Weather Information Systems
Towards a Domain Description

Version 1

Dines Bjørner
Fredsvej 11, DK-2840 Holte, Denmark
and
The Technical University of Denmark

November 17, 2016, 14:47

Abstract

This document reports *work in progress*. We show an example domain description.
It is developed and presented as outlined in [Bjø16c]. The domain being described is
that of a generic weather information system. Four main endurants (i.e., aspects) of
a generic weather information system are those of the weather, weather stations (col-
lecting weather data), weather data interpretation (i.e., metereological institute[s]),
and weather forecast consumers. There are, correspondingly, two kinds of weather 1
information: the weather data, and the weather forecasts. These forms of weather
information are acted upon: the weather data interpreter (i.e., a metereological in-
stitute) is gathering weather data; based on such interpretations the metereological
institute is "calculating" weather forecasts; and and weather forecast consumers are
requesting and further "interpreting" (i.e., rendering) such forecasts. Thus weather
data is communicated from weather stations to the weather data interpreter; and
weather forecasts are communicated from the weather data interpreter to the weather
forecast consumers. It is the dual purpose of this technical report to present a do- 2
main description of the essence of generic weather information systems, and to add
to the "pile" [Bjø10b, Bjø10a, Bjø13b, Bjø13a, Bjø13c, Bjø16b, Bjø16a, Bjø16d] of
technical reports that illustrate the use[fulness] of the principles, techniques and tools
of [Bjø16c].

Contents

The present document is based on version 0 of a document:

- **http://www.imm.dtu.dk/~dibj/2016/wis/wimea-p.pdf**.

Instead of updating that document I am now issuing the present document:

- **http://www.imm.dtu.dk/~dibj/2016/wis/wis-p.pdf**.

In this present document changes with respect to the former document are marked with these margin bars:

1 On Weather Information Systems 3

1.1 On a Base Terminology 4

From Wikipedia:

1 **Weather** is the state of the atmosphere, to the degree that it is hot or cold, wet or dry, calm or stormy, clear or cloudy, atmospheric (barometric) pressure: high or low.

2 So weather is characterized by **temperature, humidity** (incl. **rain**, **wind** (direction, velocity, center, incl. its possible mobility), **atmospheric pressure**, etcetera.

3 By **weather information** we mean

- either weather data that characterizes the weather as defined above (Item 1),
- or weather forecast, i.e., a prediction of the state of the atmosphere for a given location and time or time interval.

4 Weather data are collected by **weather stations**. We shall here not be concerned with technical means of weather data collection. 5

5 **Weather forecasts** are used by forecast consumers, anyone: you and me.

6 Weather data interpretation (i.e., **forecasting**) is the science and technology of creating weather forecasts based on **time-** or **time interval-stamped weather data** and **locations**. Weather data interpretation is amongst the charges of meteorological institutes.

7 **Meteorology** is the interdisciplinary scientific study of the atmosphere.

8 An **atmosphere** (from Greek $\alpha\tau\mu o\zeta$ (atmos), meaning "vapour", and $\sigma\phi\alpha\iota\rho\alpha$ (sphaira), meaning "sphere") is a layer of gases surrounding a planet or other material body, that is held in place by the gravity of that body. 6

9 Meteorological institutes work together with the World Meteorological Organization (WMO). Besides weather forecasting, meteorological institutes (and hence WMO) are concerned also with aviation, agricultural, nuclear, maritime, military and environmental meteorology, hydrometeorology and renewable energy.

10 Agricultural meteorologists, soil scientists, agricultural hydrologists, and agronomists are persons concerned with studying the effects of weather and climate on plant distribution, crop yield, water-use efficiency, phenology of plant and animal development, and the energy balance of managed and natural ecosystems. Conversely, they are interested in the rôle of vegetation on climate and weather.

© Dines Bjørner 2016, Fredsvej 11, DK–2840 Holte, Denmark – November 17, 2016: 14:47

7

2 Major Parts of a Weather Information System

We think of the following parts as being of concern in the kind of weather information systems that we shall analyse and describe: Figure 1 shows one **weather** (dashed rounded corner all embracing rectangle), one central **weather data interpreter** (cum meteorological institute) seven **weather stations** (rounded corner squares), nineteen **weather forecast consumers**, and one global **clock**. All are distributed, as hinted at, in some geographical
8 space. Figure 2 on the next page shows "an orderly diagram" of "the same" weather

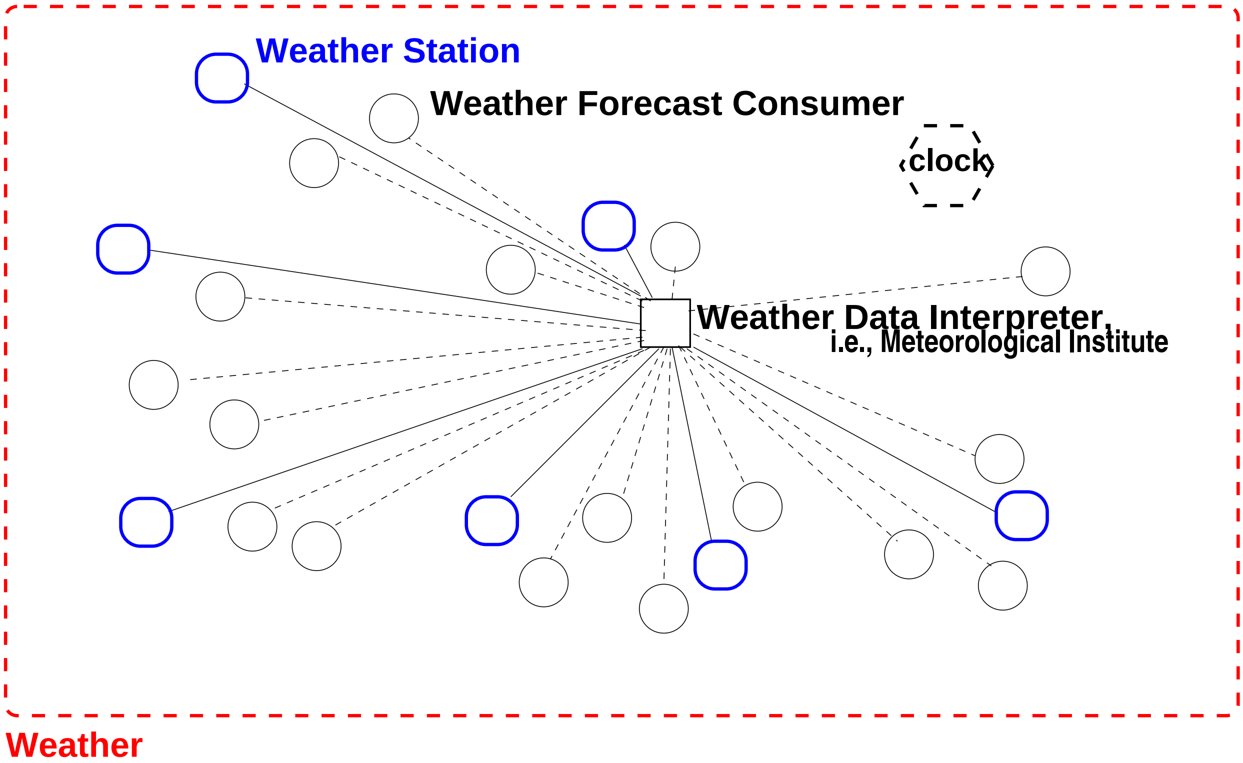

Figure 1: A Weather Information System

9 information system as Figure 1. The lines between pairs of the various parts shall indicate means communication between the pairs of (thus) connected parts. Dashed lines "crossing" bundles of these communication lines are labeled ch_xy. These labels, ch_xy, designated CSP-like channels. An input, by a weather station (wsi), of weather data from the weather (wi), is designated by the CSP expression ch_ws[wi,wsi] ?. An output, say from the weather data interpreter (wdi) to a weather forecast consumer (fci), of a forecast f, is designated by ch_ic[wdii,fci] ! f

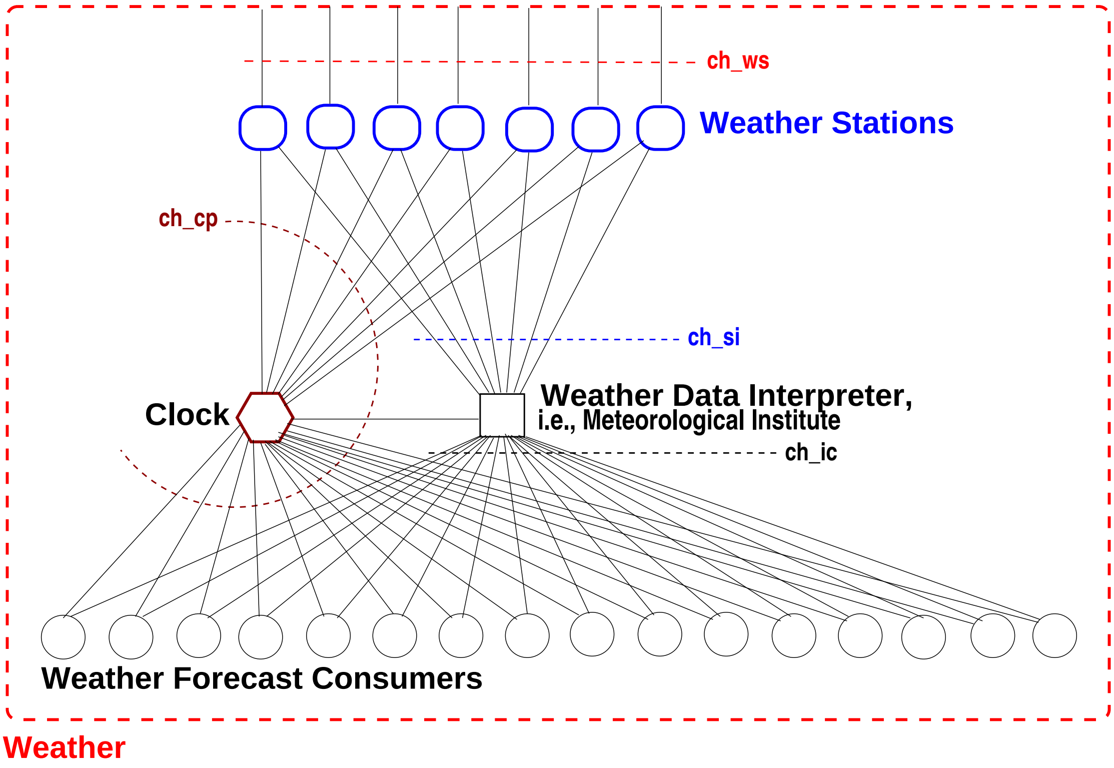

Figure 2: A Weather Information System Diagram

10

3 Endurants

3.1 Parts and Materials

11 The WIS domain contains a number of sub-domains:

a. the weather, W, which we consider a material,

b. the weather stations sub-domain, WSS (a composite part),

c. the weather data interpretation sub-domain, WDIS (an atomic part),

d. the weather forecast consumers sub-domain, WFCS (a composite part), and

e. the ("global") clock (an atomic part).

11

type
11 WIS
11a. W

 © Dines Bjørner 2016, Fredsvej 11, DK–2840 Holte, Denmark – November 17, 2016: 14:47

11b. WSS
11c. WDIS
11d. WFCS
11e. CLK
value
11a. obs_material_W: WIS → W
11b. obs_part_WSS: WIS → WSS
11c. obs_part_WDIS: WIS → WDIS
11d. obs_part_WFCS: WIS → WFCS
11e. obs_part_CLK: WIS → CLK

12

12 The weather station sub-domain, WSS, consists of a set, WSs,

13 of atomic weather stations, WS.

14 The weather forecast consumers sub-domain, WFCS, consists of a set, WFCs,

15 of atomic weather forecast consumers, WFC.

13

type
12 WSs = WS-**set**
13 WS
14 WFCs = WFC-**set**
15 WFC
value
12 obs_part_WSs: WSS → WSs
14 obs_part_WFCs: WFCS → WFCs

3.2 Unique Identifiers 14

We shall consider only atomic parts.

16 Every single weather station has a unique identifier.

17 The weather data interpretation (i.e., the weather forecast "creator") has a unique identifier.

18 Every single weather forecast consumer has a unique identifier.

19 The global clock has a unique identifier.

15

type
16 WSI
17 WDII
18 WFCI
19 CLKI
value
16 uid_WSI: WS → WSI
17 uid_WDII: WDIS → WDII
18 uid_WFCI: WFC → WFCI
18 uid_CLKI: CLK → CLKI

3.3 Mereologies 16

We shall restrict ourselves to consider the mereologies only of the atomic parts.

20 The mereology of weather stations is the pair of the unique clock identifier and the unique identifier of the weather data interpreter.

21 The mereology of weather data interpreter is the triple of the unique clock identifier, set of unique identifiers of all the weather stations and the set of unique identifiers of all the weather forecast consumers.

22 The mereology of weather forecast consumer is the the pair of the unique clock identifier and the unique identifier of the weather data interpreter.

23 The mereology of the global clock is the triple of the set of all the unique identifiers of weather stations, the unique identifier of the weather data interpreter, and the set of all the unique identifiers of weather forecast consumers.

17

type
20 WSM = CLKI × WDII
21 WDIM = CLKI × WSI-**set** × WFCI-**set**
22 WFCM = CLKI × WDII
23 CLKM = CLKI × WDGI-**set** × WDII × WFCI-**set**
value
20 mereo_WSM: WS → WSM
21 mereo_WDI: WDI → WDIM
22 mereo_WFC: WFC → WFCM
23 mereo_CLK: CLK → CLKM

 © Dines Bjørner 2016, Fredsvej 11, DK–2840 Holte, Denmark – November 17, 2016: 14:47

3.4 Attributes 18

3.4.1 Clock, Time and Time-intervals 19

24 The global clock has an autonomous time attribute.

25 Time values are further undefined, but times are considered absolute in the sense as representing some intervals since "the birth of time", an example, concrete time could be NOVEMBER 17, 2016, 14:47 .

26 Time intervals are further undefined, but time intervals can be considered relative in the sense of representing a quantity elapsed between two times, examples are: 1 day 2 hours and 3 minutes, etc. When a time interval, ti, is specified it is always to be understood to designate the times from now, or from a specified time, t, until the time $t + ti$.

27 We postulate $\oplus$, $\ominus$, and can postulate further "arithmetic" operators, and

28 we can postulate relational operators.

20

type
24 TIME
25 TI
value
24 attr_TIME: CLK $\rightarrow$ TIME
27 $\oplus$: TIME$\times$TI$\rightarrow$TIME, TI$\times$TI$\rightarrow$TI
27 $\ominus$: TIME$\times$TI$\rightarrow$TIME, TIME$\times$TIME$\rightarrow$TI
28 $=, \neq, <, \leq, \geq, >$: TIME$\times$TIME$\rightarrow$**Bool**, TI$\times$TI$\rightarrow$**Bool**, ...

We do not here define these operations and relations.

3.4.2 Locations 21

29 Locations are metric, topological spaces and can thus be considered dense spaces of three dimensional points.

30 We can speak of one location properly contained ($\subset$) within, or contained or equal ($\subseteq$), or equal ($=$), or not equal ($\neq$) to another location.

type
29. LOC
value
30. $\subset, \subseteq, =, \neq$: LOC $\times$ LOC $\rightarrow$ **Bool**

3.4.3 Weather 22

31 The weather material is considered a dense, infinite set of weather point volumes WP. Some dense, infinite subsets (still proper volumes) of such points may be liquid, i.e., rain, water in rivers, lakes and oceans. Other dense, infinite subsets (still proper volumes) of such points may be gaseous, i.e., the air, or atmosphere. These two forms of proper volumes “border” along infinite subsets (curved planes, surfaces) of weather points.

32 From the material weather one can observe its location.

type
31 W = WP-**infset**
31 WP
value
32 attr_LOC: W → LOC

23

33 Some meteorological quantities are:

a. *Hu*midity,
b. *Te*mperature,
c. *Wi*nd and
d. *Ba*rometric pressure.

34 The weather has an indefinite number of attributes at any one time.

a. Humidity distribution, at level (above sea) and by location,
b. Temperature distribution, at level (above sea) and by location,
c. Wind direction, velocity and mobility of wind center, and by location,
d. Barometric pressure, and by location,
e. etc., etc.

type
33a. Hu
33b. Te
33c. Wi
33d. Ba
34a. HDL = LOC $\underset{m}{\rightarrow}$ Hu
34b. TDL = LOC $\underset{m}{\rightarrow}$ Te
34c. WDL = LOC $\underset{m}{\rightarrow}$ Wi
34d. BPL = LOC $\underset{m}{\rightarrow}$ Ba
34e. ...
value
34a. attr_HDL: W → HDL

 © Dines Bjørner 2016, Fredsvej 11, DK–2840 Holte, Denmark – November 17, 2016: 14:47

34b. attr_TDL: W → TDL
34c. attr_WDL: W → WDL
34d. attr_APL: W → BPL
34e. ...

3.4.4 Weather Stations 24

35 Weather stations have static location attributes.

36 Weather stations sample the weather gathering humidity, temperature, wind, barometric pressure, and possibly other data, into time and location stamped weather data.

value
35 attr_LOC: WS → LOC
type
36 WD :: mkWD((TIME×LOC)×(TDL×HDL×WDL×BPL×...))

3.4.5 Weather Data Interpreter 25

37 There is a programmable attribute: weather data repository, wdr:WDR, of weather data, wd:WD, collected from weather stations.

38 And there is programmable attribute: weather forecast repository, wfr:WFR, of forecasts, wf:WF, disseminate-able to weather forecast consumers.

These repositories are updated when

39 received from the weather stations, respectively when

40 calculated by the weather data interpreter.

type
37 WDR
38 WFR
value
39 update_wdr: TIME × WD → WDR → WDR
40 update_wfr: TIME × WF → WFR → WFR

It is a standard exercise to define these two functions (say algebraically).

3.4.6 Weather Forecasts **26**

41 Weather forecasts are weather forecast format-, time- and location-stamped quantities, the latter referred to as wefo:WeFo.

42 There are a definite number ($n \geq 1$) of weather forecast formats.

43 We do not presently define these various weather forecast formats.

44 They are here thought of as being requested, mkWFReq, by weather forecast consumers.

type
41 WF = WFF × (TIME×TI) × LOC × WeFo
42 WFF = WFF1 | WFF2 | ... | WFFn
43 WFF1, WFF2, ..., WFFn
44 WFReq :: mkWFReq(s_wff:WFF,s_ti:(TIME×TI),s_loc:LOC)

3.4.7 Weather Forecast Consumer **27**

45 There is a programmable attribute, d:D, D for display (!).

46 Displays can be rendered (RND): visualized, tabularised, made audible, translated (between languages and language dialects, ...), etc.

47 A rendered display can be "abstracted back" into its basic form.

48 Any abstracted rendered display is identical to its abstracted form.

type
45 D
46 RND
value
45 attr_D: WFC → D
46 rndr_D: RND × D → D
47 abs_D: D → D
axiom
48 ∀ d:D, r:RND • abs_D(rndr(r,d)) = d

4 Perdurants **28**

4.1 A WIS Context

49 We postulate a given system, wis:WIS.

That system is characterized by

50 a dynamic weather

51 and its unique identifier,

52 a set of weather stations

53 and their unique identifiers,

© Dines Bjørner 2016, Fredsvej 11, DK–2840 Holte, Denmark – November 17, 2016: 14:47

54 a single weather data interpreter

55 and its unique identifier,

56 a set of weather forecast consumers

57 and their unique identifiers, and

58 a single clock

59 and its unique identifier.

60 Given any specific wis:WIS there is [therefore] a full set of part identifiers, is, of weather, clock, all weather stations, the weather data interpreter and all weather forecast consumers.

29

We list the above-mentioned values. They will be referenced by the channel declarations and the behaviour definitions of this section.

value
49 wis:WIS
50 w:W = obs_material_W(wis)
51 wi:WI = uid_WI(w)
52 wss:WSs = obs_part_WSs(obs_part_WSS(wis))
53 wsis:WDGI-**set** = {uid_WSI(ws)|ws:WS•ws ∈ wss}
54 wdi:WDI = obs_part_WDIS(wis)
55 wdii:WDII = uid_WDII(wdi)
56 wfcs:WFCs = obs_part_WFCs(obs_part_WFCS(wis))
57 wfcis:WFI-**set** = {uid_WFCI(wfc)|wfc:WFC•wfc ∈ wfcs}
58 clk:CLK = obs_part_CLK(wis)
59 clki:CLKI = uid_CLKI(clk)
60 is:(WI|WSI|WDII|WFCI)-**set** = {wi}∪wsis∪{wdii}∪wfcis

4.2 Channels

30

61 Weather stations share weather data, WD, with the weather data interpreter — so there is a set of channels, one each, “connecting” weather stations to the weather data interpreter.

62 The weather data interpreter shares weather forecast requests, WFReq, and interpreted weather data (i.e., forecasts), WF, with each and every forecast consumer — so there is a set of channels, one each, “connecting” the weather data interpreter to the interpreted weather data (i.e., forecast) consumers.

63 The clock offers its current time value to each and every part, except the weather, of the WIS system.

31

channel
61 { ch_si[wsi,wdii]:WD | wsi:WSI•wsi ∈ wsis }
62 { ch_ic[wdii,fci]:(WFReq|WF) | fci:FCI•fci ∈ fcis }

```
63  { ch_cp[clki,i]:TIME | i:(WI|CLKI|WSI|WDII|WFCI)•i ∈ is }
```

4.3 WIS Behaviours **32**

64 WIS behaviour, wis_beh, is the

65 parallel composition of all the weather station behaviours, in parallel with the

66 weather data interpreter behaviour, in parallel with the

67 parallel composition of all the weather forecast consumer behaviours, in parallel with the

68 clock behaviour.

value

```
64  wis_beh: Unit → Unit
64  wis_beh() ≡
65    ‖ { ws_beh(uid_WSI(ws),mereo_WS(ws),...) | ws:WS•ws ∈ wss } ‖
66    ‖ wdi_beh(uid_WDI(wdi),mereo_WDI(wdi),...)(wd_rep,wf_rep) ‖
67    ‖ { wfc_beh(uid_WFCI(wfc),mereo_WDG(wfc),...) | wfc:WFC•wfc ∈ wfcs } ‖
68      clk_beh(uid_CLKI(clk),mereo_CLK(clk),...)("November 17, 2016, 14:47")
```

4.4 Clock **33**

69 The clock behaviour has a programmable attribute, t.

70 It repeatedly offers its current time to any part of the WIS system.

It nondeterministically internally "cycles" between

71 retaining its current time, or

72 increment that time with a "small" time interval, δ, or

73 offering the current time to a requesting part.

value

```
69.  clk_beh: clki:CLKI × clkm:CLKM → TIME →
70.      out {ch_cp[clki,i]|i:(WSI|WDII|WFCI)•i ∈wsis∪{wdii}∪wfcis }  Unit
69.  clk_beh(clki,is)(t) ≡
71.        clk_beh(clki,is)(t)
72.      ⌈⌉ clk_beh(clki,is)(t⊕δ)
73.      ⌈⌉ ( ⌈⌉⌊⌋{ ch_cp[clki,i] ! t | i:(WSI|WDII|WFCI)•i ∈ is } ; clk_beh(clki,is)(t) )
```

 © Dines Bjørner 2016, Fredsvej 11, DK–2840 Holte, Denmark – November 17, 2016: 14:47

4.5 Weather Station 34

74 The weather station behaviour communicates with the global clock and the weather data interpreter.

75 The weather station behaviour simply "cycles" between sampling the weather, reporting its findings to the weather data interpreter and resume being that overall behaviour.

76 The weather station time-stamp "sample' the weather (i.e., meteorological information).

77 The meteorological information obtained is analysed with respect to temperature (distribution etc.),

78 humidity (distribution etc.),

79 wind (distribution etc.),

80 barometric pressure (distribution etc.), etcetera,

81 and this is time-stamp and location aggregated (mkWD) and "sent" to the (central ?) weather data interpreter,

82 whereupon the weather data generator behaviour resumes.

35

```
value
74  ws_beh: wsi:WSI × (clki,wi,wdii):WDGM × (LOC × ...) →
74      in ch_cp[ clki,wsi ] out ch_gi[ wsi,wdii ]  Unit
75  ws_beh(wsi,(clki,wi,wdii),(loc,...)) ≡
77      let tdl = attr_TDL(w),
78          hdl = attr_HDL(w),
79          wdl = attr_WDL(w),
80          bpl = attr_BPL(w), ... in
81      ch_gi[ wsi,wdii ] ! mkWD((ch_cp[ clki,wsi ] ?,loc),(tdl,hdl,wdl,bpl,...)) end ;
82      wdg_beh(wsi,(clki,wi,wdii),(loc,...))
```

4.6 Weather Data Interpreter 36

83 The weather data interpreter behaviour communicates with the global clock, all the weather stations and all the weather forecast consumers.

84 The weather data interpreter behaviour non-deterministically internally (⌈⌉) chooses to

85 either collect weather data,

86 or calculate some weather forecast,

87 or disseminate a weather forecast.

37

value

```
83  wdi_beh: wdii:WDII×(clki,wsis,wfcis):WDIM×...→(WD_Rep×WF_Rep)→
83          in ch_cp[clki,wdii], { ch_si[wsi,wdii] | wsi:WSI•wsi ∈ wsis },
83          out { ch_ic[wdii,wfci] | wfci:WFCI•wfci ∈ wfcis }  Unit
83  wdi_beh(wdii,(clki,wsis,wfcis),...)(wd_rep,wf_rep) ≡
85      collect_wd(wdii,(clki,wsis,wfcis),...)(wd_rep,wf_rep)
84      ⌈⌉
86      calculate_wf(wdii,(clki,wsis,wfcis),...)(wd_rep,wf_rep)
84      ⌈⌉
87      disseminate_wf(wdii,(clki,wsis,wfcis),...)(wd_rep,wf_rep)
```

4.6.1 collect_wd 38

88 The collect weather data behaviour communicates with the global clock and all the weather stations – but "passes-on" the capability to communicate with all of the weather forecast consumers.

89 The collect weather data behaviour

90 non-deterministically externally offers to accept weather data from some weather station,

91 updates the weather data repository with a time-stamped version of that weather data,

92 and resumes being a weather data interpreter behaviour, now with an updated weather data repository.

39

value

```
88  collect_wd: wdii:WDII×(clki,wsis,wfcis):WDIM×...
88          → (WD_Rep×WF_Rep) →
88          in ch_cp[clki,wdii], { ch_si[wsi,wdii] | wsi:WSI•wsi ∈ wsis },
88          out { ch_ic[wdii,wfci] | wfci:WFCI•wfci ∈ wfcis }  Unit
89  collect_wd(wdii,(clki,wsis,wfcis),...)(wd_rep,wf_rep) ≡
90     let ((ti,loc),(hdl,tdl,wdl,bpl,...)) = ⌈⌉{wsi[wsi,wdii]?|wsi:WSI•wsi∈wsis} in
91     let wd_rep′ = update_wdr(ch_cp[clki,wdii]?,((ti,loc),(hdl,tdl,wdl,bpl,...)))(wd_rep) in
92     wdi_beh(wdii,(clki,wsis,wfcis),...)(wd_rep′,wf_rep) end end
```

 © Dines Bjørner 2016, Fredsvej 11, DK–2840 Holte, Denmark – November 17, 2016: 14:47

4.6.2 calculate_wf 40

93 The calculate forecast behaviour communicates with the global clock – but "passes-on" the capability to communicate with all of weather stations and the weather forecast consumers.

94 The calculate forecast behaviour

95 non-deterministically internally chooses a forecast type from among a indefinite set of such,

96 and a current or "future" time-interval,

97 whereupon it calculates the weather forecast and updates the weather forecast repository,

98 and then resumes being a weather data interpreter behaviour now with the weather forecast repository updated with the calculated forecast.

41

```
value
93  calculate_wf: wdii:WDII×(clki,wsis,wfcis):WDIM×...→(WD_Rep×WF_Rep)→
93          in ch_cp[ clki,wdii ], { ch_si[ wsi,wdii ] | wsi:WSI•wsi ∈ wsis },
93          out { ch_ic[ wdii,wfci ] | wfci:WFCI•wfci ∈ wfcis }  Unit
94  calculate_wf(wdii,(clki,wsis,wfcis),...)(wd_rep,wf_rep) ≡
95      let tf:WWF = ft1 ⌈⌉ ft2 ⌈⌉ ... ⌈⌉ ftn,
96          ti:(TIME×TIVAL) • toti≥ch_cp[ clki,wdii ] ? in
97      let wf_rep′ = update_wfr(calc_wf(tf,ti)(wf_rep)) in
98      wdi_beh(wdii,(clki,wsis,wfcis),...)(wd_rep,wf_rep′) end end
```

42

99 The calculate_weather forecast function is, at present, further undefined.

```
value
99.  calc_wf: WFF × (TIME×TI) → WFRep → WF
99.  calc_wf(tf,ti)(wf_rep) ≡ ,,,
```

4.6.3 disseminate_wf 43

100 The disseminate weather forecast behaviour communicates with the global clock and all the weather forecast consumers – but "passes-on" the capability to communicate with all of weather stations.

101 The disseminate weather forecast behaviour non-deterministically externally offers to received a weather forecast request from any of the weather forecast consumers, wfci, that request is for a specific format forecast, tf, and either for a specific time or for a time-interval, toti, as well as for a specific location, loc.

102 The disseminate weather forecast behaviour retrieves an appropriate forecast and

103 sends it to the requesting consumer –

104 whereupon the disseminate weather forecast behaviour resumes being a weather data interpreter behaviour

44

```
value
100 disseminate_wf: wdii:WDII×(clki,wsis,wfcis):WDIM×...→(WD_Rep×WF_Rep)→
100         in ch_cp[clki,wdii] in,out { ch_ic[wdii,wfci] | wfci:WFCI•wfci ∈ wfcis } Unit
100 disseminate_wf(wdii,(clki,wsis,wfcis),...)(wd_rep,wf_rep) ≡
101     let mkReqWF((tf,toti,loc),wfci) = ⌈⌉{ch_ic[wdii,wfci] ? | wfci:WFCI•wfci∈wfcis} in
102     let wf = retr_WF((tf,toti,loc),wf_rep) in
103     ch_ic[wdii,wfci] ! wf ;
104     disseminate_wf(wdii,(clki,wsis,wfcis),...)(wd_rep,wf_rep) end end
```

45

105 The retr_WF((tf,toti,loc),wf_rep) function invocation retrieves the weather forecast from the weather forecast repository most "closely" matching the format, tf, time, toti, and location of the request received from the weather forecast consumer. We do not define this function.

```
105. retr_WF: (WFF×(TIME×TI)×LOC) × WFRep → WF
105. retr_WF((tf,toti,loc),wf_rep) ≡ ...
```

We could have included, in our model, the time-stamping of receipt (formula Item 101) of requests, and the time-stamping of delivery of requested forecast in which case we would insert ch_cp[clki,wdii]? at respective points in formula Items 101 and 103.

4.7 Weather Forecast Consumer 46

106 The weather forecast consumer communicates with the global clock and the weather data interpreter.

107 The weather forecast consumer behaviour

108 nondeterministically internally either

109 selects a suitable weather cast format, tf,

© Dines Bjørner 2016, Fredsvej 11, DK–2840 Holte, Denmark – November 17, 2016: 14:47

110 selects a suitable location, loc′, and

111 selects, toti, a suitable time (past, present or future) or a time interval (that is supposed to start when forecast request is received by the weather data interpreter.

112 With a suitable formatting of this triple, mkReqWF(tf,loc′,toti), the weather forecast consumer behaviour "outputs" a request for a forecast to the weather data interpreter (first "half" of formula Item 111) whereupon it awaits (;) its response (last "half" of
47 formula Item 111) which is a weather forecast, wf,

113 whereupon the weather forecast consumer behaviour resumes being that behaviour with it programmable attribute, d, being replaced by the received forecast suitably annotated;

108 or the weather forecast consumer behaviour

114 edits a display

115 and resumes being a weather forecast consumer behaviour with the edited programmable attribute, d′.

48

```
value
106  wfc_beh: wfci:WFCI × (clki,wdii):WFCM × (LOC × ...) → D →
106        in  ch_cp[ clki,wfci ],
106        in,out { ch_ic[ wdii,wfci ] | wfci:WFCI•wfci ∈ wfcis }  Unit
107  wfc_beh(wfci,(clki,wdii),(loc,...))(d) ≡
109       let tf = tf1 ⌈⌉ tf2 ⌈⌉ ... ⌈⌉ tfn,
110            loc′:LOC • loc′=loc∨loc′≠loc,
111            (t,ti):(TIME×TI) • ti≥0 in
112       let wf = (ch_ic[ wdii,wfci ] ! mkReqWF(tf,loc′,(t,ti))) ; ch_ic[ wdii,wfci ] ? in
113       wfc_beh(wfci,(clki,wdii),(loc,...))((tf,loc′,(t,ti)),wf) end end
108     ⌈⌉
114       let d′:D {\EQ} rndr\_D(d,{\DOTDOTDOT}) in
115       wfc_beh(wfci,(clki,wdii),(loc,...))(d′) end
```

The choice of location may be that of the weather forecast consumer location, or it may be one different from that. The choice of time and time-interval is likewise a non-deterministic internal choice.

5 Conclusion 49

5.1 Reference to Similar Work

As far as I know there are no published literature nor, to our knowledge, institutional or private works on the subject of modelling weather data collection, interpretaion and weather forecast delivery systems.

5.2 What Have We Achieved ? 50

TO BE WRITTEN

5.3 What Needs to be Done Next ? 51

TO BE WRITTEN

5.4 Acknowledgements 52

This technical cum experimental research report was begun in Bergen, Wednesday, November 9, 2016 – inspired by a presentation by Ms. Doreen Tuheirwe, Makarere University, Kampala, Uganda. I thank her, and Profs. Magne Haveraaen and Jaakko Järvi of BLDL: the Bergen Language Design Laboratory, Dept. of Informatics, University of Bergen (Norway), for their early comments, and Prof. Haveraaen for inviting me to give PhD lectures there in the week of Nov. 6–12, 2016.

5.5 References

[Bjø10a] Dines Bjørner. On Development of Web-based Software: A Divertimento of Ideas and Suggestions. Technical, Technical University of Vienna, August–October 2010. http://www.imm.dtu.dk/~dibj/wfdftp.pdf.

[Bjø10b] Dines Bjørner. The Tokyo Stock Exchange Trading Rules. R&D Experiment, Fredsvej 11, DK-2840 Holte, Denmark, January and February, 2010. Version 1, 78 pages: many auxiliary appendices, Version 2, 23 pages: omits many appendices and corrects some errors..

[Bjø13a] Dines Bjørner. Documents – a Domain Description[1]. Experimental Research Report 2013-3, DTU Compute and Fredsvej 11, DK-2840 Holte, Denmark, Spring 2013.

[Bjø13b] Dines Bjørner. Pipelines – a Domain Description[2]. Experimental Research Report 2013-2, DTU Compute and Fredsvej 11, DK-2840 Holte, Denmark, Spring 2013.

[Bjø13c] Dines Bjørner. Road Transportation – a Domain Description[3]. Experimental Research Report 2013-4, DTU Compute and Fredsvej 11, DK-2840 Holte, Denmark, Spring 2013.

[Bjø16a] Dines Bjørner. A Credit Card System: Uppsala Draft. Technical Report: Experimental Research, Fredsvej 11, DK–2840 Holte, Denmark, November 2016. http://www.imm.dtu.dk/~dibj/2016/credit/accs.pdf.

[1] http://www.imm.dtu.dk/~dibj/doc-p.pdf
[2] http://www.imm.dtu.dk/~dibj/pipe-p.pdf
[3] http://www.imm.dtu.dk/~dibj/road-p.pdf

© Dines Bjørner 2016, Fredsvej 11, DK–2840 Holte, Denmark – November 17, 2016: 14:47

[Bjø16b] Dines Bjørner. Domains: Their Simulation, Monitoring and Control – A Divertimento of Ideas and Suggestions. *Perhaps to be submitted for consideration by Formal Aspects of Computing*, 2016. http://www.imm.dtu.dk/˜dibj/2016/demos/faoc-demo.pdf.

[Bjø16c] Dines Bjørner. Manifest Domains: Analysis & Description. *Formal Aspects of Computing*, ...(...):1–51, 2016. DOI 10.1007/s00165-016-0385-z http://link.springer.com/article/10.1007/s00165-016-0385-z.

[Bjø16d] Dines Bjørner. Weather Information Systems: Towards a Domain Description. Technical Report: Experimental Research, Fredsvej 11, DK–2840 Holte, Denmark, November 2016. http://www.imm.dtu.dk/ dibj/2016/wis/wis-p.pdf.