
Meta-modelización de Expresiones de Cálculo de Vistas

**Trabajo de Pasantía
Laboratorio PRISM - Universidad de Versailles
Noviembre 2001 - Marzo 2002**

**Verónica Peralta
Abril 2002**

Plan

Contexto.

Meta-modelo

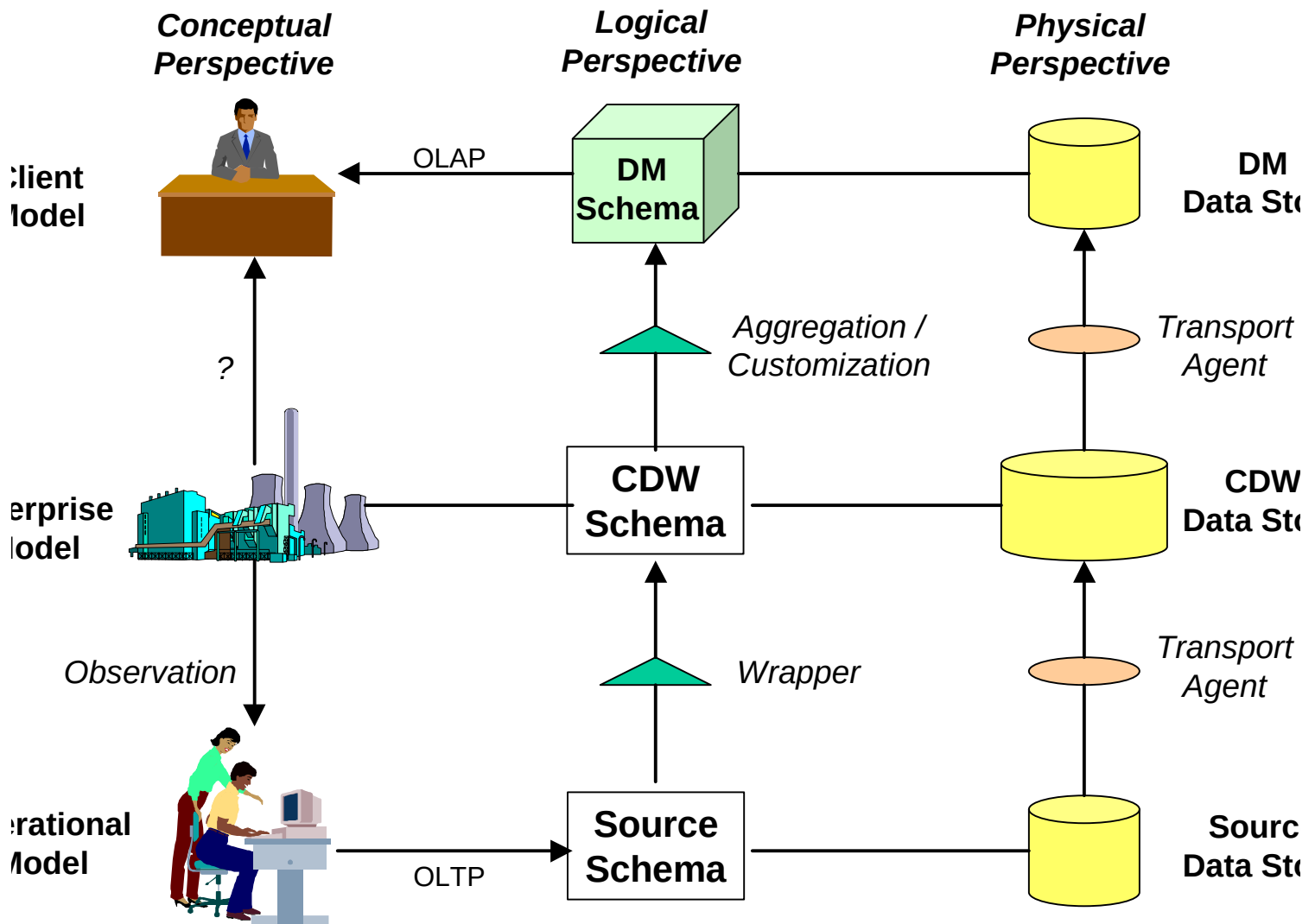
- Expresiones de Cálculo de Vistas.
- Reglas de Cálculo de Atributos.

Metaforma para MSIS.

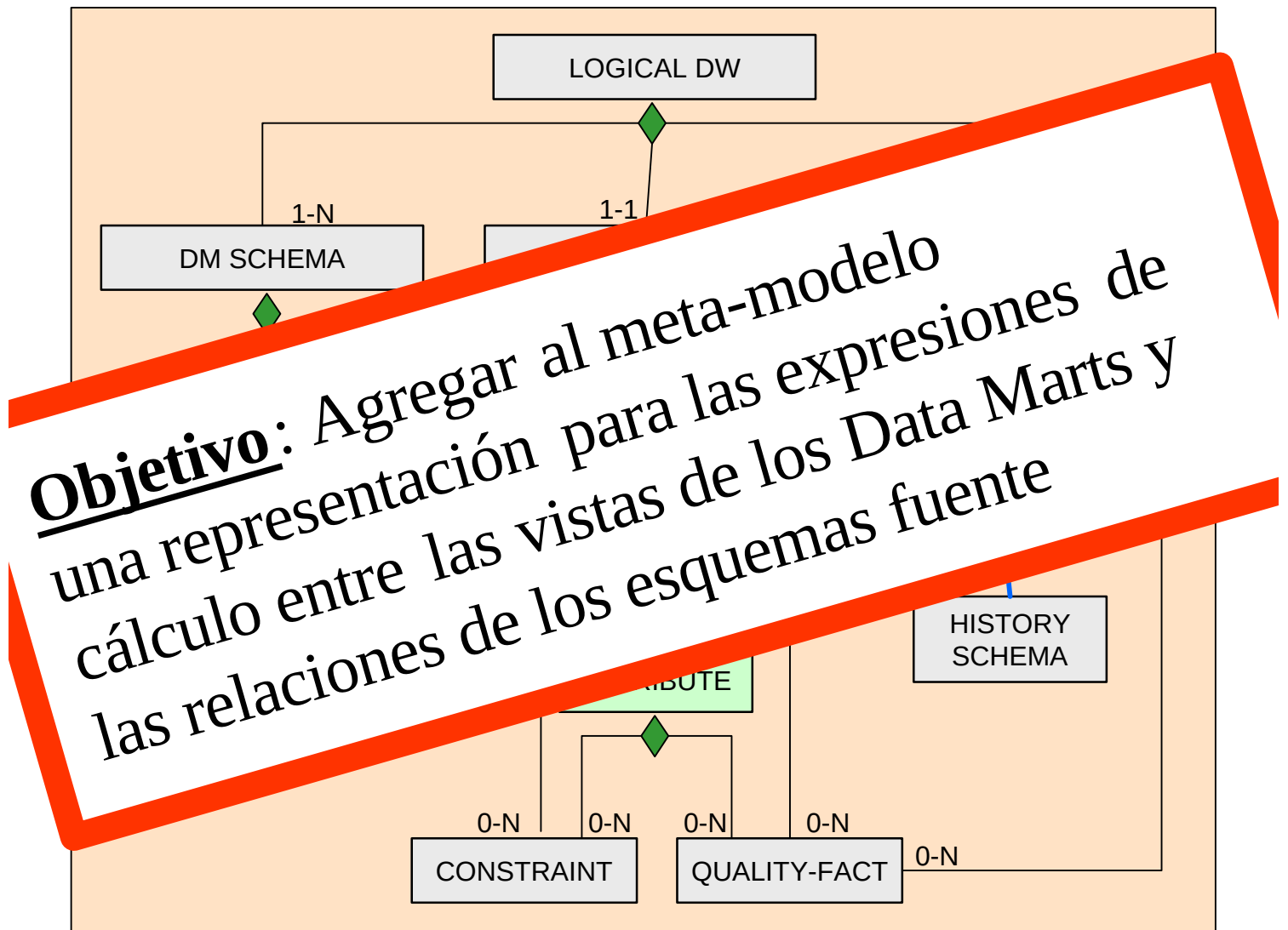
- Prototipo actual.
- Aplicaciones y herramientas.

Conclusiones.

Contexto

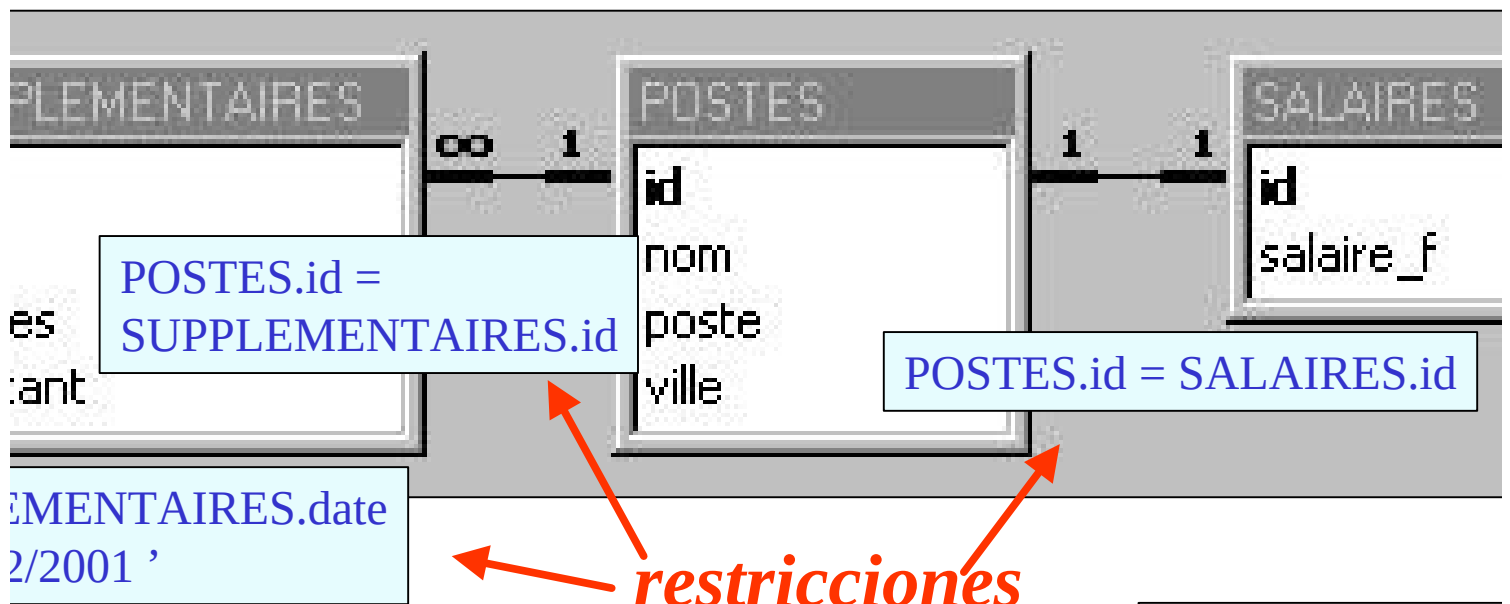


Motivación



Problema a estudiar

*relacion
funcional*



Employee ← POSTES.id
Salary_e ← SALAIRES.salaire_f * 6.56
Post ← POSTES.poste
Total ← SUM(SUPPLEMENTAIRES.montant * 6.56)

reglas de cálculo de atributos



Problema a estudiar

enriquecer el meta-modelo inicial:

- Representar las expresiones de cálculo de vistas (I)
 - Extensión del álgebra, SQL, nueva definición, ...
 - Una expresión compleja o varias expresiones simples.
- Representar las reglas de cálculo de atributos (RC,
 - Gramática, expresiones SQL, funciones de usuario, ...

explorar el meta-modelo:

- Componer las ECV a partir de las RCA.
 - Verificar la coherencia entre las ECV y las RCA.
-

Expresiones de Cálculo de Vista

Para representar una vista simple: [Aki99]

- $R = \{R_1, \dots, R_n\}$: Relaciones
- $E = S \cup A$: Expresiones formadas con atributos de las relaciones
 - S: Expresiones simples (sin agregados) para agrupar.
 - A: Expresiones con agregados.
- $C = CS \cup CA$: Restricciones (constraints) sobre los atributos de las relaciones:
 - CS: Restricciones sobre las tuplas de las relaciones.
 - CA: Restricciones sobre los agregados.
 - $CR = \{CR_1, \dots, CR_{n-1}\}$: Restricciones de join entre las relaciones

$$\sigma_{CA} (\pi_{S,A} (\sigma_{CS} (R_1 \bowtie_{CR_1} \dots \bowtie_{CR_{n-1}} R_n)))$$

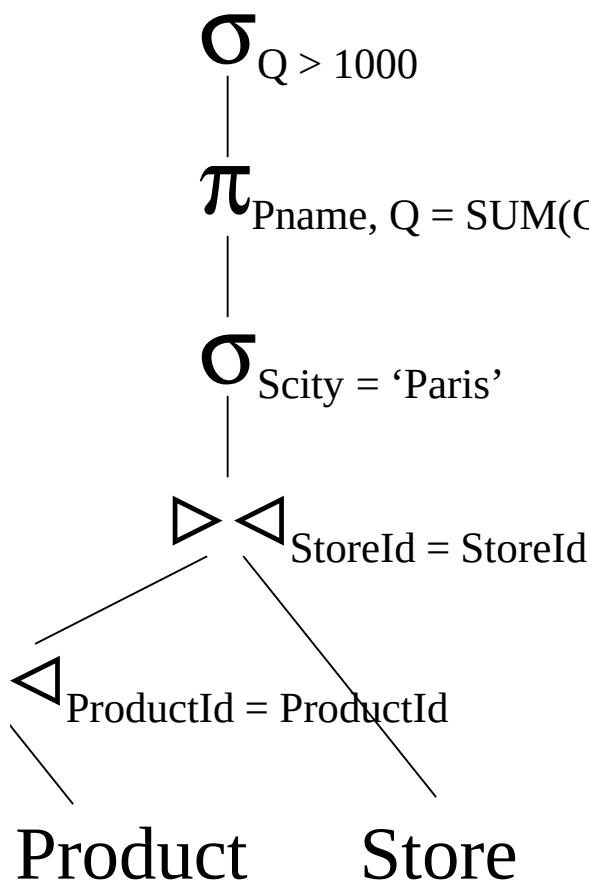
Normal Form [Gup9]

Para representar vistas anidadas (nested views) [G]

- R_i puede ser una relación o una vista.
-

Ejemplo de ECV

Find the average selling price and the total quantity sold, for each product whose total quantity sold is greater than 1000, at Paris

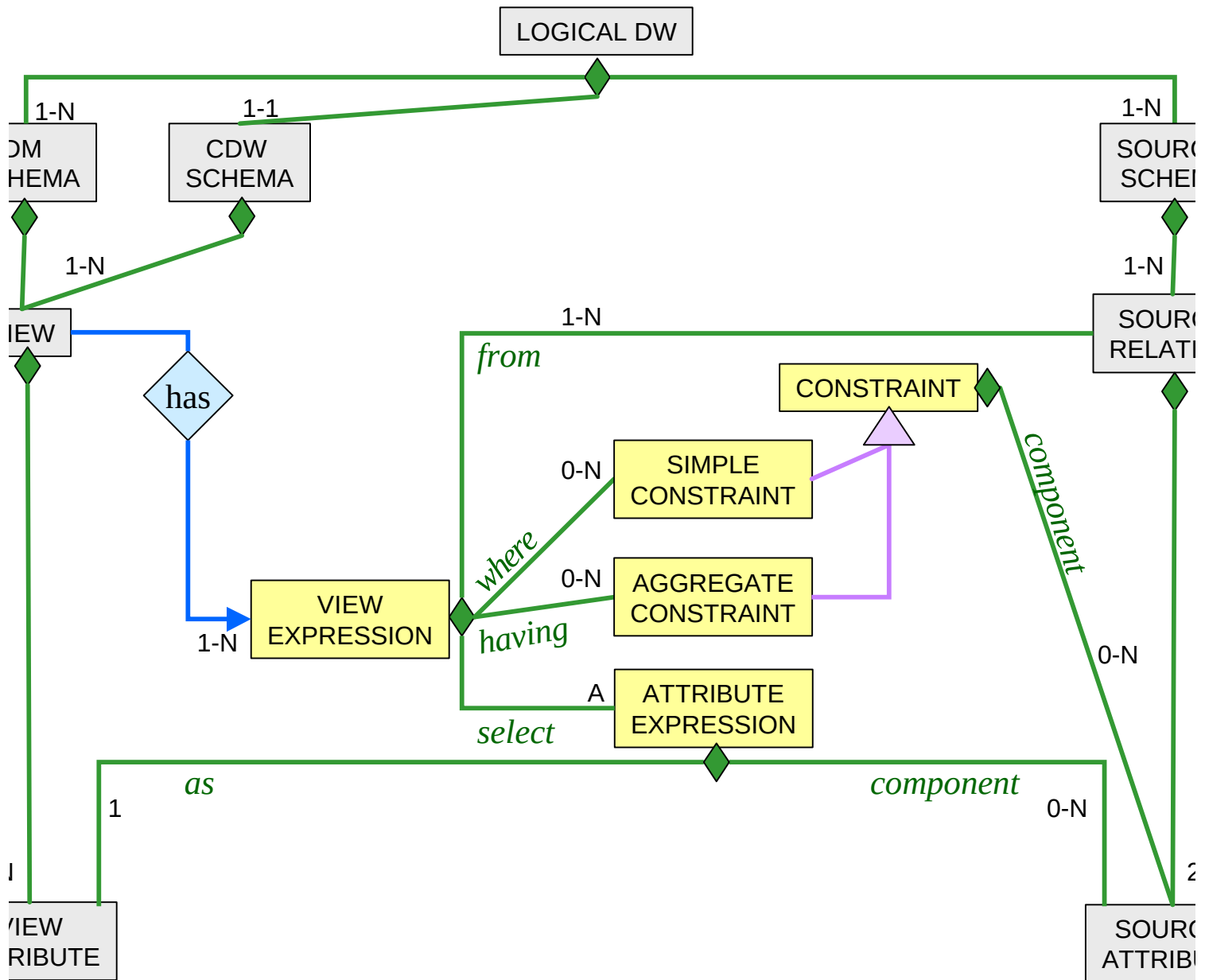


STORE (Storeid, Sname, Scity)
PRODUCT (Productid, pname, ptype, price)
SALE (Timeid, Storeid, Productid, Qty, Price)

```
SELECT  pname, sum(qty) as Q, avg(p  
FROM    Sale, Product, Store  
WHERE   Sale.storeid=Store.storeid  
        AND  Sale.productid= Product.pro  
        AND  Scity = ' Paris '  
GROUP BY  pname  
HAVING   sum(qty) > 1000
```

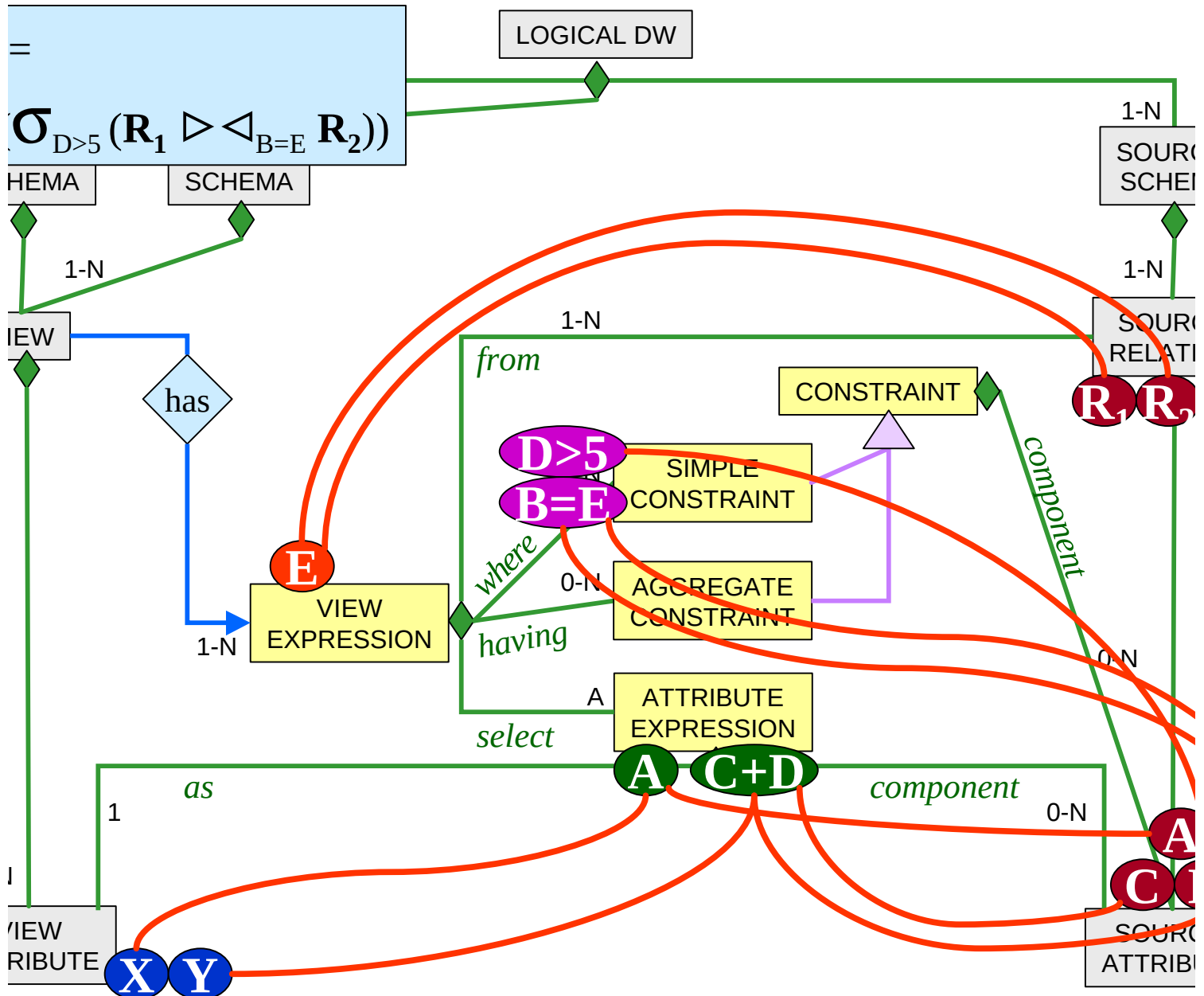
Otros ejemplos: [Go

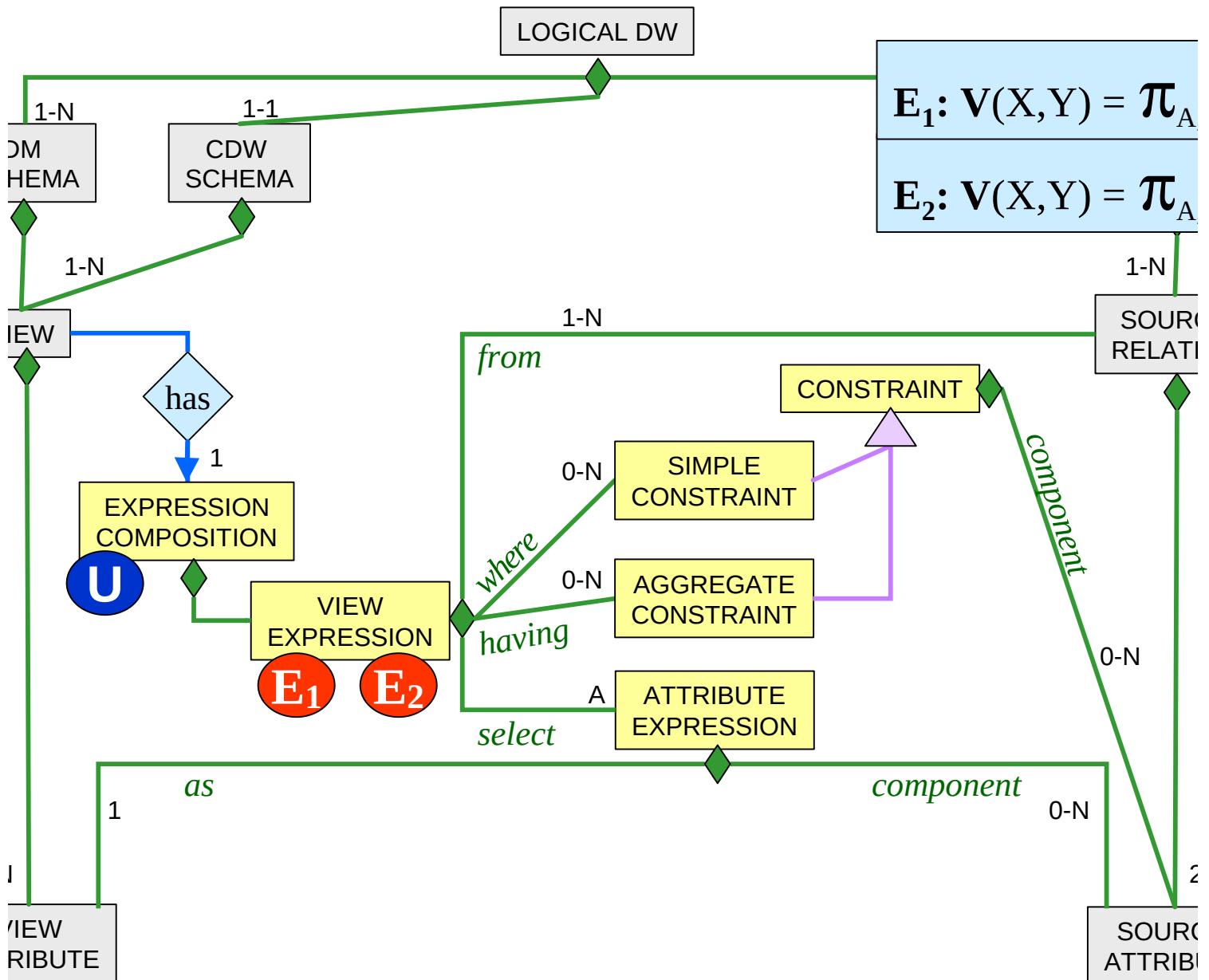
Meta-modelo simple



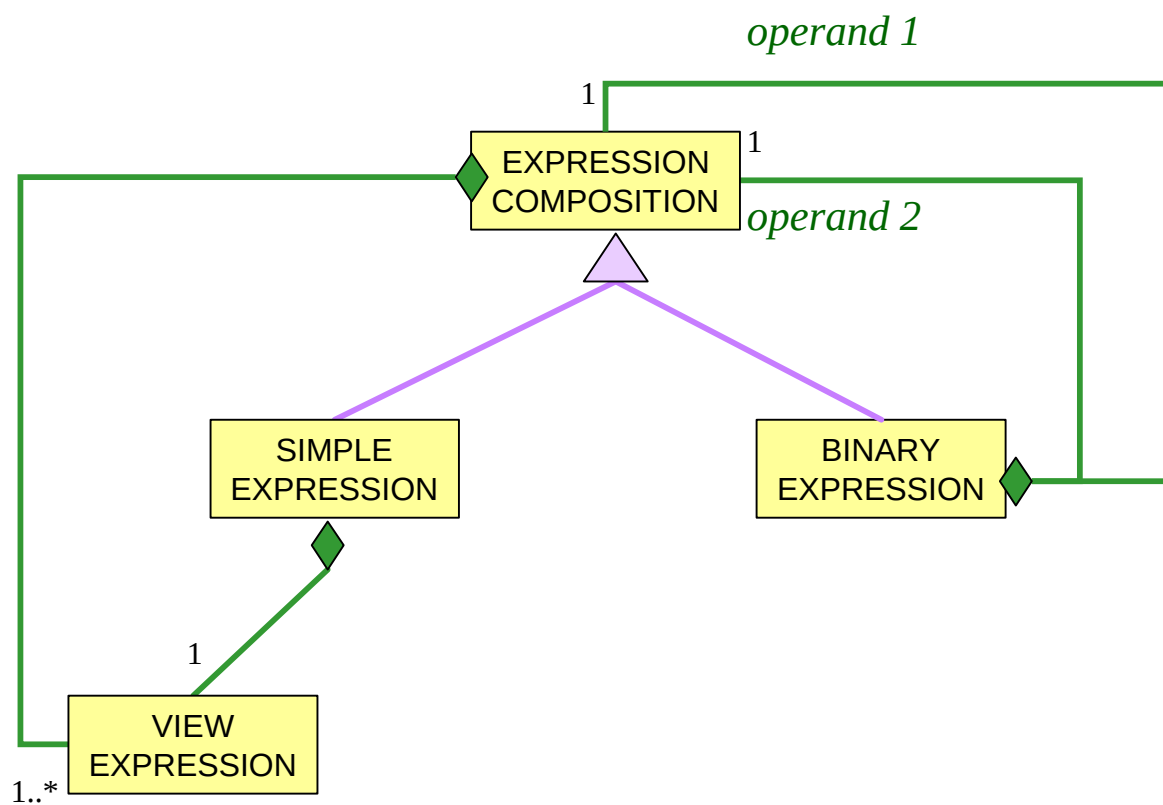
E)

Ejemplo del Meta-modelo



[illegible]

Composición de ECV



[illegible]

Reglas de Cálculo de Atributo

Un atributo de una tabla fuente.

- POSTES.nom

SE_{xj}

Un cálculo con atributos de tablas fuentes.

- Majuscule (POSTES.nom)
- Concaténation (POSTES.id, POSTES.nom)
- Premier-mot (POSTES.nom)
- SALAIRES.salaire_f * 6.56

SE_{xj}

SE_{xj}

SE_{xj}

SE_{xj}

Un agregado con atributos de tablas fuentes.

- Sum (SALAIRES.salaire_f)
- Sum (SALAIRES.salaire_f * 6.56)
- Sum (SALAIRES.salaire_f) / Count (SALAIRES.salaire_f)
- Sum (SALAIRES.salaire_f) / VILLES.viatique

AE_{xj}

AE_{xj}

AE_{xj}

CE_{xj}

Un valor externo a las fuentes.

- Aujourd'hui ()
- 6.56
- Auto-numérique ()

VE_{xj}

VE_{xj}

VE_{xj}

RCA: Un ejemplo

The diagram illustrates data relationships between four tables in a database. Yellow arrows indicate primary key to foreign key relationships, while red arrows indicate calculated values.

POSTES : Table

id	nom	poste	ville
1	Anne	Directeur	Paris
2	Gerard	Ingenieur	Paris
3	Nicole	Ingenieur	Versailles
4	Piere	Analyst	Versailles
5	Louis	Analyst	Paris

SUPPLEMENTAIRES : Table

id	date	heures	montant
1	03/01/02	5	500
1	04/01/02	4	400
1	05/01/02	4	400
2	16/01/02	5	450
5	30/01/02	5	400
5	31/01/02	3	240

SALAIRES :

id	salaire_f
1	15000
2	10000
3	10000
4	8000
5	8000

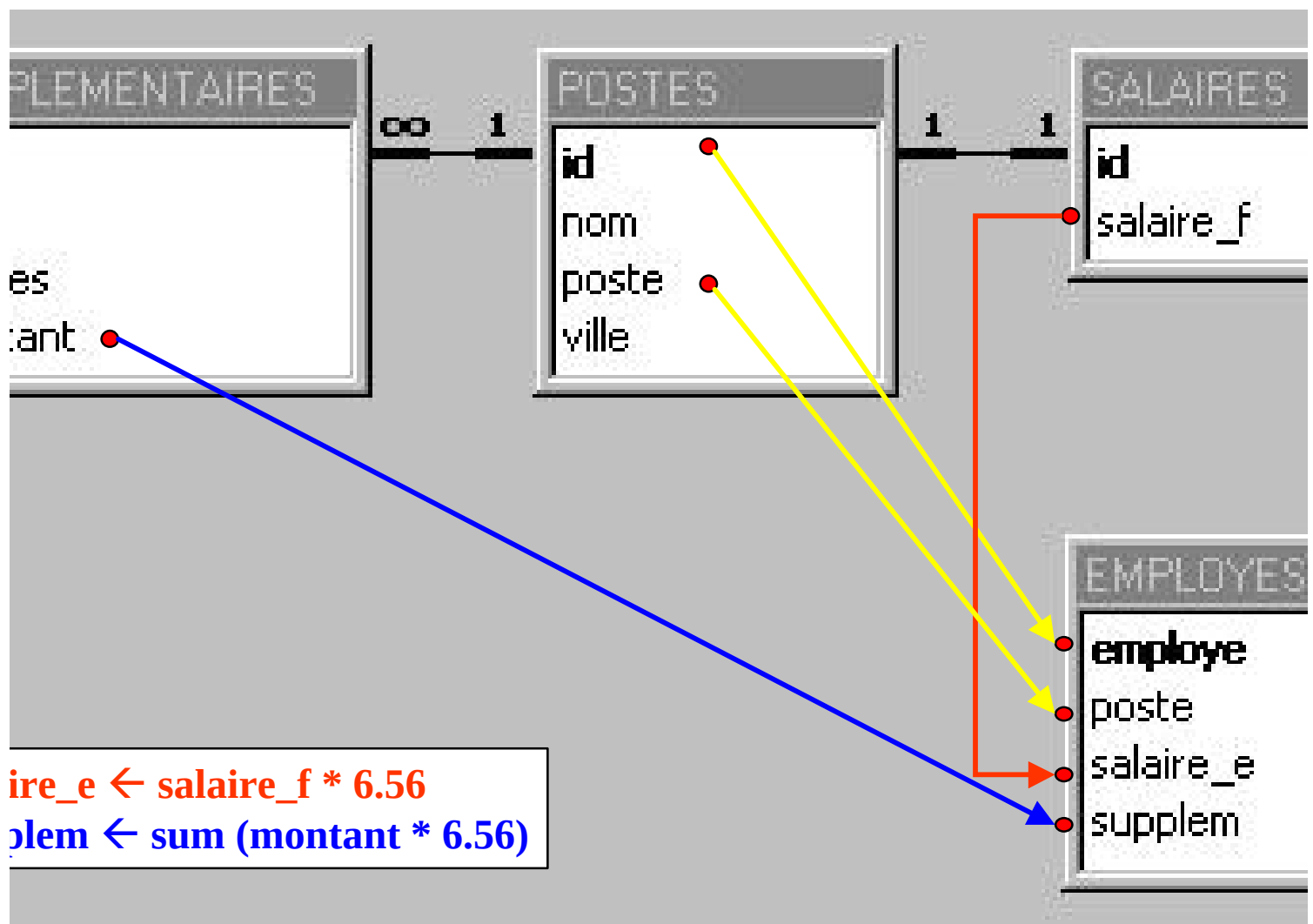
EMPLOYES : Table

employe	poste	salaire_e	supplem
1	Directeur	2286	198
2	Ingenieur	1524	69
3	Ingenieur	1524	0
4	Analyst	1220	0
5	Analyst	1220	98

Yellow arrows show relationships from **POSTES** to **EMPLOYES**: **id** to **employe** and **poste** to **poste**.

Red arrows show calculated values in **EMPLOYES**: **salaire_e** is calculated as **salaire_f** (from **SALAIRES**) divided by **6.56**, and **supplem** is calculated as **montant** (from **SUPPLEMENTAIRES**) divided by **6.56**.

Repr. Gráfica de las RCA



Gramática para las RCA

Let $BasicTypes$ be the set of basic types, such as *boolean*, *integer*, *float*, *string* and *date*.

Let $Values$ be a set of constant values of basic types.

Let $Attributes$ be a set of attributes. For each $A \in Attributes$. $Dom(A) \in BasicTypes$.

Value Expressions

If $E \in Values$

$E \in VExpr$

Simple Expressions

If $A \in Attributes$

$A \in SExpr$

If $E_1 \dots E_n \in SExpr$, $n \geq 1$, $f \in Dom(E_1) \times \dots \times Dom(E_n)$ $T, T \in BasicTypes$

$f(E_1, \dots E_n) \in SExpr$

Aggregate Expressions

If $E \in SExpr$, $g \in Set(Dom(E))$ $T, T \in BasicTypes$ $g(E) \in AExpr$

$g(E) \in SExpr$

If $E_1 \dots E_n \in AExpr$, $n \geq 1$, $f \in Dom(E_1) \times \dots \times Dom(E_n)$ $T, T \in BasicTypes$

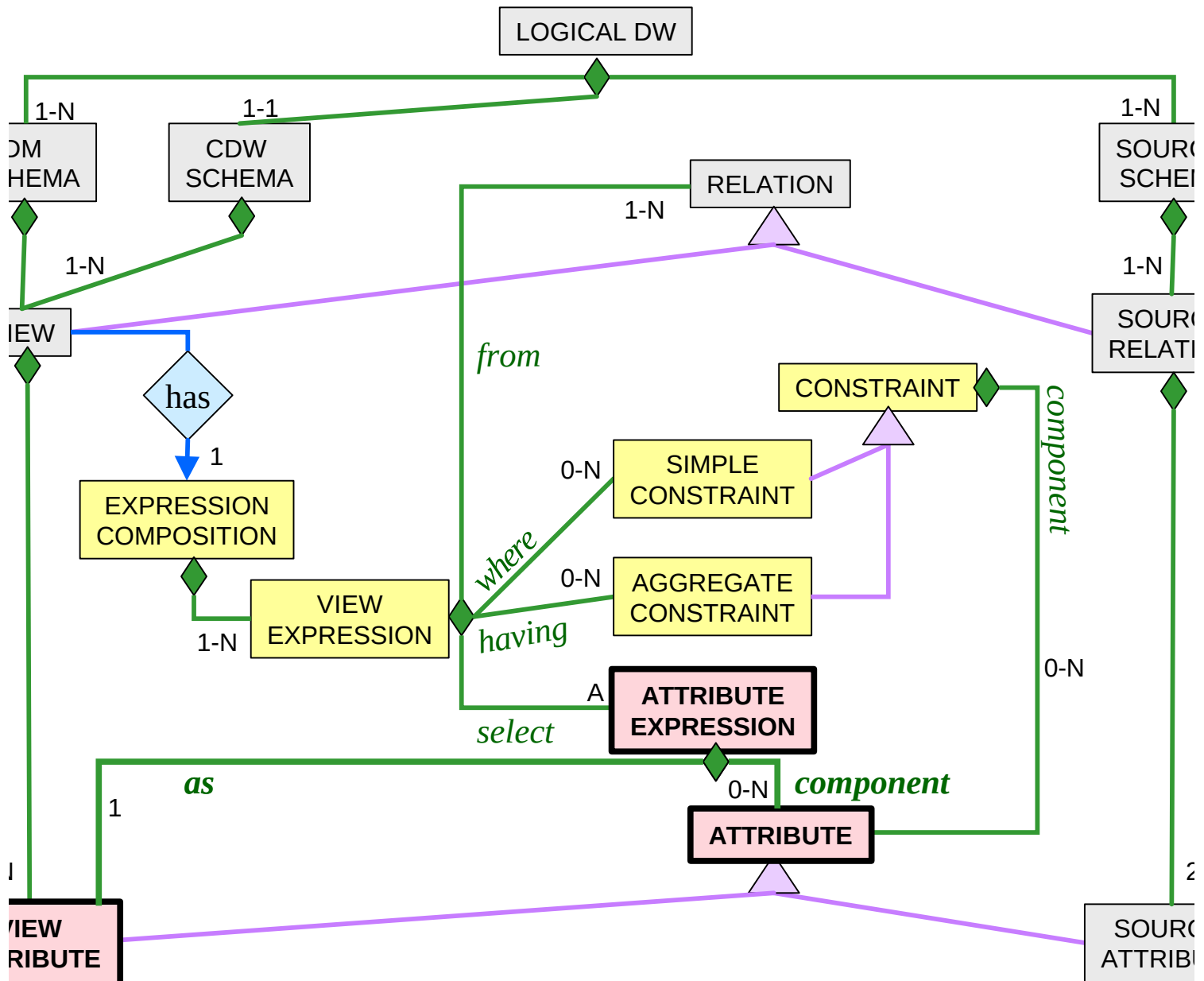
$f(E_1, \dots E_n) \in AExpr$

Composed Expressions

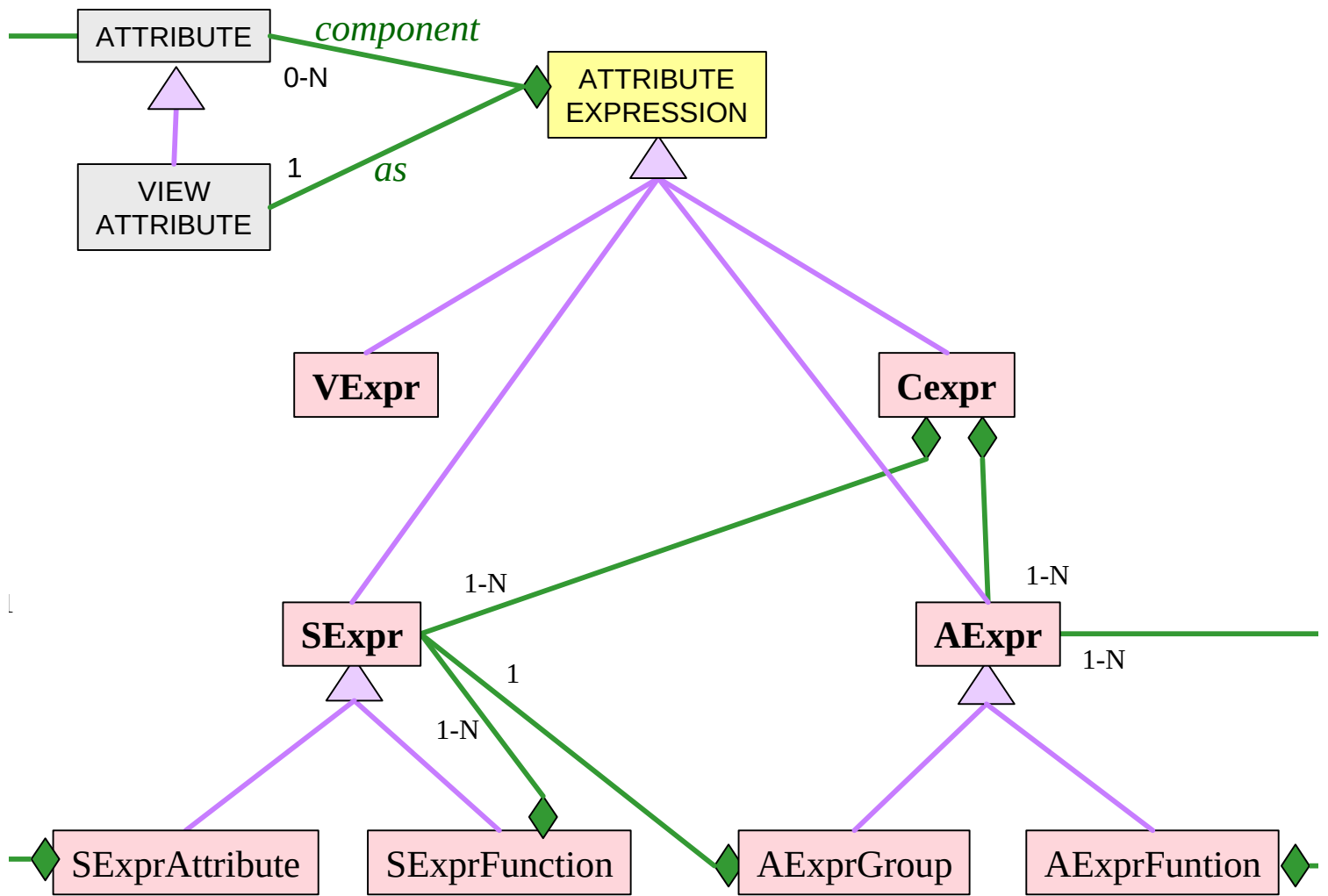
If $E_1 \in SExpr$, $E_2 \in AExpr$, $E_3 \dots E_n \in SExpr \cup AExpr$, $n \geq 2$, $f \in Dom(E_1) \times \dots \times Dom(E_n)$ $T, T \in BasicTypes$

$f(E_1, \dots E_n) \in CExpr$

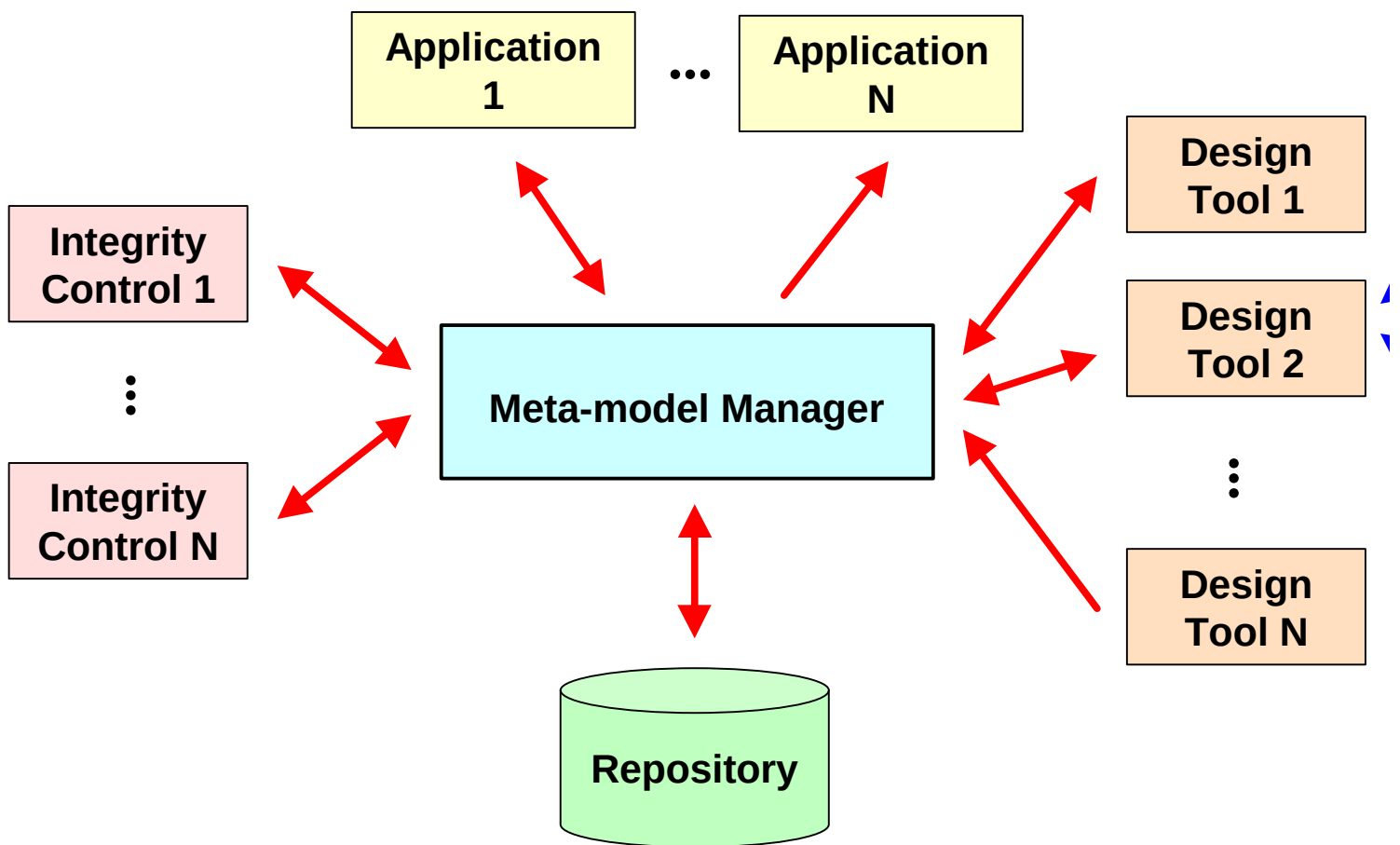
Meta-modelo lógico del DW



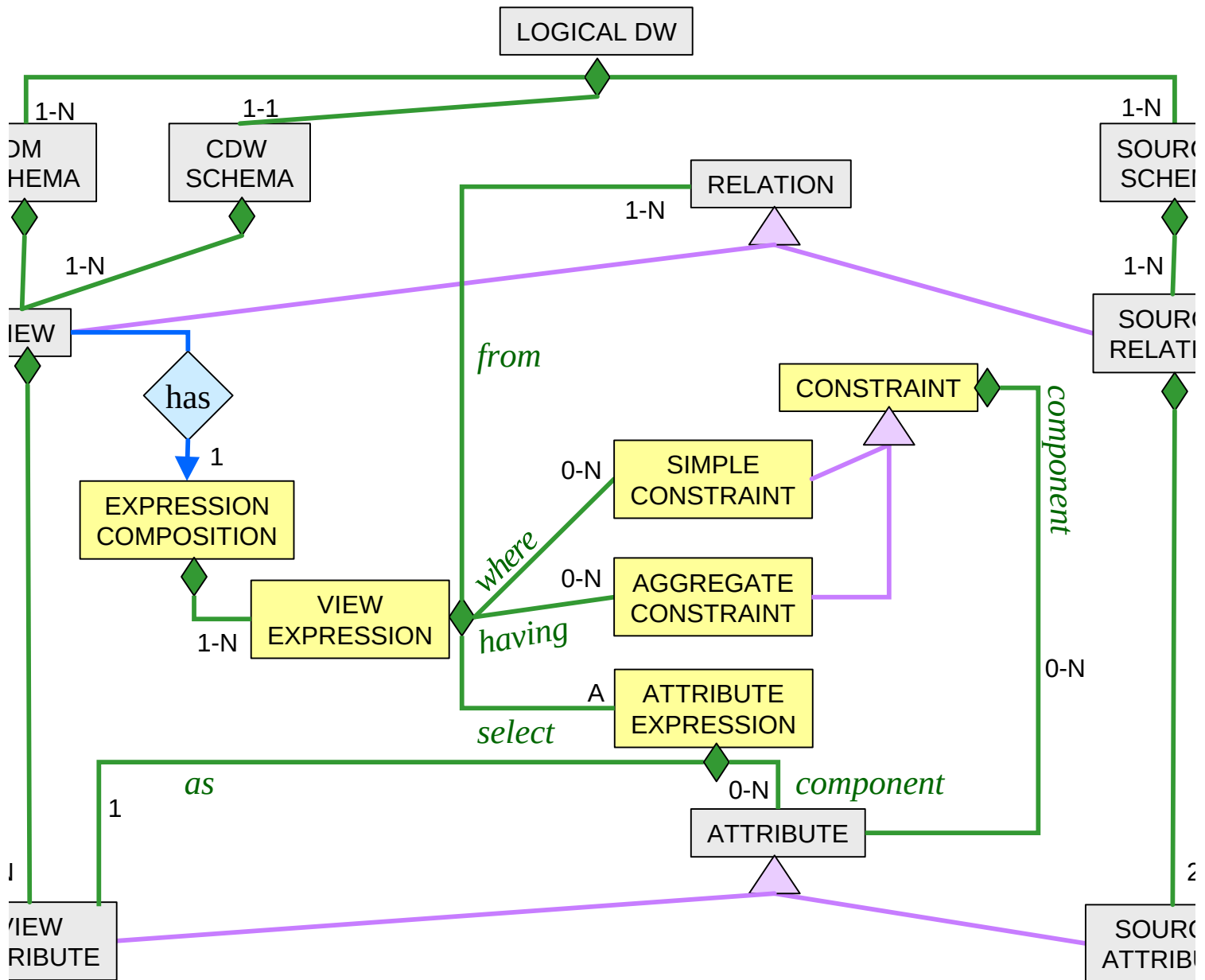
Meta-modelo de las RCA



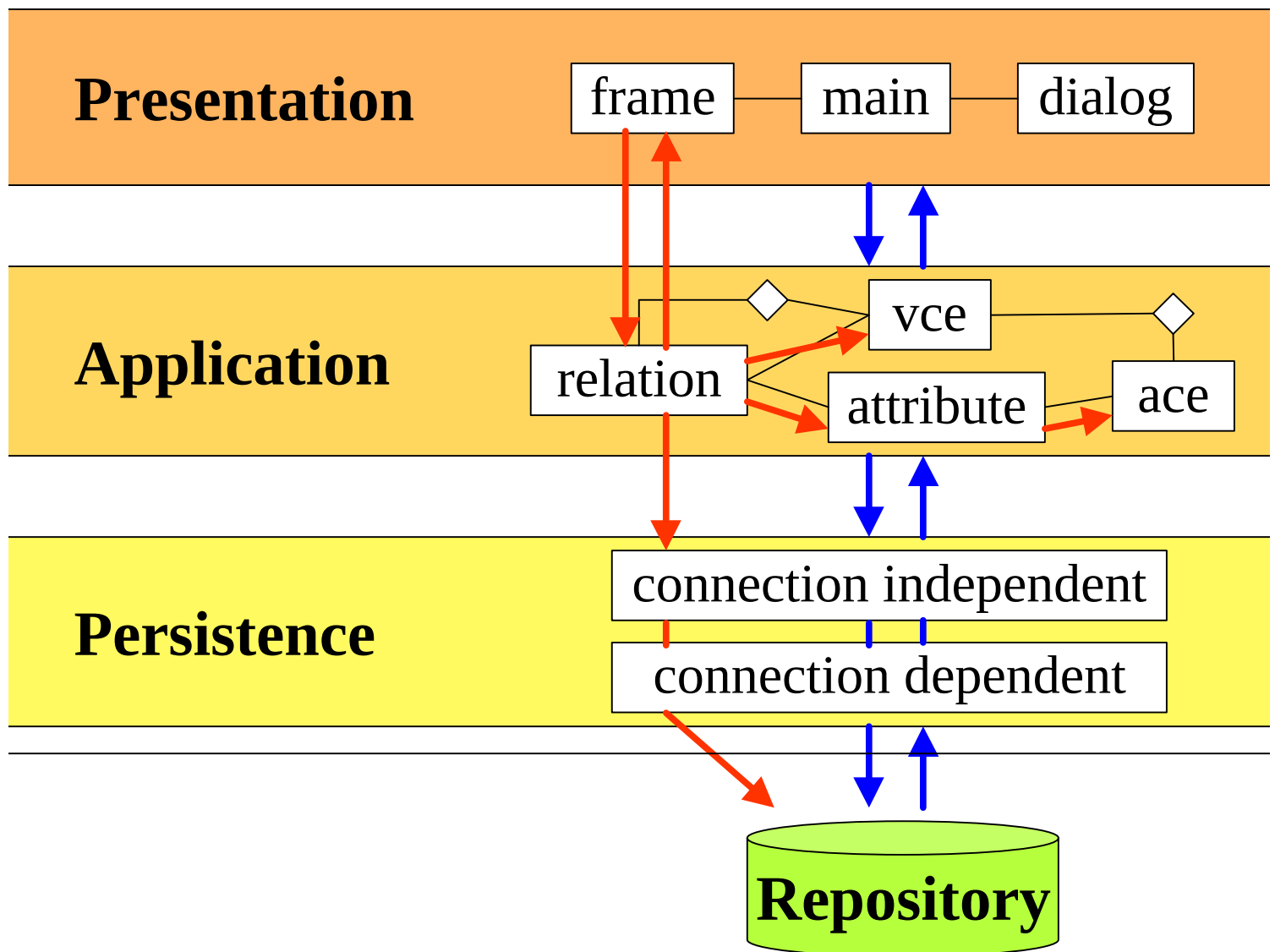
Plataforma para MSIS



Meta-modelo lógico del DW



Arquitectura del Prototipo



A framework for Multi-Source Information Systems Development

-model	Logical Meta-model	Physical Meta-model

uses

nas

Data Warehouse

Properties

Identifier: 1

Name: DW1

Description: dw probe

Size (MB): 1

Designer: vero

Creation Date: 05/02/2002

Source Schemas

- (New)
- (Existent)
- SS10
- SS2

Corporate DW Schema

- (New)

Data Mart Schemas

- (New)
- VS1
- VS2

Simple Constraints

- (New)
- T1.A1 = T2.A5
- T1.A1 not null AND
- T3.A8 = un_valeur

Aggregate Constraints

- (New)
- sum(T2.A4) > 10
- sum(T2.A4) > 5 OR
- sum(T1.A3) > sum(T1.A4)

Cancel OK

Meta-model	Logical Meta-model	Physical Meta-model

ernmas

View Calculation Expression

Properties

Identifier	1
Name	expr1-V4
Vue	V4

Relations

(Existent)
T1
T2
T3

Simple Constraints

(New)
(Existent)
T1.A1 = T2.A5
T1.A1 not null AND

Calculation Expressions

(Existent)
concat (T1.A1, T1.A2)
T2.A4 + T1.A3

Aggregate Constraints

(New)
(Existent)

Cancel OK

Aplicaciones y herramientas

Discovering view calculation expressions:

- Cómo calcular las ECV accediendo a fuentes existentes. [Ked99]
- Input: views, source relations.
- Output: ECV.

Automatic query rewriting using user profiles:

- Los requerimientos de usuarios pueden ser re-escritos integrando perfiles de usuarios (en forma de vistas). [Lah02]
 - Input: requerimientos (externos a la plataforma).
 - Output: views, ECV.
-

Aplicaciones y herramientas

W cleanning processes:

- [Sou02]
- Input: source schema, CDW schema.
- Output: ECV, loading processes information.

DW logical design and evolution

- [Mar00], [Mar01], [Per01]:
 - Input: source schemas.
 - Output: CDW schema.
-

Conclusiones

Se realizó:

- Estudio de ECV y RCA.
- Meta-modelo para ECV y RCA.
- Prototipo de la plataforma.

Trabajo futuro:

- Verificación de la coherencia entre ECV y RCA.
 - Composición de ECV a partir de RCA.
 - Enriquecer meta-modelo conceptual y físico.
 - Derivación de ECV y RCA a partir del meta-modelo conceptual.
 - Desarrollo de aplicaciones y herramientas.
 - Meta-modelizar los procesos.
-