Revista Română de Interacţiune Om-Calculator 1 (2008), 73-84

Abordări în evaluarea automată a utilizabilităţii. Studiu comparativ

Adriana-Mihaela Guran[1], Grigoreta Sofia Cojocar[2]

[1]Universitatea Babeş-Bolyai, Facultatea de Matematică şi Informatică, Cluj-Napoca
Str. Mihail Kogălniceanu nr. 1
E-mail: adriana@cs.ubbcluj.ro, grigo@cs.ubbcluj.ro

Rezumat. Utilizabilitatea este un subiect de larg interes pentru dezvoltatorii de software, astfel încât din ce în ce mai multe companii sunt preocupate de acest aspect al produselor software pe care le dezvoltă. In această lucrare sunt prezentate trei abordări folosite în evaluarea utilizabilităţii sistemelor informatice şi o analiză comparativă a acestora.

Cuvinte cheie: interacţiune om-calculator, utilizabilitate, AOP, agenţi, accesibilitate

1. Introducere

Importanţa crescândă a utilizabilităţii sistemelor informatice, în special în ultima decadă, se datorează importanţei pe care aceasta o are în succesul sau eşecul unui produs informatic. Acest fapt a determinat numeroase companii să realizeze sesiuni de evaluare a utilizabilităţii. De-a lungul timpului s-au formulat numeroase definiţii ale utilizabilităţii, câteva fiind prezentate în cele ce urmează.

Shakel defineşte utilizabilitatea ca fiind capacitatea unui sistem de a fi folosit eficient şi uşor de o categorie specifică de utilizatori, care au avut parte de o instruire specifică şi asistenţă pentru utilizarea produsului, pentru a îndeplini un domeniu specific de sarcini într-un cadru specific. Definiţia e operaţionalizată prin intermediul a patru criterii: eficienţă, uşurinţă de învăţare, flexibilitate şi atitudine (Shakel, 1991).

Preece (Preece et al, 1990) priveşte utilizabilitatea ca măsura uşurinţei cu care un sistem poate fi învăţat sau utilizat, siguranţa, eficienţa şi eficacitatea sa, şi atitudinea utilizatorilor faţă de sistem.

Standardul ISO 9241-11 defineşte utilizabilitatea ca fiind: ”măsura în care un produs poate fi folosit de utilizatori specifici pentru a atinge scopuri specifice cu eficienţă, eficacitate şi satisfacţie într-un context specific de

utilizare" (***b, 1991).

Eficiența se referă la raportul dintre resursele consumate şi acuratețea şi completitudinea cu care utilizatorii îndeplinesc scopuri.

Eficacitatea se referă la acuratețea şi completitudinea cu care utilizatorii îndeplinesc sarcini specificate.

Satisfacția este o măsură subiectivă referitoare la confortul şi acceptarea produsului de către utilizatorii finali.

Dix (Dix et al, 2004) oferă o definiție operaționalizată a utilizabilității, oferind proiectanților un punct de reper atunci când se proiectează sisteme pentru utilizabilitate. Din perspectiva sa, utilizabilitatea e asigurată de trei factori:

- uşurință de învățare - se referă la uşurința cu care utilizatorii noi pot să înceapă interacțiunea efectivă şi să atingă performanță maximă;
- flexibilitate - se referă la multitudinea canalelor prin care utilizatorul şi sistemul schimbă informație;
- robustețe - se referă la nivelul sprijinului care îi este acordat utilizatorului în îndeplinirea sarcinilor.

Pornind de la aceste perspective asupra utilizabilității, au fost definite metrici ale utilizabilității, instrumente de real folos în activitatea de evaluare (Welie, 2001).

2. Evaluarea utilizabilității sistemelor informatice

Evaluarea utilizabilității poate să aibă loc pe parcursul proiectării unui sistem sau după dezvoltarea acestuia. Ea se poate realiza prin testări cu utilizatorii, rezultatele acestor testări fiind cele mai relevante dar în acelaşi timp şi cele mai costisitoare. Prin folosirea scenariilor se pot obține date despre numărul erorilor sau viteza de execuție, care sunt indicatori ai utilizabilității. Chestionarele de utilizabilitate precum SUMI (***a, 1996) sau QUIS (Chin et al, 1988) oferă informații standardizate despre utilizabilitatea sistemelor informatice. Metodele care oferă estimări ale utilizabilității sistemelor încă din faza de proiectare reprezintă alegerea preferată de dezvoltatorii de sisteme.

Din această categorie de metode fac parte inspectarea utilizabilității sau

evaluarea respectării şabloanelor de proiectare a interfețelor utilizator. Inspectarea utilizabilității este o revizuire a sistemului pe baza unui set de recomandări. Revizuirea este condusă de un grup de experți familiarizați cu conceptele utilizabilității în proiectare. Experții se concentrează asupra unei liste de aspecte din proiectare care s-au dovedit problematice pentru utilizatori. Recomandările de utilizabilitate sunt derivate din studii de HCI, ergonomie, design grafic, proiectarea informației şi psihologie cognitivă. Aspectele evaluate pot fi: limbajul folosit în interacțiune, măsura în care se apelează la aducerea aminte a utilizatorului şi modul în care sistemul oferă feedback utilizatorilor. Inspectarea utilizabilității poate fi realizată prin diverse metode precum:

- revizuiri cognitive - revizuirile cognitive pot fi efectuate în orice etapă a dezvoltării unui produs, folosind fie un document aparținând proiectării conceptuale, un prototip sau produsul final. Pe baza scopurilor utilizatorilor un grup de evaluatori parcurg sarcina pas cu pas, evaluând la fiecare pas cât este de dificil pentru utilizator să identifice şi să opereze asupra elementului din interfață care este cel mai relevant pentru subscopul curent şi cât de clar este feedback-ul pe care sistemul îl oferă utilizatorilor. Scopul reprezintă o stare care se doreşte a fi atinsă folosind sistemul interactiv. Revizuirile cognitive iau în considerare procesul de gândire care determină luarea deciziilor, precum încărcarea memoriei şi abilitatea de a raționa. Această abordare este folosită atunci când se încearcă explorarea utilizabilității pentru utilizatorii care folosesc pentru prima dată sistemul sau utilizatorii ocazionali ai sistemului, adică acei utilizatori care învață prin explorare;
- revizuiri pluralistice - foloseşte întâlniri de grupuri în care utilizatori, dezvoltatori şi specialişti în factori umani parcurg un scenariu şi discută fiecare element al dialogului;
- inspectarea consistenței - un grup de proiectanți care participă la realizarea unor produse similare inspectează interfața pentru a vedea dacă permite realizarea funcționalității în acelaşi mod ca şi produsele proprii;
- inspectarea standardelor - un expert într-un standard verifică

respectarea standardului de către o interfață;

- inspectarea caracteristicilor sistemului - se listează secvențele folosite în realizarea unei sarcini, se caută secvențele prea lungi, paşii confuzi, acțiunile care nu sunt selectate natural de către utilizator, paşii care necesită cunoştințe extensive sau expertiză pentru a realiza un scop;
- evaluarea euristică - este o tehnică de determinare a problemelor de utilizabilitate cu o interfață. Un număr mic de evaluatori experimentați (3-5) inspectează separat o interfață folosind un set de euristici. Apoi, combină rezultatele evaluărilor şi fac o clasificare a problemelor de utilizabilitate după gravitatea acestora. Sunt general acceptate cele 10 euristici ale lui Nielsen: vizibilitatea stării sistemului, potrivirea dintre sistem şi lumea reală, controlul utilizatorului şi libertatea, consistență şi standarde, prevenirea erorilor, recunoaştere (nu reamintire), flexibilitate şi eficiență în utilizare, design minimalist şi estetic, ajutorul utilizatorilor să recunoască, diagnosticheze şi să recupereze erorile, help şi documentare (Nielsen, 1993);
- focus groups - focus grupurile reprezintă o metodă foarte eficientă de a culege reacții de la utilizatori şi de a aprecia reacțiile inițiale la proiectare, dar şi pentru a identifica diferențele dintre aşteptările utilizatorilor şi implementarea actuală.

Aceste metode necesită existența unor experți în utilizabilitate, fapt care poate fi de asemenea costisitor. În acest context, automatizarea (parțială a) procesului de evaluare a utilizabilității poate reprezenta o soluție satisfăcătoare pentru orice dezvoltator de software.

2.1 Abordări în automatizarea evaluării utilizabilității

Pentru evaluarea utilizabilității unui sistem informatic sunt efectuați, în mod general, următorii paşi (Ivory, 2001):

- colectarea de date - etapă în care sunt adunate informații despre modul în care este folosit sistemul informatic;
- analiza datelor - etapă care cuprinde metode statistice de grupare a datelor şi de identificare a problemelor;

- propunerea de sugestii de îmbunătățire a sistemului software suspus evaluării.

Tipic, la un test de utilizabilitate evaluatorul propune un scenariu de utilizare a sistemului informatic. Sarcinile incluse în scenariu au scopul de a testa anumite aspecte ale interfeței utilizator. În timpul testului de utilizabilitate specialistul în utilizabilitate observă modul în care utilizatorul execută sarcinile, comportamentul său şi comentariile pe care le face. La sfârşitul testului are loc o discuție privind modul de realizare a sarcinilor şi gradul de satisfacție a utilizatorului în raport cu sistemul folosit. Primele două activități din cele mai sus amintite pot fi supuse unui proces de automatizare. Avantajele pe care le aduce automatizarea evaluării utilizabilității sunt: reducerea necesarului de resurse umane, reducerea necesarului de expertiză umană, completitudine şi posibilitatea comparării rezultatelor testelor de utilizabilitate. Pentru a analiza interacțiunea utilizator-sistem este necesară identificarea unor "urme" utilizator în interacțiunea cu sistemul.

Jurnalizarea şi analiza evenimentelor care apar la nivelul interfeței utilizator au fost recunoscute a fi surse valoroase de determinare a problemelor de utilizabilitate. Prin jurnalizarea evenimentelor din interfața utilizator se pot obține informații precise despre ce face utilizatorul, ce date foloseşte, ce funcționalități foloseşte, în ce ordine execută acțiunile. Astfel, metrici ale utilizabilității precum timpul de execuție a unei sarcini, frecvența de folosire a unei funcționalități sau numărul de erori pot fi calculate din fişierele de jurnalizare. Există şi alte modalități de evaluare a acestor metrici (înregistrarea video a sesiunii de testare, notarea manuală), dar folosind jurnalizarea evenimentelor acest lucru se poate realiza automat. Datele înregistrate în fişierele de jurnalizare au un format specific care poate fi transformat în alte formate, operație urmată de supunerea datelor spre analiză. Desigur, jurnalizarea automată a evenimentelor din interfața utilizator are şi neajunsuri, unul din acestea fiind imposibilitatea surprinderii reacțiilor utilizatorului. În mod obişnuit jurnalizarea se realizează prin inserarea unor linii de cod în cadrul codului aplicației pentru a se surprinde elementele de interes din cadrul interacțiunii. Această abordare necesită o foarte bună cunoaştere a structurii aplicației şi acces la codul sursă al aplicației. Decizia asupra locațiilor în care să se facă instrumentarea trebuie să aparțină unui expert în utilizabilitate care să colaboreze cu dezvoltatori ai

sistemului.

2.1.1 Utilizarea AOP pentru captarea evenimentelor din interacțiunea utilizator-sistem

În (Tarța şi Moldovan, 2006a) şi (Moldovan şi Tarța, 2006b) s-a propus o abordare referitoare la jurnalizarea evenimentelor din interfața utilizator folosind o paradigmă de programare recent dezvoltată, şi anume programarea orientată pe aspecte (AOP - Aspect Oriented Programming). Această paradigmă este recomandată tocmai în astfel de situații în care trebuie să se trateze situații de funcționalități transversale. O funcționalitate transversală este o caracteristică a unui sistem informatic a cărei implementare este împrăştiată prin codul sursă al aplicației (exemplu: jurnalizarea, securizarea). Pentru implementarea unor astfel de funcționalități AOP introduce patru noțiuni noi (Kiczales, 1997): join-point, pointcut, advice, aspect.

Un aspect este unitatea de modularizare a AOP. Aspectele sunt integrate într-un sistem folosind un instrument special numit *weaver*. Există extensii AOP pentru limbaje de programare bine-cunoscute precum Java, C++ sau C#.

În această abordare s-au scris aspecte pentru diferitele metrici care s-a dorit a fi evaluate. Avantajul pe care îl aduce abordarea propusă este acela că modulele care realizează procesele de evaluare a utilizabilității sunt separate de codul sursă al aplicației. În plus, ataşarea acestor module la aplicație sau modificările apărute în modulele de evaluare a utilizabilității nu necesită modificarea codului aplicației, ci doar modificarea aspectelor şi recompilarea sistemului.

2.1.2 Utilizarea bibliotecilor pentru aplicații de tip suport pentru captarea evenimentelor din interacțiunea utilizator-sistem

O metodă alternativă de jurnalizare a informațiilor necesare evaluării utilizabilității aplicațiilor implementate folosind Java este utilizarea Java Accesibility API. Acest API este destinat dezvoltării de aplicații pentru persoane cu dizabilități. Pentru a scrie fişiere de jurnalizare folosind Java Accessibility API e necesară scrierea unei clase care implementează toate interfețele asociate evenimentelor de care suntem interesați. Clasa

SwingEventMonitor din Java Accessibility API va fi adăugată ca listener la toate evenimentele de interes. Maşina virtuală Java trebuie configurată înainte de rularea aplicației aflată în evaluare pentru a notifica clasa noastră asupra evenimentelor de interes. Această configurare trebuie făcută o singură dată şi este disponibilă pentru toate aplicațiile aflate în evaluare.

2.1.3 O abordare bazată pe agenți în evaluarea automată a utilizabilității

In (Tarța et. al, 2006) s-a propus o abordare de evaluare a utilizabilității folosind agenți şi programarea orientată pe aspecte. Un *agent* este o entitate care îşi percepe mediul prin intermediul unor senzori şi acționează asupra mediului său prin acțiuni (Russell şi Norvig, 2003). Un agent inteligent este un agent cu un set de cunoştințe inițiale care are capacitatea de a învăța. Una din sarcinile unui agent este aceea de a asista utilizatorul, de a realiza sarcini în locul utilizatorului sau de a-l învăța pe utilizator ce trebuie să facă. Un agent este caracterizat prin:

- arhitectura (comportament) - acțiunile efectuate după orice secvență;
- program - funcționalitatea internă a agentului.

Evaluarea utilizabilității porneşte în demersul propus de la compararea modelului sarcinilor construit de utilizator cu modelul sarcinilor aparținând proiectantului sistemului. In situația ideală, aceste modele sunt identice, caz în care nu vor apărea probleme de utilizabilitate majore. In majoritatea cazurilor însă, există diferențe considerabile între modelul conceptual al proiectantului şi modelul conceptual al utilizatorului. Trebuie să remarcăm faptul că modelul conceptual al utilizatorului se construieşte pe baza interacțiunii cu sistemul şi pe baza experiențelor de operare cu alte sisteme. In abordarea noastră, modelul conceptual al proiectantului se concretizează într-un arbore al sarcinilor descris cu ajutorul unui instrument de analiza sarcinilor şi salvat într-un fişier XML. Rolul agentului este de a monitoriza interacțiunea utilizatorului cu sistemul şi de a reconstrui modelul sarcinilor aparținând utilizatorului. Agentul care evaluează utilizabilitatea, denumit TAMO, este situat astfel într-un mediu software. Acesta foloseşte AOP pentru a culege informații despre mediul său. Arhitectura generală a acestui sistem este prezentată în Figura 1.

Componentele sistemului sunt descrise în cele ce urmează:

- Aplicația software (SA) - e un sistem interactiv căruia i se evaluează utilizabilitatea. Aplicația software are asociat un arbore al sarcinilor (TT) dezvoltat de către proiectantul aplicației;
- Utilizatorul (U) - o persoană care foloseşte SA pentru a realiza un set stabilit de sarcini (un scenariu);
- Agentul (TAMO) - este un agent software care are rolul de a monitoriza acțiunile utilizatorului şi de a compara acțiunile efectuate de utilizator cu TT pentru a-l asista pe utilizatorul U. Cunoştințele inițiale ale agentului constau în arborele sarcinilor TT asociat aplicației interactive SA. Percepția agentului TAMO se constituie din acțiunile pe care utilizatorul U le efectuează în timp ce foloseşte SA. Partea de program a agentului TAMO se concretizează prin compararea a doi arbori ai sarcinilor (corespunzători modelului sarcinilor proiectantului şi modelului sarcinilor utilizator). In cazul general, acțiunea agentului constă în transmiterea rezultatului comparării modulului de învățare. TAMO poate fi îmbunătățit prin înzestrarea lui cu capacitatea de învățare, situație în care el devine un agent de interfață (Maes, 1994), un fel de asistent personal al utilizatorului;
- Modulul de învățare (LM) - este un modul software care învață comportamentul utilizatorului şi transmite feedback evaluatorului interfeței utilizator. Acesta se poate concretiza într-un agent inteligent. Modulul AOP este folosit pentru a capta evenimentele utilizator care sunt folosite de TAMO pentru a reconstrui modelul sarcinilor utilizator (UTT). Cunoştințele inițiale ale agentului sunt stocate într-un fişier XML. Modulul de evaluare al agentului, care specifică comportamentul acestuia, compară TT şi UTT pentru a verifica dacă UTT este un subarbore al lui TT. In versiunea curentă a sistemului propus, TAMO salvează rezultatele comparațiilor efectuate pe un dispozitiv de stocare.

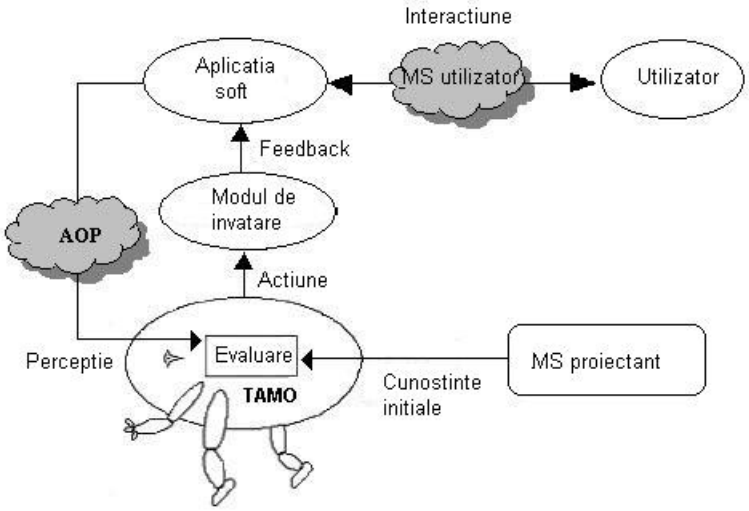

Figura 1. Arhitectura sistemului bazat pe agenți

3. Analiza comparativă a abordărilor

Pentru compararea abordărilor prezentate s-a realizat un studiu de caz. S-a luat în considerare o aplicație pentru gestiunea bugetului unei familii. Funcționalitatea oferită de aplicație include adăugarea unui venit, adăugarea unei cheltuieli, vizualizarea bilanțului pentru o anumită perioadă de timp. Participanții la test au trebuit să realizeze următoarele sarcini:

- adăugarea unui venit la o dată specificată (data1);
- înregistrarea unei cheltuieli la o dată specificată (data2);
- vizualizarea economiilor dintre data1 şi data2;
- vizualizarea cheltuielilor dintre data1 şi data2;
- vizualizarea tuturor veniturilor.

Pentru evaluarea utilizabilității am parcurs următorii paşi:

- înregistrarea fiecărei erori împreună cu momentul apariției;
- preluarea unei capturi de ecran la apariția fiecărei erori pentru a

vizualiza datele introduse de utilizator;

- calcularea frecvenței pentru fiecare tip de eroare;
- calcularea timpului de execuție a fiecărei sarcini;
- calcularea numărului de sarcini încheiate cu succes, abandonate (se începe execuția sarcinii) sau eşuate (se începe execuția sarcinii, dar apar erori).

Momentul începerii execuției unei sarcini e considerat acel moment în care utilizatorul selectează butonul corespunzător sarcinii din bara de instrumente sau selectează din meniu opțiunea corespunzătoare.

Aplicația a fost dezvoltată folosind Java, iar ca extensie AOP am folosit AspectJ.

În urma efectuării studiului de caz, s-au observat următoarele:

- Folosind AOP sau agenți, metrici de utilizabilitate precum timpul de execuție, numărul erorilor, starea execuției sarcinilor se pot determina automat, în timpul interacțiunii utilizatorului cu sistemul.
- Java Accessibility API realizează automat jurnalizarea, dar calculul metricilor trebuie să se facă ulterior interacțiunii şi folosind date din contextul interacțiunii.
- AOP permite efectuarea unor acțiuni la apariția unor evenimente (erori, apăsări de taste, selecții de opțiuni), precum preluarea unor capturi de ecran cu starea execuției, lucru imposibil de realizat folosind Java Accessibility API.
- Folosind AOP şi agenți tipul erorii poate fi stabilit cu exactitate în momentul apariției acesteia, iar folosind Java Accessibility API acest lucru presupune analiza evenimentelor şi contextelor in care a apărut.
- Codul aplicației supuse evaluării rămâne neschimbat în toate abordările, diferențele apar la integrarea modulului de evaluare a utilizabilității. La AOP e necesară scrierea de cod care să trateze situațiile de interes din momentul interacțiunii (relative unor evenimente din interfața grafică) şi recompilarea aplicației folosind weaver-ul asociat, pe când la Java Accessibility API e necesară doar configurarea maşinii virtuale Java.
- Abordarea bazată pe aspecte poate fi folosită doar pentru tehnologii care au extensii pentru AOP, iar abordarea bazată pe accesibilitate

poate fi folosită doar pentru platforme care oferă support pentru persoane cu dizabilități.

- Pentru captarea evenimentelor din interfața grafică, dezvoltatorul modulelor AOP trebuie să cunoască fiecare metodă din cadrul aplicației care tratează evenimente.
- În abordarea bazată pe tehnici de accesibilitate, codul sursă corespunzător modulului de captare a datelor de interacțiune poate fi refolosit pentru orice aplicație Java (nu este dependent de sistemul supus evaluării);

4. Concluzii

In acest articol au fost prezentate trei abordări în evaluarea automată a utilizabilității sistemelor interactive folosind metrici de utilizabilitate. Prima abordare propune utilizarea Programării Orientate pe Aspecte (AOP) pentru preluarea informațiilor despre interacțiune şi calculul unor metrici de utilizabilitate. Ca o alternativă pentru preluarea informațiilor despre interacțiune a fost prezentată utilizarea bibliotecilor pentru dezvoltarea de aplicații pentru persoane cu dizabilități, care oferă facilități în ceea ce priveşte captarea evenimentelor din interfața grafică. Cercetările au fost extinse de la calculul unor metrici de utilizabilitate spre identificarea paşilor din interacțiune care provoacă probleme de utilizabilitate. Abordarea propusă în acest sens se bazează pe agenți inteligenți cu rol de monitorizare a interacțiunii şi de identificare a problemelor de utilizabilitate prin compararea modelului conceptual al utilizatorului cu modelul sarcinilor utilizator (reconstruit din interacțiune). In viitor ne propunem înzestrarea agentului dezvoltat cu capacitatea de a învăța, determinarea unor măsuri de calitate ale rezultatelor evaluării şi extinderea abordării evaluării utilizabilității spre aspecte calitative (evaluarea satisfacției utilizatorilor).

Referințe

Alexandersson, B., Event Logging in Usability Testing, Master Thesis, Chalmers University Of Technology, Goteborg, Sweden, 2002.

Dix, A., Finlay, J., Abowd, G., and Russell, B.. *Human-Computer Interaction*. Prentice Hall, 2004.

Card, S. Moran, T. and Newell, A. *The Psychology of Human-Computer Interaction*. Cariere. Lawrence Erlbaum Associates, 1983.

Chin, J. P., Diehl, V. A., and Norman, K. L., *Development of an instrument measuring user satisfaction of the human-computer interface*. In CHI '88: Proceedings of the SIGCHI conference on Human factors in computing systems, pages 213–218, New York, NY, USA, 1988. ACM Press

Ivory, M. Y. and Hearst, M. A., *The state of the art in automating usability evaluation of user interfaces*, ACM Computing Surveys, vol. 33, no. 4, pp. 470–516, 2001.

Kiczales, G., Lamping, J., Menhdhekar, A., Maeda, C., Lopes, C. Loingtier, and Irwin, J.-M., *Aspect-Oriented Programming, in Proceedings European Conference on Object-Oriented Programming*. Springer-Verlag, 1997, vol. 1241, pp. 220–242.

Maes, P., *Social Interface Agents: Acquiring Competence by Learning from Users and Other Agents*. In O. Etzioni, editor, Softwareware Agents — Papers from the 1994 Spring Symposium (Technical Report SS-94-03, pages 71–78. AAAI Press, 1994.

Moldovan, G. S. and Tarţa. A. M., *Developing an Usability Evaluation Module Using AOP*. In International Conference on Computers, Communications & Control, ICCCC 2006, pages 320–325, Felix Spa, România, 2006 (a).

Moldovan, G. S., and Tarţa, A. M., *A Comparison of Using AOP and Java Accessibility for Usability Evaluation*. In CD Proceedings of 4th European Conference on Intelligent Systems and Technologies, ECIT 2006, Iaşi, România, 2006 (b).

Nielsen, J., *Usability Engineering*. Academic Press, 1993.

Preece, J., Benyon, D., Davies, G., Keller G., and Rogers, Y. *A Guide to Usability*, 1990.

Russell, S. and Norvig, P., *Artificial Intelligence - A Modern Approach*. Prentice-Hall,Inc., 2003.

Shakel, D., *Usability - Context, Framework, Definition, Design and Evaluation*. Cambridge, University Press, 1991.

Tarţa, A. M., and Moldovan, G. S., *Automatic Usability Evaluation using AOP*. In Proceedings of 2006 IEEE-TTTC International Conference on Automation, Quality and Testing, Robotics, volume TOME II, pages 84–89, Cluj-Napoca, România, 2006.

Tarţa, A. M., Moldovan, G. S. and Serban, G.. *An Agent Based User Interface Evaluation Using Aspect Oriented Programming Techniques*. In ICAM5 - Fifth International Conference on Applied Mathematics, pages 151–158, Baia-Mare, România, 2006.

van Welie, M., *Task-based User Interface Design*. PhD thesis, Vrije Universiteit Amsterdam, 2001.

***a. SUMI Softwareware Usability Measurement Inventory, 1996. http://www.ucc.ie/hfrg/questionnaires/sumi/.

***b. ISO9214-11 Ergonomic Requirements for office Work with VDT's – Guidance on Usability, 1991.