



IMPLEMENTACIÓ D'EINES CAD PER A CIRCUITS DE POTÈNCIA EN ENTORN WINDOWS

Alex Méndez, Josep Bordonau, Joan Peracaula

Microsoft Windows és un entorn de finestres multitasca per ordinadors personals executable amb PC-DOS i MS-DOS. Aquest entorn gràfic permet una millora considerable de les possibilitats de desenvolupar eines de CAD per ordinadors personals. En aquest document es descriuen aquestes millores en un cas concret.

INTRODUCCIÓ.

Encara que Windows existeix des de 1985, han estat les darreres versions, i més concretament la versió 3.1 les que han permès gaudir als usuaris de PC d'un sistema força semblant al *Presentation Manager* de OS/2 per a màquines més grans. Això ha estat en part possible per que les darreres versions aprofiten força els recursos que tenen els microprocessadors 386 i 486 que darrerament s'han anat incorporant als PCs.

En poc temps Windows s'ha convertit en un estàndard força emprat també pels fabricants de software, que veuen com l'entorn Windows els proporciona ja fetes moltes de les prestacions que ells haurien d'incorporar als seus productes.

En aquest document es descriuen els avantatges que suposa utilitzar aquest entorn per desenvolupar aplicacions (programes) de CAD orientades a l'Electrònica de Potència. Les característiques que es comenten s'ha agrupat al voltant dels següents temes:

- **Interface:** El sistema de finestres, els menús, capses de diàlegs, i altres tipus de recursos proporcionen un interface molt més intuïtiu i molt més senzill d'aprendre per part de l'usuari i de fer servir i molt més senzill de desenvolupar donada la seva natura gràfica (les presentacions de resultats i de dissenys queda molt reduïda).

- **Sistema de memòria:** L'organització de la memòria per part de Windows redueix i optimitza força el seu ús, i redueix l'esforç de programació considerablement, comparat amb la programació directa sobre MS-DOS. El sistema de memòria depèn del microprocessador bàsic, essent necessari un 80386 o més avançat per tal de gaudir dels avantatges reals de Windows.

- **Multitasca:** La multitasca és una característica addicional de Windows que permet executar més d'una aplicació a la vegada. La

multitasca només és possible amb màquines com el 80386 o més avançades. En un entorn de CAD la multitasca es una característica imprescindible. Fins ara el fet que els PCs no fossin multitasca ha estat resolt més o menys elegantment. Tanmateix, quan aquesta característica ha estat possible implementar-la les solucions antigues han estat oblidades ràpidament, ja que eren artificioses i molestes per als usuaris.

- **Biblioteques d'enllaç dinàmic:** són biblioteques de dades, codi o recursos que s'incorporen a les aplicacions durant l'execució i que poden ésser compartides per més d'una aplicació. A més poden estar escrites amb altres llenguatges diferents al codi client, donat que el sistema operatiu de Windows estableix uns protocols força ajustats. Aquest tipus de biblioteques no tan sols optimitzen l'ús de la memòria; també permeten una major flexibilitat en el codi que es desenvolupa, en molts sentits, com més endavant s'explicarà.

- **Intercanvi dinàmic de informació:** Com que més d'una aplicació pot estar alhora executant-se, es permeten els intercanvis de dades entre les diferents aplicacions actives a través d'una memòria global que totes comparteixen. Això optimitza molt el nombre d'eines que cal desenvolupar els interrelacions entre elles. Aquesta característica és força important en el desenvolupament d'un sistema de CAD.

- **Biblioteques incorporades del sistema:** Windows proporciona

ÀLEX MÉNDEZ FERRÉS és enginyer Industrial. Participa en diversos projectes de recerca en el Departament d'Enginyeria Electrònica de la UPC. Actualment gaudeix d'una beca de F.I. de la Generalitat de Catalunya.

JOSEP BORDONAU FARRERONS és professor Titular del Departament d'Enginyeria Electrònica de la UPC des de 1990. Actualment és sots-director de recerca del Departament d'Enginyeria Electrònica de la UPC.

JUAN PERACAULA ROURA és catedràtic del Departament d'Enginyeria Electrònica de la UPC des de 1971. Actualment és responsable del programa de doctorat d'aquest departament.

per ell mateix un nombre considerable de biblioteques que faciliten força el disseny d'aplicacions. Aquestes biblioteques integren força bé totes les característiques que es desitgen, per la qual cosa es poden utilitzar directament en els productes finals, o són la base per desenvolupar-ne de més complexes però basades en aquestes.

- Transferències i compatibilitat: la major part de les aplicacions més importants tenen la seva versió per Windows. Quan es desenvolupa una aplicació Windows, el programador queda obligat a establir mecanismes de transferència amb altres aplicacions Windows. Això suposa que les eines que es desenvolupen són compatibles directament amb les restants aplicacions que tenen una versió per Windows. Com que una de les característiques més desitjables per un sistema de CAD és que sigui compatible amb altres aplicacions alienes, però orientades en camps propers o idèntics, l'entorn Windows és un dels millors candidats.

INTERFACE.

El disseny de l'interface dins de l'entorn Windows ha estat força cuidat, i ha estat dissenyat amb el següent principis com a bàsics:

Control de l'usuari: els usuaris han de controlar les aplicacions, i no les aplicacions als usuaris. Aquest principi té diferents implicacions:

- les aplicacions han d'ésser tant interactives com sigui possible. Els usuaris no han de esperar temps massa llargs. Això es pot resoldre amb la multitasca.

- les aplicacions han de permetre un cert grau de configuració per part dels usuaris, de forma que puguin canviar colors, tamanys i aspectes globals de visualització de les aplicacions.

- el bon interface és el que permet facilitar les tasques, i no cridar l'atenció dels usuaris sobre aquestes. Els usuaris volen realitzar unes tasques amb l'ordinador, i no fer servir l'ordinador per passar el temps.

Directness: cal fer un interface que doni al usuari un camí intuïtiu i directe per aconseguir fer les tasques per les que fa servir l'aplicació. Els paradigmes objecte-acció aconsegueixen

aquest principi. Els usuaris seleccionen objectes (dibuixets, botons, textos) i fan accions amb ells.

Consistència: dos tipus de consistències cal considerar: amb el món real (en fer servir els paradigmes esmentats) i entre les aplicacions desenvolupades (de forma que el coneixem d'una faciliti molt el coneixement de les restants).

Claredat: tots els elements visualitzats han d'ésser sota el punt de vista visual, conceptual i lingüístic immediatament comprensibles i no han de portar a interpretacions ambigües.

Estètica: a més d'ésser directe, un interface que sigui agradable estèticament, facilita la tasca de l'usuari, qui s'hi troba més còmode.

Feedback: Els usuaris han de rebre sempre unes informacions de feedback que els informin sobre les accions que estan portant a terme.

Aprenentatge: Als usuaris els agrada explorar una aplicació en mode prova-error. Per aquest motiu, els programes que admeten que el usuari pugui avançar i retrocedir en les accions que fa, veient els resultats, i que el va informant dels errors que va cometent, són els que més èxit tenen. Es per això que les aplicacions en la mesura que sigui possible han d'admetre aquesta qualitat de l'interface.

Com es pot veure, fent que les aplicacions aconsegueixin aquests principis, n'hi prou per desenvolupar un Sistema de CAD amb un bon interface sense necessitat d'incorporar res de nou al sistema del propi Windows i que estigui màximament integrada dins de l'entorn.

SISTEMA DE MEMÒRIA.

La gestió de memòria que realitza Windows és prou complexa com per que ni tant sols es pugui esmentar aquí. Això es deu bàsicament per la gran varietat de configuracions que poden presentar els diferents equips i pel fet que els processadors de les famílies 80x86 tenen models segmentats de memòria.

El resultat final, però, que veu el programador és una memòria accessible determinada i ben gestionada, on

pot accedir-hi sota unes determinades normes i que fan es pugui oblidar del tipus de màquina que farà anar les aplicacions. Continua havent-hi les limitacions de tamanys de segments, de direccions, de tipus de punters, però tota la problemàtica es veu molt reduïda. Cert tipus d'aplicacions veuen millorades moltes de les seves prestacions de memòria dins de l'entorn Windows.

MULTITASCA.

La multitasca dona la possibilitat de carregar múltiples aplicacions en múltiples finestres, i que es puguin executar simultàniament, és a dir, hi han vàries aplicacions que estan processant dades i informacions simultàniament. Aquest resultat s'aconsegueix mitjançant l'assignació d'interval de temps que el microprocessador estableix entre les diferents tasques que s'estan executant.

Entre les diferents tasques en un moment determinat s'executen es poden establir fàcilment limitacions i directives respecte el temps que el microprocessador estableix per a cada tasca. Per exemple, es poden establir els percentatges de temps que rep respecte la resta d'aplicacions actives quan té el focus, quan no el té, quan està esperant per a rebre entrades, etc.

Aquest terme no s'hauria de confondre amb la multicàrrega. La multicàrrega consisteix en carregar varies aplicacions en memòria però executar-ne només una. Una aplicació carregada en memòria està llesta per ésser activada, però romandrà així fins que no es quedin desactivades totes les restants. La multicàrrega únicament permet guanyar temps i altres molèsties de sortir i entrar en les diverses aplicacions cada cop que es vol commutar entre alguna d'elles.

La multitasca permet, per tant, tenir unes aplicacions on els usuaris poden entrar les dades i visualitzar resultats o rebre altres tipus d'informacions, mentre altres aplicacions s'encarreguen de processar les dades, guardar-les, verificar integritats, etc. També permet visualitzar aplicacions que mostrin resultats anteriors



de dissenys, dades resultants d'altres processos, o aplicacions alienes que facin tasques semblants de forma que els usuaris les puguin comparar.

BIBLIOTEQUES D'ENLLAÇ DINÀMIC.

Dins de l'entorn Windows hi han dos tipus bàsics de programes: les aplicacions i les biblioteques d'enllaç dinàmic. Tant les unes com les altres poden anomenar-se mòduls de codi carregables o executables. Una aplicació pot tenir més d'una vegada carregat el seu codi, perquè és possible, si així ho ha pensat el programador, tenir varies vegades el mateix programa executant-se. Les biblioteques d'enllaç dinàmic només poden carregar-se una vegada, però poden ésser accedides per més d'una aplicació.

Les aplicacions també reben el nom de tasques executables. Cada aplicació sota Windows està dirigida per una tasca i respon a un conjunt d'events externs (missatges). Les DLL (biblioteques d'enllaç dinàmic) reben a vegades el nom de executables sense tasca donat que són dirigides per la tasca que les crida. Les DLL no tenen pila pròpia ni llaç de procés de missatges propi, pel que de fet són biblioteques de funcions i recursos disponibles per a qualsevol aplicació del sistema que les conegui. Les DLL bàsicament donen tres graus de llibertat en el desenvolupament d'aplicacions:

- Les DLL són un lloc excel·lent per situar el codi i les dades que siguin compartides per diverses aplicacions. Queda clar que en un sistema de CAD hi ha força codi que es repeteix o que és emprat per a diferents eines. Suposem una rutina que faci una anàlisi d'harmònics. La mateixa rutina es pot trobar tant en el disseny de filtres com en presentació de resultats de formes d'ona de convertidors.

- També són una bona forma de situar parts del codi que només cal carregar en memòria segons les opcions escollides pels usuaris. Això permet desenvolupar eines molt grans, o que facin usos extensius de la memòria, sense que calgui una màquina amb moltes prestacions, donat que el propi sistema d'aplicacions pot vigilar la

memòria que resta, i alliberar la que no cal en cada moment.

- Permeten afegir parts de codi desenvolupat externament o posteriorment, fins i tot per al propi usuari de les eines de CAD. Aquesta flexibilitat fa que, seguint unes normes determinades, es pugui desenvolupar un sistema d'eines que estigui en constant creixement, i que el disseny original del sistema de CAD únicament proporcioni una estructura llavor.

DYNAMIC DATA EXCHANGE.

Intercanvi dinàmic de informació (normalment referit amb les sigles angleses DDE) es una forma de processos de comunicació entre aplicacions que fan servir la memòria compartida per a intercanviar dades. Es poden utilitzar aquests processos per a transferències d'una dada per comunicació, o per canvis automatitzats en les aplicacions que tenen lligams DDE establerts.

Normalment es parla de les **interaccions client-servidor**. Dins de l'entorn Windows es considera que una transferència de dades sempre s'estableix entre un client i un servidor. Un servidor és l'aplicació que proporciona les dades que es volen transmetre, mentre que un client és l'aplicació que demana les dades. Un servidor pot tenir diferents clients, i un client pot tenir diferents servidors. També pot succeir que una mateixa aplicació sigui a la vegada servidor i client. Com es pot veure, hi ha una gran flexibilitat en les comunicacions entre aplicacions. A més, com que una aplicació pot accedir a informacions sobre quins servidors hi ha en el sistema i quines informacions poden proporcionar, es pot veure que realment les possibilitats que ofereix l'entorn Window són força grans.

Aquesta transferència de dades entre aplicacions té conseqüències pràctiques molt importants:

- s'elimina la duplictat de codi. Pesem per un moment en les bases de

dades. Sota aquest entorn sembla lògic pensar que hi hagi només una aplicació que proporcioni a les restants de forma automàtica totes les dades que requereixen les diferents aplicacions que fan servir aquella base de dades. Sense aquesta característica això no es podria fer d'una forma tant independent i apriorística com el DDE o permet, ja que es pot dissenyar aquesta aplicació sense conèixer les aplicacions que la faran servir.

- la comunicació de resultats entre les diferents aplicacions de CAD es pot realitzar de forma immediata i directa, sense tenir que recórrer a mètodes indirectes (per exemple emmagatzemar els resultats en un arxiu de dades que després una altre aplicació llegirà).

BIBLIOTEQUES INCORPORADES.

Windows incorpora per ell mateix un conjunt de biblioteques que afavoreixen el desenvolupament d'aplicacions sofisticades. Les més destacades pel disseny de CAD són:

Graphics Device Interface (GDI): aquest mòdul s'encarrega d'exportar les funcions Windows que permeten pintar, dibuixar, traçar,

imprimir i gestionar els colors en mode gràfic. Aquestes funcions són molt avançades, i permeten la gestió de sistemes de coordenades, generació automàtica de formes polinomials, gestió de paletes de colors, emmagatzemant vectoritzat de dades gràfiques. Això permet un gran estalvi de codi relacionat amb presentació de resultats i de dades gràfiques.

Serveis de Sistema: són funcions directament accessibles de l'entorn Windows i que permeten accions profundes i avançades en els temes de gestió de mòduls d'aplicacions i biblioteques, gestió del sistema global de memòria, gestió de interrupcions, gestió de tasques, gestió de mòduls de recursos, gestió de comunicació, gestió de fitxers, gestió de depuració de codi i gestió d'optimització de codi.

La multitasca dona la possibilitat de carregar múltiples aplicacions en múltiples finestres, i que es puguin executar simultàniament.

Biblioteca de Gestió de Inter-canvi Dinàmic de Dades: aquesta DLL permet simplificar les tasques de programació dels protocols de comunicació entre les diferents aplicacions. També proporciona funcions per a la gestió de memòria global i de registre dels clients i servidors que té el sistema en cada moment. Permet el desenvolupament d'eines de depuració pels processos de transmissió dinàmica de dades entre aplicacions.

Tool Helper Library: proporciona tot tipus de informació sobre el sistema, pel que fa a aplicacions, recursos i estat de la memòria. Permet també la realització d'eines de depuració en les assignacions de memòria i en la gestió de mòduls en les múltiples instances, de cada aplicació o biblioteca dinàmica..

System Resources Stress-Testing Library: permet el testeig les aplicacions o biblioteques en front de situacions diverses del sistema, on hi han pocs recursos de temps, memòria o espai de disc. Permet veure com reaccionen les diverses aplicacions en front d'aquestes situacions de pocs recursos de sistema.

32-Bit Memory Management Library: permet la gestió de dades directament en registres de 32 bits, ja que els processadors 80386 o superiors tenen aquesta capacitat.

Biblioteca per Aplicacions Internacionals: permet la creació d'aplicacions en diferents idiomes, tenint en compte les particularitats de notacions de cada cas.

Biblioteca d'Aplicacions en Xarxa: porta incorporada biblioteques que permeten gestionar el funcionament de les diverses aplicacions que s'estan executant en cada moment a dins de la xarxa a la qual estan connectats.

TRANSFERÈNCIES I COMPATIBILITAT.

Windows estableix un conjunt de formats per a cada tipus de dada que es vol transmetre. Així hi han formats generals pels textos, per les imatges

rasteritzades, per les imatges vectoritzades, per les unitats de codis, etc. Normalment quan es desenvolupen aplicacions dins de l'entorn Windows es respecten aquests formats per tal que els programes desenvolupats s'integrin plenament dins de l'entorn Windows. Tanmateix, fins i tot definint formats nous de dades, les compatibilitats queden garantides pels protocols DDE (encara que l'aplicació client desconegui quin es el format de les dades que li són enviades, les pot rebre finalment com a dades estàndard del sistema gràcies al sistema de traducció proporcionat per DDEML).

També hi ha un altre tipus de compatibilitat que normalment és obviada però que al programador li estalvia un gran nombre de problemes: les diferents configuracions de hardware dels ordinadors personals. Windows proporciona un sistema comú, prou independent dels diferents tipus de monitors, impressores, plaques base, etc., per que el programador no tingui que perdre el temps fent codis d'interface per a cada configuració. Això ja bé incorporat dins de Windows i el programador fa servir unes funcions d'interface que no

són dependents del dispositiu. Dos exemples són el GDI (*Graphical Device Interface*) i la gestió d'impressores, on es fan servir instruccions generals que no depenen

del tipus de hardware que té la màquina en concret.

CONCLUSIONS.

En aquest document han estat presentats els avantatges de implementar programes de CAD dins de l'entorn Windows. Aquests avantatges es poden resumir en:

- Només cal seguir les normes de interface marcades per Windows per aconseguir un bon interface pel Sistema de CAD.

- La gestió de memòria facilita la utilització per part del programador d'aquesta, i millora les prestacions de les aplicacions que aprofiten

de tots els recursos que Windows proporciona.

- La multitasca que proporciona Windows permet parlar veritablement d'un entorn de CAD en un ordinador personal.

- Les biblioteques d'enllaç dinàmic permeten una optimització de la memòria disponible, tamany de les aplicacions molt més gran del que en principi aparentment seria possible i del creixement del codi amb aportacions del programador o del propi usuari sense tenir que retocar el codi ja existent. Es pot parlar d'un sistema de CAD llavor que va creixent amb aportacions del programador i dels usuaris, a mida que es va utilitzant.

- La transmissió de dades de forma dinàmica permet la creació d'aplicacions servidores que eliminin duplicitats de codi o enllaços, i permet la transmissió directa de dades i resultats sense tenir que recórrer a formes d'emmagatzament transitòries.

- Hi ha un conjunt d'eines pròpies de Windows que es poden incorporar a les aplicacions i que redueixen les tasques que el programador ha de realitzar.

REFERÈNCIES.

[1] *Microsoft Windows Software Development Kit*. WA: Microsoft Press, 1991.

[2] TELLO, ERNEST: *Object-Oriented Programming for Windows*. New York: John Wiley & Sons, 1991.

[3] CLARK, JEFFREY: *Dynamic Data Exchange Programming Windows*. Carmel IN: SAMS 1991.

[4] PETZOLD, CHARLES: *Programming Windows*. 2ª edició. Redwood, WA: Microsoft Press, 1990.

[5] FAISON, TED: *Borland C++ 3 Object-Oriented Programming*. 2ª edició. Carmel IN: SAMS, 1992.

[6] LAUREL, BRENDA: *The Art of Human-Computer Interface Design*. Reading, MA: Addison-Wesley, 1990.

