

• 1400/90/89
COPIA 1

**La problemàtica d'us
dels SGBD's
en sistemes CAE**

Sebastià Vila

Report LSI-93-18-T



Facultat d'Informàtica
de Barcelona - Biblioteca

31 ENE. 1994

Abstract:

CAE applications are characterized by the complexity and volume of managed data. Current trends in CAE point towards a higher integration between currently used tools and the engineering process itself. This integration heavily relies on two main techniques: the grouping of all the tools under the same user interface shell and allowing all tools to share the same data. Because of this, the DBMS are becoming a main topic inside CAE architectures. In this report, the most salient requirements that a DBMS must satisfy in order to be satisfactorily used in a CAE system are identified and briefly discussed.

La problemàtica d'ús dels SGBD's en sistemes CAE

Sebastià Vila i Marta
Dept. Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
16 - Desembre - 1993

Resum

Les aplicacions de CAE es caracteritzen per la complexitat i volum de les dades manipulades. Conforme la tecnologia avança, la tendència a l'integració de totes les aplicacions relacionades amb el procés productiu augmenta. Aquesta integració es fonamenta en dos pilars: l'agrupació sota una mateixa interfície d'usuari i la compartició de les dades. En aquest entorn, l'ús de SGBD's per a l'emmagatzematge i manipulació de les dades es converteix en un punt fonamental de tota arquitectura. En aquest informe s'analitzen des del punt de vista de les aplicacions CAE els principals requeriments que l'SGBD ha de satisfer i fins a quin punt els SGBD's actuals són adequats.

1 Introducció

L'ús de les aplicacions de CAE és un camp en alça dins la indústria actual. A mida que s'incrementa la introducció d'aquestes aplicacions, augmenta la necessitat d'integració amb sistemes de gestió de bases de dades (SGBD's) per tal de facilitar la gestió dels volums de dades manipulades i permetre la integració amb la resta de l'entorn informàtic (vegeu per exemple [Wol87]). S'ha vist, però, que els SGBD's tradicionals —entenent per tradicionals aquells que es fonamenten en el model relacional— no són les eines adequades per a solucionar els problemes de gestió de les aplicacions CAE [Mai89].

La constatació d'aquesta realitat per part dels grups que fan recerca en SGBD's, ha impulsat aquest món de les bases de dades cap a la búsqueda de nous models que satisfegin les necessitats de les aplicacions de CAE. Hi ha moltes noves aportacions en aquest sentit, vegeu per exemple [Atk89], [Car90], [Cod79], [Jos90], [Pec88] o [Sto90]. Algunes d'aquestes han estat usades en prototipus o analitzades sota l'òptica de les aplicacions CAE: [Bar90], [Cha90], [Dul88], [Dür90], [Har87], [Har90], [Kem87], [Ket87], [Kim90], [Spo88] i [Wie86] en són un exemple. No obstant i això, la majoria de les innovacions solament presenten resultats positius respecte a un subconjunt de les necessitats.

A aquest informe es descriuen quins són els principals requeriments que un SGBD ha de satisfer per tal que siguin eines útils dins un entorn CAE. A l'apartat primer, es fa una breu caracterització de les aplicacions de CAE. Dins el segon apartat s'enumeren i estudien amb més detall els requisits que es deriven de la

caracterització. Finalment, en el tercer apartat, s'apunten algunes conclusions així com temes que resten oberts.

2 Les aplicacions de CAE. Una caracterització

En aquest apartat es subratllen alguns dels trets més característics de les aplicacions de CAE. Aquests trets diferencials determinen les necessitats de gestió de dades d'aquestes aplicacions. No obstant i això, l'ús d'aquesta informació és difícil a causa de la pobra formalització i quantificació.

També és important remarcar que, tot i l'èmfasi que fem en aplicacions CAE, la problemàtica no és únicament pròpia d'aquestes. En certa mesura hom podria dir que aquesta és una problemàtica pròpia de les eines de disseny [Atk89], [Sto90], [Car90]. Altres aplicacions de disseny tenen requisits similars, en especial:

- L'enginyeria del software i les eines CASE.
- Les aplicacions multimedia i hipertext.

Les aplicacions de CAE podriem definir-les com aquelles eines software que s'apliquen a resoldre part de la problemàtica relacionada amb el procés de disseny i manufactura d'un determinat producte. Així doncs, són eines de treball per a persones no informàtiques, dedicades al disseny i producció entès en un sentit àmpli de la paraula. Des d'aquest punt de vista hi ha algunes característiques de les eines CAE que cal prendre en consideració:

1. Els objectes manipulats per aquestes aplicacions són models dels productes en procés de disseny o manufactura. Per tal que aquests models siguin útils és necessari que siguin acurats i íntegres semànticament parlant.
2. El procés de disseny no pot modelar-se com un procés lineal sinó com un procés evolutiu. Durant aquest procés el model en creació evoluciona fins arribar al model final. La natura d'aquest procés s'ha modelitzat de diverses maneres (vegeu p. ex.: [Ng91], [Bañ91]).
3. Els productes dissenyats estan sotmesos a un cicle de vida en el temps. El disseny no és un procés lineal en el temps. Els resultats que s'obtenen de simular un cert comportament del producte o del seu funcionament en condicions reals, la variació de les especificacions o altres circumstàncies poden esdevenir factors que porten al redisseny.
4. Un model, des del punt de vista estructural, és una entitat complexa. Aquesta complexitat té dos grans aspectes:
 - Un model pot ésser format per múltiples entitats interrelacionades per múltiples interrelacions. Per tant, encara que entitat monolítica, és també una entitat disgregada en altres entitats. La majoria de les representacions usades en modelatge geomètric tenen aquestes característiques [Sam90], [Fol90].

- El volum d'informació que pot contenir un model te valors molt grans. Per tant, en pensar en models, cal no perdre de vista la qüestió de la mida.
5. Un determinat producte pot estar representat en un sistema de CAE per més d'un model. Existeixen raons de pes per actuar així, en especial:
 - Sovint és preferible tenir models especialitzats en comptes de models generals. Això permet que cada tasca treballi sobre els models millors per a dur-la a terme. Tal és el cas dels anomenats esquemes de representació híbrids [Män88].
 - També és freqüent que el dissenyador treballi amb diferents models en diferents estats del procés de disseny. En aquest cas, el producte es veuria representat per models més propers al model final conforme el procés de disseny avança.
 6. Una aplicació CAE és una eina eminentment interactiva. L'usuari ha de poder manipular amb un temps de resposta d'interacció els models o les entitats que els formen.
 7. El cicle de vida d'un producte te lloc immers en una collectivitat estructurada. En general mai és un sol individu el qui intervé en aquest cicle sino que són una collectivitat d'individus immersos en una o més organitzacions. Per que a aquest informe afecta, considerarem que aquests individus es troben organitzats jeràrquicament.

3 Requisits que ha de satisfer un SGBD

En el punt anterior hem exposat algunes de les característiques més rellevants dels sistemes CAE. Prenent com a base aquestes, durem a terme un procés d'abstracció. Com a resultat del mateix obtindrem alguns dels requeriments de gestió de dades que aquestes aplicacions posseeixen.

Aquests no són els únics requisits que hauria de complir un SGBD. Els requisits que qualsevol altre aplicació exigiria també són aplicables en aquest cas. Aquests es consideren coneguts i pot consultar-se qualsevol clàssic com [Dat90] o [Kor93] per conèixer-els. En aquest document solament considerarem aquells requisits diferencials entre les aplicacions que anomenem tradicionals i les aplicacions de CAE.

Els requisits que se'n deriven són:

1. Cal que l'SGBD pugui assegurar la integritat semàntica de les dades que emmagatzema.
2. Cal preservar la informació històrica referent als diversos estats pels que passa un objecte emmagatzemat durant el seu cicle de vida. Cal que aquest cicle de vida incorpori el temps de disseny, en el qual l'evolució històrica d'un objecte pot ser més complexa.

3. Cal suportar objectes d'estructura inherentment complexa sense perdre informació ni incrementar la dificultat d'ús.
4. Cal suportar objectes de mida molt gran com una única entitat. Això no implica necessàriament que aquests siguin estructuralment complexes.
5. Cal que l'SGBD pugui suportar el concepte d'objectes equivalents. Objectes equivalents són aquells que tenen diferent representació però modelen el mateix objecte real. Entre aquests objectes equivalents existeix generalment un conjunt molt fort de relacions semàntiques a satisfer.
6. L'SGBD ha de permetre un accés interactiu a velocitats d'interacció per als objectes emmagatzemats.
7. La gestió de la integritat ha d'ésser eminentment interactiva.
8. L'SGBD ha de permetre la manipulació concurrent del mateix objecte en tant que possible. Aquest punt és rellevant si l'aplicació cal que suporti tècniques d'enginyeria concurrent.
9. L'SGBD ha de suportar la realització d'operacions llargues i costoses amb els objectes.

Tot seguit s'analitzen cada un dels requisits amb més profunditat.

3.1 Control de la integritat semàntica

Donada la complexitat dels objectes que cal emmagatzemar en una BD de CAE, el problema de mantenir la integritat semàntica presenta una relevància notable. Tot i que el problema de la integritat semàntica també existeix en aplicacions convencionals, en les aplicacions de CAE es veu amplificat. Vegem-ne alguns exemples ben coneguts:

- Assegurar que una determinada peça mecànica verifica les restriccions geomètriques imposades pel lloc on ha d'anar encastada. Per exemple, un cigonyal ha de tenir una geometria tal que permeti encabir-lo en el lloc corresponent de la bancada del motor. Cal assegurar que la informació de montatge és coherent amb la geometria particular de cada peça.
- Seguint amb CAE mecànic, assegurar que una peça dissenyada podrà ésser mecanitzada amb els mecanismes de que es disposa i les matèries primeres existents. Per exemple, un perfil de planxa cal que tingui unes mides que no excedeixin les de les planxes servides pels trens de laminat o que el seu mecanitzat sigui possible comptant amb el conjunt de màquines eina disponibles.
- Assegurar que una certa construcció té la resistència requerida. Per exemple, una tapa d'un calderí d'aire a pressió, cal que resisteixi la pressió de seguretat del calderí en qüestió.

Tots els exemples vistos, plantegen problemes d'integritat comuns a la indústria. És obvi que aquestes restriccions són molt lluny de les expresables, per exemple, en DB2 ([Dat90]). Les dificultats que ofereixen són:

- El veure si un objecte és semànticament íntegre pot involucrar molta informació provinent de fonts molt diverses. Per exemple:
 - Altres objectes emmagatzemats en la BD. Aquest és el cas del cigonyal.
 - Informació provinent de BD tecnològiques i normatives com en el cas dels perfils de planxa.
 - Informació provinent de processos de simulació com en el cas de la tapa de calderí.
- El detectar la integritat, fins i tot tenint la informació necessària, pot no ser un procés simple. En molts dels casos caldria un llenguatge Turing-equivalent per expressar la restricció i el cost d'avaluar-la podria ser molt significatiu. Pensi's, per exemple, en el cas del calderí, possiblement per estimar-ne la resistència caldria practicar un procés de simulació mitjançant elements finits.
- En alguns casos, el control de la integritat fins i tot és difícilment automatitzable. Factors com la qualitat estètica, que són molt rellevants en alguns productes, són difícilment avaluables de manera algorítmica. En conseqüència, no és senzill automatitzar el manteniment d'aquestes propietats.

3.2 Preservació de la informació històrica

Tal i com hem vist a la caracterització dels sistemes CAE, els objectes manipulats per aquestes aplicacions, estàn immersos en un cicle de vida que s'estèn en el temps. La informació de tots els estats pels quals un objecte ha anat passant s'anomena *informació històrica*. Hi ha un cert consens en la majoria d'autors respecte la necessitat de conservar aquesta informació per als objectes dissenyats (vegeu [Ng91] i [Ket87] per exemple). No obstant això, aquesta informació ha d'ésser organitzada entorn un model respecte el qual no hi ha acord.

Per la seva completesa, aquí usarem el model que es presenta a [Kat90] com a referència. Tot seguit en describim les principals característiques.

Anomenem *manegament de versions* al conjunt de conceptes d'organització i mecanismes operatius usats per treballar amb conjunts d'informació organitzats jeràrquicament que estàn sotmesos a un cicle de vida.

En aquest context, anomenem *model de dades* al conjunt d'estructures de dades que permeten emmagatzemar la informació que volem manipular. Al conjunt d'operacions aplicables als objectes citats l'anomenem *model operacional*.

Un *objecte de disseny* és un agregat de dades tractades de manera solidària per una aplicació. Aquesta informació evoluciona en el temps —cicle de vida—. Definim *versió* a una "fotografia" semànticament significativa d'un objecte de disseny en un cert instant de temps.

Definim *configuració* com la definició jeràrquica d'un objecte de disseny a partir de certes versions d'altres objectes de disseny. En certa manera, podríem dir que una configuració és un binding entre una versió d'un objecte de disseny i les versions dels seus components.

Vistes aquestes definicions, Katz presenta un conjunt de requeriments bàsics que un model de versions ha de satisfer i acaba detallant un model particular derivat del *Version Server*. Vegem quin és el model que presenta.

El model de Katz està fonamentat en dos conceptes: objectes de disseny i interrelacions entre ells. En certa manera, el model podria veure's com un model E-R on les entitats són els objectes de disseny i les interrelacions del model Katz corresponen a interrelacions entre aquestes entitats. No obstant i això, com veurem, el model Katz defineix interrelacions més complexes que les que podríem definir amb el model E-R.

Katz distingeix dos tipus d'objectes rellevants. Per un cantó tenim els objectes *representacionals*. Aquests són els objectes o entitats del model que contenen informació referent al què hi ha representat. Per altre cantó, tenim els objectes anomenats *estructurals*. Aquests contenen informació respecte el com està estructurada la informació. Fent un paralelisme amb un *file-system*, diríem que els fitxers —entitats— poden ser de dos tipus: directoris —estructurals— o fitxers regulars —representacionals—. La part estructural del model queda definida pel propi model. Per altra banda, la part representacional és responsabilitat de les aplicacions que interaccionen amb el mateix.

Tres són les interrelacions que defineix el model de Katz. Cada una d'elles suporta un dels següents conceptes: jerarquies de components, història de les versions i equivalències entre objectes. Vegem alguns aspectes d'aquestes interrelacions.

Les jerarquies de components són suportades per la interrelació *és-part-de*. Aquesta interrelació lliga una instància d'una entitat amb instàncies de les entitats components. Hi ha dos casos d'aquesta interrelació que cal permetre especialment. El primer es dona quan una instància d'una entitat és compartida per dues instàncies compostes diferents o per la mateixa. L'altre resulta de considerar diverses instàncies essent part d'una mateixa instància composta. En certa manera, els arcs d'aquesta interrelació són els que distingeixen les instàncies entre si.

Les entitats són, per Katz, tipificades. La interrelació *és-part-de* pot agregar entitats de qualsevol tipus per formar una entitat composta. Podem pensar com a exemple en una entitat document que agrega entitats de tipus text i de tipus imatge —diguem fotografies—. En aquest cas particular, els dos casos anteriorment citats, es traduirien en tenir una mateixa fotografia a dos capítols diferents o en tenir una mateixa fotografia duplicada en el mateix capítol.

La història d'un cert objecte és suportada mitjançant les interrelacions *és-una-classe-de* i *es-deriva-de*. La interrelació *es-deriva-de* s'encarrega de mantenir la història durant el temps d'un cert objecte associant a una certa versió d'un objecte una versió filla del mateix. En aquest arbre, els nodes germans són anomenats *alternatius* i els nodes fills *derivats*.

Cada una de les versions d'un mateix objecte està lligada a una entitat estructural —essencialment abstracte— que representa l'objecte en si. Aquesta interrelació és anomenada *és-una-classe-de*. Per exemple, qualsevol versió particular

de l'objecte "biela" és una classe de l'objecte abstracte "biela".

La combinació ortogonal de les interrelacions associades al versionat d'objectes i les associades a les jerarquies de components produeixen com a resultat les configuracions. Una versió determinada d'un objecte compost, està lligat (*binding*) a unes versions particulars dels objectes components. Les diferents polítiques de *binding* possibles, determinen diferents comportaments de les configuracions davant l'evolució dels objectes en el temps. Essencialment hi ha dues grans polítiques: el *binding* estàtic, que fixa a quines versions d'objectes components es lliga un objecte compost; i el *binding* dinàmic, en la qual les versions components es determinen dinàmicament en temps d'accés a l'objecte compost.

La darrera interrelació que defineix el model de dades és l'anomenada *és-equivalent-a*. Aquesta interrelació lliga una entitat estructural a les diferents representacions de la mateixa que poden existir. Per posar un exemple, lliga l'entitat abstracte "tiristor" amb diverses representacions de la mateixa, diguem "esquema del tiristor" i "geometria 3D del tiristor".

A més del model de dades descrit, Katz presenta el model operacional. Aquest integra el manteniment de la versió corrent, el suport a configuracions dinàmiques, el model de *workspaces*, els mecanismes de propagació de canvis i restriccions i l'herència.

Amb la descripció d'aquest model hem donat una visió àmplia dels requeriments necessaris per mantenir la informació històrica dels objectes CAE.

3.3 L'estructura complexa dels objectes

Els objectes manipulats en un sistema de CAE poden arribar a tenir una estructura de gran complexitat. Potser l'exemple paradigmàtic són les estructures de dades usades per modelar la topologia i geometria d'escenes tridimensionals. A la bibliografia poden trobar-se multitud de propostes [Sam90], [Män88] o [Fol90]. La figura 1 mostra un diagrama E-R de l'estructura proposada a [Män88]. Aquesta estructura és usada per emmagatzemar un model de fronteres d'un objecte polièdric. Dins un sistema de CAE, aquesta és únicament una petita part de la informació emmagatzemada. També cal tenir en compte moltes altres informacions com propietats físiques, plans de montatge, informació organitzativa, etc. (vegeu per exemple [Har85] o [Roy88]). Podriem dir que una aplicació CAE es caracteritza per la gran quantitat d'entitats manipulades i per la gran quantitat d'interrelacions entre elles.

Aquest fet incideix en tres aspectes fonamentals d'un SGBD:

- Potència descriptiva del DDL.
- Potència de les operacions d'accés.
- Eficiència de l'accés a les dades.

Pel que a la potència descriptiva del DDL, la principal dificultat sorgeix de la necessitat de descriure estructures de dades molt complexes i no fàcilment reduïbles

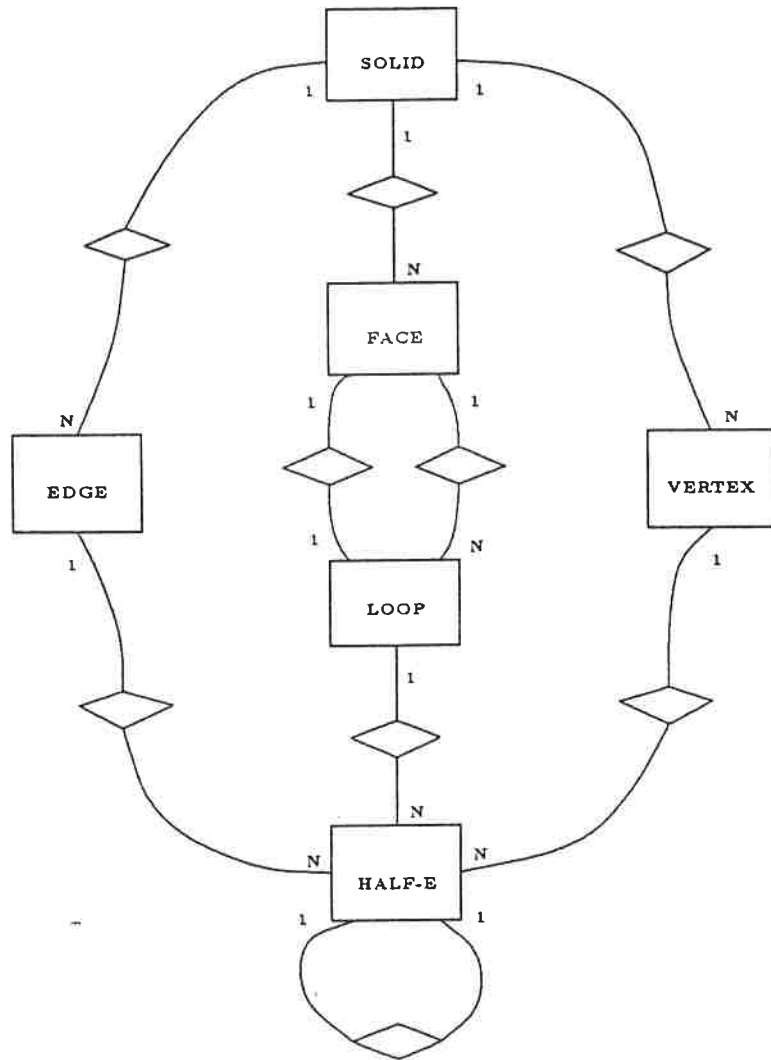


Fig. 1: Diagrama E-R de l'estructura de dades presentada a [Män88]

a taules. L'efecte que es produeix quan una estructura de dades cal reduir-la a taules és conegut com *efecte d'aplanament*. Aquesta representació "aplanada" tampoc és aquella que l'aplicació desitja veure. Això implica una feina molt notable de canvi de format per tal que una aplicació pugui fer ús de les dades que un SGBD li subministra. Aquest problema és conegut com el problema de la *impedància* entre una aplicació i un SGBD.

La potència de les operacions d'accés requerida és conseqüència immediata de la complexitat de les dades. Quan les dades no tenen una estructura lineal, hi ha una quantitat de patrons d'accés molt elevada: accesos seqüencials, recursius, a una jerarquia completa o a una part, etc. A [Kem87] pot trobar-se un recull d'exemples d'ús d'operacions d'accés diverses. Breument, les principals dificultats

amb que hom es troba són les següents:

- L'accés a estructures recursives. Moltes de les estructures accedides són eminentment recursives. Vegeu per exemple [Sam90]. Per tant, es requereixen operacions d'accés que facilitin l'accés recursiu.
- La gran variabilitat de la granularitat de l'accés. És molt freqüent tenir patrons d'accés per a un determinat objecte que tant aviat requereixen accedir a una part molt concreta de l'objecte com a la majoria de l'objecte. Per exemple, dins un procés de visualització, solament cal accedir a les arestes dels objectes de la escena si visualitzem en filferros. En canvi, pel mateix procés de visualització, un algoritme de scan-line amb suavitzat, requerirà l'accés a nivell cares i interrelacions entre cares.
- L'accés a les entitats emmagatzemades, sovint té un caire navegacional per la pròpia natura dels algoritmes que en fan ús. Aquesta navegació es tradueix dins un sistema relacional en una gran quantitat d'interrogacions molesta de manipular i ineficient.
- L'existència d'una gran quantitat d'interrelacions entre entitats. Algunes vegades aquestes interrelacions tenen estructura jeràrquica, com és el cas de l'explosió d'un cert mecanisme en un conjunt de submecanismes. D'altres, la relació pot tenir un patró més general, pensi's per exemple en les interrelacions de connectivitat entre canonades [Jua93] (veure fig. 2).

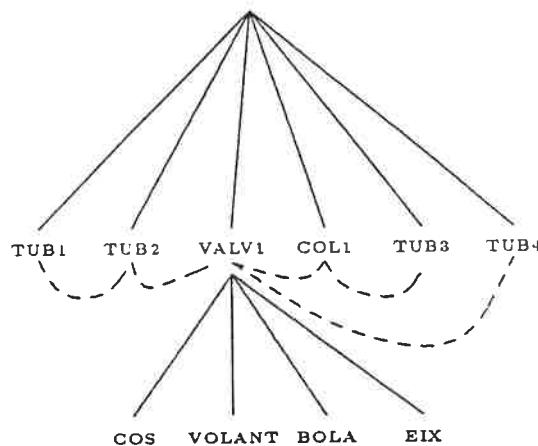


Fig. 2: Exemple d'interrelacions entre objectes

Per últim tenim el problema de l'eficiència. Aquest problema no és particularment causat per el requisit que analitzem sinó que també altres requisits afecten al mateix. Analitzem aquí aquells problemes d'eficiència derivats de la complexitat dels objectes. Essencialment es tracta de dos problemes:

- Les conseqüències de l'efecte aplanament.
- Les conseqüències de l'efecte "còpia de la resposta".

L'efecte aplanament, del que hem parlat dins d'aquest mateix punt, es produeix quan cal reduir a taules una estructura de dades complexa (graf, jerarquia). En certa manera diríem que és el cost a pagar per poder salvar el salt semàntic existent entre el model de dades d'un SGBD tradicional i el model de dades d'un llenguatge de programació modern. La conseqüència d'aquest efecte és doble. Per un cantó, les dades es dispersen en múltiples taules i per tant el cost d'operar amb la informació original augmenta —conforme augmenta el nombre de joins necessaris per obtenir-la—. Per altre cantó, al passar d'un model més ric a un model més pobre, es perd informació. En particular, es perd la informació sobre quines entitats són "afins" a altres entitats. Per exemple, si el model original era jeràrquic, es poden perdre les relacions de germà, o les de pare-fills o fills-pare. Aquesta informació no apareix en el model de dades de l'SGBD. La conseqüència és que els algorismes per optimitzar el clustering no la poden usar i, per tant, baixa notablement el rendiment del sistema.

L'efecte "còpia de resposta" fa referència al fet de que l'SGBD relacional sempre obté com a resultat d'una interrogació una taula, taula que és copiada a l'àrea de l'usuari per que pugui disposar-ne. Si les taules tenen mides raonables o be si les interrogacions són escases, aquesta còpia local pot no produir masses inconvenients. Si, com és el cas, les interrogacions són molt freqüents això esdevé un factor d'ineficiència. Pensem, de nou, en el repintat d'una escena. El volum de dades que cal copiar cada vegada és tan gran que sens dubte és un tant per cent molt significatiu del temps de repintat.

A banda d'aquesta constatació, la complexitat dels objectes, hom pot preguntar-se: fins a quin punt l'estructura dels objectes manipulats per una aplicació de CAE ha d'ésser visible a l'SGBD? Si l'estructura d'aquests objectes fos totalment opaca a l'SGBD, aquest requisit deixaria de tenir sentit. Hi ha, però, fortes raons que aconsellen no prendre aquesta decisió. Citem entre altres:

- L'opacitat de l'estructura d'un objecte impossibilita a l'SGBD assegurar la integritat interna de l'objecte. Per exemple, no es podria verificar que l'objecte és topològicament íntegre.
- L'opacitat de l'estructura d'un objecte impossibilita que l'SGBD pugui mantenir condicions d'integritat entre components de l'objecte i altres objectes emmagatzemats a ls BD. Per exemple, no es podria verificar que una certa cara d'un sòlid té un color admissible.
- L'opacitat de l'estructura d'un objecte impossibilita l'accés selectiu a algunes de les entitats que el componen. Per exemple, si volem dur a terme una cerca sobre els vèrtex d'un objecte, fóra desitjable únicament consultar aquests.

3.4 Suport a tipus d'objectes de mida gran

Dins l'apartat anterior hem estat parlant d'objectes estructuralment complexos. En aquest apartat, es tracta d'analitzar les conseqüències que es deriven del volum dels objectes i no pas per la seva complexitat.

Tot i que, generalment, objectes estructuralment complexes tenen un cost en volum elevat, també és possible haver de treballar amb objectes estructuralment simples però amb un gran volum. L'exemple paradigmàtic són les imatges. Tot i tractar-se de seqüències de píxels, habitualment tenen mides de l'ordre de centenars de Kb.

És a conseqüència d'aquest fet que es requereix el suport, per part de l'SGBD, d'objectes de mida molt gran. Típicament, els tipus de dades suportats per un SGBD tradicional són tipus elementals (enters, reals, cadenes de caràcters) i sovint de mida fixa. Amb aquests tipus és inviable la representació d'objectes voluminosos. Ampliar el conjunt de tipus que un SGBD admet no és solució suficient. Cal tenir en compte que alguns dels mecanismes més importants d'un SGBD tradicional s'arrepengen sobre la suposició d'usar objectes petits. Els principals punts febles respecte aquest requeriment són:

- Impossibilitat per definir nous tipus de dades per a usar dins del DDL. Especialment tipus de dades estructurats. S'enten que quan es parla de tipus de dades s'inclouen les operacions practicables sobre els mateixos. Operacions, aquestes, que s'han d'integrar completament dins el DML de l'SGBD.
- Impossibilitat per definir tipus de dades de mida dinàmicament variable. Quan parlem de tipus de dades de mida reduïda, el guany en espai que podríem obtenir de la dinamicitat pot no ser prou gran, però quan parlem de tipus de dades voluminosos, aquest pot ser un factor molt important.
- Augment de la ineficiència de les polítiques de logging. El fet que les dades manipulades siguin molt voluminoses fa que les actualitzacions del log siguin molt costoses.
- Augment de la ineficiència de les operacions que impliquen replicar les entitats.

3.5 Equivalència d'objectes

És una pràctica usual en CAE el representar una entitat del món real de distintes maneres simultàniament [Kat83]. Els objectius poden ser diversos, però habitualment estàn lligats a obtenir representacions adequades per a cada etapa del procés de disseny i manufactura o bé a facilitar el propi procés de disseny permetent un refinament top-down de l'entitat.

Trobariem molts exemples a qualsevol aplicació. Vet-aquí alguns d'ells.

- En una aplicació de disseny mecànic, és habitual treballar amb un model 3D de l'entitat real. Quan aquest disseny cal passar-lo a taller per construir una maqueta s'ha d'obtenir un plànol, és a dir un altre model de la mateixa entitat real. Si la maqueta és satisfactòria i cal posar l'entitat en producció, pot-ser caldrà generar un programa de CN per a mecanitzar l'objecte, el programa de CN és un altre model de la mateixa entitat real. Tenim doncs, una única entitat real i tres models diferents de la mateixa, models que, en un cert sentit, són equivalents ja que representen la mateixa entitat (vegeu fig. 3).

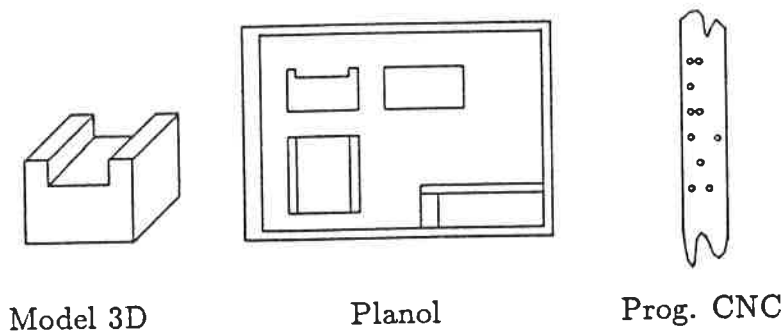


Fig. 3: Entitats equivalents

- Un exemple diferent el podem trobar en una aplicació per al disseny de circuits electrònics. En aquest cas, una primera fase de disseny pot produir com a resultat un esquema de blocs i la connectivitat entre els mateixos. Aquest és un model de l'entitat real. En una segona etapa de disseny, podem obtenir un esquema detallat del circuit. Tenim un segon model de l'entitat real. Més a més, però, aquest segon model és un refinament del primer (vegeu fig. 4).

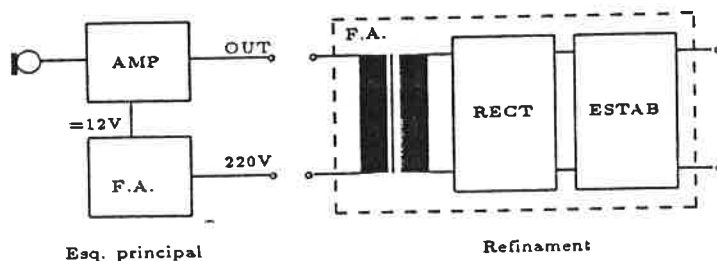


Fig. 4: Refinament d'entitats

La necessitat de suportar el modelat d'entitats equivalents prové de les fortes relacions d'integritat semàntica que lliguen dues entitats equivalents. Hi ha dues classes de relacions semàntiques que destaquen:

- Les relacions d'identificació. Mitjançant aquestes, s'identifiquen elements equivalents de diferents entitats. Un canvi a qualsevol dels elements identificats probablement desencadena un canvi de la resta d'elements identificats.
- Les relacions de refinament. Aquesta relació es dona quan un element o conjunt d'elements és un refinament d'un tercer. En aquest cas, l'entitat refinada ha de satisfer les restriccions imposades per l'entitat de més alt nivell.

En alguns casos, el manteniment d'aquestes restriccions pot fer-se de manera automàtica —com la generació d'una imatge a partir d'un model tridimensional—, en altres de manera semiautomàtica —com la generació d'un plànol acotat— que requereix la intervenció d'un operador i en alguns de manera totalment manual, tal és el cas del refinament d'un amplificador operacional.

Aquest requisit pot veure's com una particularització, sobre un cas de notable

importància, del problema general del manteniment de la integritat semàntica. Malgrat això, s'ha inclòs com a requisit aïllat per la seva importància i perquè introdueix interrelacions importants a tenir en compte en el model de dades de l'SGBD.

3.6 Accés interactiu

Moltes de les aplicacions CAE són eminentment interactives. En conseqüència, cal que l'SGBD suporti l'interactivitat àmpliament. El suport de l'interactivitat té dos aspectes diferenciats. El primer fa referència al fet intrínsec de l'accés interactiu i el segon a els requeriments de velocitats imposats per l'interacció.

El fet intrínsec de tenir un accés interactiu implica que és l'usuari de l'aplicació qui marca el ritme i qui decideix quina acció realitza a cada instant. L'usuari és, abans que res, impredecible. Aquest fet condiciona profundament l'ús que fa de la BD en molts aspectes. Tots aquells mecanismes de l'SGBD que heurísticament presuposen un comportament determinista veuen minvada la seva efectivitat. Alguns d'aquests mecanismes són fonamentals en un SGBD:

- Les polítiques de *pre-fetch* en són un exemple. Es fonamenten en el fet que si durant l'instant t_i s'ha accedit a la informació I_j , és molt probable que durant l'instant t_{i+1} sigui necessari accedir a la informació I_{j+1} . Quan això, però, depèn de l'usuari no pot fer-se aquesta hipòtesi amb seguretat.
- Un altre exemple es troba en les polítiques de control de la concurrència. En molts casos, s'assumeix que les transaccions seràn llençades en batch i no de manera interactiva per un usuari. En el darrer cas, no pot preveure's en quin moment acabarà la transacció ni és convenient abortar-la pel manegador de transaccions de manera automàtica. En conseqüència, les polítiques que fan ús del bloqueig de recursos o de la tècnica d'abortar transaccions per evitar el bloqueig del sistema no poden ser usades.

El segon cas lligat al fet interactiu és el requeriment de velocitat. La majoria de SGBD's es dissenyen sota l'òptica d'augmentar el *throughput* de transaccions. És a dir, el nombre de transaccions per unitat de temps que s'admeten en el sistema. En el cas d'aplicacions interactives, existeix un requisit més important que l'anterior: disminuir el temps mig per transacció i la desviació d'aquest mateix paràmetre [Mai89]. Requisit que és fonamental per poder assegurar un temps de resposta adequat per a la interacció.

3.7 Gestió interactiva de la integritat

Tal i com s'ha vist abans, les restriccions d'integritat a que es veuen sotmesos els objectes en un sistema de CAE són molts i complexes. Per altra banda, les operacions de disseny esdevenen llargues en el temps, conduint a transaccions de la mateixa durada o durada superior. El fet que un usuari no pugui veure verificada la coherència semàntica de l'objecte que dissenya fins el COMMIT pot tenir un

cost massa elevat. No cal dir que abortar el COMMIT per que hi ha violacions de la integritat és absolutament prohibitiu. Encara més, aquest rebuig del COMMIT per raons d'integritat cal que sigui tant explícit com es pugui i aporti la major informació possible per suportar la rectificació del disseny. Direm, doncs, que la verificació de la integritat d'un objecte cal que pugui fer-se interactivament.

Per interactiu, no s'entèn que automàticament i de manera transparent a l'usuari es duguin a terme verificacions. Tal punt és indesitjable: cal tenir en compte que és freqüent aquell cas en que per arribar a un disseny íntegre abans cal passar per dissenys intermedis sense integritat. Verificar la integritat durant aquests intervals no tindria cap sentit. Per altre cantó, aquests intervals són únicament coneguts pel propi usuari amb la qual cosa solament és l'usuari qui pot indicar els moments en els que cal fer una verificació de manera coherent. Per tant, per interactiu cal entendre "a petició de l'usuari".

Evidentment, en temps de COMMIT cal que es dugui a terme sempre una verificació semàntica. Si estem pensant en un model de transaccions jeràrquiques, llavors hi ha múltiples COMMIT's cada un dels quals fa la verificació corresponent. Es podria pensar que en aquest cas el mecanisme automàtic és suficient però caldria un estudi més acurat respecte el tema.

3.8 Manipulació concurrent de l'objecte

Si be aquest és un objectiu fonamental de qualsevol SGBD, en aquest cas concret pren una especial rellevància i dificultat. Els objectes a manipular durant un disseny són objectes de gran complexitat i per tant el seu disseny comporta una durada considerable (hores, dies, setmanes). Si les polítiques pel control de la concurrència es fonamenten en el bloqueig, llavors resulta que la disminució del paral·lisme és molt elevada. Encara més, sovint és necessari que diverses aplicacions puguin accedir simultàniament sobre el mateix objecte per tal de poder accelerar el procés de disseny. Algunes vegades, la pròpia jerarquitzaió de l'objecte indueix una partició natural en subobjectes que facilita l'accés concurrent al disminuir la granularitat. Hi ha altres casos, però, en que aquesta partició no existeix o bé no és adequada per a la distribució de les diferents tasques. Alehores solament el treball simultani sobre el mateix objecte o la partició fictícia del mateix són solucions viables. Tot i això, no cal perdre de vista que la partició fictícia comporta una feina no despreciable de "manteniment de la coherència" que ve a agreujar el problema de la integritat.

3.9 Suport per a operacions llargues i costoses

Les operacions relacionades amb el disseny són sovint llargues. Això fa que el model de transaccions usat habitualment no sigui el més adequat. A la literatura es defineix aquest tipus de transacció com a "transacció de disseny". Les transaccions de disseny són diferents essencialment per dues raons:

- la primera perquè sovint es tracta d'un procés interactiu de prova i error conduït humanament.
- La segona perquè algunes de les operacions que s'apliquen poden ser molt costoses (per exemple les operacions booleanes).

Aquesta realitat té com a conseqüents divesos fets:

- Trencar una de les suposicions en que es fonamenten alguns dels mecanismes de control de la concurrència dels SGBD's clàssics: la brevetat d'una transacció.
- Quan una transacció és llarga —potser de dies o setmanes—, pot fer-se necessari compartir dades encara no confirmades amb altres transaccions. El motiu més usual és permetre augmentar el grau de paral·lelisme en les etapes de disseny d'un cert producte (enginyeria concurrent).
- Abortar una transacció com a resposta a haver infringit una regla d'integritat o a evitar una situació de abraçada mortal te un cost tant elevat que resulta inviable.
- La regla clàssica de recuperar l'estat de l'SGBD, en cas de fallida, a transacció completa té un cost massa elevat. Les transaccions llargues que es podrien perdre en aquest procés poden significar un cost massa elevat.

Aquest tema està extensament tractat, amb un recull de les propostes més interessants, al informe [Bar91].

4 Conclusions

En aquest informe s'ha presentat la problemàtica que hom ha d'afrontar quan es veu la necessitat d'integrar un SGBD a un sistema CAE. S'han exposat nou requisits que un SGBD hauria de satisfer com a conseqüència de les característiques pròpies de les aplicacions CAE. Tot i que cada un dels requisits és prou important per se, aquests requisits poden ser agrupats en quatre grans blocs que es poden enunciar com:

1. Preservació de la informació històrica del cicle de vida del producte.
2. Eines per assegurar la integritat semàntica prou potents.
3. Suport pel tractament d'entitats grans i complexes.
4. Adaptació al requeriment d'interactivitat i immersió en un procés de disseny.

Tot i que s'està treballant en diversos fronts per aconseguir que els SGBD's satisfacin aquests requeriments, els propis requeriments no han estat prou estudiats ni clarament determinats fins el moment.

Bibliografia

- [Atk89] M. Atkinson, D. DeWitt, D. Maier, F. Barcillon, K. Dittrich and S. Zdonik, "The Object-Oriented Database System Manifesto", *ALTAIR Technical Report*, no. 30-89, GIP ALTAIR, LeChesnay, France.
- [Bañ91] R. Bañares Alcàntara, "Representing the Engineering Design Process: Two Hypotheses", *Computer Aided Design*, vol. 23, no. 9, November 1991, pp. 595-603.
- [Bar90] T. Barsalou and G. Wiederhold, "Complex Objects for Relational Databases", *Computer Aided Design*, vol. 22, no. 8, October 1990, pp. 458-468.
- [Bar91] N.S. Barghouti and G.E. Kaiser, "Concurrency Control in Advanced Database Applications", *ACM Computing Surveys*, vol. 23, no. 3, September 1991, pp. 269-317.
- [Car90] M. Carey and L. Haas, "Extensible Database Management Systems", *SIGMOD RECORD*, vol. 19, no. 4, December 1990, pp. 54-60.
- [Cha90] E.E. Chang and R.H. Katz, "Inheritance in Computer-Aided Design Databases: Semantics and Implementation Issues", *Computer Aided Design*, vol. 22, no. 8, October 1990, pp. 489-499.
- [Cod79] E.F. Codd, "Extending the Database Relational Model to Capture More Meaning", *ACM Transactions on Database Systems*, vol. 4, no. 4, December 1979, pp. 397-434.
- [Dat90] C.J. Date, *An Introduction to Database Systems vol. 1*, 5^a edició, Addison-Wesley 1990.
- [Dul88] S. Duli and V. Milanese, "A DataBase Design for Graphical Models", *ACM SIGPLAN Notices*, vol. 25, no. 12, December 1988, pp. 30-36.
- [Dür89] M. Dürr, M. Huck, A. Kemper, P. Mohrholz and M. Wallrath, "Using Conventional and Nested Relational Database Systems for Modelling CIM Data", *Computer Aided Design*, vol. 21, no. 6, August 1989, pp. 379-392.
- [Fol90] J.D. Foley, A. van Dam, S.K. Hughes, *Computer Graphics. Principles and Practice*, 2^a edició, Addison-Wesley 1990.
- [Har85] D.J. Hartzband and F.J. Maryanski, "Enhancing Knowledge Representation in Engineering Databases", *IEEE COMPUTER*, September 1985, pp. 39-48.
- [Har87] M. Hardwick and D.L. Spooner, "Comparison of Some Data Models for Engineering Objects", *IEEE Computer Graphics and Applications*, March 1987, pp. 56-66.

- [Har90] M. Hardwick, W. Uejio and D.L. Spooner, "ROSE and CHIDE: User Interface Management System Implementation as Object-Oriented Database System Application", *Computer Aided Design*, vol. 22, no. 8, October 1990, pp. 480-488.
- [Jos90] J.V. Joseph, S.M. Thatte, C.W. Thompson and D.L. Wells, "Object-Oriented Databases: Design and Implementation", *Proceedings of the IEEE*, vol. 79, no. 1, January 1991, pp. 41-64.
- [Jua93] R. Ruan, L. Solano, J. Vega, F. Alonso, J. Subías, "Sistema DMI. Modelado Geometrico de Tuberías", *Proceedings CEIG'93*, Junio 1993.
- [Kat83] R.H. Katz, "Managing the Chip Design Database", *IEEE COMPUTER*, December 1983, pp. 26-36.
- [Kat90] R.H. Katz, "Toward a Unified Framework for Version Modeling in Engineering Databases", *ACM Computing Surveys*, vol. 22, no. 4, December 1990, pp. 375-408.
- [Kem87] A. Kemper and M. Wallrath, "An Analysis of Geometric Modeling in Database Systems", *ACM Computing Surveys*, vol. 19 no. 1, March 1987, pp. 47-91.
- [Ket87] M.A. Ketabchi and V. Berzins, "Modeling and Managing CAD Databases", *IEEE COMPUTER*, February 1987, pp. 93-102.
- [Kim90] W. Kim, J. Banerjee, H. Chou and J. Garza, "Object-Oriented Database Support fo CAD", *Computer Aided Design*, vol. 22, no. 8, October 1990, pp. 469-479.
- [Kor93] H.F. Korth, A. Silberschatz. *Fundamentos de Bases de Datos*, 2ª edició, McGraw-Hill 1993. (Traducció de Database Systems Concepts).
- [Mai89] D. Maier, "Making Database Systems Fast Enough for CAD Applications", in *Object Oriented Concepts, Databases and Applications*, W. Kim and F. H. Lochovsky eds., ACM Press 1989, pp. 573-582.
- [Män88] M. Mäntylä, *An Introduction to Solid Modeling*, Computers Science Press 1988.
- [Ng91] W.Y. Ng, "Generalized Computer-Aided Design System: A Multiobjective Approach", *Computer Aided Design*, vol. 23, no. 8, October 1991, pp. 548-553.
- [Pec88] J. Peckham and F. Maryanski, "Semantic Data Models", *ACM Computing Surveys* vol. 20, no. 3, September 1988, pp. 153-189.
- [Roy88] U. Roy and C.R. Liu, "Establishment of Functional Relationships Between Product Components in Assembly Database", *Computed Aided Design*, vol. 20, no. 10, December 1988, pp. 570-580.
- [Sam90] H. Samet, *The Design and Analisis fo Spatial Data Structures*, Addison-Wesley 1990.

- [Spo88] D.L. Spooner and M. Hardwick, "A Conceptual Framework for Data Management in Mechanical CAD", in *Geometric Modeling and Applications*, M.J. Wozny, H.W. McLaughlin and J.L. Encarnação eds., Elsevier Science Publishers 1988, pp. 317-329.
- [Sto90] M. Stonebraker, L.A. Rowe, B. Lindsay, J. Gray, M. Carey, M. Brodie, P. Bernstein and D. Beech, "Third-Generation Database System Manifesto", UC Berkeley Tech. Rep. UCB/ERL M90/28, April 1990.
- [Wie86] G. Wiederhold, "Views, Objects, and Databases", *IEEE COMPUTER*, December 1986, pp. 37-44.
- [Wol87] R.N. Wolfe, M.A. Wesley, J.C. Kyle, F. Gracer and W.J. Fitzgerald, "Solid Modeling for Production Design", *IBM Journal of Research and Development*, vol. 31, no. 3, May 1987, pp. 277-295.