Managing and Using Context Information within the PerLa Language

Fabio A. Schreiber -

Letizia Tanca – Romolo Camplani – Diego Viganò

Politecnico di Milano
Dipartimento di Elettronica ed Informazione

PeDiGREE
PervasiveDataGRoup of EnginEers

PerLa
PERvasive LAnguage

Outline

- Introduction
- The CDT context model
- Context Management in PerLa
  - Language support
  - Contextual-block composition
- Examples
- Conclusions

Autonomic Pervasive Systems

- □ Pervasive systems are widely adopted to monitor many kinds of physical phenomena.

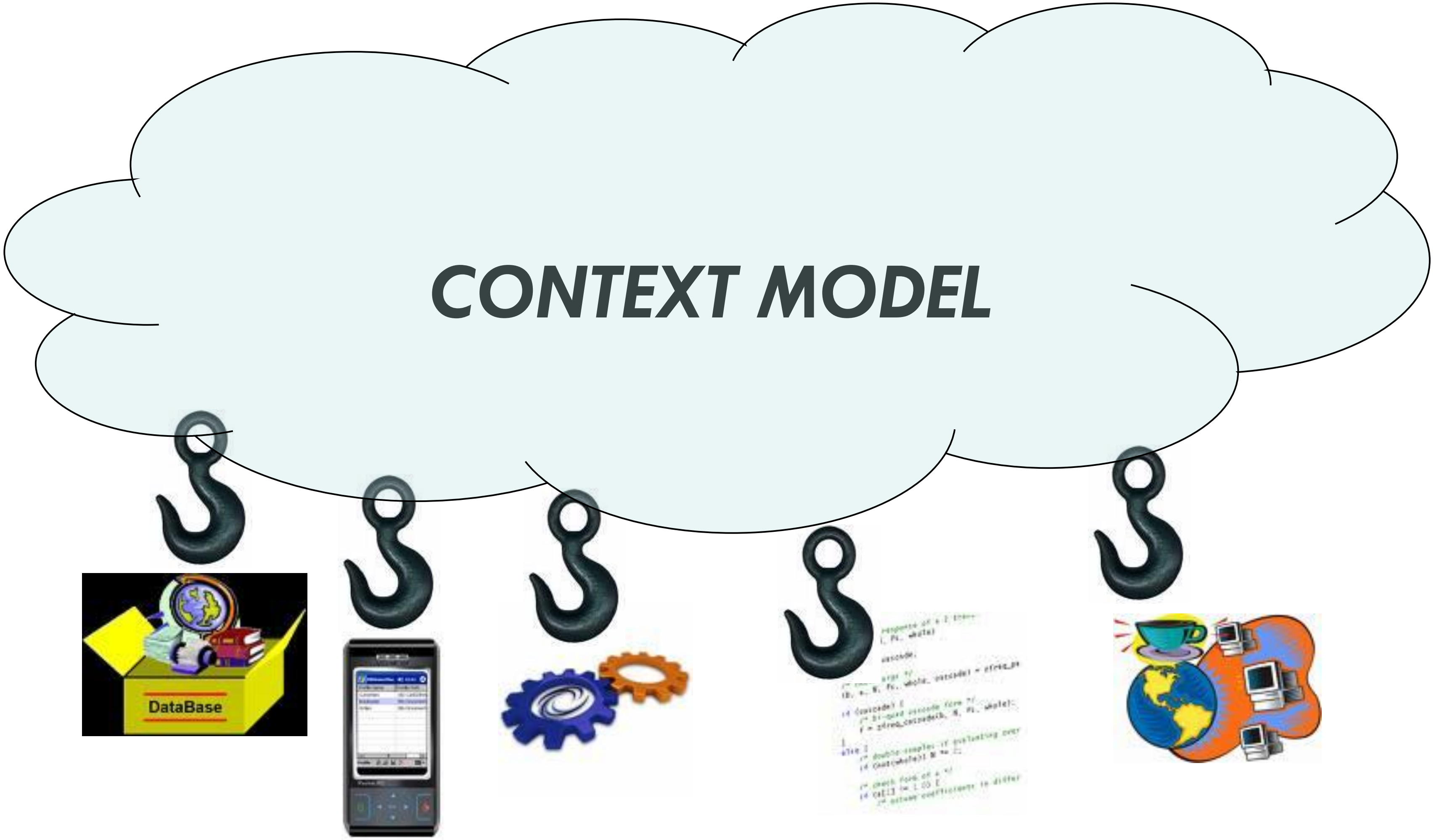
- □ **Context-awareness** plays a fundamental role since it allows, through the *perception* of the environment, to make the system *autonomic* w.r.t. environmental situations and changes.

- □ Context must be managed both at **design** and **run** time.

Context management at ***design time***

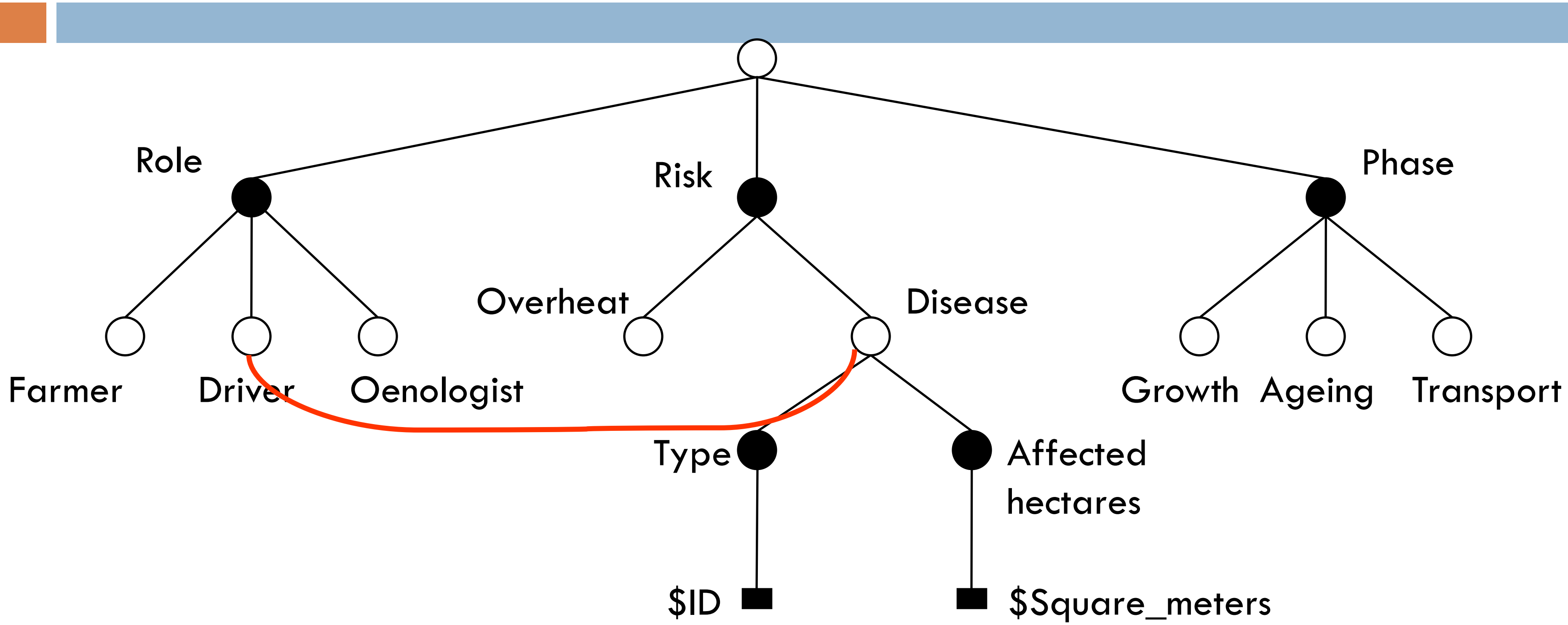
- Context modelling
- Application domain modelling (data, functions)
- Design of the relationship between the context model and the application domain.

Context management at ***design time***



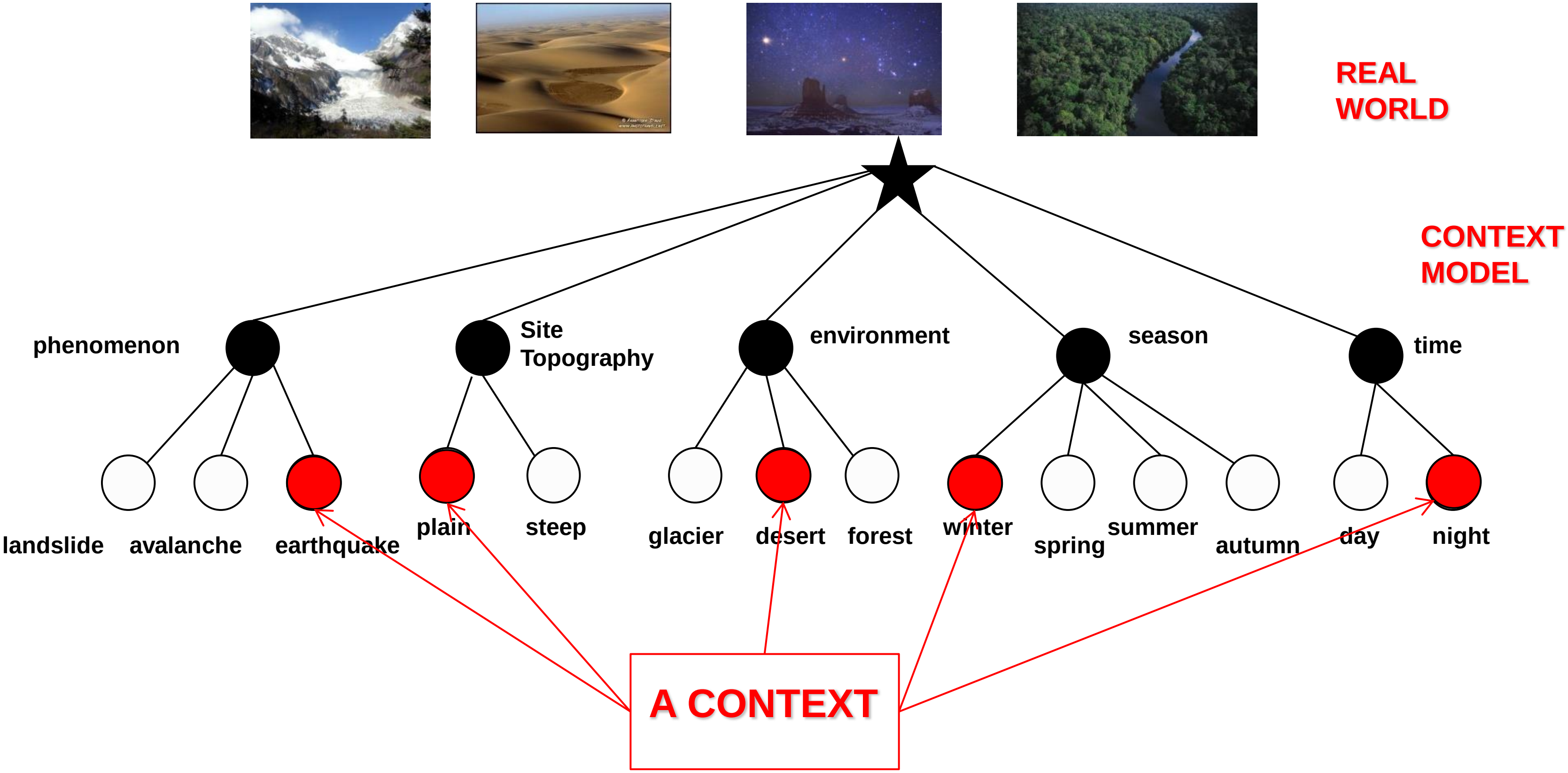
CONTEXT MODEL

CDT model



- □ The environment is modelled in terms of **dimension nodes, concept (or value) nodes, attributes, and possibly constraints**
- □ A **context element** is defined as *Dimension = Value* and a context is a conjunction of context elements
- □ A context can be represented as a particular subtree of the CDT

CDT model



REAL WORLD

CONTEXT MODEL

phenomenon

Site Topography

environment

season

time

landslide

avalanche

earthquake

plain

steep

glacier

desert

forest

winter

spring

summer

autumn

day

night

A CONTEXT

Context management at ***design time***

It is important to separate between

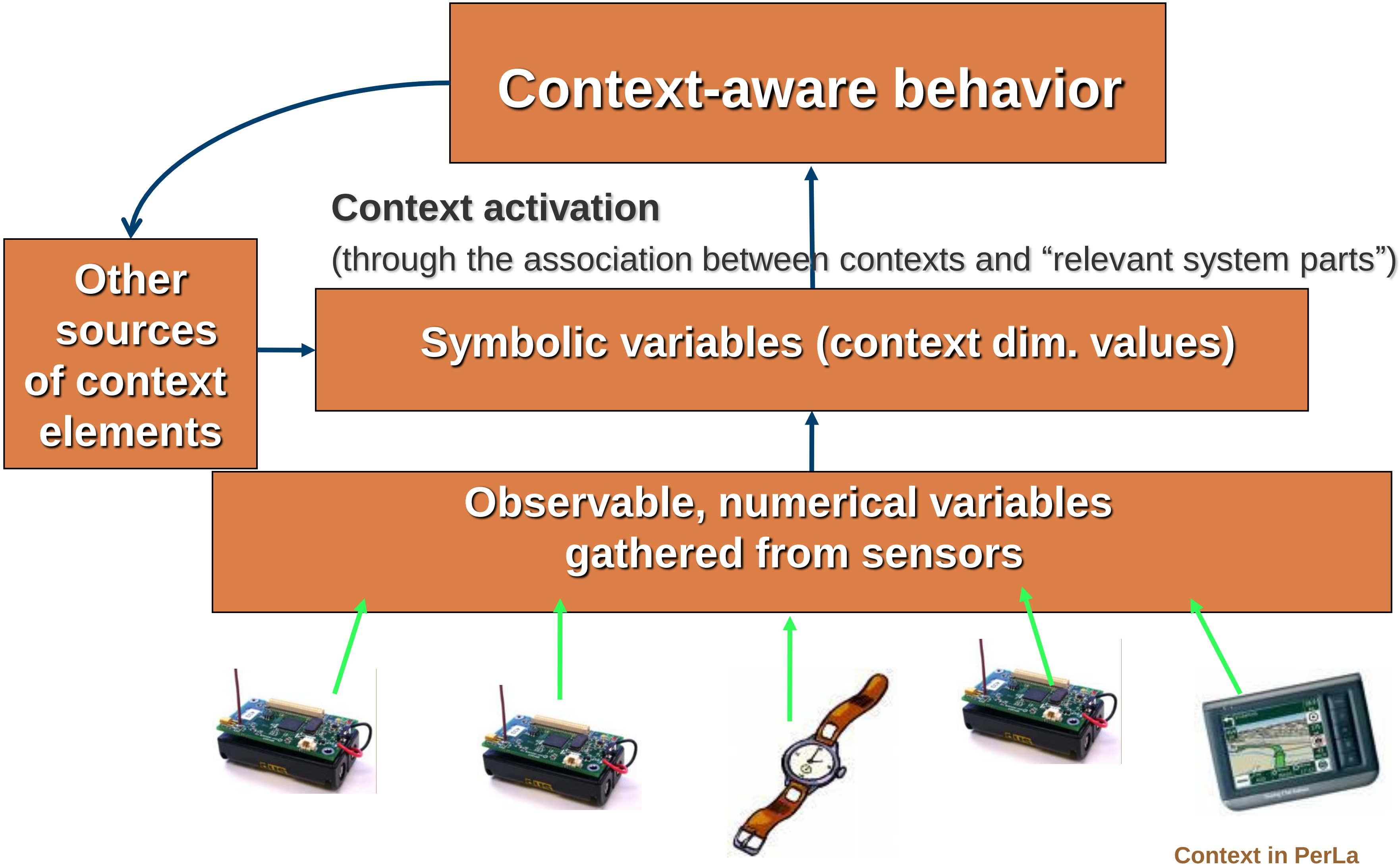
- **NUMERIC OBSERVABLES**
- **SYMBOLIC OBSERVABLES**

Example:

temperatures

- 20 ° C – 10 ° C	**COLD**
11 ° C – 25 ° C	**MILD**
26 ° C – 50 ° C	**WARM**

Context management at ***runtime***



Context-aware behavior

Context activation

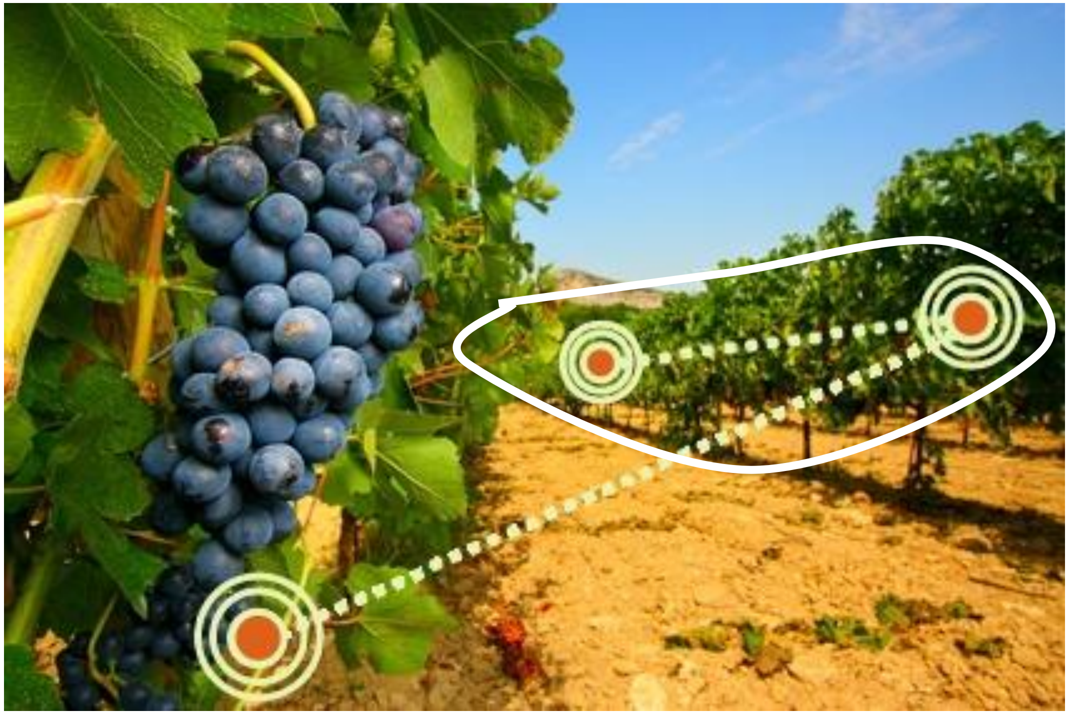
(through the association between contexts and “relevant system parts”)

Other sources of context elements

Symbolic variables (context dim. values)

Observable, numerical variables gathered from sensors

Context management at ***runtime***



Apply the sensor query only to the sensors in context:

phase = ‘growth' AND risk='overheat‘ AND orientation=‘westward’

Context in PerLa

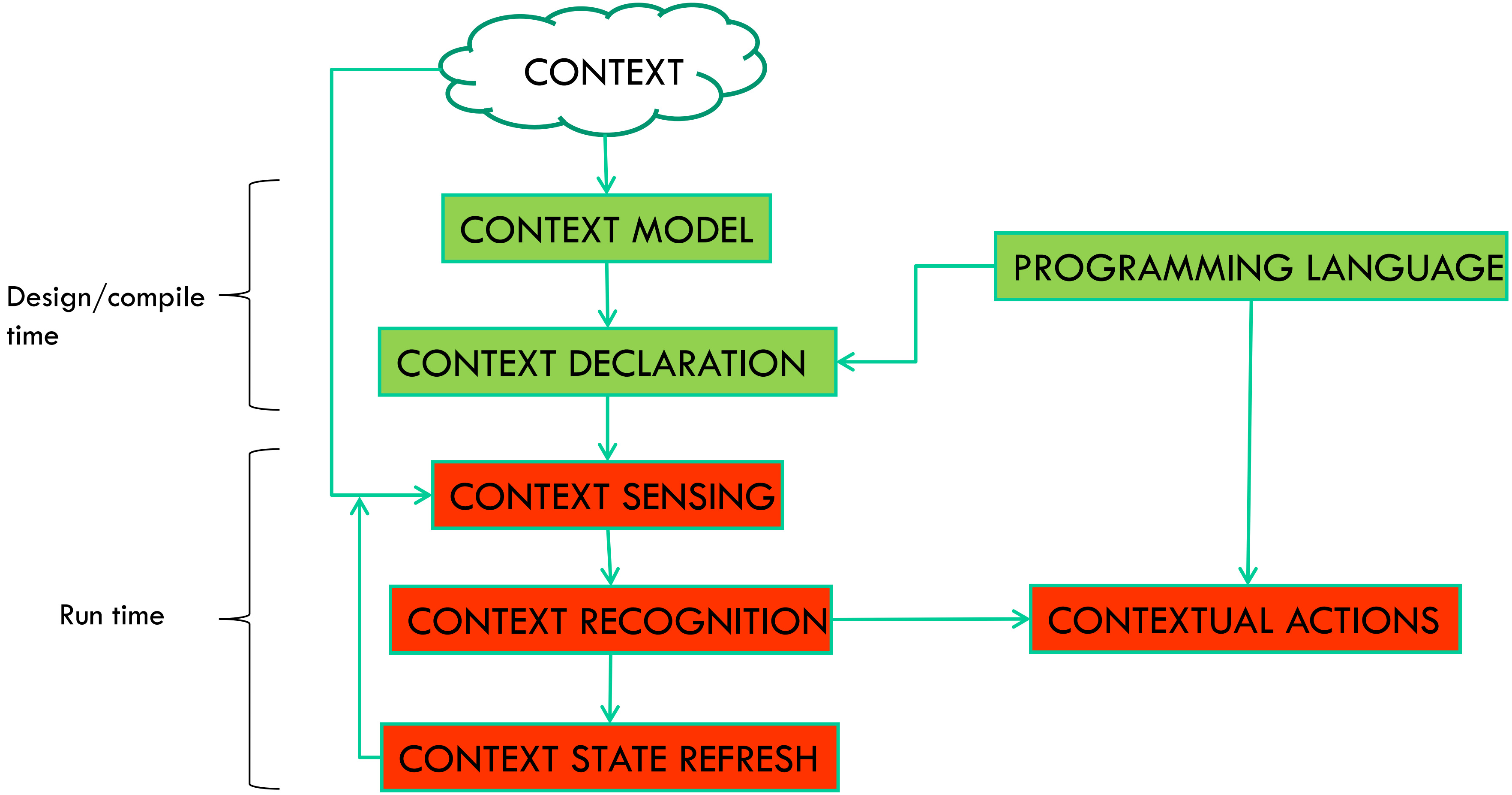
Context management at ***runtime***

- Context sensing (numeric observables)
- Context recognition (symbolic observables)
- Context activation
- Context-aware behaviour to be merged into a middleware and a language to manage pervasive systems hiding the complexity of handling different technologies

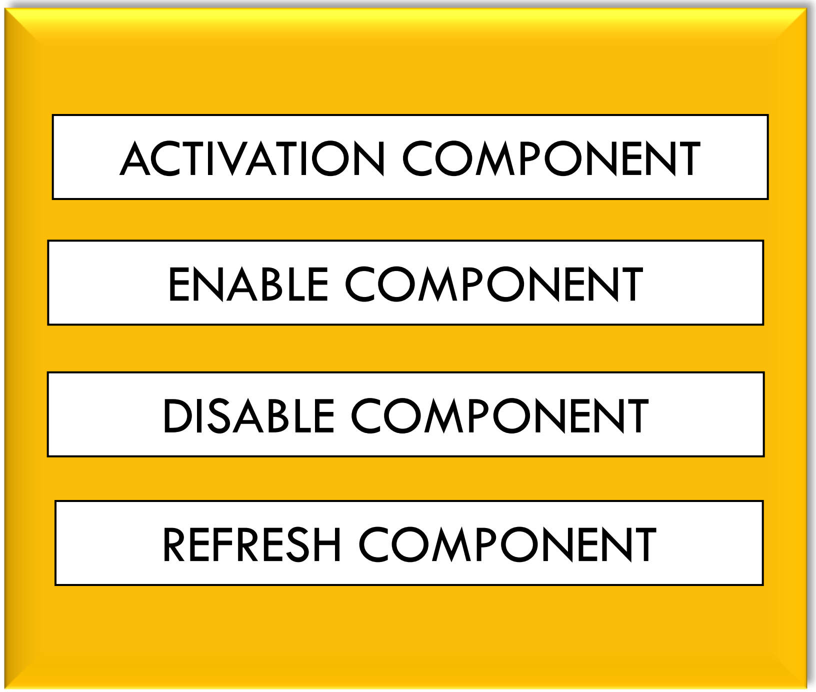
```
SELECT temperature, humidity
    WHERE temp>20
    SAMPLING EVERY 1h
    EXECUTE IF device_id > 2
```

PerLa
PERvasive LAnguage

Context-aware Sw Behaviour



Contextual block structure



ACTIVATION COMPONENT

ENABLE COMPONENT

DISABLE COMPONENT

REFRESH COMPONENT

The PerLa Context Language

```
CREATE DIMENSION <Dimension Name>
[CHILD OF <Parent Node >]
[CREATE ATTRIBUTE $<Attribute Name >] |
{CREATE  CONCEPT <Concept Name> WHEN <Condition >
[EXCLUDES <Dimension Name>.<Concept Name>]
[CREATE ATTRIBUTE $<Attribute Name >]*
[EVALUATED ON <Low Level Query >]}*
```

CDT **declaration** in terms of numeric and symbolic observables

Conversion from numeric to symbolic observable

PerLa Definition of contexts and action(s) to be performed

```
CREATE CONTEXT <Context Name>
ACTIVE IF <Dimension>= <Value>
[AND <Dimension>= <Value >]*
ON ENABLE  (<Context>): <PerLa Query>
ON DISABLE  (<Context>): <PerLa Query>
REFRESH EVERY <Period>
```

Contextual Block

Activation component

Enable component

Disable component

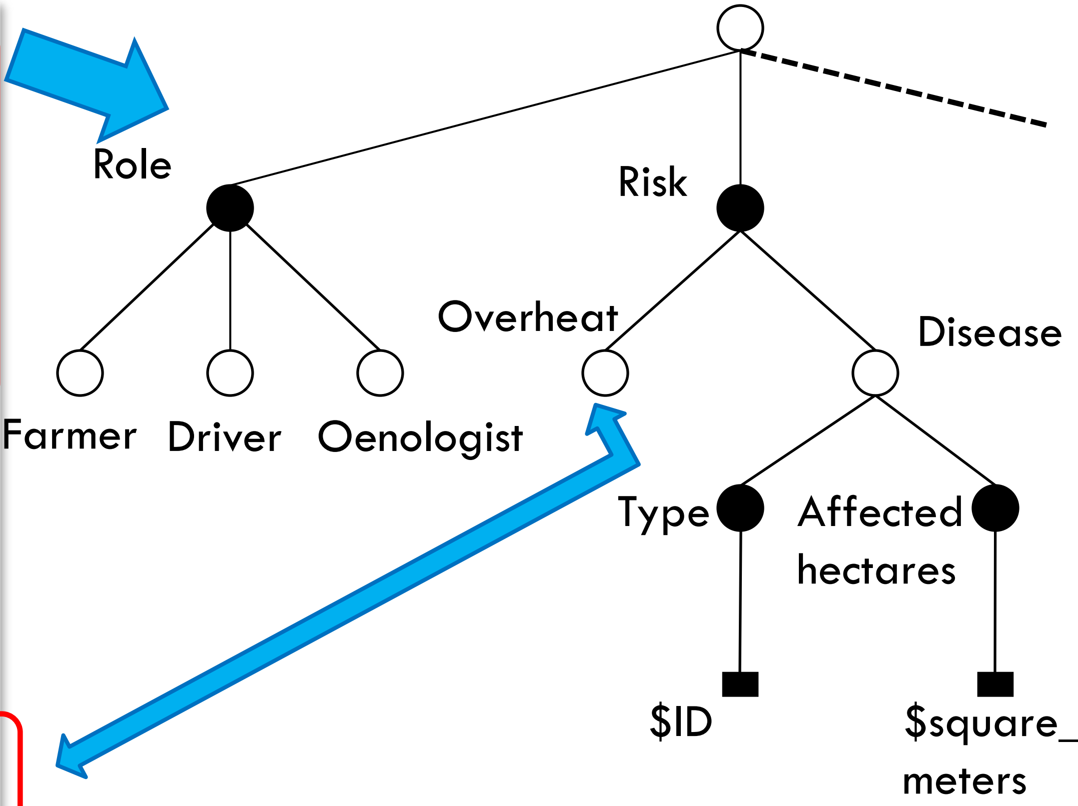
Refresh component

The PerLa Context language (2/3)

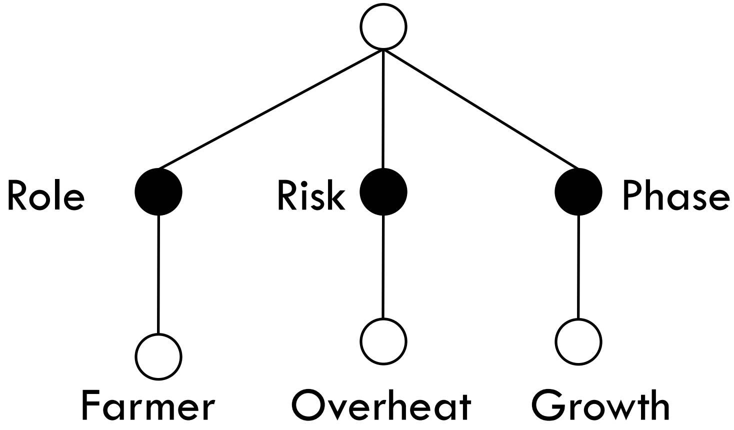
Example: Given the previous CDT

```
CREATE DIMENSION Role
  CREATE CONCEPT Farmer
    WHEN get_user_role()='farmer'
  CREATE CONCEPT Oenologist
    WHEN get_user_role()='oenologist'
  CREATE CONCEPT Driver
    WHEN get_user_role()='driver'

CREATE DIMENSION Risk
  CREATE CONCEPT Disease
    WHEN get_interest_topic()='disease'
  CREATE CONCEPT Overheat
    WHEN temperature > 30 AND
          brightness > 0.75;
.....
```

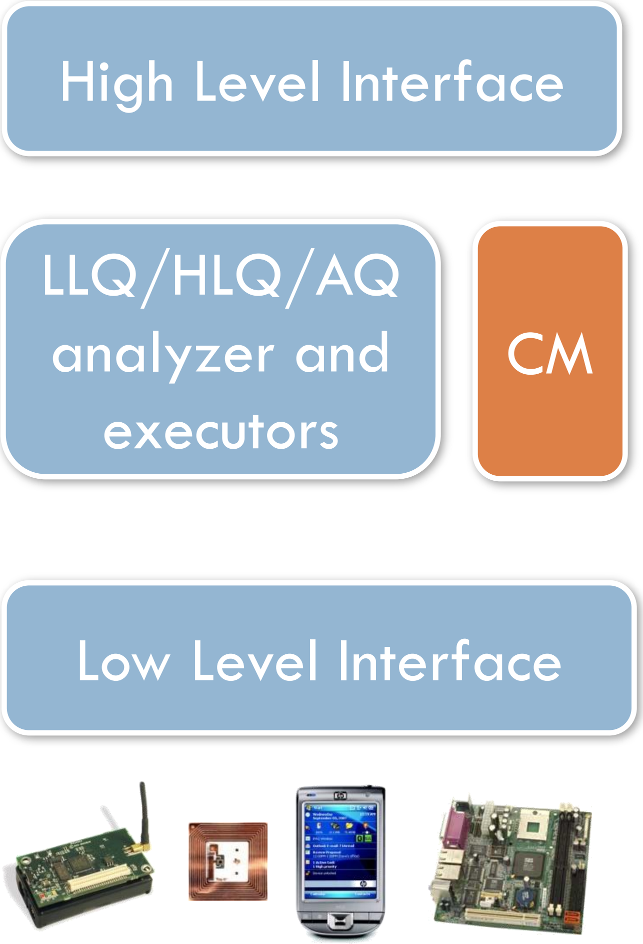


The PerLa Context Language (3/3)



```
CREATE CONTEXT Growth_Monitoring
ACTIVE IF phase = ‘growth' AND role=‘farmer’  AND Risk='overheat'
ON ENABLE:
 SELECT temperature,humidity
 SAMPLING EVERY 120 s
 EXECUTE IF location = ‘vineyard'
 SET PARAMETER 'alarm' = TRUE;
ON DISABLE:
SET PARAMETER 'alarm' = FALSE;
REFRESH EVERY 24 h;
```

PerLa language and middleware



High Level Interface

LLQ/HLQ/AQ analyzer and executors

CM

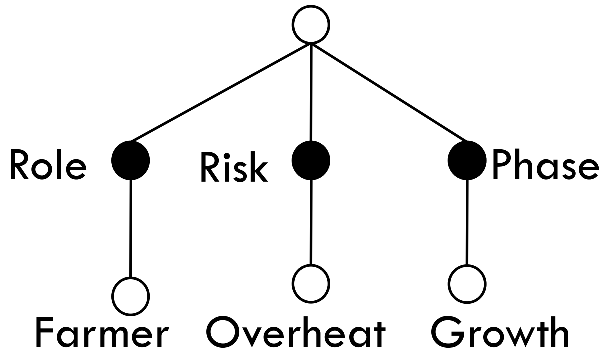
Low Level Interface

- □ **CDT creation and maintenance**
- □ **Context detection**
- □ **Perform context actions**

The CM associates to every dimension of the CDT a table that contains the values of every *numeric* observable sampled from the environment and that is used in relation with the *symbolic* observables which describe that dimension

PerLa language and middleware

In the previous example we declared a context that includes the *observable* “overheat” (declared using the *numeric* **temperature** and **brightness**):



OVERHEAT:
temperature > 30 AND brightness > 0.75

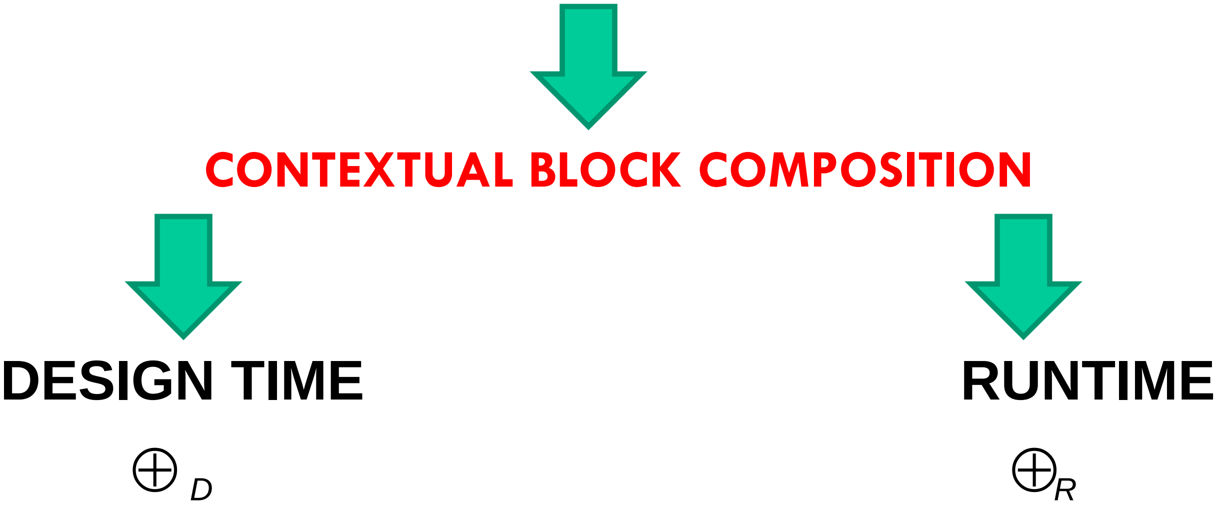
ID	Temperature	Brightness
1	28	0.60
3	31	0.71
4	33	0.80

The context can be considered as **active** for all the sensors for which the rule is **true,** and the context-aware actions will be performed only on them.

Contextual block automatic composition

- **Problem:** given a CDT, the number of possible contexts exponentially grows with the dimensions number ☹
- E. g.: 5 dimensions, 3 conceps/dim (average) → >500 contexts!

- **Solution:** automatic composition of the contextual block, based on partial components:



CONTEXTUAL BLOCK COMPOSITION

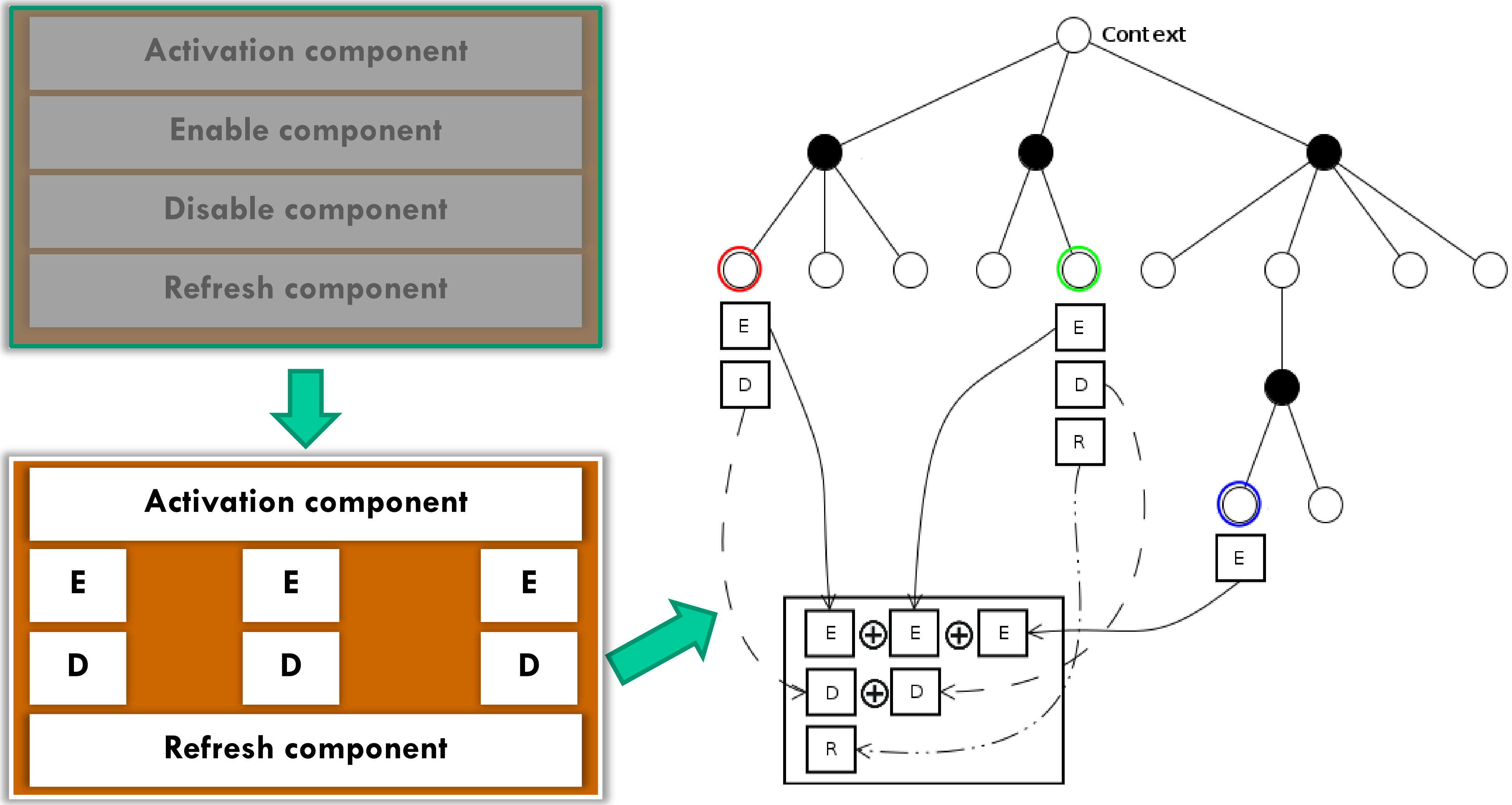
DESIGN TIME

$\oplus_D$

RUNTIME

$\oplus_R$

Contextual block automatic composition



Activation component

Enable component

Disable component

Refresh component

Activation component

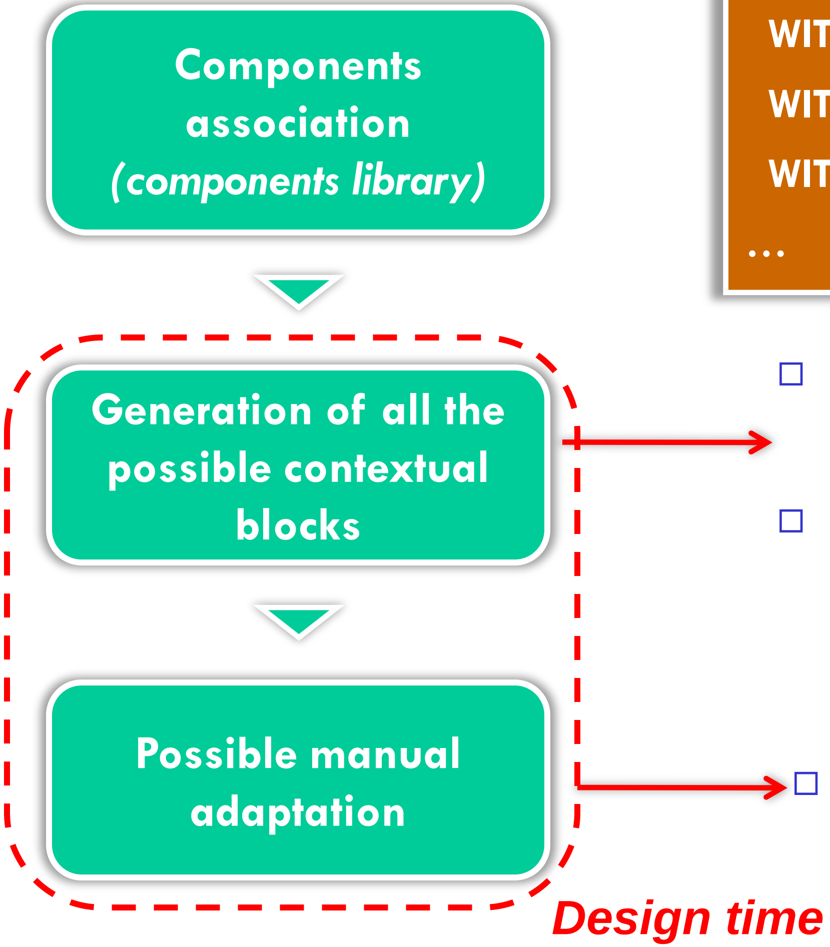
E E E

D D D

Refresh component

The designer’s tasks

```
…
{CREATE  CONCEPT <Concept Name> WHEN <Condition >
  WITH ENABLE COMPONENT <PerLa_Query>
  WITH DISABLE COMPONENT <PerLa_Query>
  WITH REFRESH COMPONENT <Period>
…
```


Components association
(components library)
Generation of all the possible contextual blocks
Possible manual adaptation
Design time

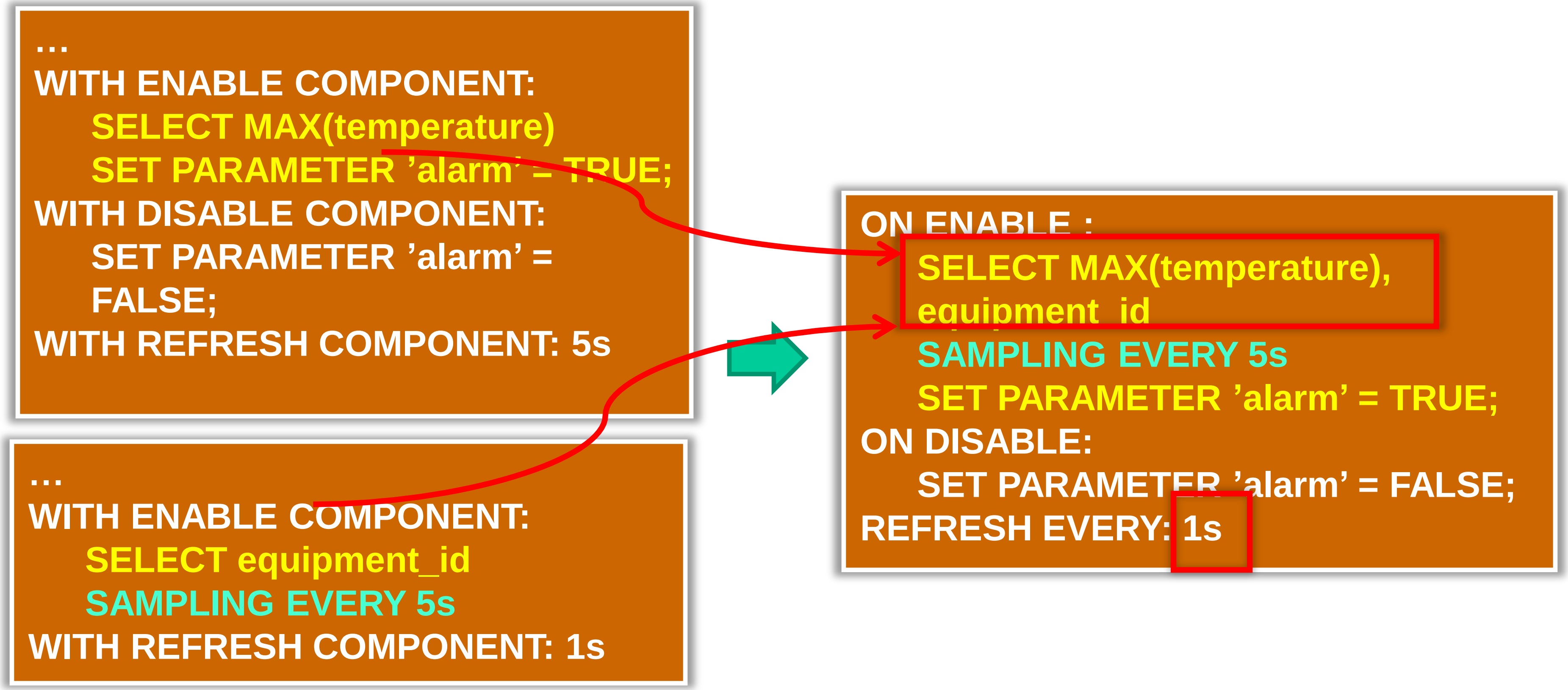
- □ Building of a composite contextual blocks library
- □ Verification of the composite block correctness (**QueryAnalyzer/Optimizer**)
- □ If required for peculiar situations

Design time vs. run time composition

- **Design time:**
  - Fully controlled by the designer
  - Static vision
- **Run time:**
  - Autonomic behaviour of the system
    - Contextual blocks are composed only for the active context
  - No further changes allowed
  - Performance issues (more contexts can be simultaneously active causing frequent context switching!)

A suitable trade-off is a designer’s choice based on the system requirements

An example



```
…
WITH ENABLE COMPONENT:
    SELECT MAX(temperature)
    SET PARAMETER ’alarm’ = TRUE;
WITH DISABLE COMPONENT:
    SET PARAMETER ’alarm’ =
    FALSE;
WITH REFRESH COMPONENT: 5s
```

```
…
WITH ENABLE COMPONENT:
    SELECT equipment_id
    SAMPLING EVERY 5s
WITH REFRESH COMPONENT: 1s
```

```
ON ENABLE :
    SELECT MAX(temperature),
    equipment_id
    SAMPLING EVERY 5s
    SET PARAMETER ’alarm’ = TRUE;
ON DISABLE:
    SET PARAMETER ’alarm’ = FALSE;
REFRESH EVERY: 1s
```

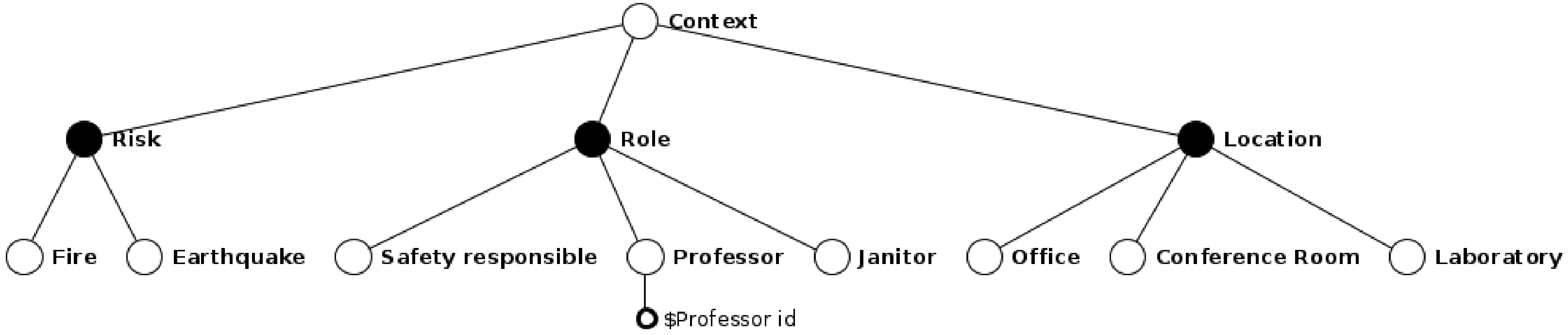
- Clause optimization (e.g: SELECT)
- Highest refresh frequency selection (lowest time constant)

Example 1: office risk management

- Suppose to have a pervasive system to monitor the potential risks at the DEI Department of Politecnico di Milano

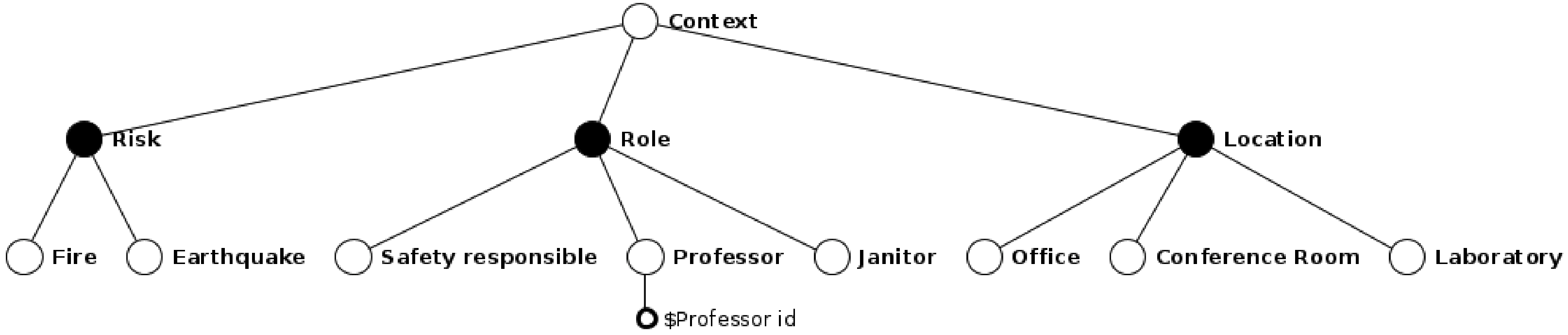
Example 1: office risk management

- Suppose to have a pervasive system to monitor the potential risks at the DEI Department of Politecnico di Milano



Example 1: office risk management

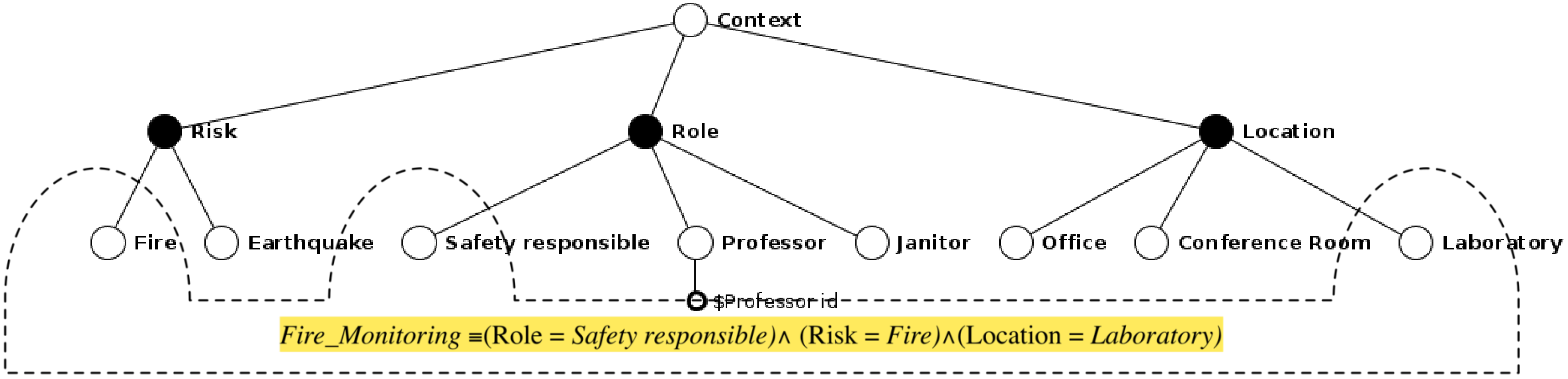
- Suppose to have a pervasive system to monitor the potential risks at the DEI Department of Politecnico di Milano



- We want to define a context-aware behaviour to control:
  - Fire
  - Earthquake

Example 1: office risk management

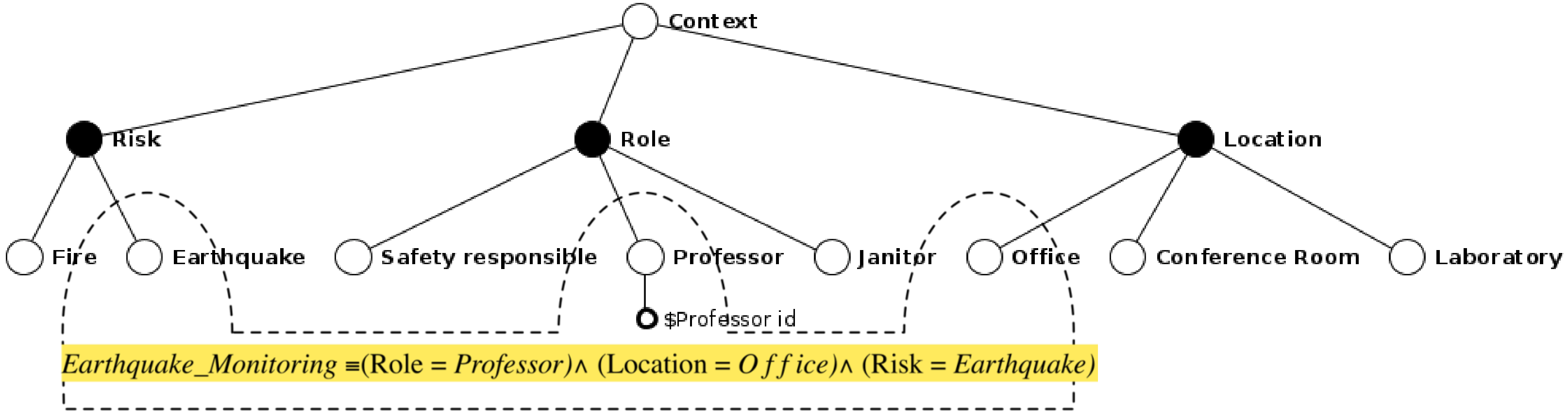
- Suppose to have a pervasive system to monitor the potential risks at the DEI Department of Politecnico di Milano



Fire_Monitoring ≡(Role = *Safety responsible*)∧ (Risk = *Fire*)∧(Location = *Laboratory*)

Example 1: office risk management

- Suppose to have a pervasive system to monitor the potential risks at the DEI Department of Politecnico di Milano



$Earthquake\_Monitoring \equiv (\text{Role} = Professor) \wedge (\text{Location} = Office) \wedge (\text{Risk} = Earthquake)$

Example 1: office risk management(2)

Fire Risk concept A

```
CREATE CONCEPT Fire
WHEN temperature > 40
        WITH ENABLE COMPONENT:
                SELECT MAX(temperature)
                SET PARAMETER ’alarm’ = TRUE;
        WITH DISABLE COMPONENT:
                SET PARAMETER ’alarm’ = FALSE;

WITH REFRESH COMPONENT: 5s
```

Earthquake Risk concept B

```
CREATE CONCEPT Earthquake
        WHEN delta_x > 2 AND delta_y > 3
         WITH ENABLE COMPONENT:
                SELECT delta_x,delta_y;
        WITH REFRESH COMPONENT: 1s
```

Example 1: office risk management(3)

Location

C

```
CREATE CONCEPT Office
    WHEN get_current_location() = ’Office’
    WITH ENABLE COMPONENT:
        SELECT office_floor
        SAMPLING EVERY 2m
```

D

```
CREATE CONCEPT Laboratory
    WHEN get_current_location() = ’Laboratory’
    WITH ENABLE COMPONENT:
        SELECT equipment_id
        SAMPLING EVERY 5s
    WITH REFRESH COMPONENT: 1s
```

E

```
CREATE CONCEPT Conference Room
    WHEN get_current_location() = ’Conf. room’
    WITH ENABLE COMPONENT:
        SELECT room_name
        SAMPLING EVERY 20s
    WITH REFRESH COMPONENT: 1s
```

Role

F

```
CREATE CONCEPT Professor
    CREATE ATTRIBUTE $Professor_Id
    WHEN get_current_role() = ’Professor’
    WITH ENABLE COMPONENT:
        SELECT professor_name,professor_surname
        WHERE ’professor_id’ = CDT.Role.Professor_ID
```

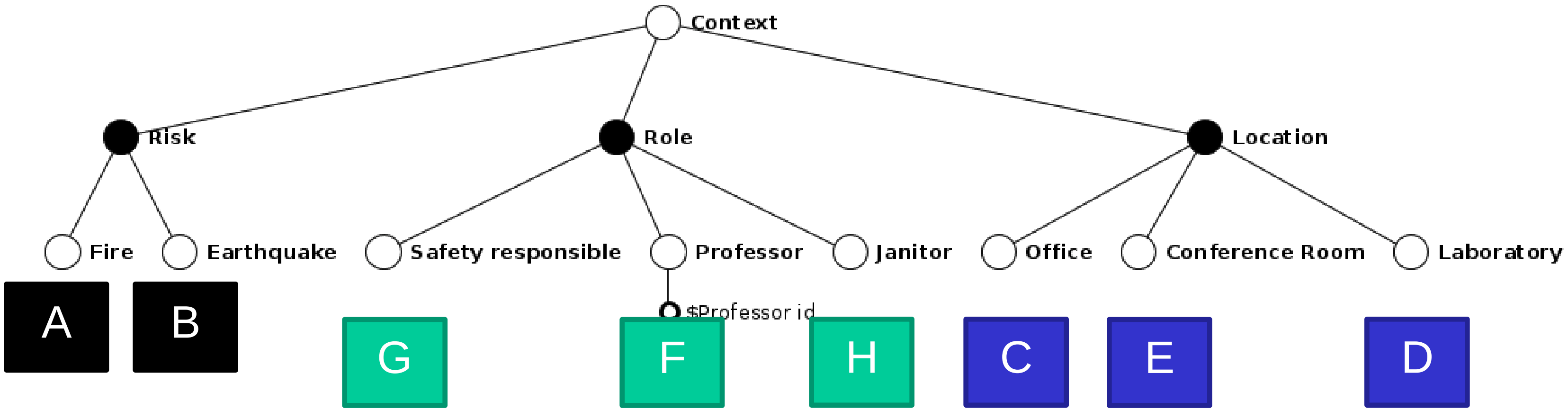
G

```
CREATE CONCEPT Safety_Responsible
    WHEN get_current_role() = ’Safety Responsible’
```

H

```
CREATE CONCEPT Janitor
    WHEN get_current_role() = ’Janitor’
```

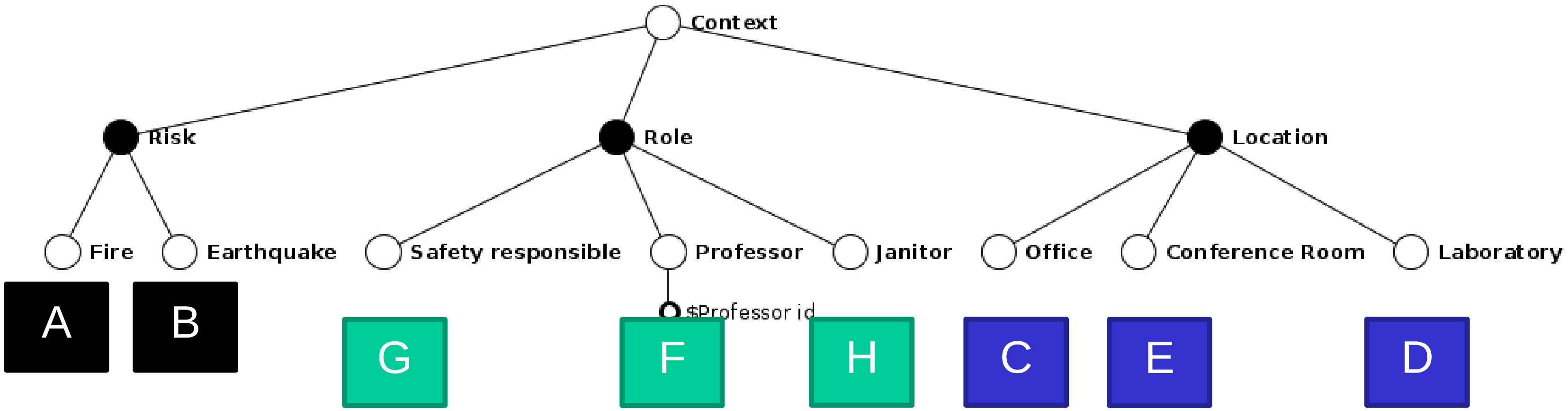
Example 1: office risk management



Fire_Monitoring º (Role = *Safety responsible*)Ù (Risk = *Fire*)Ù(Location = *Laboratory*)

Fire_Monitoring ≡ A ⊕ G ⊕ D

Example 1: office risk management



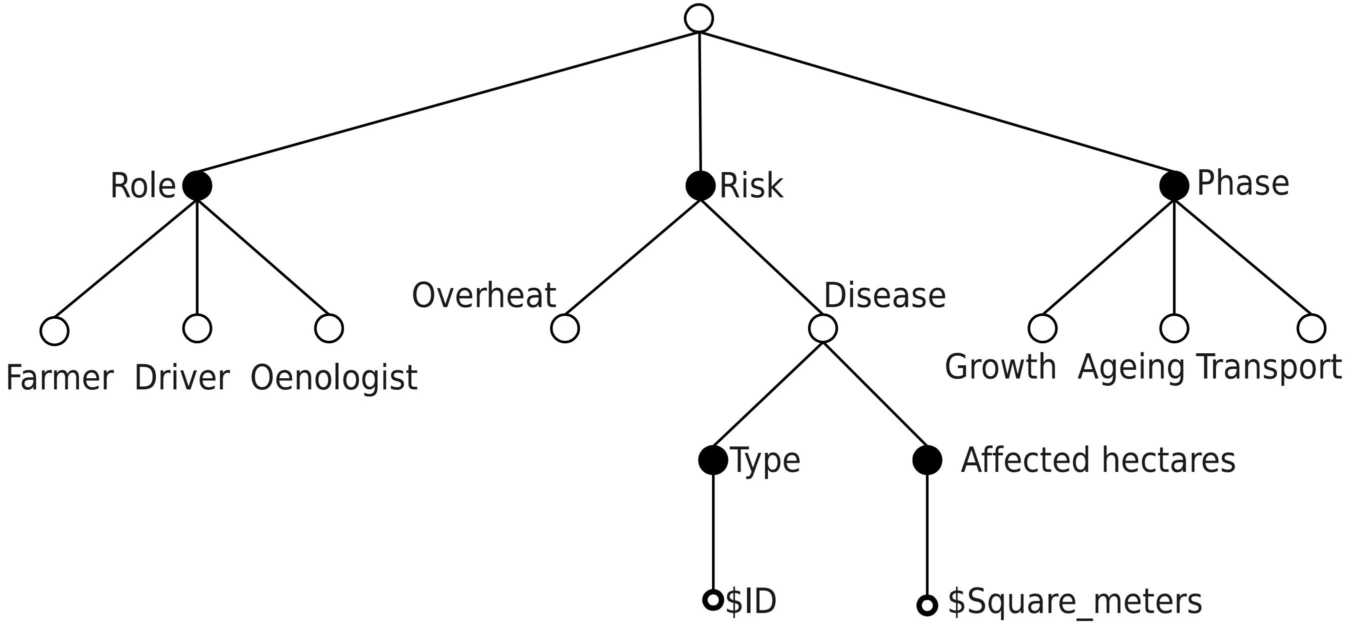
Fire_Monitoring º (Role = *Safety responsible*)Ù (Risk = *Fire*)Ù(Location = *Laboratory)*

Fire_Monitoring ≡ A ⊕ G ⊕ D

```
CREATE CONTEXT Fire_Monitoring
ACTIVE IF (temperature > 40 AND ...)
ON ENABLE:
        SELECT MAX(temperature), equipment_id     SAMPLING EVERY 5s
        SET PARAMETER ’alarm’ = TRUE;

ON DISABLE:
        SET PARAMETER ’alarm’ = FALSE;
REFRESH EVERY 1s;
```

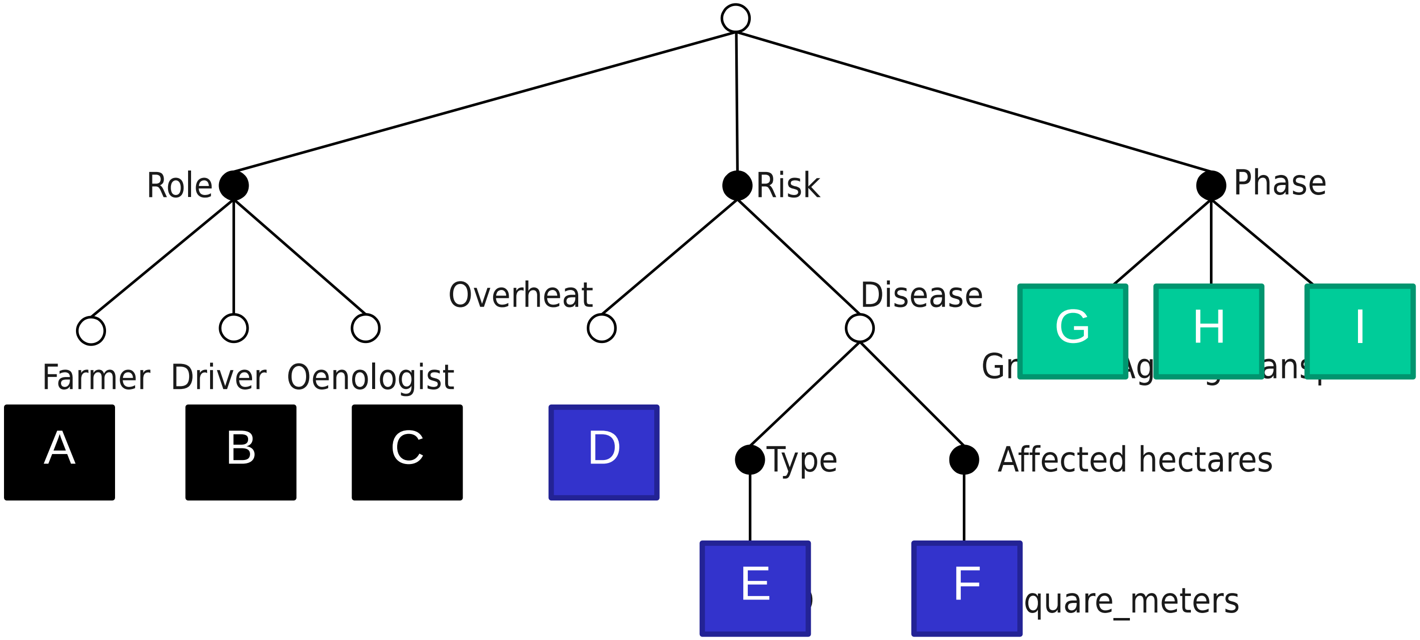
Example 2: vineyard monitoring



$Growth\_Monitoring \equiv (\text{Role} = Farmer) \wedge (\text{Phase} = Growth) \wedge (\text{Disease.Type} = 3) \wedge (\text{Disease.}AffectedHectares = 200)$

$Transport\_Monitoring \equiv (\text{Role} = Driver) \wedge (\text{Phase} = Transport) \wedge (\text{Risk} = Overheat)$

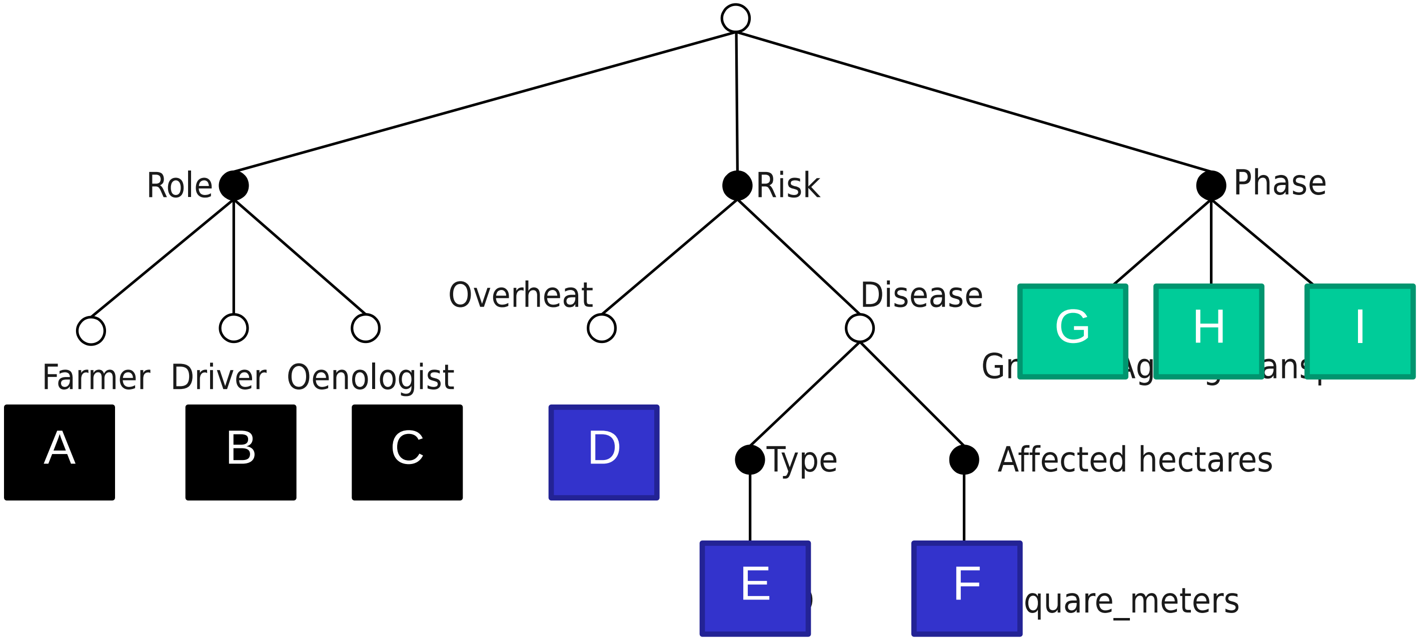
Example 2: vineyard monitoring



$Growth\_Monitoring \equiv (\mathrm{Role} = Farmer) \wedge (\mathrm{Phase} = Growth) \wedge (\mathrm{Disease.Type} = 3) \wedge (\mathrm{Disease.}AffectedHectares = 200)$

$Transport\_Monitoring \equiv (\mathrm{Role} = Driver) \wedge (\mathrm{Phase} = Transport) \wedge (\mathrm{Risk} = Overheat)$

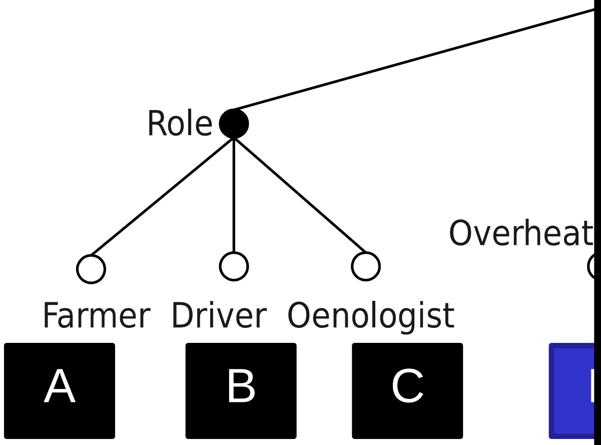
Example 2: vineyard monitoring



$\mathit{Growth\_Monitoring} \equiv A \oplus G \oplus E \oplus F$

$\mathit{Transport\ Monitoring} \equiv B \oplus I \oplus D$

Example 2: vineyard monitoring



```
CREATE CONTEXT Growth_Monitoring
ACTIVE IF phase = ’growth’ AND role=’farmer’ AND
  Disease.Type=3 AND
Disease.Affected_Hectares = 200 REFRESH EVERY 1 d;
ON ENABLE (Growth_Monitoring)
       SELECT humidity,temperature
       WHERE humidity > 0 AND temperature > 0
       SAMPLING EVERY 6 h
       EXECUTE IF EXISTS humidity,temperature AND
    location=’vineyard’
ON DISABLE (Growth_Monitoring)
       DROP CONTEXT Growth_Monitoring;


CREATE CONTEXT Transport_Monitoring
ACTIVE IF phase = ’transport’ AND role=’driver’ AND
   Risk=’overheat’ REFRESH EVERY 24 h;
ON ENABLE (Transport_Monitoring)
       SELECT temperature,gps_latitude,gps_longitude
       WHERE temperature > 30
       SAMPLING EVERY 120 s
       EXECUTE IF location = ’truck_departing_zone’
       SET PARAMETER ’alarm’ = TRUE;
ON DISABLE (Transport_Monitoring)
       DROP Transport_Monitoring;
       SET PARAMETER ’alarm’ = FALSE;
```

Growth_Monitoring ≡ A ⊕ G ⊕ E ⊕ F

Transport Monitoring ≡ B ⊕ I ⊕ D

Comparison with Active DB

ACTIVE DATABASES	PerLa FOR CONTEXT
EVENT data modification: insert, delete, update	EVENT general system events, clock, …
CONDITION (optional) SQL predicate	CONDITION context definition formula
ACTION sequence of SQL statements (or extensions, e.g. PL/SQL in Oracle)	ACTION Data, code, services tailoring, whatever action on the physical system

Comparison with Active DB

ACTIVE DATABASES	PerLa FOR CONTEXT
COUPLING (immediate/deferred)	ONLY IMMEDIATE
ATOMIC/INTERRUPTIBLE ACTIONS	ONLY ATOMIC
EVENT CONSUMPTION (never, local, global)	EVENT CONSUMPTION (never, only at context change)
CONFLICT RESOLUTION (serial/parallel)	CONFLICT RESOLUTION (serial, managed by priority policies)

Comparison with programming languages

Philosophy without Science is empty,
Science without Philosophy is blind

I. Kant

PARAPHRASE

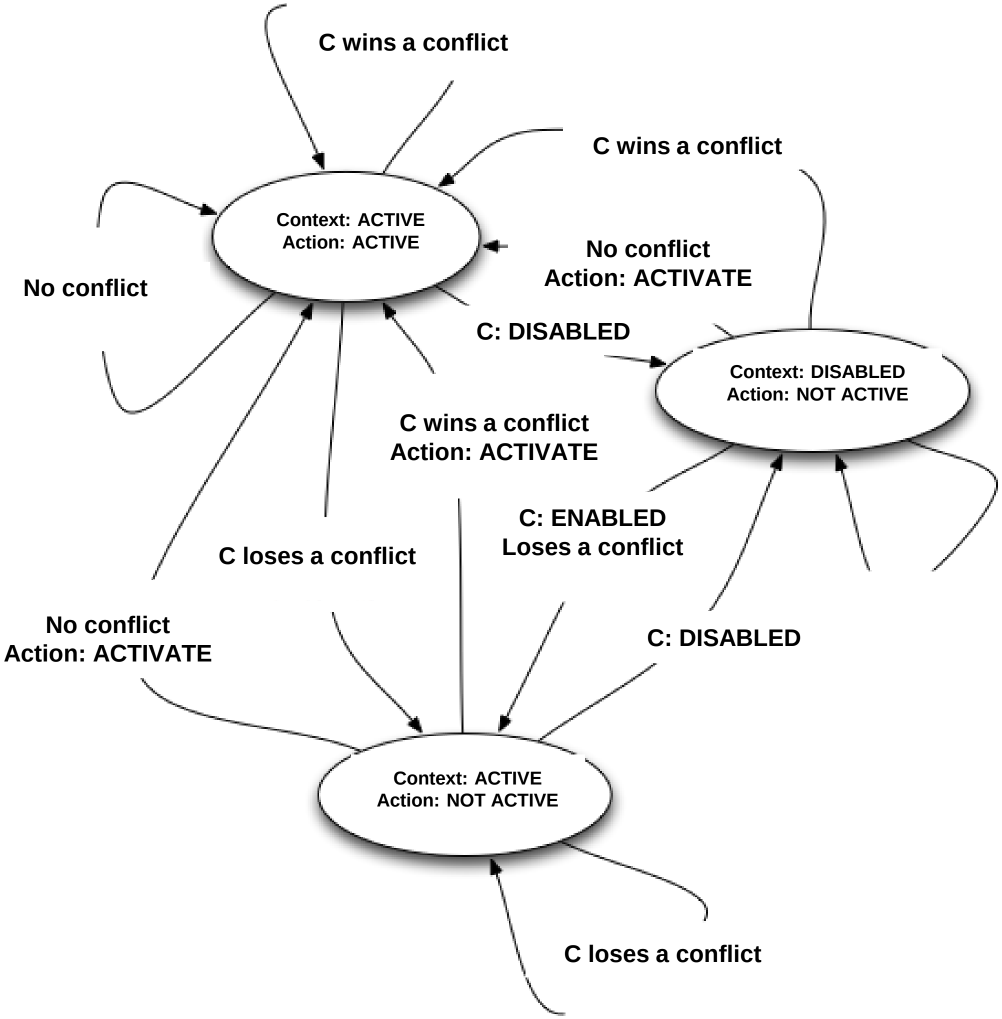
Programs without Data are empty,
Data without Programs are blind

F. A. Schreiber

Comparison with programming languages

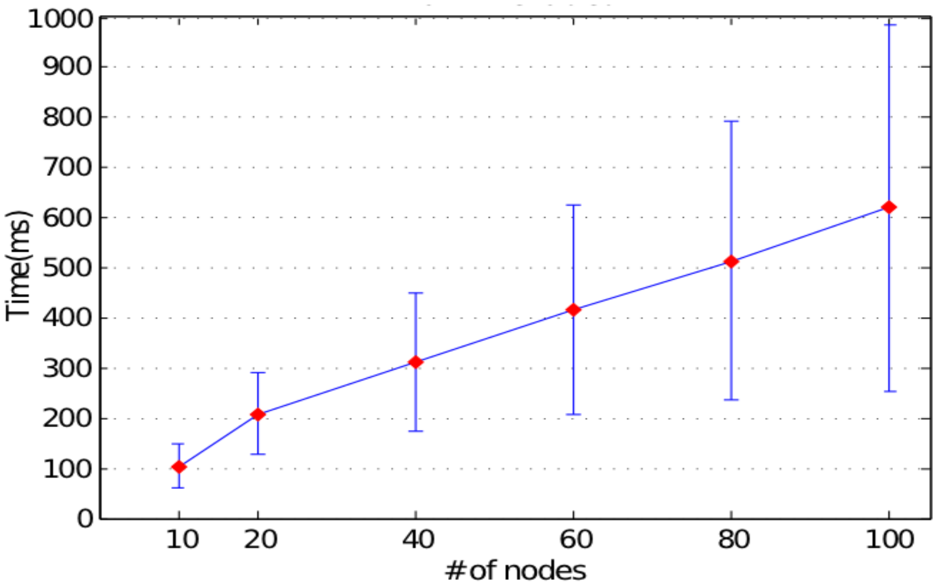
	PerLa	COP
Context	Numeric observables → Symbolic observables (Coutaz, CACM, 2005)	Any computationally accessible information (Hirschfeld, JOT, 2008) Numeric observables → Symbolic observables
Context model	Context Dimension Tree (CDT) Context Element ≡ {Dimension$_i$ = Value$_i$} Multiple active contexts	Left to application software Multiple active contexts
Context declaration	Contextual Block • activation • Enabling • Disabling • Change detection	Left to application software
Context sensing and recognition	LLQ from sensors User declared variables GET_ ACTIVE IF REFRESH EVERY	Layer activation mechanisms
Contextual actions	ON ENABLE ={TRUE/FALSE} → LLQ/HLQ/AQ ON DISABLE ={TRUE/FALSE} → LLQ/HLQ/AQ Partial components associated with each Context-element WITH {ENABLE/DISABLE/REFRESH} COMPONENT	Behavioural variations Partial methods WITH WITHOUT

Conflict resolution



Performance evaluation

- In its original configuration, PerLa's middleware scales linearly w.r.t. the operations (i.e.: LLQs, HLQs, AQs) that are performed on the deployed devices.



- The creation of the CDT and the search for active contexts scale linearly too (simple lookup control in every table) and thus _do not_ impact PerLa's linear behaviour.

Conclusions

- PerLa allows for an easy and rapid passage between *numeric* and *symbolic* observables.
- Morover it allows to model and define the context with the preferred granularity, and to actuate context-aware actions within the same language.
- It offers design support tools through the contextual block composition both at design and at run time.
- The PerLa system is operating in a rockfall monitoring project in Mt. San Martino in Lecco (MI) since April 2010.
- We are currently focused on the following issues:
  - the management of possible context conflicts.
  - the management of context evolution
  - Assessing C-A systems stability

Further readings

- on CDT:
  - Bolchini C., Curino C.A., Orsi G., Rossato R., Quintarelli E., Schreiber F.A., Tanca L. - And What Can Context Do For Data? - Communications of ACM (VE), Vol.52, n. 11, p.136-140, (2009)
  - Bolchini C. , Quintarelli E. , Tanca L. - Carve: Context-aware automatic view definition over relational databases - Information Systems, Accepted manuscript (unedited version available online: 12-MAY-2012).
  - **http://tanca.dei.polimi.it/images/documents/sac2012.pdf**
- on PerLa:
  - Schreiber F.A., Camplani R., Fortunato M., Marelli M., Rota G. - PerLa: A Language and Middleware Architecture for Data Management and Integration in Pervasive Information Systems - IEEE Transactions on Software Engineering, Vol. 38, n. 2, pp. 478-496, (2012)
  - **http://perlawsn.sourceforge.net**

?

THANK YOU