

STANDARDE ȘI MODELE DE REFERINȚĂ ÎN GRAFICA PE CALCULATOR

ec. Costin Pribeanu

Institutul de Cercetări în Informatică

Rezumat:

Standardele internaționale de grafică pe calculator au avut o influență majoră asupra conceptelor, arhitecturilor și algoritmilor asociați cu programarea grafică, independentă de dispozitiv. În acest sens, articolul propune o privire retrospectivă asupra dezvoltării standardelor de grafică pe calculator din ultimul deceniu și evidențiază tendințele care se manifestă în prezent. Sînt descrise succint modelele de interfață cu programul de aplicație GKS și PHIGS, standardele de metafișier și standardele de interfață cu dispozitivul. O atenție deosebită este acordată modelului de referință în grafica pe calculator, adoptat de ISO în 1992, care are ca obiectiv crearea unui cadru conceptual pentru activitatea de cercetare/dezvoltare în acest domeniu și în disciplinele legate de grafica pe calculator.

Cuvinte cheie: grafica pe calculator, GKS, PHIGS, GKS-3D, CGRM, interfețe cu dispozitive grafice, metafișiere grafice, portabilitate.

1. Introducere

Grafica pe calculator s-a conturat ca o tehnologie distinctă, de prelucrare a informației, cu mult înainte de apariția standardelor oficiale. Perioada de început s-a caracterizat printr-o orientare către sisteme dedicate, fiecare programator dezvoltînd propriul său pachet de rutine grafice cu intenția de a utiliza în mod optim resursele disponibile. Scăderea prețurilor la echipamente grafice și creșterea interesului pentru informația prezentată vizual (grafic) care se adresează percepției umane într-o manieră mai naturală, au condus la dezvoltarea de aplicații complexe în care instrumentele de grafică pe calculator reprezintă o investiție majoră. În aceste condiții, eforturi importante de cercetare au fost îndreptate spre definirea conceptelor de grafică pe calculator și spre studierea cerințelor de integrare a acestora în sisteme de tip CAD, CAM, CAE.

Grupuri de lucru și comitete internaționale, pornind de la experiența diferitelor sisteme grafice, au fost implicate, începînd din 1975, în standar-

dizarea în acest domeniu. Circa 100 de experți din 16 țări au participat la proiectarea și redarea standardului GKS. Un volum de muncă însemnat a fost investit în institute naționale de standardizare și organizații internaționale (ISO, ANSI). Interesul pentru această activitate a fost evidențiat de majoritatea conferințelor și manifestărilor științifice din acest domeniu, atestînd influența majoră pe care standardele de grafica pe calculator au avut-o asupra conceptelor, arhitecturilor și algoritmilor asociați cu programarea sistemelor grafice.

Conferința Seillac I din 1976 a condus la o definire mai precisă a noțiunii de portabilitate, distingînd între portabilitatea programelor de aplicație, independentă de dispozitiv, portabilitatea informației grafice și portabilitatea instruirii. Activitatea de standardizare a fost demarată din dorința de a orienta realizarea de instrumente grafice către programator, pentru a conferi astfel flexibilitate în dezvoltarea de aplicații. Această activitate s-a extins de la definirea interfeței cu programul de aplicație, prin modelele GKS, GKS-3D, PHIGS la transferul datelor grafice – modelul CGM și interfața cu dispozitivul – modelul CGI.

Efortul de standardizare a fost susținut în această perioadă de două surse de finanțare importante, care au reflectat, pe de o parte, interesul proiectanților de echipamente grafice și software de a satisface cererea pieței (rezultatul a fost consolidarea unor standarde "de facto", cum sînt Windows, IGES) și pe de altă parte, interesul institutelor de învățămînt superior și de cercetare în fundamentarea teoretică (avînd ca rezultat standarde ISO cum sînt GKS, PHIGS, CGM, CGI, CGRM). Prezentarea succintă a acestor standarde și a problematicei activității de cercetare și dezvoltare relaționate face obiectul următoarelor trei secțiuni din această lucrare.

Modelul de referință în grafica pe calculator CGRM, adoptat în 1992, reflectă tendința actuală de formalizare a conceptelor în grafica pe calculator și se constituie într-o bază metodologică pentru activitatea de standardizare viitoare. CGRM realizează un cadru de lucru larg, care își propune să cuprindă întreaga arie a graficii pe calculator - interfața cu programe de aplicație, transferul informației grafice, interfața cu dispozitivul și interfața cu operatorul. Din aceste motive, articolul rezervă un spațiu mai mare acestei problematice prezentînd, în cadrul unei secțiuni separate, atât modelul și conceptele de baza cît și o reconsiderare a locului și rolului standardelor existente în acest cadru.

2. Modele de interfață cu aplicații grafice

Separarea unui nucleu de grafică de partea specifică aplicației a fost una din problemele importante legate de definirea standardelor. Un rezultat important al activității grupului de lucru Seillac I a fost distincția făcută între sistemele de grafică și sistemele de modelare. Sistemele de modelare permit definirea de obiecte în sisteme proprii de coordonate și depind de structuri de date și output specific aplicației. Sistemul de grafică este răspunzător de transformarea și afișarea acestui model pe suprafața de afișare. Cu toate acestea, o separare netă nu poate fi făcută, problematica modelării avînd o influență importantă asupra proiectării nucleului de grafică. Soluția adoptată în definirea modelului GKS [4] a constituit-o amînarea problemei pînă la definitivarea sistemului de grafică, în final considerînd că aparține de grafică ceea ce este în interiorul nucleului și de modelare ceea ce rămîne în afară.

Modelele de grafică pe calculator GKS [8, 11] și PHIGS [12] definesc, într-o manieră independentă de limbaj, un set de funcții pentru executarea de operații grafice. Cu toate acestea, implementarea se realizează într-un limbaj de programare, interfețele cu limbajele FORTRAN, Pascal, C și Ada pentru standardele GKS și PHIGS fiind standardizate separat [10, 13].

GKS a fost proiectat pentru a furniza un set de funcții de programare grafică utilizabile de majoritatea aplicațiilor care produc imagini generate pe calculator. Deși setul este definit într-o manieră independentă de dispozitiv, nu cuprinde o interfață cu acesta, alegerea ei rămînînd în sarcina implementatorului. Obiectivele standardului sînt în strînsă relație cu funcțiile pe care trebuie să le îndeplinească:

- generarea și reprezentarea de imagini;
- transferul de părți de imagini create în diferite sisteme de coordonate utilizator către diferite stații de lucru și transformarea acestora în coordonate de dispozitiv;
- controlul stațiilor de lucru atașate sistemului;
- gestiunea inputului grafic;
- furnizarea de mijloace de structurare a imaginilor în părți care pot fi manipulate separat;
- stocare pe termen lung a informației grafice.

Informațiile grafice se pot structura în segmente și păstra într-un format independent de dispozitiv într-o colecție de segmente WISS (work-station

independent segment storage). Segmentele constituie entitatea manipulabilă, suportînd operații de transformare, ștergere, copiere și asociere la diferite stații de lucru. Stațiile de lucru pot păstra segmentele asociate într-o colecție dependentă de dispozitiv (WDSS).

GKS nu suportă facilități de editare de nivel malt, manipularea efectuîndu-se la nivel de segment fără a putea modifica conținutul acestuia. Segmentarea oferă o structurare liniară, pe un singur nivel, construirea unei ierarhii de obiecte pentru reprezentarea lumii reale rămînînd în sarcina programului de aplicație. O singură funcție de modelare, cea de inserare a conținutului unui segment, este suportată cu scopul de a furniza unele mijloace de editare și de a putea extinde un segment după ce a fost închis.

Primitivele grafice de ieșire, definite în coordonate utilizator (WC), sînt transformate înainte de intrarea în colecția de segmente în coordonate normalizate de dispozitiv (NDC) prin transformarea curentă de normalizare. Spațiul normalizat permite compunerea unor imagini, independente de dispozitiv, din părți definite în diferite sisteme de coordonate utilizator.

Standardul distinge între atribute definite individual, care sînt atașate la crearea primitivei (în mod static) rămînînd nemodificate pe toată durata de viață a primitivei, și atribute definite grupat (legat) numite și reprezentări, care sînt atașate la momentul afișării (în mod dinamic). Atributele individuale sînt definite în mod implicit, printr-un mecanism de setare modală care presupune atașarea valorii curente a atributelor din lista de stare a nucleului. Acest mod de definire prezintă dezavantajul unei redundanțe în memorare la începutul fiecărui segment.

Inputul grafic prevede șase clase de dispozitive virtuale de intrare (locator, stroke, valuator, choice, pick și string) care permit poziționarea și interceptia grafică, introducerea de valori reale, alternative și șiruri de caractere. Dispozitivele pot fi acționate în trei moduri de intrare – la cerere (request), prin eșantionare (sample) și prin evenimente (event).

Modelul GKS a introdus un concept nou în grafica pe calculator – stația grafică de lucru definită ca o stație de lucru virtuală, menită să abstractizeze capacități ale diferitelor echipamente grafice. Conceptul de multiple stații de lucru, deschise simultan, corespunde, pe de o parte, cerinței de a adresa simultan o stație de lucru interactivă și un plotter sau o imprimantă grafică, iar, pe de altă parte, ideii de a trata metaafișierele și colecția de segmente ca stații de lucru, în scopul facilitării

transferului de date. Dezavantajul acestui concept constă în faptul că mai multe drivere sînt simultan rezidente în memorie.

O extensie a acestui concept care permite deschiderea de multiple stații de lucru, simultan pe același ecran, a fost implementată în sistemul GKS /PC [22]. Avantajele constau în furnizarea de multiple ferestre de lucru, fapt care permite o editare mai convenabilă a entităților grafice structurate. Abordarea are însă dezavantajul unei alocări suplimentare de memorie, pentru listele de stare ale fiecărei stații de lucru, și al unui timp de răspuns mai slab, datorită operațiilor de inițializare și de actualizare a suprafeței de afișare la crearea sau ștergerea unei ferestre de lucru.

O abordare complementară este prezentată în [6], managerul de ferestre fiind amplasat sub nucleul GKS, cît mai aproape de hardware, astfel încît timpul de acces să fie cît mai mic. Un dezavantaj îl constituie faptul că aplicația nu are un control direct prin intermediul funcțiilor GKS asupra poziției unde va fi afișată informația grafică. Un dialog local între utilizator (operator) și acest manager este realizat în acest scop prin intermediul unor funcții nestandard.

Extensia GKS-3D [11] a avut ca scop rezolvarea problemei reprezentării informației grafice tridimensionale. În GKS-3D, primitivele sînt specificate în coordonate utilizator (WC3) și sînt transformate în coordonate normalizate de dispozitiv (NDC3) prin transformarea de normalizare care permite compunerea unei imagini din părți separate, definite fiecare într-un sistem de coordonate convenabil. O transformare de segment este posibilă înainte de transformarea de vizualizare. Transformarea de proiecție transformă primitivele în coordonate normalizate de proiecție (NPC) care sînt apoi transformate în coordonate de dispozitiv prin transformarea stației de lucru.

O serie de cerințe identificate de utilizatorii de sisteme grafice și nuclee de dezvoltare a unor asemenea sisteme nu sînt satisfăcute de standardul GKS. Cauzele principale sînt tehnica de segmentare, metoda de legare a atributelor și imposibilitatea de modificare a corpului segmentului grafic, fapt care a condus la dezvoltarea standardului de grafica PHIGS [12].

Cerințele avute în vedere au fost :

- grad ridicat de interactivitate;
- structurare ierarhizată multinivel a datelor grafice;
- modificarea ușoară a structurilor (echivalentul segmentelor grafice GKS) și a relațiilor între acestea;

- articulare geometrică cu posibilitatea reprezentării mișcării legăturilor mecanice;
- informații grafice 2D și 3D.

PHIGS a preluat din GKS, cu unele modificări, conceptul de stație grafică de lucru, definiția primitivelor de ieșire și a inputului grafic. Standardul prevede un mecanism de moștenire a atributelor individuale de la structurile de nivel inferior, înlocuind astfel mecanismul mai rigid al definirii statice. Setarea modală a atributelor este dinamică la momentul traversării structurii pentru afișare.

Structura cuprinde primitive, atribute, transformări, invocări de structuri și date specifice aplicației, care pot fi editate interactiv. Prin facilitatea de apelare a unei structuri pot fi construite ierarhii pe mai multe niveluri, evitînd astfel multiplicarea definiției unei entități.

În PHIGS primitivele sînt specificate în coordonate locale de modelare (MC3) și transformate la traversarea structurii de transformarea compozită de modelare în coordonate utilizator (WC3). În continuare, fluxul de vizualizare este similar cu cel din GKS-3D.

O comparație la nivelul fluxului de afișare (pipeline) între standardele GKS-3D și PHIGS, compatibile în privința dimensionalității, este prezentată în figura 1. Se relevă o serie de diferențe care derivă din incompatibilitatea între cele două modele în ceea ce privește sistemele de coordonate, transformările și structurarea datelor. O analiză detaliată a celor două modele din acest punct de vedere este făcută în [19].

Obiecțiile majore la adresa specificației PHIGS se referă la caracterul de monolit al standardului care nu permite desprinderea de părți pentru a servi nevoile specifice unei aplicații. Standardul nu prevede un subset pentru grafică exclusiv 2D sau pentru aplicații care utilizează propria lor structură de date. O explicație a acestei situații se poate deduce din unul din obiectivele standardului care prevede definirea, afișarea și modificarea obiectelor 3D relaționate geometric, obiectiv care depășește aria unui suport grafic pentru dezvoltare și cuprinde cerințe funcționale specifice unui sistem de modelare.

O altă critică adusă standardului PHIGS constă în faptul că se oferă funcții de nivel înalt la un nivel foarte coborît, prin operații de definire a structurilor la nivel de pointer, lăsînd verificarea consistenței datelor și realizarea funcțiilor de gestiune a bazei de date mai mult în sarcina programului de aplicație decît în sarcina sistemului grafic.

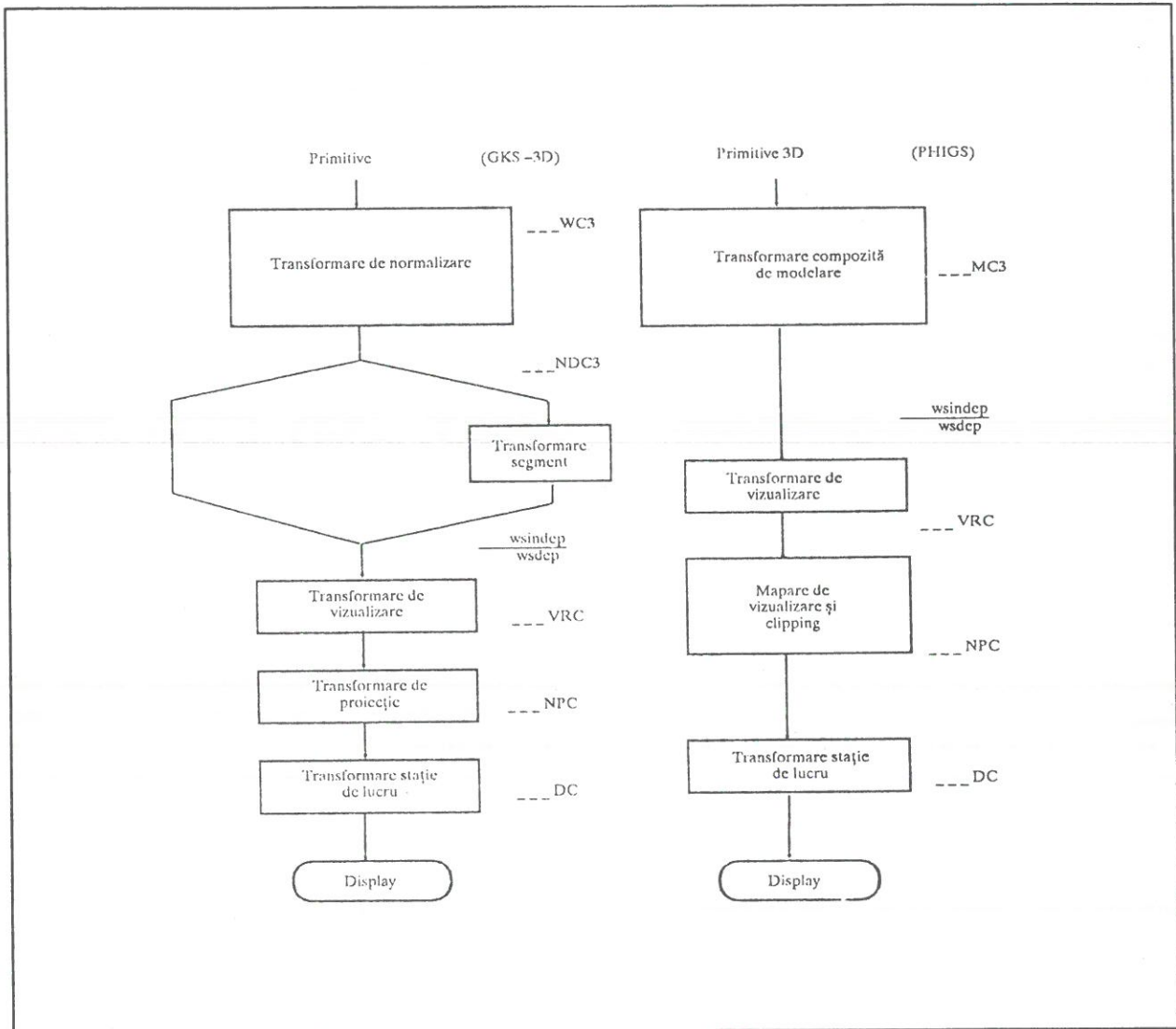


Figura 1: Fluxul outputului grafic în GKS-3D și PHIGS

L3	nivel aplicație	L2 + date orientate spre aplicație (obiecte CAD)
L2	mecanism definiție	L1 + definiție și referințe subimagini
L1	imagine structurată	L0 + segmentare și denumire subimagini
L0	imagine	primitive grafice și atribute

Figura 2: Niveluri de metafișier

O abordare care încearcă să îmbine avantajele celor două standarde GKS-3D și PHIGS o constituie specificația PHI-GKS, propusă la Universitatea din Darmstadt [3], de extindere a conceptului GKS-3D la mai multe niveluri de segmentare ierarhică și la editare, pentru a asigura o compatibilitate cu PHIGS. Modelul de atașare a atributelor este combinat, soluția de introducere a unei valori denumită *tbi* (to be inherited) pentru atributele setate modal, permițând o comutare între mecanismele PHIGS și GKS.

Cerințele de construire și întreținere a unor ierarhii sofisticate de obiecte pot fi satisfăcute de conceptele de clase multinivel, crearea de instanțe și definirea de metode, specifice programării orientate pe obiecte. Abordări privind potențialul de implementare a modelului PHIGS sînt prezentate în [20, 21]. O arhitectură GKS, bazată pe mesaje output și input care utilizează modelul computațional de lucru cu actori al lui C. Hewitt, este prezentată în [18] și constă din definirea unui număr de procese concurente (generare output, atașare atribute, transformări, preluare input etc.) care comunică prin transmitere de mesaje. Sistemul poate fi considerat ca o arhitectură condusă de date (data driven) întrucît conținutul mesajelor determină acțiunea ce va fi executată.

3. Modele de interfață cu metafișiere grafice

Existența mai multor surse de informații grafice, care sînt produse ca rezultat al procesării imaginilor, calculelor de simulare sau înregistrărilor experimentale, face necesară cuprinderea lor într-o formă unitară.

Metafișierele grafice sînt fișiere care furnizează un mecanism de stocare a informațiilor grafice în mod independent de dispozitiv și de programul de aplicație. Obiectivele urmărite prin introducerea metafișierelor în conexiune cu sistemele de grafică pe calculator sînt:

- prezentarea informațiilor grafice pe diferite suprafețe de afișare, utilizatorul avînd posibilitatea de a alege display-ul dorit;
- reținerea datelor pentru o utilizare ulterioară într-un format independent de dispozitiv astfel încît alegerea dispozitivului să poată fi făcută după generarea imaginii;
- transportarea informațiilor grafice între diferite platforme de calcul;
- furnizarea unei modalități de editare (modificare, ștergere, adăugare) a informației grafice care a fost produsă și stocată anterior.

Deși modelul GKS oferă o interfață standard de gestiune metafișier, structura și conținutul metafișierului GKSM sînt descrise într-o anexă care nu face parte din standard. Această separare a fost făcută pentru a se putea standardiza interfața cu metafișierul grafic în mod independent de interfața cu programul de aplicație.

În GKSM sînt reținute majoritatea funcțiilor legate de producerea output-ului grafic, interfața avînd practic rolul unui protocol de înregistrare (audit trail). Această facilitate de a înregistra funcții de control conferă avantaje în restaurarea unei sesiuni de lucru, realizarea de programe de demonstrație sau generarea unor secvențe pentru animație.

Intrucît GKS oferă facilități slabe de modelare a entităților grafice, singura posibilitate de editare la nivel de primitivă o constituie funcțiile de gestiune metafișier. Trebuie menționat însă că restaurarea unor obiecte complexe (cuprinzînd sute de primitive) este lentă, deoarece informația nu intră direct în fluxul de afișare, fiind mai întîi transferată programului de aplicație și returnată (după o eventuală editare) nucleului pentru interpretare.

Standardul ISO de metafișier CGM [9], care implementează conceptul VDM (Virtual Device Metafile), conține articole care descriu o imagine statică, fără a include funcții care au ca efect modificarea imaginii. Standardul oferă trei modalități de codificare a informației (caracter, binar și text). Modificările ulterioare aduse standardului au vizat includerea funcțiilor de segmentare ca facilități de structurare a imaginii și de mărire a compatibilității cu GKSM.

În diferite arii de aplicație a graficii pe calculator, a fost dezvoltat un număr mare de formate de metafișier pornind de la specificații generale pentru reținerea de primitive și atribute la formate sofisticate, specifice unor programe de aplicație.

Un tip distinct de metafișier îl reprezintă cel pentru stocarea definiției informației unui produs, utilizat în domeniile CAD/CAM. Un standard ANSI cu o largă utilizare, care furnizează o schemă complexă, orientată spre aplicație, pentru a descrie obiecte și caracteristici ale acestora din sfera proiectării asistate de calculator, este IGES.

O strategie de a dezvolta metafișiere de nivel înalt, orientate spre aplicație, fără a pierde generalitatea metafișierelor grafice, este prezentată în [4] și se bazează pe o grupare pe niveluri după schema prezentată în figura 2. Un interpretor poate prelua informația din metafișiere de nivel înalt și produce metafișiere de nivel inferior, al cărui conținut poate fi afișat pe orice echipament disponibil.

4. Modele de interfață cu dispozitive grafice

Independența de dispozitiv și de procesor, care definește un nucleu portabil de grafică pe calculator, nu cere ca o implementare să nu țină seama de echipamentele disponibile și de procesoarele pe care va fi executată. Dimpotrivă, acestea trebuie să fie adaptate la mediul de exploatare utilizând din plin caracteristicile și facilitățile oferite de dispozitiv și de calculatorul gazdă. Totodată, efortul de includere a unui nou dispozitiv grafic în configurație trebuie să fie minimizat. În [4] sînt prezentate trei strategii posibile pentru realizarea unui nucleu grafic ca sistem specific mediului de exploatare, sistem independent de dispozitiv sau sistem configurabil.

Sistemele specifice unui mediu de exploatare sînt o consecință a facilităților foarte puternice oferite de stații grafice care merg pînă la implementare pe un cip a funcționalității unui standard de grafică. În aceste cazuri, arhitectura sistemului are o influență considerabilă asupra implementării, funcțiile de interfață cu aplicația fiind realizate în mare parte direct de către funcții de dispozitiv. Dezavantajele acestei abordări provin din faptul că majoritatea configurațiilor necesită mai mult decît o stație de lucru, ceea ce conduce la o multiplicare a funcțiilor emulate software.

Cea de a doua abordare, argumentată și în [5], necesită construirea unui driver general de dispozitiv pentru implementarea funcționalității diferitelor stații de lucru. Software-ul care implementează conceptul de stație grafică de lucru cuprinde o parte independentă de dispozitiv, dar dependentă de stație (DI) și o altă dependentă de dispozitiv (DD). Efortul de introducere în configurație a unui echipament nou este astfel redus prin utilizarea unei interfețe DI/DD, soluția conducînd însă la încărcarea cu un "overhead" considerabil pentru asigurarea flexibilității. Abordări în acest sens sînt prezentate în [5, 7, 17].

Realizarea unui set de funcții de bază pentru controlul și transferul de date între nivelurile DI/DD, în fluxul de vizualizare și în gestiunea input-ului grafic, a constituit obiectivul unor eforturi de standardizare, concretizate în standardul CGI (Computer Graphics Interface) [14]. Acest standard implementează conceptul de interfață virtuală cu dispozitivul VDI (Virtual Device Interface).

CGI oferă o interfață standard pentru implementatori, producători și furnizori de echipamente grafice, furnizează un mecanism de interogare și răspuns pentru diferite facilități hardware și un mecanism standard, de tip Escape, pentru funcții

nestandard. Încurajînd portabilitatea, protejează investițiile software care nu devin perimate la apariția noilor echipamente, reduce costurile de întreținere și permite concentrarea activității de dezvoltare asupra funcțiilor de nivel înalt și a aplicațiilor.

Standardul cuprinde șase părți care descriu structura funcțională de ansamblu, funcții de control (dispozitiv, suprafața de afișare, tratare erori), output grafic (primitive și atribute), input grafic, segmentare și funcții pentru grafică raster.

Implementarea poate fi realizată printr-o corespondență software – software sau software – hardware în funcție de caracteristicile platformei de calcul. În prezent nu există un limbaj de legătură standardizat pentru funcțiile CGI.

Tendențele manifestate pe parcursul ultimilor ani evidențiază două direcții majore în activitatea de cercetare-dezvoltare în domeniul problematicii VDI :

- optimizarea părții independente de dispozitiv și minimizarea celei dependente pentru a utiliza în mod maximal funcționalitatea modelului;
- minimizarea părții independente de dispozitiv și optimizarea celei dependente de dispozitiv în vederea utilizării maxime a facilităților oferite de echipament.

Două căi pentru asigurarea unui acces mai bun la facilitățile hardware fără a pierde avantajele unei interfețe standard de programare, pot fi scurtcircuitarea standardului prin funcții Escape [5, 6] și rezolvarea unor probleme de optimizare și maximizare a funcționalității, prin combinații CGI-GKS [3].

O alternativă care încearcă să îmbine specificitatea hardware cu independența de dispozitiv, o constituie sistemele configurabile, definite în termeni de model de referință prin care se separă nivelurile de abstractizare a datelor ca aplicație, nucleu, stație de lucru, dispozitiv, fiecare constituind o sursă virtuală pentru nivelurilor superioare.

5. Modelul de referință CGRM

O preocupare constantă în activitatea de cercetare și dezvoltare a constituit-o specificarea unui model de referință pentru grafica pe calculator. O serie de abordări în acest sens sînt prezentate succint în [1, 18] și constau în descrierea graficii pe calculator în termeni de fluxuri de output grafic (output pipeline) și input grafic (input pipeline). Eforturile făcute pentru definirea unor asemenea

modele au fost determinate de o serie de factori cum sînt :

- dezvoltarea și utilizarea în paralel a mai multor standarde pentru același tip de interfață;
- preocuparea de a asigura o mai mare compatibilitate între standardele de interfață cu programul de aplicație și conceptele VDI și VDM;
- apariția și utilizarea pe scară largă a unor medii de dezvoltare, cum este Windows, și a sistemelor bazate pe cunoștințe, care au influențat modul de abordare a interfeței cu utilizatorul;
- impactul progresului tehnologic, care reclamă sisteme configurabile, cu flexibilitate mărită.

Adoptarea de către ISO în 1992 a standardului CGRM (Computer Graphics Reference Model) [15] constituie un pas important în rafinarea unui cadru metodologic în grafica pe calculator, definind conceptele care trebuie utilizate pentru a dezvolta și compara sisteme și standarde de grafică pe calculator.

Modelul se bazează pe conceptul de mediu, care reprezintă o subdiviziune a CGRM la un anumit nivel de abstractizare, definind grafica pe calculator în termenii a cinci medii abstracte, situate între aplicație (nivelul cel mai înalt) și operator (nivelul cel mai coborît):

- mediul construcție (interfață cu programul de aplicație);
- mediul virtual;
- mediul vizualizare;
- mediul logic;
- mediul realizare (interfața cu operatorul).

Structura generală a CGRM, prezentată în figura 3, evidențiază interfețele externe prin care modelul comunică cu aplicația, operatorul, metafișierul de înregistrare comenzi (ATMF - audit trail metafile) și metafișierul de captare date (DCMF - data capture metafile). În CGRM aceste obiecte sînt externe și nu fac parte din grafica pe calculator.

Înțelegerea modelului și analiza relației dintre acesta și standardele de grafică pe calculator necesită o prezentare a conceptelor de bază, așa cum sînt definite de CGRM.

Articolul de informație, stocat în interiorul unui mediu sau transmis între două medii, se numește

entitate. Entitățile se împart în trei clase: output, input și control.

Primitiva de ieșire constituie unitatea atomică pentru output grafic. Proprietățile geometrice sau de altă natură pot fi legate de primitive la momentul creerii sau ulterior, utilizînd mecanisme ca setarea modală sau moștenirea. Unele proprietăți pot fi consumate în procesul de elaborare, pierzîndu-și identitatea ca proprietăți.

Mesajul de intrare constituie unitatea atomică pentru descrierea inputului grafic. Mesajele pot fi transformate în mai multe noi mesaje prin emanație. Proprietățile pot fi legate la crearea sau ulterior. Anumite proprietăți pot fi consumate.

Un set de entități de ieșire care sînt denumite și pot fi structurate, se numește colecție. O colecție poate fi manipulată pentru a produce o compoziție (set spațial, structurat de primitive de ieșire) în același mediu. Compoziția are nume specifice în fiecare mediu: model, scenă, imagine, imagine grafică, display.

Un set de entități de intrare într-un mediu se numește agregare. Agregarea poate fi manipulată pentru a produce entități în stocul de mesaje de intrare în același mediu.

Fiecare mediu constă din elemente de informație (stocări de informație într-un mediu dat) și elemente de procesare (processe într-un mediu dat). Procesele din fiecare mediu au nume specifice. Simetria dintre input și output în fiecare mediu reflectă o simetrie a obiectivelor, și nu a complexității.

Absorbția este procesul care primește entități de la mediul imediat superior și le procesează pentru utilizare în propriul său mediu. Denumiri specifice mediilor: pregătire, producție, proiecție, completare, prezentare. Emanația este procesul simetric care emană entități către nivelul imediat superior după procesarea lor. Nume specifice: acumulare, abstractizare, elevație, generare, utilizare.

Distribuția este procesul care distribuie compoziția și entitățile de control output către mediul imediat inferior. Asamblarea este procesul care primește entități de la nivelul imediat inferior pentru utilizare în propriul său mediu.

Manipularea este procesul care accesează și schimbă conținutul elementelor de informație, furnizînd astfel legături între input și output în cadrul aceluiași mediu.

Sistemele de coordonate utilizate în interiorul unui mediu sînt aceleași. În consecință, toate transformările apar în procesele de absorbție și emanație. Dacă este prevăzut clipping, rezultatul trebuie specificat în termeni de primitive de ieșire din același mediu.

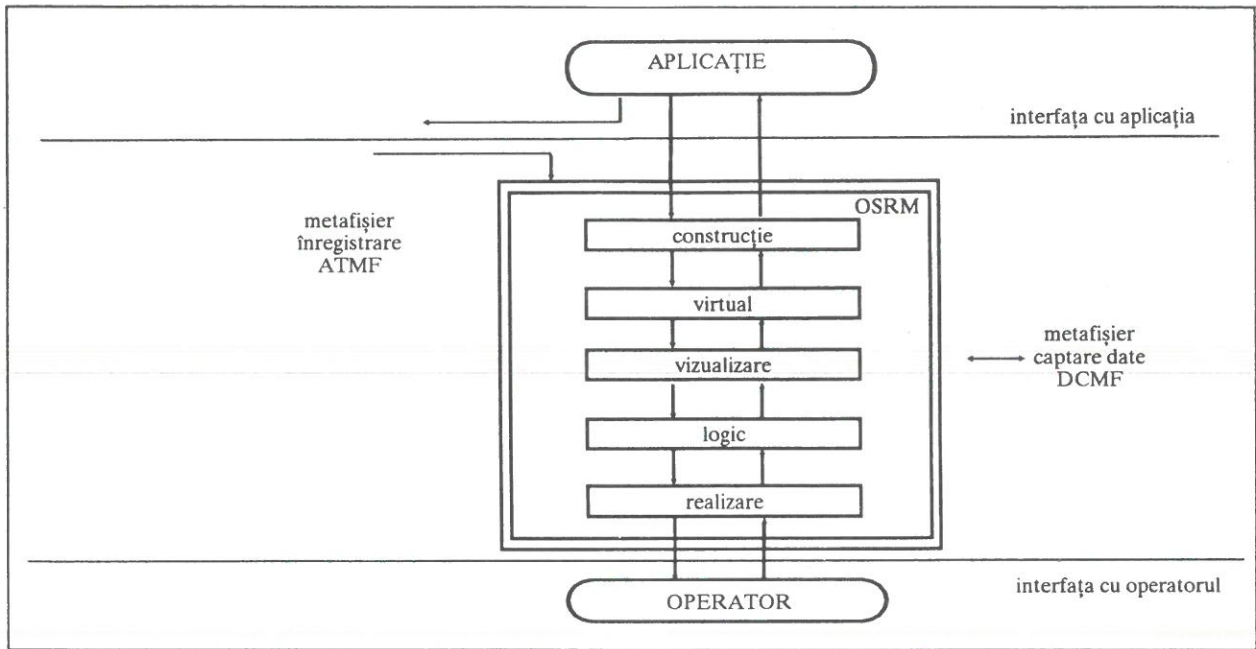


Figura 3: Structura generală a CGRM

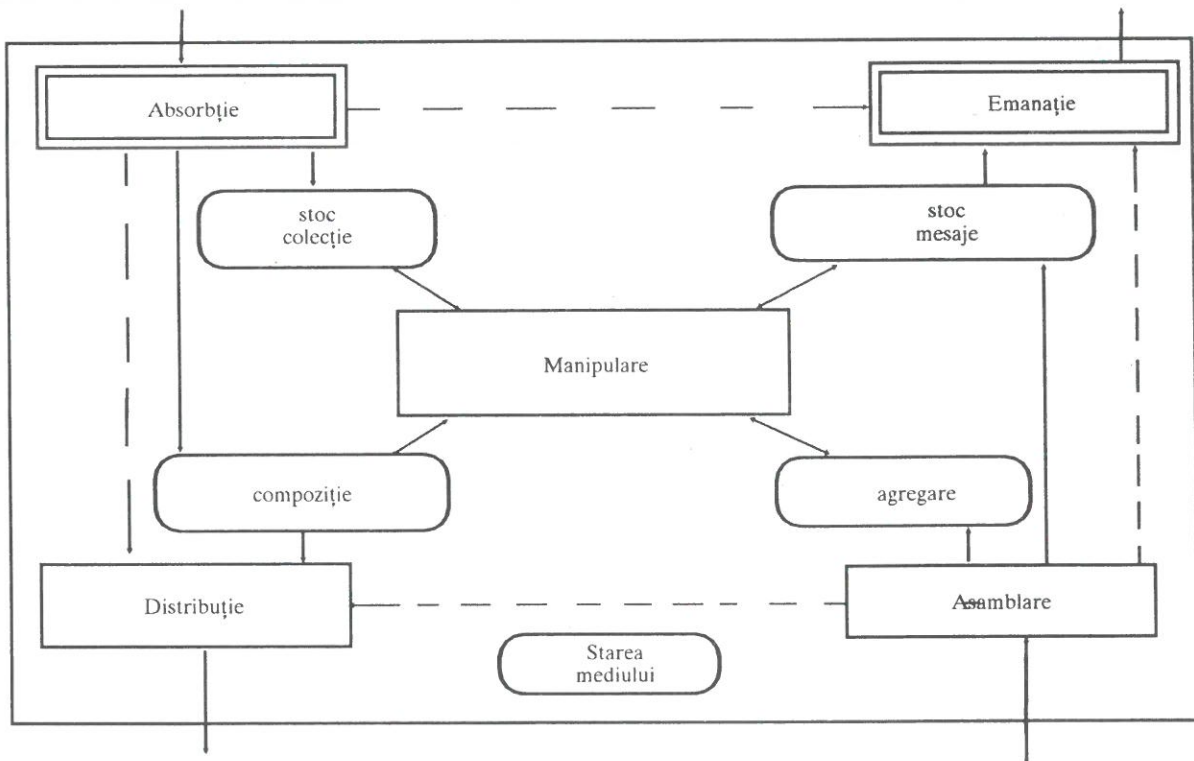


Figura 4: Modelul unui mediu CGRM

Starea mediului se definește prin entități separate de alte informații din cadrul unui mediu, care pot fi propagate spre medii adiacente prin operații de control. Informațiile de stare pot fi partajate de toate procesele din mediu și de operațiile care se fac asupra entităților din acest mediu. Pot fi utilizate pentru a verifica dacă o operație particulară poate fi executată.

În figura 4 este prezentată structura generală a unui mediu care evidențiază procese, elemente de informație și legături între acestea. Fiecare mediu este descris conceptual, în termenii de sistem grafic, care se interfațează atât cu aplicația, cât și cu operatorul. Este posibil pentru un mediu să fie nul. În continuare, se prezintă câteva din caracteristicile mediilor specifice.

În mediul construcție, informația aplicației care urmează să fie afișată este pregătită sub forma unui model din care să poată fi produse scene grafice specifice. Aplicația poate să editeze modelul (compoziția) și colecția din mediul construcție. Denumirea de "părți de colecții" este permisă și poate să ajute interacțiunea. Mesajele din stocul de mesaje input sînt construite la precizia necesară aplicației.

În mediul virtual este produsă o scenă din model. Scena constă dintr-un set de primitive virtuale, pregătite pentru vizualizare. Geometria este complet definită astfel încît scenele sînt complete din punct de vedere geometric. Mesajele input pot fi diferențiate după proprietățile asociate.

În mediul vizualizare, o imagine a scenei este generată prin proiecție.

În mediul logic, este completată o imagine grafică în vederea realizării. Întregul set de proprietăți este legat la primitive.

Mesajele input și informațiile de stare sînt convertite la forme independente de dispozitiv cu proprietăți adăugate pentru a diferenția originea inputului. O interpretare concretă a stocului de mesaje s-ar putea să nu fie cunoscută în acest mediu.

În mediul realizare, este prezentată o afișare a imaginii grafice (mediul nu corespunde neapărat unui display fizic). Mesajele input din stocul de mesaje sînt primite de la dispozitiv. Operatorul poate edita conținutul stocului de mesaje și al agregării în acest mediu. La acest nivel, sînt cunoscute toate proprietățile asociate dispozitivului fizic.

Standardele existente nu se încadrează cu precizie în CGRM deși experiența acumulată a influențat semnificativ dezvoltarea acestui model. Din acest motiv, o comparație succintă cu modelele GKS, GKS-3D, PHIGS, CGM și CGI per-

mite o delimitare mai clară a conceptelor privite prin prisma modelului de referință.

În GKS producția de primitive din spațiul de coordonate normalizate (NDC) corespunde cu nivelul virtual. Stația grafică de lucru GKS corespunde nivelului logic și de realizare. Atașarea atributelor legate și transformarea stației se fac în mediul logic. Atributele individuale se atașează în NDC, modelul individual/grupat fiind compatibil cu modelul de referință CGRM în măsura în care întreaga geometrie este specificată în NDC. Stocul de segmente este identificat cu o colecție virtuală WISS și o colecție vizualizată WDSS. Coada de evenimente corespunde cu stocul de mesaje de intrare din mediul virtual.

Arhiva centrală de structuri din PHIGS corespunde unei colecții în mediul construcție. Stația grafică de lucru PHIGS corespunde cu mediile virtual, vizualizare, logic și realizare. Setul structurilor trimise spre afișare formează stocul conceptual al colecției în mediul vizualizare. Outputul grafic este generat prin traversarea structurilor care constituie un proces de manipulare în acest mediu.

Standardul CGM este un standard de captare date 2D (compoziție) în mediul vizualizare. Termenul de metaafișier este utilizat diferit de CGRM care presupune captarea unei imagini și de CGM care presupune multiple imagini.

Standardul CGI poate fi considerat ca o interfață pentru mediile vizualizare, logic și realizare. Interfața CGI corespunde cu interfața dintre nivelul virtual și de vizualizare în modelul de referință CGRM. Primitivele de ieșire pot fi colectate în mediul de vizualizare în colecția de segmente. Atributele sînt atașate în același mediu. Nu există o colecție în mediul logic. CGI prevede funcții raster în mediul de realizare, care corespund cu operațiile de manipulare pe colecția (bitmap store) din acest mediu.

Un model care a avut o influență considerabilă în ultimii ani îl reprezintă mediul de dezvoltare Windows. Sistemele de grafică pe calculator și sistemele Windows se referă la crearea, manipularea și interacțiunea cu obiecte afișate pe suprafețe de ecran partajate. Funcțiile definatorii sînt totuși diferite, sistemele de grafică referindu-se în primul rînd la output și input grafic, în timp ce sistemele Windows se referă în primul rînd la gestiunea resurselor partajate de multiple aplicații.

6. Concluzii

Activitatea de standardizare în grafica pe calculator a avut ca efect crearea unui cadru metodologic pentru dezvoltarea conceptelor și precizarea ter-

minologiei din acest domeniu, ca bază comună de înțelegere în crearea și utilizarea de sisteme grafice, în instruire și în transferul de informații grafice.

Evoluția și tendințele actuale în activitatea de cercetare și dezvoltare relaționată definirii standardelor oficiale în grafica pe calculator justifică evidențierea următoarelor concluzii:

1. creșterea performanțelor este relaționată de mărirea flexibilității pentru integrarea progresului tehnologic fără modificări radicale ale software-ului;
2. o modalitate de a construi sisteme configurabile de grafică pe calculator poate fi indusă de orientarea pe obiecte, care permite implementarea unei ierării de clase de obiecte cu facilități de moștenire a metodelor;
3. cerințe noi în abordarea interfeței utilizator, ca urmare a dezvoltării sistemelor inteligente și a facilităților oferite de medii de dezvoltare de tip Windows, reclamă extinderea funcționalității modelelor la nivelul interfeței cu aplicația și cu operatorul;
4. modelul de referință pentru grafica pe calculator capătă o importanță deosebită pentru implementarea conceptului de "arhitectură deschisă", atât în dezvoltarea de sisteme grafice portabile, cât și în contextul relaționării graficii pe calculator cu alte domenii ale cercetării și dezvoltării în informatică.

BIBLIOGRAFIE

1. Encarnação, J.L.: *Incorporating Knowledge Engineering and Computer Graphics for Efficient and User-friendly Interactive Graphics Applications*. În: C.E. Vandoni (Ed.), Eurographics '85, Elsevier Science Publisher B.V., Amsterdam, 1985.
2. Encarnação, J.L., Schonhut, J.: *Interface and Data Formats for Transfer and Communications in Computer Graphics Systems*. În: D.F. Rogers și R.A. Earnshaw (Eds.), *Techniques for Computer Graphics*, Springer-Verlag, Berlin, 1987, pp. 237-254.
3. Encarnação, J.L.: *R&D Issues and Trends Consequent Upon GKS and Related Standards*. În: D.F. Rogers și R.A. Earnshaw (Eds.), *Techniques for Computer Graphics*, Springer-Verlag, Berlin, 1987, pp.443-454.
4. Enderle, G., Kansy, K., Plaff, G.: *Computer Graphics Programming. GKS - The Graphics Standard*, Springer-Verlag, Berlin, 1987.
5. Herman, I., Revitzky J.: *A General Device-Driver for GKS*. În: P.R. Bono și I. Herman (Eds.), *GKS Theory and Practice*, Springer-Verlag, 1987, pp. 37-46.
6. Herman, I., Tolnay-Knefely, T., Vincze, A.: *XGKS - A Multitask Implementation of GKS Machine*. În: P.R. Bono și I. Herman (Eds.), *GKS Theory and Practice*, Springer-Verlag, 1987, pp. 47-57.
7. Herman, I., Revitzky, J., Tolnay-Knefely, T.: *A Concept for a GKS Machine*. În: C.E. Vandoni (Ed.), *Eurographics '85*, Elsevier Science Publisher, B.V., Amsterdam, 1985, pp.203-212.
8. ISO 7942:1985. *Information Processing Systems - Computer Graphics - Graphical Kernel System (GKS) Functional Description*.
9. ISO 8632-1,2,3,4: 1992. *Information Technology - Computer Graphics - Metafile for Storage and Transfer of Picture Description Information*.
10. ISO 8651-1,2,3:1988 și ISO/IEC 8651-4:1991. *Information Processing Systems - Computer Graphics - Graphical Kernel System (GKS) Language Bindings*.
11. ISO 8805:1988. *Information Processing Systems - Computer Graphics - Graphical Kernel System for Three Dimensions (GKS-3D) Functional Description*.
12. ISO/IEC 9592-1,2:1989. *Information Processing Systems - Computer Graphics - Programmer's Hierarchical Interactive Graphics System (PHIGS) - Functional Description*.
13. ISO/IEC 9593-1,2:1990 și ISO/IEC-4:1991. *Information Technology - Computer Graphics - Programmer's Hierarchical Interactive Graphics System (PHIGS) Language Bindings*.
14. ISO/IEC 9636 - 1,2,3,4,5,6:1991. *Information Technology - Computer Graphics - Interfacing Techniques for Dialogs with Graphical Devices (CGI)*.
15. ISO/IEC 11072:1992. *Information Technology - Computer Graphics - Computer Graphics Reference Model (CGRM)*.
16. Klimenko, S.V., Kochin V.N., Samarin, A.V.: *A Pipeline Approach for Building the Presentation Graphics Systems*. În: C.E. Vandoni

- (Ed.), Eurographics '85, Elsevier Science Publisher B.V., Amsterdam, 1985, pp. 427-438.
17. Pribeanu, C., Ionescu, M.L.: *O implementare multitasking a standardului GKS*. În: Revista de Informatică și Automatică nr.2/1991, pp. 35-36.
 18. Reynolds, G.J.: *A Token Based Graphics System*. În: P.R. Bono și I. Herman (Eds.), GKS Theory and Practice, Springer - Verlag, Berlin, 1987, pp.67-77.
 19. Singleton, K.: *An Implementation of GKS-3D /PHIGS Viewing Pipeline*. În: P.R. Bono și I. Herman (Eds.), GKS Theory and Practice, Springer-Verlag, Berlin, 1987, pp. 145-183.
 20. Wisskirchen, P.: *Towards Object Oriented Graphics Standards*. În: C.E. Vandoni (Ed.), Eurographics '85, Elsevier Science Publisher B.V., Amsterdam, 1985, pp.391-400.
 21. Wisskirchen, P.: *Graphics System with Editable Objects*. În: T.I. Kunii (Ed.), Advanced Computer Graphics, Springer-Verlag, Tokio, 1986.
 22. *Sistem grafic în timp real - Manual de prezentare suport grafic*, ICI, 1991.



PROGRAF-ADC

Programarea realizării producției complexe prin metoda analizei drumului critic

Produsele industriale complexe și obiectivele de construcții industriale și civile sînt rezultatul unor activități înlănțuite și interconținute. La rîndul lor, activitățile pot fi decupate în subactivități și operații, fiecare dintre acestea utilizînd resurse materiale și umane, precum și capacități tehnologice.

Planificarea activităților și subactivităților pentru realizarea unor produse complexe (construcții navale, aeronave, locomotive etc.), construcții civile și industriale (drumuri, poduri, centrale electrice, linii electrice etc.) sau a unor mari proiecte de dezvoltare are ca obiectiv corelarea desfășurării activităților astfel încît să se obțină o durată de realizare minimă a obiectivului, rezultată pe baza determinării drumului critic, cu respectarea interconținuturilor tehnologice și ținînd seama de nivelul resurselor disponibile ale executantului.

Pentru elaborarea graficului unui obiectiv complex, fiecare activitate trebuie caracterizată prin durata sa (rezultată din subactivitățile și operațiile componente, de execuție și logistice), printr-o relație de dependență în raport cu alte activități și prin resursele specifice necesare pentru îndeplinirea sa.

Produsul PROGRAF-ADC asigură funcțiunile privind:

- *gestionarea activităților, subactivităților și operațiilor, a atributelor acestora precum și resursele necesare. Numărul activităților pentru fiecare din cele 3 niveluri este de pînă la 100;*
- *determinarea duratei activităților de nivel superior pentru acele activități care au un graf subordonat;*
- *determinarea drumului critic în grafurile de la fiecare nivel subordonat și în graful principal, evidențierea activităților, subactivităților și operațiilor situate pe drumul critic și a intervalelor de timp disponibile pentru activitățile, subactivitățile și operațiile care nu sînt situate pe drumul critic, cu indicarea momentului celui mai tîrziu de începere;*

- *modificarea inițială a activităților și subactivităților sau actualizarea lor pe parcursul desfășurării, precum și refacerea corespunzătoare a drumului critic ținând seama de datele reale, obținute din urmărire;*
- *editarea graficului activităților și a caracteristicilor acestora sub forma unor grafice GANTT, în condițiile indicării datei de început sau impunerii unor termene de sfârșit pentru realizarea obiectivului;*
- *adaptarea graficului și respectiv a drumului critic în funcție de restricții privind capacitățile disponibile sau de termene de disponibilitate pentru resurse materiale;*
- *determinarea necesarului de resurse pe intervale de timp definite;*
- *furnizarea unor situații selective privind activități, subactivități sau operații și resurse.*

Rapoartele furnizate informează asupra coordonatelor de începere și de terminare a activităților planificate, asupra activităților începute, precum și asupra activităților terminate sau îrziate, pentru anumite intervale de timp sau față de o anumită dată calendaristică.

Produsul **PROGRAF-ADC** permite analize de detaliu a desfășurării producției prin rapoarte, sum sînt:

- *comparația grafică a necesarului și disponibilului de forță de muncă pe diferite intervale;*
- *refacerea drumului critic și a graficelor plecînd de la stadiul execuției obiectivului la un moment dat;*
- *nivelarea utilizării forței de muncă pentru activitățile cu rezervă de timp.*

Resursele producției sînt tratate diferit, forța de muncă condiționează prin nivelul sau pe tot intervalul activității, iar utilajele, sculele și materialele - prin data de la care sînt disponibile pentru întreaga durată a activității.

Produsul **PROGRAF-ADC** dispune de o interfață prietenoasă și furnizează grafice și situații afișate pe ecran sau tipărite la imprimantă.

Pentru integrarea în sistemele informatice se cuplează la o bază de date, conținînd date tehnoloice (pentru domeniul construcțiilor de mașini) sau conținînd normele de construcții (pentru domeniul construcțiilor civile și industriale).

Elaborarea repetată, pentru mai multe produse sau obiective a graficelor, permite o evaluare a posibilităților organizației industriale de a asigura resursele necesare, cumulate pentru toate obiectivele, pe diferite intervale de timp.

PROGRAF-ADC, realizat utilizînd SGMD Fox Prol/LAN, poate fi exploatat pe echipamente de tip PC/AT sub MS/DOS.

Livrarea programelor este însoțită de manualele de prezentare și utilizare.

Garanția acordată produsului PROGRAF-ADC este de 12 luni de la instalarea sa pe echipamentul utilizatorului.

Elaborator: Institutul de Cercetări în Informatică

Laboratorul 2.8 - Sisteme pentru conducerea producției.

Adresa: Bd. Mareșal Al. Averescu 8-10, București 1.

telefon: 665.60.60/208, camera 611.

