

Relatia dintre probabilitatea de eroare si metricile OO

Lect. Marian CRISTESCU
Universitatea "Lucian Blaga" Sibiu

Object-oriented design and development is becoming very popular in today's software development environment. Object oriented development requires not only a different approach to design and implementation, it requires a different approach to software metrics. Since object oriented technology uses objects and not algorithms as its fundamental building blocks, the approach to software metrics for object oriented programs must be different from the standard metrics set. Some metrics, such as lines of code and cyclomatic complexity, have become accepted as "standard" for traditional functional/ procedural programs, but for object-oriented, there are many proposed object oriented metrics in the literature. The question is, "Which object oriented metrics should a project use, and can any of the traditional metrics be adapted to the object oriented environment?"

Keywords: metrics, object-oriented software development, software reliability, cyclomatic complexity, error prediction, C++ programming language.

1 Relatia dintre probabilitatea de eroare si metricile OO

1.1. Metodologia de analiza

Variabila raspuns care se utilizeaza pentru a evalua metricile design-ului OO este binara. De exemplu, daca a fost detectata o clasa care cauzeaza probleme în timpul fazei de testare se recomanda folosirea metodei regresiei logistice pentru a analiza relatia dintre metrici si clasele predispuse la erori. Regresia

logistica este o tehnica de clasificare folosita în multe experimente stiintifice bazate pe estimari de probabilitate maxima [BARO1]. În acest caz, trebuie întreprinsa o analiza atenta care sa ofere o anumita certitudine referitoare la faptul ca trendul observat nu este doar rezultatul câtorva observatii, desi regresia logistica este considerata a fi mai robusta, pentru evidentiere, decât regresia realizata prin metoda celor mai mici patrute.

Tabelul 1. Analiza univariata – sumarul rezultatelor

Metrica	Coefficient	DY	p	R ²	Clase
WMC(1)	-0.022	98%	0.0607	0.007	Toate
WMC(2)	-0.086	92%	0.00035	0.024	Clase noi
WMC(3)	-0.027	103%	0.0656	0.0154	DB
WMC(4)	-0.0944	91%	0.0019	0.0467	UI
DIT(1)	-0.485	62%	0.0000	0.0648	Toate
DIT(2)	-0.868	42%	0.0000	0.1314	Clase noi
DIT(3)	-0.475	62%	0.043	0.0187	DB
DIT(4)	-0.29	75%	0.024	0.017	UI
RFC(1)	-0.085	92%	0.0000	0.0648	Toate
RFC(2)	-0.087	92%	0.0000	0.2477	Clase noi
RFC(3)	-0.077	93%	0.0000	0.188	DB
RFC(4)	-0.108	90%	0.0000	0.3624	UI
NOC(1)	3.3848	3000%	0.0000	0.1426	Toate
NOC(2)	3.62	3734%	0.0011	0.362	Clase noi
NOC(3)	2.05	777%	0.0000	0.0826	DB
CBO(1)	-0.142	87%	0.0000	0.068	Toate
CBO(2)	-0.079	92%	0.017	0.02	Clase noi
CBO(3)	-0.086	92%	0.006	0.034	DB
CBO(4)	-0.284	75%	0.0000	0.17	UI

În particular, a fost folosită regresia logistică univariată, pentru a evalua relația fiecărei metrici în condiții de izolare și predispunere la erori. Apoi, s-a utilizat regresia logistică multivariată pentru a evalua capacitatea predictivă a acelor metrici care au fost presupuse ca fiind suficient de semnificative în analiza

univariată (de exemplu, $p < 0.25$ este considerată a fi suficient de euristica).

Conform [CHK1], în forma sa cea mai simplă, un model de regresie logistic se bazează pe următoarea ecuație de relație (modelul de regresie logistică univariată este un caz particular al acesteia, unde apare doar o singură variabilă):

$$\log\left(\frac{p}{1-p}\right) = C_0 + C_1 X_1 + C_2 X_2 + \dots + C_n X_n \quad (1)$$

unde p este probabilitatea ca o eroare va fi găsită într-o clasă în timpul fazei de validare și valorile X_i exprimă metricile OO incluse ca și predictorii în model (numite covariante ale ecuației de regresie logistică). În două cazuri extreme, (de exemplu, atunci când o variabilă este: fie insignifiantă, fie diferențiază total clasele predispușe la erori), curba dintre p și orice X_i , (de exemplu, presupunând ca toate celelalte valori X_j sunt constante) aproximează o linie orizontală și respectiv o linie verticală. Între aceste cazuri extreme, curba se prezintă sub o formă flexibilă de S. Oricum, de vreme ce p este necunoscut, coeficienții

C_i vor putea fi estimați printr-o funcție de optimizare probabilă [HOS1]. Aceasta presupune ca toate observațiile sunt, din punct de vedere statistic, independente. În timpul construirii ecuațiilor de regresie, fiecare observație a fost "apreciată" în conformitate cu numărul de erori depistate în fiecare clasă.

Raționamentul folosit este următorul "fiecare depistare a unei erori este un eveniment independent" - clasele care nu conțin erori vor fi detectate și li se va asocia nivelul de "importantă (semnificativitate)" 1.

Tabelul 2. Analiza multivariată a metricilor de design OO

	Coeficient	P
Înterupere	3.13	0.0000
DIT	-0.50	0.0004
RFC	-0.11	0.0000
NOC	-2.01	0.0178
RFC	-0.13	0.0072
CBO	-0.238	0.0001
Origine	-1.84	0.0000

Tabelele 1 și 2 conțin rezultatele obținute, respectiv regresia univariată și multivariată în toate clasele examinate. Aceste rezultate sunt raportate la metricile care s-au dovedit a fi cele mai semnificative în toate proiectele de dezvoltare. Pentru fiecare metrică, sunt disponibile următoarele statistici:

- **Coeficienți** (prezentati în tabelele 1 și 2) exprimă coeficienții de regresie estimați. Cu cât valoarea absolută a coeficientului este mai mare, cu atât impactul variabilelor explicative asupra probabilității p , de detectare a erorilor dintr-o clasă, este mai mare;
- **DY** - (apare numai în tabelul 1) - se bazează pe noțiunea de *rata impară* [HOS1] și oferă o

evaluare a impactului metricii asupra variabilei răspuns. Mai precis, *rata impară* $Y(X)$ reprezintă raportul dintre probabilitatea de a avea o eroare și probabilitatea de a nu avea o eroare atunci când valoarea unei metrici este X . De exemplu, dacă pentru o valoare dată X , $Y(X)$ este 2, atunci este de două ori mai probabil ca o clasă să conțină erori decât ca ea să nu conțină erori. Valoarea lui $Y(X)$ este calculată ca medie a următoarei formule:

$$\Delta y = \frac{y^{(X+1)}}{y^{(X)}} \quad (2)$$

Deci, $Y(X)$ exprimă reducerea/creșterea ratei impare (exprimată ca procent în tabelul 2) atunci când valoarea X crește cu o unitate.

Aceasta este destinata pentru a furniza o imagine intuitiva asupra impactului pe care îl au variabilele explicative.

• *Semnificatia statistica p* (apare în tabelele 1 si 2) ofera o imagine mai detaliata asupra acuratetei estimarii coeficientilor. Furnizeaza informatii despre probabilitatea ca acel coeficient sa fie diferit de 0, în mod întâmplator. Din punct de vedere istoric un prag de semnificatie a lui $p = 0.05$ (o probabilitate de 5%) a fost adesea folosita pentru a determina daca o variabila explicativa a fost un predictor semnificativ. Oricum, alegerea unui nivel particular de semnificatie este, în ultima instanta, o decizie subiectiva si alte niveluri precum $p = 0.01$ sau $p = 0.1$ sunt valori obisnuite. De asemenea, cu cât nivelul de semnificatie este mai mare, cu atât creste deviatia standard a coeficientilor estimati si scade credibilitatea impactului calculat al variabilelor explicative. Testul de semnificatie se bazeaza pe un test al ratei de probabilitate [HOS1], folosit, în mod curent, în cadrul regresiei logistice.

Luând în considerare ecuatia (1), functia de probabilitate a setului de date de dimensiune D este:

$$L = \prod_{i=1}^D p(x_i) \quad (3)$$

unde:

$$p(x_i) = \frac{e^{(C_0 + C_1 x_{i1} + \dots + C_n x_{in}) \cdot Y_i}}{1 + e^{(C_0 + C_1 x_{i1} + \dots + C_n x_{in})}} \quad (4)$$

unde Y_i este asignat ca având valoarea 1 daca clasa nu contine nici o eroare si 0 în caz contrar. Vectorul n -dimensional X_i contine metricile de design OO caracterizate fiecare de cele D observatii. Deci $p(X_i)$ reprezinta probabilitatea estimata ca o clasa sa contina sau nu erori. Coeficientul care va maximiza functia de probabilitate va fi obtinut din estimarile coeficientului de regresie. Prin conventie matematica, $l = Ln[L]$ (logaritmul probabilitatii), este, în mod uzual maximizat.

O masura globala a gradului de adecvare care este folosita în modelele de regresie logistica este R^2 , definita statistic astfel [HOS1]:

$$R^2 = \frac{(l_0 - l_n)}{(l_0 - l_s)} \quad (5)$$

unde:

• l_0 este logaritmul functiei de probabilitate fara a folosi vreo covarianta (doar întreruperea);

• l_n este logaritmul probabilitatii modelului incluzând un numar de n metrici de design selectate ca si covariatii;

• l_s este logaritmul probabilitatii modelului saturat, unde Y_i , (0 sau 1) este înlocuit pentru fiecare probabilitate $p(X_i)$ din l . Logaritmul probabilitatii l_s este valoarea maxima care îi poate fi atribuita lui l .

Cu cât R^2 este mai mare, cu atât modelul este mai fidel. R^2 poate fi descris ca o masura a *proportiei de totala incertitudine* care îi este atribuita modelului cel mai adecvat. Oricum, în contrast cu R^2 pentru regresia bazata pe metoda celor mai mici patrate, o valoare mai mare a acestui parametru este destul de rar întâlnita pentru regresia logistica, deoarece valorile lui l_n sunt rareori aproximari ale valorilor lui l_s , de vreme ce valoarea calculata $p(X_i)$ pentru l_n se apropie rareori de 1.

1.2. Analiza univariata

Reprezinta o metoda de analiza a cator sase metrici de design OO, introduse de Chidamber si Kemerer în [CHK1], cu privire la probabilitatea de detectare a defectelor dintr-o clasa în timpul fazelor de testare. În cazul analizat acest lucru este echivalent, pentru modelul logistic, cu calcularea probabilitatii ca o singura eroare sa fie detectata într-o clasa.

• *Importanta (semnificatia) metodelor dintr-o clasa (WMC)* a fost prezentata (vezi tabelele 1 si 2) ca fiind oarecum semnificativa ($p = 0.06$). Pentru clasele noi si cele modificate în mod extensiv, precum si pentru clasele UI (interfata grafica si textuala cu utilizatorul), rezultatele sunt mult mai bune: $p = 0.0003$ si respectiv $p = 0.0001$. Dupa cum era de asteptat, cu cât WMC este mai mare cu atât mai mult creste probabilitatea detectarii erorilor. Aceste rezultate pot fi explicate prin urmatorul fapt: *complexitatea interna nu are un impact puternic daca clasa este refolosita cu mici modificari*. În acest caz, proprietatile

clasei interfata vor avea un impact mai semnificativ;

- *Adâncimea arborelui de mostenire al unei clase* (DIT) – asa cum rezulta din tabelele 1 si 2 este prezentata ca fiind extrem de semnificativa ($p = 0.0000$). Dupa cum era de asteptat, cu cât DIT este mai mare, cu atât probabilitatea de detectare a erorilor creste. Rezultatele sunt îmbunătățite din nou, (R^2 logistic creste de la 0.06 până la 0.13) atunci când sunt luate în considerare numai clasele noi si cele modificate în mod extensiv;

- *Raspunsul dintr-o clasa* (RFC) s-a demonstrat, pe baza datelor din tabelele 1 si 2 ca este extrem de importanta ($p = 0.0000$). Cu cât RFC este mai mare, cu atât creste probabilitatea detectării unei erori. Oricum, valoarea parametrului R^2 logistic a fost ajustata, în mod semnificativ, pentru clasele noi, pentru cele modificate în mod extensiv si pentru clasele de interfata cu utilizatorul - UI (de la 0.06 la 0.24 respectiv 0.36). Clasele UI prezinta o distributie care difera, în mod semnificativ, de cea a claselor DB: media si mediana sunt în mod semnificativ mai mari. Ca rezultat, aceasta poate întări impactul pe care îl are RFC atunci când se efectueaza analiza;

- *Numarul de succesori ai unei clase* (NOC) pare a fi foarte semnificativ (cu exceptia cazului claselor UI) dar tendinta observata (pe baza datelor din tabelele 1 si 2) este contrara cu ceea ce se asteapta. Cu cât NOC este mai mare, cu atât probabilitatea de detectare a erorii scade. Aceasta tendinta surprinzătoare poate fi explicata prin faptul ca majoritatea claselor nu au mai mult de un mostenitor, si ca re folosirea propriu-zisa a claselor este oarecum asociata cu o valoare mai mare pentru NOC. Deoarece sa demonstrat ca reutilizarea este un factor de influenta important al densitatii erorilor [BAS2], aceasta explica de ce clasele care au valori mai mari pentru NOC sunt mai puțin dispuse la erori. Cu toate acestea, exista o oarecare instabilitate printre subseturile de clase în legatura cu impactul metricii NOC asupra probabilitatii de detectare a unei erori într-o clasa (vezi $\Delta\Psi$ în tabelul 1). Acest lucru poate fi explicat, în parte, prin lipsa de variabilitate pe aceasta

scala de masurare (vezi distributiile din figura 1);

- *Lipsa de coeziune a metodelor* (LCOM) - în urma observatiilor efectuate s-a dovedit a fi lipsita de semnificatie pentru toate cazurile (motiv pentru care rezultatele nu sunt prezentate în tabelul 1), ceea ce era de asteptat în conditiile în care distributia LCOM indica o lipsa de varietate si doar câteva lucruri de importanta majora;

- *Cuplarea între clasele de obiecte* (CBO) - este semnificativa si cu un plus de particularitate pentru clasele UI ($p = 0.0000$ si $R^2 = 0.17$). Nu exista o explicatie satisfăcătoare pentru diferentele existente între modelul claselor UI si DB.

Ca o consecinta, unii coeficienti arata un impact foarte mic (de exemplu, $Y(X)$) atunci când sunt comparati cu ceilalti. Oricum, acesta nu este un criteriu de validare pentru a evalua utilitatea predictiva a acestor metrici. Mai important este faptul ca, exceptând metrica NOC, toate celelalte metrici par a avea un impact foarte stabil asupra diferitelor categorii de clase (DB, UI, etc.), iar acest lucru este încurajator pentru ca permite extragerea unei concluzii referitoare la faptul ca *tipurile de variabile sunt, în general, comparabile*. Daca ar fi fost luate în considerare diverse tipuri de defecte, în mod separat, rezultatele ar fi putut fi diferite.

1.3. Analiza multivariata

Metricile OO, prezentate anterior, pot fi folosite de la început în ciclul de viata pentru a construi modele predictive ale claselor dispuse la erori. Pentru a obtine un model optim, aceste metrici sunt incluse într-un model de regresie logistica multivariata [LIH1]. Oricum, doar metricile care îmbunătățesc, în mod semnificativ, puterea de predictie a modelului multivariat este recomandabil sa fie incluse într-un proces de selectie iterativ.

Un alt predictor semnificativ al predisunerii la erori este *nivelul de reutilizare al unei clase* (denumit "origine" în tabelul 2). Aceasta informatie este disponibila la finalul fazei de proiectare, atunci când au fost identificate clasele ce urmeaza a fi reutilizate si poate fi estimat volumul de schimbari necesar. Tabe-

lul 2 prezinta informatii referitoare la modelul multivariat calculat.

Folosind un astfel de model, pentru clasificare, rezultatele prezentate în tabelul 3 au fost obtinute prin utilizarea ca si prag de clasificare a valorii $p = 0.5$ pentru probabilitatea detectarii unei singure erori într-o clasa data (valoarea utilizata este recomandata în [BRO1] si [CHK1]). De exemplu: când $p > 0.5$, clasa va fi considerata ca având erori, iar în caz contrar ca fiind lipsita de erori. Dupa cum era de asteptat, clasele predictate ca eronate contin un numar mai mare de erori (250 erori în 48 clase) deoarece acele clase tind sa prezinta o mai buna acuratete în clasificare.

Analizând impactul utilizarii unui model de predictie prin ipoteze, cu scopul de a simpli-

fica modul de calcul, se observa ca procesul de "inspectare" a claselor prezinta o eficienta de aproape 100% în ceea ce priveste detectarea erorilor. Astfel, 80 de clase (predictate ca având erori), dintr-un total de 180, au fost "inspectate" si 48 de clase care contin erori, dintr-un total de 58, au fost detectate în timpul inspectiei. Luând în considerare contorizarea individuala a erorilor, se observa ca din numarul total de 258 de erori, 250 au fost detectate în timpul inspectiei. Aceste rezultate sustin ipoteza conform careia "modelul predictional ofera rezultate mai precise pentru clasele ce contin mai multe erori". Pentru a concluziona, rezultatele arata ca *metricile OO analizate sunt utile ca predictor de pre-dispunere la erori.*

Tabelul 3. Rezultatele clasificarii folosind metricile de design OO

Predictii		
Valori curente	Clase fara erori	Clase eronate
Clase fara erori	90	32
Clase eronate	10 (18)	48 (250)

În tabelul 3 valorile dispuse înaintea parantezelor, în coloanele din dreapta, reprezinta numarul de clase trecute în categoria celor eronate, iar valorile plasate în interiorul parantezelor exprima numarul de erori continute de aceste clase. Pentru a evalua acuratetea acestor metrici OO, apare ca interesanta compararea capacitatii lor de predictie cu una dintre metricile uzuale pentru cod, care poate fi obtinuta mai târziu în dezvoltarea ciclului de viata. Pentru efectuarea comparatiei au fost selectate, printr-o procedura de regresie logistica iterativa, trei metrici pentru cod din setul pus la dispozitie de instrumentul AMADEUS [AMA1].

Instrumentul AMADEUS ofera 35 metrici pentru cod dintre care cele mai importante sunt: linii de cod cu sau fara spatii, "secvente executabile", blocuri declarate, functii declarate, definitii de functii, apeluri de functii, complexitatea ciclomatica, declaratii iterati-

ve, maximul de adâncime si largime a claselor într-un fisier, numarul metodelor declarate, numarul mediu al metodelor si al definitiilor. Tabelul 4 prezinta rezultatele estimarilor parametrilor modelului logistic multivariat unde:

- *MaxStatNest* este nivelul maxim de imbricare a declaratiilor în cadrul unei clase;
- *FunctDef* este numarul declaratiilor de functii;
- *FunctCall* este numarul de apeluri ale functiilor.

Oricum, bazându-se pe întregul set de metrici pus la dispozitie de AMADEUS, si alte modele multivariate au scos la iveala rezultate similare din punct de vedere al acuratetei [AMA1]. Totusi, acest model este, în mod întâmplator, modelul rezultat din utilizarea unei proceduri standard de analiza regresiva logistica iterativa.

Tabelul 4. Analiza multivariata folosind metricile pentru cod

	Coefficient	P
Întreupere	0.39	0.0384
MaxStatNest	-0.286	0.0252
FunctDef	0.166	0.0010
FunctCall	-0.0277	0.0000

În plus, pentru a fi colectabile doar mai târziu în cadrul procesului, metricile pentru cod par a fi într-un fel mai "sarace" decât predictorii claselor predispușe la erori (vezi tabelul 5). În acest caz, 112 clase (estimate ca fiind clase cu erori) din 118 au fost cercetate, iar dintr-un total de 58 de clase au fost detectate 51 clase care contin erori. Dacă este luată în considerare contorizarea individuala a erorilor, 231 de erori dintr-un total de 268 au fost detectate în timpul procesului de inspectie.

Ar mai fi posibil de corectat încă trei clase cu erori (51 fata de 48), dar ar trebui cercetate cu 32 de clase mai mult (112 fata de 80). Asadar, metricile de design OO sunt mai buni predictorii ai claselor ce contin un numar mare de erori, deoarece permit detectarea unui numar de erori cu 19 mai mare decât s-a estimat initial (250 fata de 231). De aceea, predictiile bazate pe metricile pentru cod par a fi mai "sarace".

Tabelul 5. Rezultate ale clasificării bazată pe metricile pentru cod prezentate în tabelul 4

Predictate Actuale	Clase fara erori	Clase Eronate
Clase fara erori	61	61
Clase eronate	7 (37)	51 (231)

Tabelul 6. Clasificarea acuratetei bazată pe metricile OO și pentru cod prezentate în tabelele 2 și 4

Acuratetea Modelului	Metrici OO	Metrici pentru cod
Completitudine	88% (93%)	83% (86%)
Corectitudine	60% (92%)	45.5% (86%)

Tabelul 6 confirmă rezultatele experimentului prin prezentarea valorilor *corectitudinii* (procent al claselor estimate corect ca fiind purtătoare de erori) și *completitudinii* (procent al claselor eronate detectate). Valorile dintre paranteze prezintă corectitudinea predicțiilor și valorile completitudinii atunci când clasele sunt evaluate potrivit numărului de erori pe care îl contin.

2. Compararea rezultatelor studiului de caz cu alte experimente

Pentru a evalua rezultatele studiului de caz efectuat au fost analizate rezultatele obținute

de către alți specialiști din domeniu, în urma efectuării de experimente, și publicate în literatura de specialitate. Unele din cele mai cunoscute experimente, realizate pe o temă asemănătoare cu cea a prezentului studiu de caz, sunt cele ale lui Li și Henry [LIH1] și Briand [BRI1], care au încercat să valideze experimental metrici ale tehnologiei OO (vezi tabelul 8). În [BRI1] metrici folosiți pentru a măsura tipurile de date abstracte (ADT) coeziunea și cuplarea sunt propuse și validate, în mod experimental, ca predictorii ai tipurilor de date abstracte eronate.

Tabelul 7. Diferențe și asemănări între lucrările analizate

CRITERII	LUCRARI ANALIZATE		
	Briand	Li și Henry	Studiul de caz propus
<i>Setul de metrici</i>	Tipul de date abstracte, coeziunea și cuplarea	Metriicile Chidamber și Kemerer	Metriicile Chidamber și Kemerer
<i>Tipul componentelor</i>	Ada	Dialect OO al limbajului Ada	C++
<i>Variabila dependentă</i>	Apariția erorilor în tipurile de date abstracte Regresia logistică	Numărul de schimbări în cadrul componentelor Regresia prin metoda celor mai mici pătrate	Apariția erorilor în clasele C++ Regresia logistică
<i>Tehnica statistică</i>			

Comparând rezultatele acestui studiu de caz cu cele obținute de Briand [BRI1], au fost "descoperite" următoarele diferențe și asemănări. În studiul amintit anterior validarea experimentală a metricilor utilizate nu s-a efectuat pe programe realizate în conformitate cu tehnologia OO, în contextul mostenirii, însă modelul de validare utilizat este similar. În ambele cazuri, modelul statistic a fost construit pentru estimarea componentei (de exemplu: tipurile abstracte de date respectiv clase) și a predispușiei la erori (de exemplu: probabilitatea detectării de erori) prin utilizarea regresiei logistice multiple.

Pe de altă parte, Li și Henry [LIH1] au efectuat o validare a metricilor OO propuse de Chidamber și Kemerer [CHK1], studiind numărul de schimbări realizate în două sisteme comerciale implementate cu un dialect OO al limbajului Ada. Potrivit acestei lucrări, metricile OO propuse de Chidamber și Kemerer par a fi adecvate pentru predicția frecvenței cu care se efectuează modificările din clase, în timpul etapei de întreținere. Modelul realizat de ei este destinat efectuării de predicții referitoare la numărul de modificări dintr-o clasă, care au presupus ca este proporțional cu efortul necesar pentru realizarea modificărilor și este reprezentativ pentru nivelul de mentenanță al clasei. Li și Henry au folosit același set de metrici OO care a fost folosit și în studiul de caz pe care l-am propus. Au folosit, de asemenea, date referitoare la produse implementate într-un limbaj OO care oferă facilități precum: mostenirea multiplă și polimorfismul.

Studiul de caz efectuat a urmărit să evalueze dacă metricile OO propuse de Chidamber și Kemerer [CHK1] au fost predictorii utili ai claselor predispușe la erori. În alta ordine de idei, a fost folosită probabilitatea de detectare a erorilor ca variabilă dependentă a modelului statistic, aceeași variabilă fiind întâlnită și în lucrarea lui Li și Henry [LIH1]. În plus, Li și Henry au folosit regresia liniară multivariată bazată pe metoda celor mai mici pătrate pentru a construi un model predictiv, în timp ce în studiul propus am folosit regresia logistică (de exemplu: tehnică clasificării pentru variabilele binare dependente). Natura

variabilei dependente, utilizată în studiul realizat (de exemplu: apariția sau neapariția unei erori) m-a condus la folosirea regresiei logistice care este des întâlnită și în alte experimente științifice [HOS1].

3. Concluzii

În acest experiment, au fost colectate date despre defectele întâlnite în clasele orientate pe obiecte. Pe baza acestor date s-a verificat cât de mult este influențată predispușia la erori, de caracteristicile interne (de exemplu: dimensiunea și coeziunea) și externe (de exemplu: cuplarea) ale claselor OO. Din rezultatele prezentate mai sus, cinci din cele șase metrici propuse de Chidamber și Kemerer [CHK1], par a fi utile pentru a prezice clasele predispușe la erori în timpul fazelor primare ale ciclului de viață. Aceasta validare empirică demonstrează că **majoritatea acestor metrici pot fi folosite drept indicatori de calitate**. În plus, **multe dintre aceste metrici par a fi indicatori complementari care sunt relativ independenți fata de alții**. De asemenea, se poate spune că metricile OO propuse de Chidamber și Kemerer par a fi predictorii mai buni decât cel mai bun set de metrici "traditionale" pentru cod, care pot fi aplicate pe setul de date colectat, și care, în plus, pot fi colectate numai în fazele finale ale procesului de dezvoltare a sistemelor de programe OO.

Prin studiul de caz efectuat s-a urmărit obținerea unei mai bune înțelegeri a impactului strategiilor de proiectare OO (de exemplu: mostenirea simplă față de cea multiplă) asupra densității erorilor și a reprocesării. În acest studiu, din cauza "greutăților" întâlnite în procesul de colectare a datelor, nu s-a putut analiza capacitatea metricilor OO de a realiza predicții referitoare la reprocesare. Trebuie precizat că acest impediment ar putea fi depășit prin "îmbunătățirea" procesului de colectare a datelor, pentru a determina cât efort a fost depus pentru fiecare clasă în parte. Studiarea diferențelor dintre diversele limbaje OO, în concordanță cu definițiile metricilor și cu rezultatele experimentale, reprezintă un alt factor de interes care merita a fi menționat. Astfel, capacitățile de predicție a

predispunerii la erori ale setului de metrice, analizat în cadrul acestui experiment, pot fi diferite, depinzând de limbajul de programare utilizat. Pentru evaluarea acestor capabilitati setul metricilor de design OO trebuie validat în cadrul unor limbaje de programare OO (de exemplu: C++, Java, etc.).

Bibliografie

[AMA1] – Amadeus Software Research, "Getting Started with Amadeus", Amadeus Measurement System, <http://www.amadeus/system/uk/cs/docs.htm>
[BARO1] - Baron T. coordonator, "Calitate si fiabilitate - manual practic", vol. 1 si vol. 2, Editura Tehnica, Bucuresti, 1988;
[BRI1] - Briand L., Morasca S, Basili V., "Defining and Validating High-Level Design Metrics", CS-TR-3301, University of Maryland, College Park, MD, 20742, 1999

[CHK1] - Chidamber S.R., Kemerer C.F., "A Metrics Suite for Object-Oriented Design", IEEE Transactions on Software Engineering, 20(6), 1998, pp.476-493;

[DEV1] - Devanbu P., "GENOA/GENII - A Customizable Language and Front-End Independent Code Analyzer", in Proceedings of the 14th International Conference on Software Engineering, Melbourne, Australia, 1992, pp.234-261;

[HEL1] - Heller G., Valett J., Wild M., "Data Collection Procedure for the Software Engineering Laboratory (SEL) Database", SEL Series, 1998, SEL-98-002;

[HOS1] – Hosmer D., Lemeshow S., "Applied Logistic Regression", Wiley-Intersciences, 1999, pp.56-92

[LIH1] - Li W., Henry S., "Object-oriented Metrics that Predict Maintainability", Journal of System and Software, 23(2), 1998, pp. 111-122

[YOU1] - Young D.A. "Object-Oriented Programming with C++ and OSF/MOTIF", Reference Manual, Prentice-Hall, 1992;