

# MODELAREA, RECUNOAȘTE- REA ȘI EXTRAGEREA TRĂSĂ- TURILOR ÎN SISTEMELE DE INGINERIE CONCURENTĂ

ec. Claudia Ionescu  
ing. Constantin Vasiliu

*Institutul de Cercetare în Informatică*

**Rezumat:** În articolul de față, se prezintă o dezvoltare relativ nouă în CAD/CAM privind modelarea, recunoașterea și extragerea de trăsături în sistemele de inginerie concurrentă. Sunt evidențiate de asemenea metodele avansate de modelare a solidelor, reprezentările trăsăturilor, tehnici de recunoaștere pentru modelele B-rep și CSG, de recunoaștere a formelor, rețele neuronale și conversia trăsăturilor.

**Cuvinte cheie:** trăsătură, modelare prin trăsături, modelare a solidelor, reprezentare trăsături, CAD/CAM, rețele neuronale.

Modelarea prin trăsături ("feature modelling") este o dezvoltare relativ nouă în CAD/CAM. Dacă în modelarea solidă sunt memorate numai informații despre geometria obiectelor, în modelarea prin trăsături, în modelul obiectului sunt memorate și informații funcționale.

Întrucât modelarea prin trăsături se grefează pe modelarea solidă, se va face o trecere în revistă a metodelor de modelare solidă avansate, punând accent pe modelarea parametrică și pe modelarea bazată pe restricții. Cele două căi mai importante de definire a trăsăturilor ("features"), aferente modelului produsului, "recunoașterea de trăsături" într-un model geometric și "proiectarea de trăsături" vor fi abordate în acest articol.

## 1. Modelarea prin trăsături

În sistemele CAD/CAM, informațiile despre forme și alte aspecte relevante ale obiectelor sunt memorate într-o bază de date. Călea tradițională de memorare a informațiilor privind forma obiectelor este arhivarea desenelor tehnice. Problema principală a desenelor tehnice (în 2D) este ca acestea să nu conducă la o reprezentare ambiguă a obiectului tridimensional.

De multe ori, desenele tehnice nu pot servi ca model general al unui obiect, model care să fie utilizabil pentru toate tipurile de programe de aplicații în sistemele CAD/CAM. Cele două scheme de reprezentare mai importante, B-rep ("boundary representation") și CSG ("constructive solid geometry") pot fi utilizate de multe programe de aplicații, dar, pentru realizarea sistemelor CAD/CAM puternic integrate sunt necesare și alte informații decât cele referitoare la forma obiectelor.

În acest context, o dezvoltare foarte importantă este modelarea trăsăturilor în care, pe lângă informații referitoare la formă, sunt memorate și alte informații despre un obiect, într-un așa numit model al produsului ("product model"). Cea mai importantă adăugire este informația funcțională despre obiect care poate reprezenta fie funcția unei anumite părți din obiect, fie informația despre modul în care o anumită parte a obiectului este prelucrată sau asamblată. O mare parte din aceste informații funcționale pot fi modelate prin trăsături de tip formă (TTF), care sunt elemente de formă cu anumite funcții sau semnificații (exemple: găuri, șanțuri).

În alte situații, termenul "trăsătură" este desori utilizat referitor la proprietățile materialului, la utilizarea și la întreținerea informațiilor. Ca exemple de aplicații care utilizează trăsături se pot menționa: (a) planificarea procesului de fabricație, (b) aplicații de analiză, (c) proiectarea obiectelor, (d) standardizarea modelului produsului și schimbul de date despre produs.

Utilizând modele de produse cu trăsături și alte tipuri de informații, integrarea completă CAD/CAM devine posibilă cel puțin la nivel teoretic. Toate programele de aplicații ale unui sistem au acces la un model central al produsului și utilizatorul nu trebuie să introducă informații noi, specifice pentru fiecare aplicație.

Într-un mediu ingineresc ideal, inginerii având specialități diferite pot să lucreze simultan la proiectarea unui produs. După ce un model conceptual inițial a fost definit, inginerii pot lucra simultan la un model al produsului, adăugând detalii de proiectare și, de asemenea, executând programe de aplicații pentru analiza costurilor, planificarea proceselor, analiza de eforturi (structuri de rezistență). Acest obiectiv este numit **inginerie simultană sau concurrentă**.

Dacă trăsăturile sunt utilizate în contextul ingineriei concurente, aceasta înseamnă că toate procesele de proiectare (conceptuale sau mai detaliate) și toate aplicațiile au la dispoziție setul lor propriu de trăsături, fapt ce presupune asigurarea conversiei automate între trăsături.

Modelarea trăsăturilor se construiește pe un număr de tehnici de modelare solidă avansate precum modelare parametrică și modelare bazată pe restricții.

## 2. Definierea și clasificarea trăsăturilor

În general, prin trăsătură se desemnează o parte distinctivă sau proeminentă a unui obiect sau sistem, concret sau abstract. În această definiție apar trei aspecte relevante:

a) o trăsătură este o parte a unei entități mai mari

b) partea respectivă are anumite proprietăți care o disting dintr-un întreg

c) trăsăturile pot să apară în toate tipurile de entități și pot să aibă un înțeles specific pentru fiecare entitate.

În contextul CAD/CAM, în ultimii ani, s-au elaborat diferite definiții pentru termenul de "trăsătură". Anumite definiții sunt legate de aplicații specifice și sunt orientate pe geometrie; astfel, în această accepțiune o "trăsătură" este:

a) o geometrie care corespunde la operații de prelucrare primare

sau

b) o parte distinctivă sau caracteristică a unei etape de prelucrare, definind o formă geometrică care este, fie specifică unui proces de prelucrare sau care poate fi utilizată în scopuri de fixare și/sau măsurare.

Alte definiții sunt independente de aplicație, dar orientate încă pe geometrie: (a) o zonă de interes pe suprafața unei piese sau (b) o zonă de interes în modelul unei piese.

Anumite definiții independente de aplicație sunt generale, în sensul în care nu sunt restricționate de aspecte geometrice:

a) un set de informații legat de o descriere a unei piese

b) elemente utilizate în generarea, analiza sau evaluarea proiectării

c) un aspect legat de forma funcțională pentru proiectare și fabricație.

În anumite tipuri de definiții, alte informații și, în particular, informația funcțională joacă un rol important, și nu informațiile legate de formă sau de părți ale obiectelor. Uneori, funcția unei piese este mai importantă decât forma ei exactă, iar memorarea informațiilor funcționale este suficientă. Trăsăturile pot, de asemenea, să descrie și alte informații despre produs, cum ar fi: proprietăți de material, parametri tehnologici și precizii legate de fabricație. Pe de altă parte însă, forma ca atare (uneori incomplet specificată) joacă un rol important în multe trăsături, care sunt denumite trăsături de tip formă (TTF).

Exemple și clasificări ale trăsăturilor:

a) O primă distincție poate fi făcută între trăsături elementare și compuse.

Trăsăturile elementare (atonic) sunt trăsăturile simple care nu mai pot fi descompuse în trăsături mai simple (de exemplu, o gaură).

Trăsăturile compuse sunt mai complexe, alcătuite din trăsături elementare (de exemplu, o gaură în trepte - compusă din două găuri concentrice).

b) Altă distincție poate fi făcută între trăsături implicite și explicite.

Trăsăturile implicite (neevaluate sau procedurale) sunt trăsături care nu sunt definite neambiguu. Acestea sunt date printr-o descriere generică și printr-un număr de parametri pentru un eveniment specific, dar neevaluate într-o descriere geometrică explicită (de exemplu, un șurub al cărui tip și mărime au fost specificate, care determină complet șurubul, dar a cărui formă exactă, în mod special pasul, nu este reprezentată nefiind relevant pentru multe scopuri).

Trăsăturile explicite (evaluate sau enumerative) sunt trăsături ale căror forme sunt descrise explicit printr-un model geometric. În modelarea ierarhică, anumite trăsături pot fi implicite la nivelurile înalte ale modelului și explicite la niveluri joase.

Modelarea trăsăturilor, în general, și conversia trăsăturilor, în particular, sunt puternic dependente de modelarea parametrică și de modelarea bazată pe restricții.

### 3. Metode avansate de modelare a solidelor

#### a) Dimensiuni și toleranțe

Reprezentările principale, utilizate în modelarea solidă, sunt bazate pe ecuații matematice și pe relații matematice între elementele geometrice elementare: fețe, colțuri, muchii (reprezentări prin frontiere) și pe o combinație de seturi ale obiectelor primitive transformate (CSG). Aceste reprezentări nu corespund direct entităților normal utilizate în practica ingineriei: în particular, se afectează dimensiuni și toleranțe. De exemplu, o gaură cilindrică este reprezentată printr-o ecuație a suprafeței, într-o reprezentare pe suprafețe și prin scăderea unei primitive cilindrice, în reprezentarea CSG. Relația dintre reprezentarea prin modelare solidă și dimensiuni nu este directă.

Dimensiunile unui obiect specifică un obiect nominal. În practică, uneori, este imposibil să se producă un obiect în acord cu dimensiunile nominale, datorită preciziei finite a echipamentului de prelucrare. Deviațiile ce pot fi admise de la dimensiunile nominale sunt specificate prin toleranță. Informațiile despre dimensiuni și toleranță sunt indispensabile în multe aplicații

CAD/CAM, în particular, în planificarea procesului pentru fabricație și asamblare.

Pentru ca modelarea solidă cu dimensiuni și toleranțe și, în general, modelarea trăsăturilor să fie operații efective, sunt necesare tehnici aparținând următoarelor zone din:

- modelarea bazată pe nonvarietăți, pentru a specifica linii de referință și de suprafețe
- modelarea parametrică, pentru a specifica dimensiuni
- modelare bazată pe restricții, pentru a specifica toleranțe.

#### b) Modelare bazată pe nonvarietăți

Cu ajutorul metodelor de modelare CSG și a frontierelor, se pot prezenta numai obiectele reprezentate prin varietăți bidimensionale. Acestea sunt definite formal în termeni matematici ca obiecte de la care fiecare punct de pe frontiera obiectului are o vecinătate care este echivalentă cu un disc 2D.

În practică, aceasta înseamnă printre altele că:

- fiecare muchie aparține la exact 2 fețe
- fiecare vârf este înconjurat de o secvență de muchii și fețe
- fețele intersectează numai muchii și vârfuri comune
- volumul se află numai pe o parte a unei fețe.

În general, pentru a specifica dimensiuni și toleranțe, pentru modelarea trăsăturilor și pentru multe alte scopuri (cum ar fi analiza elementului finit, unde un obiect solid este împărțit într-o rețea de tetraedre) este necesară o reprezentare a obiectelor bazată pe nonvarietăți. Astfel de obiecte pot avea fețe, muchii și vârfuri care nu respectă condițiile ce decurg din definiția varietăților geometrice.

Modelarea bazată pe nonvarietăți oferă următoarele posibilități și avantaje:

- domeniul formei reprezentărilor bazate pe nonvarietăți este mult mai larg decât al reprezentărilor bazate pe varietăți (exemplu: este posibil ca două muchii sau fețe să se atingă și sunt posibile muchii și fețe în suspensie);

- modelele "wire frame", prin frontiere și volumetrie pot fi integrate într-o reprezentare care este utilă în aplicații
- informația geometrică, cum ar fi liniile de referință și suprafețele, poate fi inclusă în modelul geometric de bază.

Un dezavantaj al modelării bazate pe nonvarietăți este creșterea complexității structurii de date și a rutinelor sale de acces.

#### c) Modelare parametrică

În modelarea solidelor, obiectele pot fi adesea caracterizate printr-un număr restrâns de parametri. Exemplele cele mai simple sunt primitivele din sistemele de modelare CSG. Un paralelipiped poate fi caracterizat prin lungime, lățime și înălțime, iar un cilindru prin rază și înălțime. Astfel de obiecte sunt denumite **obiecte parametrizate**.

Parametrizarea nu se restricționează la aceste primitive și se bazează pe faptul că, variind unul sau mai mulți parametri, pot fi definite obiecte diferite. Dacă într-un sistem de modelare, obiectele pot fi definite într-un astfel de mod, acest gen de modelare se numește **modelare parametrică**.

În acest sens, sunt puse la dispoziție colecții de obiecte predefinite, iar sistemul de modelare oferă un mecanism de specificare a parametrilor pentru un obiect concret. Parametrii sunt, mai ales, dimensiuni ale obiectului. Pentru a implementa modelarea parametrică, un sistem trebuie să ofere un mecanism de creare a descrierii parametrizate a obiectului și de memorare a acestuia într-o bibliotecă.

O astfel de descriere este denumită **definiție generică** a obiectului și fiecare apariție a obiectului este numită o **instanțiere** a obiectului.

Pe baza parametrilor specificați și a definiției generice, reprezentarea cerută a unei instanțieri poate deriva într-un anumit sistem de modelare solidă (exemplu: reprezentări prin frontiere sau CSG). Dacă definiția generică a unui obiect este modificată, toate aparițiile sunt modificate automat în acord cu noua definiție. O definiție generică poate fi comparată cu o declarație de procedură într-un limbaj de programare, iar instanțierea cu o procedură de apel.

Se pot distinge două tipuri de parametrizare: geometrică și topologică.

În parametrizarea geometrică, pot varia numai geometria sau poziția și dimensiunile unui obiect, dar nu structura.

În parametrizarea topologică sau structurală, poate varia structura obiectului (exemplu: numărul de găuri într-un cilindru).

Modelarea parametrică are două avantaje:

- specificația modelului obiectelor utilizate frecvent folosind un număr restrâns de parametri este mult mai simplă pentru proiectant decât într-un sistem de modelare fără obiecte parametrizate;
- asigură intensificarea implicită a standardizării, permițând proiectanților să utilizeze numai obiecte parametrizate.

Puterea modelării parametrice crește prin adăugarea de restricții.

#### d) Modelarea bazată pe restricții

Prin modelarea parametrică se creează instanțieri diferite ale unui obiect generic specificând parametri potriviți. Uneori, în toate instanțierile, trebuie să se mențină anumite relații între elementele obiectului. De exemplu, două puncte trebuie să aibă: aceleași coordonate  $x$ ,  $y$ ,  $z$  sau o distanță fixată sau axele a doi cilindri sau a două fețe trebuie să fie paralele.

Restricțiile pot ajuta la menținerea unor astfel de relații și astfel să permită un nivel înalt de modelare. Elementele modelului nu trebuie să fie poziționate explicit în spațiul 3D, dar pot fi definite în relație unele cu altele. Toleranțele pot fi specificate de asemenea prin restricții.

O restricție descrie o relație între două elemente în definirea unui obiect generic. După ce utilizatorul a stabilit un set de restricții printr-o definiție generică, se pot genera diverse instanțieri ale obiectului care satisfac restricțiile, variind parametrii din definiția obiectului. Restricțiile sunt menținute automat de către sistemul de modelare. De exemplu, dacă o față este declarată ca fiind paralelă cu o altă față și a doua față este rotită, prima față este și ea rotită automat, astfel încât cele două fețe să rămână paralele. Acest tip de modelare este modelarea bazată pe restricții.

Pentru un model pot fi specificate mai multe restricții. Prelucrarea restricțiilor pentru anumite seturi de valori ale parametrilor cu ajutorul sistemului de modelare se numește rezolvarea restricțiilor. Acest proces poate fi foarte complicat, în special dacă multe restricții sunt specificate din cauza interdependenței dintre diverse restricții. Timpul de calcul poate crește foarte mult dacă sunt specificate multe restricții. De asemenea, pot să apară restricții contradictorii (să nu existe nici o configurație de obiecte care să

satisfacă toate restricțiile sau setul de restricții să fie incomplet). Aceste situații sunt denumite subconstrânse, respectiv supraconstrânse.

## 4. Reprezentarea trăsăturilor

Modurile de abordare cele mai importante pentru reprezentarea elementară a TTF sunt strâns legate de cele mai importante reprezentări utilizate în modelarea solidă: modele de frontieră cu trăsături, modele CSG cu trăsături și modele cu trăsături bazate pe semispații. Relațiile dintre trăsăturile elementare într-o trăsătură compusă și relațiile de adiacență dintre trăsături pot fi descrise printr-un graf al trăsăturilor.

### a) Modele de frontieră cu trăsături

Sunt posibile două tipuri de modele cu trăsături utilizând reprezentarea pe frontieră pentru reprezentarea TTF: trăsături de suprafață și trăsături de volum.

Trăsăturile de suprafață sunt TTF reprezentate printr-un număr de fețe (și muchii și vârfuri) care nu formează un volum închis.

Trăsăturile de volum sunt TTF reprezentate printr-un număr de fețe care formează un volum închis. Avantajele utilizării trăsăturilor de volum:

- ștergerea și poziționarea trăsăturilor de volum sunt simple din cauză că fețele care închid o trăsătură volum, indică vechea frontieră a obiectului; utilizând trăsăturile de suprafață, această informație este pierdută;
- reprezentarea unor trăsături mai complexe compuse dintr-un număr de trăsături simple este mai simplă utilizând trăsăturile de volum;
- pentru operațiile de fabricație, cum ar fi frezarea sau găurirea, suprafața de pornire a sculei este prezentă în trăsăturile de volum sub forma unei fețe închise; utilizând trăsături de suprafață, această informație nu există.

Dezavantajul utilizării trăsăturilor de volum este complexitatea mare a modelului: fețele sunt incluse în model, dar nu aparțin geometriei obiectului.

Avantajele importante ale utilizării modelelor de frontieră cu trăsături sunt:

- trăsăturile pot fi conectate cu fețe individuale, muchii sau vârfuri, acestea fiind prezente explicit în model;
- detectarea interferenței între două trăsături, o operație importantă în unele aplicații, este relativ simplă.

#### b) Modelele CSG cu trăsături.

Într-un model CSG cu trăsături, trăsăturile sunt reprezentate prin obiecte primitive care sunt combinate cu obiecte inițiale printr-un set de operatori, cu preponderență operatorul de diferență.

Avantajele importante ale CSG cu trăsături sunt:

- legătura strânsă între entități, cum ar fi parametrii și axe în trăsături și în modele CSG;
- un model poate fi construit într-un mod simplu, rapid și intuitiv;
- un model este întotdeauna valid.

#### c) Modele cu trăsături bazate pe semispații

Modelele care utilizează reprezentarea bazată pe semispații stau la baza combinării metodelor de frontieră cu trăsături cu modele CSG cu trăsături. Semispațiile sunt o tehnică de reprezentare pentru primitive într-o reprezentare CSG: o primitivă poate fi reprezentată ca intersecția unui număr de semispații. Trăsăturile sunt, de asemenea, reprezentate printr-un număr de semispații, dar intersecția acestor semispații nu trebuie să fie în mod necesar un volum închis ca în cazul primitivelor într-o reprezentare CSG.

Avantajele importante ale modelelor cu trăsături bazate pe semispații sunt:

- trăsăturile pot fi conectate la semispații individuale;
- modelele pot fi compacte;
- semispațiile sunt strâns legate de operații de fabricație elementare.

#### d) Grafuri de trăsături

Pe lângă tehnicile de reprezentare a TTF elementare, sunt necesare tehnici de reprezentare între diferite trăsături într-un model bazat pe trăsături. În acest context se utilizează un graf al relațiilor dintre trăsături sau un graf al trăsăturilor.

Nodurile terminale în graf sunt utilizate pentru a memora informația despre trăsăturile elementare din model și legăturile dintre noduri pentru a memora relațiile dintre trăsături. Memorând într-un graf informația despre geometrie și topologie, crește eficiența realizării, a modificării și a accesării modelului.

Într-un graf de trăsături există două tipuri de relații care pot fi reprezentate. În primul rând, se pot memora relațiile între trăsăturile elementare dintr-o trăsătură compusă. Într-o trăsătură compusă este adesea posibil să se considere o trăsătură ca trăsătură de bază și celelalte trăsături ca trăsături dependente. Trăsătura de bază se definește și se poziționează prima, iar trăsăturile dependente sunt definite și poziționate relativ la trăsătura de bază.

Modificările în trăsătura de bază vor fi transferate automat trăsăturilor dependente. Trăsăturile dependente pot fi la rândul lor trăsături de bază pentru alte trăsături dependente. În acest mod se construiește o structură ierarhică, ce poate fi foarte bine prezentată printr-un graf.

În al doilea rând, se pot reprezenta relațiile de adiacență între trăsături. Aceste relații sunt importante pentru determinarea succesiunii operațiilor de fabricație.

Pentru a determina trăsături într-un model al produsului, există trei abordări de bază:

1. recunoașterea trăsăturilor pornind de la un model geometric (în care sistemul de recunoaștere identifică trăsăturile relevante pentru o aplicație care utilizează acest model);
2. construirea *ab initio* a unui model bazat pe trăsături de interes într-o aplicație dată (i. e., proiectantul nu specifică modelul geometric, ci un model al trăsăturilor);
3. conversia (automată) a trăsăturilor cuprinse într-un model dat pentru un obiect, într-un alt model al aceluiași obiect dar cu alte trăsături.

## 5. Recunoașterea trăsăturilor

Recunoașterea trăsăturilor desemnează interpretarea modelului geometric al unui obiect pentru identificarea trăsăturilor relevante ale acestuia. Trăsăturile care trebuie recunoscute sunt dependente de aplicație. Este evident că, dacă trăsăturile sunt utilizate deja din faza de proiectare sau, dacă se dispune de tehnici de conversie a trăsăturilor, se pot avea în vedere abordări mai puternice decât ar putea permite numai simpla disponibilitate a unui model pur geometric.

Recunoașterea de trăsături se poate efectua, în principiu, de către utilizator. De exemplu, utilizatorul poate puncta (selecta)

anumite entități de pe o imagine a modelului pe ecran și poate defini faptul că acestea formează o trăsătură. Deși perfect realizabilă pentru obiecte simple, cu un număr redus de trăsături, o astfel de procedură devine inaplicabilă pentru situațiile complexe.

Dacă se urmărește realizarea unei integrări complete CAD/CAM, recunoașterea de trăsături trebuie să se facă în mod automat. Desigur, se poate imagina o situație mixtă, în care o parte a unei trăsături este selectată de către utilizator, iar sistemul de recunoaștere completează în mod automat restul acestei trăsături.

Intrarea unui sistem automat de recunoaștere este reprezentată de descrierea formală a geometriei obiectului (de exemplu, un model geometric), precum și de o descriere formală a trăsăturilor generice care urmează a fi recunoscute; în mod corespunzător, ieșirea va fi reprezentată de o listă a instanțierilor recunoscute ale trăsăturilor cuprinzând parametrii și organizarea trăsăturilor generice recunoscute.

## 6. Tehnici de recunoaștere pentru modele B-rep

Tehnicile elaborate pentru cazul în care modelul geometric este o reprezentare de tip frontieră (denumite în continuare B-rep) se încadrează în două categorii: tehnici de recunoaștere (sintactică, pe reguli și pe grafuri) și tehnici utilizând descompunerea în volume.

Primele trei clase de metode, metode de recunoaștere sintactică, metode bazate pe reguli și metode bazate pe grafuri, sunt toate bazate pe potrivirea de pattern-uri pentru fețe, muchii și colțuri, dar diferă prin modul în care sunt descrise aceste pattern-uri pentru trăsături și model. Topologia și geometria de pattern-uri trebuie luate în considerare: două elemente pot fi topologic identice, dar pot să constituie o trăsătură diferită.

Cea de a patra metodă este metoda descompunerii volumelor care are un mod de abordare complet diferit.

### a) Metode de recunoaștere sintactică

Metodele de recunoaștere sintactică utilizează secvențe de elemente geometrice (segmente de dreaptă, segmente de cerc), pentru a descrie pattern-uri geometrice 2D.

O astfel de secvență de elemente geometrice este reprezentată ca un șir de coduri, fiecare cod reprezentând un element geometric. În acest șir, sunt căutate subșiruri reprezentând trăsături particulare sau, mai specific, un analizor sintactic verifică dacă subșirurile din șir pot fi

generate cu o gramatică descriind trăsăturile. Metodele de recunoaștere sintactică lucrează numai în 2D.

Problemele în 3D nu au o ordonare secvențială unică.

### b) Metode bazate pe reguli

Metodele bazate pe reguli utilizează reguli de producție pentru a descrie trăsături.

O regulă de producție specifică o serie de condiții necesare și suficiente pentru elemente din model, de exemplu convexitate, perpendicularitate sau adiacență, pentru a constitui un tip particular de trăsătură. Regulile sunt formalizate în clauze în Prolog și modelul este convertit în fapte în Prolog. Pattern-urile sunt căutate cu mecanismul de unificare standard Prolog.

Metodele bazate pe reguli se pot aplica și la trăsături generale 3D, dar scrierea de reguli pentru fiecare tip de trăsătură este incomodă, iar procesul de recunoaștere este lent.

### c) Metode bazate pe grafuri

După cum s-a prezentat anterior, graful este o structură de date comună pentru B-rep.

Nodurile reprezintă elemente de frontieră și legăturile dintre noduri reprezintă relații de adiacență între elementele de frontieră.

În recunoașterea trăsăturilor, în particular, graful față-muchie a devenit foarte cunoscut. Toate fețele sunt reprezentate prin noduri și toate muchiile prin legături între noduri (a două fețe adiacente). De fapt, un graf cu atribute sau completat, de tip față-muchie, este utilizat în recunoașterea trăsăturilor, adică un atribut care descrie clasificarea de convexitate a unei muchii este adăugat fiecărei legături.

Toate trăsăturile care trebuie să fie recunoscute sunt descrise într-un astfel de graf și astfel se obține reprezentarea pe frontiere a modelului.

Unificarea grafului (exemplu: verificarea dacă subgrafuri din graful modelului corespund trăsăturilor grafului) se face pentru a recunoaște trăsături.

Există și metode (Chuang și Henderson) care utilizează un graf vârf-muchie în locul grafului față-muchie.

Metodele bazate pe grafuri sunt cel mai frecvent utilizate în tehnicile de recunoaștere a trăsăturilor.

#### d) Metode de descompunere a volumelor

Ideea acestor metode este de a determina materialul care trebuie înlăturat dintr-o piesă de bază pentru a obține obiectul care este căutat prin trăsături și pentru a descompune acest material în mai multe volume. După cum se observă, există o corespondență foarte apropiată cu fabricația: volumele pot fi considerate ca operații-mașină.

Un algoritm utilizat este și algoritmul pentru reducerea acoperirii convexe. În acest algoritm, diferența mulțimilor între obiect și înfășurarea convexă este recursiv determinată până când rămâne un obiect convex. Obiectul poate fi reprezentat ca o secvență de operații pe volume de convexe cu operatori de reuniune și diferență.

### 7. Tehnici de recunoaștere pentru modele CSG

În modelele CSG, multe trăsături de interes sunt deja în modelul geometric sub forma primitivelor care sunt reunite cu, sau scăzute din alte primitive și, astfel, pot fi ușor recunoscute.

Recunoașterea trăsăturilor în B-rep are dificultăți inerente:

- problema generală a reprezentării prin frontiere
- unele informații spațiale sunt greu de calculat dintr-o reprezentare pe frontiere.

Acestea sunt rezolvate prin operarea pe modele CSG. Pe de altă parte, există și situații în care recunoașterea trăsăturilor este dificilă în modelele CSG:

- neunicitatea modelelor CSG (un obiect poate fi reprezentat prin mai multe modele CSG);
- natura globală a modelului CSG (primitivele care constituie anumite aspecte locale de formă geometrică și, astfel de trăsături în obiectele compuse, nu sunt în mod necesar izolate într-o parte a arborelui CSG, ci pot fi distribuite în tot arborele).

Prima soluție este restricționarea complexității modelelor CSG prin restricționarea rangului primitivelor, a orientării și interacțiunii spațiale cu alte primitive și tipuri de expresii permise; aceasta poate conduce uneori la restricții neacceptabile în tipurile de modele care pot fi manipulate.

A doua soluție este utilizarea unor "șabloane" pentru trăsături constând din combinații de primitive și încercând potrivirea lor cu modelul CSG.

A treia soluție este de a încerca utilizarea unui gen de reprezentare pe frontiere care este mai ușor de calculat decât reprezentarea pe frontiere tradițională, permițând calcule explicitate cât mai multe posibile și este, în plus, mai bine orientat spațial.

O metodă de rearanjare a modelului CSG pentru o clasă restrânsă de obiecte a fost propusă de Lee și Fu. Se pornește de la faptul că modelele CSG nu sunt unice și sunt greu de unificat, fiind necesară dezvoltarea unei tehnici care să unifice cel puțin reprezentările prin trăsături.

Problema semnalată este: cum pot fi transformate configurații echivalente de primitive corespunzătoare unei trăsături într-o reprezentare unificată, dorită în vederea prelucrării.

Se utilizează axele principale ale primitivelor modelului; în primul rând, toate aceste axe sunt determinate și partiționate în grupuri bazate pe relații spațiale. În aceste grupuri, axele presupuse într-o trăsătură pot fi localizate în concordanță cu condițiile specificate pentru această trăsătură.

Pasul final este unificarea reprezentărilor trăsăturilor prin restructurarea arborelui CSG, grupând mai întâi împreună în arborele CSG nodurile care se leagă unele de altele, dar care sunt răspândite în arborele CSG original.

După ce nodurile care participă la o trăsătură sunt grupate într-un subarbore în acord cu o ordine unificată, reprezentarea trăsăturii devine, de asemenea, unificată sau poate fi acum transformată într-o reprezentare unificată prin înlocuirea subarborelui cu un subarbore echivalent.

Abordarea lui Lee și Fu este foarte interesantă, dar este problematică generalizarea la modele geometrice arbitrare și trăsături.

### 8. Probleme deschise în recunoașterea trăsăturilor

Recunoașterea trăsăturilor oferă o cale de a extrage trăsături dintr-un model geometric fără multe alte informații orientate pe trăsături cunoscute despre obiecte.

Pentru această abordare există probleme inerente:

- procesul de recunoaștere este foarte complex și adesea operează numai pentru o clasă restrânsă de geometrii și, în special, în recunoașterea trăsăturilor

compuse complexe, ceea ce poate conduce ușor la erori;

- algoritmi de recunoaștere sunt dependenți de aplicație (pot recunoaște numai trăsături pentru aplicațiile pentru care au fost programați);
- informațiile care nu sunt prezentate în modelul geometric (toleranțe, suprafețe) nu pot fi recunoscute.

Într-un sens, recunoașterea trăsăturilor este o activitate redundantă: în faza de proiectare, informația de nivel înalt referitoare la produsul conceptual, este translatată de proiectant în informație geometrică de nivel scăzut, care este prelucrată pe rând în faza de recunoaștere a trăsăturii pentru a acoperi din nou informația de nivel înalt aferentă produsului.

Extragerea trăsăturilor ar putea fi mult mai ușoară, dacă modelul produsului ar conține o descriere de nivel înalt în locul unei descrieri geometrice de nivel scăzut, încă de la început. Acesta este cazul în care se aplică proiectarea prin trăsături.

#### a) Tehnici de recunoaștere de forme

Recunoașterea de forme are ca obiect descrierea și/sau clasificarea măsurătorilor preluate din procese fizice sau mentale. Se dorește o exprimare cât mai redusă a entităților considerate, cu o pondere cât mai redusă de informație, utilă operației de recunoaștere. Într-un anumit sens, aceasta este o problemă de codare și reprezintă primul pas pentru orice procedură de recunoaștere.

Recunoașterea propriu-zisă a formelor este al doilea pas al procedurii și reprezintă o problemă de clasificare și/sau o problemă de descriere:

- clasificarea formelor, care înseamnă plasarea unei entități (forme) necunoscute într-una din cele  $M$  clase ( $M > 2$ ) predefinite;
- descrierea formelor, care înseamnă reprezentarea unor forme complexe ca o compoziție (juxtapunere) de subforme simple.

Clasificarea utilizează, în special, metode matematice de teoria deciziilor statistice și, acest demers a primit numele de recunoaștere statistică a formelor.

Descrierea formelor utilizează metode din teoria limbajelor formale pentru a reprezenta structura obiectelor, demersul corespunzător purtând numele de recunoaștere sintactică (sau structurală) a formelor.

În recunoașterea statistică, o formă este reprezentată printr-un vector  $n$ -dimensional al caracteristicilor și decizia de clasificare se bazează pe o anumită măsură a similarității.

În recunoașterea sintactică, o formă este reprezentată ca o secvență, un arbore sau un graf de forme primitive, împreună cu relații ale acestora. O astfel de abordare stabilește un izomorfism între structura formelor și sintaxa unui limbaj. Procesul de clasificare devine o operație de analiză sintactică, de evaluare a distanței dintre șirurile de primitive, a distanței dintre arbori sau a distanței dintre grafuri.

Dacă formele de recunoscut sunt foarte complexe, implicând un număr mare de caracteristici și/sau dacă numărul de clase este foarte mare (așa cum se întâmplă, de exemplu, în cazul reprezentărilor CSG), atunci este nepractic să se realizeze o abordare statistică și se recurge la una sintactică (structurală). Prin această abordare se reprezintă o formă complexă ca o combinație de forme mai simple luând în considerație relațiile dintre subforme. Dacă fiecare subformă este la rândul ei complexă, aceasta se reprezintă prin subforme și mai simple, până când se ajunge la cele mai simple subforme denumite primitive (sau primare), acestea putând fi recunoscute prin metode de teoria deciziilor statistice. Acest demers stabilește o analogie între structura formelor și sintaxa limbajelor.

În recunoașterea sintactică, o formă este reprezentată ca un șir sau arbore (tree) de subforme. Limbajul care asigură o descriere structurală a formelor, se numește limbaj de descriere a formelor, iar regulile care guvernează compoziția primitivelor în forme sunt specificate de gramatici ale limbajului.

Prima etapă a abordării sintactice o reprezintă selecția primitivelor. Pentru selecția primitivelor formelor nu există o regulă generală. Primitivele sunt componente de bază ale unei forme și se presupune că sunt ușor de recunoscut. Acestea trebuie să ofere o descriere compactă și adecvată a datelor în funcție de relațiile de structură și, în plus, ele trebuie să fie ușor de extras, de codificat și de recunoscut.

După ce au fost selectate primitivele formelor, pasul următor este construcția uneia sau a mai multor gramatici care vor genera unul sau mai multe limbaje pentru descrierea formelor aflate în studiu. Construcția unei gramatici se bazează pe cunoașterea existentă a priori asupra procesului și pe experiența proiectantului. Creșterea puterii unei gramatici contribuie la creșterea complexității sistemului de analiză sintactică, permițând recunoașterea. Automatele cu stări finite sunt capabile să recunoască sau să accepte limbajele cu stări finite, deși puterea acestor limbaje este mai mică decât cea a limbajelor independente de



context (necontextuale) și a celor contextuale. Pe de altă parte, pentru acceptarea limbajelor generate de gramaticile necontextuale sau contextuale, sunt necesare procedee infinite și nedeterministe. Deci, selecția unei gramatici pentru recunoașterea formelor este funcție de primitivele selectate și de compromisul între puterea descriptivă a gramaticii și eficiența analizei sintactice.

**Procedeul Fu-Lee:** un agent uman (eventual folosind un monitor pentru grafica interactivă) descompune fiecare cuvânt din eșantionul  $C_+ = \{X_1, X_2, \dots\}$  în  $n$  subcuvinte distincte. Fiecare set de subcuvinte reprezintă un set de subforme cu structuri simple, care poate fi descris printr-o gramatică. Pentru segmentarea formelor în subforme se pot utiliza etapele:

1. se selectează subformele cu structuri simple astfel încât gramaticile să fie ușor de construit;
2. se segmentează formele astfel încât să apară cât mai multe subforme repetate.

După ce mulțimea eșantion de cuvinte  $C_+$  a fost împărțită în  $n$  mulțimi de subcuvinte  $C_1, C_2, \dots, C_n$ , putem construi  $n$  gramatici  $G_1, G_2, \dots, G_n$  pentru fiecare mulțime de subcuvinte:

Fie  $G_j = (V_N^{(j)}, V_T^{(j)}, P^{(j)}, S^{(j)})$  și gramatica descriind toate formele  $G = (V_N, V_T, P, S)$ .

Fiecare mulțime de terminale  $V_T^{(j)}$  este inclusă în  $V_T$ , cu excepția, eventual, a unei mulțimi de conectori  $C_j$ . Simbolurile  $C_j$  sunt considerate ca terminale în  $G_j$  și ca neterminale în  $G$  adică  $C_j$  aparține lui  $V_T^{(j)}$  și  $C_j$  lui  $V_N$ .

## b) Rețele neuronale

O rețea neuronală este alcătuită dintr-o mulțime de noduri în care se găsesc elemente de procesare, denumite neuroni. Ele reprezintă elemente de calcul neliniare, care operează în paralel și sunt organizate în moduri similare celor din rețelele biologice.

Prin analogie cu neuronul biologic, un neuron artificial are un număr  $N$  de intrări și o singură ieșire, dar care se poate conecta la intrările mai multor neuroni. Fiecare intrare are asociată o pondere care reprezintă importanța pe care un impuls prezent la intrarea respectivă o are în activarea neuronului. Conexiunile se realizează prin legarea ieșirilor unor neuroni la intrările altora. Impulsurile de la intrările unor neuroni pot avea un caracter excitant (impulsuri pozitive) sau inhibitor (impulsuri negative, totalitatea lor determinând activarea sau nu a respectivului neuron. Neuronul artificial este caracterizat, de

asemenea, de un prag de sensibilitate, respectiv de o funcție de activare neliniară.

Ponderile asociate intrărilor unui neuron se notează  $W_i$ . Ele multiplică valorile impulsurilor  $X_i$ , corespunzătoare respectivelor intrări. Neuronul calculează suma ponderată a tuturor impulsurilor de intrare din care scade valoarea de prag pentru a determina dacă și când combinația de la intrare a depășit această valoare. În funcție de rezultatul calculului, precum și de tipul funcției de activare caracteristice neuronului, acesta generează un semnal  $Y$  la ieșire.

Există mai multe tipuri de neliniarități specifice comportării unui neuron: funcția treaptă, funcția rampă, funcția sigmoidală etc..

În cadrul unor rețele neuronale complexe, se pot întâlni variații temporale ale parametrilor, precum și operații mai complicate în locul simplei însumări.

Elementele de procesare sunt organizate în mai multe straturi (niveluri). Introducerea informațiilor în rețea se face prin intermediul unui strat de neuroni de intrare, iar rezultatele se obțin printr-unul de ieșire. Între aceste două straturi care asigură comunicarea cu mediul înconjurător este posibilă existența unor straturi intermediare, numite și straturi ascunse. Legăturile între neuronii din diversele straturi sunt specificate printr-o matrice de conexiuni ponderate.

Asigurarea de performanțe ridicate se face prin gradul ridicat de interconectare care determină un paralelism accentuat. Datorită acestuia, tehnicile de prelucrare a informațiilor, specifice calculatoarelor neuronale, sunt preferate calculatoarelor clasice într-o serie de aplicații.

Principalele avantaje prezentate de un neurocalculator sunt:

- toleranța la defecte (continuarea normală a activității rețelei chiar și în cazul în care un număr de neuroni individuali au încetat să mai funcționeze corect);
- recuperarea asociativă (regăsirea informațiilor stocate pe bază de conținut și formularea unor răspunsuri adecvate, chiar dacă nu se găsește un corespondent extern la datele de intrare);
- degradare lentă (recuperarea situațiilor de încetare a funcționării unor neuroni).

Diferența față de un calculator secvențial constă tocmai în existența unui număr foarte mare de elemente de procesare, care operează în paralel.

## 9. Conversia trăsăturilor

Conversia trăsăturilor, numită, de asemenea, transformarea trăsăturilor și maparea trăsăturilor, este definită ca proces de transformare a trăsăturilor unui obiect dintr-un punct de vedere, în trăsături ale aceluiași obiect, văzute din alt punct de vedere. Conversia trăsăturilor este necesară, în particular, pentru a permite conceptul de inginerie concurentă cu multiple puncte de vedere ale trăsăturilor. În implementări relativ simple, pentru fiecare aplicație, trăsăturile de proiectare sunt convertite automat în trăsături specifice pentru aceste aplicații. Toate modificările trebuie făcute pornind de la trăsăturile de proiectare.

## Bibliografie

1. BRAID, I.C.: Geometric Modelling. In: *Advances in Computer Graphics I* (G. Enderle, M.Grave, and F. Lillehagen, Eds.), Springer-Verlag, Berlin, pp.325-362.
2. BRONSVOORT, W.F., F.W. JANSEN: Feature Modelling and Conversion-Key Concepts to Concurrent Engineering. In: *Computers in Industry*, Vol.21, No.1, 1993, pp.61-86.
3. LEAU, J.J., TSAL, W.H.: The Minimum Feature Point Set Representing a Convex Polyhedral Object. In: *Pattern Recognition Letters*, Vol.11, No.3, 1990, pp.225-229.
4. LEE, Y.C., FU, S.K.: Machine Understanding of CSG: Extraction and Unification of Manufacturing Features. In: *IEEE Transactions on Computer Graphics and Applications*, January 1987, 1987, pp.20-32.
5. MEHLHORN, K.: *Multi-dimensional Searching and Computational Geometry*, Springer-Verlag, Berlin, 1984.
6. PATEL, M.: A Method for The Representation, Evaluation and Display of CSG Models in PHIGS and PHIGS+. In: *Computer Graphics Forum*, Vol.8, 1989, pp.337-345.
7. PRATT, M.J. : Interactive Geometric Modelling for Integrated CAD/CAM. In: *Advances in Computer Graphics I* (G. Enderle, M.Grave, and F. Lillehagen, Eds.), Springer-Verlag, Berlin, pp.363-380.
8. PURFURST, R.: Ein Tiefenpufferalgorithmus. In: *CAD und Computergraphik*, Vol.12, No.3/4, 1989, pp.93-96.
9. QIAO, L.H., ZHANG, C., Liu, T.H., BEN WANG, H.P., FISCHER, G.W.: A PDES/STEP-based Product Data Preparation Procedure for Computer-Aided Process Planning. In: *Computers in Industry*, Vol.21, No.1, 1993, pp.11-22.
10. REQUICHA, A.A.G.: Representations for Rigid Solids: Theory, Methods, and Systems. In: *ACM Computing Survery*, Vol.12, No.4, 1980, pp.437-464.
11. ROGERS, D.F., EARNSHAW, R.A.: *Techniques for Computer Graphics*, Springer-Verlag, Berlin, 1987.
12. ROGERS, D.F., ADAMS, A.J.: *Mathematical Elements for Computer Graphics* (Second Edition), Mc.Graw-Hill, New York, 1989.
13. ROSSIGNAC, J.R., VOELCKER, H.B.: Active Zones in CSG for Accelerating Boundary Evaluation, Redundancy Elimination, Interference Detection, and Shading Algorithms. In: *ACM Transactions on Graphics*, Vol.8, No.1, 1989, pp. 51-87.
14. ROSSIGNAC, J.R., REQUICHA, A.A.G.: Piecewise-Circular Curves for Geometric Modelling. In: *IBM Journal of Research and Development*, Vol.31, No.3, 1987, pp. 296-313.
15. SAMET, H.: Bibliography on Quadrees and Related Hierarchical Data Structures. In: *Data Structures for Raster Graphics* (L.R.A. Kessener, F.J. Peters and M.L.P. van Lierep, Eds.), Springer-Verlag, Berlin, 1986, pp. 9-38.
16. WOLFE, R.N., WESLEY, M.A., KYLE, J.C.Jr., GRACER, F., FITZGERALD, W.J.: Solid Modelling for Production Design. In: *IBM Journal of Research and Development*, Vol.31, No.3, 1987, pp.277-295.