A rule-based methodology for the adaptation of Web based Information System

Roberto De Virgilio

Dipartimento di Informatica e Automazione
Università degli studi Roma Tre

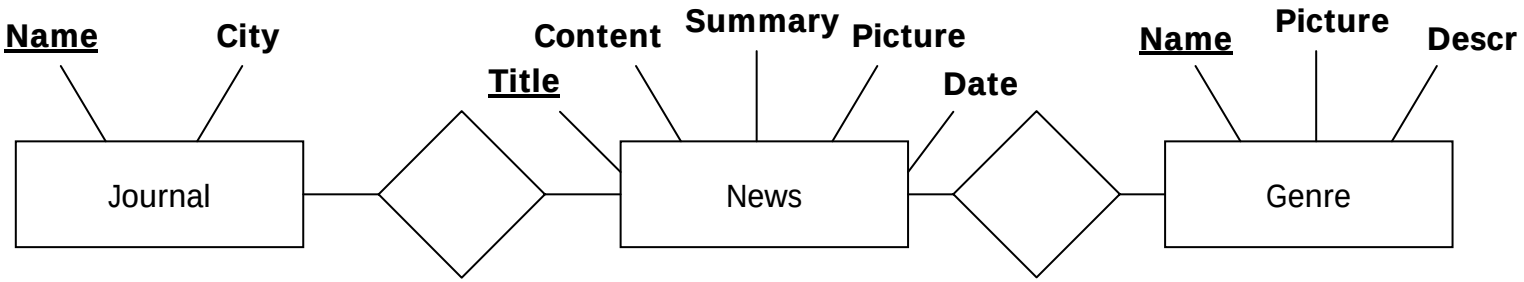
Overview

- Scenario
  - An ever increasing number of mobile devices can provide ***everywhere*** and ***any time*** access to the Web: they offer limited computing capabilities
- Problem
  - content delivery adaptation of Web information to the **context** of the client
- Goal
  - general approach to adapt a Web Information System according to different and possibly heterogeneous contexts

design of WIS data-intensive

- we focus our attention on the large category of ***data intensive*** Web Information Systems
- a *Web access* to large amounts of structured data
- ***model-driven*** methodologies to design a WIS

An Example: content layer



Journal

J	Name	City
J1	Roma	Rome
J2	Oggi	Naples

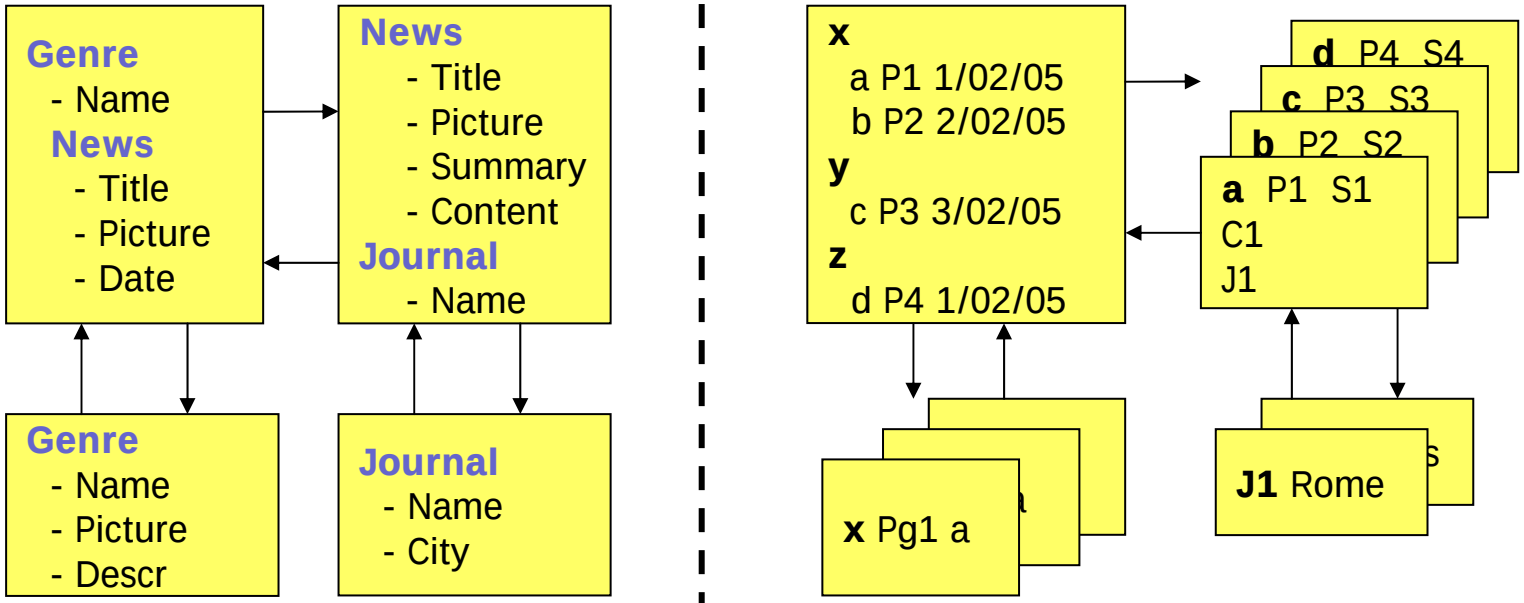
News

Title	Summary	Content	Picture	Date	Gn	J
a	S1	1	P1	1/02/05	x	J1
b	S2	2	P2	2/02/05	x	J2
c	S3	3	P3	3/02/05	y	J1
d	S4	4	P4	1/02/05	z	J2

Genre

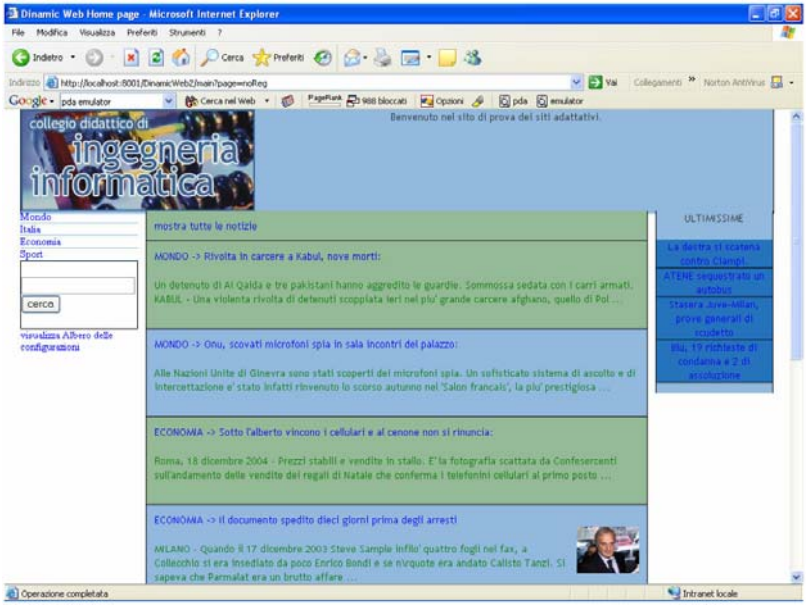
Gn	Pict	Descr
x	Pg1	aa
y	Pg2	bb
z	Pg3	cc

An Example: navigation layer



An Example: presentation layer

```
body {
    background-color: yellow;
    font-family: georgia;
    font-size:16pt;
    color: black;
}
h1 {
    font-size: 24pt;  font-weight: navy;
    color:purple;
}
#picture span {
    width: 100px;  height: 100px;
    float: center;
}
...
```



Adaptation on WIS (several aspects)

- a WIS is **adaptive** if it is able to modify and personalize delivery of contents and services according to the **context** of the client (human being or application).

definition of context

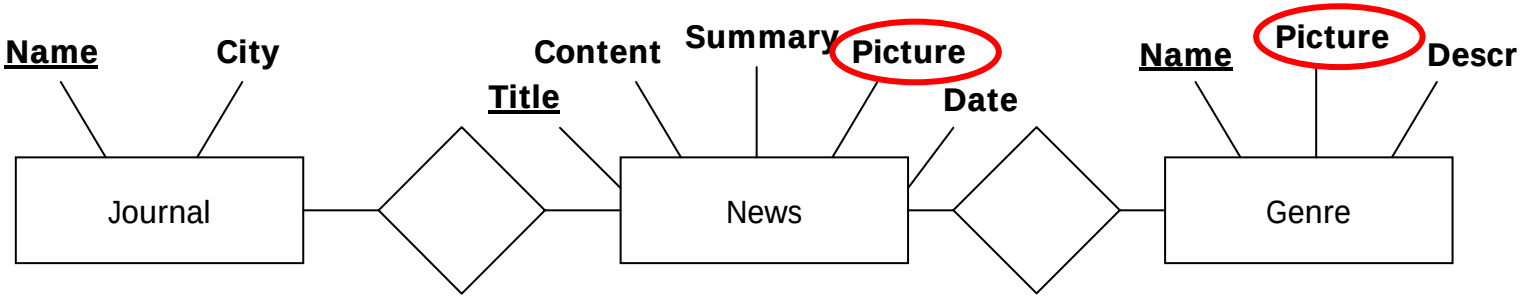
- "*a set of **attributes** that characterizes the capabilities of the access mechanism, the preferences of the user and other aspects of the context into which a Web page is to be delivered*"

W3C Working Group on Device Independence

an example of context

- "*a **user** having a **preference** of summaries for **genre x** and **z** with his/her **mobile phone** with a **black and white** display of **limited size** and **without graphical capabilities***"

The “adapted” example



Journal

J	Name	City
J1	Roma	Rome
J2	Oggi	Naples

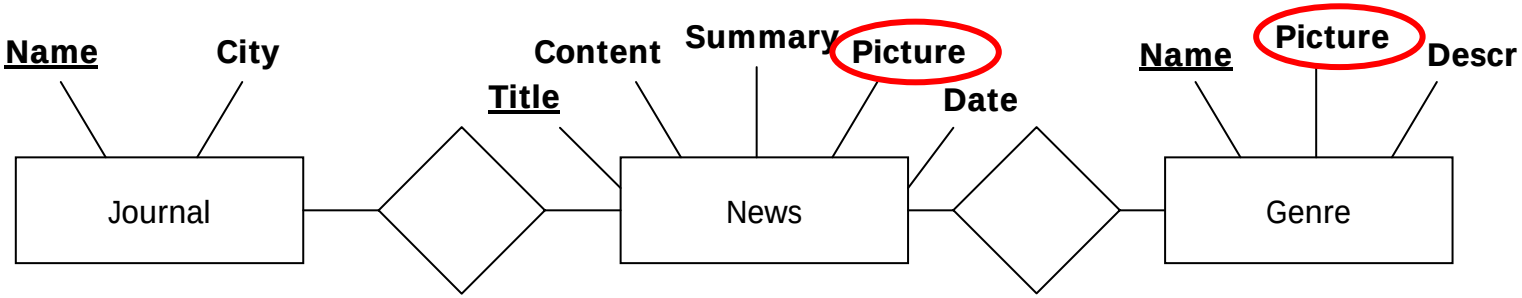
News

Title	Summary	Content	Picture	Date	Gn	J
a	S1	1	P1	1/02/05	x	J1
b	S2	2	P2	2/02/05	x	J2
c	S3	3	P3	3/02/05	y	J1
d	S4	4	P4	1/02/05	z	J2

Genre

Gn	Pict	Descr
x	Pg1	aa
y	Pg2	bb
z	Pg3	cc

The “adapted” example



Journal

J	Name	City
J1	Roma	Rome
J2	Oggi	Naples

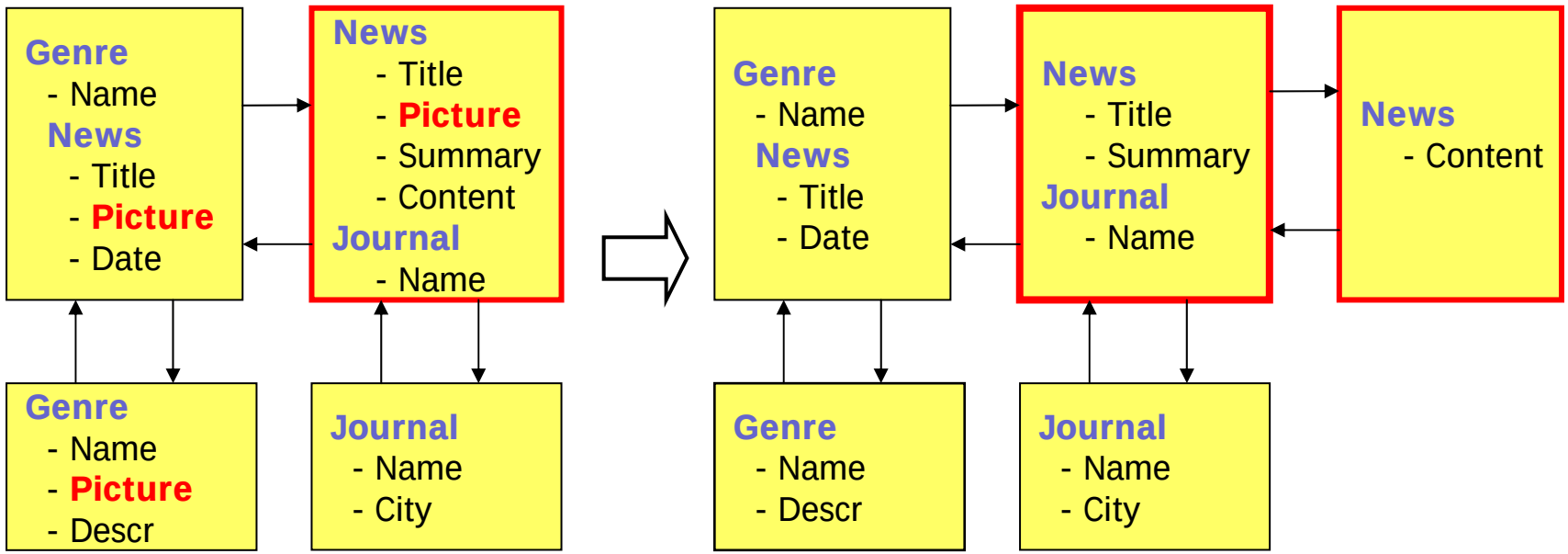
News

Title	Summary	Content	Picture	Date	Gn	J
a	S1	1	P1	1/02/05	x	J1
b	S2	2	P2	2/02/05	x	J2
c	S3	3	P3	3/02/05	y	J1
d	S4	4	P4	1/02/05	z	J2

Genre

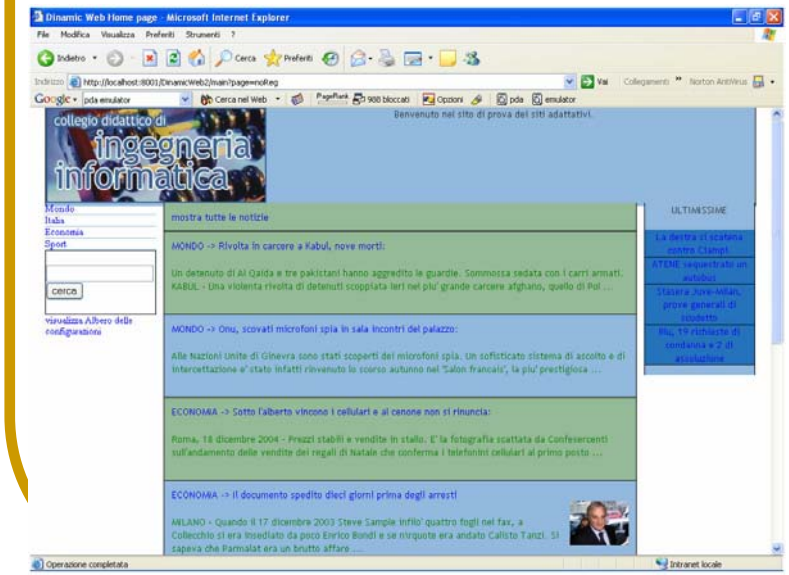
Gn	Pict	Descr
x	Pg1	aa
y	Pg2	bb
z	Pg3	cc

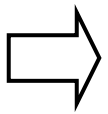
The “adapted” example



```
body {
    background-color: yellow;
    font-family: georgia;
    font-size:16pt;
    color: black;
}
h1 {
    font-size: 24pt;  font-weight: navy;
    color:purple;
}
#picture span {
    width: 100px;  height: 100px;
    float: center;
}
...
```

```
body {
    background-color: white;
    font-family: arial;
    font-size:10pt;
    color: black;
}
h1 {
    font-size: 12pt;  font-weight: black;
    color:black;
}

...
```





Several approaches

- many approaches provide a solution for the **whole** development process: design, implementation, configuration and deployment

- they are often **specific solutions**, suited only for predefined coordinates of adaptation and **hardly reusable** for adding new adaptation capabilities to existing systems

Profiles

- **profile**: a description of an autonomous aspect of the context in which the Web site is accessed.

device

user preference

temporal aspects

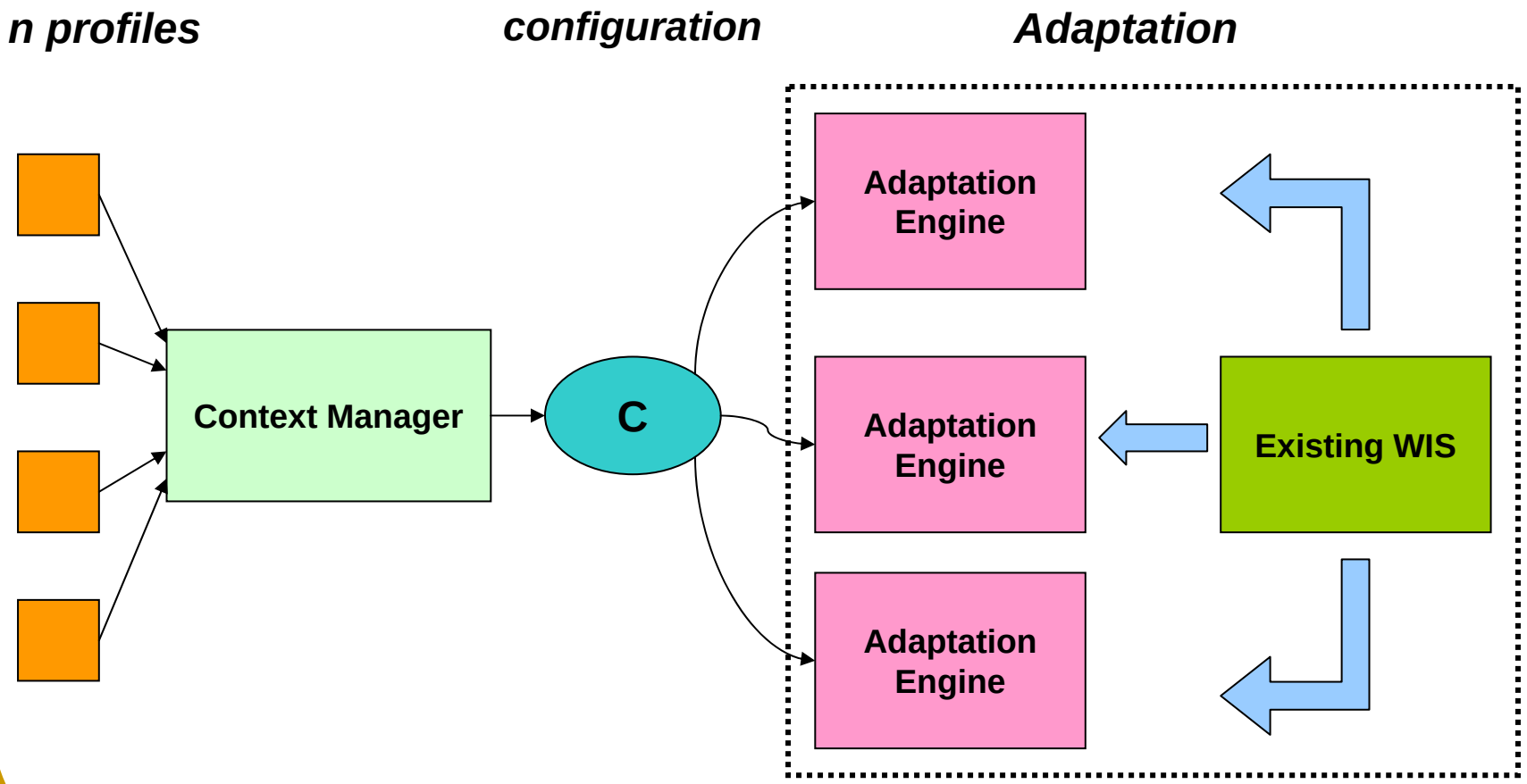
location

network

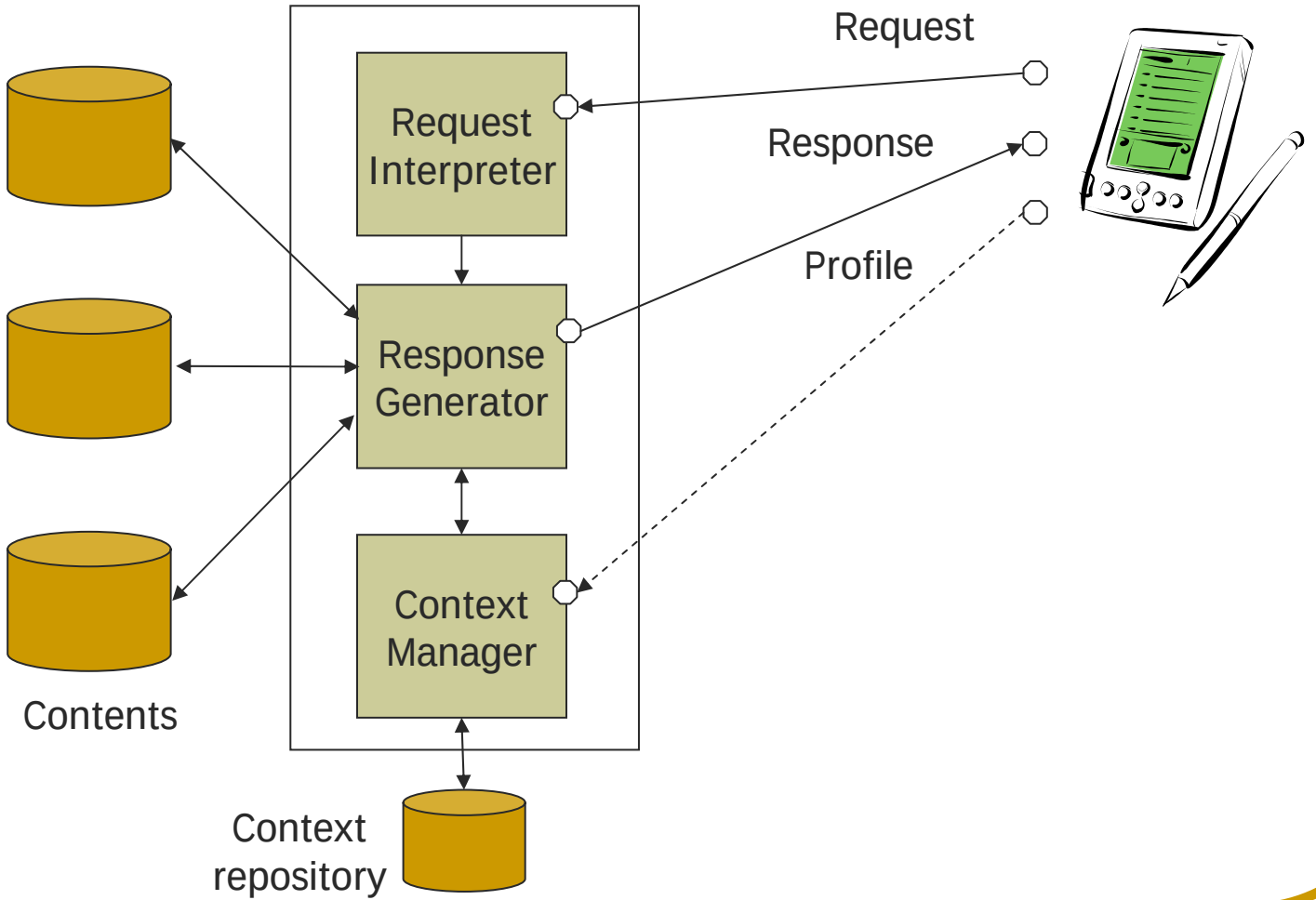
Configurations

- **configuration**: **abstract** description that specifies how information has to be delivered by taking into account the requirements of adaptation for a profile.

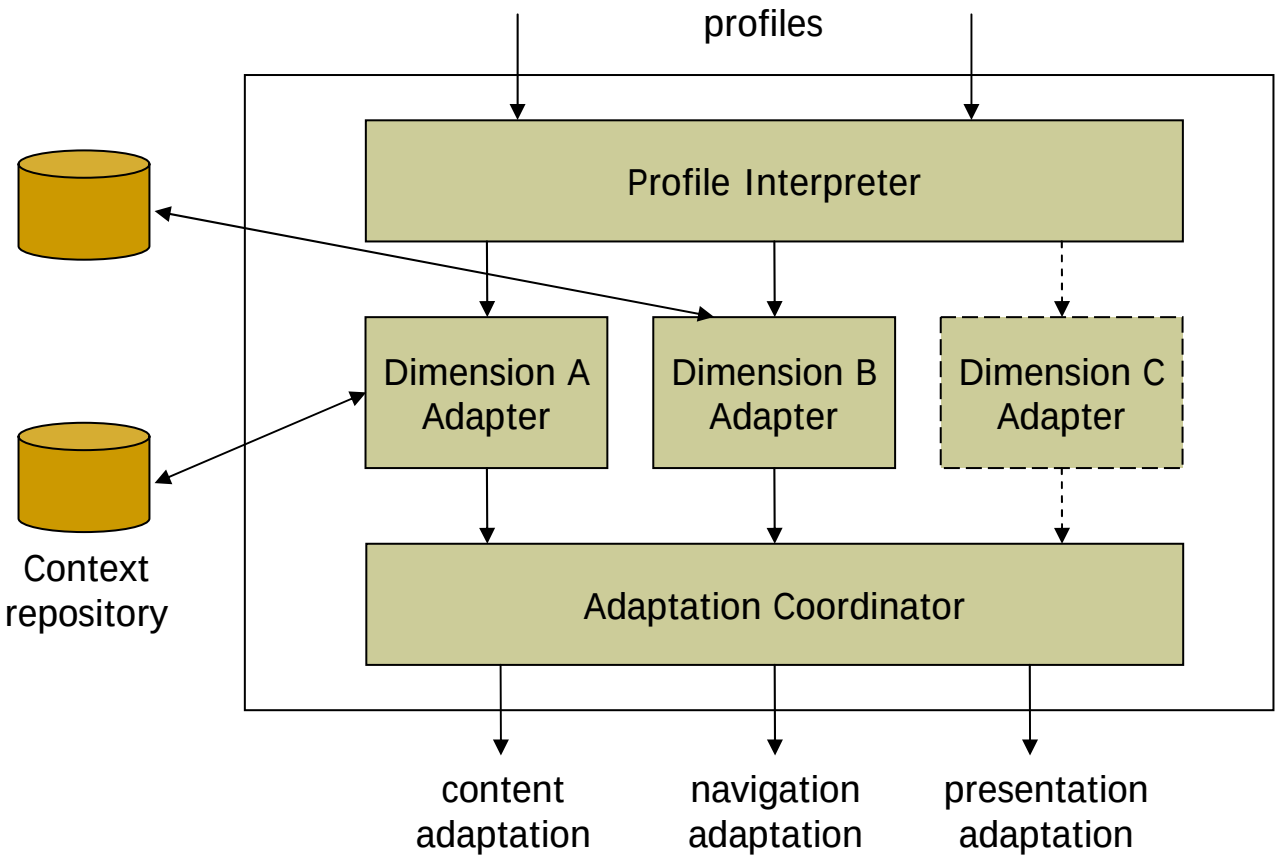
from profiles to configuration



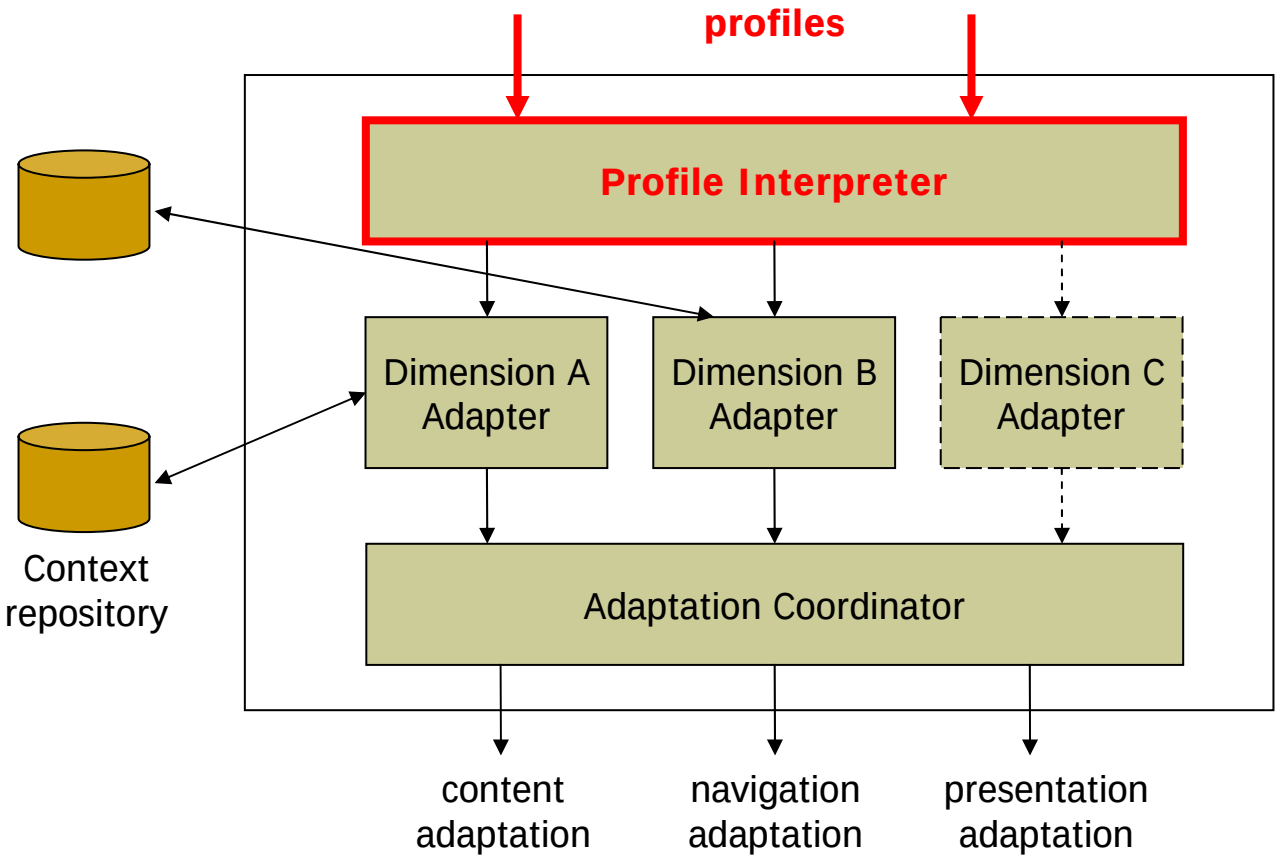
An architecture of reference



Context Manager



The adaptation process: context analysis



Different Formats to express a profile

text files in ad-hoc format

XML

DTDs

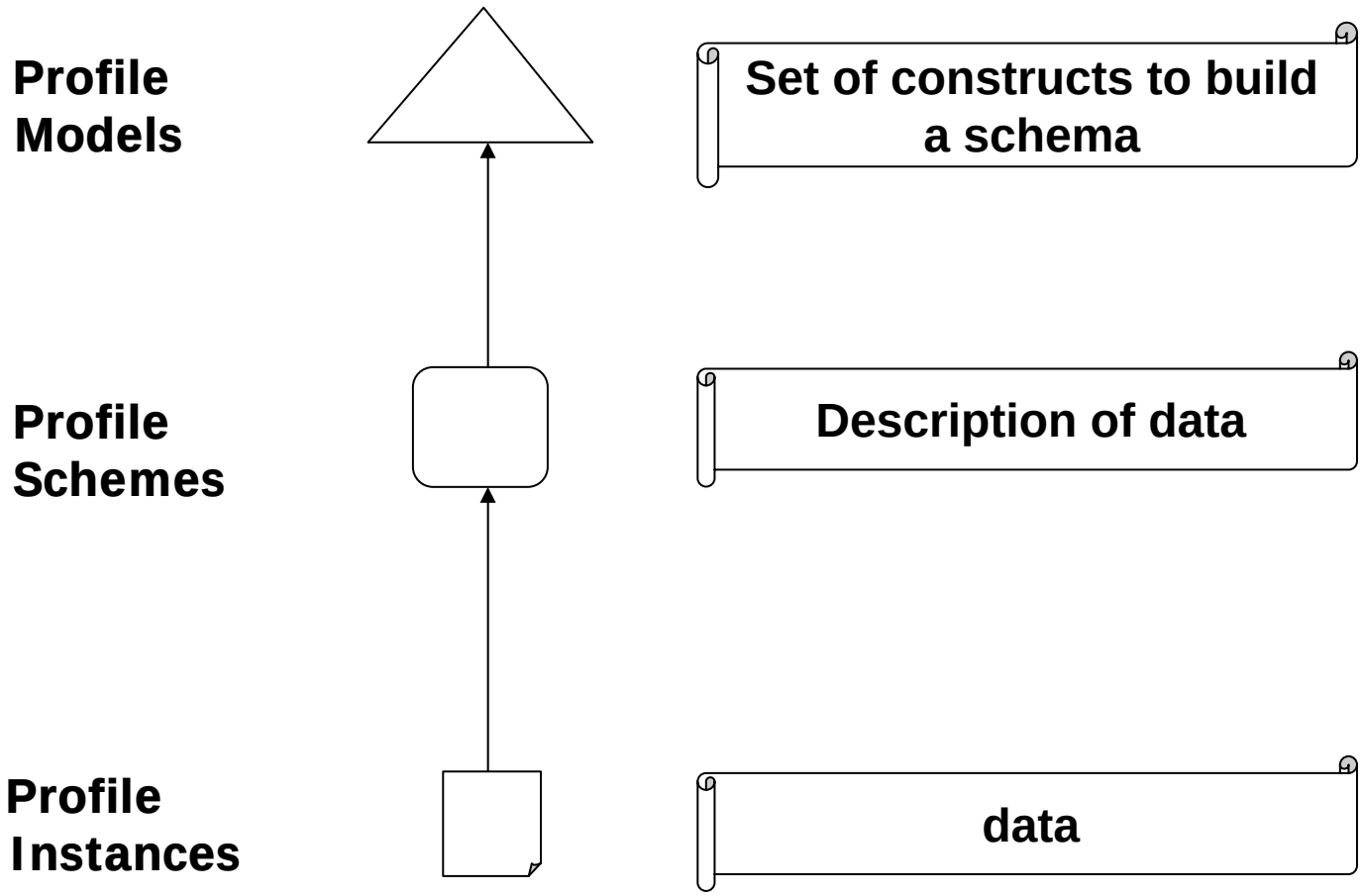
CC/PP

HTTP headers

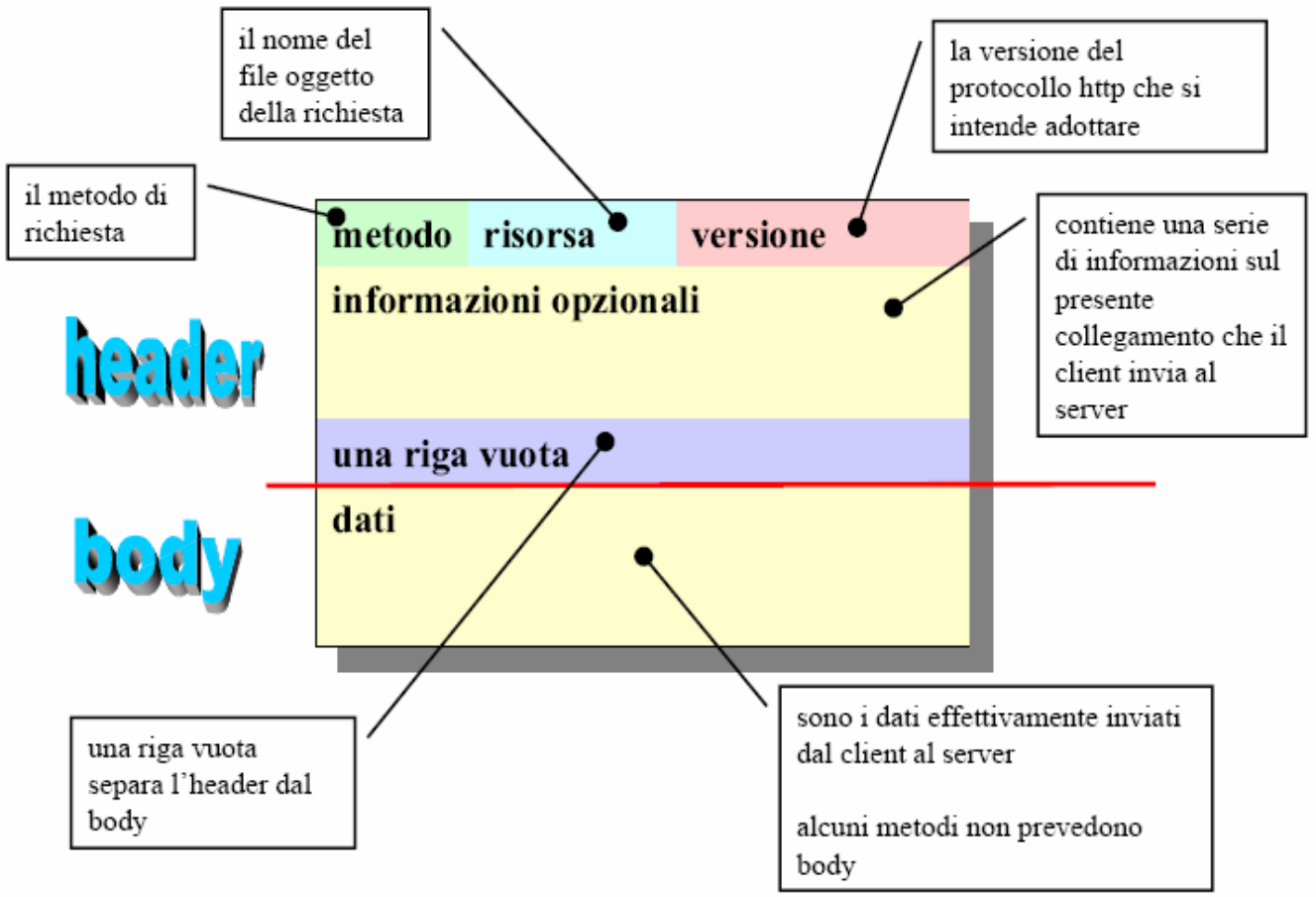
RDF

... and their dialects.

Abstraction Levels



Headers HTTP



Headers HTTP

Request = Request-Line

**((general-header | request-header | entity-header) CRLF)*

CRLF

[message-body]

http://www.w3.org/Protocols/rfc2616/rfc2616-sec5.html#sec5

Headers HTTP

POST register.jsp HTTP/1.1

Host: hi.iq

User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.7) Gecko/20040616 MultiZilla/1.6.4.0b

Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9, text/plaitn;q=0.8,video/x-mng,image/png,image/jpeg,image/gif;q=0.2, text/css,*/*;q=0.1

Accept-Language: en-us, en;q=0.50

Accept-Encoding: gzip, deflate, compress;q=0.9

Accept-Charset: ISO-8859-1, utf-8;q=0.66, *;q=0.66

From: rde79@yahoo.com

Content-Type: application/x-www-form-urlencoded

Content-Length: 36

Log File

Elemento	Descrizione
1) Indirizzo utente	*Indirizzo IP o nome del dominio dell'utente che accede al sito*
2) Rfc931	*Campo utilizzato per registrare il dominio nel caso di un web server con più homepage*
3) Autenticazione utente	*Identificativo (user name) dell'utente se richiesto dal documento consultato*
4) Data e orario	*Data e orario di accesso al documento*
5) GMT	*Differenza di jet lag dal fuso orario utilizzato*
6) Action	*GET o POST e documento richiesto*
7) Codice di risposta	*Codice di risposta (eventualmente per mostrare condizioni di errore)*
8) Dimensioni	*Dimensioni del documento trasferito*
9) Referrer	*URL di origine dell'utente*
10) Browser/system	*Browser e sistema da cui l'utente accede*

Log File

host144-98.pool80117.interbusiness.it **1)**

- **2)**

- **3)**

[01/Apr/2003:08:18:49 +0200] **4) + 5)**

"GET /~coppola/didactics/twm/aa2002-2003 HTTP/1.1" **6)**

302 **7)**

284 **8)**

"http://www.google.it/" **9)**

"Mozilla/2.065 (WinNT; I)" **10)**

XML

- *XML: client is able to send structured information*

```
<context>
  <coordinate 1> … </coordinate1>
  <coordinate 2> … </coordinate2>
  <coordinate 3> … </coordinate3>
  …
</context>
```

DTD schema

```
<!DOCTYPE TVSCHEDULE
 [<!ELEMENT TVSCHEDULE (CHANNEL+)>
  <!ELEMENT CHANNEL (BANNER,DAY+)>
  <!ELEMENT BANNER (#PCDATA)>
  <!ELEMENT DAY (DATE,(HOLIDAY|PROGRAMSLOT+)+)>
  <!ELEMENT HOLIDAY (#PCDATA)>
  <!ELEMENT DATE (#PCDATA)>
  <!ELEMENT PROGRAMSLOT (TIME,TITLE,DESCRIPTION?)>
  <!ELEMENT TIME (#PCDATA)>
  <!ELEMENT TITLE (#PCDATA)>
  <!ELEMENT DESCRIPTION (#PCDATA)>
  <!ATTLIST TVSCHEDULE NAME CDATA #REQUIRED>
  <!ATTLIST CHANNEL CHAN CDATA #REQUIRED>
  <!ATTLIST PROGRAMSLOT VTR CDATA #IMPLIED>
  <!ATTLIST TITLE RATING CDATA #IMPLIED>
  <!ATTLIST TITLE LANGUAGE CDATA #IMPLIED> ]>
```

XML schema

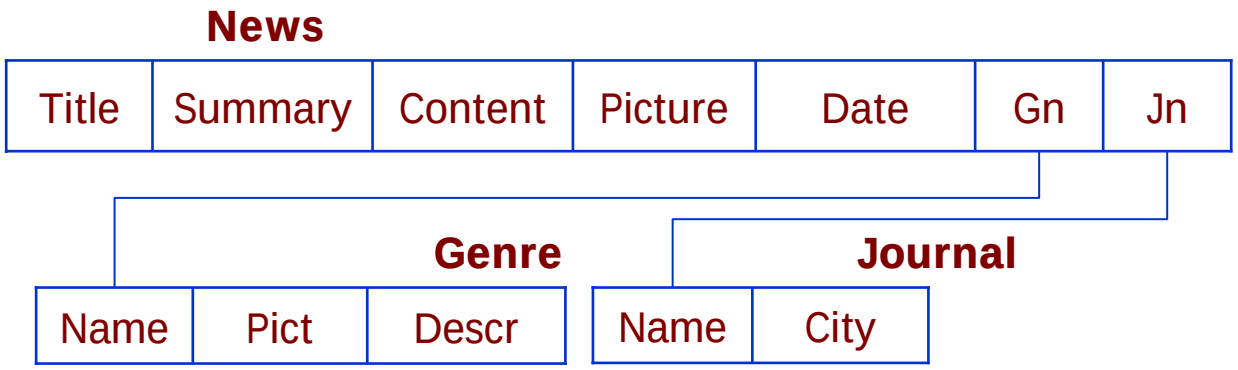
```
<xs:element name="shipto">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="name" type="xs:string"/>
   <xs:element name="address" type="xs:string"/>
   <xs:element name="city" type="xs:string"/>
   <xs:element name="country" type="xs:string"/>
  </xs:sequence>
 </xs:complexType>
</xs:element>
```

Relational schema

- *Common model to store data: Relational Model*

Table (Field1, Field2, …)

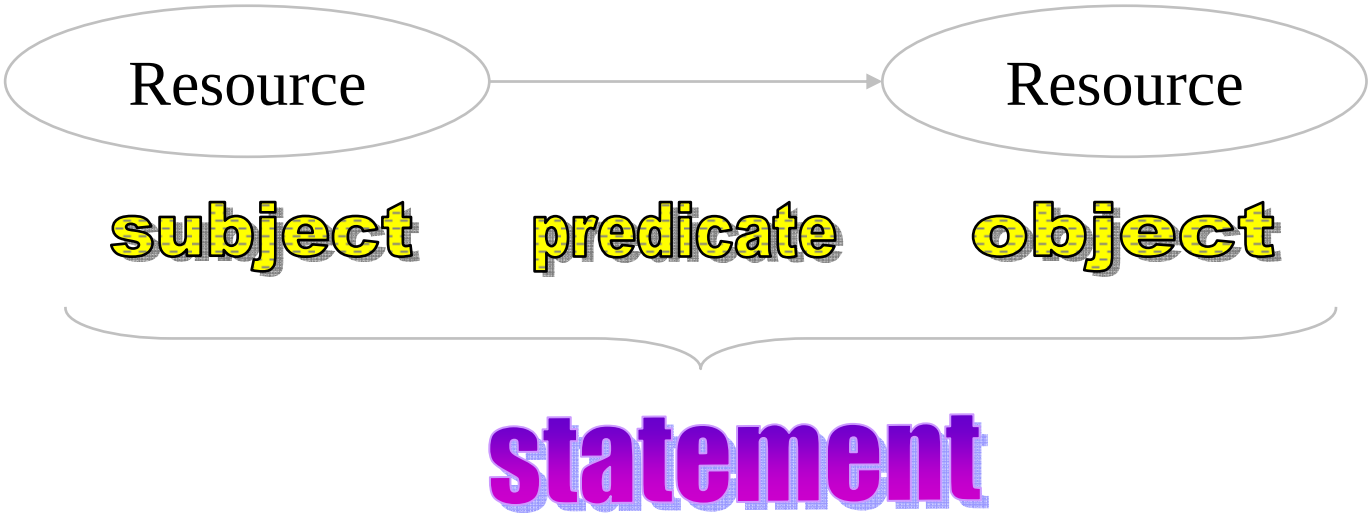
tuple



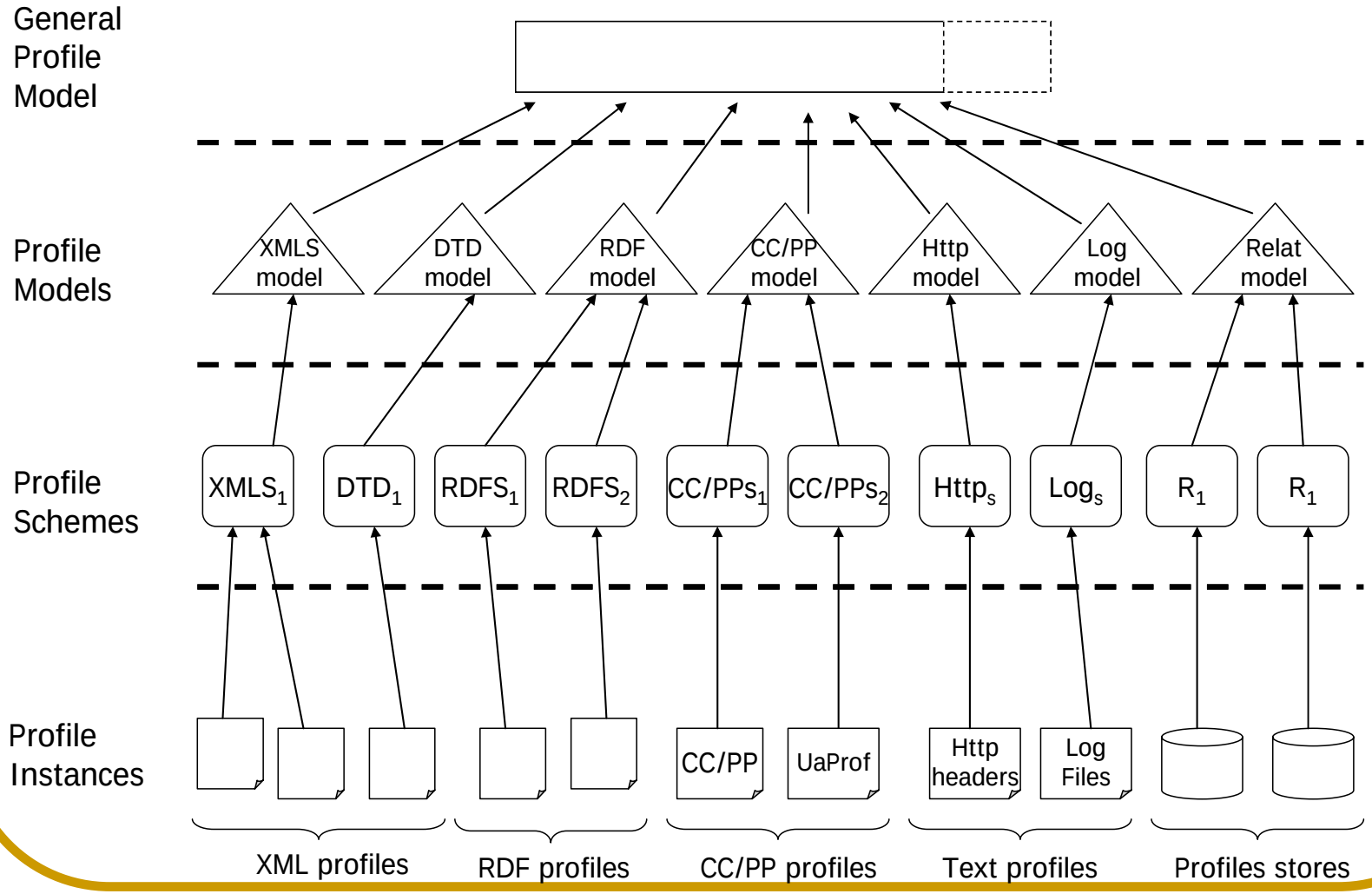
RDF

RDF: Resource Description Framework

- Resource (Web)
- Property (subclass of Resource)



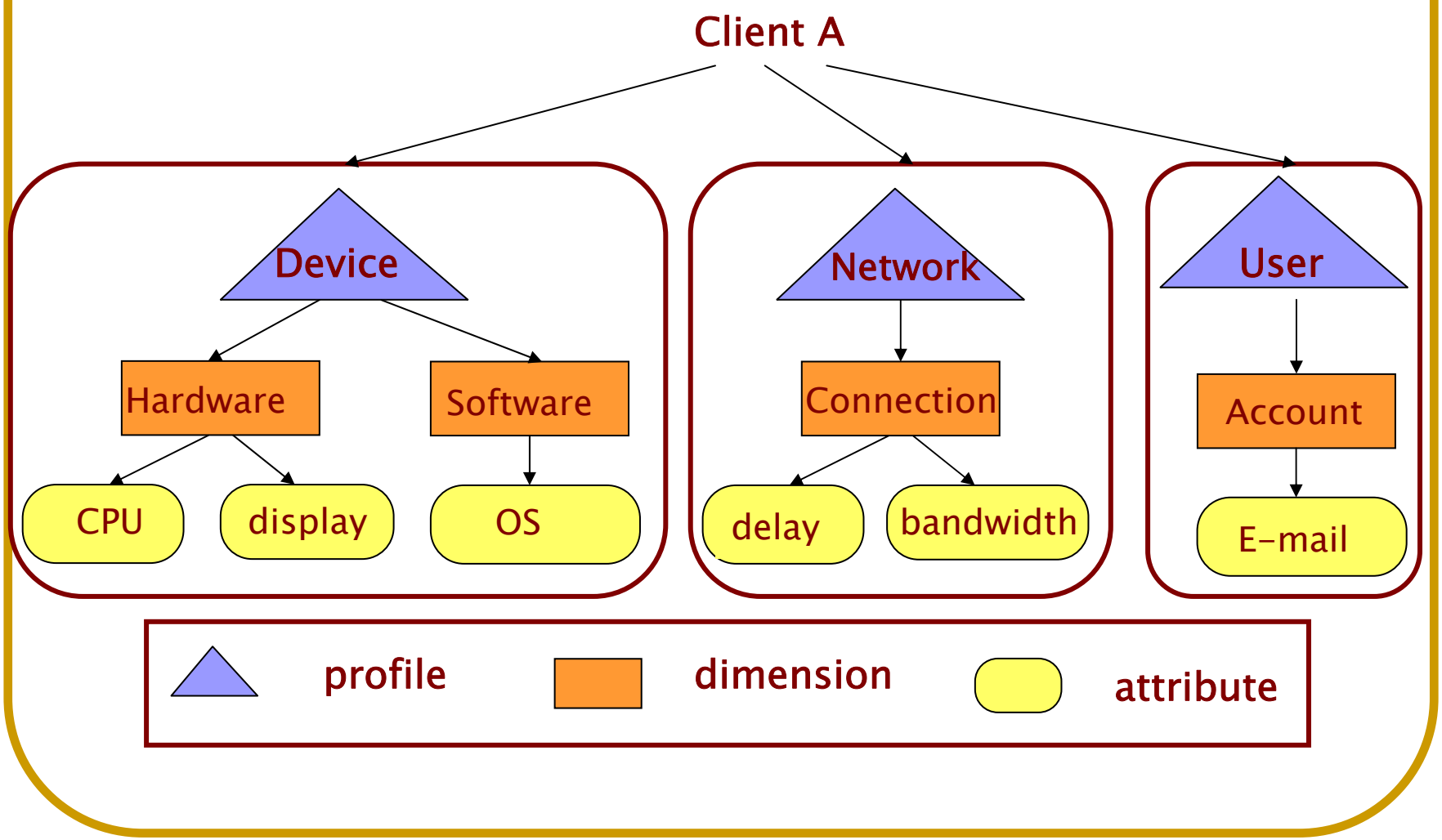
General Profile Model



“Generic” profile

- a **profile** is a **description** of an autonomous aspect of the context (device, user, network…)
- a **dimension** is property that characterizes a profile
- each dimension can present a set of **attributes**
  - **simple**: a value associated
  - **complex:** a set of (simple or complex) attributes associated

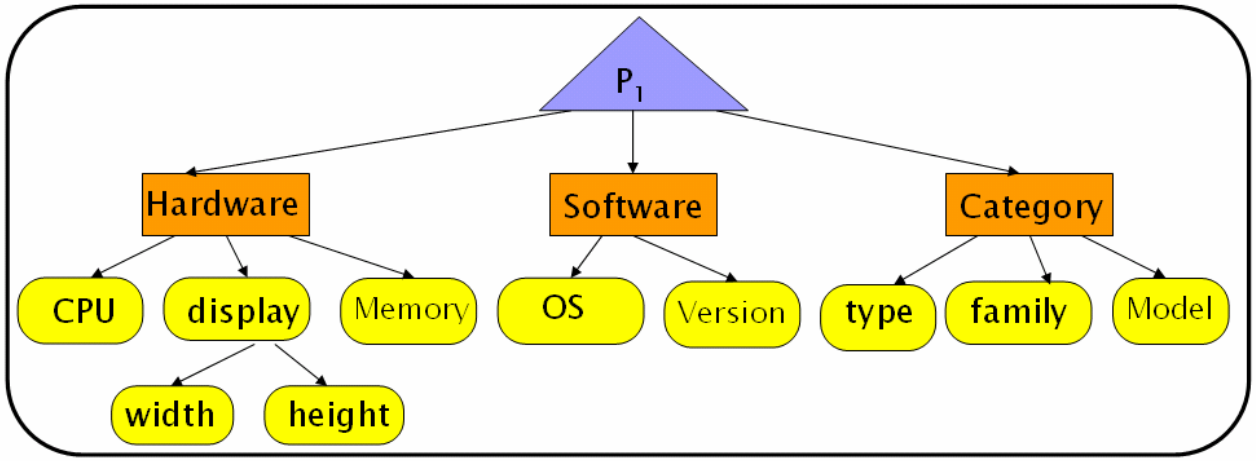
General Profile Model

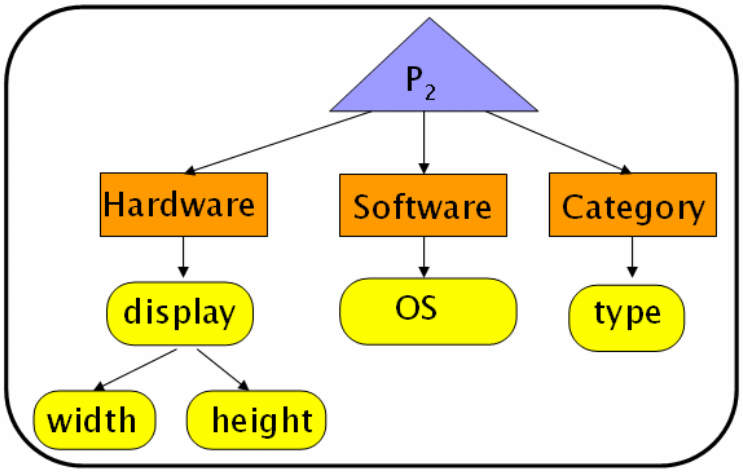


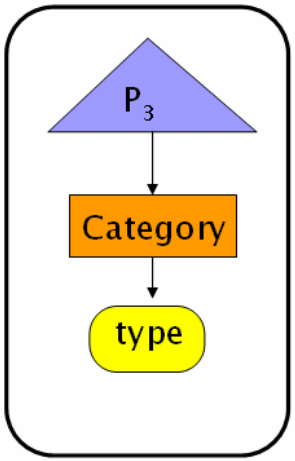
Instance of a profile

- The *instance* I_A of an attribute A is
  - pair (A, v), where v is a value (if A is **simple**)
  - a set of pairs (A, I_{A_k}), for each attribute A_k that composes A, where I_{A_k} is an instance of A_k (if A is **complex**)
- The *instance* I_D of a dimension D is
  - a set of pairs (D, I_A), where I_A is an attribute instance, for each attribute A of D.
- The *instance* of a profile P is a set of dimension instances, for each dimension of P.

Subsumption relation between profiles



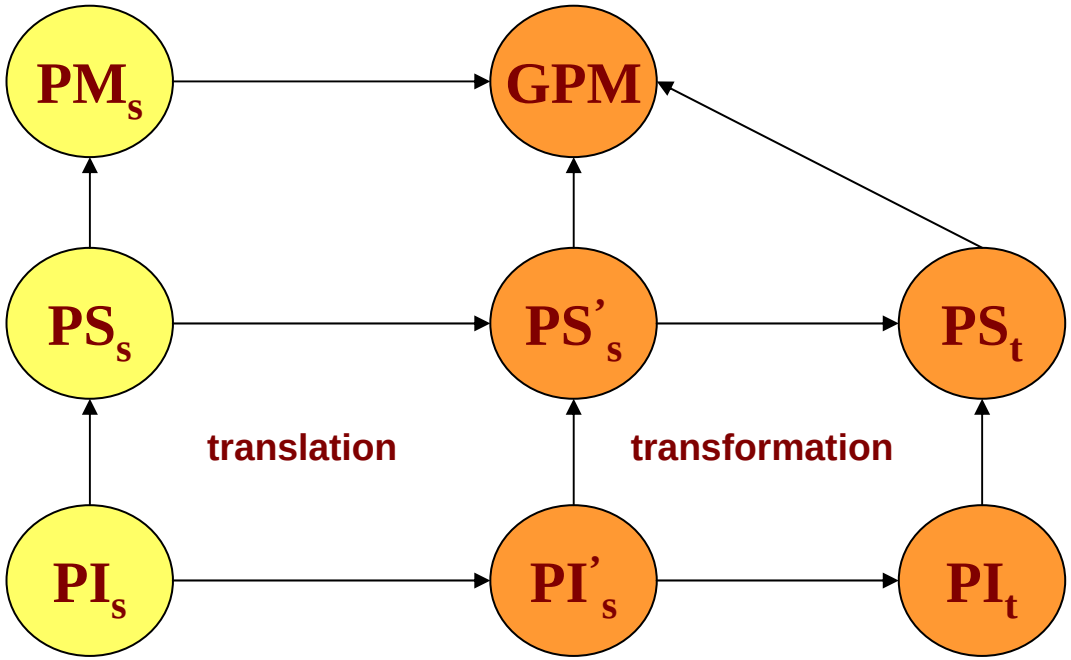




$$P_3 \lhd P_2$$

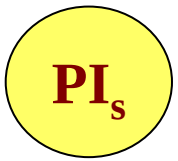
$$P_2 \lhd P_1$$

Interpretation of Profiles

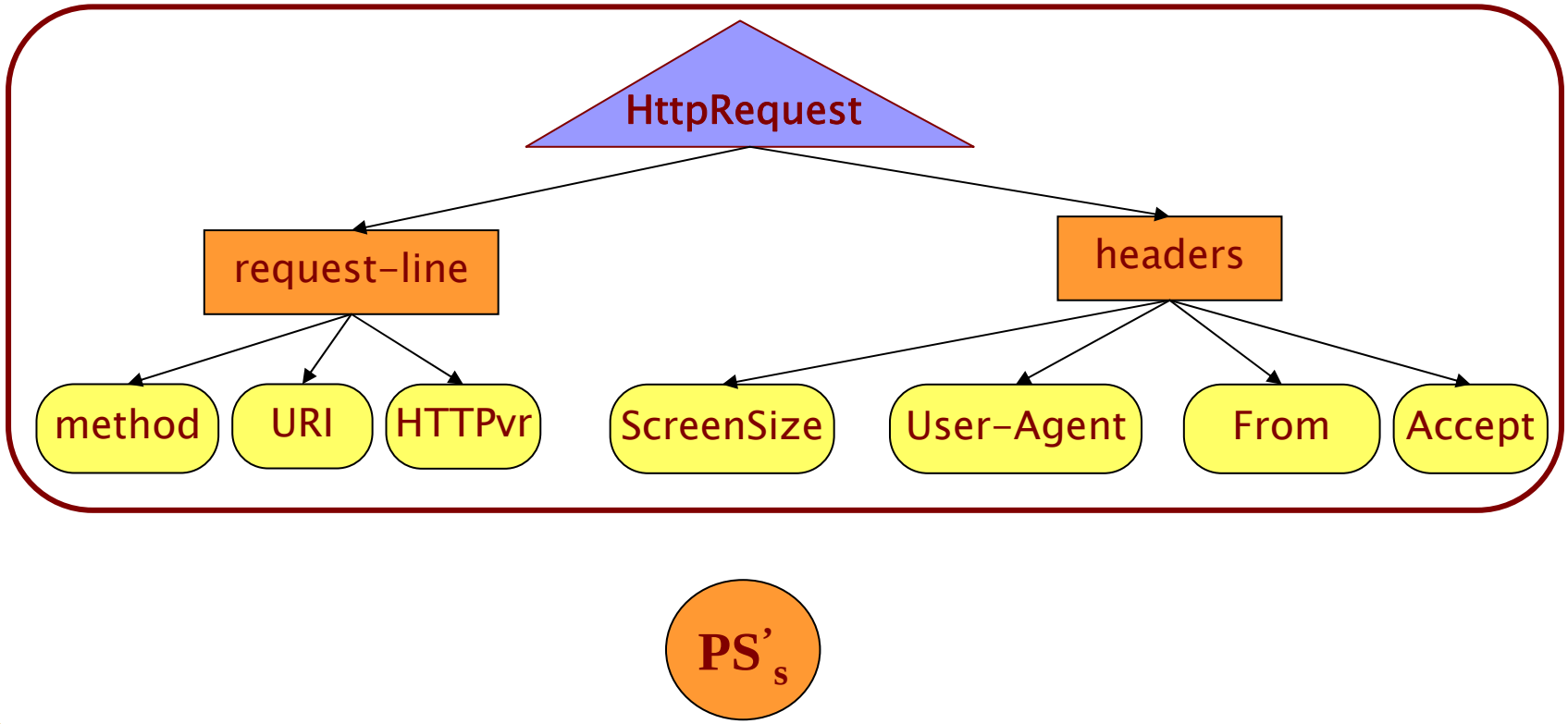


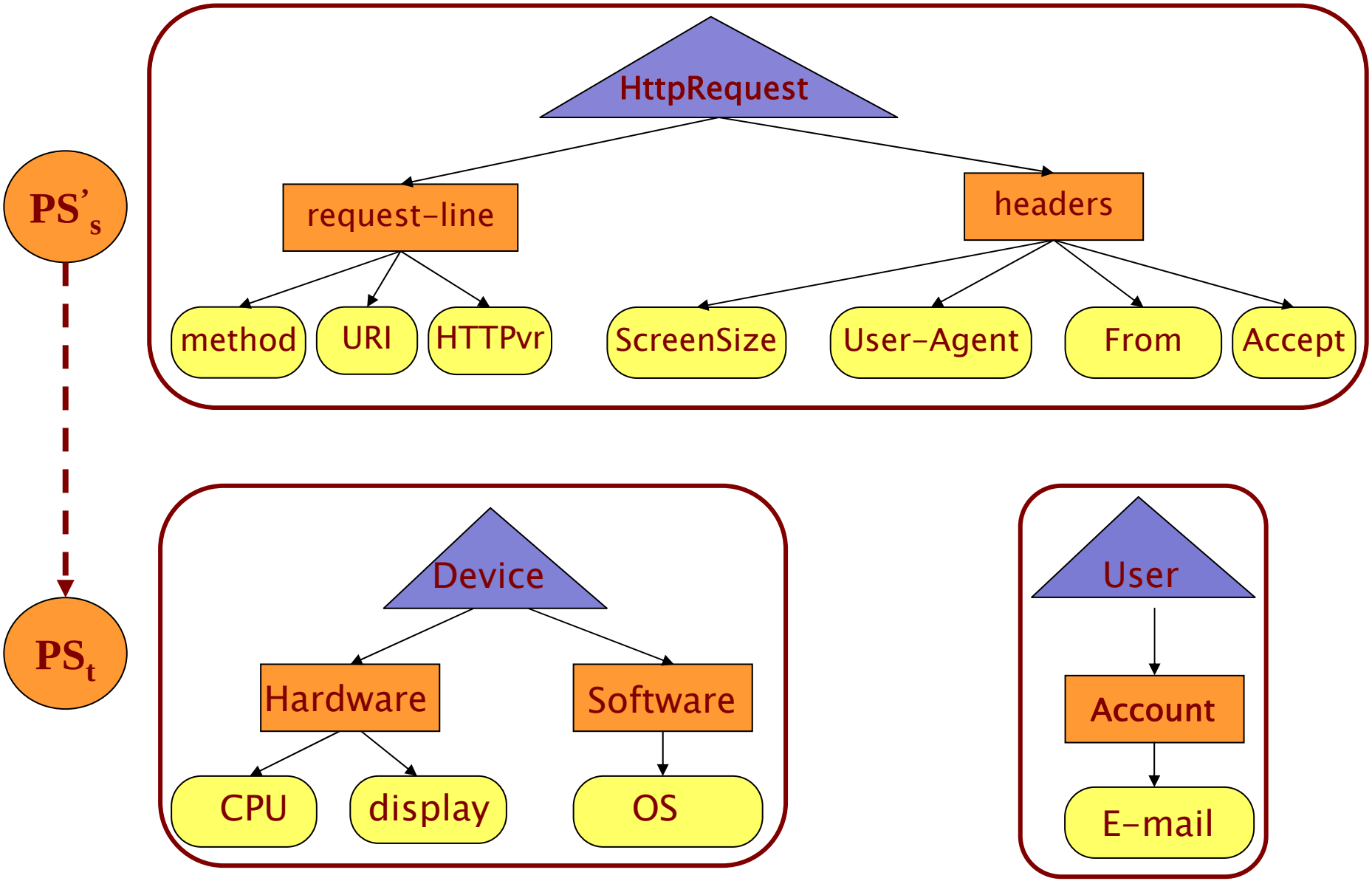
Translation of Profiles

```
POST register.jsp HTTP/1.1
Host: hi.iq
User-Agent: Mozilla/5.0
Accept: text/xml, application/xml, application/xhtml+xml,
        text/html, text/plain, video/xmng, image/png,
        image/jpeg, image/gif, text/css, */*
From: w3c@yahoo.com
ScreenSize: 15x10
```



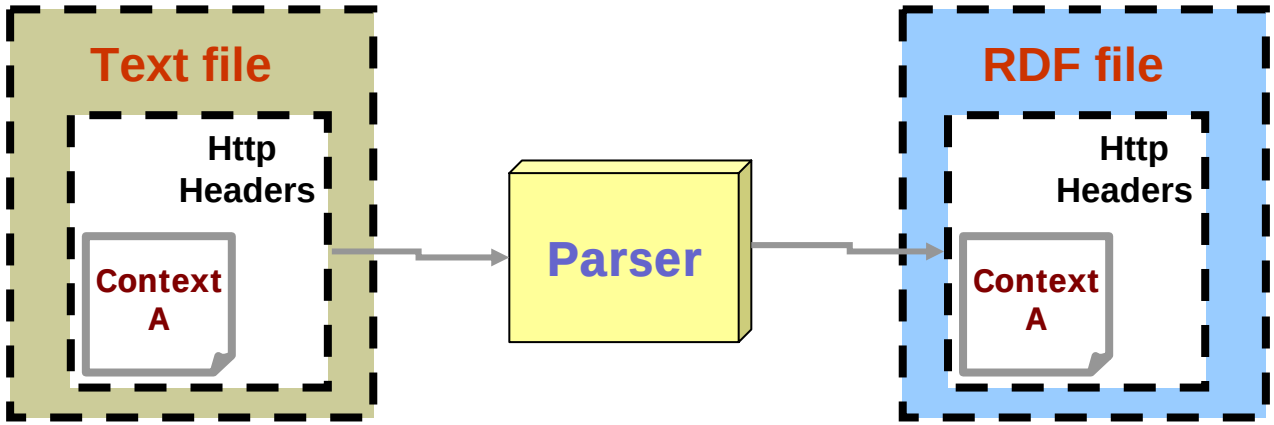
Translation of Profiles




PS's
PSt
HttpRequest
request-line
headers
method
URI
HTTPvr
ScreenSize
User-Agent
From
Accept
Device
Hardware
Software
CPU
display
OS
User
Account
E-mail

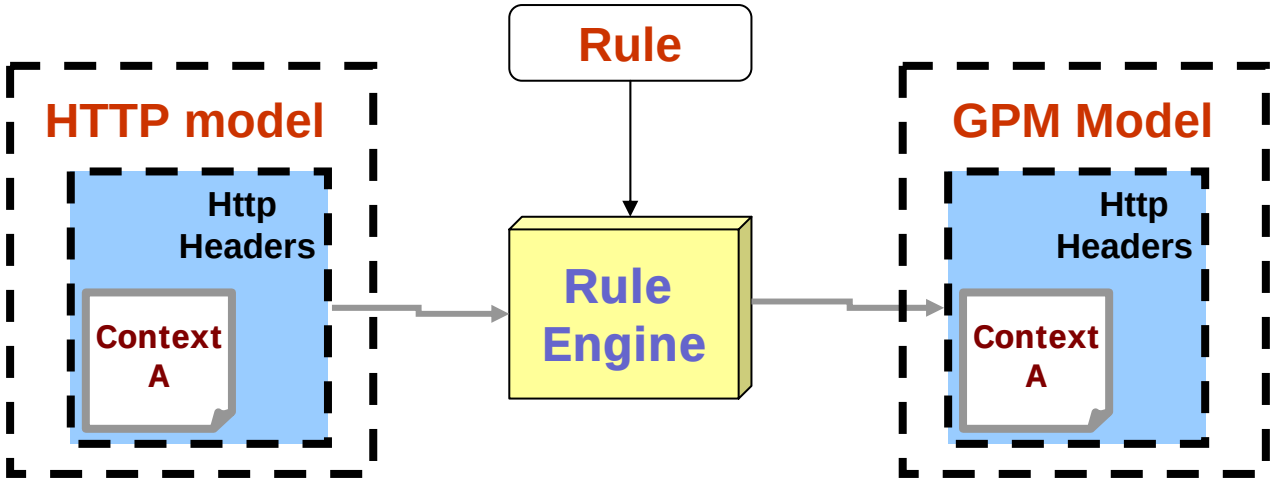
Translation step

- Given a context expressed in a model, we want to serialize it in RDF.

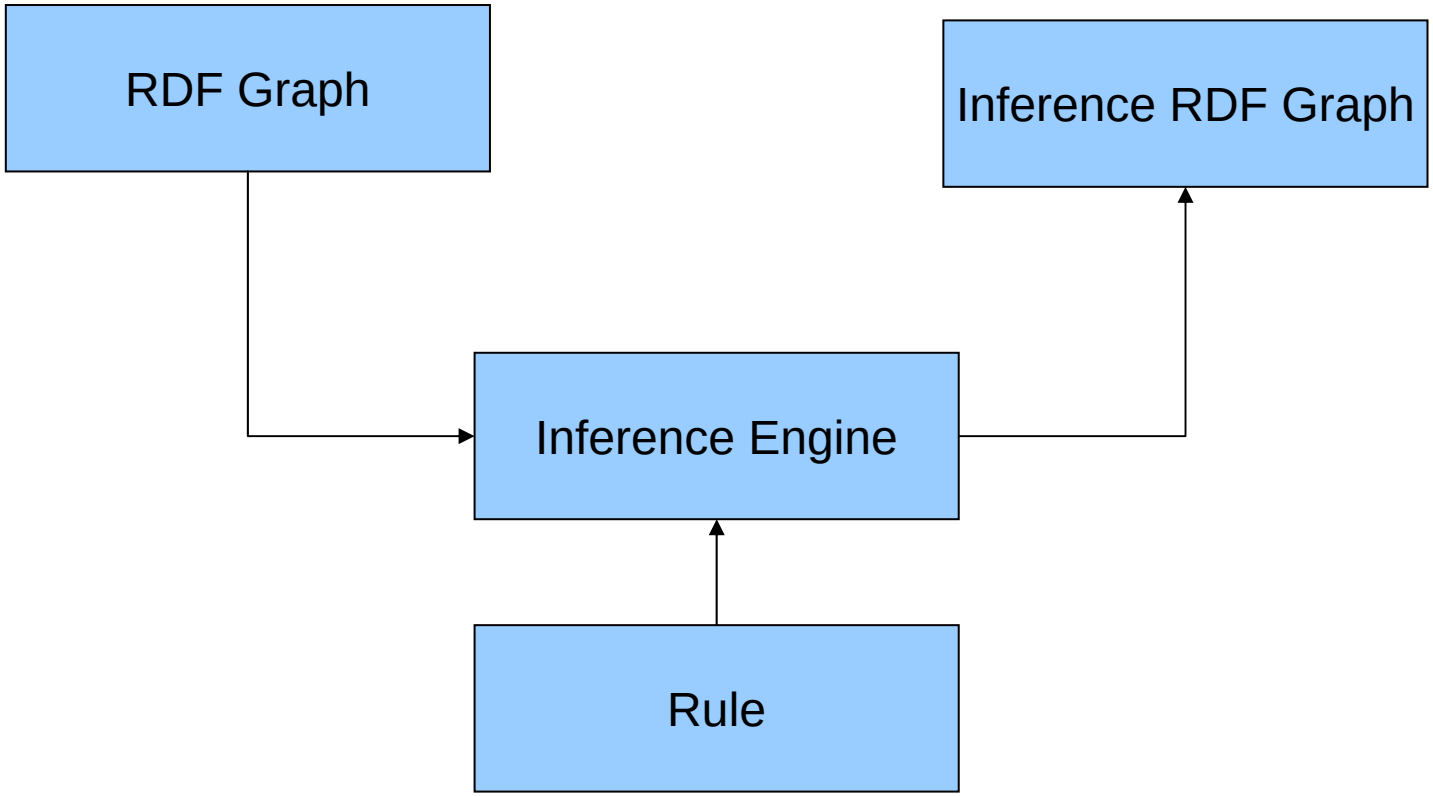


Translation step

- Then we want to build a “rule” to translate primitives of source model in metaprimitives of our General Profile Model.



Jena: a Java Framework to manage RDF files
http://jena.sourceforge.net/



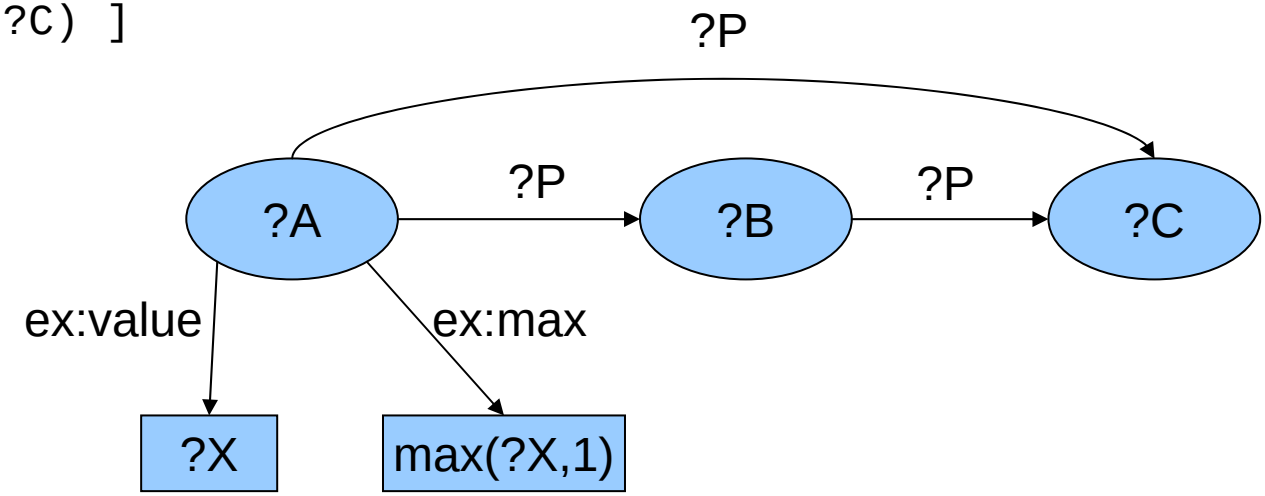
Jena: a Java Framework to manage RDF files
http://jena.sourceforge.net/

```
Rule      :=   bare-rule .
          or   [ bare-rule ]
               or   [ ruleName : bare-rule ]

bare-rule :=   term, ... term -> hterm, ... hterm    // forward rule
          or   term, ... term <- term, ... term    // backward rule

hterm     :=   term
          or   [ bare-rule ]

term      :=   (node, node, node)            // triple pattern
          or   (node, node, functor)         // extended triple pattern
          or   builtin(node, ... node)       // invoke procedural primitive

functor   :=   functorName(node, ... node)   // structured literal

node      :=   uri-ref                          // e.g. http://foo.com/eg
          or   prefix:localname                 // e.g. rdf:type
          or   ?varname                      // variable
          or   'a literal'                   // a plain string literal
          or   'lex'^^typeURI                // a typed literal, xsd:* type names supported
          or   number                        // e.g. 42 or 25.5
```

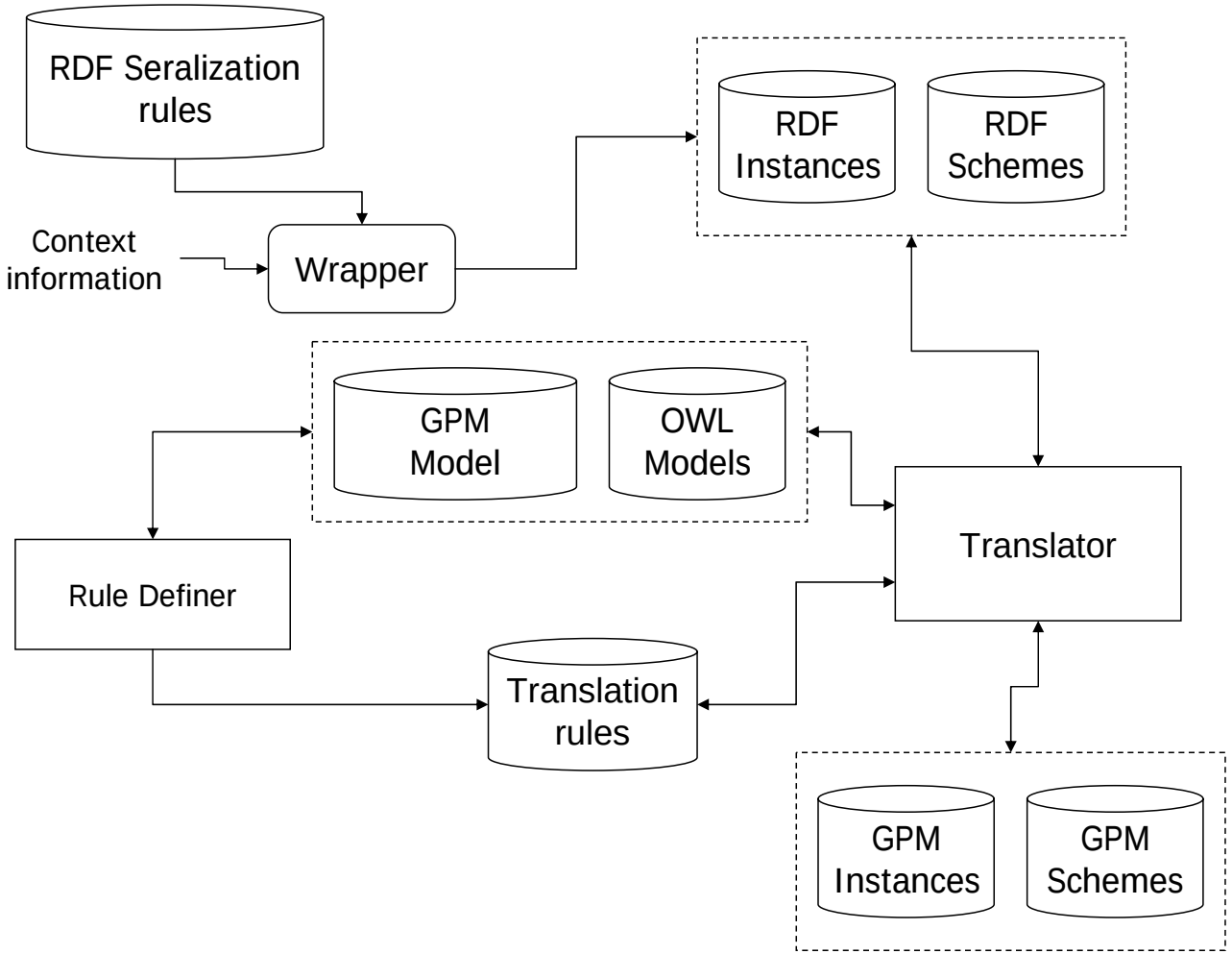
Jena: a Java Framework to manage RDF files
http://jena.sourceforge.net/

Es.

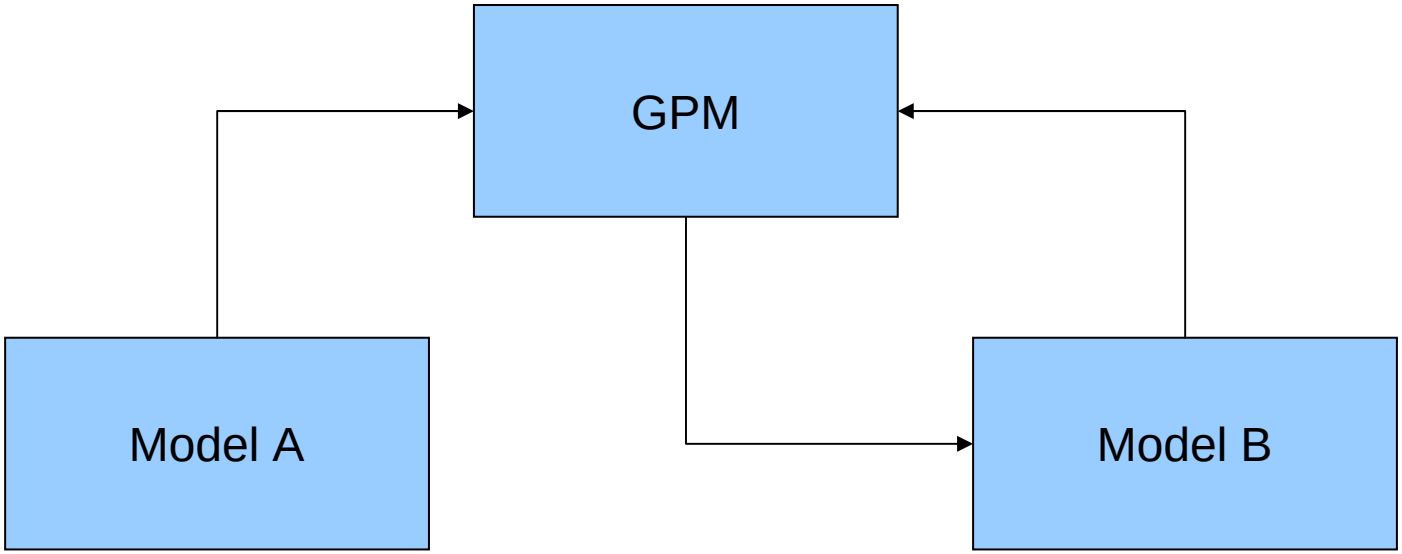
```
[max1:
 (?A ex:value ?X), (?A ex:max max(?X, 1)), (?A ?P ?B), (?B ?P ?C)
    ->
 (?A ?P ?C) ]
```



A tool for translation



A tool for translation



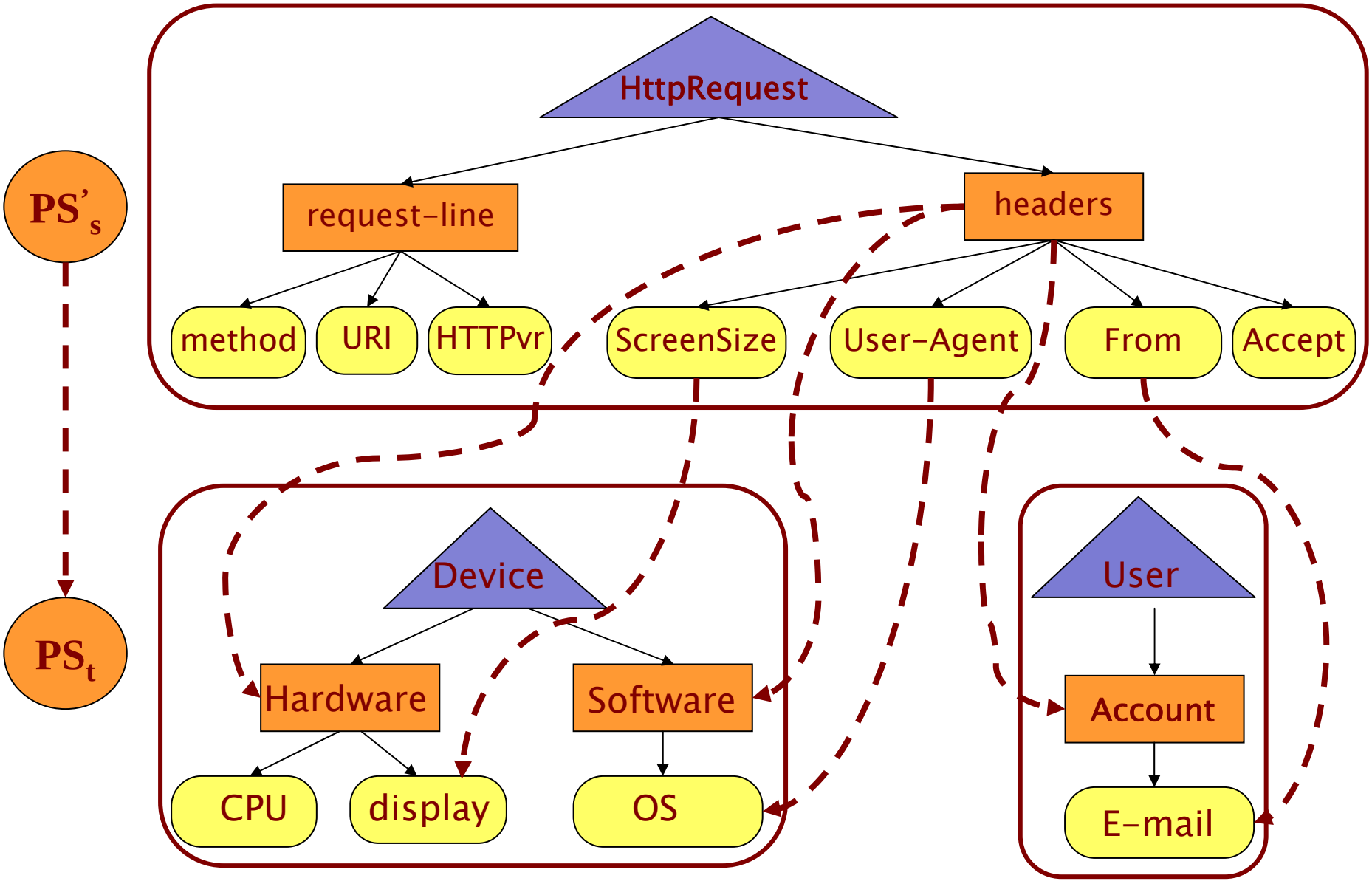
Seminario

- Documentazione sul modello RDF, OWL, Jena e Inference Rules
- Scelta di due modelli: individuazione di primitive
- Costruzione del Parser RDF per i due modelli scelti (definizione della grammatica)
- Costruzione delle regole (Jena Rule) per tradurre i costrutti dei due modelli nei costrutti del GPM

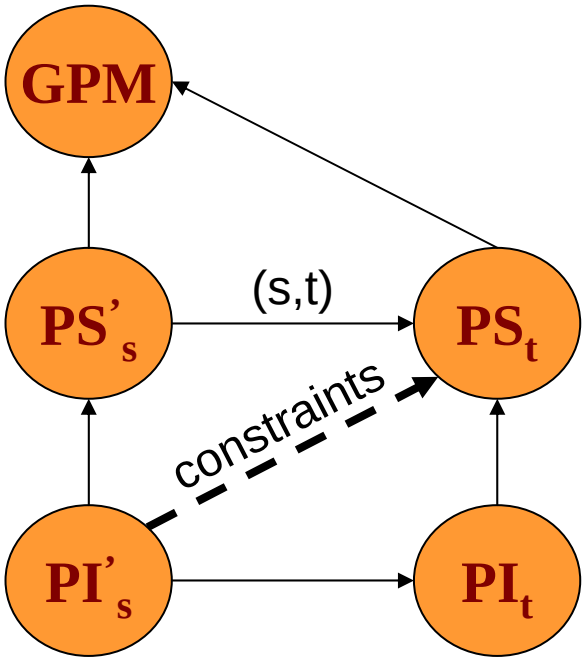
Transformation step

- The second step is more involved and is based on a *mapping* between *profiles* that drives the profile instance PI_t

Definition (Mapping of profiles) *Given two profiles* ***P1*** *and* ***P2*** *in* ***GPM*** *model, a* ***mapping M*** *is a set of pairs* (***S***, ***t***) *where S is a set of one or more components (dimension or attribute) of P1 and* ***t*** *is a component of P2.*


PS'$_s$
PS$_t$
HttpRequest
request-line
headers
method
URI
HTTPvr
ScreenSize
User-Agent
From
Accept
Device
Hardware
Software
CPU
display
OS
User
Account
E-mail

Transformation step



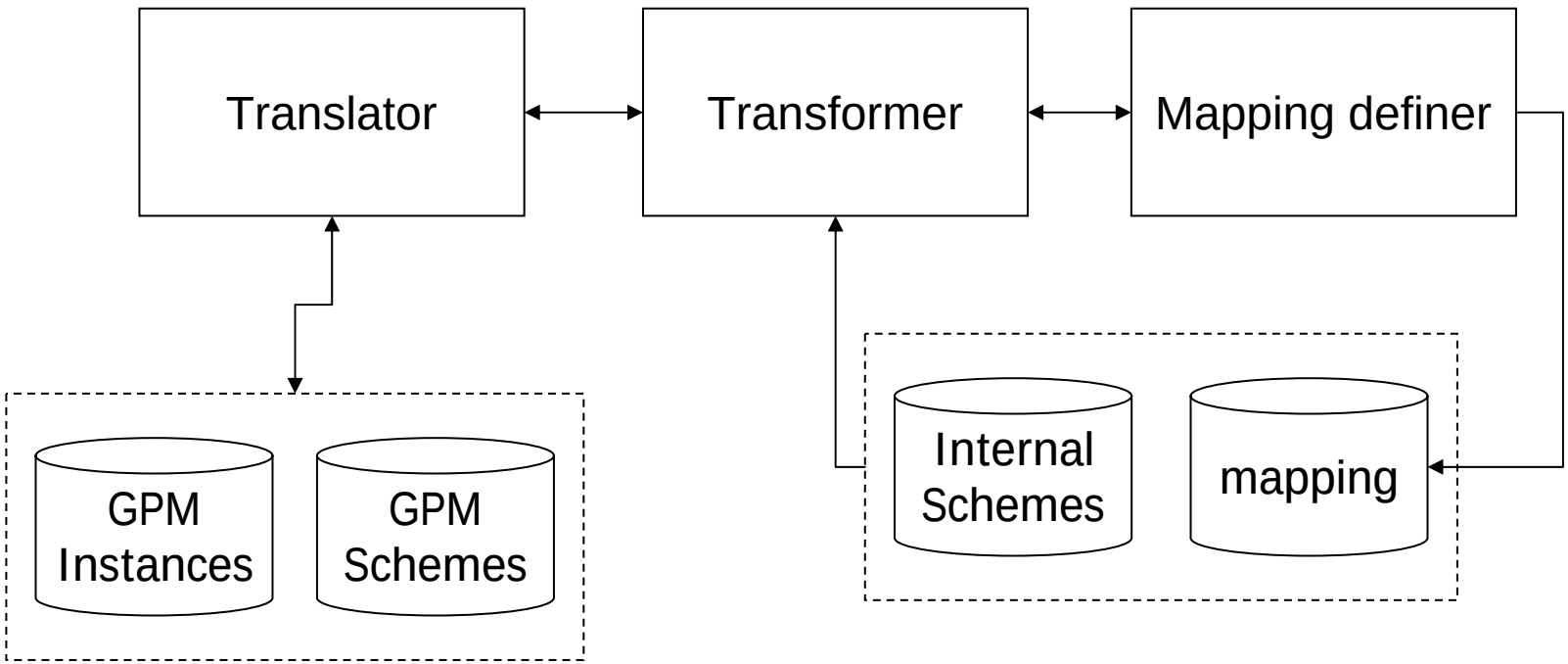
- Input: PI_s, PS_s, PS_t, M
- Output: PI_t
- We have to calculate the instance of t for each (S,t) in M
- Instance(t) = f((S,t),costr)
- f is an operator to calculate the instance of t from S

Transformation step

Algorithm 1
Input: PI_s, PS_s, M, PS_t
Output: PI_t
(1) ***begin***
(2) $\forall$ *dimension* $D \in PS_t$
(3) *if* $\exists m \in M / m = (D',D)$
(4) $\forall$ *attribute A of* $D \in PS_t$
(5) *if* $\exists m' \in M / m' = (A',A)$
(6) $PI_t \cup (D,VALUE(A,m'))$;
(7) ***end***

Procedure 1
(1a) I_A *VALUE(attribute A, pair m) {*
(2a) ***if*** *(A is simple)* ***return*** *(A,value of m.s);*
(3a) ***else*** *{*
(4a) I_A $vA = \varnothing$;
(5α) $\forall$ *attribute A' of A {*
(6a) *if* $\exists m' \in M / m' = (A'',A')$
(7a) $vA \cup (A', VALUE(A',m'))$;
(8a) *}*
(9a) ***return*** *vA;*
(11a) *}*
(12a) *}*

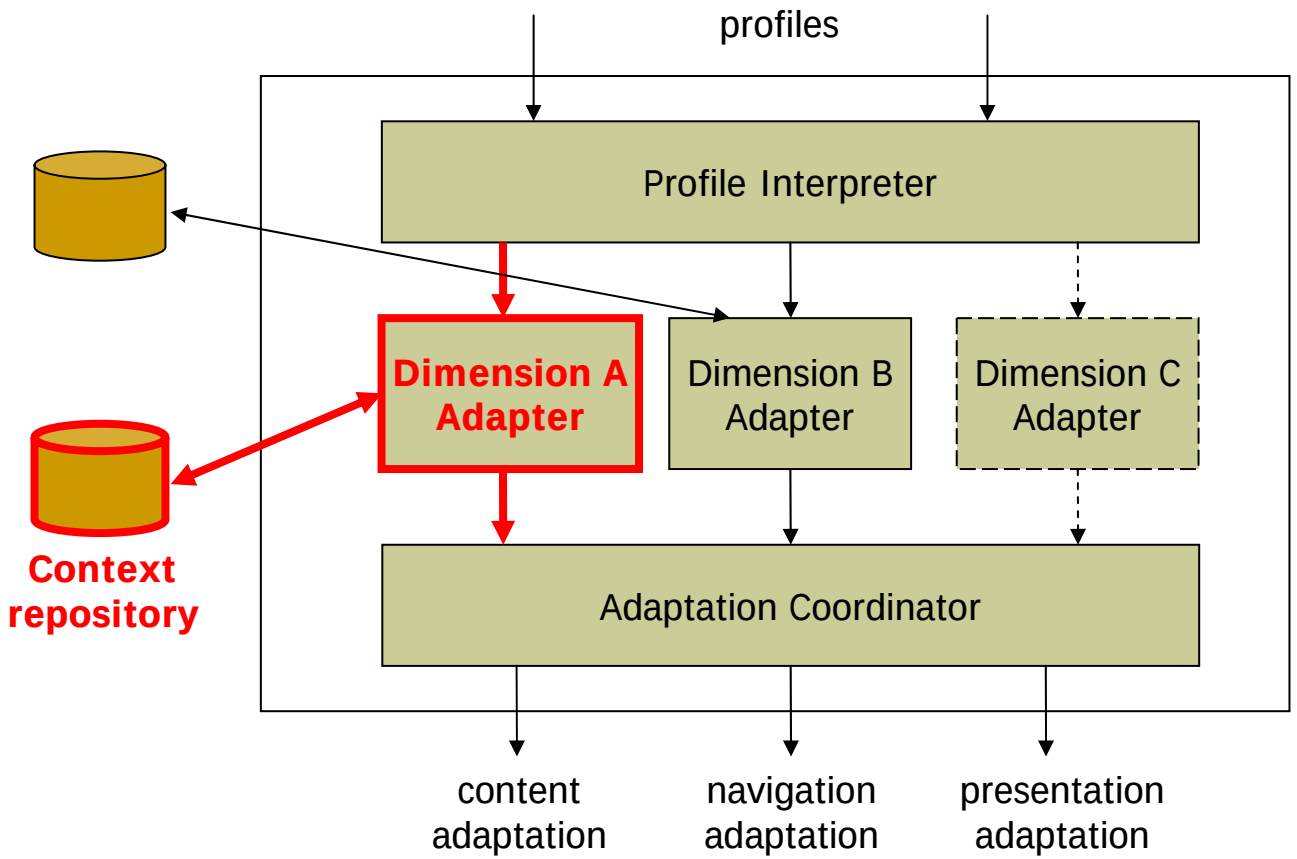
A tool for transformation



Seminario

- Documentazione sul modello RDF, OWL, SPARQL, Jena e Inference Rules
- Studio di tecniche per il mapping semi-automatico tra schemi
- Scelta di uno schema sorgente
- Scrittura dei mapping (Jena Rule o SPARQL) tra lo schema sorgente e lo schema target (fornito)

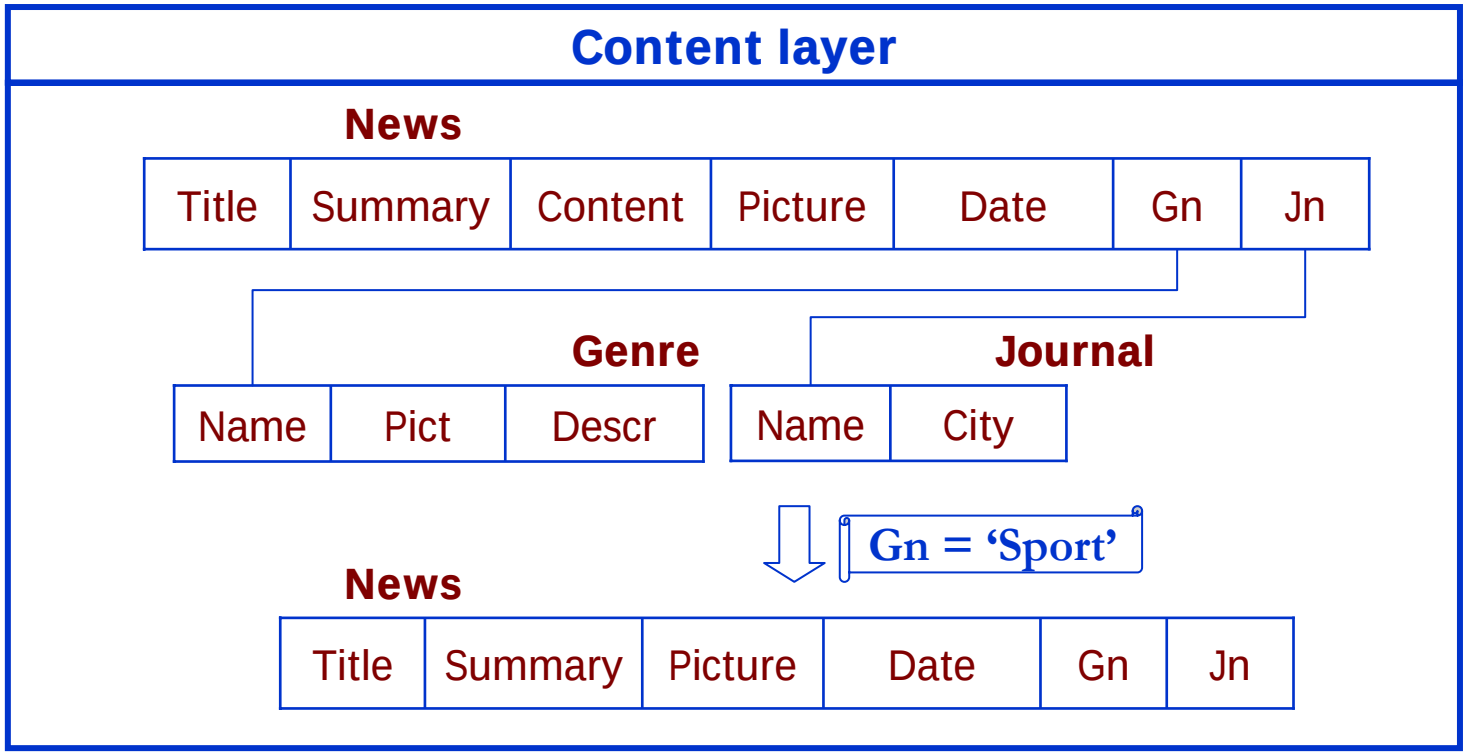
Automatic Generation of Adaptation



Configuration

- **C = {q,h,p}**
  - q is a **query** expressed in standard **relational calculus**
  - h is an **abstract hypertext** expressed in **WebML** model
  - p is **presentation specification** expressed in terms of **Logical style sheet**

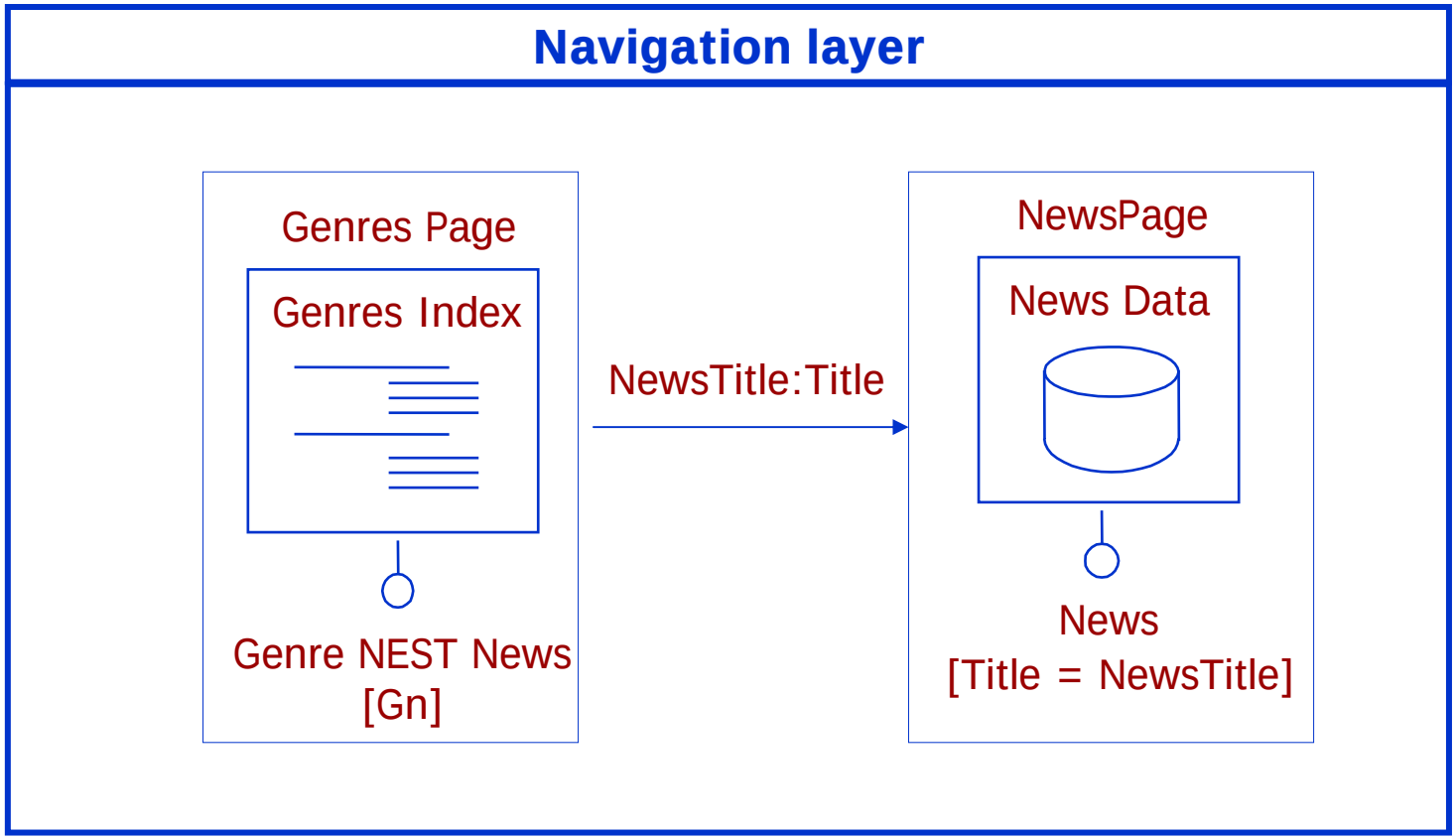
Content layer



Content layer

$$\{Title : x_1, Summary : x_2, Picture : x_3, Date : x_4, Gn : x_5, Jn : x_6 \mid$$
$$News(T : x_1, S : x_2, C : y_1, P : x_3, D : x_4, G : x_5, J : x_6),$$
$$Genre(Name : x_5, Pict : y_2, Descr : y_3)$$
$$Journal(Name : x_6, City : y_4), x_5 = 'Sport'\}$$

Navigation layer



Navigation layer

IndexUnit *GenresIndex* **hierarchical**

(**source** *Genre*; **attributes** *Name*;

orderby *Name*; **NEST** *News* **selector** *GN=Name*;

attributes *Title, Picture, Date*; **orderby** *Title*;)

DataUnit *NewsData*

(**source** *News*; **selector** *NewsTitle=Title*;

attributes *Title*, *Content, Date*)

link *GenreToNews*

(**from** *GenresIndex* **to** *NewsData*;

parameters *NewsTitle = Title*)

Presentation Layer

Text	Link	Image	Video
Font: Arial **Size**: 10pt **Style**: Normal **Color**: Black **Border:** 0pt	**Note**: FALSE **Font**: 10pt **Size**: Normal **Style**: Underline **Color**: Blue	**Resolution**: jpeg **Size**: 176 x 208 **Border**: 0pt **Color**: TRUE **Alignment**: left	**Resolution**: mpeg **Size**: 240 x 320 **Border**: 2pt **Color**: FALSE **Alignment**: center

```
{
  Text {Font: Arial; Size: 10pt; … ; Border: 0pt},
  Link {Note: FALSE; … ; Color: Blue},
  … ,
  Video {Resolution: mpeg; … }
}
```

Configuration is an abstract entity

- a **configuration** can be implemented in several ways and with different syntaxes
  - SQL (on content layer)
  - XHTML (on navigation layer)
  - CSS or XSL (on presentation layer)

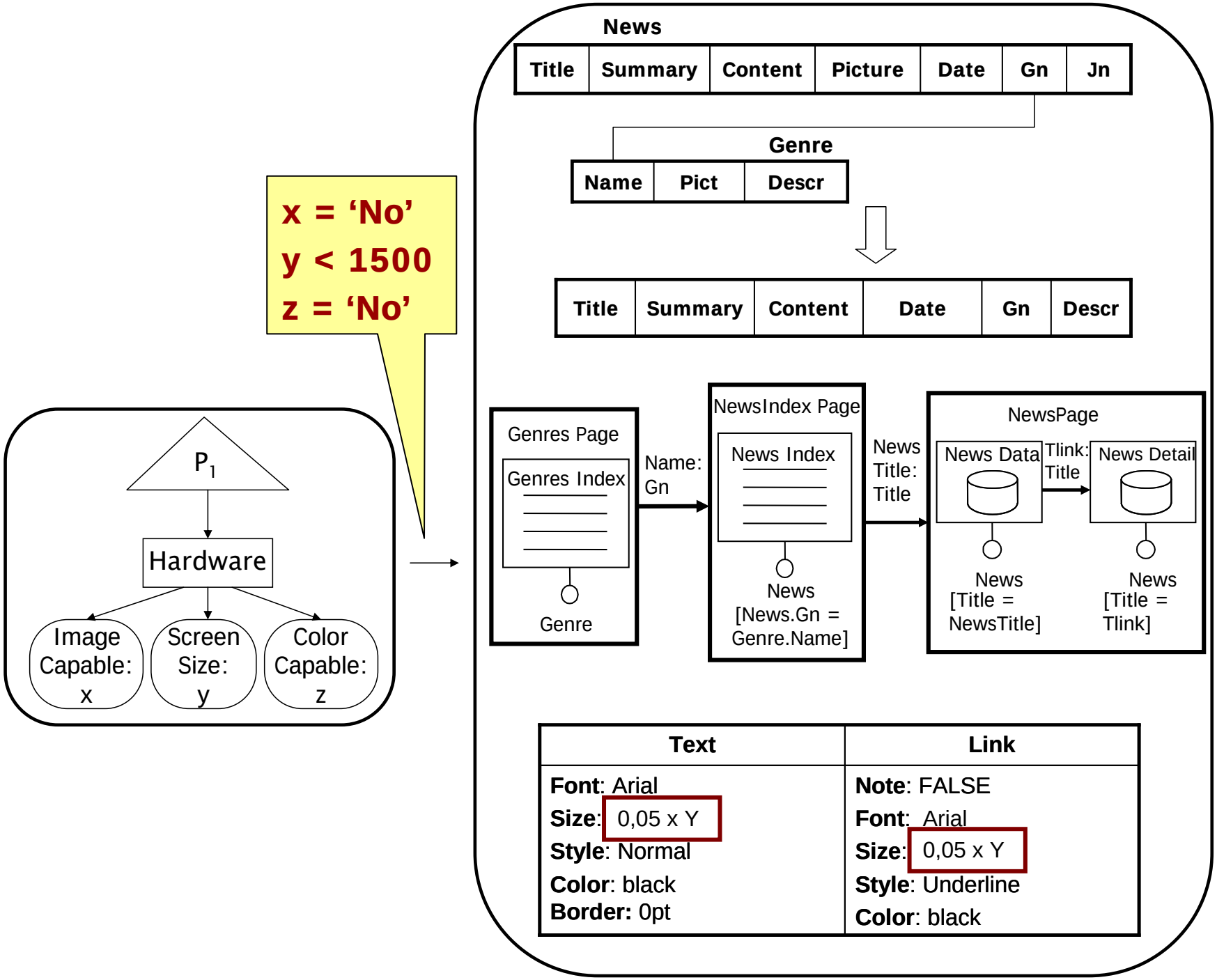
Adaptation Rule

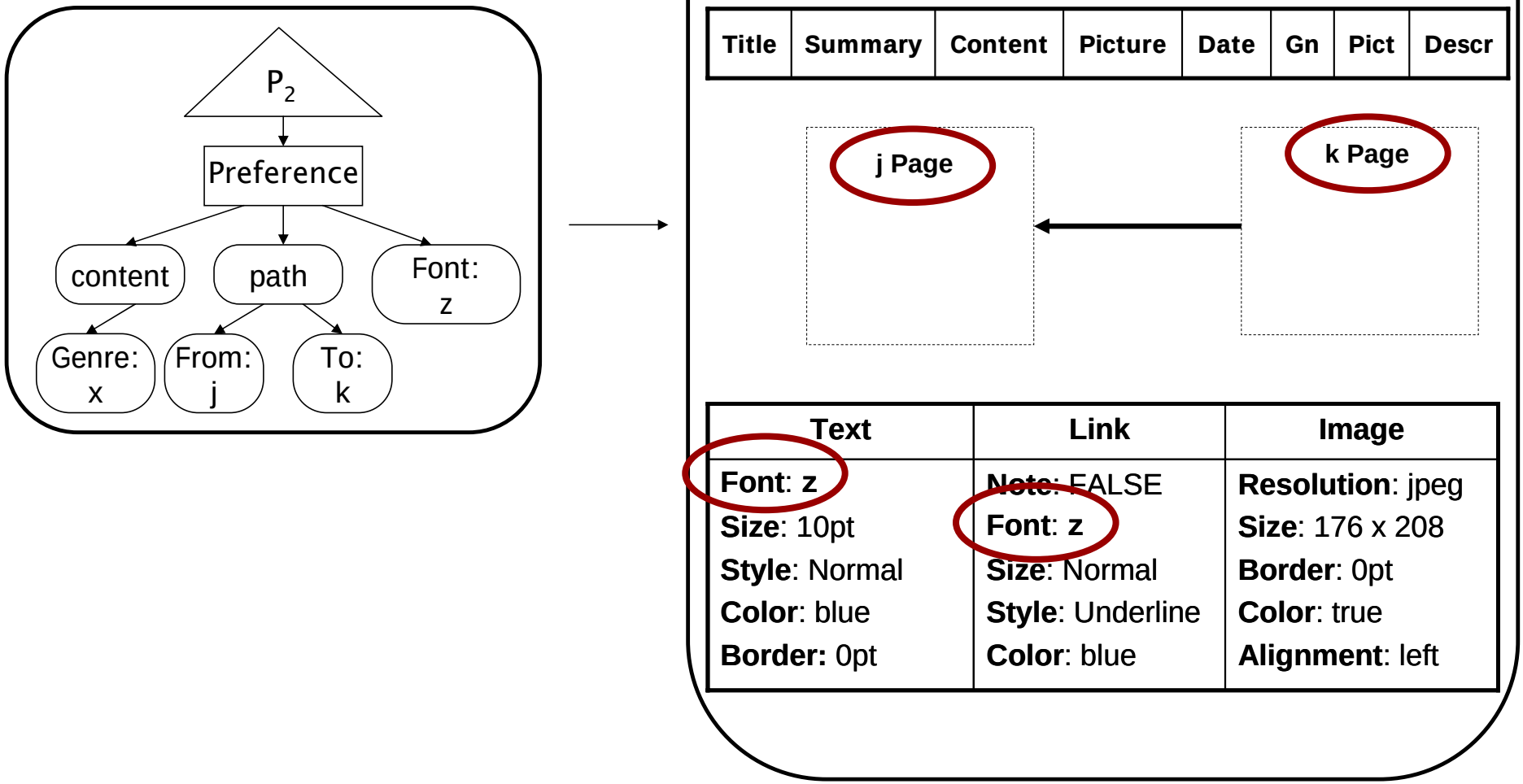
- a **rule** looks as $P_r : C_d \rightarrow C_f$

  - ***P_r*** is a parametric profile, that is, a profile in which parameters can appear in place of values,

  - ***C_d*** is a condition, made of a a conjunction of atoms of the form $A = c$ or $A = B$ where A and B are parameters occurring in P_r and c is a constant value, and

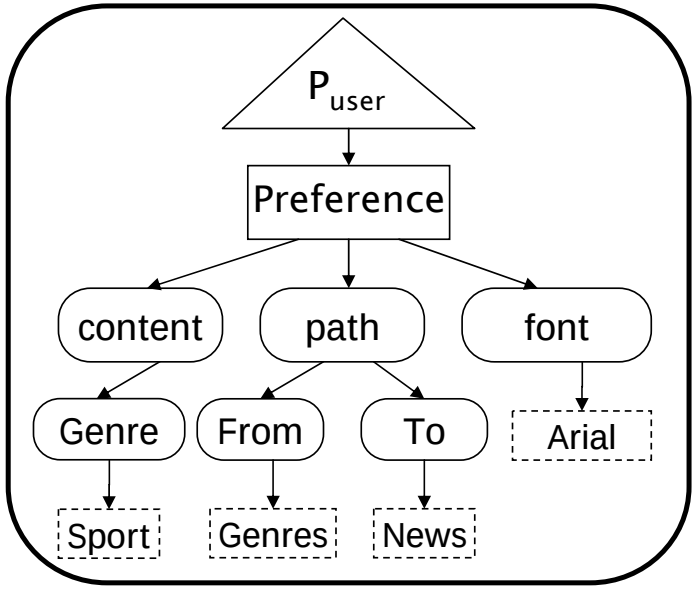
  - ***C_f*** is a parametric configuration in which parameters occurring in P_r can appear in place of values.

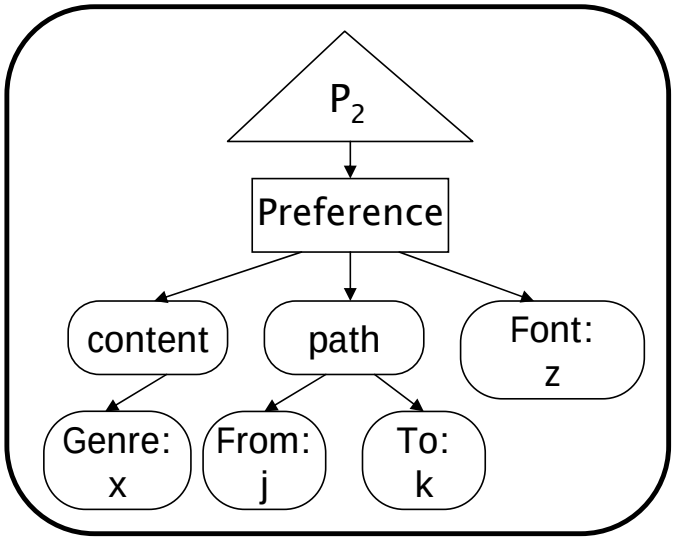

News
Title
Summary
Content
Picture
Date
Gn
Jn
Genre
Name
Pict
Descr
Title
Summary
Content
Date
Gn
Descr
x = 'No'
y < 1500
z = 'No'
P1
Hardware
Image Capable: x
Screen Size: y
Color Capable: z
Genres Page
Genres Index
Genre
Name: Gn
NewsIndex Page
News Index
News [News.Gn = Genre.Name]
News Title: Title
NewsPage
News Data
Tlink: Title
News Detail
News [Title = NewsTitle]
News [Title = Tlink]
Text
Font: Arial
Size: 0,05 x Y
Style: Normal
Color: black
Border: 0pt
Link
Note: FALSE
Font: Arial
Size: 0,05 x Y
Style: Underline
Color: black

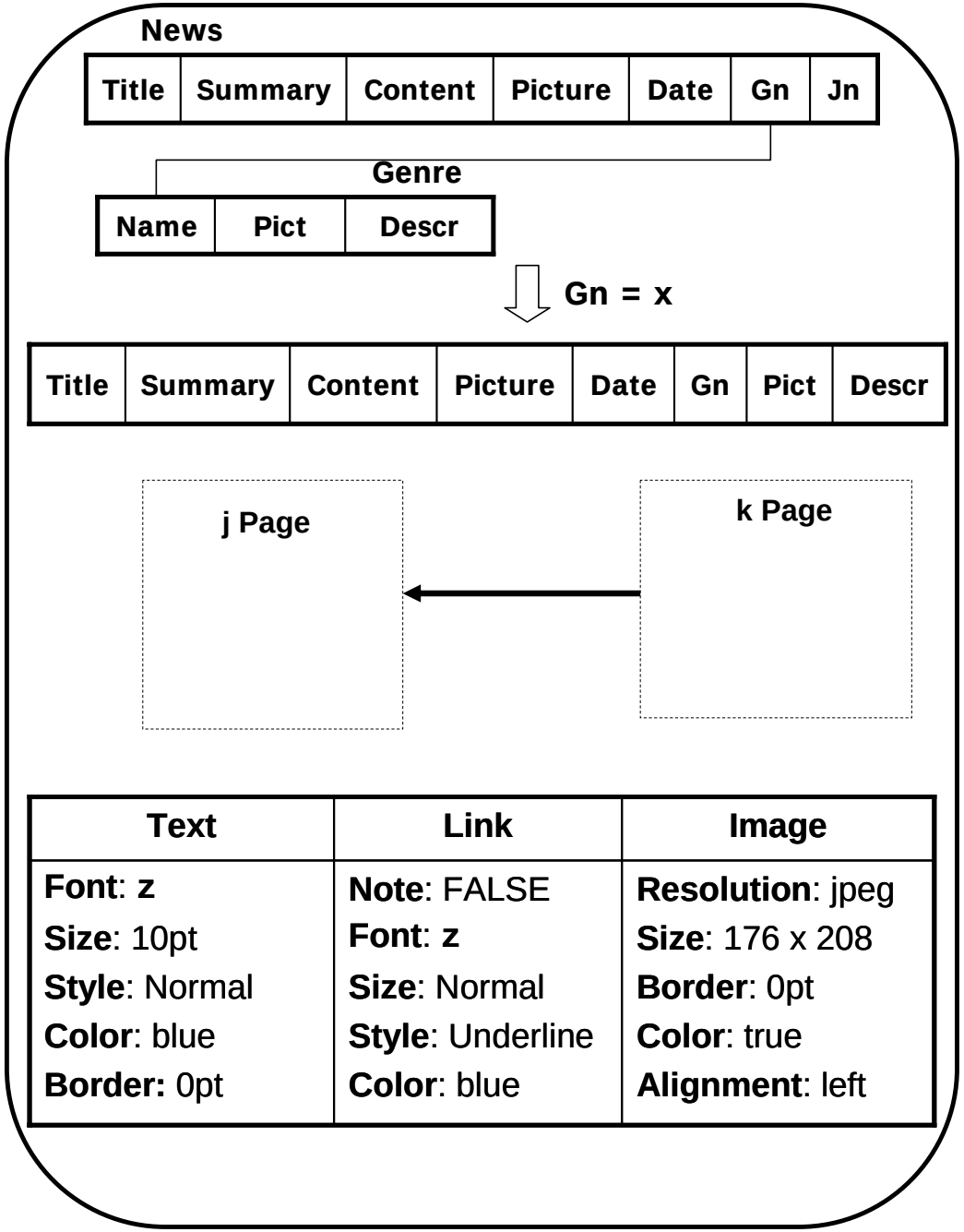

P2
Preference
content
path
Font: z
Genre: x
From: j
To: k
News
Title
Summary
Content
Picture
Date
Gn
Jn
Genre
Name
Pict
Descr
Gn = x
Title
Summary
Content
Picture
Date
Gn
Pict
Descr
j Page
k Page
Text
Font: z
Size: 10pt
Style: Normal
Color: blue
Border: 0pt
Link
Note: FALSE
Font: z
Size: Normal
Style: Underline
Color: blue
Image
Resolution: jpeg
Size: 176 x 208
Border: 0pt
Color: true
Alignment: left

Activation of a rule

- A profile P ***activates*** a rule $P_r : C_d \rightarrow C_f$ if $P_r \triangleleft P$.
- Hence, a profile P can activate a rule for a profile that is more general than P.







Text	Link	Image
Font: z **Size**: 10pt **Style**: Normal **Color**: blue **Border:** 0pt	**Note**: FALSE **Font**: z **Size**: Normal **Style**: Underline **Color**: blue	**Resolution**: jpeg **Size**: 176 x 208 **Border**: 0pt **Color**: true **Alignment**: left

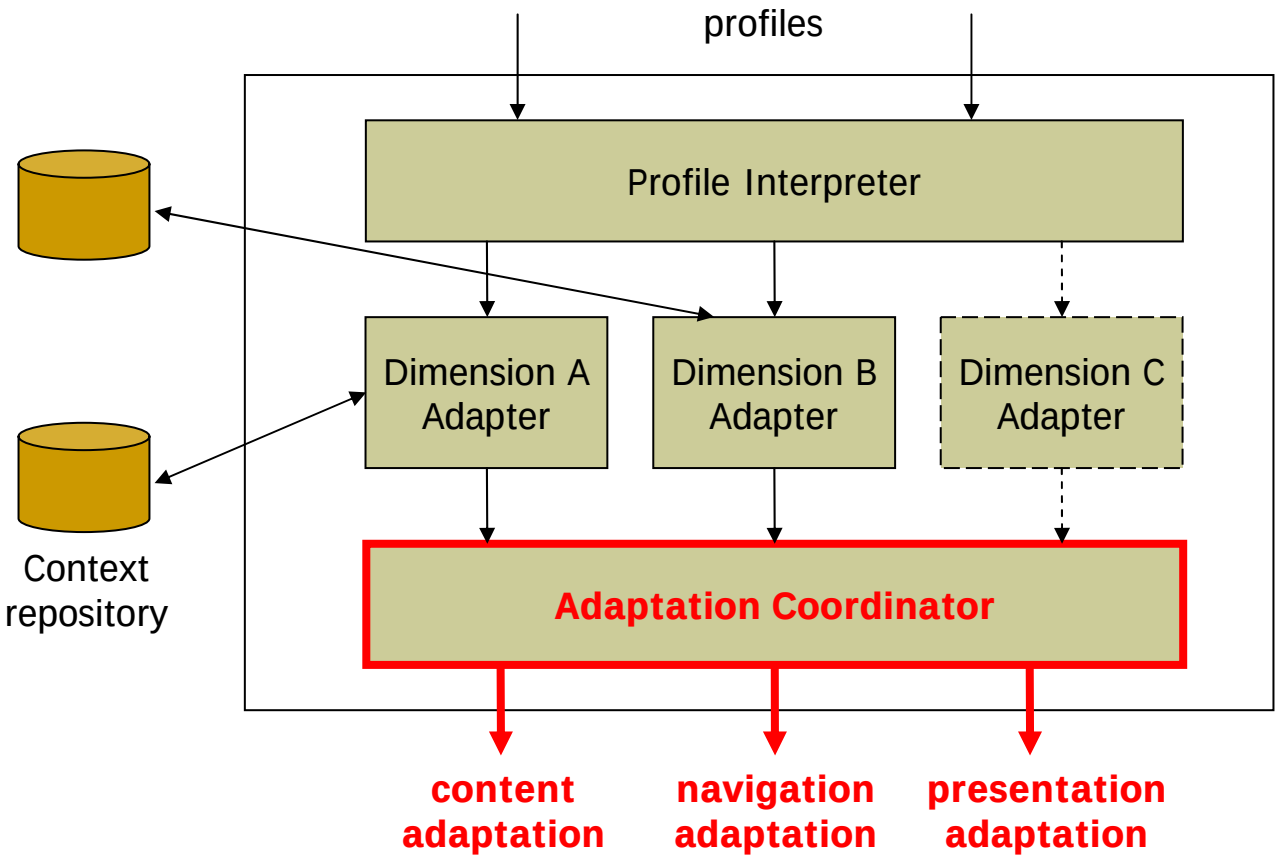
Seminario

- Documentazione sul modello RDF, OWL, SPARQL, Jena e Inference Rules
- Studio sull'adattamento dei vari livelli di un WIS
- Scrittura delle regole di adattamento
- Parser per serializzare le regole (RDF model)

Seminario

- Documentazione sul modello RDF, OWL, SPARQL, Jena e Inference Rules
- Scrittura di regole di adattamento sui tre livelli
- Studio di un processo di attivazione delle regole che sfrutti la relazione di sussunzione

Coordination of adaptation

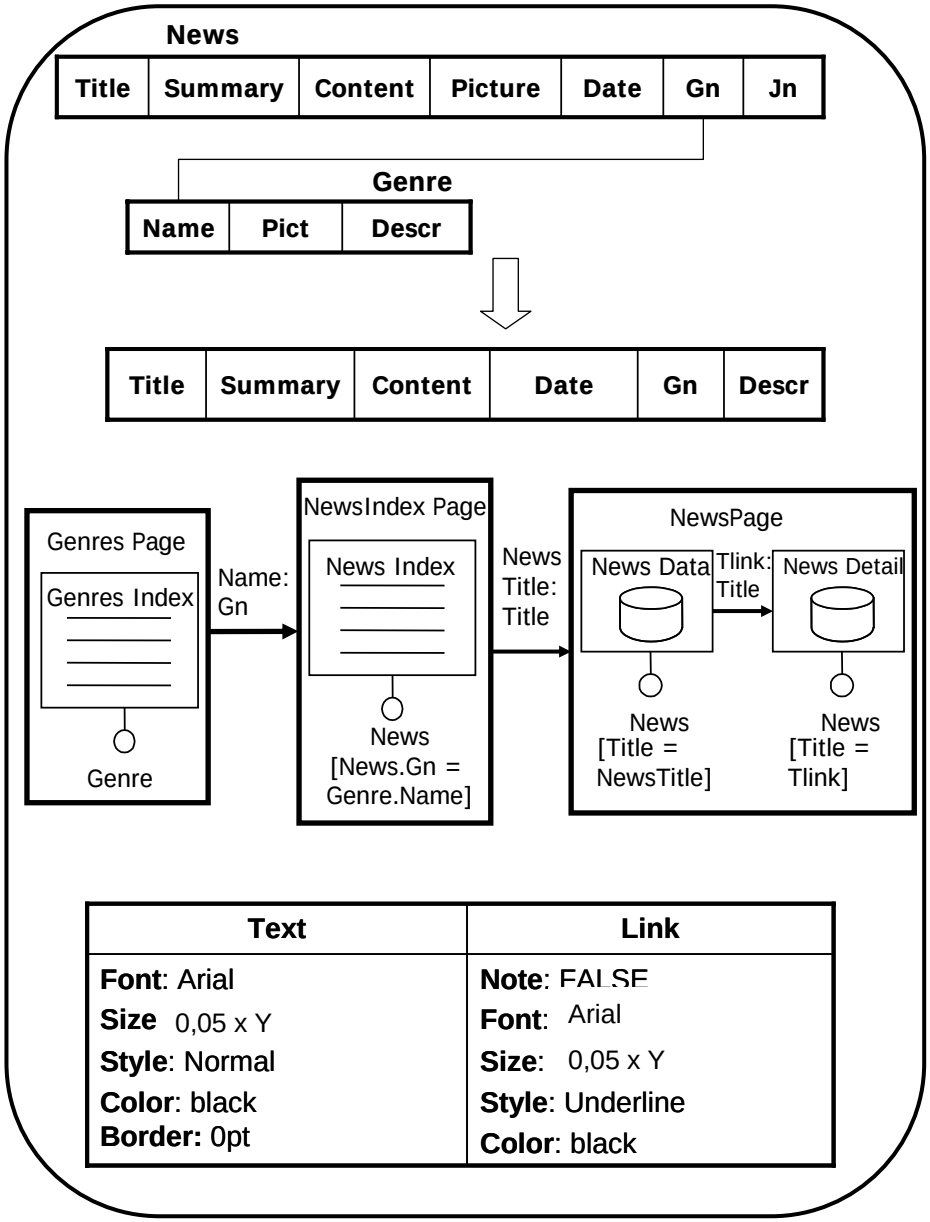


Composition of configurations (⊕)

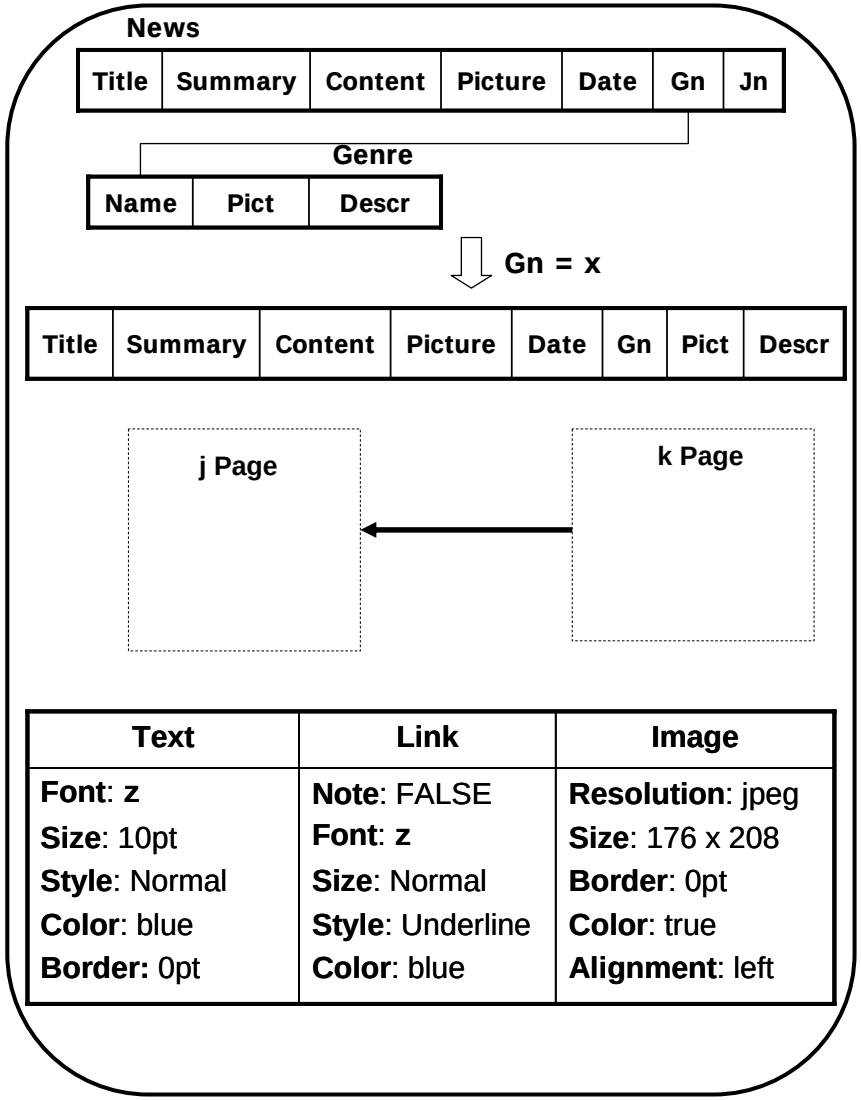
Given a pair configurations $C_1(q_1, h_1, p_1)$ and $C_2(q_2, h_2, p_2)$:

$C_1 \oplus C_2$ is a configuration $C(q, h, p)$ is defined as follows:

- $q = q_1 \circ q_2$, that is, q is obtained as the composition of q_1 followed by q_2;
- h is obtained by merging h_1 and h_2: if some conflict arises, the choices of h_1 are preferred to those of h_2; and
- $p(w_i) = p_1(w_i)$ if w_i is a WOT occurring in p_1 and $p(w_i) = p_2(w_i)$ otherwise (that is, if w_i is a WOT occurring only in p_2).

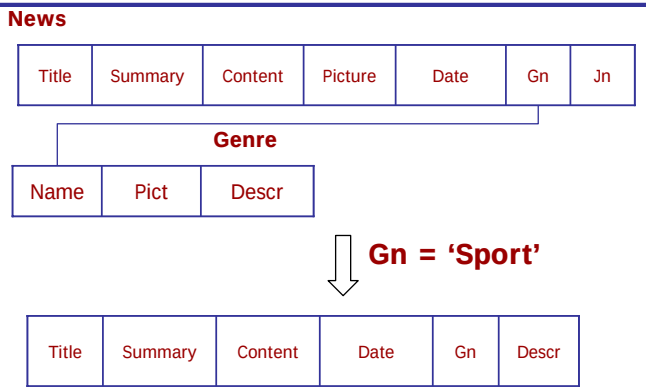


Text	Link
Font: Arial **Size** 0,05 x Y **Style**: Normal **Color**: black **Border:** 0pt	**Note**: FALSE **Font**: Arial **Size**: 0,05 x Y **Style**: Underline **Color**: black

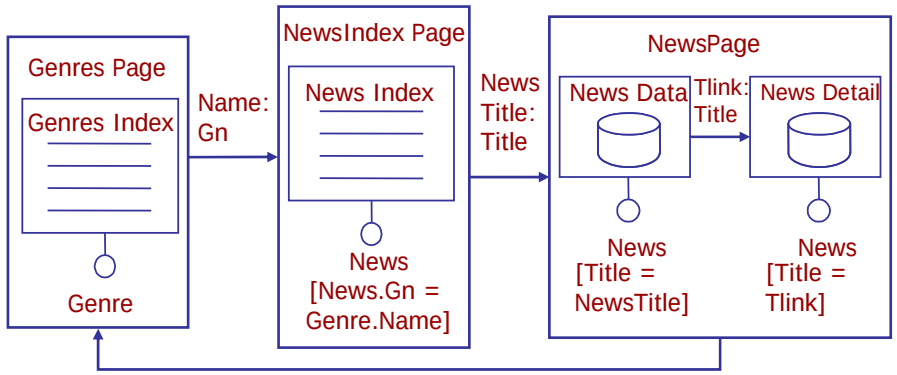


Text	Link	Image
Font: z **Size**: 10pt **Style**: Normal **Color**: blue **Border:** 0pt	**Note**: FALSE **Font**: z **Size**: Normal **Style**: Underline **Color**: blue	**Resolution**: jpeg **Size**: 176 x 208 **Border**: 0pt **Color**: true **Alignment**: left

Content layer



Navigation layer



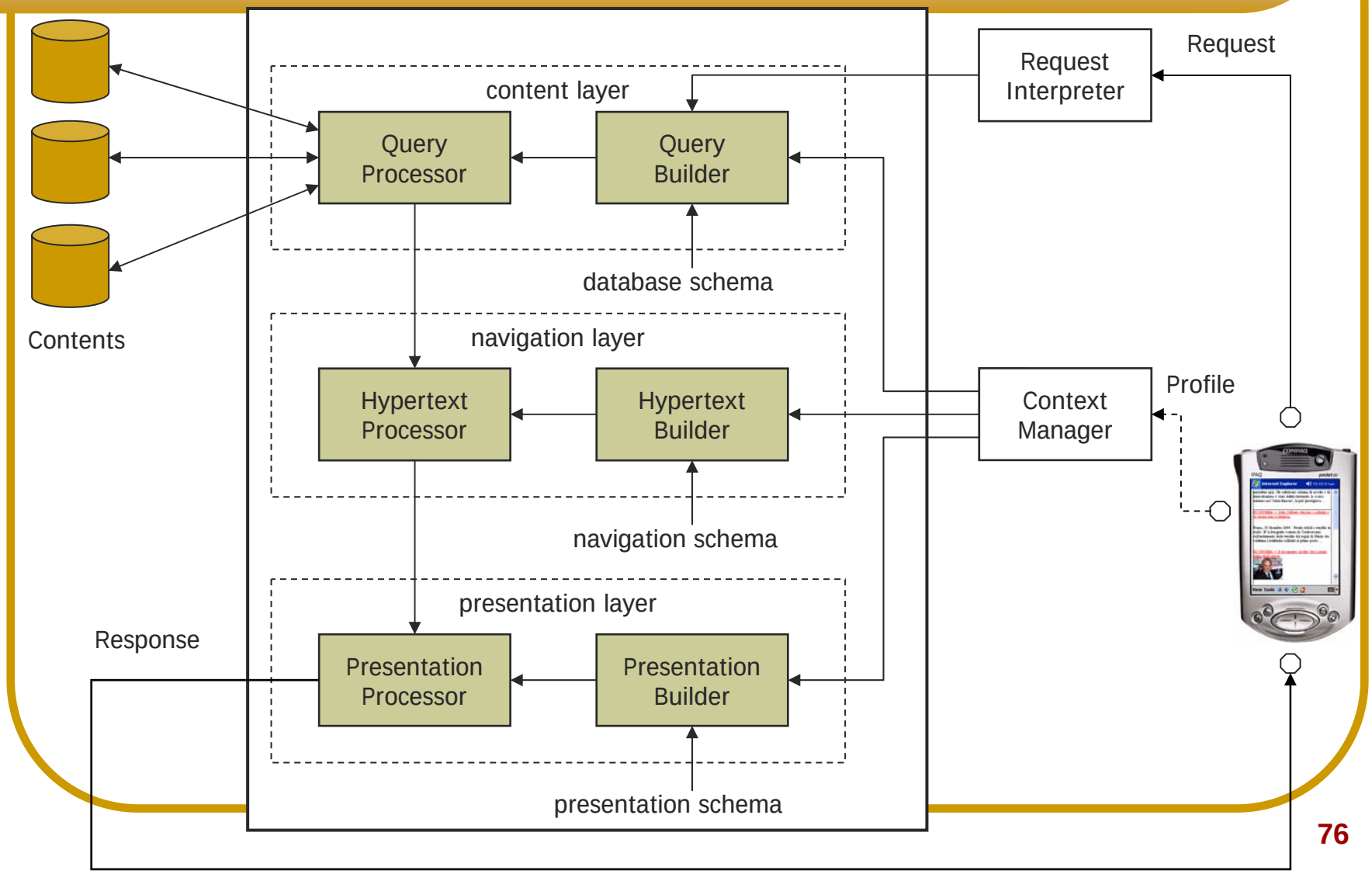
Presentation layer

Text	Link
Font: Arial **Size**: 10pt **Style**: Normal **Color**: black **Border:** 0pt	**Note**: FALSE **Font**: 10pt **Size**: Normal **Style**: Underline **Color**: black

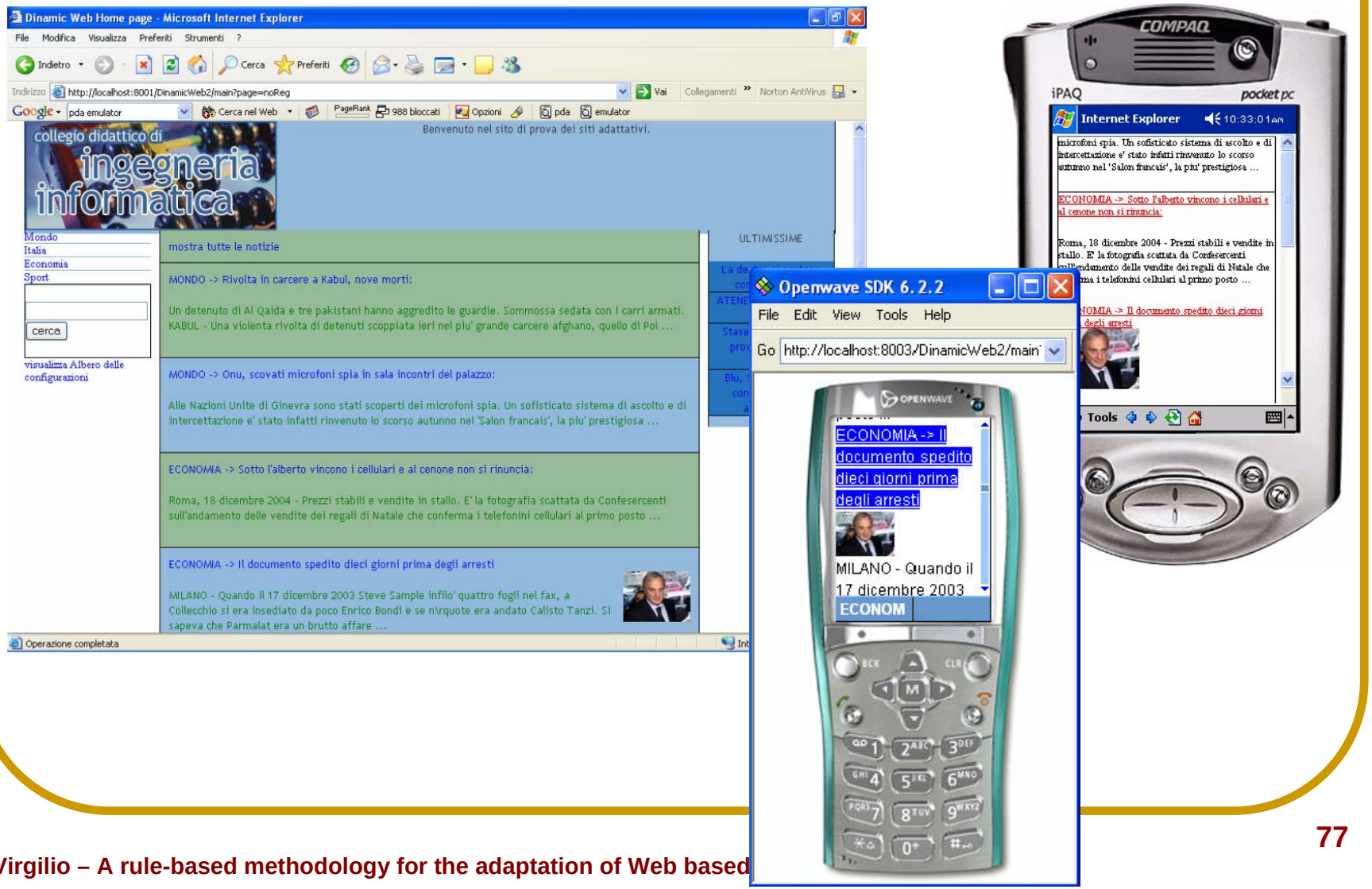
Seminario

- Documentazione sul modello RDF, OWL, SPARQL, Jena e Inference Rules
- Studio dell'operatore di composizione
- Studio di una relazione formale per stabilire conflitti tra configurazioni
- Definizione di un algoritmo per la composizione di configurazioni

FAWIS: a prototype application



Several responses



Thanks for the attention

rde79@yahoo.com

http://mais.dia.uniroma3.it/rodevirg