because all non-neighbours cancel out, which yields exactly $\sum_{ij} a_i \cdot w_{ij} = \text{net}_j$. It follows that, in order to increase coherence, it suffices to repeatedly adapt node activation of variables in small steps δ accordance with (3) and (4),

$$a_j(t + 1) = (1 - \delta)a_j(t) + \delta \cdot \text{net}_j$$

either simultaneously (all nodes together) or in random order (pick a random node and change its activation in the right direction). The latter is known as *stochastic gradient ascent* and is generally considered to be more effective because it introduces non-determinism in search.

My own experiments show that gradient ascent with restarts almost always yield discrete activation values, i.e. values in $\{-1, 1\}$. McClelland's method on the other hand, typically yields less extreme activation values. Interestingly, gradient ascent with restarts lead to a better coherence than the relaxation method of McClelland and Rumelhart used by Thagard. This is illustrated with a few cases in Table 1. In this table the coherence is computed for a few cases that were discussed in the framework of TEC. Thus, if one aims for a better coherence, then gradient ascent with restarts seem to be a better method to achieve that. Still, in most cases both methods yield identical acceptation–rejection divisions, if only for the cases displayed in Table 1. A possible advantage of McClelland's method is that activation amplitude might be interpreted as the degree, or intensity, with which a certain proposition is accepted or rejected. As far as I know, however, Thagard did not elaborate on this aspect of McClelland's method.

7. Alternative schemes of coherence

A problem with TEC is that it cannot deal with the competition among more than two explanations.[6] This calls into need other schemes of coherence that are able to deal with this problem. Let me first describe the problem, and then let me explain how a particular alternative scheme of coherence, such as for example continuous satisfiability (there may very well be other schemes), is able to deal with this problem.

To introduce the problem, let

$a \to p$ and $b \to p$ and $c \to p$

be three competing explanations for p. If we assume that these three explanations are *all* explanations known for p, then TEC prescribes

$$a \sim p, \quad b \sim p, \quad c \sim p, \quad a \nsim b, \quad a \nsim c, \quad \text{and} \quad b \nsim c.$$

(I hope the notation is intuitive: '$\sim$' denotes coherence between two atomic propositions, and '$\nsim$' denotes incoherence between two atomic propositions.) Of these coherence and incoherence relations, the incoherence relations form a problem because they form a negatively connected 3-clique.[7] Therefore, it is impossible to satisfy a, b and c simultaneously. For example, if we decide to accept a, then $a \nsim b$ and $a \nsim c$ prescribe that we must reject b and c, which conflicts with $b \nsim c$. Similarly, other discrete acceptance–rejection valuations, like $b = 1, a = c = -1$ yield incoherence. Continuous acceptance–rejection valuations, like $a = b = c = 0.5$, also fail because they yield sub-optimal coherence at best.

A solution to this problem is to follow the idea that, out of n competing explanations, $n > 1$, only one explanation can win.[8] Continuing our running example with the three explanations of p, this would mean that either

- $a = 1, a \nsim b, a \nsim c, b \sim c$, hence $b = c = -1$, or
- $b = 1, b \nsim a, b \nsim c, a \sim c$, hence $a = c = -1$, or
- $c = 1, c \nsim a, c \nsim b, a \sim b$, hence $a = b = -1$.

The idea is that, if one atom gets accepted, then the other atoms must be rejected. Notice that in all cases it is necessary to add that precisely one element is accepted, otherwise a case such as $a \nsim b$, $a \nsim c$, $b \sim c$ would not rule out $a = -1$, $b = c = 1$, i.e. it would not rule out the mirror setting in which a itself is rejected. Further notice that the 'or' is exclusive which will be of importance later.

[6] Schoch (2000) made a similar remark but aims at something different, namely that his formalism is able to express the incoherence between more than two propositions. Schoch's work is further discussed in Section 13 ('Related work').

[7] A clique is a set of points where all point pairs are connected.

[8] Conceptual Revolutions, page 69: 'Normally, [. . .], if hypotheses are proposed to explain the same evidence, they will be treated as competitors. For example, in the debate over dinosaur extinction, scientists generally treat as contradictory the hypotheses:

1. Dinosaurs became extinct because of a meteorite collision.
2. Dinosaurs became extinct because the sea level fell.

Logically, (1) and (2) could both be true but scientists treat them as conflicting explanations. According to Principle 6, they incohere because both are claimed to explain why dinosaurs became extinct and there is no explanatory relation between them.' (Thagard, 1992, p. 69).

A problem for TEC is that this solution is an exclusive disjunction of three separate coherence networks, which TEC cannot manage. Thus, the above three networks (actually, three network pieces) cannot be combined into one coherence network (one network piece). To see why this is a problem, suppose that we have a case with k propositions that each have, on average, n competing explanations. According to the idea that each cluster of competing explanations generates n different coherence networks, this would in total yield n^k different coherence networks to explore, which clearly is intractable.

This problem is one of the occasions on the basis of which alternative schemes for coherence might be proposed. One such an alternative scheme of coherence is the scheme of *continuous propositional satisfiability*. In the literature, continuous satisfiability is often referred to as fuzzy satisfiability. I purposively avoid the adjective 'fuzzy' here, because it possesses an implacable negative scientific connotation. Moreover, the idea is to use continuous satisfiability only for optimization and not for the interpretation of formulas. (The latter is the approach of fuzzy satisfiability.) Before describing the semantics of continuous propositional logic, and explaining how to satisfy continuous propositions, I first explain how cases can in the spirit of TEC be converted to the language of propositional logic.

It will now be described how an individual case can be converted to the language of propositional logic. First, every piece of evidence, e, is translated into the propositional equivalence

$$e \equiv \text{TRUE}.$$

Further, every contradiction, 'a contradicts b' is translated into

$$a \equiv \neg b.$$

Notice that, although the formula itself is not symmetric, the underlying semantics is.

The translation of explanations is a bit more involved, because it must take into account the closed world assumption which, as we recall, says that in a normal course of affairs precisely one explanation is responsible for a certain proposition. To this end, let us assume for example that 'a and b explain p', and 'c and d and e explain p' and 'f explains p' are all the available explanations for p. (There are no more explanations for p.) Then the following logical formulas are added. First all corresponding ordinary implications $a, b \to p$ and $c, d, e \to p$ and $f \to p$ are added. Then, to ensure that one, and only one, implication is chosen for explanation, the formula

$$p \to \oplus(a \wedge b, c \wedge d \wedge e, f), \tag{5}$$

is added, where $\oplus$ is the exclusive disjunction on multiple coordinates:

$$\oplus(x_1, x_2, x_3) \equiv (x_1 \wedge \neg x_2 \wedge \neg x_3) \vee (\neg x_1 \wedge x_2 \wedge \neg x_3) \vee (\neg x_1 \wedge \neg x_2 \wedge x_3)$$

This formula is a DNF, and equivalent to saying that exactly one of x_1, x_2, x_3 is true. In this way (5) written out in full becomes

$$\begin{array}{lccccccc} p \to & (& (a \wedge b) & \wedge & \neg(c \wedge d \wedge e) & \wedge & \neg f &) \\ \vee & (& \neg(a \wedge b) & \wedge & (c \wedge d \wedge e) & \wedge & \neg f &) \\ \vee & (& \neg(a \wedge b) & \wedge & \neg(c \wedge d \wedge e) & \wedge & f &). \end{array}$$

Another intuitive format (that within a few steps is convertible to a CNF) is

$$\oplus(x_1, x_2, x_3) \equiv (x_1 \vee x_2 \vee x_3) \wedge (x_1 \rightarrow \neg(x_2 \vee x_3)) \wedge (x_2 \rightarrow \neg(x_1 \vee x_3)) \wedge (x_3 \rightarrow \neg(x_1 \vee x_2)).$$

Unfortunately, exclusive disjunction on multiple coordinates cannot be obtained by means of a simple iteration of binary exclusive disjunctions:

$$\oplus(x_1, x_2, x_3) \neq (x_1 \oplus x_2) \oplus x_3 = x_1 \oplus (x_2 \oplus x_3)$$

Fortunately, n competing explanations generate a closed world formula that consists of $n+1$ components, so that the entire conversion remains linear. Consider for instance a number of explanations from Thagard's case on the Darwinian revolution (Thagard, 1992, p. 144).

'DF5, DF6 explains DH1,'	'DH1, DF4 explains DH2,'
'DH2 explains DH3,'	'DH2, DH3, DH4 explains E1,'
'CH1 explains E1,'	'DH2, DH3, DH5 explains E2,'
'CH1 explains E2,'	etc.

Translating for $e1$, for example, yields $dh2 \wedge dh3 \wedge dh4 \rightarrow e1$ and $ch1 \rightarrow e1$ and $e1 \rightarrow ((dh2 \wedge dh3 \wedge dh4) \oplus ch1)$.

Generally, j facts, m inconsistencies, and k propositions with each on the average n different explanations generate $j + m + k(n + 1)$ propositions.

7.1 Continuous satisfiability

Above it was explained how individual cases can in the spirit of TEC be converted into a conjunction of propositions. Let us call such a conjunction K. It is now described how to satisfy K.

First, notice that each simple conventional $\{0, 1\}$ valuation that satisfies K is in fact a solution. Sometimes, however, K is possesses no models (is not satisfiable) or, at the other extreme, possesses multiple models (is satisfiable but in many ways so). If K has multiple models, these models can be considered as multiple possible worlds that the end user can view and compare.[9] The possibility to view the same scenario from multiple angles must clearly be seen as an advantage. Also, it should be indicated to the end user which propositions are always true in all models, which propositions are always false in all models, and which propositions are contingent in all models. Accordingly, the user must be able to switch between the different models (which, by the way, is something different than switching between scenarios). If K is unsatisfiable with discrete valuations $P \rightarrow \{0, 1\}$, then one can try to satisfy K with continuous ('fuzzy') valuations $P \rightarrow [0, 1]$ where logical connectives are conservatively extended to their continuous counterparts. Thus, the conjunction is extended to a so-called t-norm, for example 'min'.

Furthermore, classical negation must be extended from $\{0, 1\}$ to $[0, 1]$. This is almost always done by

$$[0, 1] \rightarrow [0, 1] : x \mapsto 1 - x. \tag{6}$$

[9] The notion 'possible world' differs from the notion 'scenario'. The notion 'scenario' is already reserved here for different purposes.

There are in fact many candidate extensions for negation. It can be shown, however, that all functions that (1) extend classical negation conservatively (i.e. $\neg 0 = 1$, $\neg 1 = 0$), (2) are strictly decreasing (i.e. $x < y \Rightarrow f(y) < f(x)$), and (3) involute $[f(f(x)) = x]$, are so-called conjugates of $x \mapsto 1 - x$ in the group of automorphisms and anti-automorphisms *Map*([0, 1]) (Kruse *et al.*, 1994). So (6) is a logical choice.

Classical disjunction, then, can be derived from the t-norm through one of DeMorgan's laws, i.e. $a \vee b \equiv \neg(\neg a \wedge \neg b)$, which amounts to

$$a \vee b = 1 - (1 - a) \wedge (1 - b) = (\text{in this case})\ 1 - \min\{1 - a, 1 - b\}$$

Accordingly, continuous extensions of the classical disjunction are called t-conorms. A suitable choice that turns out practical later, is to use 'min' as a t-norm, and $x \mapsto 1 - x$ for negation. From these choices it follows that 'max' is the t-conorm.

For implication there are two choices: to define fuzzy $a \to b$ compositionally through $(\neg a) \vee b$ (called *S-implication*) or more by the nature of implication through $b \to c = \inf\{a \mid a \wedge b \leq c\}$ (called *residuum* or *R-implication*). Since S-implication preserves compositionality I implemented that. Together, the collection 'min', 'max', S-implication and standard fuzzy negation is called the Zadeh family of fuzzy operators.

As an example, suppose A explains D, and B, C explain D, and A contradicts E, and E is data. Then

$$\begin{array}{rcl} A \text{ explains } D & \Rightarrow & \text{add } A \to D; \\ B, C \text{ explains } D & \Rightarrow & \text{add } (B \wedge C) \to D; \\ & \Rightarrow & \text{add } D \to (A \wedge \neg(B \wedge C)) \vee (\neg A \wedge (B \wedge C)) \text{ due to competition;} \\ A \text{ contradicts } E & \Rightarrow & \text{add } A \equiv \neg C; \\ E \text{ is data} & \Rightarrow & \text{add } E \equiv \text{TRUE}. \end{array}$$

The arithmetisation of K then becomes

$$\begin{aligned} \min\{\ & \max\{1 - A, D\}, \\ & \max\{1 - \min\{B, C\}, D\}, \\ & \max\{1 - D, \min\{A, 1 - \min\{B, C\}\}, \min\{1 - A, \min\{B, C\}\}\}, \\ & \min\{\max\{1 - A, C\}, \max\{A, 1 - C\}\}, \\ & \min\{\max\{1 - E, 1\}, \max\{E, 0\}\}\ \} \end{aligned}$$

Some automated simplifications can be made, for example the expression

$$\min\{\max\{1 - E, 1\}, \max\{E, 0\}\}$$

reduces to E. In general the arithmetisation of K can become pretty complex but nevertheless remains linear in the size of K. For purposes of efficiency, my implementation pre-compiles the arithmetisation upon each request by applying the languages' `eval` function once on a string representation on the arithmetisation. Such an operation can be performed in almost all scripting languages.

Maximal satisfiability of K can then be sought by means of gradient ascent with restarts to avoid local optima. To this end, the min/max norm is the best candidate because it can be proven that 'min' is the largest t-norm, and 'max' is the smallest t-conorm. Therefore, the truth-value of large conjunctions does not vanish, so that the gradient always points in a specific direction. If we would, for example, use the product norm $x \wedge y = x \cdot y$, then a conjunction of $|K|$ propositions would result in very small

floating point numbers that computers cannot handle. In my implementation, gradient ascent with restarts on arithmetised formulas proved to be rather slow in comparison with gradient ascent on coherence graphs. This was because certain aspects of the implementation were not optimised. For example, I did not implement algebraic rewrite rules to simplify arithmetic expressions, and I did not partitioned the evaluation of partial derivatives according to K. (For K is a grand minimum of individual expressions. To compute the partial derivative with respect to one atom, it is not necessary to evaluate all elements of K.) I expect that if these optimizations are carried out, convergence will be as fast as convergence in coherence graphs. I ground my expectations on the observation that, once the optimization has been carried out, each node only needs to evaluate the values of its direct neighbours once per episode, and the latter also happens with optimization in coherence graphs.

7.2 No unique optimal valuations

The most striking difference with classical TEC on the one hand and propositional satisfiability on the other hand is that with propositional satisfiability unique convergence is no longer assured. In general there may be many limit points. This is a property of discrete (i.e., classical) as well as in continuous (i.e., fuzzy) propositional logic, and can best be understood by considering simple disjunctions. For example, the disjunction $a \vee b$ is satisfied by

$$\{(x, y) \in \{0, 1\}^2 \mid x = 1 \vee y = 1\}$$

in discrete logic, which amounts to three optima, and by

$$\{(x, y) \in [0, 1]^2 \mid x = 1 \vee y = 1\}$$

in continuous logic, which amounts to an infinite number of optima. The same phenomenon then of course also occurs with larger sets of propositions, in particular K. One way to deal with this phenomenon is to register the truth-values of all atoms when an optimum has been reached. At the end of optimization, all minimum and maximum values that atoms have assumed in global optima is displayed, so that the user obtains an impression of the range of values that individual atoms assume in optimal assignments.

The use of continuous satisfiability is interesting for yet another reason. For some cases, for instance Example 4.2.5 (3) of (Thagard, 1992, p. 76):

```
H1 explains NE1          H2 explains E2
H1 explains E2           H2 explains E3
H1 explains E3           E1 contradicts NE1
```

the activation values of all nodes converge to discrete values, i.e. values in $\{0, 1\}$. In particular: $E1, E2, E3, H2 \rightsquigarrow 1.00$, and $H1, NE1 \rightsquigarrow 0.00$. For other cases, for instance Example 4.2.5 (2) of (Thagard, 1992, p. 76):

```
H1 explains NE1          H1 explains E2
E1 contradicts NE1       H1 explains E3
```

the activation values of all nodes inevitably (that is, after every restart) converge to 'soft' values, i.e. values somewhere in [0, 1]. In particular: $E1 \rightsquigarrow 0.97$, $E2 \rightsquigarrow 0.66$, $E3 \rightsquigarrow 0.85$, $H2 \rightsquigarrow 1.0$, $H1 \rightsquigarrow 0.51$, and $NE1 \rightsquigarrow 0.50$. This seems to suggest that continuous propositional satisfiability distinguished between cases with scenarios that are clearly 'in' or 'out', and cases in which such a distinction between

scenarios is less clear. Thus continuous propositional satisfiability seems to be able to approach cases differently than classical TEC, which is an interesting addition.

This completes my list of proposals to modify TEC. The next sections contain proposals to extend TEC.

8. Addition of meta-explanations

TEC does not view explanations themselves as hypotheses. However, explanations might be considered as hypotheses just as well, because explanations are assumptions that, if asked for, must be justified just as another ordinary hypothesis. Thus, it is imaginable that crucial and critical explanations can (must?) be supported by explanations as well. Of course, not all explanations can be supported by meta-explanations, on pain of infinite regress.

If meta-explanations are allowed, users must be enabled to assign names to explanations of their choice, and provide justifications to named explanations. For example, suppose

```
PH1, PH2, PH3 explains E1
```

and suppose we think this explanation must be justified with the help of another explanation, then it must be possible to input something like[10]

```
J7: PH1, PH2, PH3 explain E1
A1, A2 explain J7
// or: B1 contradicts J7
```

This principle can simply be implemented by supplementing the list in Section 2 with an additional principle:

– *Justification.* If J is a name for '$P_1, \ldots, P_n$ explain Q', then the weights of edges JP_i and JQ are increased with a small amount, say 0.04.

Of course, the parser must be adapted to also recognize these new expressions, but if that happens then the concept of meta-justification can be considered as implemented and can do its work. Notice that contradictions and analogies can also, in principle, be justified by meta-explanations.

Another way to justify explanations is to let them be endorsed or validated by domain experts. With this method, the significance of an explanation linearly depends on the number of endorsements of authorities that work on the case. Giving support to input is vital in applied professional environments such as sense-making systems for crime investigation, cf. (Bex *et al.*, 2007; van den Braak *et al.*, 2007) and others. There may be different ways to implement this, which I did not further explored.

9. Addition of scenarios

Often users of TEC wish to compare different scenarios. For example, in crime investigation a user might wish to compare a scenario in which someone acted in good faith against a scenario in which the same person acted with bad intentions (Bex *et al.*, 2007). In philosophy of science one rather speaks of theories. Scenarios and theories are already present in TEC (Thagard, 1992). For example, the Lavoisier case involves two theories, for instance, 'the oxygen theory' versus 'the phlogiston theory'.

[10] Identifier J ('justification') is used here, rather than E ('explanation') because the letter E is already taken since it is used for 'evidence'.

A problem with TEC is that it does not represent scenarios explicitly. A small addition to basic TEC will now be presented to deal with scenarios.

First I give an informal definition of the notion of scenario and then a formal definition. Informally, a scenario is a consistent collection of hypotheses, a 'theory', that explains what might be the case, or what might have happened. A scenario does not necessarily have to be coherent or true, it only needs to be free of contradictions. The formal definition is not very different.

Definition. A *scenario* is a consistent (i.e. contradiction-free) and possibly empty set of hypotheses. Scenarios may overlap. The *plausibility* of a scenario is defined as the average node activation after network convergence.

With n hypotheses, maximally 2^n different scenarios exist. Usually, however, a case typically consists of two to four interesting scenarios at the very most.

Scenarios may be hand-crafted but may also be produced automatically. To avoid generating too many scenarios, an automatically generated scenario is taken to be maximal (in the set-theoretic sense). It follows that if there are k contradictions, say

$$(a_1, b_1), \ldots, (a_k, b_k),$$

there are 2^k automatically generated scenarios, viz. one maximal set that contains $a_1, \ldots, a_{k-1}, a_k$, one maximal set that contains $a_1, \ldots, a_{k-1}, b_k$, one maximal set that contains $a_1, \ldots, b_{k-1}, a_k$, one maximal set that contains $a_1, \ldots, b_{k-1}, b_k$, etc. It follows that the maximal scenarios are

$$H\backslash\{b_1, \ldots, b_{k-1}, b_k\}, H\backslash\{b_1, \ldots, b_{k-1}, a_k\}, H\backslash\{b_1, \ldots, a_{k-1}, b_k\}, \text{ etc.},$$

which is easy to compute.

In a user interface, users must be able to switch between different scenarios. When a scenario is selected for inspection, the idea is that its elements are highlighted and its plausibility (see above) is shown. Other nodes not in the scenario are still present to provide context but are greyed out.

I have implemented the possibility to input different scenarios in a prototype. Cf. Fig. 1 for an input example. Scenarios obtain their own identifier, typically $S1, S2, \ldots$ (or $T1, T2, \ldots$ if we have a case in which we prefer to call scenarios theories). Each scenario identifier is supplemented by a piece of text that is meant as a scenario title or else to shortly describe the scenario. Once a case has been entered, the user is able to switch between different scenarios through the interface. If a scenario is selected, only the nodes and edges between nodes in a scenario are coloured; the rest is greyed out. Moreover, the title (or description) of the scenario is shown together with its plausibility. I did not conduct user experiments on a large scale. My own most important observation of switching between different scenarios is that their relative plausibility can be read off in a glance by simply keeping an eye on node colours. The exact numerical plausibility of each scenario can if necessary and in doubt always be read off.

10. Addition of leaking hypotheses

In activating explanations and suppressing others, TEC uses a kind of closed world assumption (CWA). A CWA is the presumption that what is not in the input does not exist. More specifically, if

$$\text{explanations-for}(e) = \{h_1^i, \ldots, h_{n_i}^i \rightarrow e | 1 \leq i \leq m\} \tag{7}$$

are all explanations entered for a certain proposition e, then the mechanism of TEC is such that one or more of these explanations receive an activation that is above average. This may be called a CWA,

because TEC acts like the input is known to be complete. (Or the input is known to be incomplete but a best answer must be derived from incomplete information.) Thus, for every e, TEC acts like the set explanations $-$ for(e) always contains one or more true and proper explanations for e.

The cases that are studied in Conceptual Revolutions (Thagard, 1992) indeed all have the property that explanatory sets like (7) are complete, i.e. contain at least one explanation of which one may beforehand suspect it might well be the right one. For example, in the Lavoisier case (Thagard, 1992), there are explanations that belong to the phlogiston theory and there are explanations that belong to the oxygen theory. Since nowadays we do know that the oxygen theory is true, we also know that the input contains the right theory, so that TEC has a chance to converge to the right explanations.

What if the input contains propositions that are supported only by spurious explanations? Then according to TEC's CWA the mechanism of TEC still converges to one of these spurious explanations, which obviously is undesired. A way to get rid of the CWA is to introduce, what I call, *leaky explanations*. The idea is to add, for every explained proposition, e, something like

$$\text{leak-for-}e \rightarrow e.$$

A leaky explanation is a dummy explanation that supplements (7) and is meant as a kind of catch-all, or 'slack', that becomes activated if other explanations fail (due to incohere). If after convergence one or more leaky explanations become activated we may conclude that, for these propositions, the right explanations were absent in the input. For example, the input

```
A explains P          A contradicts TRUE
B explains P          B contradicts TRUE
```

yields activation

$$(A, B, P) = (-0.49, -0.49, -043)$$

in conventional TEC. Adding the input 'LEAK explains P' yields activation

$$(A, B, \text{LEAK}, P) = (-0.57, -0.57, 0.37, -037)$$

in conventional TEC, which indicates that both A and B are poor explanations.

In my current implementation, I experimented with leaky explanations simply by entering them explicitly as input. In a more finished implementation, users need not get bothered with entering leaky explanations, or with viewing them in the converged network. Instead, the implementation should generate leaky explanations in the background when parsing the input, and show leaky explanations in the user interface only when requested. Moreover, there might be additional help in the form of a mechanism that filters out the most deficient explanatory sets (7) and presents them to the end user. The user is then warned that some of the inputted explanations remain inactivated and must be replaced by, or be supplemented by, ones that cohere more with the rest of the input and have a better chance of getting activated.

11. Addition of knowledge acquisition

As indicated in (Abdul-Gader and Kozar, 1990) and (Koponen and Pehkonen, 2010), a serious bottleneck in TEC, and actually for all sense-making systems, is to ensure that cases are translated as faithfully as possible into facts, explanations, contradictions and analogies. The problem is that there are so many degrees of freedom. First, one has to choose the proper atoms, then one has to

decide which of these are evidence, and which are hypotheses. Finally one has to fabricate explanations and find appropriate contradictions and analogies, all not necessarily in that order. If there is a method to this process at all, then this method usually is subjective and *ad hoc*. Unfortunately, in Thagard's major works on explanatory coherence the author is not very clear or explicit about how this process might work, or how Thagard executes this process (Thagard, 1992, 2000). There are a large number of extensive descriptions how theoretical principles and considerations translate into ECHO (for example 'ECHO analysis of Darwin' (Thagard, 1992, pp. 140–148)). Thus, there are a large number of extensive descriptions on the *method*. However, there is little description on the way in which explanations are composed (what goes in the antecedent? when are explanations serial and when are they concurrent?) and selected (which implications do we select as explanations and why? which implications are dropped and why?). Thus, there is little description on the *methodology*. I can only briefly describe what I usually do. What I usually do is to start with the facts, and then begin to think what might explain these. Finally, I scan propositions for possible contradictions. Meanwhile I never bother much about analogies unless, perhaps, they clearly stand out as such. This 'method' is, of course, extraordinarily *ad hoc*.

Of course, knowledge acquisition as a discipline is all about ways to make the process of acquiring knowledge more methodical, structured and reproducible. An important and major insight in this respect is that acquired knowledge must be *sound* (are explanations correct?) and *complete* (do we have all relevant explanations?).[11]

11.1 Completeness

To ensure completeness, one might employ what might be called *structural exhaustive knowledge acquisition*. This is not an official term, rather it is my broom term for a method where information by means of an algorithm is structurally retrieved from human subjects on the basis of the data retrieved thus far. The latter implies that structural exhaustive knowledge acquisition is a bootstrap process where already acquired data is used to generate new queries. Querying already acquired data may happen breadth-first or with the help of a priority queue (McGraw and Harbison-Briggs, 1989). A structured acquisition system might, for example, first ask for evidence. For every proposition entered, then, the system may ask recursively for possible explanations of that proposition. At any point a user may stop giving explanations for that particular proposition. The system then resumes querying that proposition if that proposition returns on the top of the stack in breadth-first querying, or if that proposition possesses the highest priority in prioritized querying. Similar queries may then be generated for contradictions and analogies. Structural exhaustive knowledge acquisition is called exhaustive because the algorithm ensures that every proposition will be queried eventually for all modalities (explanation, contradiction, analogy). In this way, the probability that an explanation or some other piece of knowledge is overlooked and not critically reviewed will be very small. Structural exhaustive knowledge acquisition puts honour to its name because it literally proves to be exhaustive for human subjects. The exhaustive character of structured knowledge acquisition can be alleviated by setting lower priorities to propositions that have low priority or are 'off-center' (further away from the main propositions) and by enabling intermediate saves.

An annoying pitfall in structural exhaustive knowledge acquisition is a possible semantic duplication of propositions. This is the phenomenon that unwary users (i.e. most of us) may enter identical

[11] These notions are related to but different from the notions of soundness and completeness in mathematical logic.

propositions under different names. For example we may in the beginning of a KE session enter the proposition that 'Harry smokes'. Suppose the KE system dubs this proposition $P12$. After a save and few nights sleep we (or some collaborator) may then enter 'Harry is a smoker' (or something similar) in response to another query of the KE system, without us knowing that we actually intended to enter something that is already present, viz. $P12$. The system gives this proposition the name $P23$, say, and, unless there is a sophisticated natural language recognition module, does not recognize that $P12$ and $P23$ actually denote the same proposition. There are ways to reduce the likelihood of semantic duplication, cf. (McGraw and Harbison-Briggs, 1989; Milton, 2007). Discussing these, however, is beyond the scope of this article. For more generalistic considerations on automated knowledge-acquisition, cf. for instance (Marcus, 2013).

11.2 Soundness

Compared to completeness there are no formal ways to enforce soundness. Here, honesty is the best policy, which means that domain experts will have to scrutinize the knowledge available and, if possible, must validate it. Again, cf. (McGraw and Harbison-Briggs, 1989; Milton, 2007).

In my current implementation, I programmed a knowledge acquisition module separately. Experimentation has shown that further research is needed to better integrate exhaustive knowledge acquisition with TEC. Especially appropriate timing (when to bother users) and location (where, i.e. at which propositions to bother users) of queries, and the avoidance of semantic duplication contributes to the success of such a knowledge acquisition module.

12. Addition of contextual explanation

Once ECHO presents a division between accepted and rejected propositions, it is not possible to ask why a certain proposition is accepted (or rejected). An important addition to ECHO (and any implementation of TEC for that matter) would be to add a possibility to ask *why* a certain proposition got accepted (or rejected).

There are different ways to add opportunities for justification. One such way is already presented in Conceptual Revolutions (Thagard, 1992), albeit implicitly. More specifically, pp. 87, 146 and 174 of (Thagard, 1992) contains what may be called *star diagrams*[12] of propositions in a converged network. A star diagram (some may say: *wheel diagram*) is a figure where a centre node, say P, is connected to all neighbouring nodes in the coherence graph G. The centre node P is typically selected by an end user who wants to know why P got accepted (or rejected). Numbers along the radii represent weights of the links between P and its neighbours, and numbers besides nodes represent degrees of activation.

The idea of contextual explanation is to generate a verbal explanation on the basis of a star diagram. To avoid the communication of numerical activation values, I made up a rather arbitrary and *ad hoc* division into verbal activation values:

true	[1.00, 099)	not impossible	[0.00, −025)
highly likely	[0.99, 075)	not very likely	[−0.25, −050)
likely	[0.75, 050)	unlikely	[−0.50, −075)
plausible	[0.50, 025)	highly unlikely	[−0.75, −099)
possible	[0.25, 000)	false	[−0.99, −1.00]

[12] My own terminology.

As an example, Fig. 2 depicts the contextual explanation of $W11$ in Thagard's case on Wegener. Since $W11$ is related to exceptionally many other propositions (the usual connectivity of an arbitrary node remains between 2 and 10), Fig. 2 very well illustrates what a detailed explanation may look like.

13. Related work

Although Thagard's work is often cited and applied, there actually is, as far as I know, surprisingly little work published on technical modifications and extensions to TEC, perhaps because TEC is such a mature and established work. In fact it has been critiqued, defended, implemented and re-implemented, for over more than a decade. Nevertheless, during this time a number of modifications were proposed. Of these modifications I discuss work of Danenberg and Marsella (2010), Joseph *et al.* (Joseph, 2010; Joseph and Prakken, 2009; Joseph *et al.*, 2010), Schoch (2000), and Vreeswijk (2005).

Schoch (2000) proposes an alternative way to compute coherence, namely on the basis of consistent sets of literals. I first list Schoch's principles of coherence and then discuss them.

- If E is data, then $\{E\}$ is coherent.
- If $P_1, \ldots, P_n$ explain Q and both $\{P_1, \ldots, P_n, Q\}$ and $\{P_1, \ldots, P_n, \neg Q\}$ are consistent, then Schoch stipulates that $\{P_1, \ldots, P_n, Q\}$ is coherent and $\{P_1, \ldots, P_n, \neg Q\}$ is incoherent.
- If $P_1, \ldots, P_n$ is contradictory or competing then $\{P_1, \ldots, P_n\}$ is incoherent.

All the P_I, Q and E are literals now, instead of atoms. Further notice that contradiction may now involve more than two elements, and that competition seems to have become isolated principle, i.e. disconnected from competing explanations. Finally, a way to deal with analogies is absent in Schoch's system.

Interestingly, Schoch indicates that TEC's coherence function, viz. equation (1), can be generalized to a first-degree polynomial

$$\text{coherence}(c) = \sum_{0 \leq r_1, \ldots, r_n \leq 1} c_{r_1, \ldots, r_n} \cdot a_1^{r_1} \cdots a_n^{r_n} \tag{8}$$

With this insight and based on the above principles, Schoch proposes a coherence function

$$c : \text{ATOMS} \rightarrow \mathbb{R}$$

as follows. If the set $\mathbf{P} = \{P_1, \ldots, P_n\}$ is coherent then $c_{\mathbf{P}} v(P_1) \ldots v(P_n)$ is added as a term to the coherence function, where $c_{\mathbf{P}}$ is a constant that is determined by $\mathbf{P}$, and $v(P_n)$ is the value of atom P_n, if P_n is a positive literal, and one minus the value of atom in P_n, if P_n is a negative literal. Similarly, if $\mathbf{P}$ is incoherent, then a term $-c_{\mathbf{P}} P_1 \ldots P_n$ is added to the coherence function.

Schoch's approach differs from the one presented here because among other things it uses a different t-norm (multiplication instead of 'min') and computes coherence differently. Furthermore, it seems that index sets with different constants seems overly complex. (With n different atoms this would require the definition of 2^{n+1} different constants.) A simpler indexing mechanism with less constants could for example be an indexing that depends only on the cardinality of sets of literals, i.e. $c_{|\mathbf{P}|}$. It is difficult to compare Schoch's approach to the one presented here because Schoch did not seem to have implemented his system. Therefore there are no results to compare with.

Danenberg and Marsella (2010) propose a method to create coherence models directly from data, which they call *data-driven model construction*. An advantage of this approach is that it frees end users from the burden to decide which nodes cohere with which other nodes, thus ameliorating the

$W11$ is highly likely because of the following reasons:

1. $[-0.03]$ $W11$ is a competitor of C4 in the following explanations, and $C4$ is highly unlikely.
   $(W11, W4, W8) \rightarrow E2$ vs. $(C3, C5, C4) \rightarrow E2$
   $(W11, W4, W9) \rightarrow E3$ $(C3, C5, C4) \rightarrow E3$
2. $[-0.04]$ $W11$ is a competitor of $C5$ in the following explanations, and $C5$ is highly unlikely.
   $(W11, W4, W8) \rightarrow E2$ vs. $(C3, C5, C4) \rightarrow E2$
   $(W11, W4, W9) \rightarrow E3$ $(C3, C5, C4) \rightarrow E3$
   $(W11, W5) \rightarrow E5$ $(C6, C5) \rightarrow E5$
3. $[-0.03]$ $W11$ is a competitor of $C6$ in the following explanations, and $C6$ is unlikely.
   $(W11, W5) \rightarrow E5$ vs. $(C6, C5) \rightarrow E5$
4. $[-0.02]$ $W11$ is a competitor of $C9$ in the following explanations, and $C9$ is plausible.
   $(W11, W6, W9) \rightarrow E15$ vs. $(C1, C9) \rightarrow E15$
5. $[+0.02]$ $W11$ explains $E1$, and $E1$ is likely.
6. $[+0.01]$ $W11$ explains $E2$, and $E2$ is likely.
7. $[+0.01]$ $W11$ explains $E3$, and $E3$ is likely.
8. $[+0.02]$ $W11$ explains $E4$, and $E4$ is likely.
9. $[+0.02]$ $W11$ explains $E5$, and $E5$ is likely.
10. $[+0.01]$ $W11$ explains $E13$, and $E13$ is likely.
11. $[+0.01]$ $W11$ explains $E14$, and $E14$ is likely.
12. $[+0.01]$ $W11$ explains $E15$, and $E15$ is likely.
13. $[+0.01]$ $W11$ explains $E16$, and $E16$ is likely.
14. $[+0.01]$ $W11$ explains $E17$, and $E17$ is likely.
15. $[+0.02]$ $W11$ explains $E19$, and $E19$ is likely.
16. $[+0.02]$ $W11$ explains $E20$, and $E20$ is likely.
17. $[+0.07]$ $W11$ is a co-hypothesis of $W4$ in $(W11, W4, W8) \rightarrow E2$, and $W4$ is highly likely.
18. $[+0.07]$ $W11$ is a co-hypothesis of $W4$ in $(W11, W4, W9) \rightarrow E3$, and $W4$ is highly likely.
19. $[+0.07]$ $W11$ is a co-hypothesis of $W4$ in $(W11, W4) \rightarrow E4$, and $W4$ is highly likely.
20. $[+0.07]$ $W11$ is a co-hypothesis of $W4$ in $(W11, W4) \rightarrow E19$, and $W4$ is highly likely.
21. $[+0.11]$ $W11$ is a co-hypothesis of $W5$ in $(W11, W5) \rightarrow E1$, and $W5$ is highly likely.
22. $[+0.11]$ $W11$ is a co-hypothesis of $W5$ in $(W11, W5) \rightarrow E5$, and $W5$ is highly likely.
23. $[+0.11]$ $W11$ is a co-hypothesis of $W5$ in $(W11, W5, W8) \rightarrow E13$, and $W5$ is highly likely.
24. $[+0.11]$ $W11$ is a co-hypothesis of $W5$ in $(W11, W5, W8) \rightarrow E14$, and $W5$ is highly likely.
25. $[+0.11]$ $W11$ is a co-hypothesis of $W5$ in $(W11, W5, W6, W9) \rightarrow E16$, and $W5$ is highly likely.
26. $[+0.11]$ $W11$ is a co-hypothesis of $W5$ in $(W11, W5, W8) \rightarrow E17$, and $W5$ is highly likely.
27. $[+0.11]$ $W11$ is a co-hypothesis of $W5$ in $(W11, W5) \rightarrow E20$, and $W5$ is highly likely.
28. $[+0.02]$ $W11$ is a co-hypothesis of $W6$ in $(W11, W6, W9) \rightarrow E15$, and $W6$ is likely.
29. $[+0.02]$ $W11$ is a co-hypothesis of $W6$ in $(W11, W5, W6, W9) \rightarrow E16$, and $W6$ is likely.
30. $[+0.07]$ $W11$ is explained by W8, and $W8$ is highly likely.
31. $[+0.07]$ $W11$ is a co-hypothesis of $W8$ in $(W11, W4, W8) \rightarrow E2$, and $W8$ is highly likely.
32. $[+0.07]$ $W11$ is a co-hypothesis of $W8$ in $(W11, W5, W8) \rightarrow E13$, and $W8$ is highly likely.
33. $[+0.07]$ $W11$ is a co-hypothesis of $W8$ in $(W11, W5, W8) \rightarrow E14$, and $W8$ is highly likely.
34. $[+0.07]$ $W11$ is a co-hypothesis of $W8$ in $(W11, W5, W8) \rightarrow E17$, and $W8$ is highly likely.
35. $[+0.06]$ $W11$ is explained by W9, and $W9$ is highly likely.
36. $[+0.06]$ $W11$ is a co-hypothesis of $W9$ in $(W11, W4, W9) \rightarrow E3$, and $W9$ is highly likely.
37. $[+0.06]$ $W11$ is a co-hypothesis of $W9$ in $(W11, W6, W9) \rightarrow E15$, and $W9$ is highly likely.
38. $[+0.06]$ $W11$ is a co-hypothesis of $W9$ in $(W11, W5, W6, W9) \rightarrow E16$, and $W9$ is highly likely.

FIG. 2. Contextual explanation of the acceptance of $W11$ in the Wegener case.

knowledge-acquisition bottleneck. Danenberg *et al.*'s approach works as follows. The algorithm starts by forming a complete undirected graph. Then edges corresponding to conditional independencies are deleted. The remaining links are oriented according to Pearl's IC algorithm (Pearl, 2000). The remaining graph is then explored by, what the authors call, perturbation. Each time a target node is set to a maximal (or minimal) activation value and then the network is made to relax. Also arbitrary cross-

sections of the data are singled out and from these cross-sections the graph structure, node activations and internodal relationships are recreated.

Danenberg *et al.* have implemented their approach and tested it on the (at that time pressing) problem whether it was right to invade Iraq. They published a poll ('We need your opinion on Iraq. Take our Iraq War survey!') and received 442 surveys of which 344 were deemed valid. Besides a conclusion concerning mass psychology (which is interesting in itself but irrelevant in the scope of this article), their experience with data preprocessing was positive. In their conclusion they declared that they would like to test mutating craft of the thus prescribed results in a follow-up study, wherein appropriate or inappropriate messages preface the administration of the administration of the instrument and attitude deviation is tested against a null hypothesis.

Joseph *et al.* (Joseph, 2010; Joseph and Prakken, 2009; Joseph *et al.*, 2008, 2010) define a theory of coherence that is inspired on Thagard's work on coherence, but otherwise clearly deviates from it. The main difference with Thagard's work and ours is that nodes are no longer atomic propositions but may in fact be propositional formulas that are constituted from atoms and logical connectives. Also logical provability $|-$ is involved, which complicates matters, because besides a mechanism for coherence also a mechanism for provability (or some other mechanism to verify logical validity) is involved and the two mechanisms must interact sensibly. On top of that, modal belief and norm operators are introduced to enable epistemic and deontic reasoning. Algorithms for computing coherence are described in a PhD Thesis (Joseph, 2010) and elsewhere (Joseph *et al.*, 2008, 2010). In (Joseph, 2010) a reference to a Prolog implementation is given and it is claimed that this implementation has given good results for testing purposes.

Vreeswijk (2005) argues that, to be able to make accurate scientific statements, languages are needed that are more expressive than the language of TEC. In (Vreeswijk, 2005) it is also discussed how the language of TEC might be extended to the language of propositional logic, including additional coherence principles that express the relation between propositions and their sub-formulas. The idea, then, is not to derive a coherence network from the input (à la TEC), but to construct a coherence network right from the input itself. These ideas have been elaborated here in Sections 5 and 6.

14. Conclusion

In this article I described ways to either modify or extend Thagard's mechanism of explanatory coherence.

Among the proposed modifications there is a proposal to work with more intuitive input formats and a proposal to work with input formats that recognize scenarios. Further we find a proposal to represent false formulas, proposals to use a richer representation language, proposals to relaxation methods that are more intuitive and/or are more in line with notions of convergence in system dynamics, and proposals to use alternative schemes of coherence that are based on continuous propositions satisfiability.

Among the proposed extensions there is a proposal to use meta-explanations, use scenarios, use leaking hypotheses, integrate semi-automated knowledge acquisition, and apply contextual explanation.

Most proposed modifications and extensions are implemented in a public prototype that can be experimented with. The address of this prototype is `http://www.projects.science.uu.nl/pipo/TEC++/`.

It appears that a canonical formal theory of coherence at this moment does not exist and maybe cannot exist. (As opposed to, for instance, the theory of probabilistic inference which converged to

Bayesian belief networks.) Therefore the search for a satisfactory theory of coherence still is an ongoing matter, and much interesting work is still to be done.

Acknowledgements

I am very much indebted to the reviewers for their help in improving the text.

References

Abdul-Gader, A. H. and Kozar, K. A. (1990). Discourse analysis for knowledge acquisition: the coherence method. *Journal of Management Information Systems*, **6**(4):61–82.

Audi, R. (1998). *Epistemology: A Contemporary Introduction to the Theory of Knowledge*. Routledge, London.

Bex, F. (2011). *Arguments, Stories and Criminal Evidence: A Formal Hybrid Theory, volume 92 of Law and Philosophy Library*. Springer.

Bex, F., Braak, S. v. d., van Oostendorp, H., Prakken, H., Verheij, B., and Vreeswijk, G. (2007). Sense-making software for crime investigation: How to combine stories and arguments? *Law, Probability and Risk*, **6**(1–4): 145–168

Chesñevar, C., McGinnis, J., Modgil, S., Rahwan, I., Reed, C., Simari, G., South, M., Vreeswijk, G., and Willmott, S. (2006). Towards an argument interchange format. *The Knowledge Engineering Review*, **21**: 293–316.

Danenberg, P. and Marsella, S. (2010). Data-driven coherence models. In *Proc. of the 19th Int. Conf. on Behavior Representation in Modeling and Simulation*, pages 296–303.

Everitt, N. and Fisher, A. (1995). *Modern Epistemology: A New Introduction*. McGraw-Hill.

Joseph, S. (2010). Coherence-Based Computational Agency. PhD thesis, Universitat Autònoma de Barcelona, Barcelona.

Joseph, S. and Prakken, H. (2009). Coherence-driven argumentation to norm consensus. In *Proc. of the 12th Int. Conf. on Artificial Intelligence and Law (ICAIL)*, pages 58–67. ACM.

Joseph, S., Sierra, C., and Schorlemmer, M. (2008). A coherence based framework for institutional agents. In Sichman, J. S., Padget, J., Ossowski, S., and Noriega, P., editors, *Coordination, Organizations, Institutions, and Norms in Agent Systems III,* volume 4870 of *Lecture Notes in Computer Science (LNCS)*, pages 287–300. Springer.

Joseph, S., Sierra, C., Schorlemmer, W. M., and Dellunde, P. (2010). Deductive coherence and norm adoption. *Logic Journal of the IGPL*, **18**(1):118–156.

Koponen, I. and Pehkonen, M. (2010). Coherent knowledge structures of physics represented as concept networks in teacher education. *Science and Education*, **19**:259–282.

Kruse, R., Gebhardt, J., and Klawonn, F. (1994). *Foundations of Fuzzy Systems*. J. Wiley and Sons, Chichester, England.

Mackonis, A. (2013). Inference to the best explanation, coherence and other explanatory virtues. *Synthese*, pages 1–21.

Marcus, S. (2013). *Automating Knowledge Acquisition for Expert Systems*, volume Vol. 57. Springer Science & Business Media.

McClelland, J. L. (2013). Explorations in parallel distributed processing: A handbook of models, programs, and exercises. Published through https://www.stanford.edu/group/pdplab/pdphandbook/. URL last checked at Feb. 11, 2014.

McClelland, J. L. and Rumelhart, D. E. (1987). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. MIT Press.

McGraw, K. and Harbison-Briggs, K. (1989). *Knowledge Acquisition: Principles and Guidelines*. Prentice-Hall, New-York, etc.

Milton, N. R. (2007). *Knowledge Acquisition in Practice: A Step-by-step Guide*. Springer.
O'Brien, D. (2006). *An Introduction to the Theory of Knowledge*. Polity Press.
Olsson, E. J. (2005). *Against Coherence: Truth, Probability, and Justification*. Oxford scholarship online. Oxford University Press.
Pasquier, P. and Draa, B. C. (2005). Agent communication pragmatics: the cognitive coherence approach. *Cognitive Systems Research*, **6**(4):364–395.
Pearl, J. (2000). *Causality: Models, Reasoning, and Inference*. Cambridge UP.
Rahwan, I., Zablith, F., and Reed, C. (2007). Laying the foundations for a world wide argument web. *Artificial Intelligence*, **171**(10-15):897–921.
Ritchey, T. (2011). *Wicked Problems – Social Messes: Decision support Modelling with Morphological Analysis*. Springer.
Rittel, H. W. and Webber, M. M. (1973). Dilemmas in a general theory of planning. *Policy Sciences*, **4**:155–169.
Rittel, H. W. J. (1972). Second generation design methods. In *Interview in Design Methods Group, 5th Anniversary Report,* pages 5–10. DMG Occasional Paper 1. Reprinted in Cross, N. (ed.) Developments in Design Methodology, J. Wiley & Sons, Chichester, Vol. 1984, pp. 317–327.
Schoch, D. (2000). A fuzzy measure for explanatory coherence. *Synthese*, **122:**291–311.
Thagard, P. (1992). *Conceptual Revolutions*. Princeton University Press, Princeton. Paperback edition, 1993. Italian translation published by Guerini e Associati, 1994.
Thagard, P. (2000). *Coherence in Thought and Action*. MIT Press, Cambridge, MA.
Thagard, P. and Findlay, S. (2011). Changing minds about climate change: Belief revision, coherence, and emotion. In Olsson, E. J. and Enqvist, S., editors, *Belief Revision Meets Philosophy of Science*, Logic, Epistemology, and the Unity of Science, pages 329–345. Springer.
van den Braak, S. W., Vreeswijk, G. A. W., and Prakken, H. (2007). AVERs: an argument visualization tool for representing stories about evidence. In *Proc. of the 11th Int. Conf. on Artificial intelligence and law*, ICAIL '07, pages 11–15, New York, NY, USA. ACM.
Vreeswijk, G. A. W. (2005). Direct connectionist methods for scientific theory formation. In Festa, R., Aliseda, A. and Peijnenburg, J., editors, *Cognitive Structures in Scientific Inquiry. Essays in Debate with Theo Kuipers,* volume 84 (2) of *Poznan Studies in the Philosophy of the Sciences and the Humanities*, pages 375–403. Rodopi.
Wagenaar, W. A., van Koppen, P. J., and Crombag, H. F. (1993). *Anchored narratives: The psychology of criminal evidence*. Harvester Wheatsheaf.