

Utilizari ale metodelor structural-semiotice în realizarea sistemelor informatice (partea I)

Ec. Robert KOMARTIN
Oracle România

The discrepancy between the efficiency of the application development phases against the analysis and design phases in almost all methods and methodologies currently in use it is obvious, especially due to a fundamentally different level of formalization of the two areas: while the programming languages area has a strong theoretical foundation for few decades already (formal grammars, compilers theory, etc.), the theoretical foundation of the analysis – and also of the design – are still at a ‘folklore and craftsmanship’ level.

Moreover, due to this discrepancy, which generates huge differences in operational performance, the focus of the ‘practicians’ moved towards these phases/moments in the development process. Consequently – and correlated with the lack of adequate mathematical support – the appearance of a ‘de-mathematization’ of the information systems development methods was created – the modern system analysis and IS project management courses are rather belonging to the social sciences field (sociology, ethnology), rather than the field originating them – computer science.

That’s why we consider that what is actually necessary to improve the performances of the information systems development processes is not ‘de-mathematization’ but rather ‘re-mathematizing’ them in a proper manner. One possible way – presented below – in order to obtain significant increase in methodological quality is a structuralist and semiotic approach of the information systems development processes. The proposed approach is based on semiotic algebra, as defined by the group led by prof. Joseph Goguen at the University of San Diego.

Keywords: *information systems, structuralism, semiotics, semiotic algebra, semiotic morphisms, semiotic projections, best practice and representation support systems.*

1 Problematizare

Structuralismul, asemenea scolii de gândire sistemică (mai precis, curentele ce se revendică de la Teoria Generală a Sistemelor - TGS) analizează realitatea ca fiind compusă din ansambluri – la diferite nivele – de elemente grupate într-o manieră, însă spre deosebire de acesta, pune accentul pe formalizarea relațiilor existente între elementele ansamblului (structura) și mai puțin pe definirea elementelor, considerate de cele mai multe ori ca fiind atomice (i.e. indivizibile).

Este interesant de observat că dihotomia este de dată recentă, F. de Saussure – pionier al semioticii și predecesor al structuralismului – folosind în lucrările sale numai termenul de “sistem” și niciodată pe acela de “structură”. În definiția dată de R. Barthes, structuralismul în sens larg privește toate teoriile care examinează sistemul limbii în

immanentă și în genere toate cercetările sistematice subordonate „pertinenței semantice și inspirate de modelul lingvistic”.

Abordările structuraliste și-au găsit un domeniu fertil de dezvoltare în științele umane (ele precedând în mod direct gramaticile generativ-transformaționale inițiate de Noam Chomsky) și în ultimele două decenii și în matematici, însă au “ocolit” domeniul informaticii aplicate.

În cele ce urmează vom introduce cititorului o metodă de sorginte structural-semiotică, bine conturată în arii sau domenii conexe celui studiat, însă neaplicată – din câte știm – în domeniul realizării sistemelor informatice. Metoda introdusă este bazată pe algebra semiotică, în accepțiunea scolii de la Universitatea din San Diego condusă de prof. Joseph Goguen.

Aplicabilitatea acestei metode consta atât în introducerea unui mod sistematic de analiza a metodologiilor existente, cât și în utilizarea ei ca instrument de lucru în diferitele faze ale procesului de realizare a sistemelor informatice integrate. De asemenea, vor fi introduse o serie de concepte originale, utile în particular în analiza diverselor metode și metodologii de analiza existente.

Trebuie subliniat ca alături de metoda mai sus amintită, ce va fi prezentată în detaliu în cele ce urmează, considerăm ca există o serie de alte metode de inspirație structuralist/semiotica cu aplicații potențiale în realizarea sistemelor informatice, între care amintim:

- LSA (Latent Semantic Analysis – analiza semantica latentă) – vezi [LAND98] pentru mai multe detalii
- Gramatici dinamice – metoda aplicată de asemenea în analiza semantica – vezi [BOUL94] pentru informații suplimentare.
- DSM (Design Structure Matrix – matricea de structura a proiectării) – utilizabilă în analiza structurală a metodologiilor cadru de implementare

2. Algebra semiotica

În cele ce urmează vom încerca să descriem – pomind de la considerațiile menționate în paragraful anterior și bazându-ne – în ceea ce privește partea de definiții și exemplificări „standard” – pe lucrările lui Joseph Goguen [GOGU91], [GOGU99] – o abordare semiotica formalizată algebric referitoare la analiza și modelarea sistemelor informatice.

Următoarele idei sunt fundamentale pentru considerațiile ce urmează:

- Semnele apar ca membre ale unor *sisteme de semne*, și nu în izolare (concepție datorată lingvistului elvețian Ferdinand de Saussure, a cărui operă a inspirat mișcările franceze structuraliste și post-structuraliste)
- Cele mai multe semne sunt obiecte complexe, construite din alte semne de nivel mai scăzut
- Sistemele de semne sunt mai degrabă *teorii* – declarații ale unor simboluri plus propoziții, numite „axiome”, care restricționează folosirea simbolurilor – decât modele.
- Reprezentanții în general și diversele diagrame din cadrul metodelor de analiza și proiectare

în particular sunt „morfisme” (legături) între sisteme de semne.

2.1 Sisteme de semne

Charles Saunders Peirce [PEIR65], logician și filozof american din secolul XIX, a introdus termenul „semiotica” și a definit multe din conceptele sale de bază. Peirce a subliniat că înțeleșurile nu sunt atașate direct semnificanților ci că înțeleșul (sensul) este *mediat* prin intermediul unor evenimente (sau procese) de semioza, fiecare implicând un **semnificant** (un „token” de un fel oarecare), un **semnificat** (un „obiect” oarecare – cum ar fi o idee) și un **interpretant** care le leagă pe cele două (a nu se confunda interpretantul cu interpretorul – cel care interpretează).

Cele trei elemente sunt numite un „semn” sau – pentru a evita confuziile posibile – „triada semiotica”; cele trei apar ori de câte ori există un sens. Semnele și înțeleșurile există numai pentru o semioza dată, care include și un context; prin urmare, înțeleșul este întotdeauna *încorporat*. În general, semnificatul nu este dat, ci trebuie dedus de persoana/persoanele implicate. Proiectanții creează în direcția opusă, construind semne pentru un semnificat dat.

Deși abordarea lui Peirce poate părea simplă, dar este foarte diferită de abordările „naive”, cum ar fi semantica denotatională din limbajele de programare. Teoria semnelor a lui Peirce nu este o *teorie reprezentatională a semnificatiei*, în care un semn are o semnificație, ci – prin intermediul interpretantului – este o *teorie relatională a semnificatiei*. Vom discuta mai jos și alte noțiuni introduse de Peirce (icon, index și simbol) și vom folosi termenul *semnal* pentru o configurație fizică ce poate sau nu să fie semn.

Compunerea semnelor din alte semne este o strategie fundamentală pentru comunicațiile netriviabile, prin abordarea semnelor complexe (compuse) de pe un nivel ca semne individuale de un nivel superior. Un exemplu direct îl constituie nivelele diferite de detaliu ale DFD-urilor (diagramele de flux de date) din metodologia SSADM. Această structură ierarhică parte/întreg plasează fiecare semn într-un context de alte semne împreună cu care formează mai departe nivele de nivel mai înalt. Utilizări mai recente ale sistemelor de semne (cum ar fi studiile lui Roland Barthes) tind să fie mai puțin rigide decât

lingvistica lui Saussure sau antropologia structurala a lui Claude Levi-Strauss. În locul opozitiilor binare, avem de a face cu scale multi-valoare sau chiar continue; în locul construcției semnelor de nivel înalt prin compunere secvențială, se folosesc relații complexe de inter-relationare sau influență; poate cel mai important, este recunoscută într-o mai mare măsură dependența de context (vezi în acest sens și paragraful 4.4 de mai jos).

Sistemele vocalice cu accente diferite din cadrul aceleiași limbi arată ca același sistem de semne poate fi realizat în mai multe feluri; vom numi aceste modalități **modele** ale sistemului de semne. Pentru informaticieni, poate fi util să interpreteze sistemele de semne ca (un fel de) tipuri abstracte de date (ADT), deoarece acestea includ ideea că aceeași informație poate fi reprezentată în modalități multiple; spre exemplu, datele calendaristice au modalități multiple de reprezentare.

Alfabetele grec, latin și chirilic demonstrează că multimile ce stau la baza modelelor de reprezentare se pot suprapune (în toate trei există elementul A); acest exemplu mai demonstrează și că un semnal care poate avea sens într-un sistem de semne poate să nu aibă sens într-altul (de exemplu, Δ nu are sens în alfabetul latin), cu toate că se manifestă în același mediu (un mediu exprimă dimensiunile în cadrul cărora semnalele pot varia).

De asemenea, același semnal într-un alfabet diferit este un semn diferit (spre exemplu, B latin și B chirilic), deoarece este într-un sistem de semne diferit. Sistemul vocalic mai demonstrează și faptul că modele diferite ale aceluiași sistem de semne pot folosi exact aceleași semnale în feluri diferite; prin urmare, diferențierea dintre modele este dată de modul de utilizare al elementelor, și nu de elementele în sine.

Să observăm că în economia semioticii sistemele de semne sunt mult mai importante decât proprietățile și parametrii semnelor individuale. De exemplu, formalizarea literei „a” este mult mai puțin utilă decât formalizarea relațiilor dintre litere și cuvinte.

Algebra semiotică propune un cadru de studiu precis al sistemelor de semne și al reprezentărilor lor, precum și studiul calitatii sistemelor de semne (care sisteme de semne sunt mai adecvate pentru un anumit scop și de ce), precum și modalități de

combinare/decombinare a reprezentărilor. Așa după cum subliniază Goguen, numele metodei putea la fel de bine să fie **semiotica structurală**, pentru a sublinia caracterul structural al sensului, sau **semiotica morfică**, pentru a pune accentul pe caracterul dinamic, contextual, încorporat și social al înțeleșului.

Vom da în cele ce urmează definiția formală a unui sistem de semne S, urmând mai apoi să furnizăm clarificări cu privire la termenii utilizați:

Definiție: Un sistem de semne S constă din:

1. O mulțime S de **sorturi** pentru semne, nu neapărat disjuncte;
2. O relație parțială de ordine pe S, numită relație **subsort** și notată \leq ;
3. O mulțime V de **sorturi de date**, pentru informații despre semne;
4. O relație de ordine parțială a sorturilor pe nivele, în așa fel încât sorturile de date sunt inferioare sorturilor de semne și astfel încât există un singur sort de nivel maximal, numit **top sort**;
5. O mulțime C_n de **constructori de nivel n**, folosiți pentru a construi semnele de nivel n din alte semne, de nivel n sau mai mic, și scris $r: s_1 \dots s_k d_1 \dots d_l \rightarrow s$, indicând faptul că argumentul i trebuie să aibă sortul s_i și parametrul j sortul de date d_j , și sortul rezultat este s. Sunt permise și construcțiile $c \rightarrow s$, unde c este o constantă.
6. O relație de ordine (parțială) pe fiecare C_n numită **prioritate**;
7. O mulțime R de relații și funcții aplicabile semnelor, numită **atribute** și
8. O mulțime A de propoziții (în sensul logicii), numite **axiome**, care constrâng/limitează mulțimea semnelor posibile.

În mod deliberat a fost evitat termenul de tip și a fost utilizat cel mai puțin familiar de sort – deoarece termenul tip are un mare număr de sensuri în informatică, această polisemie putând îngreuna înțelegerea. Un exemplu simplu de sort îl constituie așa numitele parti de vorbire din sintaxă (subiect, predicat, atribut, complement). Vom presupune că sorturile au o structură ierarhică – definită prin relația de ordine parțială de subsort. Ca exemplu, sortul *timp* este un subsort al sortului *dimensiune*. Pe lângă constructori, într-un sistem de semne putem avea **atribute** definite pe semne; acestea sunt funcții sau **predicte** definite

pe semnele unui anumit sort si luând valori în sorturi ce nu sunt neaparat semne (numite **sorturi de date**). Spre exemplu, o functie locatie va oferi coordonatele unui anumit obiect grafic. Exemple de sorturi de date: culori, locatii, valori de adevar. Se poate face – pentru a usura înțelegerea - o analogie între sistemele de semne si spatiile vectoriale: în timp ce semnele sunt vectorii, attributele (sau mai precis, valorile aparținând sorturilor de date) sunt scalarii sistemului.

Constructorii sunt reguli de combinare ale semnelor de sorturi bine definite într-un nou semn, de un alt sort. Modul lor de definire sugereaza faptul ca, aditional sorturilor, constructorii pot avea parametri, ce nu afecteaza natura semnului obtinut (schimbarea dimensiunilor si culorii unei anumite forme geometrice nu altereaza natura/identitatea formei geometrice).

O notiune echivalenta cu sistemul de semne este aceea de **spatiu conceptual** (Fauconier si Turner), cu observatia ca într-un spatiu conceptual nu exista decât constructori constante si ca în plus sunt definite relatii între aceste constante.

Sa ilustram definitia de mai sus cu câteva sisteme de semne foarte simple. Un prim exemplu este sistemul *ora din zi*. Are un singur sort (timpul) si doi constructori (constanta de timp 0 si operatia de succesiune $s(t)$ ce semnifica minutul urmator lui t). Nu exista subsorturi, sorturi de date, nivele sau prioritati. Exista o singura axioma: $s^{1440}(t) = t$, unde s^{1440} indica aplicarea de 1440 de ori a lui s .

Axioma captureaza natura ciclica a timpului (sau, mai precis, a reprezentarii timpului) pe durata unei zile. Orice reprezentare rezonabila a orei din zi trebuie sa satisfaca aceasta conditie.

Reiese aproape imediat din definitia de mai sus ca un sistem de semne se refera exclusiv la aspectul conceptual. Abia **modelul** (vezi definitia de mai sus) ofera o multime de semne pentru fiecare sort si operatiile efective aplicate acestor sorturi care construiesc noi semne din cele deja existente. Diferenta între model si sistemul de semne este ca sistemul reprezinta mai degraba o teorie, un limbaj, în timp ce modelul ofera interpretarea concreta a elementelor din teorie (sorturile sunt interpretate ca multimi, constantele sim-

bolice ca elemente, constructorii ca functii, etc.). Asupra modelelor se pot impune doua conditii:

- Conditia de **neambiguitate** – daca doi termeni nu pot fi demonstrati ca fiind egali folosind axiomele din teorie, atunci ei trebuie sa denote elemente diferite din model (se doreste excluderea modelelor în care doi termeni diferiti denota acelasi lucru).

- Conditia de **accesibilitate** – se doreste sa se restrictioneze gama reprezentarilor la acele modele în care toate semnele sunt denotate prin termeni din teoria subiacenta.

În rezumat, modelele de sisteme de semne nu trebuie sa fie numai multimi de elemente pentru reprezentarea semnelor, ci trebuie sa aiba si structura aditionala ceruta de sistemul de semne dat, inclusiv operatiile si attributele sale. În particular, se poate demonstra ca modele unui sistem de semne sunt *algebre*

2.2 Morfisme semiotice

Una din marile idei ale matematicii secolului XX, cu consecinte înca neexplorate integral, este aceea ca transformarile ce conserva structura sunt adesea cel puțin la fel de importante ca si structurile însele. Un exemplu clasic si oarecum „avant la lettre” este algebra liniara, preocupata în mod fundamental de transformari (adesea reprezentate prin matrice) între spatii vectoriale, mai degraba decât de studiul spatiilor vectoriale însele. Ideea fundamentala este ca reprezentarile, metaforele, interpretarile, interfetele, etc. – sunt morfisme de la un sistem de semne la altul.

Aceasta revolutie conceptuala a capatat o forma mai bine definita o data cu inventarea teoriei categoriilor în anii '40 de catre Eilenberg si MacLane (vezi [GOGU91] pentru mai multe detalii despre teoria categoriilor). Desi în plina dezvoltare în matematicile fundamentale, teoria categoriilor nu a fost aplicata decât marginal în informatica – si cu predilectie în domenii „exotice” cum ar fi inteligenta artificiala, însa nu (din câte stim) în domeniul realizarii sistemelor informatice. Reprezentarile sunt semne în cadrul unui sistem de semne care se relationeaza sistematic cu semnele din cadrul unui alt sistem. În general, este la fel de inutil sa studiem reprezentari ale unor semne izolate ca si studiul semnelor izolate în sine, deoarece reprezentarile apar de asemenea ca sistem, existând regularitati sistematice în modul

în care semnele unui sistem sunt reprezentate ca semne ale unui alt sistem. Vom folosi notatia $M : S_1 \textcircled{R} S_2$ pentru un morfism din sistemul de semne S_1 în sistemul de semne S_2 , cu observatia ca în cazul general nu exista un singur morfism între cele doua sisteme. Scopul unui morfism semiotic este prin urmare de a furniza o modalitate de a descrie miscarea (translatarea, interpretarea, reprezentarea) semnelor din cadrul unui sistem într-un alt sistem. Aceasta include atât notiuni „poetice”, cum ar fi explicarea unei metafore, cât si chestiuni mai apropiate de domeniul studiat, cum ar fi reprezentarea diagramatica a domeniului unei probleme date. Asa dupa cum am definit sistemele de semne ca fiind teorii mai degraba decât modele, tot asa morfismele au loc între teorii, translatând din limbajul unui sistem de semne în limbajul unui alt sistem de semne, în loc sa se margineasca la translatarea semnelor din modele. Un morfism semiotic bun trebuie sa conserve cât mai mult din structura sistemului de semne sursa – aceasta implicând reprezentarea sorturilor, a subsorturilor, a sorturilor de date, a constantelor, a constructorilor, etc. Cu toate acestea, morfismele pot fi pariale (nedefinite pentru toate semnele din sistemul sursa, unele semne putând fi netranslatabile sau cel puțin netranslate de un anumit morfism).

Definitie: Fiind date sistemele S_1 si S_2 , un **morfism semiotic** $M : S_1 \textcircled{R} S_2$ de la S_1 la S_2 consta în urmatoarele functii pariale (notate toate M):

1. Sorturile lui $S_1 \textcircled{R}$ sorturile lui S_2 ;
2. Constructorii lui $S_1 \textcircled{R}$ constructorii lui S_2 ;
3. Predicatele si functiile lui $S_1 \textcircled{R}$ predicatele si functiile lui S_2 ; în asa fel încât:
 1. Daca $s \leq s'$ atunci $M(s) \textcircled{R} M(s')$
 2. Daca $c: s_1...s_k \textcircled{R} s$ este un constructor (sau o functie) a lui S_1 , atunci (daca este definit) $M(c) : M(s_1)...M(s_k) \textcircled{R} M(s)$ este un constructor (sau o functie) a lui S_2
 3. Daca $p: s_1...s_k$ este un predicat al lui S_1 , atunci (daca este definit) $M(p) : M(s_1)...M(s_k)$ este un predicat al lui S_2
 4. M este functia identitate pe toate sorturile si operatiile pentru date din S_1

Trebuie remarcat ca – din punct de vedere conceptual – morfismele semiotice sunt de fapt ex-

tensii ale notiunii de model; desi mult mai puțin intuitiv, prin morfisme se pot obtine rezultate si generalizari mai ample decât prin specificarea de modele.

Câteva exemple simple: fie N_1 sistemul de numerotatie arab si N_2 sistemul de numerotatie roman. Atunci exista un morfism evident $M : N_1 \textcircled{R} N_2$, care însa este nedefinit pentru 0 arab, deoarece conceptul de zero era inexistent în sistemul roman. Putem de asemenea sa consideram transliteratii între alfabetul grec si cel latin, însa unele litere nu vor putea fi transformate, neavând un echivalent direct.

Un alt exemplu îl constituie reprezentarea sistemului de semne *ora din zi*, descris în paragraful anterior într-un sistem grafic standard (monitorul unui calculator). Putem identifica cu usurinta o mare varietate de modalitati de reprezentare, mergând de la simpla afisare a numarului t (cuprins între 0 si 1439) pâna la modele de ceas analogic, cu ac orar si minutar.

Bibliografie

- [BOUL94] - P. Boullier, *Dynamic grammars & semantic analysis*, Rapport de recherche No. 2322, Institut National De Recherche en Informatique et en Automatique, 1994
- [FORR79] - J.W. Forrester, *Principiile sistemelor. Teorie si autoinstruire programata*, Ed. Tehnica, Bucuresti, 1979
- [GOGU91] - J. Goguen - *A categorical manifesto*, Mathematical Structures in Computer Science, 1(1):49-67, March 1991
- [GOGU99] - J. Goguen - *An Introduction to Algebraic Semiotics, with Applications to User Interface Design*, Computation for Metaphors, Analogy and Agents, Lecture Notes in Artificial Intelligence, Volume 1562, pages 242-291, Springer, 1999
- [LAND98] - T.K. Landauer, P.W. Foltz, D. Laham, An Introduction to Latent Semantic Analysis, in *Discourse Processes*, 25, pg. 259-284, 1998
- [LUNG94] - I. Lungu, Gh. Sabau, T. Surcel, C. Bodea, *Sisteme informatice pentru conducere*, Ed. SIAJ, Bucuresti, 1994
- [OPRE99] - D. Oprea, *Analiza si proiectarea sistemelor informatice economice*, Ed. Polirom, Iasi, 1999

[PEIR65] - C. S. Peirce. *Collected Papers of Charles Saunders Peirce*, Harvard, 1965

[SERB87] - L. D. Serbanati, *Limbaje de programare si compilatoare*, Editura Academiei Republicii Socialiste România, Bucuresti, 1987