

6. PROIECTAREA ASISTATĂ DE CALCULATOR (CAD)

6.1. Proiectarea axiomatică

6.1.1. Introducere

Tehnologia de proiectare axiomatică reduce riscurile de dezvoltare ale produsului, reduce costurile și crește viteza de analiză a pieții [6.41]. Pașii de realizat în proiectarea axiomatică (indiferent că este vorba de produs, serviciu, software sau proces) sunt concretizați astfel:

- înțelegerea necesităților enunțate de utilizator / consumator;
- definirea problemei care poate rezolva aceste necesități;
- crearea și selectarea soluției;
- analiza și optimizarea soluției;
- verificarea rezultatelor proiectării în contrast cu cerințele utilizatorului.

Proiectarea axiomatică are la bază 3 concepte:

- *conceptul domeniului;*
- *conceptul ierarhic;*
- *conceptul zi-zag*

și 2 axiome de proiectare.

6.1.2. Conceptul domeniului

Conceptul fundamental al proiectării axiomatică este cel de domeniu de lucru evidențiat pentru procesul de proiectare reprezentat în figura 6.1.

În ordine ierarhică, de la stânga la dreapta, fiecare pereche domeniu definește o dependență: domeniul din stânga înseamnă scopul proiectării iar cel din dreapta soluția de proiectare.

Conținutul, semnificația și simbolizarea utilizată pentru fiecare domeniu se poate descrie conform cu tabelul 6.1.

Deciziile dintr-un domeniu sunt reprezentate în domeniul din dreapta conform cu un principiu: un "CE ?" pentru domeniu din stânga este reprezentat în domeniul adiacent prin "CUM". Această echivalență este dată de matricea de proiectare ca relație dintre {FRs} și {DPs} sau {DPs} și {PVs}. Un exemplu de matrice de proiectare este prezentat în figura 6.2

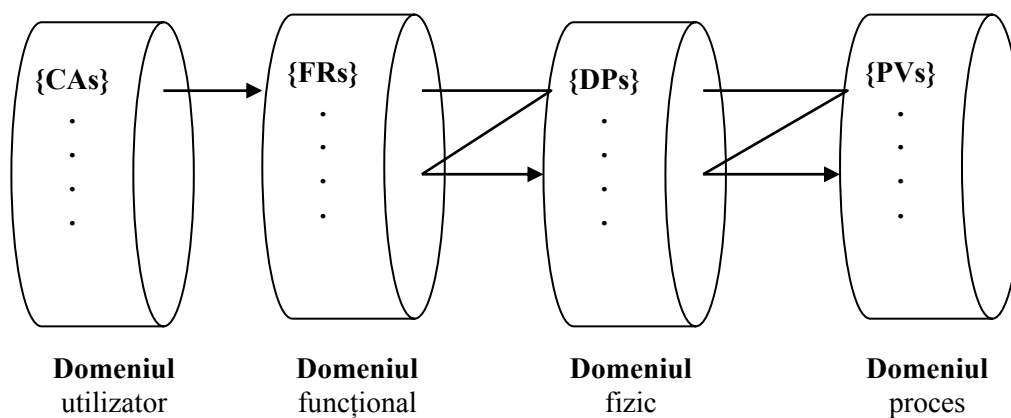


Fig. 6.1 Conceptul proiectării axiomatice

Tabelul 6.1

Utilizator	Cumulează beneficiile urmărite din partea utilizatorului.
Funcțional	Cumulează cerințele funcționale pentru soluțiile procesului de proiectare, notație {FRs} (F unctional R equirements)
Fizic	Cumulează totalitatea parametrilor de proiectare pentru soluția din proiect, notație {DPs} (D esign P arameters)
Proces	Cumulează totalitatea variabilelor procesului, notație {PVs}

	DP1	DP2	DP3
FR1	X	X	0
FR2	X	0	0
FR3	X	X	X

a)

	DP1	DP2	DP3
FR1	X	X	0
FR2	X	0	0
FR3	X	X	X

b)

	DP1	DP2	DP3
FR1	X	X	0
FR2	X	0	0
FR3	X	X	X

c)

Fig. 6.2 Matricea de proiectare

6.1.3. Conceptul ierarhizării și zig-zag

Procesul de proiectare progresaază de la nivelul cel mai ridicat de abstractizare spre nivele cu detalii multiple, de la sistem la subsisteme, de la ansamblu la

caracteristicile componentelor.

Ierarhizarea există în oricare din domeniile specificate: funcțional, fizic, proces. O descompunere ierarhică cu utilizarea conceptului zig-zag este ilustrată în figura 6.3

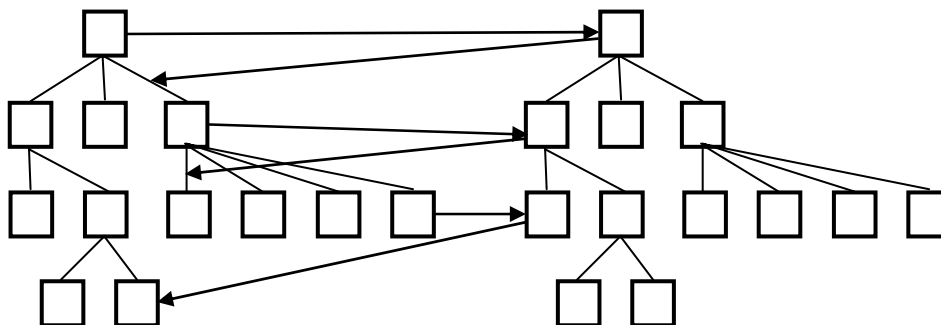


Fig. 6.3 Conceptul zig - zag

Deciziile luate la nivelele superioare, afectează formularea problemei la nivelul inferior. Proiectantul va executa astfel un proces în zig-zag între domenii (fig.6.3). La orice nivel de abstractizare există un set **FRs**. Înainte ca acest set să fie descompus se pot selecta corespondențele **DPs** și **PVs**. O singură corespondență **DP** poate satisface **FR** și **DP** care este satisfăcut de **PV**. **FR** se poate descompune într-un nivel inferior al cerințelor.

Utilizarea axiomelor de proiectare și a bazei de cunoștințe, permite proiectantului să aleagă soluțiile cele mai bune.

6.1.4. Axiomele de proiectare

Proiectantul în procesul de proiectare urmează un algoritm de decizie pornind de la nivelul superior și coborând spre nivelul inferior, componenta.

Axiomele de proiectare servesc ca unelte de analiză. Două axiome sunt precizate în literatura de specialitate [6.41]:

- Axioma independenței – susține independența cerințelor funcționale. Secțiunile de proiectare trebuie să fie separabile astfel că schimbările efectuate într-o secțiune să nu afecteze (pe cât posibil) restul proiectului.
- Axioma informației – se referă la informația conținută într-un proiect de produs, la complexitatea proiectului. Se recomandă ca informația să fie minimă.

Dacă un set de cerințe funcționale **FRs** au fost bine precizate și s-a reușit și sintetizarea setului de parametri de proiectare **DPs**, cele două axiome se pot folosi pentru evaluarea proiectului propus.

6.1.5. Matricea de proiectare

Utilizând reprezentarea sistemică, cerințele funcționale **FRs** se constituie în mărimea de ieșire în timp ce parametri de proiectare **DPs** corespund mărimii de intrare. Legătura dintre o cerință funcțională și parametri de proiectare aferenți

exprimă funcția de transfer f a sistemului:

$$FR = f(DP_1, DP_2, \dots, DP_n) \quad (6.1)$$

S-a arătat în capitolul 4 legătura dintre variațiile la nivelul cerinței funcționale determinate de variațiile parametrilor de proiectare. Pentru FRs multiple funcții asemănătoare cu (6.1) definesc corespondența cu parametrii de proiectare :

$$\begin{cases} FR_1 = f_1(DP_1, DP_2, \dots, DP_n) \\ FR_2 = f_2(DP_1, DP_2, \dots, DP_n) \\ \dots\dots\dots \\ FR_n = f_n(DP_1, DP_2, \dots, DP_n) \end{cases} \quad (6.2)$$

sau

$$\mathbf{FR} = \mathbf{f}(\mathbf{DP}) \quad (6.3)$$

Matricea de proiectare (MP) - $[A]$, pune în evidență relația dintre două seturi de date la un nivel de ierarhizare, **FRs** și **DPs**, **DPs** și **PVs** sub forma unei ecuații de proiectare :

$$\begin{Bmatrix} FR_1 \\ FR_2 \\ \dots \\ FR_n \end{Bmatrix} = [A] \cdot \begin{Bmatrix} DP_1 \\ DP_2 \\ \dots \\ DP_n \end{Bmatrix} \quad (6.4)$$

$$\begin{Bmatrix} \Delta FR_1 \\ \Delta FR_2 \\ \dots \\ \Delta FR_n \end{Bmatrix} = \begin{bmatrix} \frac{\partial FR_1}{\partial DP_1} & \frac{\partial FR_1}{\partial DP_2} & \frac{\partial FR_1}{\partial DP_n} \\ \frac{\partial FR_2}{\partial DP_1} & \frac{\partial FR_2}{\partial DP_2} & \frac{\partial FR_2}{\partial DP_n} \\ \dots & \dots & \dots \\ \frac{\partial FR_n}{\partial DP_1} & \frac{\partial FR_n}{\partial DP_2} & \frac{\partial FR_n}{\partial DP_n} \end{bmatrix} \cdot \begin{Bmatrix} \Delta DP_1 \\ \Delta DP_2 \\ \dots \\ \Delta DP_n \end{Bmatrix} \quad (6.5)$$

Elementele matricii de proiectare se determină din matricea de sensibilitate scrisă pe baza ecuației anterioare. Matricea $[A]$ prin valorile sale indică efectele dintre parametrii de proiectare și cerințe. Valoarea zero indică lipsa unui efect relativ la toleranțele asociate în timp ce o valoare « x » indică un efect mai mult sau mai puțin puternic. Referitor la matricile prezentate în figura 6.2 acestea sunt cu un efect: a) de cuplare; b) decuplate; c) necuplate.

Ideal ar fi ca proiectantul să poată avea o matrice pătrată din categoria matricilor unitate. Practic se realizează un compromis prin admiterea unor posibilități de ajustare :

- Prin separarea elementelor de proiectat se poate obține o matrice decuplată;
- Minimizarea cerințelor funcționale FRs simplifică procesul de proiectare;
- Utilizarea componentelor integrate, dacă este posibil și fără compromisuri semnificative, reduce numărul de componente;
- Standardizarea tinde să satisfacă cerințele axiomelor de proiectare. În plus această opțiune reduce informația conținută;
- Simetria, dacă este posibilă, reduce semnificativ informația inclusă în proiect;
- Câmpul de toleranțe se recomandă a fi cât mai larg posibil pentru a reduce informația conținută;
- Utilizarea soluției de necuplare în matricea de proiectare și minimizarea informației este o țintă care trebuie urmărită de fiecare proiectant.

6.1.6. Cantitatea de informație

În general se consideră că se primește o informație atunci când se obțin despre un subiect cunoștințe mai complete decât cele avute anterior. Teoria informației, sub forma elaborată de C. Shannon, evită orice element subiectiv din definirea noțiunii de cantitate de informație. *Cantitatea de informație* primită depinde de raportul dintre numărul răspunsurilor considerate corecte înainte și după primirea informației [6.32].

În general pentru evenimente echiprobabile se poate defini cantitatea de informație I ca fiind :

$$I = \log_2 N \quad (6.6)$$

unde N este numărul de cazuri posibile.

Prin definiție, diferența dintre entropia informațională înainte și după primirea informației constituie cantitatea de informație primită:

$$I = H(p_i) - H(p_f) = -\sum_{i=1}^n p_i \cdot \log_2 p_i + \sum_{i=1}^n p_f \cdot \log_2 p_i' \quad (6.7)$$

unde p_i este probabilitatea inițială iar p_f este probabilitatea finală pentru evenimentul i din cele n disponibile.

În axioma informației, parametrii de proiectare sunt selectați în corespondență cu informația conținută. În general informația conținută este definită prin probabilitatea de succes de a satisface cerința funcțională [6.38] :

$$I = \log_2 \frac{1}{P_s} \quad (6.8)$$

$$P_s = \frac{\text{domeniu_comun}}{\text{domeniul_sistemului}} \quad (6.9)$$

unde P_s este probabilitatea de succes. Dacă sunt posibile mai multe variante de soluții pornind de la axioma independenței, se recomandă să se aleagă varianta care satisface condiția de minim :

$$I_{\text{var}} = \min \{I_j; j = 1, 2, \dots, n\} \quad (6.10)$$

Pentru componentele mecanice informația I conținută și atribuită componente se

poate exprima sub forma :

$$I = \sum_{i=1}^n \left[\log \left(\frac{l_i}{T_i} \right) \right] \quad (6.11)$$

unde n este numărul de cote cu toleranțe, l_i este dimensiunea de gabarit pentru caracteristica « i » iar T_i este toleranța pentru dimensiunea « i ».

Existența posibilității de calcul a coeficientului de capabilitate pentru caracteristica « i » permite evaluarea informației conform relației :

$$I = \sum_{i=1}^n 10 \cdot \log C_{pki} \quad (6.12)$$

Algoritmul de desfășurare a procesului de proiectare prin integrarea celor două axiome este prezentat în figura 6.4.

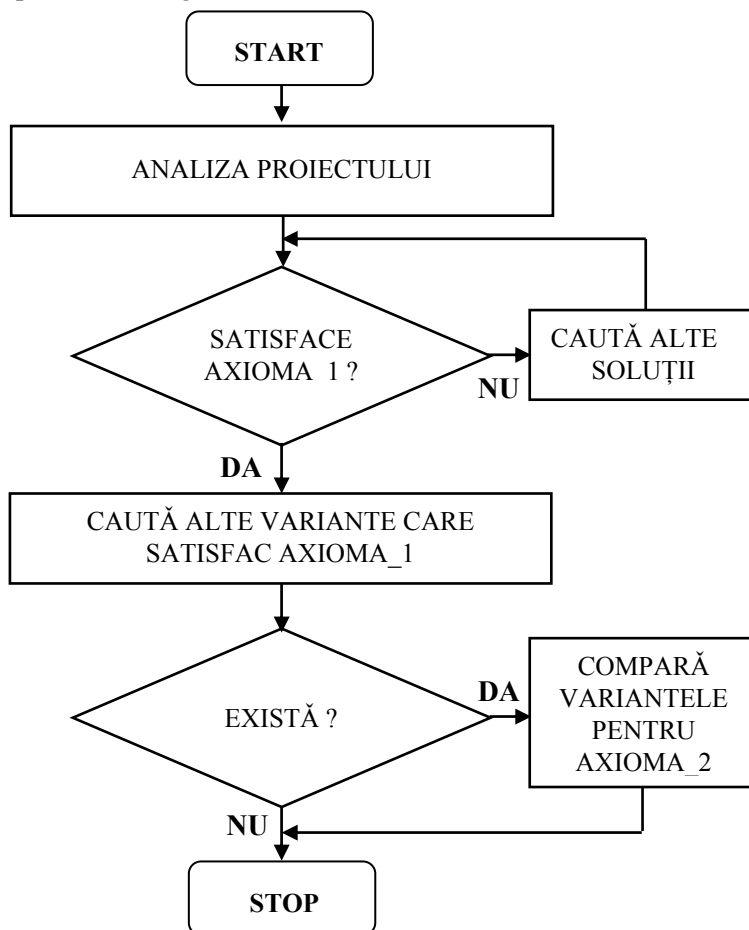


Fig. 6.4 Algoritmul de proiectare

Modalități de evaluare a acestei informații, pentru forme posibile ale matricii de proiectare, sunt prezentate în [6.38] : metoda grafică, metoda de integrare.

6.1.7. Generalizări în proiectarea axiomatică

Axiomele de proiectare au creat cadrul pentru o proiectare metodologică. În acest fel în scurt timp proiectarea axiomatică a fost utilizată cu succes într-o serie de aplicații ingineresti.

Au fost dezvoltate teorii referitoare la aceste aplicații sau cu un caracter general. Analiza costului efectiv și al complexității este abordată [6.18]. Pe baza axiomei de independență a fost dezvoltată metodologia de optimizare structurală [6.27].

Proiectarea produselor de calitate, la un preț de cost scăzut este un deziderat al oricărui proiectant. Este un motiv pentru care problema se regăsește și în abordările teoretice de conexiune proiectare axiomatică – metodele Taguchi având ca suport informația inclusă în proiect [6.25]. Proiectarea 6 sigma este corelată de asemenea prin metodologia axiomatică [6.4].

Analiza fiabilității unui sistem prin utilizarea conceptului axiomatic este concretizată în [6.3] pentru ușa glisantă a automobilului Fiat. Este o dezvoltare a metodei FMEA (Failure Modes and Effects Analysis) prin conceptul axiomatic. Reamintim că metoda FMEA este utilizată pentru a identifica modurile posibile de defectare, cauzele și efectele acestor defecte. Se utilizează în acest sens matricea de corelare (MC) și matricea de risc (MR). Prima se constituie într-un document în care se regăsesc toate funcțiile produsului, toate componentele și corelațiile dintre acestea. A doua matrice descrie aceste aspecte sub forma FMEA. Algoritmul de lucru are la bază:

- Aplicarea axiomelor de proiectare asupra sistemului studiat;
- Alcătuirea matricii de corelare;
- Evaluarea rezultatelor obținute prin comparare cu standardele întreprinderii;
- Identificarea elementelor critice;
- Interpretarea FMEA asupra acestor elemente;
- Memorarea rezultatelor și eventual corecția standardelor.

O unificare a proiectării axiomatică și a celei robuste prin funcția de transfer este susținută în [6.19]. Se întărește astfel aspectul de generalitate a conceptului de proiectare axiomatică și de abordare sistemică. Introducând în mod direct concretizarea cerinței funcționale sub formă cantitativă (funcție dependentă de parametrii de proiectare) se prezintă modul de tratare a problemei în spiritul axiomatic în condițiile unui număr sporit de cerințe funcționale.

Proiectarea axiomatică se poate aplica cu succes în proiectarea sistemului software în mod distinct față de limbajele de programare sau algoritmi specifici [6.41].

Un astfel de exemplu este sistemul Acclaro destinat proiectanților care utilizează proiectarea axiomatică. Acest software de proiectare este bazat pe proiectarea axiomatică și este implementat ca o modificare a versiunii tehnicii orientate-obiect și a limbajului de programare Java. Utilizarea programării orientate-obiect (OO) este o cerință impusă de limbajul Java. Prin utilizarea proiectării axiomatică, OO este mult simplificată. Activitatea de proiectare pe conceptul domeniu este aplicabilă și în acest caz cu următoarele precizări:

- Necesitățile utilizatorului sau atributele beneficiarului pentru programul software se regăsesc în domeniul beneficiar **CAs**;
- **FRs** se identifică prin ieșirile, specificațiile sau cerințele sistemului software. FRs sunt “obiectele” din programarea OO;

- **DPs** se regăsesc ca și :
intrări pentru module în cazul algoritmilor primari (fără zone de date). În cazul programării OO, DPs pot fi tratate ca și “informații (date)”;
semnal de la senzori și aplicații specifice circuitelor integrate în cazul în care sistemele implică atât software cât și hardware;
cod program pentru generarea intrărilor destinate modulelor.

- **PVs** se concretizează ca subrutine, coduri mașină, compilatoare.

Controlul sistemelor, proiectarea sistemelor de acționare pentru micro-vehicule aeroperțate, proiectarea transmisiilor mecanice ș.a. sunt aplicații soluționate prin proiectarea axiomatică sugerând generalitatea metodologiei.

6.1.8. Analiza procesului de proiectare axiomatică

Metodologia de proiectare axiomatică se poate prezenta în mod succint sub forma algoritmului:

- Determinarea atributelor consumatorului / beneficiarului;
- Descrierea cerințelor funcționale FRs și ale parametrilor de proiectare DPs pentru procesul de proiectare convențional;
- Construirea matricii de proiectare;
- Căutarea surselor de cuplare;
- Stabilirea direcției de eliminarea cuplajelor;
- Determinarea FRs și DPs și a matricii de cuplare pentru noul proces de proiectare;
- Descompunerea în mod ierarhic a setului FRs – DPs;
- Utilizarea noilor stări în proiectare.

Proiectarea documentației necesare realizării unui tub cinescop se constituie într-o aplicație specifică proiectării axiomatice [6.35]. Schița produsului care face obiectul procesului de proiectare este prezentată în figura 6.5: partea frontală este ecranul (din sticlă) care se continuă cu partea finală – manșonul.

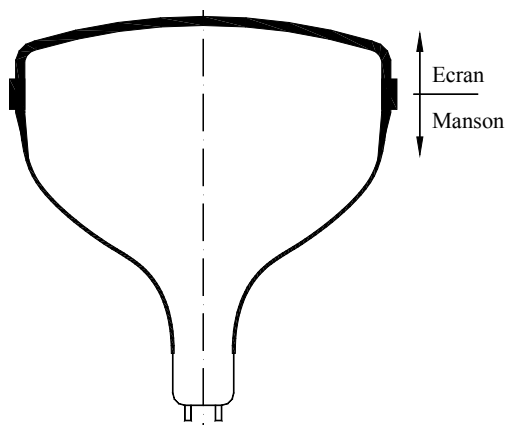


Fig. 6.5 Exemplu de obiect al procesului de proiectare

Procesul de proiectare convențional este divizat în 5 etape conform algoritmului prezentat în figura 6.6.

Primul pas constă în stabilirea bazei de date: date referitoare la curbura (raza, profilul, ecuația polinomială), specificațiile componente, dimensiunea ecranului, desene. Se stabilesc astfel care sunt cerințele funcționale:

- **FR1** – construirea bazei de date pentru noul produs;
- **FR2** – stabilirea configurației pentru produs;
- **FR3** – verificarea caracteristicilor produsului;
- **FR4** – generarea documentației grafice pentru produs.

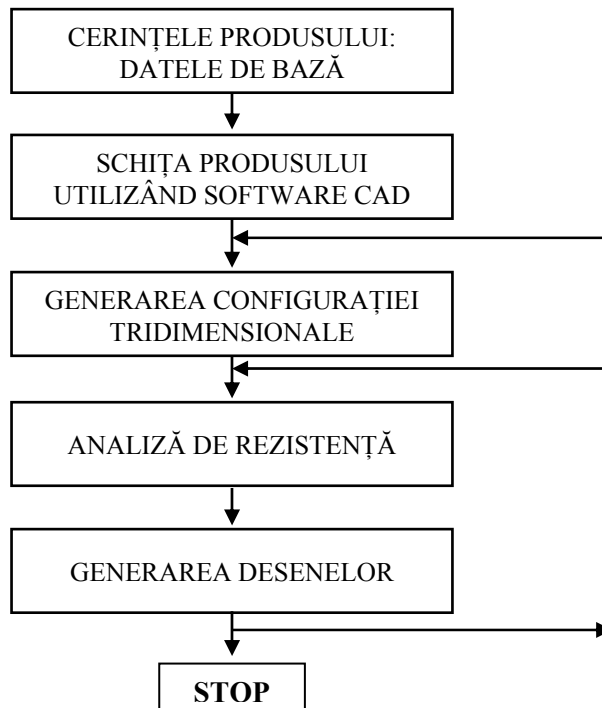
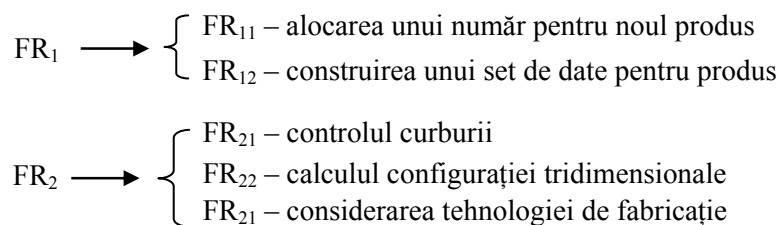


Fig. 6.6 Schema logică a procesului de proiectare

Corespunzător cu cele expuse, se pot stabili parametrii de proiectare care să satisfacă FRs:

- DP1 – setul de date pentru noul produs;
- DP2 – configurația tridimensională;
- DP3 – condițiile de lucru;
- DP4 – un set de date suplimentar pentru desen.

De la corespondența descrisă se poate trece la o descompunere ierarhică. Un exemplu edificator pentru primele două cerințe este prezentat în continuare:



În mod ierarhic asemănător, se poate stabili și setul de parametri de proiectare DP. Pentru FR11 și FR12 parametrii DP ar corespunde după cum urmează: DP11 ar reprezenta codul pentru noul produs iar DP12 un set specific de date. În final se poate stabili matricea de proiectare:

$$\begin{Bmatrix} FR_{11} \\ FR_{12} \end{Bmatrix} = \begin{Bmatrix} x & 0 \\ x & x \end{Bmatrix} \cdot \begin{Bmatrix} DP_{11} \\ DP_{12} \end{Bmatrix} \quad (6.13)$$

Pe baza matricilor de proiectare se poate defini structura modulelor de joncțiune și implicit schema bloc a software-ului de proiectare axiomatice. Structura unui astfel de software este prezentată în figura 6.7.

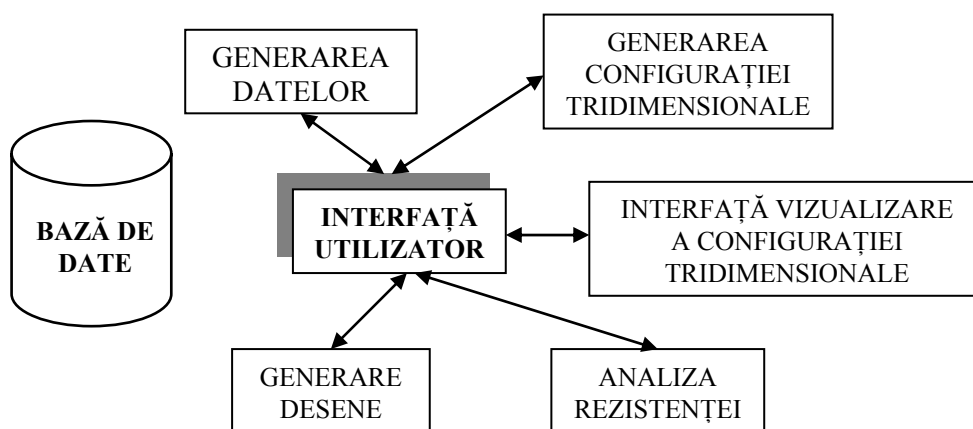


Fig. 6.7 Structura unui software pentru proiectare axiomatică

Secvențele de lucru sunt coordonate prin interfața utilizator (GUI) dezvoltată în sistemul X-Window (MOTIF).

Generarea configurației utilizează interrelații geometrice și algoritmi de analiză numerică. De la variabilele de proiectare se construiește geometria produsului. Se utilizează în acest sens o bibliotecă grafică pentru stațiile grafice HP735.

Analiza stării de solicitare utilizează programul comercial ANSYS.

Generarea părții grafice are la bază sistemul CAD comercial Unigraphics.

6.2. Proiectare și TRIZ

6.2.1. Metoda TRIZ

Metoda TRIZ este acronimul în limba rusă a teoriei de rezolvare a problemelor de invenție (Theory of inventive problem solving) introdusă de Genrikn Altshuller. Abordarea problemei pornește de la conceptul sistemelor tehnice care realizează o conversie a energiei, a materialelor și a semnalelor. În această metodologie se introduce noțiunea de “substanță” referitor la aceleași obiecte cu complexități diferite la orice nivel. TRIZ se constituie într-o metodologie analitică pentru modelarea problemelor din sistemele tehnologice. Domeniul - “substanță” (Su-Domeniu) este un

model al sistemelor tehnice minimale, acționabile și controlabile.

Principiul minimal al sistemelor tehnice este edificator în sensul celor specificate anterior. Conform acestui principiu un sistem tehnic constă dintr-un domeniu (F – de la *field*) și două “substanțe” (S₁ și S₂) cu relații reciproce conform cu figura 6.8.

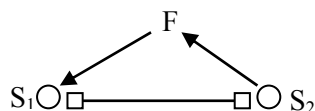


Fig. 6.8 Principiul minimal al sistemelor tehnice

Soluțiile standard se constituie deasemenea într-un instrument valoros în rezolvarea rapidă a problemei de proiectare printr-o analiză Su-Domeniu. “Soluțiile standard 76” din TRIZ au fost dezvoltate de G.S. Altshuller și colaboratorii săi între anii 1975 și 1985. Această bază de date este structurată în 5 clase conform tabelului 6.2[6.43]. Fiecare clasă este subdivizată într-o serie de grupe: clasa 1 în 2 grupe, clasa 2 în 4 grupe, clasa 3 în 2 grupe, clasa 4 în 5 grupe și clasa 5 în 5 grupe [6.40]. Drept exemplu, în tabelul 6.3 se prezintă grupele clasei 4.

Tabelul 6.2

Clasa	Categoria	Număr de soluții
1	Îmbunătățirea sistemului cu sau fără uşoare schimbări	13 soluții standard
2	Îmbunătățirea sistemului prin schimbarea acestuia	23 soluții standard
3	Tranziții ale sistemului	6 soluții standard
4	Detectie și măsurare	17 soluții standard
5	Strategii pentru simplificare și îmbunătățire	17 soluții standard

Tabelul 6.3

Grupa 4-1	În loc de măsurare și detecție – schimbă sistemul
Grupa 4-2	Sinteza sistemului de măsurare
Grupa 4-3	Extinderea sistemului de măsurare
Grupa 4-4	Tranziția spre sistemele de măsurare feromagnetice
Grupa 4-5	Evoluția sistemelor de măsurare

Metologia TRIZ se bazează pe un set de metode și strategii de lucru. Într-o enumerare sumară acestea ar fi:

- *Conceptul idealist*. Orice sistem prin funcțiile sale are efecte utile și efecte nocive. Un scop primordial al rezolvării oricărei probleme de proiectare este maximizarea efectelor utile. Conceptul idealist are două direcții scop: toate sistemele tehnice evoluează în sensul creșterii proporției de ideal; direcționarea rezolvitorului de probleme spre conceptualizarea perfecțiunii și ruperea inerției psihologice sau a paradigmatelor.
- *ARIZ* – un algoritm necomputațional pentru rezolvarea problemelor de invenție;
- *Tabelul contradicțiilor* – un instrument la dispoziția inventatorului cu un

conținut de 1263 de contradicții ingineresti. Contradicția se definește ca efectul negativ asupra unui parametru datorită unui efect pozitiv asupra altui parametru;

- *Principiile invenției* – un instrument de lucru cu 40 de principii și aproximativ 50 de subdomenii;
- *Principiul separației*
- *Legile evoluției sistemelor ingineresti*
- *Analiza funcțională*

Aceste instrumente de lucru sunt rodul unei intense activități. De exemplu cele 40 de principii pentru invenție au fost dezvoltate între anii 1940 și 1972 [6.40]. Aceste principii sunt fundamentate pe un studiu asupra 400.000 de patente (studiu inițiat pe 4.000.000 de patente). Analiza s-a concretizat printr-o clasificare ierarhică pe nivele inovative și selecția celor mai bune (~ 21 %). În continuare bazei de date formate i-au fost aduse completări și cu alte principii.

Structura metodologiei TRIZ este prezentată în figura 6.9 [6.46].

Premiza proiectării TRIZ este proiectarea ideală. Măsura idealității, care stabilește și soluția problemei, este indicele de ideal definit prin raportul:

$$ideal = \frac{\sum \text{beneficii}}{\sum \text{cost} + \sum \text{efecte_nocive}} \quad (6.14)$$

Termenul $\sum \text{beneficii}$ cuantifică funcțiile utile ale sistemului în timp ce $\sum \text{cost}$ se referă la costurile directe și cele pentru societate. Componenta $\sum \text{efecte_nocive}$ ia în considerare moduri de defectare, funcții nocive și alte aspecte nedorite din ieșirile sistemului proiectat.

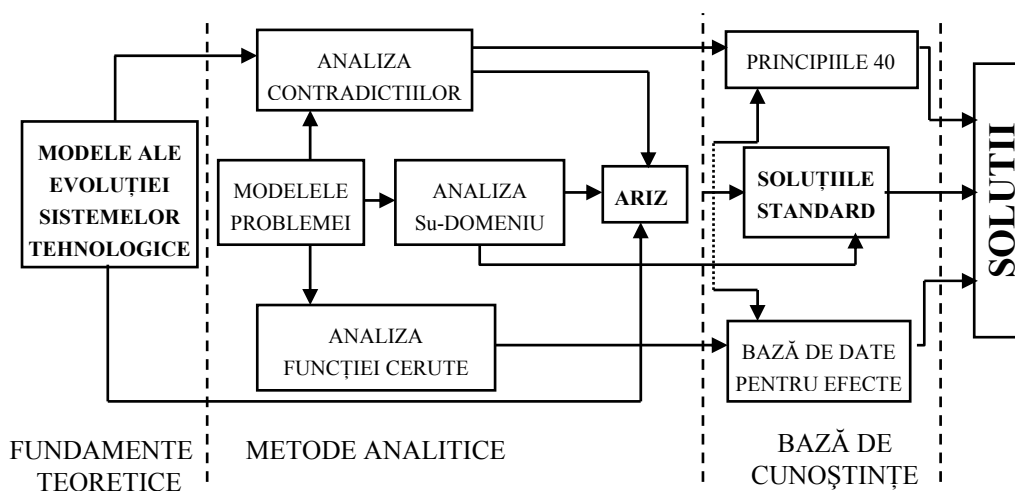


Fig. 6.9 Structura metodologiei TRIZ

6.2.2. Comparație TRIZ - proiectarea axiomatică – metoda Taguchi

Metoda de proiectare robustă a lui Taguchi se identifică cu o metodologie sistematică pentru proiectarea statistică experimentală în vederea îmbunătățirii procesului de proiectare și fabricație a produselor.

Rezolvarea unei probleme de proiectare prin metoda Taguchi presupune trei stadii :

- Proiectarea conceptuală a sistemului. Acest stadiu nu este un punct central al metodei.
- Proiectarea parametrică care presupune un proces de optimizare și oportunități de reducere a costurilor produsului și a fabricației ;
- Proiectarea toleranțelor, considerat ca un stadiu independent în condițiile unor mici variații ale parametrilor de proiectare.

În baza celor prezentate se poate stabili o comparație critică între metodele de proiectare analizate (tabelul 6.4)[6.21].

O comparație pe baza axiomelor de proiectare este sugerată în tabelul 6.5.

Tabelul 6.4

	FUNCȚIA CENTRALĂ	APLICABILĂ CEL MAI BINE	IDEEA PROCESULUI DE PROIECTARE	ACCENTUL METODEI
PROIECTARE AXIOMATICĂ	Răspuns dorit	..pe structura sistemului & fundamente în proiectare conceptuală	- Gândire pozitivă - Cum se poate realiza un proiect perfect și imun la « zgomot » ?	Aplicație dinspre cerințele funcționale spre cerințele de proiectare
TRIZ	Funcția de bază	..pe structura sistemului & fundamente în proiectare conceptuală	- Gândire negativă - Punct de start conflicte și contradicții - Cum se poate rezolva o contradicție	- Contradicțiile - Start cu parametrii de proiectare și apoi cerințe funcționale
PROIECTARE ROBUSTĂ	Transformări energetice	- Optimizare tehnologie -Optimizare proiectare conceptuală	Cum un sistem ingineresc poate fi optimizat astfel ca efectele secundare din condiții necontrolate să fie reduse	- Identificarea funcției ideale - Start cu răspunsul sistemului în vederea maximizării funcției uzuale

Tabelul 6.5

	AXIOMA INDEPENDENȚEI	AXIOMA INFORMAȚIEI
PROIECTARE AXIOMATICĂ	Independența cerințelor funcționale	Minimizarea informației
TRIZ	Se menține independența parametrilor prin eliminarea contradicțiilor tehnice sau fizice	Conceptul de idealitate
PROIECTARE ROBUSTĂ	Identificarea funcției ideale; Se controlează factorii care promovează efecte pozitive	Maximizarea raportului semnal / zgomot

6.3. Proiectare CAD inteligentă

6.3.1. Introducere

Facilitățile CAD se constituie într-un instrument util la îndemâna proiectantului pentru generarea de produse competitive.

Un pas important în dezvoltarea sistemelor tehnice este pasul de la procesarea geometrică simplă la manipularea proprietăților funcționale. Aceste eforturi au permis accelerarea procesului de proiectare, eliminarea erorilor din procesele inginerești și utilizarea datelor în format electronic.

Obținerea unor produse competitive care să corespundă cerințelor utilizatorului / beneficiarului impune existența posibilității unei simulări a funcționării pe parcursul procesului de proiectare.

Forma, funcțiile și funcționarea unui produs interacționează și se condiționează reciproc. O realizare a proiectării asistată de calculator impune condiția obținerii informației privind funcționarea din modelul CAD. Relația dintre cele trei atribute este prezentată în figura 6.10 [6.39].

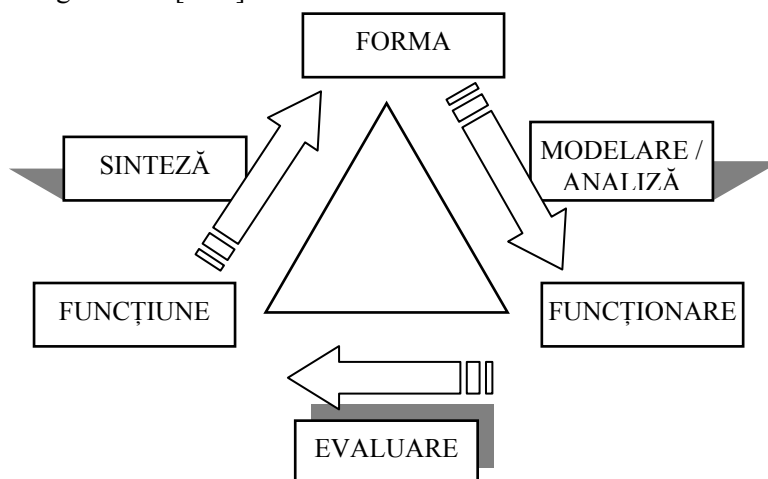


Fig. 6.10 Relația formă – funcțiune - funcționare

Modelul produsului CAD inteligent apelează la tehnologiile bazate pe caracteristici orientate – obiect și sisteme expert.

6.3.2. Considerații privind programarea orientată obiect

Faza de proiectare se situează între *analiză* și *implementare*. Toate aceste faze fac parte din procesul de *dezvoltare orientat obiect*.

Abordarea orientată spre obiecte se consideră ca fiind inițiată încă din anul 1957, o dată cu proiectul american pentru racheta balistică Minuteman, dar cu o extindere apreciată în ultima perioadă.

Abordarea orientată spre obiecte este specifică *lucrului în echipă*. Prin acest concept se asigură o comunicare eficientă între membrii diverselor colective care lucrează în cadrul unui proiect. Acest stil de programare permite împărțirea aplicațiilor în mai multe module, astfel încât cel ce dezvoltă un modul nu trebuie să cunoască detaliile de implementare a altor module.

Se poate considera că programarea orientată pe obiecte este un concept natural: zi de zi avem de-a face în activitatea obișnuită cu *obiecte conectate* între ele, *comunicând* unele cu altele într-un anumit mod. În plus în natură o entitate este caracterizată atât prin structura sa cât și printr-un anumit comportament. În natură obiectele evoluează în timp, adeseori modificându-și structura și funcționalitatea.

Proiectarea orientată obiect este o strategie în care sistemul se gândește în termeni de “obiecte”, în loc de operații și funcții. Structurile de date se grupează cu operațiile care prelucrează datele respective. Un astfel de ansamblu poartă denumirea de *obiect* sau clasă. Proiectarea de programe utilizând clase se numește programare orientată pe obiecte (OO). Programul nu este conceput ca o mulțime de funcții care schimbă date prin parametri comună și memorie comună ci ca și o mulțime de obiecte care interacționează, oferă servicii altor obiecte și își gestionează starea internă [6.28].

Conceptele fundamentale avute în vedere de acest principiu sunt [6.28]:

- **Încapsularea**

În mod frecvent pentru structurile de date se utilizează denumirea de *date membre* iar pentru procedurile ce prelucrează aceste date, termenul de *funcții membre* sau *metode*. Ansamblul definește *obiectul* sau *clasa*:

$$date + metode = obiect$$

Categoria de “clasă” se asociază unui obiect real sau virtual înglobând o serie de subcategorii:

- Structură informațională care cuprinde un set de atribute asociat clasei;
- Funcții de acces autorizat la această structură informațională care permit operații de manipulare a obiectului respectiv;
- Funcții suplimentare (ascunse utilizatorului clasic) care definesc comportamentul obiectului în mediul său de evoluție

Asemănător limbajului *C*, pentru a putea utiliza efectiv un tip de date – în cazul de față o clasă - trebuie să definim o variabilă de acel tip:

NumeClasă variabilă

Prin *încapsulare* clasele sunt protejate față de “tentativele altor clase” de a regăsi sau modifica valorile propriilor lor atribute. O sintaxă simplificată a declarației

unei clase este următoarea:

```
class NumeClasă
{
.....
declarații variabile membre
.....
declarații funcții membre
..
}
```

- **Moștenirea**

Continuăm analogia dintre activitatea din natură și (OO) printr-un exemplu referitor la clasificarea animalelor în regn, clasă, ordin, familie, gen și specie. Acest mod de clasificare definește o relație de forma “*este un / este o*”. Conceptul de *cal* primește în mod automat un răspuns de forma: *este un ierbivor*. Un ierbivor *este un mamifer* ș.a.m.d. Conceptul de *cal* înglobează – *moștenește* - în acest fel toate caracteristicile unui ierbivor și cele a unui mamifer. Dar acest concept are în vedere și faptul că un cal nechează, dă din coadă ș.a.m.d. Putem clasifica în mod suplimentar caii în cai de tracțiune, cai de curse iar caii de curse în alte subcategorii etc.

Conceptul de *moștenire* permite construirea de *clase derivate* pentru care se păstrează unele structuri informaționale și funcții de acces la acestea, preluate din una sau mai multe clase de referință, originală care poartă denumirea de clasă de bază. Se constituie astfel o ierarhizare a obiectelor bazată pe criterii logice și funcționale.

Dintr-o clasă de bază pot fi derivate mai multe clase și fiecare clasă derivată poate servi mai departe ca bază pentru alte clase derivate. În acest mod se poate realiza o ierarhie de clase care să modeleze adecvat sisteme complexe. Pornind de la clase simple și generale, fiecare nivel al ierarhiei acumulează caracteristicile claselor “părinte” și le adaugă un anumit grad de originalitate. În plus, modul de lucru în C++ permite ca o clasă să moștenească simultan proprietățile mai multor clase, procedura numindu-se moștenire multiplă [6.1].

- **Polimorfismul**

Polimorfismul reprezintă atribuirea aceluiași nume unei funcții de tratare a unor structuri informaționale, implementarea realizându-se în mod dinamic în momentul rulării programului, în funcție de recunoașterea automată a tipului clasei obiectului referit.

6.3.3. Programare OO în domeniul electromecanic

Proiectarea orientată – obiect (OO) a contribuit esențial la evoluția proiectării asistate. Elementul proiectat se află într-o reprezentare încapsulată prin includerea proprietăților și comportamentului.

Proprietățile sunt modelate prin parametrii (atributul nodului) ce descriu geometria și proprietățile fizice, forțe generalizate, poziții, viteze etc. Sub forma fizică cantitativă acestea se exprimă prin utilizarea terminologiei de *date*. Se pot distinge date care descriu proprietăți permanente (asemănări geometrice, masă, coeficient de frecare etc.) și date care descriu o stare temporară a obiectului (forță, poziție, viteză). Prima categorie poate fi modificată doar prin metode de dimensionare relativ la proces în

timp ce a doua categorie corespunde unor metode ce descriu o funcționare dinamică.

Funcționarea *obiectului* este pusă în vedere prin interdependențele acestor parametri (de ex.: masă, accelerație și forță). Instrucțiunile care descriu combinațiile sunt *metodele* sau *constrângeri*. Într-o reprezentare schematică obiectul are componența din figura 6.11.

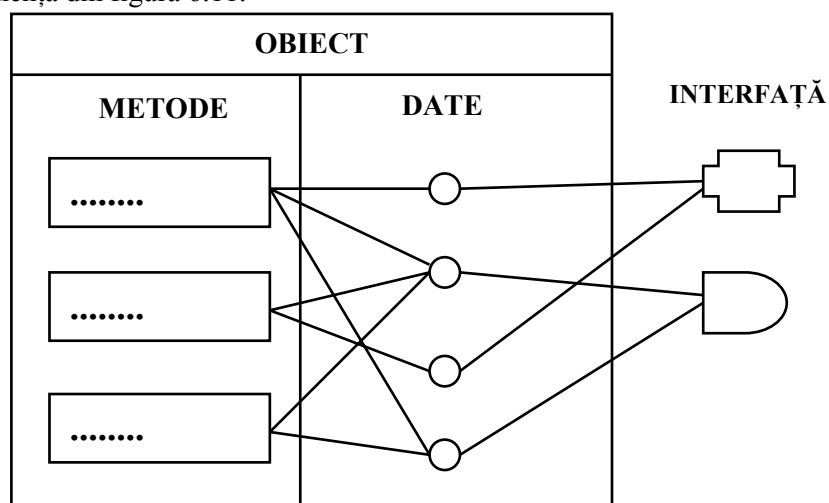


Fig. 6.11 Proiectarea orientată - obiect

Componenta – *obiectul* - care se identifică cu problema de rezolvat poate fi individualizată în mediu de lucru sau într-o conexiune mecanică, electrică sau termică cu o altă componentă. O descriere completă a *obiectului* impune descrierea interacțiunii acestuia. Această descriere este realizată de *interfața obiect*. Aceasta definește un set de parametri care trebuie să se găsească într-o anumită relație funcțională. Trebuie făcută o distincție între două tipuri de interfețe:

- Interfață care descrie o interacțiune fizică energetică, masă, informație a obiectului cu mediu;
- Interfață internă – *implementare* – care pune în vedere funcționarea internă.

Elementele unui sistem – obiectele prin prisma OO - interacționează unul cu altul în procesul de funcționare (figura 6.12). Această interacțiune se poate descrie pe baza noțiunii de *port* sau *conector* și care intră în componența interfeței obiect (reprezentat printr-un ● în figura 6.12).

Portul se identifică cu punctele sau conturul prin care componentele interacționează între ele prin schimb de energie și în mod particular cu mediul exterior. În mod grafic interacțiunea se reprezintă prin linii trasate între porturi (fig.6.12) Un port trebuie să conțină toate informațiile necesare descrierii interacțiunii.

Energia vehiculată printr-un port este descrisă de două variabile:

- o variabilă peste / transversală (*across*);
- o variabilă de transfer / trecere (*through*).

O conexiune între două porturi este descrisă prin constrângeri algebrice conform sistemului de ecuații (**) din figura 6.12 (va_{acr_A} și va_{acr_B} sunt variabilele *across*, va_{tr} sunt variabilele *through*).

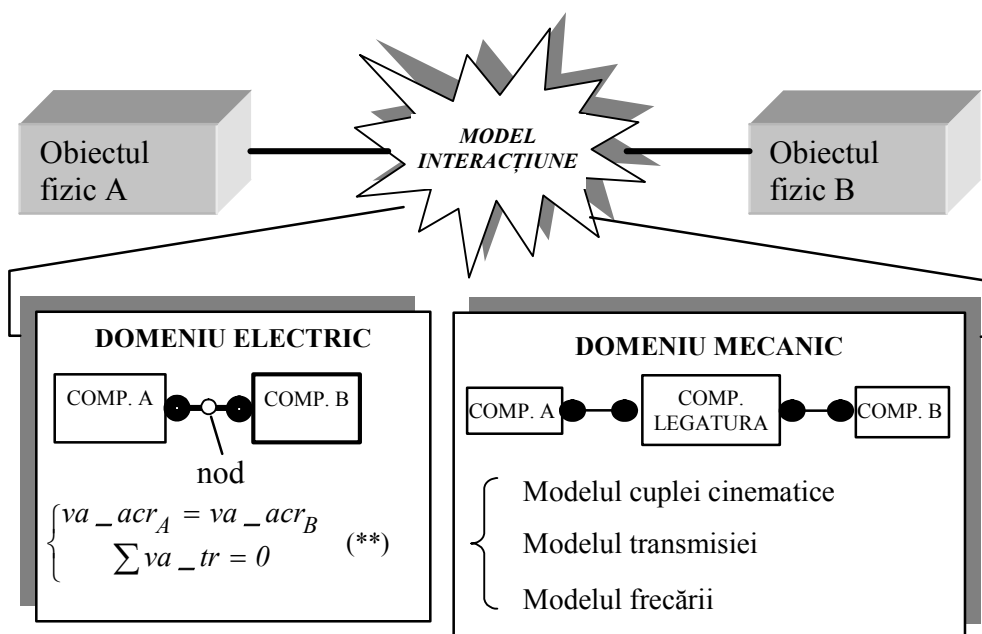


Fig. 6.12 Interacțiunea obiectelor din sistem

Ca o exemplificare a celor arătate anterior se prezintă în figura 6.13 o componentă obiect cu evidențierea interfeței de configurare, modelul CAD, modelul de funcționare și relațiile dintre acestea.

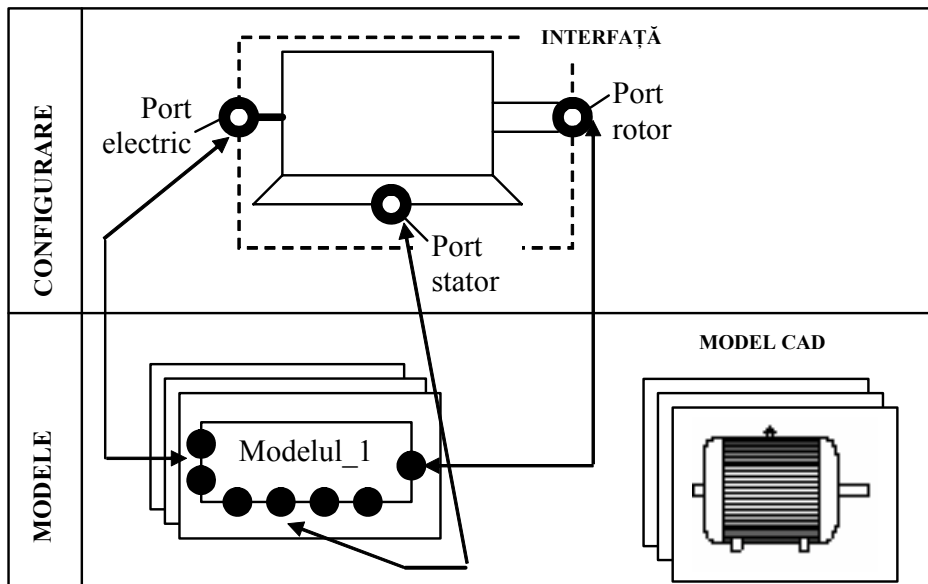


Fig. 6.13 Porturile unui motor electric

Interfața de configurare a componentei constă în porturi și parametri / date. Parametrii permit modelarea componentei. Portul (porturile) definesc interacțiunea cu mediul. Motorul electric din figura 6.13 prezintă trei porturi materializate prin:

- Portul 1 – reprezintă modul de fixare a statorului față de mediu și se concretizează printr-un număr de orificii pentru fixare;
- Portul 2 – reprezintă calea prin care motorul se conectează în sistem vehiculând energie mecanică și se concretizează prin arborele rotorului;
- Portul 3 – reprezintă calea prin care motorul primește energie electrică dinspre exterior și este concretizat prin conectorul electric.

Modelul funcțional este concretizat din nou pe baza porturilor interfață. Este întâlnit modelul de tip interfață (funcționarea porturilor și parametrii modelului) și componenta de implementare. Implementarea este realizată sub forma seturilor de ecuații referitoare la funcționare.

Modelul CAD are două scopuri majore:

- O reprezentare cu specificații referitoare la forma componentei (dimensiuni nominale, toleranțe, materiale);
- O reprezentare matematică a geometriei obiectului (de ex.: vizualizare componente).

Considerentele anterioare sunt direct aplicabile în domeniul electric. Cele două variabile avute în vedere sunt tensiunea și respectiv curentul. În spiritul metodologiei OO cele două variabile se declară sub forma (în mediul Modelica) [6.15]:

```

class voltage = Real
class current = Real
    
```

(6.15)

unde **Real** este numele tipului de variabilă predefinită. Variabila reală conține un set inițial de atribute referitoare la unitatea de măsură, valoarea inițială, valoarea minimă și maximă. De exemplu o tensiune alternativă este definită în modul:

```

class Voltage = Real(unit ="V", min = -220.0, max = 220.0);
    
```

(6.16)

Clasa port / conector este definită în modul:

```

connector Pin
Voltage v;
Flow Current i;
end Pin
    
```

Declararea conectorului este utilizată așa cum arătam la exemplificarea modului de conectare a claselor.

O proprietate comună multor componente electrice este definirea lor în raport cu doi conectori. Acest considerent a permis definirea unei clase model corespunzătoare *Doi_pini* (TwoPin) (fig.6.14):

```

class TwoPin
Pin p, n;
Voltage v;
Current i;
Equation
v = p.v - n.v;
0 = p.i + n.i;
i = p.i;
end TwoPin;
    
```

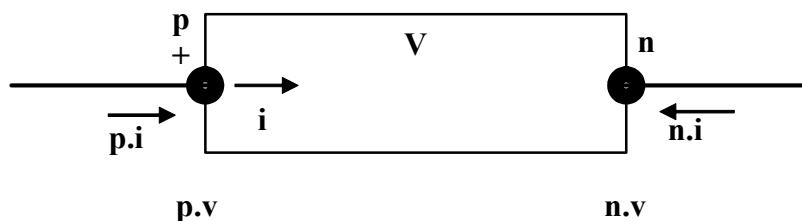


Fig. 6.14 Clasa model doi_pini

În același mod se poate defini acum subclasa pentru rezistor, capacitate etc. . Această clasa beneficiază de toate proprietățile clasei TwoPin.

```
class Rezistor "rezistorul electric ideal"
    extends TwoPin
    parameter Real R (unit = "Ohm");
```

```
Equation
    R * i = v;
```

```
end Resistor;
```

Beneficiind de o bibliotecă de modele pentru diverse clase, există posibilitatea realizării modelului pentru circuitul electric și realizarea unor operații de simulare corespunzătoare.

În domeniul mecanic interacțiunea presupune includerea unui număr sporit de modele care se pot însă descrie pe principiile modelării fizice.

O componentă importantă a sistemelor tehnice este cea concretizată prin mecanisme. Utilizând metodologia OO se poate defini modelul elementului, a cuplei cinematice etc.

Conform principiilor mecanicii clasice un element rigid este descris printr-un punct corespunzător centrului de masă și tensorul de inerție. Situația acestuia – poziția și orientarea – se exprimă relativ la un sistem de referință global. Modelul de funcționare constă dintr-un set de ecuații referitoare la variabilele portului $\{p, \Phi, F, M\}$ unde:

- perechea de variabile *across*:

$$\left\{ \begin{array}{l} \text{variabila } p \text{ (un vector) definește poziția centrului de masă;} \\ \text{variabila } \Phi \text{ definește orientarea corpului în raport cu sistemul de axe global.} \end{array} \right.$$

- perechea de variabile *through*:

$$\left\{ \begin{array}{l} \text{variabila } F \text{ definește forța aplicată asupra elementului;} \\ \text{variabila } M \text{ definește momentul aplicat asupra elementului.} \end{array} \right.$$

Reprezentarea componentelor pentru elementul rigid în concordanță cu metodologia OO este dată în figura 6.15.

În corespondență cu teoria mecanismelor și utilizând modul de definire a elementului rigid se poate reprezenta modelul cuplei cinematice (fig.6.16).

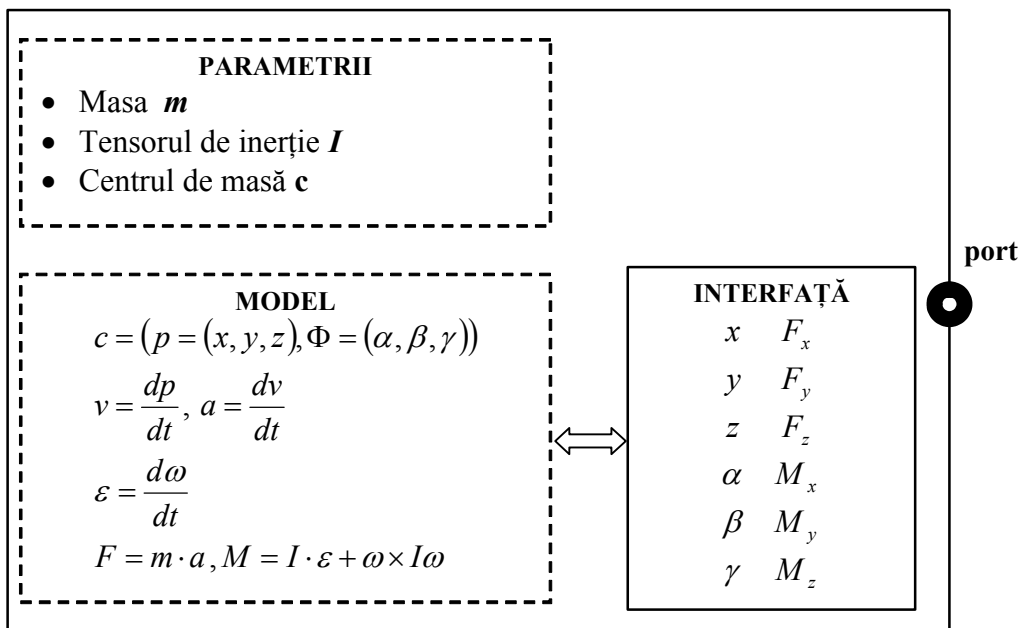


Fig. 6.15 Elementul rigid și OO

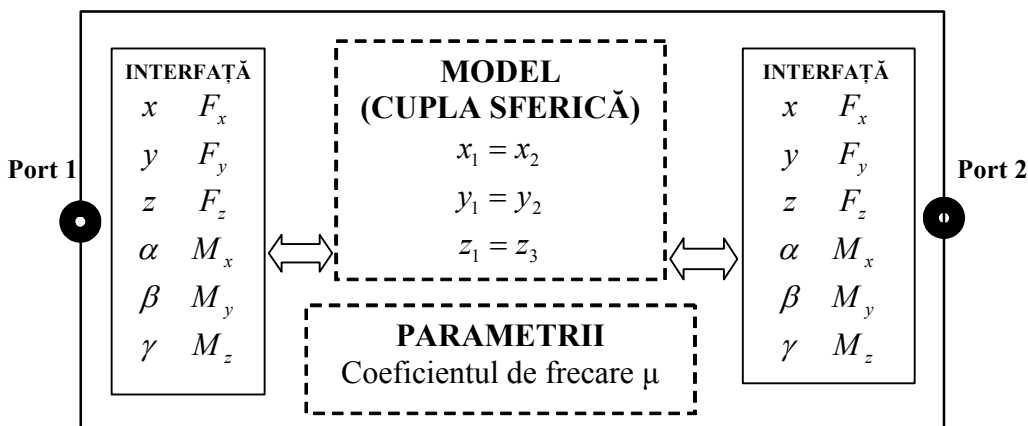


Fig. 6.16 Modelul cuplei cinematice

Pe un principiu asemănător celor expuse anterior, pentru un conector / port de natură mecanică echivalent unei piese de conexiune, există posibilitatea creării clasei corespunzătoare:

```

connector Flange
    Angle    phi;
    Flow     Torque;
end Flange
    
```

În final, modelele create se includ în biblioteca de lucru al software-lui creat. Un astfel de exemplu îl reprezintă Modelica unde domeniul mecanic este structurat în mod ierarhic:

- **modelica_mecanică_mișcarea de rotație** – cu o serie de subclase: componentă rotațională cu inerție, transmisie ideală R-R, transmisie ideală planetară, element elastic liniar, element amortizor liniar, ...etc.
- **modelica_mecanică_mișcarea de translație** – cu o serie de subclase specifice: componentă inerțială în translație, sensor de forță, de poziție, ...etc.

În figura 6.17 se prezintă posibilitatea de a construi modelul unei transmisii reale cu pierdere energetică – subclasa din domeniul ...*mecanic / rotațional* - pe baza a două componente:

- prima clasă corespunde unei transmisii cu pierderi energetice (include randamentul transmisiei și frecarea în lagăre);
- a doua clasă corespunde aspectului real al transmisiei și include elasticitatea, amortizarea și jocul din angrenare

Acest mod de abordare a interacțiunii se cuplează cu sistemul CAD. Se beneficiază de informațiile referitoare la parametrii elementelor și a cuplelor cinematice.

Din obiectele proiectate, prin cuplarea într-o formă nouă și complexă se obține obiectul compus în baza ecuațiilor de sistem sau constrângeri de rețea.

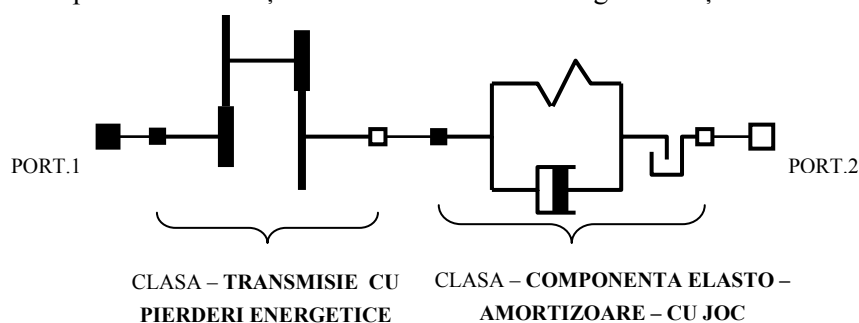


Fig. 6.17 Modelul unei transmisii reale

În mod ierarhic următorul nivel este cel al clasei. Prin achiziționarea seturilor de parametri și metode se definesc module de constrângere. Caracteristicile (features) pot fi organizate prin astfel de module.

Literatura de specialitate face referiri la o serie de medii integrate care există astăzi la dispoziția inginerului proiectant:

- Visual Nastran 4D – destinat analizei mișcării și prin element finit;
- ADAMS (Automatic Dynamic Analysis of Mechanical Systems);
- Dynamic Designer / Motion and Simply Motion – rulează integral cu Mechanical Desktop;
- DADS (Dynamic Analysis and Design System);
- CATIA;
- Pro-engineer , ...s.a.

În figura 6.18 se prezintă o structură principală a unui mediu de lucru pentru proiectare, simulare și vizualizare și care combină metodologiile prezentate anterior.

În figura 6.19 se prezintă o dezvoltare a principiului de lucru pentru un mediu integrat, dezvoltat pe structura mediului de lucru Mechanical Desktop.

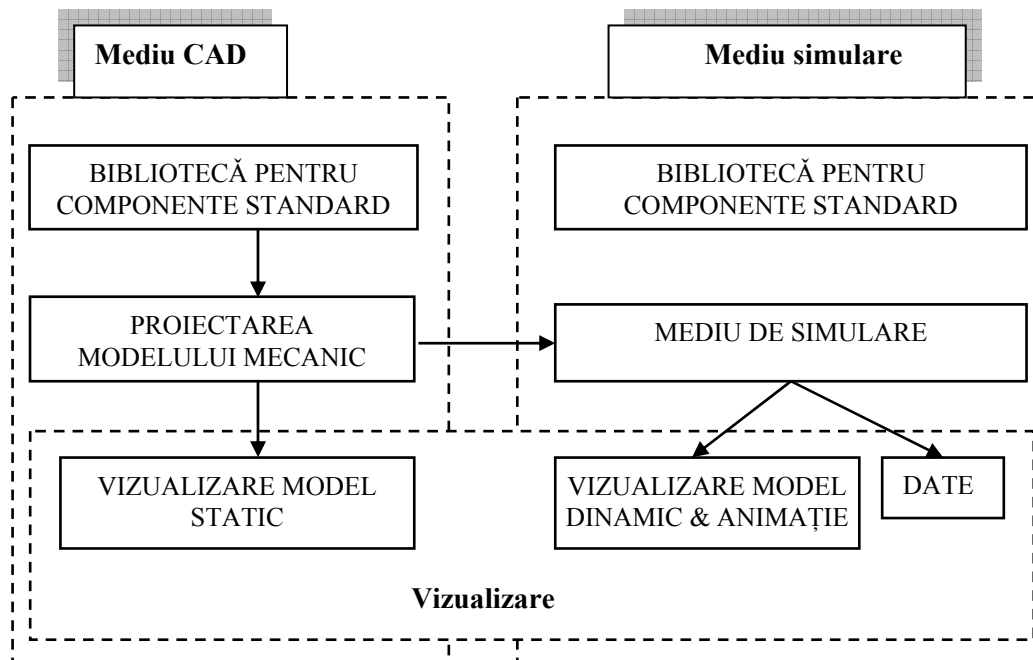


Fig. 6.18 Structura principală a unui mediu de lucru pentru proiectare

Punctul de start este oferit de realizarea modelului mecanic în mediul de lucru Mechanical Desktop (AutoDesk). Modelul salvat în format *.dwg conține toate informațiile referitoare la componente, informații referitoare la montajul componentelor și constrângeri.

Geometria fiecărei componente este exportată în format STL. În același timp informațiile referitoare la masa elementelor și tensorul inerțial sunt obținute din ansamblul mecanic. Translatorul ARX (atașat mediului Mechanical Desktop) utilizează aceste informații pentru a genera setul corespunzător de clase Modelica cu conexiunile corespunzătoare dintre ele (pe principiul programare orientată – obiect). Modelul generat în mod automat este procesat într-un mediu de simulare adecvat (Dymola, MathModelica) [6.7].

Concepția modulară în proiectarea obiect este direcționată în proiectarea organologică. O serie de noi aspecte pentru un produs pot fi schimbate și modificate în mod simplu.

Interfața obiect permite inginerului alăturarea unor componente prin accesarea bazelor de date. Proiectarea unui produs nou pornind de la o structură funcțională Black Boxes este mult simplificată.

În mediul de lucru AutoCAD noțiunea de obiect este utilizată în mod nativ chiar dacă inițial nu în sensul “clasic”. Prin introducerea interfeței programate de tip VBA (Visual Basic Applications) specifică mediului Microsoft Windows, abordarea obiect a căpătat o consistență mult mai mare [6.22]. Au fost definite în mod precis obiecte specifice: ‘Application’, ‘Preferences’, ‘Document’, ‘Blocks’, ..., ‘Linetypes’, ‘Linetype’, ‘Circle’, ‘Ellipse’, ‘Hatch’, ..

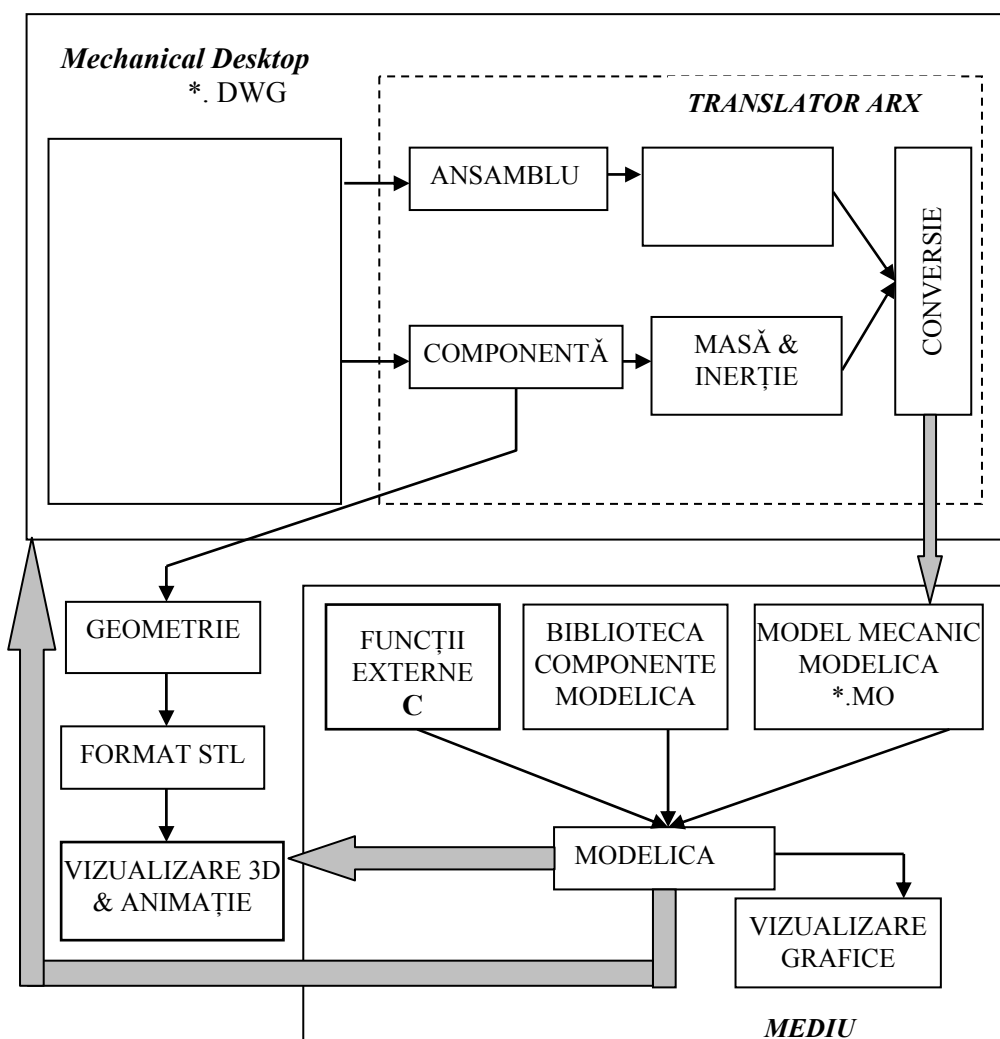


Fig. 6.19 Principiu de lucru într-un mediu integrat

6.3.4. Proiectarea pe bază de caracteristici și sistem expert

Modelarea produselor, în scopul reprezentării lor și procesării cu ajutorul calculatorului, impune construirea unei structuri logice, flexibile bazată pe reprezentarea cunoștințelor: informații cu semnificații atașate incluzând generalizări, reguli, metode, proceduri. Se înțelege prin informație un set de evenimente și relațiile dintre ele: de ex. profilul unei palete descris printr-o listă de coordonate. Datele referitoare la produs includ toate informațiile referitoare la proiectare, fabricație, analiza calității, testare. Se pune accent în mod special pe cele referitoare la modelul geometric [6.6].

Modelarea geometrică a produsului are ca scop primordial reprezentarea geometrică într-un format accesibil sistemului de calcul. O istorie parțială a modelării

geometrice ar include:

- 1963 – modelul de tip rețea sârmă (wireframe) (Ivan Sutherland, MIT);
- 1967 – modele bazate pe suprafață (surface modeling) (S.A.Coons);
- 1973 – modelarea prin solide:
 - construcția geometrică (Laning s.a., Draper Lab.);
 - reprezentarea frontierelor (Ian Braid, Cambridge Univ.);
- 1985 modelarea prin caracteristici (Pratt & Wilson);
- 1990 modelarea parametrică.

Modelul geometric prezintă o serie de deficiențe [6.16], [6.31]:

- multe dintre activitățile ingineresti sunt legate de entități “macro” (de ex.: orificii, cavități, etc.) și mai puțin de informații “micro” (de ex.: muchii, suprafețe, curbe etc.);
- reprezentarea multor elemente de bază pentru descrierea produsului nu sunt adecvate cu modelul geometric;
- reprezentarea componentelor exclusiv prin primitive geometrice nu reflectă existența unor aspecte în termenii de abstractizare necesari înțelegerii proiectului.

În contextul creării unui algoritm corespunzător, evoluția metodologiilor de modelare a condus la un nivel înalt de abstractizare – caracteristica (*feature*).

În corespondență cu Shah (1988) o caracteristică este definită ca un set de informații grupate adecvat pentru a reprezenta un produs [6.37]. În [6.6] caracteristica reprezintă o semnificație inginerescă sau o semnificație a geometriei unei componente sau produs. În [6.11] caracteristica se definește ca un element distinctiv al unei componente, definind o formă geometrică, care este specifică pentru un anumit proces de prelucrare sau poate fi utilizată pentru asamblare sau raționamente asupra formei.

Reprezentarea componentelor prin caracteristici – metodologie OO a modulelor de constrângere – a fost analizată în literatura de specialitate consemnându-se printre altele:

- accelerează procesul de proiectare;
- garantează posibilitatea de a procesa, prelucra prin fabricație caracteristici individuale pe componente fizice individuale;
- sunt elemente potențiale de integrare între proiectare și fabricație.
- necesitatea rezolvării unor aspecte legate de:
 - posibilitatea de a crea și utiliza caracteristici predefinite;
 - dificultățile generate de lipsa unui set suficient de caracteristici referitoare la un domeniu.

Un număr sporit de sisteme CAD de tip comercial: CATIA, Pro-Engineer, AUTODESK / Inventor etc. utilizează aceste facilități;

Clasificări ale caracteristicilor, după diverse criterii, sunt prezentate în [6.6], [6.37], [6.11] iar o schemă a taxonomiei caracteristicilor este ilustrată în [6.8].

Așa cum rezultă din cele prezentate, metodologia caracteristicilor este o extensie a tehnicilor de modelare solidă. Din acest motiv se impune precizarea unor aspecte esențiale pentru modelarea solidelor. În baza modului de reprezentare această tehnologie poate fi divizată în trei clase [6.37]:

- descompunerea modelelor: enumerare exhaustivă, descompunere celulară,

- subdiviziunea spațiului;
- modele constructive, construirea bazându-se pe o combinație de primitive și operații Booleene.
 - Modelul semi-spațial. În figura 6.20 este prezentat modul de descriere a unui obiect cilindric.

$$H_1 : x^2 + y^2 - r^2 < 0 \quad (6.17)$$

$$H_2 : z > 0 \quad (6.18)$$

$$H_3 : z - h < 0 \quad (6.19)$$

$$C = H_1 \cap H_2 \cap H_3 \quad (6.20)$$

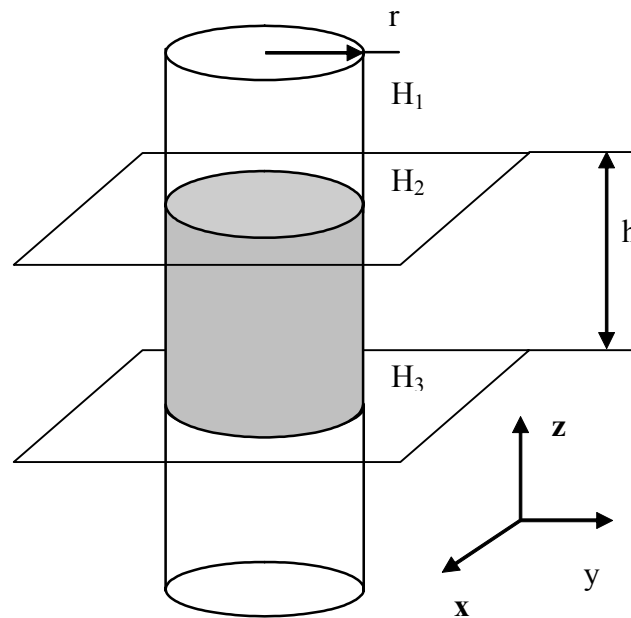


Fig. 6.20 Descrierea unui obiect cilindric

- Modelul CSG (Constructive Solid Geometric) operează numai cu primitive solide și operații booleene. Fiecare primitivă este definită prin combinații bazate pe modelul semi-spațial. Modul de configurare a unui obiect prin metodologia CGS este prezentat în figura 6.21
- Modelul frontieră reprezintă un obiect solid prin divizarea suprafeței acestuia într-o sumă de fețe (suprafețe frontale) dispuse într-o formă convenabilă. Metodologia este performantă dacă divizarea se poate reprezenta compact matematic. Construcția unui obiect prin această metodă este prezentată în figura 6.22
- Modelul parametric și variațional conform algoritmului: definirea topologiei, definirea constrângerilor, evaluarea modelului și crearea de variante.

Într-o evoluție logică modul de construcție a modelelor a devenit cel bazat pe caracteristici. Structura unui sistem de proiectare bazat pe caracteristici este prezentată în figura 6.23 [6.16] iar lucrul în Mechanical Desktop pentru un caz concret în figura 6.24.

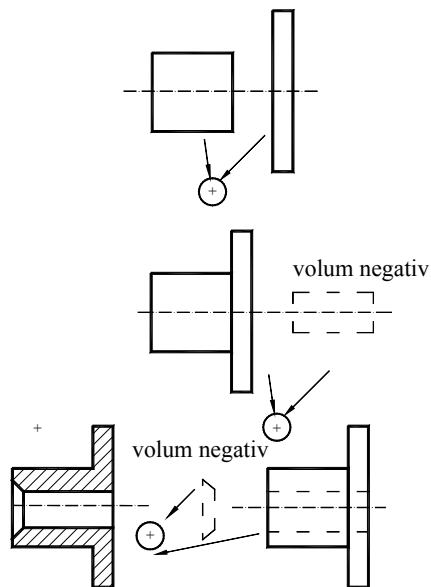


Fig. 6.21 Descrierea unui obiect prin metodologia CGS

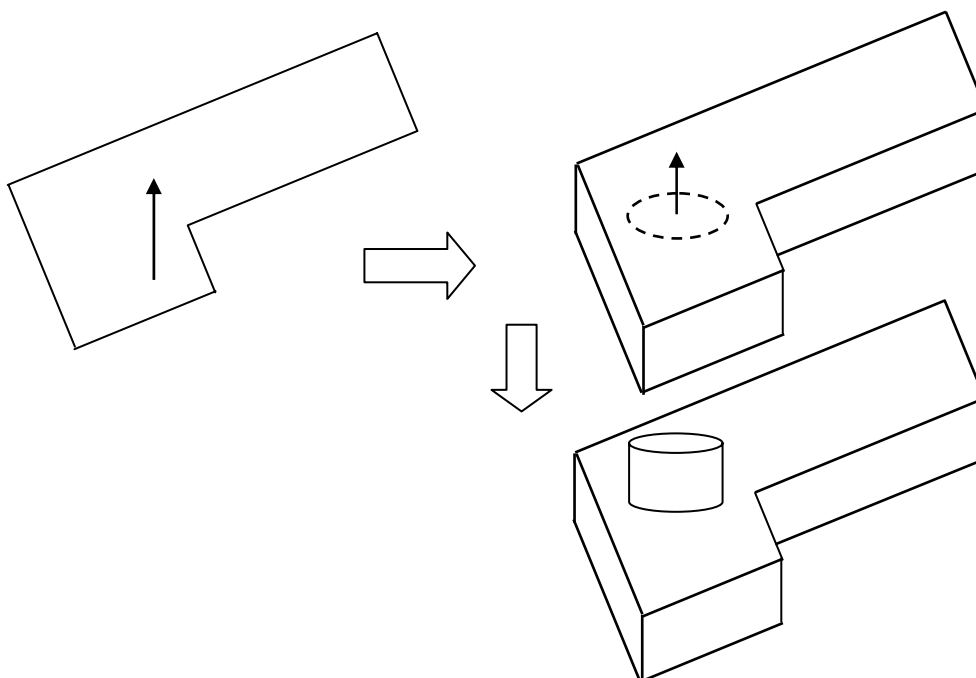


Fig. 6.22 Modelul frontieră în construcția unui obiect

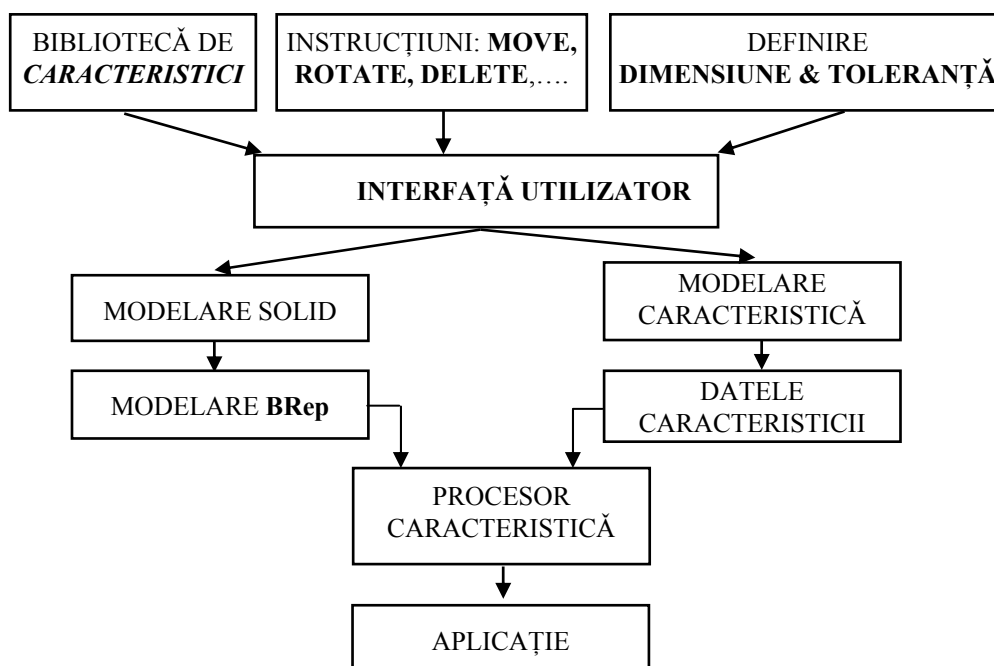


Fig. 6.23 Structura unui sistem de proiectare bazat pe caracteristici

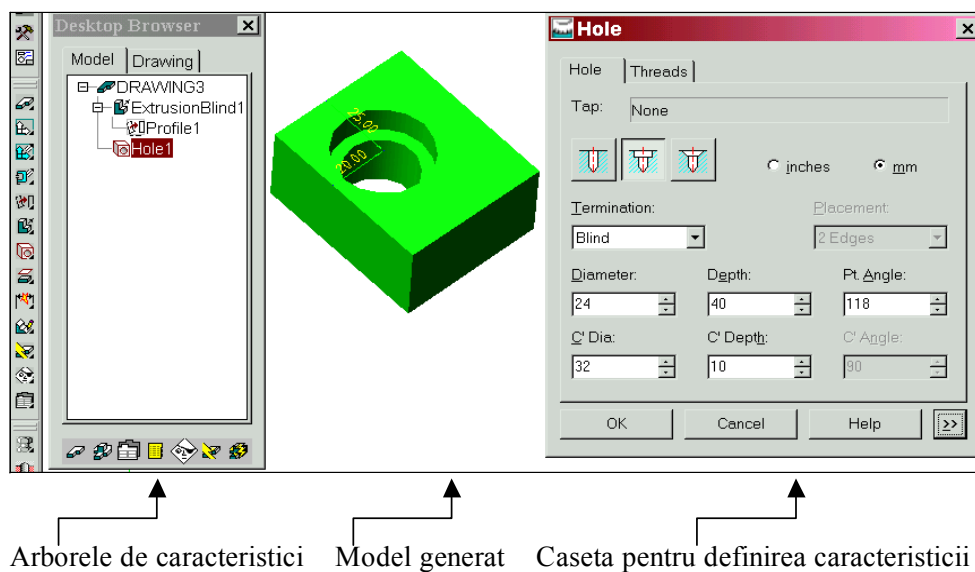


Fig. 6.24 Concretizarea modului de lucru în Mechanical Desktop

Structura unui sistem CAD inteligent structurat pe principiile enunțate este prezentată în figura 6.25[6.31].

Așa cum rezultă din algoritmul prezentat, programul CAD dispune de o interfață grafică cu facilități de accesare a bazelor de date necesare în cadrul sistemului

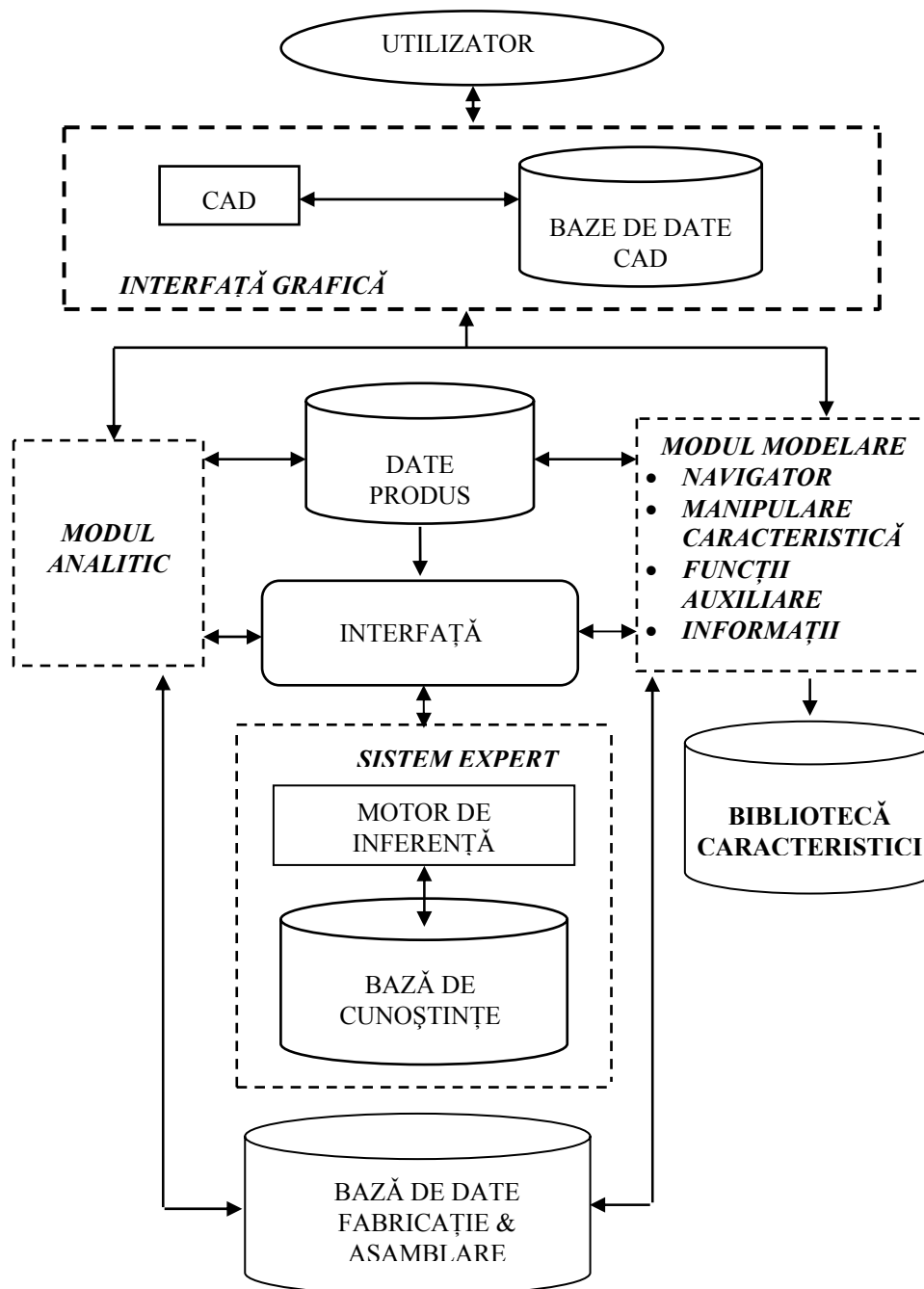


Fig. 6.25 Structura unui sistem CAD inteligent

Modulul pentru modelare are ca scop definit:

- Asigurarea funcției de “navigare” în structura de date ale produsului;
- Apelarea și utilizarea caracteristicilor;

- Construcții grafice prin funcțiile auxiliare disponibile în mediul CAD;
- Funcția de distribuție / utilizare a informațiilor necesare pentru înscrierea toleranțelor, prelucrarea suprafeței etc.

Modulul analitic permite analizarea procesului de asamblare, dimensionare, toleranțe, identificarea suprafețelor de prelucrat etc.

Informația despre produs este prezentă în două formate:

- *Formatul intern* care corespunde informației stocate în structura de date ale produsului;
- *Format grafic* care reprezintă informația despre produs necesară vizualizării.

Vizualizarea produsului este realizată prin utilizarea interfeței grafice. Schema de ordonare ierarhică a produsului pe layere este prezentată în figura 6.26. Dacă ansamblul, subansamblurile, componentele au fost proiectate graphic atunci corespondentul *dimensiuni* (*cote*, *toleranțe*, *note informative*) este creat în mod automat.

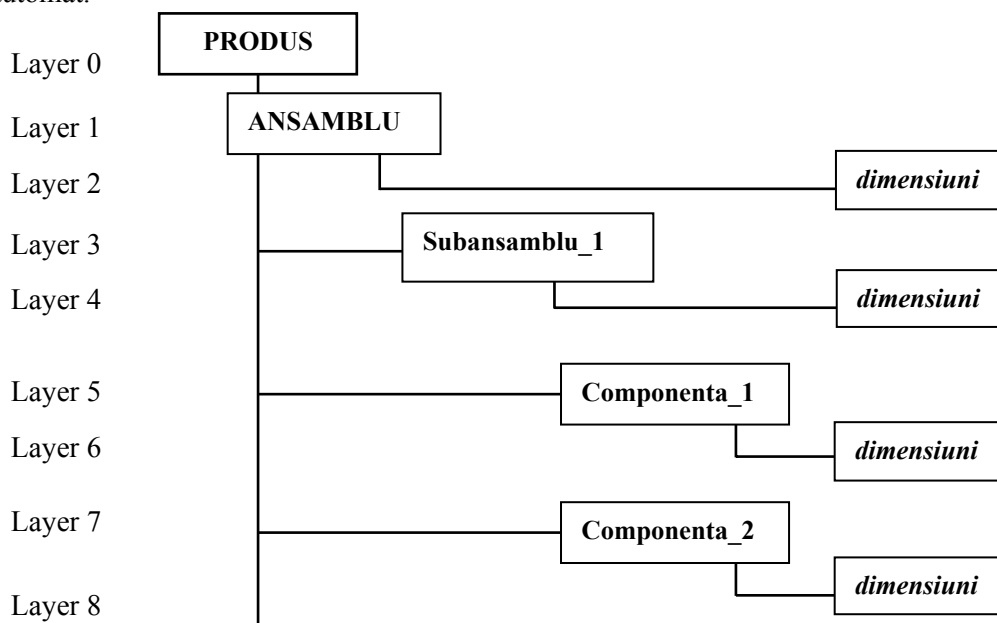


Fig. 6.26 Schema de ordonare ierarhică a produsului

În acord cu metodologia orientată - obiect, clasele vor fi definite pe baza funcției îndeplinite în contextul proiectului. Este momentul în care intervine din punct de vedere O_O noțiunea de caracteristică. Atributele caracteristicilor trebuie să se refere la:

- *Atributele configurației* - corespund unei descrieri geometrice a caracteristicii (de ex.: diametrul unui tronson);
- *Atributele geometrice ale informației* - care corespund pentru informația referitoare la situarea caracteristicii în spațiu;
- *Atributele toleranței și a prelucrării suprafeței* - utilizabile pentru o proiectare funcțională performantă;

- *Atribute manageriale* - care asigură performanțele sistemului referitoare la comunicarea informațiilor dintre diversele module.

Un aspect referitor la definirea unei caracteristici *tronson_cilindric* este prezentat în figura 6.27. Poziționarea caracteristicii s-a realizat prin referire la o singur aspect geometric: axa de rotație a caracteristicii.

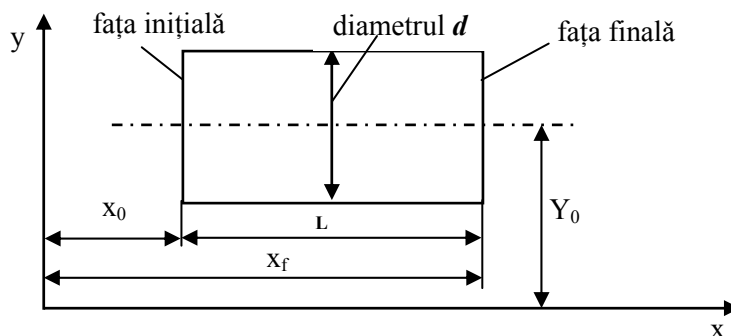


Fig. 6.27 Caracteristicile unui tronson cilindric

Structura de date ale produsului este extrem de importantă pentru asigurarea unui proces de proiectare corect, fără ambiguități. În figura 6.28 se prezintă un exemplu de *sub-ansamblu arbore_1* din cadrul ansamblului intitulat *transmisie*.

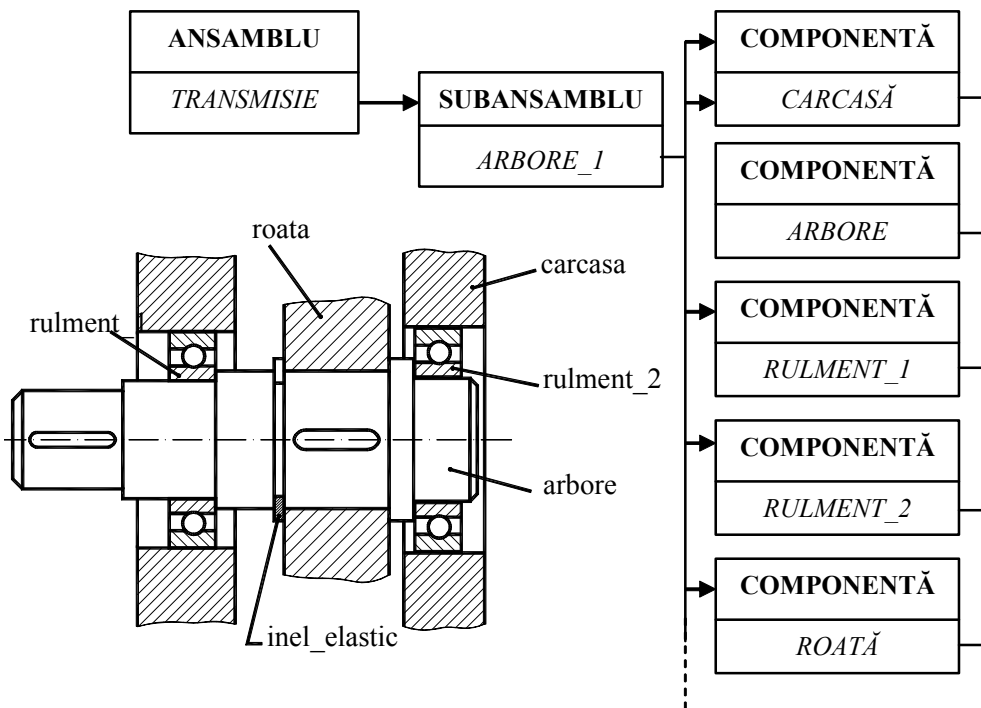


Fig. 6.28 Un subansamblu din ansamblul transmisie

Lista componentelor generate de acest subansamblu este precizată pentru o parte din componente.

Structura pe $tronson_i$ a componentei arbore este prezentată în figura 6.29 în corespondență cu aspectul grafic de reprezentare al componentei din figura 6.30.

Existența bazelor de date grafice pentru componente standardizate este esențială pentru sistemele CAD inteligente.

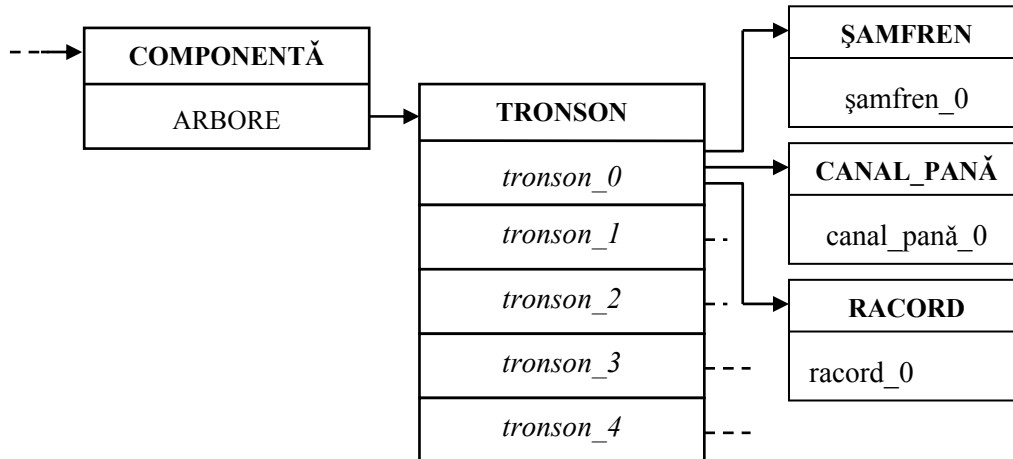


Fig. 6.29 Structura pe tronsoane a componentei arbore

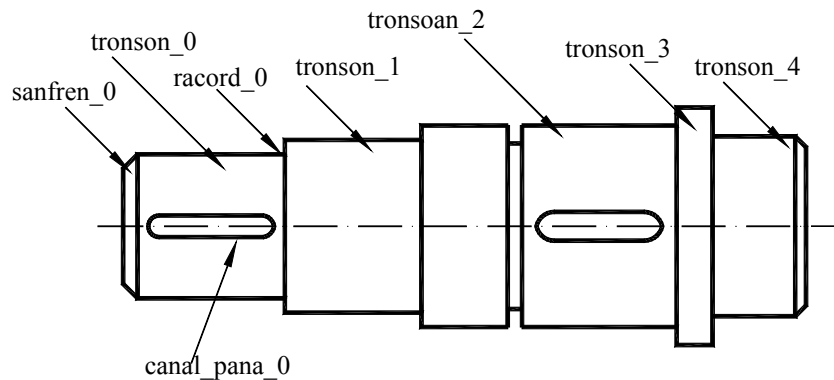


Fig. 6.30 Reprezentarea grafică a componentei arbore

6.4. Procesul de selectare optimală a variantelor proiectate

Se poate spune că în general într-un proces există mai multe posibilități de acțiune la un moment dat. Fie V_i ($i = 1, 2, \dots, m$) mulțimea variantelor posibile (fig. 6.31) din care trebuie aleasă *varianta optimă* care corespunde cel bine în raport cu o mulțime de criterii de comparație.

Decizia este hotărârea de a alege o anumită variantă de acțiune, din mai multe posibile, atașate unui anumit proces sau fenomen.

Procesul de decizie presupune o analiză atentă având un suport matematic corespunzător. Metodele de decizie în baza mai multor criterii se împart în două mari categorii:

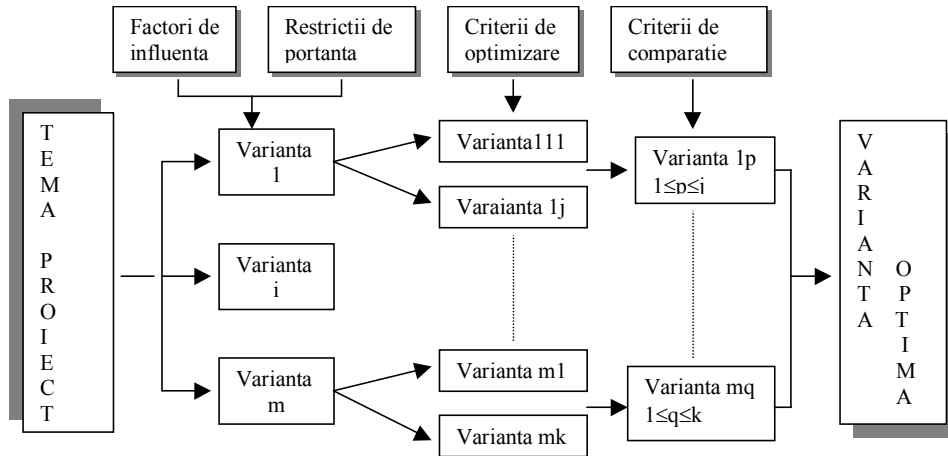


Fig. 6.31 Algoritm de proiectare optimă

A. **Modele de decizie multiobiectiv**, obiect al programării matematice. În faza de proiectare produsul este caracterizat printr-un set de *parametri de proiectare*, o serie de restricții de portanță și criterii de optimizare posibile.

Parametrii de proiectare pot fi grupați în trei categorii conform cu aspectele pe care le iau în considerare:

- *parametrii geometrici* definiți prin vectorul $X = (x_1, x_2, \dots, x_n)$ a cărui componente se identifică cu date dimensionale și de poziție privind elementele componente: lungimi "L" ale unor bare, lățimi "B" și grosimi "H" ale acestora, cote de gabarit, de legătură, de montaj etc.;
- *parametrii de material* definiți prin vectorul $Y = (y_1, y_2, \dots, y_n)$ a cărui componente se identifică cu rezistențe admisibile ale materialului ($\sigma_{ai}, \sigma_{0,2}$), module de elasticitate longitudinal și transversal (E, G), densitate ρ ;
- *parametrii de solicitare* definiți prin vectorul $Z = (z_1, z_2, \dots, z_n)$ a cărui componente se identifică cu componentele torsorului din sistem τ ;

Acești parametri permit definirea unor mărimi cantitative care să caracterizeze fiecare variantă de produs, proces dezvoltat în faza de proiectare: volumul poliedrului care înscrie construcția dată, suprafața, masa etc.

Restricțiile (de portanță, constructive) se pot defini prin relații matematice de constrângere și care trebuie să fie respectate, pentru ca varianta dezvoltată să fie acceptată:

$$G_i(X, Y, Z) \begin{cases} > \\ = \\ < \end{cases} G_{i0} \quad (6.21)$$

unde $i = 1 \dots n$ reprezintă numărul restricțiilor introduse.

Efortul maxim în secțiunile cele mai solicitate, parametrii geometrici de gabarit, masa produsului, frecvența de rezonanță, sensibilitatea etc. sunt câteva dintre

restricțiile care se pot introduce.

Criteriile de optimizare se exprimă matematic prin funcțiile obiectiv $F_j(X,Y,Z)$ și au în vedere o serie de caracteristici constructive sau funcționale ale produsului.

În baza celor prezentate, formularea problemei de optimizare, pentru o structură dată, impune determinarea parametrilor de proiectare

$$U = U(X, Y, Z) \tag{6.22}$$

care asigură extremul – minim sau maxim în dependență de criteriul ales – funcției obiectiv:

$$F_{j(U)} \Rightarrow \min(\max) \tag{6.23}$$

în prezența restricțiilor G_i .

Problema de optimizare poate să evidențieze insuficiența unei singure funcții – obiectiv. De regulă, aceste funcții obiectiv sunt contradictorii, ceea ce înseamnă că luarea în considerare a uneia dintre ele poate conduce la afectarea celorlalte.

B. Modele de decizie multiatribut. În activitatea complexă de proiectare proiectantul se găsește în situația de a selecta o variantă de produs dintr-un set de variante posibile (optimizate multiobiectiv). Sunt necesari algoritmi adecvați de clasificare, de sortare și de ierarhizare a informațiilor pe baza unor criterii anterior definite $C_j (j = 1, \dots, m)$.

Una dintre metodele de decizie are la baza matricea de decizie. *Matricea de decizie* este un instrument de proiectare utilă pentru stabilirea unor decizii. Forma de exprimare este tabelară (fig.6.32). Matricea este uzual aranjată cu criteriile de decizie (scop, obiective, constrângeri etc.) pe liniile matricei iar variantele de decizie formează coloanele. Fiecare variantă va fi notată în raport cu criteriile admise. În final se va alege varianta cea mai bine cotate pe întregul pachet de criterii.

Matricea de decizie	Pondere	Varianta A	Varianta B	Varianta C
Scorul total				
Criteriul 1				
Criteriul 2				
Criteriul 3				
Criteriul 4				

Fig. 6.32 Forma tabelară pentru matricea de decizie

Evaluarea fiecărei variante V_j în raport cu fiecare criteriu este oglindită în matricea consecințelor:

$$A = \left\| a_{ij} \right\|_{m \times n} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{pmatrix} \tag{6.24}$$

Criteriile de comparare a variantelor pot fi criterii *calitative sau cantitative*.

Criteriile calitative realizează compararea variantelor prin intermediul a 4 nivele – “mic”, “mediu”, “mare”, “foarte mare” – sau 5 nivele - – “mic”, “suficient”, “mediu”, “mare”, “foarte mare”. În baza acestei comparații, fiecare variantă primește în mod corespunzător notele 1, 3.5, 6.5, 9 și respectiv 1, 3, 5, 7, 9. În acest mod și aceste criterii devin cantitative.

Importanța acordată fiecărui criteriu se apreciază prin coeficienții de importanță. Literatura de specialitate utilizează o scară de importanță (tabelul 6.6) [6.49], [6.44], [6.29].

Tabelul 6.6

Intensitatea importanței $\frac{p_i}{p_j}$	Definiție	Explicație
1	Importanță egală	Cele două criterii contribuie egal la obiectiv
3	Importanță slabă	Experiența demonstrează o ușoară importanță a unui criteriu față de celălalt
5	Importanță puternică	Experiența demonstrează o importanță mai puternică a unui criteriu față de celălalt
7	Importanță demonstrată	Practica a dovedit importanța unui criteriu în raport cu celălalt
9	Importanță absolută	În mod evident un criteriu este mai important decât celălalt

În acest mod se poate defini un vector al coeficienților de importanță $P = (p_1, p_2, \dots, p_n)$ cu $\sum_{i=1}^n p_i = 1$. Totodată se poate stabili matricea importanței relative a criteriilor:

$$B = \begin{bmatrix} \frac{p_1}{p_1} & \frac{p_1}{p_2} & \frac{p_1}{p_3} & \dots & \frac{p_1}{p_n} \\ \frac{p_2}{p_1} & \frac{p_2}{p_2} & \frac{p_2}{p_3} & \dots & \frac{p_2}{p_n} \\ \frac{p_3}{p_1} & \frac{p_3}{p_2} & \frac{p_3}{p_3} & \dots & \frac{p_3}{p_n} \\ \dots & \dots & \dots & \dots & \dots \\ \frac{p_n}{p_1} & \frac{p_n}{p_2} & \frac{p_n}{p_3} & \dots & \frac{p_n}{p_n} \end{bmatrix} \quad (6.25)$$

Elementele matricei B au proprietățile:

$$b_{ij} = \frac{p_i}{p_j} = \frac{1}{b_{ji}} \quad (6.26)$$

Deoarece matricea consecințelor conține în cele mai multe cazuri date neomogene, numerice sau nenumere, pentru a fi comparate, este necesară omogenizarea acestora printr-un proces de scalare: ordinală, într-un interval și normalizare.

Scalarea ordinală are la bază o corespondență cu mulțimea numerelor naturale. Această metodă stabilește doar ordinea unor entități fără a indica și distanța dintre soluții.

Criteriile de decizie sunt comparate cu funcții obiectiv pe care trebuie să le îndeplinească variantele proiectate. Pentru a putea fi comparate aceste valori heterogene, se impune ca acestea să fie raportate pe un domeniu de evaluare $[0, M]$ unde M este valoarea maximă evaluată.

Metoda de construcție a modelului utilizează câteva facilități de evaluare a criteriilor, prin cuvinte cheie, în scopul utilizării unei funcții de agregare.

Tabelul 6.7

Ignoră	Ignoră acest criteriu în decizia modelului dominant
Crește	Alternativă ar trebui să aibă o valoare aproape de valoarea maximă (maximizează)
Descrește	Alternativa ar trebui să aibă o valoare aproape de valoarea minimă (minimizează)
Țintă	Alternativa ar trebui să aibă aceeași valoare ca baza de referință (constrângere de egalitate)

Metodele de proiectare pot stabili ca un parametru să se situeze într-un interval $[V_{\min}, V_{\max}]$. Modelul de decizie se poate construi astfel ca:

$$\text{IF NOT } (V_{\min} \leq V \leq V_{\max}) \text{ THEN evaluare} = 0 \quad (6.27)$$

Pentru criteriile de tip egalitate modelul de evaluare respectă relația de omogenizare:

$$\text{IF } V = \text{țintă THEN evaluare} = 1, \text{ OTHERWISE evaluare} = 0 \quad (6.28)$$

Se vor prefera variantele cu cele mai puține zerouri.

Scalarea pe un interval permite stabilirea distanței dintre variante. Mulțimea de corespondență este intervalul $[0, M]$. Pentru un criteriu de maxim valoarea de evaluare se definește ca fiind:

$$V_{scal} = M \cdot \left(1 - \frac{V_{\max} - V}{V_{\max} - V_{\min}} \right) \quad (6.29)$$

În mod similar pentru un criteriu de minim:

$$V_{scal} = M \cdot \left(1 - \frac{V - V_{\min}}{V_{\max} - V_{\min}} \right) \quad (6.30)$$

Normalizarea reprezintă transformarea matricii consecințelor $A = (a_{ij})$ într-o

matrice $R = (r_{ij})$. Există mai multe moduri de normalizare:

a) normalizare vectorială:

$$r_{ij} = \frac{a_{ij}}{\sqrt{\sum_1^n a_{ij}^2}} \quad \text{sau} \quad r_{ij} = \frac{a_{ij}}{\sum_1^n a_{ij}} \quad (6.31)$$

b) normalizarea prin transformări liniare:

• pentru criterii de maxim:

$$r_{ij} = \frac{a_{ij}}{a_j^{\max}} \quad \therefore a_j^{\max} = \max_i(a_{ij}) \quad \text{sau} \quad r_{ij} = \frac{a_j^{\max} - a_{ij}}{a_j^{\max} - a_j^{\min}} \quad (6.32)$$

• pentru criterii de minim:

$$r_{ij} = 1 - \frac{a_{ij}}{a_j^{\max}} \quad \text{sau} \quad r_{ij} = \frac{a_{ij} - a_j^{\min}}{a_j^{\max} - a_j^{\min}} \quad \therefore a_j^{\min} = \min_i(a_{ij}) \quad (6.33)$$

Determinarea coeficienților se realizează pe baza algoritmului:

1. Se determină valoarea proprie maximă λ_{\max} prin rezolvarea ecuației caracteristice:

$$\det[B - \lambda \cdot I] = 0 \quad (6.34)$$

unde $[I]$ este matricea unitate.

2. Se rezolvă sistemul de ecuații:

$$\begin{cases} (B \cdot P^T) = \lambda_{\max} \cdot P^T \\ \sum_{i=1}^n p_i = 1 \end{cases} \quad (6.35)$$

În cele prezentate anterior s-a presupus că fiecare variantă a fost evaluată în raport cu fiecare criteriu printr-o valoare cunoscută și unică. Deseori însă proiectantul dispune de o informație cu un grad mare de incertitudine. Acest lucru a determinat apelarea la teoria mulțimilor fuzzy pentru evaluarea variantelor. Metode referitoare la aplicarea acestei teorii în analiza multicriterială sunt prezentate în [6.49].

În metodele fuzzy aprecierea unei variante în raport cu un criteriu se face printr-un număr fuzzy care este de fapt un interval al dreptei reale [6.43].

Există o serie de metode matematice de ordonare a variantelor. Una dintre acestea este metoda TOPSIS (Technique for Order Preference by Similarity to Ideal Solution). Metoda se fundamentează pe următorul algoritm:

- se construiește matricea normalizată " R ";
- se construiește matricea normalizată ponderată $V=(v_{ij})$ unde $v_{ij} = p_j \cdot r_{ij}$;
- se determină soluția ideală V^+ :

$$v_j^+ = \begin{cases} \max_{1 \leq i \leq m} (v_{ij}) & \text{daca } C_j \text{ este de max} \\ \min_{1 \leq i \leq m} (v_{ij}) & \text{daca } C_j \text{ este de min} \end{cases} \quad (6.36)$$

și cea negativă V^- :

$$v_i^- = \begin{cases} \min_{1 \leq i \leq m} (v_{ij}) & \text{daca } C_j \text{ este de max} \\ \max_{1 \leq i \leq m} (v_{ij}) & \text{daca } C_j \text{ este de min} \end{cases} \quad (6.37)$$

- se calculează distanța între soluții:

$$S_{i^+} = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^+)^2} \quad (6.38)$$

$$S_{i^-} = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^-)^2} \quad (6.39)$$

unde $i = 1, 2, 3, \dots, m$.

- se calculează apropierea de soluția ideală:

$$C_i = \frac{S_{i^-}}{S_{i^+} + S_{i^-}} \quad (6.40)$$

- se realizează o ierarhizare a variantelor în ordine descrescătoare a criteriilor C_i . Metoda **AHP** (Analytic Hierarchy Process) face parte din metodele de decizie ierarhică și are la bază patru etape de stabilire a deciziei:

- *inițierea* – criteriile de definire a deciziei sunt stabilite adesea prin *brystorming* sau pe baza decidentului. Relațiile ierarhice între criterii sunt reprezentate sub formă matriceală.

De ex.: - *nivelul 1* se identifică cu problema de decizie abordată;

- *nivelul 2* se poate referi la prețul de cost, la fiabilitate, la ergonomie etc.

- *nivelul 3* ia în considerare o descompunere a criteriilor de la nivelul anterior. De ex.: criteriul prețul de cost poate fi divizat în: prețul de cost de laborator, costuri maximale, costuri variabile.

În figura 6.33 se prezintă o ierarhizare a criteriilor iar în figurile 6.34 – 6.35 o reprezentare matriceală a criteriului fiabilitate.

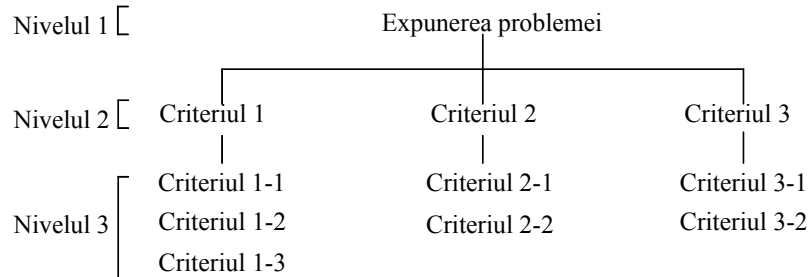


Fig. 6.33 Criterii ierarhice

Nivelul 2	Cost	Fiabilitate	Ergonomie
Cost			
Fiabilitate			
Ergonomie			

Fig. 6.34 Criterii de analiză

Fiabilitate	Sistem comanda	Sistem actionare	Transmisie
Sistem comanda			
Sistem actionare			
Transmisie			

Fig. 6.35 Subcriterii

- *evaluarea* – criteriile introduse în matricile de ierarhizare sunt comparate între ele referitor la importanța relativă. O scară a importanței relative utilizează valorile numerice reprezentate în tabelul 6.6. Utilizând relația (6.25) se poate completa întreaga matrice (fig.6.36).

Nivelul 2	Cost	Fiabilitate	Ergonomie	
Cost	1,000	$x = 3,000$	$y = 0,333$	
Fiabilitate	$1/x = 0.333$	1,000	$y/x = 0.111$	
Ergonomie	$1/y = 3,000$	$x/y = 9,000$	1,000	

Fig. 6.36 Completarea matricii de evaluare

Pe baza analizei anterioare se poate determina vectorul de evaluare (fig.6.37). Procesul de comparare se repetă pentru toate matricile de analiză. Vectorii de evaluare (de pondere) ale matricilor inferioare vor fi normalizați prin ponderea asociată matricii subcriteriului, astfel încât ponderea lor totală o va egala pe cea a criteriului de nivel precedent. De exemplu pentru subcriteriul cost vectorul de evaluare normalizat este prezentat în figura 6.38.

Nivelul 2	Cost	Fiabilitate	Ergonomie	Vector de evaluare
Cost	1,000	$x = 3,000$	$y = 0,333$	$x/(x+y+z) = 23.1 \%$
Fiabilitate	$1/x = 0.333$	1,000	$y/x = 0.111$	$y/(x+y+z) = 7.7 \%$
Ergonomie	$1/y = 3,000$	$x/y = 9,000$	1,000	$z/(x+y+z) = 69.2 \%$

Fig. 6.37 Vectorul de evaluare

Cost	Laborator	Cheltuieli generale	Cheltuieli variabile	Vector evaluare	Vector normalizat
Laborator	1,000	3	5	65.4 %	15.1 %
Cheltuieli generale	0.33	1	1,67	21.6 %	5.00 %
Cheltuieli variabile	0,2	0,6	1	13.1 %	3.00 %
				100 %	23.1 %

Fig. 6.38 Vector de evaluare normalizat

- *ierarhizarea* – fiecărei soluții potențiale după fiecare criteriu generat în primul pas. Ordonările sunt realizate pornind de la nivelul cel mai detaliat al fiecărei ierarhii. Abordarea în echipă este benefică pentru această metodă.
- *evaluarea finală* - reprezintă ultimul pas al metodei. Matricea finală pentru un anumit criteriu și soluțiile posibile vor indica varianta optimă pentru criteriul avut în vedere.

6.5. Bibliografia capitolului 6

- [6.1] Achostachioaie, D., Programarea în C++, www.biosfarm.ro/~dragos
- [6.2] Andrașiu, M., ș.a., Metode de decizie multicriteriale, Editura Tehnică, București, 1986
- [6.3] Arcidiacomo, G., s.a., Reliability improvement of car sliding door using axiomatic approach, Proc. of ICAD2004, The Third Intern. Conf. on Axiomatic Design, ICAD-2004-18, Seul, 2004
- [6.4] Arcidiacomo, G., s.a., Axiomatic design for six sigma, Proc. of ICAD2002, Second Intern. Conf. on Axiomatic Design, ICAD004, Cambridge, 2002
- [6.5] Basem, E.-H., Axiomatic design quality engineering – a transmission planetary case study, Proc. of ICAD2004, The Third Intern. Conf. on Axiomatic Design, ICAD-2004-10, Seul, 2004
- [6.6] Betting, B., Design Automation. Theories & Implementation, www.me.mtu.edu/~betting/MEEM5408
- [6.7] Bunus, P., Engelson, V., Fritzson, P., Mechanical Models Translation, Simulation and Visualization in Modelica, Proc. Of Modelica Workshop 2000, p.199-208, Oct., Lund (Sweden), 2000
- [6.8] Case, K., Gao, J.X., Gindy, N.Z., The Implementation of a Feature-Based Component Representation for CAD/CAM Integration, http://www.lboro.ac.uk/departments/mm/research/product-realisation/res_int/features/pplan/pplan.html
- [6.9] Dolga, V., Considerații privind alegerea dispozitivului de prehensiune optim, Simp. Național de RI, Brașov, 1986, p.147-154
- [6.10] Dolga, V., The optimal selection of the industrial robot for a certain industrial process, 6th Intern. Conf. On Flexible Tech., p.275-282, 1997, (Sombor – Yugoslavia)
- [6.11] Dolga, L., Revencu, M., Maci, C., Giuchici, M., Parametric and feature –based modeling with applications in CAIA and INVENTOR, Ed. Politehnica, Tmișoara, 2004
- [6.12] Domb, E., Terninko, J., Miller, J., MacGran, E., The seventy-six Standard

- Solutions: How They Relate to the 40 Principles of Inventive Problem Solving, <http://www.triz-journal.com/archives/1999/05/e/index.htm>
- [6.13]Dukhyun, C., A suggestion and a contribution for the improvement of Axiomatic Design, Proc. of ICAD2004, The Third Intern. Conf. on Axiomatic Design, ICAD-2004-02, Seoul, 2004
- [6.14]Esawi, A.M.K., Ashby, M.F., The Development and Use of a Software Tool for selecting Manufacturing Processes at the early stages of Design, Trans. of SDPS, june 2000, vol.4, n.2, p.27-43
- [6.15]Fritzson, P., Engelson, V., Modelica – A Unified Object-Oriented Language for System Modeling and Simulation, ...
- [6.16]Gao, J., Case, K., Gindy, N., Geometric Elements for Tolerance Definition in Feature-Based Product Models, http://www.lboro.ac.uk/departments/mm/research/product-realisation/res_int/features/...
- [6.17]Goncalves-Coelho, A., s.a., A rational way to select a measuring system for mechanical parts inspection, Proc. of ICAD2004, The Third Intern. Conf. on Axiomatic Design, ICAD-2004-42, Seoul, 2004
- [6.18]Guenov, M.D., Complexity and Cost Effectiveness Measures for Systems Design, Manufacturing Complexity Network, Cambridge, 2002
- [6.19]Hilario, L.O., Unifying axiomatic design and robust design through the transfer function, Proc. of ICAD2004, The Third Intern. Conf. on Axiomatic Design, ICAD-2004-09, Seoul, 2004
- [6.20]Hrishikesh, V.D., Suh, N.P., Axiomatic design of customizable automotive suspension, Proc. of ICAD2004, The Third Intern. Conf. on Axiomatic Design, ICAD-2004-38, Seoul, 2004
- [6.21]Hu, M., Yang, K., Taguchi, S., Enhancing Robust Design with the Aid of TRIZ and Axiomatic Design(Part I),<http://www.triz-journal.com/archives/2000/10/e/index>
- [6.22]Ioachim, A. A., Abordarea orientată spre obiecte în aplicațiile AutoCAD, Tehnică și tehnologie, nr. 1, 2001, p.32 – 34
- [6.23]Jonghyup, J., Moon, W., Jonghyu, P., Design improvement in a micro-gripper system by axiomatic approach, Proc. of ICAD2004, The Third Intern. Conf. on Axiomatic Design, ICAD-2004-40, Seoul, 2004
- [6.24]Kalak, O., s.a., Multi-attribute material handling equipment selection using information axiom, Proc. of ICAD2004, The Third Intern. Conf. on Axiomatic Design, ICAD-2004-30, Seoul, 2004
- [6.25]Kar, A.K., Linking axiomatic design and Taguchi methods via information content in design, Proc. of ICAD2000, First Intern. Conf. on Axiomatic Design, ICAD033, Cambridge, 2000
- [6.26]Kirschman, C.F., Fedel, G.M., Jara-Almonte, C.C., Classifying Functions For Mechanical Design, Proc.ASME , 1996, Irvine, California, 96-DETC/DTM-1504
- [6.27]Lee, K.W., s.a., A structural optimization methodology using the independence axiom, Proc. of ICAD2000, First Intern. Conf. on Axiomatic Design, ICAD002, Cambridge, 2000
- [6.28]Leon, F., Ingineria programării, http://eureka.cs.tuiasi.ro/~fleon/curs_ip.htm
- [6.29]Marsh, E.R., s.a., Hierarchical Decision Making in Machine Design, (-)
- [6.30]Maziero, N.L., Ferreira, J.C.E., Pacheco, F.S., Prim, M.F., A Feature-Based

- Object-Oriented Expert System to Model and Support Product Design, J. Braz. Society of Mech. Sci., v.22, n.4, 2000, p.1-29
- [6.31]Maziero, L.N., Ferreira, J.C.E., Automatic Dimensioning of Cylindrical Parts in an Intelligent Feature-Based Design System, <http://www.grima.ufsc.br/papers/PapFeats2001Nilson.pdf>
- [6.32]Millea,A.,Cartea metrologului. Metrologie generală, Ed.Tehnică, București, 1985
- [6.33]Pappalardo, M., Axiomatic approach in the analysis of data for describing complex shapes, Proc. of ICAD2004, The Third Intern. Conf. on Axiomatic Design, ICAD-2004-43, Seul, 2004
- [6.34]Park, J.H., s.a., Axiomatic design of a wing driving mechanism for micro air vehicle with flapping motion, Proc. of ICAD2004, The Third Intern. Conf. on Axiomatic Design, ICAD-2004-05, Seul, 2004
- [6.35]S.H., Park, G.J., Application of design axioms for glass bulb design and software development for design automation, Trans. of ASME, vol.123, sept. 2001, p.322-329
- [6.36]Shah, J.J., Rogers, M.T., Feature Based Modelling Shell: Design and Implementation, Proc. of ASME Computers in Eng. Conference, San Francisco, 1988
- [6.37]Shah, J.J., Mantyla, M., Parametric and Feature-based CAD / CAM. Concepts, Techniques, and Applications, John Wiley & Sons, Inc., New York, 1995
- [6.38]Shin, G.S., s.a., Calculation of information content in axiomatic design, Proc. of ICAD2004, The Third Intern. Conf. on Axiomatic Design, ICAD-2004-22, Seul, 2004
- [6.39]Sinha, R., Paredis, C.J.J., Khosla, P.K., Kinematics Support for Design and Simulation of Mechatronic Systems, (-)
- [6.40]Slocum, S.M., Measurement and Detection Standards from the Theory of Inventive Problem Solving, <http://www.triz-journal.com/archives/2002/09/a/index.htm>
- [6.41]Suh, N.P., Axiomatic Design – Advances and Applications (chapter 5, p.239-298), Oxford University Press, New York, Oxford 2001
- [6.42]Ullah, S., Integration of high-level design information with axiomatic design formulation, Proc. of ICAD2004, The Third Intern. Conf. on Axiomatic Design, ICAD-2004-44, Seul, 2004
- [6.43]Terninko, J., Domb, E., Miller, J., The seventy-six Standard Solutions, with Exemples-Class 5, <http://www.triz-journal.com/archives/2000/07/b/index.htm>
- [6.44]Terninko, J., Selecting the Best Direction to Create the Ideal Product Design, <http://www.triz-journal.com/archives/1998/07/c/index.htm>
- [6.45]Werner, H., Muth, M., Weber, C., Functional Modelling Using an Object-oriented Design System, <http://www.cad.uni-sb.de/dokumente/werner/hwiced97.htm>
- [6.46]Yang, K., Zhang, H., A Comparison of TRIZ and Axiomatic Design, <http://www.triz-journal.com/archives/2000/08/d/index.htm>
- [6.47]***, Controlled convergence, <http://betterproductdesign.net/tools/concept/convergence.htm>
- [6.48]***, Boothroyd & Dewhurst complexity factor, http://betterproductdesign.net/tools/production/B&D_complexity.htm
- [6.49]***, Boothroyd & Dewhurst complexity factor, <http://betterproductdesign.net/tools/production/efficiency.htm>