

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Departament de llenguatges i sistemes informàtics

**ACTUALITZACIÓ CONSISTENT DE
BASES DE DADES DEDUCTIVES**

Autor: Enric Mayol Sarroca
Director: Ernest Teniente i López

Barcelona, 2000

8. Implementació del mètode

Al capítol 6 d'aquesta tesi hem definit l'arquitectura general del mètode i una descripció detallada de cada un dels mòduls que la componen. En aquest capítol farem una breu descripció de com s'ha realitzat la implementació d'aquesta arquitectura.

Aquesta implementació s'ha realitzat utilitzant el llenguatge Quintus-Prolog (versió 3.1.4) en una estació de treball Sun. Aquesta implementació s'ha realitzat amb la col·laboració de dos estudiants d'Enginyeria Superior en Informàtica que han realitzat els seus projectes final de carrera emmarcats en la implementació del mètode. El treball de [Alb98] se centra en el mòdul encarregat de la generació del Graf de Precedències, mentre que el treball de [Fer99] es correspon a la implementació de la resta de mòduls i la integració del treball de [Alb98] a l'arquitectura del mètode. Més detalls de la implementació es poden trobar en els documents [Alb98] i [Fer99].

La implementació del sistema s'ha dividit en dues parts clarament diferenciades. Un conjunt de mòduls que implementen la component del mètode a executar en temps de definició i un segon conjunt de mòduls que implementen la component de temps d'execució.

Donada una base de dades deductiva D , la generació de la base de dades augmentada $A(D)$ queda fóra de l'àmbit d'aquesta implementació del mètode, per tant, existeix un procés previ de generació de l' $A(D)$ que es realitza invocant l'aplicació implementada per [BFU92]. Aquesta aplicació implementa el generador de la base de dades ampliada definida a [Urp93].

Així doncs, la informació d'entrada requerida per aquesta implementació del mètode serà la base de dades augmentada $A(D)$ i la petició d'actualització U requerida per l'usuari. El resultat que es genera amb aquesta implementació és el conjunt de solucions que satisfan la petició U sense violar cap restricció d'integritat. A més a més, s'obtenen una reescriptura de la $A(D)$ i el Graf de Precedències definit a partir d'aquesta $A(D)$.

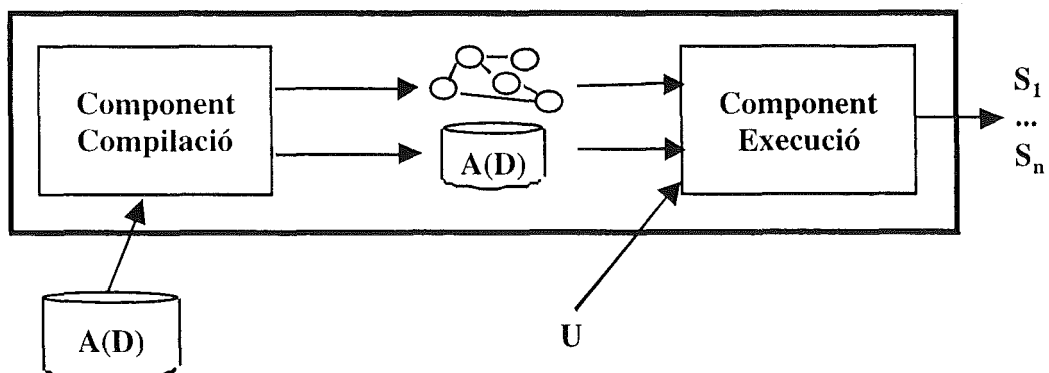


Fig. 8.1 Estructura global del mètode

8.1 Mòduls en temps de compilació

Aquest component té per objectiu reescriure les regles de la base de dades augmentada A(D) i generar el Graf de Precedències. L'estructura modular d'aquest component és la representada a la figura 8.2.

La única font d'informació necessària per a l'execució d'aquest component és la base de dades augmentada generada per [BFU92]. La notació utilitzada en la definició i reescriptura de l'A(D) és la mateixa que s'utilitza a [BFU92].

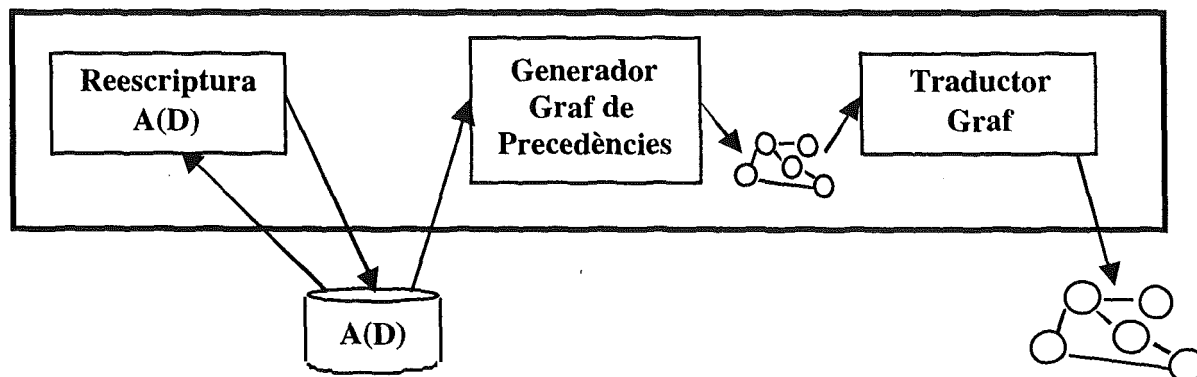


Fig. 8.2 Component en temps de Compilació

Aquest component està subdividit en tres mòduls:

- **Mòdul de Reescriptura A(D):** Aquest mòdul és l'encarregat d'aplicar les reescriptures de les regles d'esdeveniment definides al capítol 3 d'aquest document.
- **Mòdul Generador del Graf de Precedències:** Aquest mòdul és l'encarregat de generar i optimitzar el Graf de Precedències a partir de la base de dades augmentada A(D) reescrita. El mecanisme per a realitzar aquesta generació segueix el procediment descrit a la secció 6.3.
- **Mòdul Traductor del Graf:** Aquest mòdul és l'encarregat d'adaptar l'estructura interna del Graf de Precedències a la notació utilitzada a la resta de mòduls.

La base de dades augmentada reescrita serà la que s'utilitzarà en tots els mòduls que componen la implementació del mètode, i el Graf de Precedències serà utilitzat, en temps d'execució, en el mòdul de manteniment de la consistència.

La notació utilitzada per a descriure els elements del Graf de Precedències és la següent:

- Les arestes contenen informació del node origen, del node destí i de l'esdeveniment bàsic que etiqueta l'aresta.

Aresta(origen, destí, etiqueta)

- Cada node conté la informació referent a:

Node(etiqueta, pare, actiu, marcat, condició, origen, tipus)

on

etiqueta: es correspon a l'identificador del node ($C_i, I_i, D_i, M_i, A_i, g_i$)

pare: identificador del subgraf al que pertany (si és el cas)

actiu: si el node està actiu (on) o no (off)

marcat: si el node està marcat (y) o no (n)

condició: conté la llista de parelles següent:

(*node, marcat*): El node es correspon a un subgraf i conté la llista de nodes i marques que el componen.

(*condIC, marcat*): El node conté una única condició de restricció d'integritat i la seva marca.

(*condAV, marcat*): El node conté la llista de condicions d'actualització de vistes (especialitzades) i les seves marques.

origen: indica el tipus de condició associada al node. Si és una condició de restricció d'integritat (r), si és una condició d'actualització de vistes (v), o si és un cicle (c).

tipus: indica si la condició associada és de comprovació (c) o de generació (g).

8.2 Mòduls en temps d'execució

Aquest component té per objectiu el traduir la petició d'actualització U en un conjunt de transaccions que satisfacin la petició inicial i que no violin cap restricció d'integritat. L'estructura modular d'aquest component és la representada a la figura 8.3.

La font d'informació necessària per a l'execució d'aquest component és la petició d'actualització U requerida per l'usuari, la base de dades augmentada rescrita i el Graf de Precedències generats pel component en temps de compilació.

Aquest component està dividit en dos grans mòduls:

- **Mòdul d'Actualització de Vistes**: Donada una petició d'actualització U (o de reparació R) i una base de dades augmentada A(D) (que inclou la base de dades extensional), aquest mòdul obté un conjunt de traduccions compostes, cada una d'elles, per una transacció T i un conjunt C de condicions. Aquest mòdul es divideix en dos submòduls:
 - Mòdul d'Anàlisi: especialitza les regles d'esdeveniment que s'utilitzaran per a generar les traduccions, tenint en compte la petició U i el contingut de la base de dades extensional.
 - Mòdul de Traducció: construeix els conjunt T i C de cada traducció a partir de les regles especialitzades en el mòdul d'anàlisi.

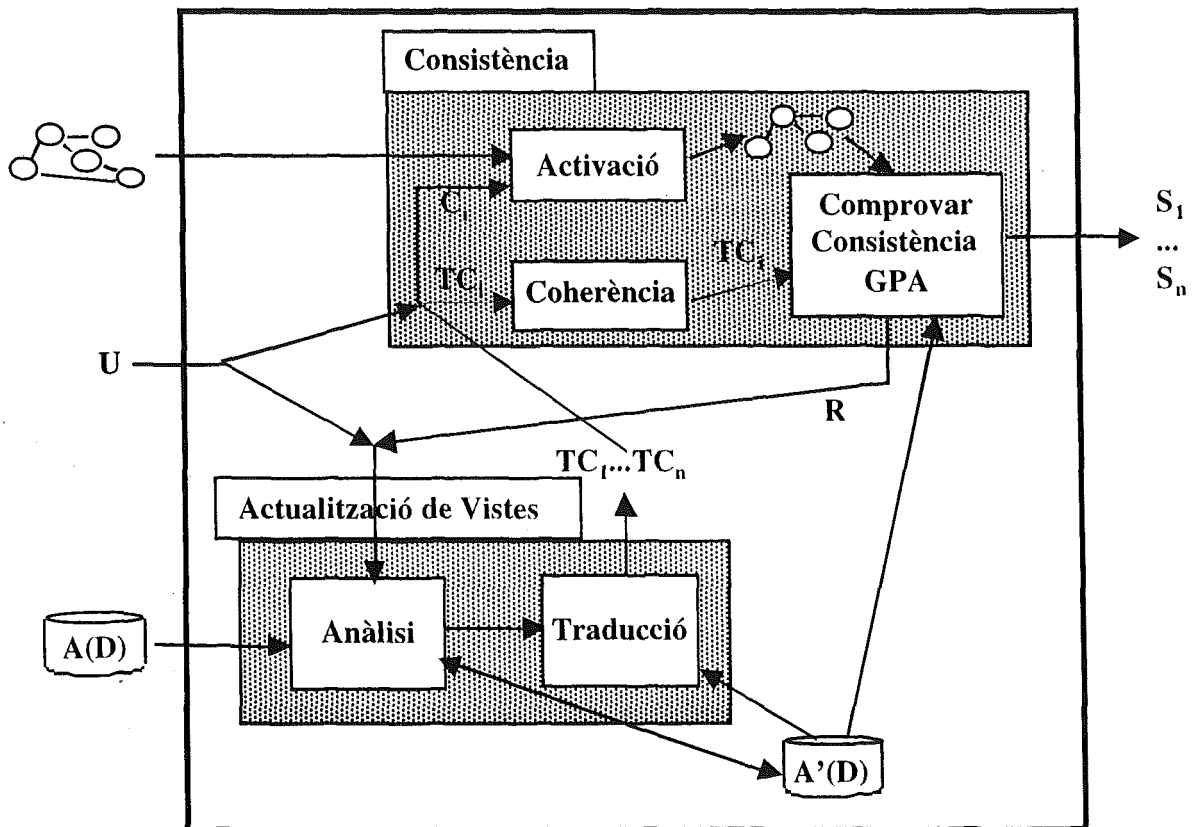


Fig. 8.3 Component en temps d'Execució

- **Mòdul de Consistència:** Donada una traducció TC (o conjunt de traduccions), una base de dades augmentada A(D) i un Graf de Precedències G, aquest mòdul assegura que tota transacció T de TC satisfà les condicions del conjunt C i satisfà les restriccions d'integritat. Aquest mòdul es divideix en tres submòduls:
 - **Mòdul d'Activació:** Aquest mòdul s'encarrega de definir el Graf de Precedències Actiu associat a la traducció TC. Donat un conjunt de condicions C i el Graf de Precedències G, activa (i marca) els nodes associats a les condicions de les restriccions d'integritat i les condicions d'actualització de vistes del conjunt C. Cada cop que es repara una condició mitjançant l'actualització d'una vista, s'activen (i marquen) els nous nodes associats a les condicions del conjunt C.
 - **Mòdul de Coherència:** Comprova que els esdeveniments del conjunt T i els esdeveniments obtinguts durant l'actualització d'una vista no són mútuament exclusius. A més a més, assegura que les condicions de comprovació del tipus ($\leftarrow Ev$) no esdevenen violades per T.
 - **Mòdul Comprovar Consistència GPA:** Donat un Graf de Precedències Actiu, el conjunt de condicions de comprovació del tipus ($\leftarrow Ev$) i una transacció T, comprova que T no viola cap de les condicions del Graf. En cas de ser necessari reparar una condició definida per predicats derivats, pot invocar

l'execució del mòdul d'Actualització de Vistes amb la petició de reparació R.
La comprovació de les condicions es realitza segons l'ordre definit pel Graf
de Precedències Actiu.

Al final de l'execució d'aquests dos mòduls, s'obtenen totes les solucions (transaccions) que
satisfan la petició inicial d'actualització U i que no violen cap restricció d'integritat.