

Evaluarea si compararea instrumentelor CASE bazate pe UML

Dan Laurentiu JISA

EWIR – Est West Informatique Roumanie, dan.jisa@ewir.ro

CASE tools (Computer Aided Software Engineering) represent those applications supporting analysts, designers, programmers, testing teams, to analyze, design, implement (at least partially) modify (expand), build, respectively tests software applications. This work outline some criteria for the evaluation of UML based CASE tools, and performs a comparative study of four CASE instruments based on these criteria. The goal of the work is to survey the instruments (RationalRose, MagicDraw, OpenTool and Microsoft Visio 2000) and to probe the criteria utility in order to choose the instrument that fit better to the specific of an organization and to the software programs developed in the organisation.

Keywords: UML, CASE, object-oriented.

Criterii de evaluare a instrumentelor CASE

Criteriile de apreciere ale instrumentelor pot fi împartite în: *criterii dependente de limbajul de modelare si criterii independente de limbajul de modelare* [CHI00]. În continuare vor fi analizate instrumentele Rational Rose 2001, Microsoft Visio 2000, OpenTool 3.1 si MagicDraw 5.0, luând în considerare criterii din ambele categorii. Instrumentele mentionate au fost selectate din multitudinea de instrumente/medii CASE existente avându-se în vedere, în principal, suportul oferit limbajului de modelare (UML), calitatea interfetei grafice (aceasta incluzând si suportul oferit navigarii prin model), limbajele de programare si tehnologiile pentru care se genereaza cod, platformele pe care pot functiona instrumentele (cu exceptia lui Visio 2000, celelalte instrumente sunt disponibile atât pentru platforme UNIX, cât si pentru platforme Microsoft).

a) Din categoria criteriilor dependente de limbaj:

–*Suportul oferit de instrument limbajului de modelare se materializeaza în numarul de concepte puse la dispozitia utilizatorului si în tipurile de diagrame ce pot fi construite* [CHI00];

–*Suportul pentru adnotari formale textuale se refera la posibilitatea utilizarii a doua formalisme diferite (unul grafic si altul textual) si reprezinta una din trasaturile definitorii ale limbajului unificat de modelare* [CHI00];

–*Pastrarea consistentei informatiilor între diferite diagrame - modificarea efectuata într-o*

anumita vedere trebuie sa se reflecte adecvat în celelalte vederi [CHI00].

b) Din categoria criteriilor independente de limbajul de modelare, au fost considerate :

–*Suportul oferit pentru navigarea prin model - instrumentul trebuie sa includa un browser care sa permita navigarea simpla a modelului, gasirea rapida si eficienta a entitatilor proiectului* [CHI00];

–*Generarea codului - acest aspect este important, în acest fel se economiseste timp, se economisesc bani si exista certitudinea conformitatii între specificare si implementare* [CHI00];

–*Reverse Engineering si Round Trip Engineering – sunt importante în cazul produselor care au fost realizate fara ajutorul instrumentelor CASE; uneori acestea nu dispun de “documente” de proiectare adecvate sau cel puțin actualizate* [CHI00];

–*Suport pentru modelarea aplicatiilor bazate pe componente – este important ca instrumentele CASE sa aiba notiuni despre principalele standarde: Enterprise Java Beans (EJB), COM/DCOM, CORBA etc.;*

–*Suport pentru modelarea datelor – UML poate modela datele prin intermediul diagramelor de clase. Un aspect important îl reprezinta generarea de cod DDL (un set de comenzi SQL necesare pentru descrierea si crearea structurii bazei de date în conformitate cu proiectul);*

–*Reutilizabilitatea - se refera la suportul acordat de instrument pentru utilizarea în faza de proiectare a unor componente de biblioteci*

(clase, componente, sabloane de proiectare sau chiar proiecte – fiecare cu particularitatile lor) [CHI00];

–*Support pentru sabloane de proiectare (design patterns)* – se refera la suportul pentru adaugarea unui sablon de proiectare în model;

–*Documentatia produsa pentru proiect* - se refera la informatiile care pot fi obtinute folosind informatia existenta în diagrame si repository. Trebuie avuta în vedere posibilitatea salvarii diagramelor într-un format recunoscut de browser-ele WWW (jpeg sau gif) [CHI00];

–*Schimbul cu alte instrumente* - se refera la posibilitatea citirii proiectelor realizate cu alte instrumente sau/si la posibilitatea salvarii într-un alt format [CHI00];

–*Integrarea cu instrumente de dezvoltare* – pentru a putea fi folosite facilitatile de forward si reverse engineering este necesar sa se foloseasca un instrument de dezvoltare suportat de instrumentul CASE;

–*Supportul acordat muncii în echipa* - are doua conotatii: prima tine cont de faptul ca la unul si acelasi proiect lucreaza mai multi realizatori care nu actioneaza concurent asupra aceluiasi model; a doua ia în considerare accesul concurent la acelasi model, caz în care trebuie gesti-

onat un mecanism de acces mai complex [CHI00].

Comparatie între patru instrumente CASE bazate pe UML

Dupa cum s-a mentionat mai sus, instrumentele CASE studiate sunt:

–Rational Rose 2001 - este un produs al companiei Rational Software Corp., disponibil atât pentru platforme Windows cât si Unix;

–Microsoft Visio 2000 – este un produs Microsoft, disponibil numai pentru platforme Windows;

–OpenTool 3.1 – un produs TNI-VALIOSYS, disponibil pentru platforme Windows si UNIX (Solaris 2, HP-UX, AIX, Linux);

–MagicDraw 5.0 – un produs No Magic Inc. disponibil, teoretic, pe orice platforma pe care poate rula o masina virtuala de java versiunea 1.3, dar practic în documentatie se precizeaza ca majoritatea testelor s-au facut pe platforme Windows si Linux.

Tabelele 1 si 2 arata conceptele si notatiile UML suportate de fiecare dintre instrumentele considerate.

Tabelul 1

Concepte folosite de UML	Rational Rose 2001	Visio 2000	OpenTool	MagicDraw
Caz de utilizare	√	√	√	√
Caz de utilizare abstract	√	√		√
Actor	√	√	√	√
Actor abstract	√	√	√	√
Colaborare				√
Pachet	√	√	√	√
Clasa	√	√	√	√
Clasa abstracta	√	√	√	√
Clasa radacina		√		√
Clasa frunza		√		√
Clasa activa	√	√		√
Clasa utilitara	√	√	√	√
Clasa parametrizabila	√	√		√
Metaclasa	√	√		√
Atribut	√	√	√	√
Atribut calificator	√	√		
Operatie	√	√	√	√
Operatie abstracta	√	√	√	√
Operatie frunza				√
Vizibilitate operatie/atribut	√	√	√	√
Obiect	√	√	√	√

Interfata	√	√		√
Asociere	√	√	√	√
Clasa a asocierii	√	√		√
Agregare	√	√	√	√
Generalizare	√	√	√	√
Relatie de dependenta	√	√	√	√
Relatie de realizare	√	√		√
Legatura	√	√	√	
Mesaj	√	√	√	√
Stare	√	√	√	√
Stare initiala/finala	√	√	√	√
Stare compozita	√	√	√	√
Istoric al starilor imbricate	√	√	√	√
Activitate	√	√	√	√
Actiune	√	√	√	√
Eveniment	√	√	√	√
Tranzitie	√	√	√	√
Componenta	√	√		√
Nod	√	√		√

Tabelul 2

Diagrame folosite de UML	Rational Rose 2001	Visio 2000	OpenTool	MagicDraw
Diagrama cazurilor de utilizare	√	√	√	√
Diagrama de clase	√	√	√	√
Diagrama de pachete	√	√		√
Diagrama de secventa	√	√	√	√
Diagrama de colaborare	√	√	√	√
Diagrama de stare	√	√	√	√
Diagrama de activitati	√	√		√
Diagrama de componente	√	√	√	√
Diagrama de amplasament	√	√	√	√

Verificarea consistentei se face atât în cadrul aceleiasi diagrame (de exemplu o relatie de asociere trebuie sa uneasca doua clase, o tranzitie trebuie sa aiba o stare sursa si una destinatie etc.), cât si între diagrame (adaugarea unui mesaj într-o diagrama de interactiune trebuie sa se reflecte în diagrama de clase etc.). Instrumentele studiate asigura consistenta în cadrul aceleiasi diagrame si între diagramele de clase si cele de interactiune.

Suportul oferit pentru navigarea prin model face parte din a doua categorie de criterii. Toate cele patru instrumente au o fereastră de tip *tree-control*, fereastră ce permite navigarea prin model. În cazul Rational Rose, Visio 2000 si MagicDraw fiecare element din diagrame (reprezentate în fereastră din partea dreapta) se reflecta în *tree-control* din partea stânga.

Tabelul 3 arata limbajele si tehnologiile pentru care este generat cod de fiecare din cele trei instrumente. De asemenea se poate observa

suportul oferit pentru modelare aplicatiilor bazate pe componente, precum si pentru modelarea bazelor de date. În plus fata de celelalte instrumente, Visio 2000 permite construirea de diagrame entitate-asociere

Rational Rose 2001 permite ingineria inversa pentru limbajele de programare Java/J2EE si C++.

Instrumentul Visio 2000 permite ingineria inversa a codului în UML, pentru proiectele dezvoltate în Visual C++ 6.0, Visual J++ 6.0 si Visual Basic 6.0. Procesul genereaza automat un model în *tree view-ul* navigatorului UML, dupa care se poate crea o diagrama a structurii statice prin operatii de "drag and drop" a claselor în fereastră de editare a diagramelor. Instrumentul OpenTool permite ingineria inversa pentru cod Java. Instrumentul MagicDraw permite ingineria inversa din proiecte dezvoltate în Java/EJB, C++, CORBA IDL, DDL.

Tabelul 3

Genereaza cod	Rational Rose 2001	Visio 2000	OpenTool	MagicDraw
C++	√		√	√
Java	√		√	√
Smaltalk			√	
VB	√			
CORBA IDL	√			√
XML DTD	√			
Ada 83	√			
Ada 95	√			
DDL	√		√	√

În ce priveste suportul acordat reutilizării, Rational Rose ofera următoarele facilitati: *framework wizard add-in* si *type library importer*. În primul caz este permisa dezvoltarea si refolosirea de cadre care sa fie incluse (sub forma unor simboluri) în modelul în curs de dezvoltare. În cel de al doilea caz este permis importul unei componente COM. De asemenea, instrumentul permite integrarea cu Microsoft Repository.

Microsoft Visio 2000 nu ofera posibilitatea refolosirii modelelor dezvoltate folosind UML (permite integrarea cu Microsoft Repository numai pentru modelarea bazelor de date). MagicDraw permite exportul în fisiere xml a

unor module dintr-un proiect, în vederea reutilizării lor în alte proiecte (se vor importa în proiectul dorit). OpenTool permite importul unei entitati (pachet, căsa, operatie a unei căse etc.) dintr-un pachet extern. Atât Rational Rose cât si MagicDraw ofera suport pentru integrarea în model a sabloanelor de proiectare descrise de Erich Gamma în cartea *Design Patterns, Elements of reusable Object Oriented Software*.

Tabelul 4 arata metodele utilizate de fiecare dintre instrumentele considerate pentru a genera documentatia proiectelor dezvoltate (generare de documentatie sau exportul diagramelor în diverse formate).

Tabelul 4

Documentatie	Rational Rose 2001	Visio 2000	OpenTool	MagicDraw
HTML	√	√	√	√
GIF		√		
JPEG		√		√
PostScript		√	√	
SGML			√	
SVG				√
Bitmap		√		
Transfer prin clipboard	√	√	√	√

Pentru schimbul cu alte instrumente Rational Rose foloseste formatul petal (formatul companiei Rational). De asemenea este disponibil un *add-in* (<http://www.rational.com/products/rose/forms/xmisupport/xmisupport.jsp>) care, o data instalat, permite schimbul de modele cu alte instrumente, folosind formatul XMI. În ce priveste suportul acordat muncii în echipa, instrumentul permite integrarea cu Microsoft

SourceSafe (când se foloseste un mediu Microsoft), realizându-se astfel atât un control al versiunilor cât si accesul concurrent la acelasi model.

OpenTool foloseste pentru schimbul cu alte instrumente formatul petal (permite importul modelelor în format petal) si formatul XMI. Suportul acordat muncii în echipa consta în împartirea modelului în pachete si subpachete,

pentru fiecare dintre acestea putându-se bloca accesul sau acorda numai drepturi read-only. MagicDraw permite schimbul cu alte instrumente folosind formatul XML. Suportul oferit muncii în echipa este oferit prin serverul MagicDraw UML Teamwork Server.

Integrarea cu instrumentele de dezvoltare a fost studiată pentru mediile Windows. Astfel, Rational Rose permite integrarea cu Microsoft Visual C++, Microsoft Visual Basic, Forte for Java. Microsoft Visio 2000 se integrează (poate face reverse engineering) cu Visual C++ 6.0, Visual J++ 6.0 și Visual Basic 6.0. MagicDraw permite integrarea cu Forte for Java și Borland JBuilder.

Concluzii

În urma studiului comparativ a celor patru instrumente CASE bazate pe UML se pot desprinde următoarele concluzii:

–Conceptele și notațiile UML sunt foarte bine acoperite de Rational Rose, Visio 2000 și MagicDraw (figura 1)

–Rational Rose și MagicDraw sunt potrivite pentru modelarea aplicațiilor bazate pe componente, ele permitând generarea de cod pentru COM, CORBA sau Enterprise Java Beans;

–Rational Rose și MagicDraw oferă suport pentru folosirea sabloanelor de proiectare în modelele dezvoltate cu aceste instrumente.

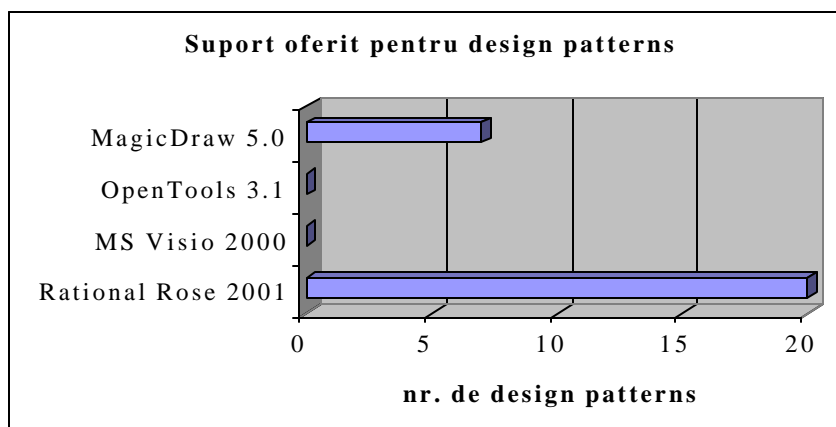


Fig.1. Suport oferit pentru design patterns

–Este necesară folosirea unei metodologii care să sprijine întreprinderile producătoare de software atunci când doresc să achiziționeze un astfel de instrument, metodologie care să aibă în vedere importanța pe care fiecare criteriu o are pentru organizația producătoare de software (este posibil ca organizația să nu acorde aceeași atenție tuturor criteriilor).

Bibliografie

[MFU98] Mihalca R., Fabian C., Uta A., Simion F., *Analiza și proiectare orientate obiect. Instrumente de tip CASE.*, Societatea Autonomă de informatică, București 1998.
 [BRJ99] Booch G., Rumbaugh J., Jacobson I., *The Unified Modelling Language*, Addison/Wesley 1999.

[RRO] *Technical Support – Rational Software*, <http://www.rational.com/support/>

[MDR] *UML diagramming, OO Software Modelling, Source Code Engineering Tool MagicDraw UML from No Magic*, www.MagicDraw.com

[TNI] TNI – VALIOSYS, www.tni-valiosys.com/?p=industry&s=aerospace&ss=opentool&type=overview

[CHI00] Chiorean D., *Instrumente CASE UML. Obiective, evoluție, criterii de apreciere și perspective*, PC Report, Ianuarie 2000.

[MDN01] Microsoft Developer Network Library – October 2001