“It could be worse, it could be raining”: reliable automatic meteorological forecasting for holiday planning

Matteo Cristani¹, Francesco Domenichini², Claudio Tomazzoli¹, and Luca Viganò³ and Margherita Zorzi¹

¹ Dipartimento di Informatica, Università di Verona, Italy
² ARPAV, Italy
³ Department of Informatics, King’s College London, UK

Abstract. Meteorological forecasting provides reliable predictions about the weather within a given interval of time. The automation of the forecasting process would be helpful in a number of contexts. For instance, when forecasting about underpopulated or small geographic areas is out of the human forecasters’ tasks but is central, e.g., for tourism. In this paper, we start to tame this challenging tasks: we develop a defeasible reasoner for meteorological forecasting, which we evaluate on of a real-world example with applications to tourism and holiday planning.

1 Introduction

In the last ten years or so, meteorological forecasting has become a commonly required web service and meteorological forecasting websites are nowadays one of the most expensive websites for advertisement. Producing a meteorological forecast is, however, an expert task to perform. It is a human activity that is typically provided as a partially automated pipeline in which a first step consists in generating models of the evolution of the weather in a given geographic area. These models are often not directly accessible online because of their sizes, so the forecaster relies on the execution of a sophisticated reasoning on the data, comparing the models and evaluating the confidence degree in a range of possibilities compatible with the models themselves.

Although this process is an expert one, and it is driven by heuristic knowledge of the domain, there exist some traits of the reasoning method that are systematic, and used in force of the nature of the domain itself. In other words, there exists a specific form of meteorological reasoning consisting in the pipeline described above, and possibly in other collateral steps including comparison with real world data against models, and local methods used to provide the inference as terms of comparison in the range itself.

The computer technologies currently available don’t satisfy the requirements of quality in general application configurations. In this paper, we investigate how to simulate the behavior of a meteorologist in this processes, in an extended way
with respect to what has been done in various previous studies, and specifically inspired by the approach followed by Ramos-Soto et al. \cite{27}.

In particular the current technologies suffer from a number of severe drawbacks: (i) They are based on the crisp interpretation of one single model so that the reliability depends on that model; (ii) They refer to vast areas and tend to infer forecasting of smaller areas without considerations of the local effects; (iii) They sometimes include extreme simplifications of the forecast itself, providing information in graphical terms and thus unifying complex judgments into single ones.

It therefore make senses to envision an automated expert technology able to support or even substitute the forecaster in many use cases, including: (i) As a learning tool: when an expert forecaster helps students, or newbies at the workplace, to become experts of the pipeline mentioned above, it would be beneficial to provide a learning environment simulating reasonably correct forecasts. (ii) As a decision support system: comparing human decisions where the level of confidence is not particularly high with decisions made by an AI tool would help the forecaster in providing a better forecast. (iii) When forecasting is not sustainable: an AI tool may substitute entirely humans when the requested forecasting is too finely associated with territory portions. The latter is the main motivation behind our research.

We aim at defining an AI tool for meteorological forecasting that could be applied to real-world situations in which the tourism flow is not intense enough to make the employment of human forecasters acceptable from a business viewpoint, whilst an automated tool, which provides the forecasting, albeit possibly in a not particularly accurate way, would be beneficial.

We use non monotonic reasoning and make thus possible to accommodate conflicting rules in the system, so that the process of decision making not only results fuzzy, but is also subject to revisions and constrained by confidence; To better explain what we focus upon, consider the following example of a specific situation managed by the framework that we devise here.

\textit{Example 1.} Let us consider a remote area, such as a small island in the middle of the Pacific Ocean. This island has a limited tourism flow and it is thus unlikely that forecasting by authorities on the island itself would be provided. Hence, the level of forecasting is derived by mid-scale models on the Ocean that are very unreliable for this and other small islands. In this island, however, there is a satellite high-speed Internet connection and thus a web server can be installed. Moreover, on the island’s coasts and its sole mountain, there exist a group of wind and rain sensors connected to the server through a network. It is therefore possible to install a sophisticated technology that makes a forecast available through a web service to the tourists interested in visiting the island.

We proceed as follows. In Section 2 we introduce the architecture of our framework. In Section 3 we describe the logic and the algorithm we use to produce input rules for a reasoner that will derive a \textit{weather scenario}; we also describe the steps behind the release of the final weather bulletin. Section 4 is
devoted to a real-world example. In Section 6 we discuss related work, and in Section 7 we draw conclusions.

2 System architecture

![Diagram of system architecture]

The system aim is to produce a better weather forecast, given meteorological models and data gathered from the field, as summarized in Figure 1. Figure 2 shows the logic model of the system architecture, which comprises several modules, each one with a single responsibility as described below.

**Source Forecast Map:** this module aims at the retrieval of raw informations from a specific source (i.e., Temperature, Humidity, ...) connecting to a sensor network or a data source. To add a new source to the system (e.g., Sea Status) one will need to extend only the implementation of this module. The output of this module is a “source forecast map”

**Tournament:** this module takes as input source forecast maps and their accuracy and fragility data, and gives forecast rules to be examined by the **Reasoner**; see Section 7 for an informal description of the algorithm.

**Decision Maker:** this module decides which model will give best performances and is thus the actual core of the system.

**Reasoner:** this module is the “brain” of the system as it applies our deduction system to make decisions about which forecast draft is the “best one”.

**Knowledge base:** this is where the knowledge base is stored; the **Reasoner** will access it for reasoning and the **Decision Maker** will increment it after the evaluation of the results of the reasoner.

**Sharp Forecast:** this module provides a mapping from quantitative forecasting (numerical) to qualitative forecasting (words); see Section 3.3 for details.
Smooth Forecast: this module transforms the forecasting expressed in words into natural language sentences suitable to be delivered to the public; see Section 3.3 for details.

Bulletin generator: this module provides visualization of all data, building the output of the system as a “pdf” file or a hypertext.

Fig. 2. Logic model of system architecture

3 A defeasible reasoner for meteorological forecasting

What is commonly intended as “weather forecasting” can be logically model as a conclusion the forecaster derives from a set of premises, by the application of some (both deductive and empirical) rules.

3.1 The logic MeteoLOG

The reasoning process of the weather forecaster is formally built upon technical steps implementing the workflow described in Section 2. In this section, we develop a logical framework called MeteoLOG, that formalizes the hybrid reasoning at the basis of meteorological forecasting. MeteoLOG, informally introduced in [6], benefits from three standard logical approaches: defeasible logic [15], labeled deduction systems [22, 29, 30, 7] and fuzzy/non-deterministic/probabilistic frameworks [10, 2].

We introduce the syntax of formulae and of labels, along with a notion of prevalence, which imports a defeasible flavour into the system. We also provide an intuitive description of the label-elimination algorithm Tournament, which represents the basis of the reasoning process that we develop below.

In this paper, we only deal with ground formulas modeling meteorological forecasting values, which we will call Assertional Maps (AMs). AMs provide
quantitative information and they represent the basic piece of knowledge used for forecasting. In the real world, they are collected worldwide, from different forecasting sites and through a number of different technologies. The internationally accepted set of numerical weather conditions revealed in AMs concerns: Temperature, Pressure, Humidity, Snowfalls, Wind, Precipitations, Visibility.

From an abstract viewpoint, AMs express rough assertions about weather to be processed and evaluated. They are simply represented by suitable predicates on space-time coordinates pointing out a numerical weather condition, expressed in a suitable measuring system. In particular, the temperature is expressed in graders, pressure is expressed in HPa, humidity in percentage, rains are expressed in millimetres, snowfalls in centimetres, visibility is expressed in metres, Wind in Knots, Cloudiness (C) in percentage.

Formally, an AM is a five-ary predicate $Q(x, y, z, \tau', q)$, where $Q$ is a numerical weather condition, $x$, $y$ and $z$ represent geographic coordinates, $\tau'$ represents the forecasting time (the interval of the validity of the assertion) and $q$ is the effective revealed value, represented by a 2-dimensional vector $(v, d)$, where $v$ is the numerical value and $d$ is the direction. For instance, Rain(45.43, 11.80, 06/04/2018, 14:05:00 CET, 5 mm) represents that the assertion Rain on ground level, point of measure (45.43, 11.80) on GPS coordinates, on 06/04/2018 at 14:05:00 CET was 5 millimetres.

Since we are interested in the reasoning process behind the forecasting, we now focus on models experts apply to derive information from AM. We formalize such methods and related notions by means of labels, and import into the formulas additional information such as the precision of the method and the detection time (the instant in which the method has been applied to generate the map). This information is crucial for the forecaster’s work, since the choice of the (as much as possible) correct maps is mainly based on methodological information.

Labeled Assertional Maps (LAMs) are obtained by labelling AMs. This formally models the additional information the forecaster have to evaluate and decide if a rough AM expressing a prediction is admissible for forecasting or not.

Labels represent contextualised methods, i.e., a forecasting method applied to a data gathering sample, performed in a given instant of time, weighted with some accuracy information; they are pairs of the kind $(\lambda, \tau')$, where $\lambda$ represents a model and $\tau'$ represents the instant in which the map has been generated.

Each method can be associated with an accuracy value $\lambda.a$, a function that extracts the accuracy information from the method $\lambda$.

An LAM is then a labeled formula $(\lambda, \tau'): Q(x, y, z, \tau', q)$.

Labeled ground formulas that express numerical weather conditions permits to compare different formulas expressing the same forecasting concept on the basis of different methods and time. Some priority rules allow us to decide what set of sources is the more reliable one. As explained in the following, we will use some priority rules to order AMs, in order to eliminate the ones that do not overcome a given threshold of reliability.

We introduce now some relation between labels. This step also imports a defeasible behaviour in the system. There are two main kind of priority relation.
The first one, called here *quantitative* or *algorithmic* priority, simply automatically check and compare labeled assertions on the basis of their quantitative information (accuracy and time). \((\lambda_1, \tau_1) : Q_1(x, y, z, \tau^r, q_1) \triangleright (\lambda_2, \tau_2) : Q_2(x, y, z, \tau^r, q_2)\), with \(Q_1 = Q_2\) holds in one of the following cases:

1. \(\lambda_1.a > \lambda_2.a\);
2. \(\lambda_1.a = \lambda_2.a\) and \(\tau_1 < \tau_2\)

Notice that more accurate model-based assertions always (quantitatively) prevails, and, up to equal accuracy, prevails the most recent LAM.

Given a set of LAM, each assertional maps is weighted according to the relations described above and the whole set is ordered as a consequence.

The second kind of priority relation is called *experience-based* or *specific*, and models the empirical knowledge of the expert of the domain.

Following the real world setting, we state that a specific priority always prevails on an algorithmic one (w.r.t. the same assertion).

Having defined the syntax of Meteolog, we are currently working at the definition of a suitable semantics and a natural deduction system [26,3] for Meteolog.

### 3.2 The Tournament algorithm

Tournament is an algorithm whose aim is to return a a defeasible theory, given an ordered set of Metarules, a set of accuracies and actual time. Informally, the algorithm maps assertions into defeasible rules and facts; when it finds possible conflicts it generates a set of defeasible conflicting rules and then, using the accuracy information, it generates the priority rules to solve the conflict so that the method with best accuracy prevails; in case of even accuracy, the latest LAM prevails.

Once a set of LAMs has been collected and an accuracy set has been acknowledged, our Tournament algorithm starts with a sifting action on the set of labeled assertions. First, it discharges LAMs that are out of date. Second, it orders LAMs on the basis of priorities, obtaining some AM for each numerical weather condition we are interested in. This operation corresponds to a label-elimination: once priorities have been derived, the majority of information about the forecasting method became useless. As an output of this step, we obtain a set of defeasible rules to be given as input to the Reasoner, which will derive a set of numerical weather condition also called a weather scenario. Priorities of these generated defeasible conflicting rules are given by a function (named “supremacy”) that takes into account two conflicting rules so that, depending on this function definition, the resulting output can differ from both source rules. The pseudocode of the Tournament algorithm is in Appendix 7.

### 3.3 Weather Forecasting Reasoning

Forecasting reasoning can be divided into three steps: (i) the quantitative forecasting (invisible to the final user) the reasoner generates; (ii) the qualitative
forecasting (also invisible to the final user) called in the following *sharp forecasting*; i(ii) the qualitative, natural language based forecasting destined to the final user, called in the following *smoothed forecasting*. Between the first two phases, a mapping between data and a suitable forecasting lexicon occurs.

**From data to pre-bulletin: Sharp Forecasting** Once the final set of reliable assertional maps has been collected, the forecaster can proceed with data analysis and the releasing of the weather bulletin. Meteorological conditions represent a crucial step of the forecasting reasoning. The internationally accepted ranges for meteorological conditions are shown in Figure 3. To each value of the range a precise interval in a suitable measure scale can be associated.

![Fig. 3. Meteorological Conditions](image)

The weather forecasting lexicon: smoothed forecasting Every one has a wide experience in weather forecasting as a final user. It is well known that weather bulletins are offered in a friendly form. For example, if the forecaster deduces that the probability that tomorrow it will rain is very small, she doesn’t release the assertion “Rain(45.43, 11.80, 06/04/2018, 14:05:00 CET, 5mm) with probability 15%” but the understandable natural language sentence “partially cloudy, possible scattered rains”. This final step provides a “smoothing” phase to the output of the previous one in which some adjectives can be added to give evidence to the uncertainty of the event. We don’t fully model this final step of weather forecasting; nonetheless, in our reference implementation we propose an example of a possible automatisation of the human task, leaving the full development for future work (see Section 7).

**4 Reference Implementation**

To illustrate concretely how our approach can fit a real-life scenario, let us consider a weather forecast considering the seaside part of Veneto, our region, which
is located in the north-east of Italy and albeit being not a remote area it exploits several neighbor small touristic places. For the sake of space, but without loss of
generality, we limit the weather forecast to cloud, wind and sea conditions and to only three points; we label these points North, South and Center, the latter representing roughly the position of the famous city of Venice (see Figure 4). Sea Conditions have only one point, representing the sea in the area. We use only two forecasting maps and we limit the time-frame to only two values, representing two and one days after the present: respectively $t_2, t_1, t_0$ We have as input two forecasting sources, coming from different forecasting models such as IFS (also known as ECMWF for European Center Medium Weather Forecast) and GFS (Global Forecast System), plus the map of observations.

The first source obtained with the GFS prevision model asserts, using $N$ for

*form North and $E$ for form East*, at time $t_0$

- **North**: {cloudiness: 90%, Wind: 18 knots N}
- **Center**: {cloudiness: 90%, Wind: 18 knots N}
- **South**: {cloudiness: 90%, Wind: 10 knots N} Sea: 190 cm wave

at time $t_1$

- **North**: {cloudiness: 90%, Wind: 8 knots N}
- **Center**: {cloudiness: 90%, Wind: 8 knots E}
- **South**: {cloudiness: 90%, Wind: 5 knots E,} Sea: 100 cm wave

at time $t_2$

- **North**: {cloudiness: 90%, Wind: 8 knots N}
- **Center**: {cloudiness: 90%, Wind: 8 knots E}
- **South**: {cloudiness: 90%, Wind: 5 knots E,} Sea: 100 cm wave

The second source obtained with ECMWF asserts, at time $t_0$

- **North**: {cloudiness: 90%, Wind: 15 knots NE}
- **Center**: {cloudiness: 90%, Wind: 15 knots NE}
- **South**: {cloudiness: 90%, Wind: 15 knots NE,} Sea: 160 cm wave

at time $t_1$

- **North**: {cloudiness: 75%, Wind: 5 knots NE}
- **Center**: {cloudiness: 75%, Wind: 5 knots NE}
- **South**: {cloudiness: 75%, Wind: 5 knots N,} Sea: 90 cm wave
at time $t_2$

North: \{cloudiness: 30\%, Wind: 5\text{ knots N}\}
Center: \{cloudiness: 30\%, Wind: 5\text{ knots N}\}
South: \{cloudiness: 30\%, Wind: 5\text{ knots N}, \} Sea: 50\text{ cm wave}\n
The observation map, which only relates data at $t_0$ states that
North: \{cloudiness: 90\%, Wind: 15\text{ knots NE}\}
Center: \{cloudiness: 90\%, Wind: 15\text{ knots NE}\}
South: \{cloudiness: 90\%, Wind: 15\text{ knots NE}, \} Sea: 190\text{ cm wave}\n
We know from knowledge experts that ECMWF has a better accuracy than GFS: numerically $a(\text{ECMWF}, t_1) = 0.85$, $a(\text{ECMWF}, t_2) = 0.80$, $a(\text{GFS}, t_1) = 0.45$, $a(\text{GFS}, t_2) = 0.40$. These assertions, using “E” for ECMWF, “G” for GFS, “O” for observation and “C” for “cloudiness”, “W” for “wind” and “S” for “sea conditions” can be represented in our formalism as

\[
\begin{align*}
(G, t_0) & : C(\text{North}, t_0, 90) & (G, t_0) & : C(\text{Center}, t_0, 90) & (G, t_0) & : C(\text{South}, t_0, 90) \\
(G, t_0) & : C(\text{North}, t_1, 90) & (G, t_0) & : C(\text{Center}, t_1, 90) & (G, t_0) & : C(\text{South}, t_1, 90) \\
(G, t_0) & : C(\text{North}, t_2, 90) & (G, t_0) & : C(\text{Center}, t_2, 90) & (G, t_0) & : C(\text{South}, t_2, 90) \\
(E, t_0) & : C(\text{North}, t_0, 75) & (E, t_0) & : C(\text{Center}, t_0, 75) & (E, t_0) & : C(\text{South}, t_0, 75) \\
(E, t_0) & : C(\text{North}, t_2, 50) & (E, t_0) & : C(\text{Center}, t_2, 50) & (E, t_0) & : C(\text{South}, t_2, 50) \\
(O, t_0) & : C(\text{North}, t_0, 90) & (O, t_0) & : C(\text{Center}, t_0, 90) & (O, t_0) & : C(\text{South}, t_0, 90) \\
(G, t_0) & : W(\text{North}, t_0, [N, 18]) & (G, t_0) & : W(\text{Center}, t_0, [N, 18]) & (G, t_0) & : W(\text{South}, t_0, [N, 18]) \\
(G, t_0) & : W(\text{North}, t_1, [N, 8]) & (G, t_0) & : W(\text{Center}, t_1, [E, 8]) & (G, t_0) & : W(\text{South}, t_1, [E, 5]) \\
(G, t_0) & : W(\text{North}, t_2, [N, 8]) & (G, t_0) & : W(\text{Center}, t_2, [E, 8]) & (G, t_0) & : W(\text{South}, t_2, [E, 5]) \\
(E, t_0) & : W(\text{North}, t_0, [NE, 15]) & (E, t_0) & : W(\text{Center}, t_0, [NE, 15]) & (E, t_0) & : W(\text{South}, t_0, [NE, 15]) \\
(E, t_0) & : W(\text{North}, t_1, [NE, 5]) & (E, t_0) & : W(\text{Center}, t_1, [NE, 5]) & (E, t_0) & : W(\text{South}, t_1, [NE, 5]) \\
(E, t_0) & : W(\text{North}, t_2, [N, 5]) & (E, t_0) & : W(\text{Center}, t_2, [N, 5]) & (E, t_0) & : W(\text{South}, t_2, [N, 5]) \\
(O, t_0) & : W(\text{North}, t_0, [NE, 15]) & (O, t_0) & : W(\text{Center}, t_0, [NE, 15]) & (O, t_0) & : W(\text{South}, t_0, [NE, 15]) \\
(G, t_0) & : S(\text{Sea}, t_0, 190) & (G, t_0) & : S(\text{Sea}, t_1, 100) & (G, t_0) & : S(\text{Sea}, t_2, 100) \\
(E, t_0) & : S(\text{Sea}, t_0, 160) & (E, t_0) & : S(\text{Sea}, t_1, 50) & (E, t_0) & : S(\text{Sea}, t_2, 10) \\
(O, t_0) & : S(\text{Sea}, t_0, 190) & (O, t_0) & : S(\text{Sea}, t_1, 100) & (O, t_0) & : S(\text{Sea}, t_2, 10) \\
\end{align*}
\]

This is therefore our set of metarules, so after the Translator has done its elaboration using algorithm described in [7] we can have:
Given this theory, the Reasoner concludes $+\partial CN_{t1}78, +\partial CC_{t1}78, +\partial CS_{t1}78, +\partial WN_{t1}NE6, +\partial WC_{t1}NE6, +\partial WS_{t1}N5, +\partial SE_{t1}65, +\partial CN_{t2}38, +\partial CC_{t2}38, +\partial CS_{t2}38, +\partial WN_{t2}N6, +\partial WC_{t2}N6, +\partial WS_{t2}N5, +\partial SE_{t2}20$.

Therefore, translating numerical value into words, we have at at time $t1$
North : Mostly Cloudy, Light Winds from North East
Center : Mostly Cloudy, Light Winds from North East
South : Mostly Cloudy, Light Winds from North
Sea : Slight

and at time $t_2$

North : Partly Cloudy, Light Winds from North
Center : Partly Cloudy, Light Winds from North
South : Partly Cloudy, Light Winds from North
Sea : Calm

which can be expressed iconographically as in Figure 5.

![Weather forecast](image.png)

**Fig. 5.** Weather forecast at $t_1$ (tomorrow; left) and at $t_2$ (day after tomorrow; right)

## 5 SPINdle

One the **Tournament** algorithm described in the paper produced a defeasible theory, we can process the theory by means of well-established reasoning technologies, such as Spindle. SPINdle is a logic reasoner that can be used to compute the consequence of defeasible logic theories in an efficient and it can be downloaded at [http://spindle.data61.csiro.au/spindle/](http://spindle.data61.csiro.au/spindle/).

### 5.1 Spindle conclusions for rules of the reference implementation

*****************************************************************************
* SPINdle (version 2.2.4) * * Copyright (C) 2009-2014 NICTA Ltd. * * This software and its documentation is distributed under the terms of the * * FSF Lesser GNU Public License (LGPL). * * * * This program comes with ABSOLUTELY NO WARRANTY; This is a free software *
Initialize application context - start
  load application configuration - start
    app.showProgress=false
    app.showStatistics=false
    reasoner.version=2
  load application configuration - end
configurating I/O classes - start
  generating outputter [spindle.io.outputter.DflTheoryOutputter]...success, type=[df1]
  generating outputter [spindle.io.outputter.XmlTheoryOutputter2]...success, type=[xml]
  generating parser [spindle.io.parser.DflTheoryParser2]...success, type=[df1]
  generating parser [spindle.io.parser.XmlTheoryParser2]...success, type=[xml]
configurating I/O classes - end
Initialize application context - end

=== System info: Load theory from url: file:/temp/Meteo_SPINDLE_RULES
=== System info: Theory loaded successfully, theory type: SDL.
=== System info: Theory contains no literal variable or boolean function.
=== System info: transform theory to regular form
=== System info: Generate conclusions.
=== System info: Conclusions.
+D CEt090(X)
+D CNt090(X)
+D CSSt090(X)
+D Seaot0190(X)
+D WCt0NE15(X)
+D WNt0NE15(X)
+D WSt0NE15(X)
+D Wo_t0_5(X)
-D CCet175(X)
-D CCet190(X)
-D CCet230(X)
-D CCgt090(X)
-D CCgt190(X)
-D CCgt290(X)
-D CNet175(X)
-D CNet190(X)
+d CSet175(X)
+d CSet190(X)
+d CSet230(X)
+d CSgt090(X)
+d CSgt190(X)
+d CSgt290(X)
+d CSt090(X)
+d CSt178(X)
+d -CSt188(X)
+d CSt238(X)
+d -CSt268(X)
+d Seaet0160(X)
+d Seaet150(X)
+d Seaet210(X)
+d Seagt0190(X)
+d Seagt1100(X)
+d Seagt2100(X)
+d Seaot0190(X)
+d Seat165(X)
+d -Seat195(X)
+d Seat220(X)
+d -Seat280(X)
+d WCet1NE5(X)
+d WCet2N5(X)
+d WCgt0N18(X)
+d WCgt1E8(X)
+d WCgt2E8(X)
+d WCt0NE15(X)
+d -WCt1E7(X)
+d WCt1NE6(X)
+d WCt2N6(X)
+d -WCt2NE7(X)
+d WNet1NE15(X)
+d WNet1NE5(X)
+d WNet2N5(X)
+d WNgt0N18(X)
+d WNgt1N8(X)
+d WNgt2N8(X)
+d WNet0NE15(X)
+d -WNt1N7(X)
+d WNet1NE6(X)
+d WNet2N6(X)
+d -WNt2NE7(X)
+d WSet0N5(X)
+d WSet1N5(X)
+d WSet2N5(X)
+d WSt0N10(X)
+d WSt1E5(X)
+d WSt2E5(X)
+d WSt0NE15(X)
+d -WSt1E5(X)
+d WSt1N5(X)
+d WSt2N5(X)
+d -WSt2NE5(X)
+d Wo_t0_5(X)
-d -CCT179(X)
-d CCT188(X)
-d -CCT238(X)
-d CCT268(X)
-d CNgt290(X)
-d -CNt178(X)
-d CNt188(X)
-d CNt238(X)
-d -CNt238(X)
-d CNt268(X)
-d -CNt268(X)
-d -CST178(X)
-d CST188(X)
-d -CST238(X)
-d CST268(X)
-d -Seat185(X)
-d Seat195(X)
-d -Seat220(X)
-d Seat280(X)
-d WCT1E7(X)
-d -WCT1NE6(X)
-d -WCT2N6(X)
-d WCT2NE7(X)
-d -WNT1N7(X)
-d -WNT1NE6(X)
-d -WNT2N6(X)
-d WNT2NE7(X)
-d WSt1E5(X)
-d -WSt1N5(X)
-d -WSt2N5(X)
-d WSt2NE5(X)

=====================================================================
== Performance statistics summary ==
=====================================================================
== I/O classes configuration time used: 32 ms
== No. of record(s) found: 1
== --- start

=====================================================================
| No. of | No. of | Loading theory | Transform theory | Remove defeater | Reasoning | used | used | Filename |
|--------|--------|----------------|-----------------|-----------------|----------|------|------|----------|
| 105    | 105    | 0.069 sec      | 0.006 sec       | 0.000 sec       | 0.035 sec| 0.11usha    | 0.63 MB | file:/temp/Meteo_SPINDLE_RULES |
=====================================================================
== --- end

Calling the shutdown routine...
6 Related work
Since the pioneering studies [5,9,23] and further engineering investigations on the commercial solutions [24], a first attempt going in the same direction that we are following in this paper appeared in the 1990s [13] and inspired many specialized studies [19,21,17,12,4,20,10,11]. The ontological approach and the usage of the Internet of Things have been applied to forecasting quite recently [118] and we acknowledge that the main technical inspirations of our framework trace back these works, whereas the main influences come from the usage of non-monotonic deduction systems for sensor-based applications (clearly related to the initial part of the forecasting process) [28,8], and non-monotonic reasoning[16,25,14,15].

7 Conclusions
In this paper, we propose an architecture to support meteorologists in producing weather forecasts. The basic work is a reasoning framework able to simulate in a quite refined way the decision process made by the forecasters in producing weather bulletins. There are several ways of extending this study. The research team includes a forecaster of the ARPA Veneto weather forecasting service, one of the most valuable forecasting services in Italy, who will lead the development of both the definition of the supremacy function and the Tournament algorithm.

We are currently working at the full formal definition the logical framework MeteoLOG. We plan to include more specific features in order to improve the precision of the automatic bulletin, aiming to a completely automatic and (potentially) unsupervised bulletin generator.

References
1. A. Agresta, G. Fattoruso, M. Pollino, F. Pasanisi, C. Tebano, S. De Vito, and G. Di Francia. An ontology framework for flooding forecasting. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 8582 LNCS(PART 4):417–428, 2014.
2. F. Aschieri and M. Zorzi. Non-determinism, non-termination and the strong normalization of system t. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 7941 LNCS:31–47, 2013.
3. F. Aschieri and M. Zorzi. On natural deduction in classical first-order logic: Curry-Howard correspondence, strong normalization and Herbrand’s theorem. Theoretical Computer Science, 625:125–146, 2016.
4. L. Carr III, R. Elsberry, and J. Peak. Beta test of the systematic approach expert system prototype as a tropical cyclone track forecasting aid. Weather and Forecasting, 16(3):355–368, 2001.
5. B. Conway. Expert systems and weather forecasting. *Meteorological Magazine*, 118(1399):23–30, 1989.

6. M. Cristani, F. Domenichini, F. Olivieri, C. Tomazzoli, and M. Zorzi. It could rain: weather forecasting as a reasoning process. In R. J. Howlett, L. C. Jain, Z. Popovic, D. B. Popovic, S. N. Vukosavic, C. Toro, and Y. Hicks, editors, *Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 22nd International Conference KES-2018, Belgrade, Serbia, 3-5 September 2018.*, volume 126 of *Procedia Computer Science*, pages 850–859. Elsevier, 2018.

7. M. Cristani, F. Olivieri, C. Tomazzoli, and M. Zorzi. Towards a logical framework for diagnostic reasoning. In *Agents and Multi-Agent Systems: Technologies and Applications 2018. Proc. of KES-AMSTA-18*, https://doi.org/10.1007/978-3-319-92031-3_1, pages 1–12. Springer, 2018.

8. M. Cristani, C. Tomazzoli, E. Karafili, and F. Olivieri. Defeasible reasoning about electric consumptions. In 30th IEEE International Conference on Advanced Information Networking and Applications, AINA 2016, Crans-Montana, Switzerland, 23-25 March, 2016, pages 885–892, 2016.

9. M. C. Desmarais, F. de Verteuil, P. Zwack, and D. Jacob. Stratus: A prototype expert advisory system for terminal weather forecasting. pages 150–154, 1990.

10. D. Domanska and M. Wojtylak. Application of fuzzy time series models for forecasting pollution concentrations. *Expert Systems with Applications*, 39(9):7673–7679, 2012.

11. D. Domanska and M. Wojtylak. Explorative forecasting of air pollution. *Atmospheric Environment*, 92:19–30, 2014.

12. P. Fohm. An overview of avalanche forecasting models and methods. *Publikasjon - Norges Geotekniske Institutt*, (203):19–27, 1998.

13. E. Goldberg. Fog. synthesizing forecast text directly from weather maps. pages 156–162, 1993.

14. G. Governatori, F. Olivieri, E. Calardo, A. Rotolo, and M. Cristani. Sequence semantics for normative agents. *Lecture Notes in Computer Science* (including subseries *Lecture Notes in Artificial Intelligence* and *Lecture Notes in Bioinformatics*), 9862 LNCS:230–246, 2016.

15. G. Governatori, F. Olivieri, S. Scannapieco, and M. Cristani. The hardness of revising defeasible preferences. *Lecture Notes in Computer Science* (including subseries *Lecture Notes in Artificial Intelligence* and *Lecture Notes in Bioinformatics*), 8620 LNCS:168–177, 2014.

16. G. Governatori, F. Olivieri, S. Scannapieco, A. Rotolo, and M. Cristani. The rationale behind the concept of goal. *Theory and Practice of Logic Programming*, 16(3):296–324, 2016.

17. B. Hansen. Sigmar: a fuzzy expert system for critiquing marine forecasts. *AI Applications*, 11(1):59–68, 1997.

18. A. Kulkarni and D. Mukhopadhyay. Internet of things based weather forecast monitoring system. *Indonesian Journal of Electrical Engineering and Computer Science*, 9(3):555–557, 2018.

19. V. Kumar, C. Chung, and C. Lindley. Toward building an expert system for weather forecasting operations. *Expert Systems With Applications*, 7(2):373–381, 1994.

20. C. Mahabir, F. Hicks, and A. Fayek. Application of fuzzy logic to forecast seasonal runoff. *Hydrological Processes*, 17(18):3749–3762, 2003.

21. R. M. Marra, D. H. Jonassen, and P. Knight. Use of expert system generation to promote knowledge synthesis in a meteorology forecasting course. pages 475–479, 1996.
22. A. Masini, L. Viganò, and M. Zorzi. Modal deduction systems for quantum state transformations. *J. Mult.-Valued Logic Soft Comput.*, 17(5-6):475–519, 2011.
23. J. McCarthy and R. J. Serafin. Advanced aviation weather system based on new weather sensing technologies. pages 228–239, 1990.
24. W. Moninger. Shootout-89, an evaluation of knowledge-based weather forecasting systems. *Machine Intelligence and Pattern Recognition*, 10(C):457–458, 1990.
25. F. Olivieri, G. Governatori, S. Scannapieco, and M. Cristani. Compliant business process design by declarative specifications. *Lecture Notes in Computer Science*, 8291 LNAI:213–228, 2013.
26. D. Prawitz. Ideas and results in proof theory. In J. Fenstad, editor, *Proceedings of the Second Scandinavian Logic Symposium*, volume 63 of *Studies in Logic and the Foundations of Mathematics*, pages 235 – 307. Elsevier, 1971.
27. A. Ramos-Soto, A. Bugarn, S. Barro, and J. Taboada. Linguistic descriptions for automatic generation of textual short-term weather forecasts on real prediction data. *IEEE Transactions on Fuzzy Systems*, 23(1):44–57, 2015.
28. C. Tomazzoli, M. Cristani, E. Karafili, and F. Olivieri. Non-monotonic reasoning rules for energy efficiency. *Journal of Ambient Intelligence and Smart Environments*, 9(3):345–360, 2017.
29. L. Viganò, M. Volpe, and M. Zorzi. Quantum state transformations and branching distributed temporal logic - (invited paper). In *Logic, Language, Information, and Computation - 21st International Workshop, WoLLIC 2014, Valparaíso, Chile, September 1-4, 2014. Proceedings*, pages 1–19, 2014.
30. L. Viganò, M. Volpe, and M. Zorzi. A branching distributed temporal logic for reasoning about entanglement-free quantum state transformations. *Inf. Comput.*, 255:311–333, 2017.
Input: a set of accuracy of the methods $\Psi$, current time $t$, an ordered set of Metarules $\Theta$ in which metarules are in the form label : assertion where labels are in the form (method, time) and assertions are in the form name(position, time, value);

Output: a defeasible theory $T = \{F, R, P\}$ (facts, rules, priorities);

$\Theta' \leftarrow \Theta$

repeat

$m \leftarrow \text{pop}(\Theta')$;

if $\text{label}(m).\text{time} \leq t$ then

if $\text{label}(m).\text{method} = 'O'$ then

$f \leftarrow \text{assertion}(m)$;

else

$r \leftarrow \emptyset \Rightarrow \text{assertion}(m)$; $R \leftarrow \text{push}(r, R)$; // $r$ is in the form $A(r) \Rightarrow C(r)$

name $\leftarrow C(r).\text{name}$; position $\leftarrow C(r).\text{position}$; time $\leftarrow C(r).\text{time}$;

$\overline{R} \leftarrow R - r$;

repeat

$r \leftarrow \text{pop}(\overline{R}); 1 \leftarrow \text{getLabel}(\Theta, C(r));$

if $\text{name} = \text{mname}$ then

if $\text{position} = \text{position}$ then

if $C(r).\text{value} \neq C(\overline{r}).\text{value}$ then

$sr_1 \leftarrow \text{createNewRule}(l);$

$A(sr_1) \leftarrow C(r), C(\overline{r}); C(sr_1) \leftarrow \text{supremacy}(C(r), C(\overline{r})$, first$)$;

$sr_2 \leftarrow \text{createNewRule}(l);$

$A(sr_2) \leftarrow C(r), C(\overline{r}); C(sr_2) \leftarrow \text{supremacy}(C(r), C(\overline{r})$, last$)$;

$vc_1 \leftarrow \text{createNewRule}(l);$

$A(vc_1) \leftarrow C(r), C(vc_1) \leftarrow \neg C(r).\text{value};$

$vc_2 \leftarrow \text{createNewRule}(l);$

$A(vc_2) \leftarrow C(r), C(vc_2) \leftarrow \neg C(r).\text{value};$

accuracy$_1 \leftarrow \text{get}(\Psi, \text{label}(m).\text{method})$; accuracy$_2 \leftarrow \text{get}(\Psi, \text{method})$;

if accuracy$_1 \geq$ accuracy$_2$ then

$p_1 \leftarrow \text{createNewPriority}(sr_1, sr_2)$; $P \leftarrow \text{push}(p_1, P)$;

else if accuracy$_1 <$ accuracy$_2$ then

$p_2 \leftarrow \text{createNewPriority}(sr_2, sr_1)$; $P \leftarrow \text{push}(p_2, P)$;

else

$p_1 \leftarrow \text{createNewPriority}(sr_1, sr_2)$; $P \leftarrow \text{push}(p_1, P)$;

else if $\text{label}(m).\text{time} \geq \overline{r}$.time$)$ then

$p_1 \leftarrow \text{createNewPriority}(sr_2, sr_1)$; $P \leftarrow \text{push}(p_1, P)$;

else

$p_2 \leftarrow \text{createNewPriority}(sr_1, sr_2)$; $P \leftarrow \text{push}(p_2, P)$;

else

$p_1 \leftarrow \text{createNewPriority}(sr_2, sr_1)$; $P \leftarrow \text{push}(p_1, P)$;

else

$p_2 \leftarrow \text{createNewPriority}(sr_1, sr_2)$; $P \leftarrow \text{push}(p_2, P)$;

end if

end if

end if

end if

until $\overline{R} = \emptyset$

end if

until $\Theta' = \emptyset$

return $T = \{F, R, P\}$;

\begin{itemize}
\item \textbf{The Tournament algorithm} A certain rule is a candidate for rewriting, only if the synchroniser acknowledged that its clock time falls within the validity interval of the rule (if the rule has a validity interval) or at the exact instant of the rule if the rule is simply instantaneous.
\end{itemize}
A Another example

We would like make an example of a weather forecast considering our region, located in the north eastern part of Italy, to give a better evidence of how our model can fit a real life scenario. For the sake of space we will make some limitations: we limit the weather forecast to rain conditions and to only four points; we will label these points North, East, South, West. We will use only two forecasting maps and we will limit the time frame to only two values, representing two and one day after the present: respectively $t_2, t_1, t_0$

We have as input two forecasting sources, coming from different forecasting models such as IFS (also known as ECMWF for European Center Medium Weather Forecast) and GFS (Global Forecast System), plus the map of observations. The first source obtained with the GFS prevision model asserts, at time $t_0$ \{North = 5mm, East = 4mm, South = 4mm, West = 4mm\}, at time $t_1$ \{North = 4mm, East = 4mm, South = 4mm, West = 5mm\}, at time $t_2$ \{North = 6mm, East = 6mm, South = 6mm, West = 6mm\}. The second source obtained with the ECMWF prevision model asserts, at time $t_0$ \{North = 5mm, East = 5mm, South = 5mm, West = 5mm\}, at time $t_1$ \{North = 24mm, East = 14mm, South = 24mm, West = 24mm\}, at time $t_2$ \{North = 16mm, East = 16mm, South = 16mm, West = 16mm\}. The observation map, which only relates data at $t_0$ states that \{North = 5mm, East = 5mm, South = 5mm, West = 5mm\}.

We know from knowledge experts that ECMWF has a better accuracy than GFS: numerically $a(ECMWF, t_1) = 0.85, a(ECMWF, t_2) = 0.80, a(GFS, t_1) = 0.45, a(GFS, t_2) = 0.40$.

These assertions, using “E” for ECMWF, “G” for GFS, “O” for observation and “R” for “rain” can be represented as

\[
\langle E, t_0 \rangle : R(North, t_0, 4) \quad \langle E, t_0 \rangle : R(East, t_0, 4) \quad \langle E, t_0 \rangle : R(South, t_0, 4) \quad \langle E, t_0 \rangle : R(West, t_0, 4) \\
\langle E, t_0 \rangle : R(North, t_1, 14) \quad \langle E, t_0 \rangle : R(East, t_1, 14) \quad \langle E, t_0 \rangle : R(South, t_1, 14) \quad \langle E, t_0 \rangle : R(West, t_1, 14) \\
\langle G, t_0 \rangle : R(North, t_0, 5) \quad \langle G, t_0 \rangle : R(East, t_0, 5) \quad \langle G, t_0 \rangle : R(South, t_0, 5) \quad \langle G, t_0 \rangle : R(West, t_0, 5) \\
\langle G, t_0 \rangle : R(North, t_1, 24) \quad \langle G, t_0 \rangle : R(East, t_1, 24) \quad \langle G, t_0 \rangle : R(South, t_1, 24) \quad \langle G, t_0 \rangle : R(West, t_1, 24) \\
\langle O, t_0 \rangle : R(North, t_0, 6) \quad \langle O, t_0 \rangle : R(East, t_0, 6) \quad \langle O, t_0 \rangle : R(South, t_0, 6) \quad \langle O, t_0 \rangle : R(West, t_0, 6)
\]

This is therefore our set of metarules, so after the Translator has done its elaboration using algorithm described in [7] we can have:

\[
\begin{align*}
\text{r}_{911} & : N \rightarrow N_{t_0} \quad \text{r}_{921} : N \rightarrow N_{t_1} \quad \text{r}_{931} : N \rightarrow N_{t_2} \quad \text{r}_{941} : N \rightarrow N_{t_4} \\
\text{r}_{912} & : E \rightarrow E_{t_0} \quad \text{r}_{922} : E \rightarrow E_{t_1} \quad \text{r}_{932} : E \rightarrow E_{t_2} \quad \text{r}_{942} : E \rightarrow E_{t_4} \\
\text{r}_{913} & : S \rightarrow S_{t_0} \quad \text{r}_{923} : S \rightarrow S_{t_1} \quad \text{r}_{933} : S \rightarrow S_{t_2} \quad \text{r}_{943} : S \rightarrow S_{t_4} \\
\text{r}_{914} & : W \rightarrow W_{t_0} \quad \text{r}_{924} : W \rightarrow W_{t_1} \quad \text{r}_{934} : W \rightarrow W_{t_2} \quad \text{r}_{944} : W \rightarrow W_{t_4} \\
\end{align*}
\]

\[
\begin{align*}
\text{r}_{915} & : N \rightarrow N_{t_0} \quad \text{r}_{925} : N \rightarrow N_{t_1} \quad \text{r}_{935} : N \rightarrow N_{t_2} \quad \text{r}_{945} : N \rightarrow N_{t_4} \\
\text{r}_{916} & : E \rightarrow E_{t_0} \quad \text{r}_{926} : E \rightarrow E_{t_1} \quad \text{r}_{936} : E \rightarrow E_{t_2} \quad \text{r}_{946} : E \rightarrow E_{t_4} \\
\text{r}_{917} & : S \rightarrow S_{t_0} \quad \text{r}_{927} : S \rightarrow S_{t_1} \quad \text{r}_{937} : S \rightarrow S_{t_2} \quad \text{r}_{947} : S \rightarrow S_{t_4} \\
\text{r}_{918} & : W \rightarrow W_{t_0} \quad \text{r}_{928} : W \rightarrow W_{t_1} \quad \text{r}_{938} : W \rightarrow W_{t_2} \quad \text{r}_{948} : W \rightarrow W_{t_4} \\
\end{align*}
\]

\[
\begin{align*}
\text{r}_{919} & : N \rightarrow N_{t_0} \quad \text{r}_{929} : N \rightarrow N_{t_1} \quad \text{r}_{939} : N \rightarrow N_{t_2} \quad \text{r}_{949} : N \rightarrow N_{t_4} \\
\text{r}_{9110} & : E \rightarrow E_{t_0} \quad \text{r}_{9210} : E \rightarrow E_{t_1} \quad \text{r}_{9310} : E \rightarrow E_{t_2} \quad \text{r}_{9410} : E \rightarrow E_{t_4} \\
\text{r}_{9111} & : S \rightarrow S_{t_0} \quad \text{r}_{9211} : S \rightarrow S_{t_1} \quad \text{r}_{9311} : S \rightarrow S_{t_2} \quad \text{r}_{9411} : S \rightarrow S_{t_4} \\
\text{r}_{9112} & : W \rightarrow W_{t_0} \quad \text{r}_{9212} : W \rightarrow W_{t_1} \quad \text{r}_{9312} : W \rightarrow W_{t_2} \quad \text{r}_{9412} : W \rightarrow W_{t_4} \\
\end{align*}
\]
Given that theory, the Reasoner concludes $\partial N_{t1} 21$, $\partial E_{t1} 21$, $\partial S_{t1} 21$, $\partial W_{t1} 21$, $\partial \neg N_{t1} 7$, $\partial \neg E_{t1} 7$, $\partial \neg S_{t1} 7$, $\partial \neg W_{t1} 7$, $\partial N_{t2} 14$, $\partial E_{t2} 14$, $\partial S_{t2} 14$, $\partial W_{t2} 14$, $\partial \neg N_{t2} 8$, $\partial \neg E_{t2} 8$, $\partial \neg S_{t2} 8$, $\partial \neg W_{t2} 8$.

Given that, translating numerical value into words, we have at time $t_1$ that {North=Heavy Rain, East=Rain, South=Heavy Rain, West=Heavy Rain} while at time $t_2$ {North=Strong Rain, East=Strong Rain, South=Possible Showers, West=Possible Showers} which translates in the following figures.

A.1 Spindle conclusions for rules of the reference implementation

***************SPINdle (version 2.2.4)***************
* Copyright (C) 2009-2013 NICTA Ltd.
.....
* java -jar spindle-<version>.jar --app.license
***************SPINdle (version 2.2.4)***************

= application start!! =

Initialize application context - start
.....

+\!d E_t0_5(X)
+\!d E_t1_14(X)
+\!d E_t1_21(X)
+d -E_t1_7(X)
+d E_t1_8(X)
+d Ee_t0_5(X)
+d Ee_t1_24(X)
+d Ee_t2_16(X)
+d Eg_t0_4(X)
+d Eg_t1_4(X)
+d Eg_t2_6(X)
+d N_t0_5(X)
+d N_t1_14(X)
+d N_t1_21(X)
+d -N_t1_7(X)
+d N_t1_8(X)
+d Ne_t0_5(X)
+d Ne_t1_24(X)
+d Ne_t2_16(X)
+d Ng_t0_4(X)
+d Ng_t1_4(X)
+d Ng_t2_6(X)
+d S_t0_5(X)
+d S_t1_14(X)
+d S_t1_21(X)
+d -S_t1_7(X)
+d S_t1_8(X)
+d Se_t0_5(X)
+d Se_t1_24(X)
+d Se_t2_16(X)
+d Sg_t0_4(X)
+d Sg_t1_4(X)
+d Sg_t2_6(X)
+d W_t0_5(X)
+d W_t1_14(X)
+d W_t1_21(X)
+d -W_t1_7(X)
+d W_t1_8(X)
+d We_t0_5(X)
+d We_t1_24(X)
+d We_t2_16(X)
+d Wg_t0_4(X)
+d Wg_t1_4(X)
+d Wg_t2_6(X)
-d -E_t1_14(X)
-d -E_t1_21(X)
-d E_t1_7(X)
-d E_t1_8(X)
Calling the shutdown routine... 
Terminate application context - start 
Terminate application context - end 
=============================================================================== 
=== Application shutdown completed! === 
===============================================================================