

PROIECTARE ORIENTATĂ SPRE DOMENIU ȘI DEZVOLTARE ORIENTATĂ SPRE MASHUP BAZATE PE OPEN SOURCE JAVA METAFRAMEWORK PENTRU DEZVOLTAREA DE SOFTWARE WEB PRAGMATICĂ, FIABILĂ ȘI SIGURĂ:

PREZENTARE GENERALĂ[*]



<http://www.ict-romulus.eu>, <http://romulus.ici.ro>

Rezumat: Această lucrare constituie o prezentare a proiectului european Romulus. Proiectul oferă o metodologie adecvată pentru dezvoltarea de aplicații bazate pe conceptele de tip metaframework-uri și pentru a îmbunătăți metaframework-ul Roma. Această metodologie se bazează pe dezvoltarea iterativă, proiectare axată pe domeniu (Domeniu Driven Design), dezvoltare bazată pe teste, obiective software, divizate în Aspect și Best Practices (cele mai bune practici).

Cuvinte Cheie: metaframework, proiectare axată pe domeniu, Rich Client Application, Mashup-uri (de date, Web, portal, întreprindere), testare automată.

Abstract: This paper presents the European project Romulus. The project provides a suitable agile methodology to develop applications with the Meta Framework concepts and improve the Roma Meta Framework itself. This methodology is based on Iterative development, Domain Driven Design, Test Driven Development, Soft goals, division by Aspect and Best Practices.

Keywords: metaframework, Domeniu Driven Design, Rich Client Application, Mashups (data, Web, portal, enterprise), automat testing.

1. Introducere

Dezvoltarea de software destinat Web-ului este unul dintre domeniile cele mai active în dezvoltarea de software în Europa; Java Enterprise Edition este opțiunea preferată pentru un milion de europeni, cu peste 38,6% mai mult decât alte software-uri similare, conform informațiilor furnizate de Evans Data. Dezvoltarea de software destinat Web-ului nu este încă o zonă matură, care a produs numeroase tehnologii și framework-uri noi. Inginerii de software nu sunt capabili să stăpânească toate aceste tehnologii, care afectează în mod serios productivitatea dezvoltatorilor de aplicații.

Apariția unei noi paradigme de dezvoltare software pentru aplicații web nu se bazează numai pe Java, cum ar fi Ruby on Rails; tehnici moderne de aplicare a mashup-ului au pus la îndoială abordarea tradițională de dezvoltare de software web, și au arătat creșteri relevante în productivitate și fiabilitate. Cu toate acestea, adoptarea pe scară largă a soft-ului Java [Java] în companiile europene cere un răspuns Java adecvat pentru dezvoltarea de aplicații Web.

Proiectul Romulus are ca scop să contribuie la promovarea unei noi paradigme Open Source pentru dezvoltarea de sisteme destinate Web utilizând tehnologia Java. Proiectul s-a ocupat de cercetarea privind proiectarea orientată pe domeniu pentru dezvoltarea de aplicații Java și oferă un metaframework Java de tip Open Source pentru dezvoltarea de aplicații web <http://www.ict-romulus.eu>.

2. Consorțiul Romulus

Consortiul este format din: **Informatica GESFOR** S.A Spania, **ASSET Data** S.R.L Italia, **LIFERAY GMBH** Germania, **UPM - Universidad Politécnica de Madrid** Spania, **IMOLA Informatica** S.R.L, Italia, **DERI - National University of Ireland** Irlanda, **ICI – National Institute for Research and Development in Informatics**, Romania.

Romulus a reunit competențele complementare din industrie și mediul academic. Gesfor contribuie cu expertiza sa în securitate, contribuind la OWASP (Open Web Application Security Project) și dezvoltare Web. ASSETDATA aduce expertiza sa în dezvoltarea metodologiei AGILE și în standardizarea Java. LIFERAY aduce expertiza sa în domeniul tehnologiei PORTAL prin produsul Liferay Portal. Imola contribuie cu expertiza sa în integrarea sistemelor de întreprindere și standardizării sale în cadrul comunității Java. Déri oferă expertiză în tehnologia semantic web aplicată ingineriei software. UPM și ICI contribuie cu expertiza lor în cercetarea cu privire la noile metode de dezvoltare software.

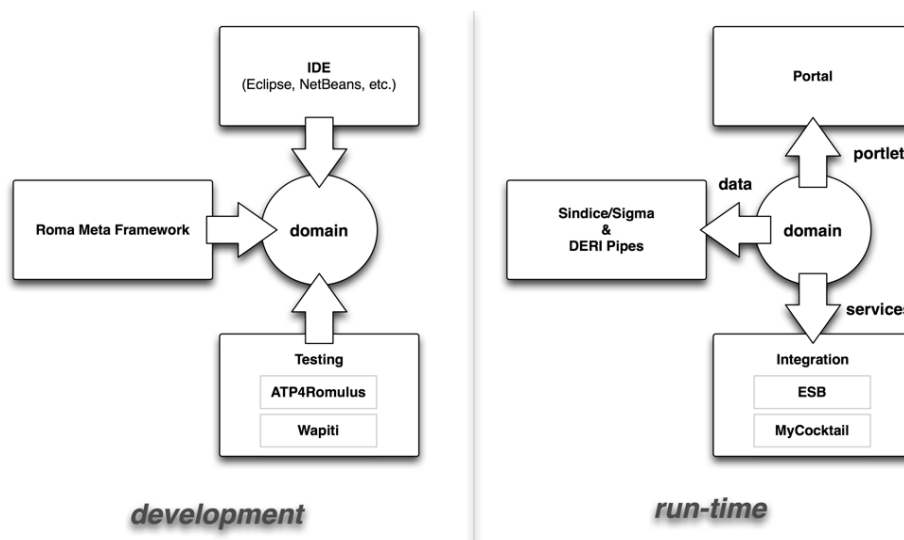
3. Obiective științifice și tehnice

3.1 Metaframework și proiectare bazată pe domeniu (Domain Driven Design)

Romulus oferă o metodologie adecvată **AGILE** pentru a dezvolta aplicații bazate pe conceptele de tip metaframework-uri și pentru a îmbunătăți metaframework-ul Roma. Această metodologie se bazează pe dezvoltarea iterativă, proiectare bazată pe domeniu (Domeniu Driven Design), dezvoltare bazată pe teste, obiective software, divizate în Aspect și Best Practices (cele mai bune practici).

Îmbunătățirile principale pentru metaframework-ul Roma au fost: elaborarea unei arhitecturi pentru un nou plug-in, un wizards (expert) pentru consolă, pentru a verifica și instala actualizările în mod transparent, crearea IDE4Romulus (noul plug-in pentru IDE Eclipse) și integrarea Aspectului **Logging** (Conectare).

Cel mai important modul dezvoltat în cadrul Romulus este **Janiculum**, o implementare alternativă pentru modulul Echo2 View (Vizualizare). Janiculum nu mai are limitările modulului Echo2, permițând o personalizare la nivel de componentă și respectarea constrângerilor de Accesibilitate W3C; aplicațiile anterioare sunt 100% portabile în spiritul Metaframework-ului.



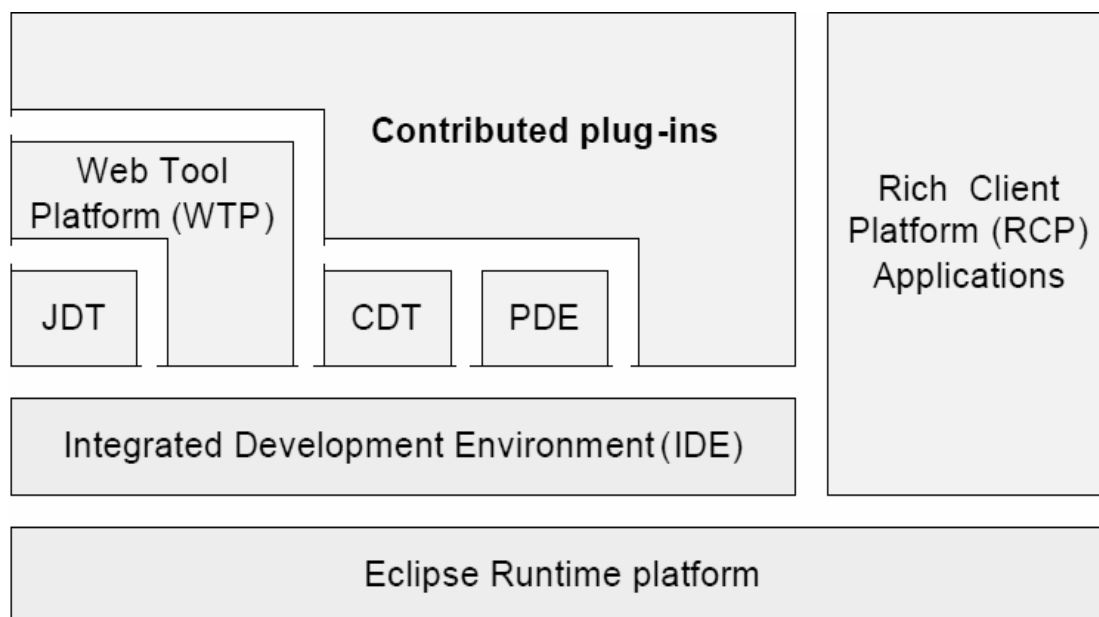
3.2 Roma Meta framework

În fiecare zi un nou instrument sau framework apare pe piață. Folosirea unei tehnologii înseamnă “o căsătorie” cu ea, de foarte multe ori pentru totdeauna. Ce se întâmplă dacă se dorește schimbarea SGBD-ului utilizat sau motorul care asigură gestionarea bazei de date pentru obiecte?

Sau un alt framework web... Multe tehnologii nu respectă standardele, astfel încât costul migrației este imens. Adesea este mai ieftin să fie rescris de la zero.

Roma este “meta” framework. Folosirea framework-ului Roma face, ca aplicația să devină independentă de tehnologia utilizată. Roma [Roma] furnizează interfețe un comportament foarte generic numit “Aspect”. “Aspect”- ele cuprind cazurile de utilizare cele mai frecvente. Folosirea instrumentului Aspect în cadrul framework-ului permite generarea unui cod curat față de tehnologie.

Deci, de multe ori se observă că aplicațiile au fost destul de similare. Tehnologiile sunt orizontale și se pot utiliza aceleași tehnologii în contexte foarte diferite. Ce este cu adevărat schimbat este **Domeniul**. Domeniul este valoarea reală a aplicației, deoarece este analizat, studiat și partajat cu clientul. Se începe prin definirea entităților / claselor specifice domeniului, care se pot scrie folosind n-IDE-uri la alegere sau instrumentul preferat UML, generator de clasă. Odată ce modelul este bine documentat, Roma [RomaURL] face restul, adică prezentarea, persistența obiectelor, internaționalizare, etc. Acestea pot fi autodeterminate, la analiza modelului domeniului.



În plus, proiectul Romulus a realizat o integrare cu produsul Open Source Tevere Flow. Aceasta este un motor pentru fluxul de lucru (workflow engine) complet tranzacțional având integrat un editor Web. Tevere Flow suportă șabloanele cele mai utilizate în fluxul de lucru (workflow), integrat în ROMA în Workflow Aspect prin intermediul Aspect Service. Nevoia de a avea un Semantic Aspect a apărut pentru a suporta integrarea de servicii de Mashup-uri pentru Date și Întreprinderi și de a integra instrumente semantice. O implementare de bază a fost dezvoltată pentru a exporta domeniul ca RDF doar prin adnotarea claselor interesate.

3.3 Dezvoltarea orientată pe Mashup-uri

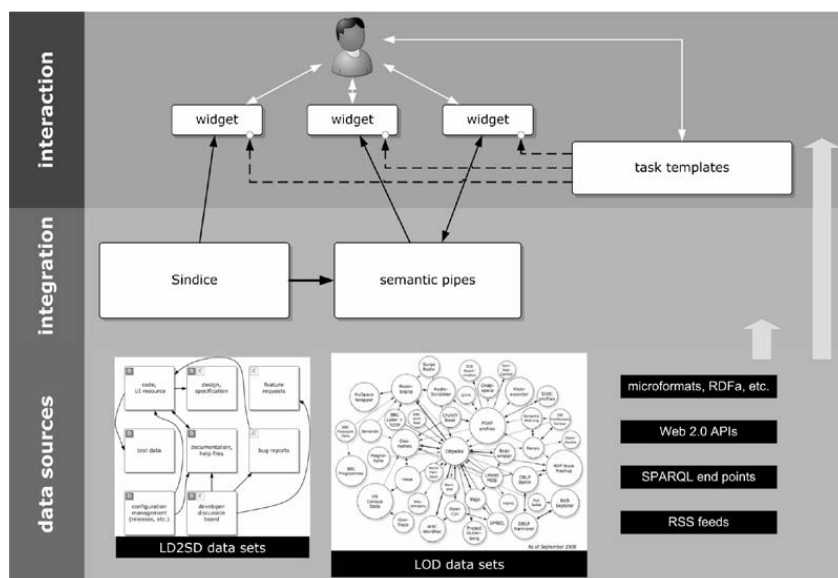
Cercetările din proiectul Romulus privesc know-how-ul de dezvoltare a cererilor Web - acestea pot fi „grăbite” datorită reutilizării serviciilor și a componentelor existente, precum și metodologiile aferente.

Proiectul a realizat cercetări pe mai multe tipuri de mashup-uri (servicii web Mashup [AawURL], mashup-uri de date de nivel [AmdURL], mashup întreprinderi [BwoURL] și mashup-uri portal [BEAAL]) pentru a integra servicii externe, sisteme de tip enterprise, surse de date sau componente de portal.

În ceea ce privește **mashup-urile la nivel de date**, au fost aplicate metode pentru exportarea și importarea de date structurate din urmărirea de bug-uri (bug tracking) de software prin

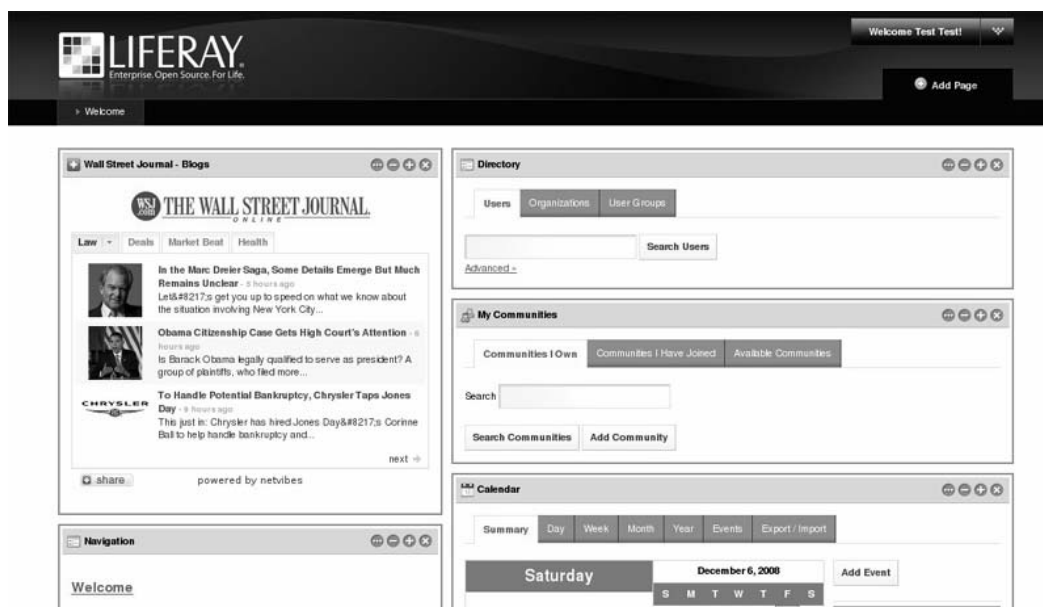
intermediul datelor legate și au fost dezvoltate widget-uri diferite pentru a lucra cu aceste date.

În plus, a fost efectuată o cercetare în agregarea de date folosind conducte de atribuire semantice (pipes) DĒRI.



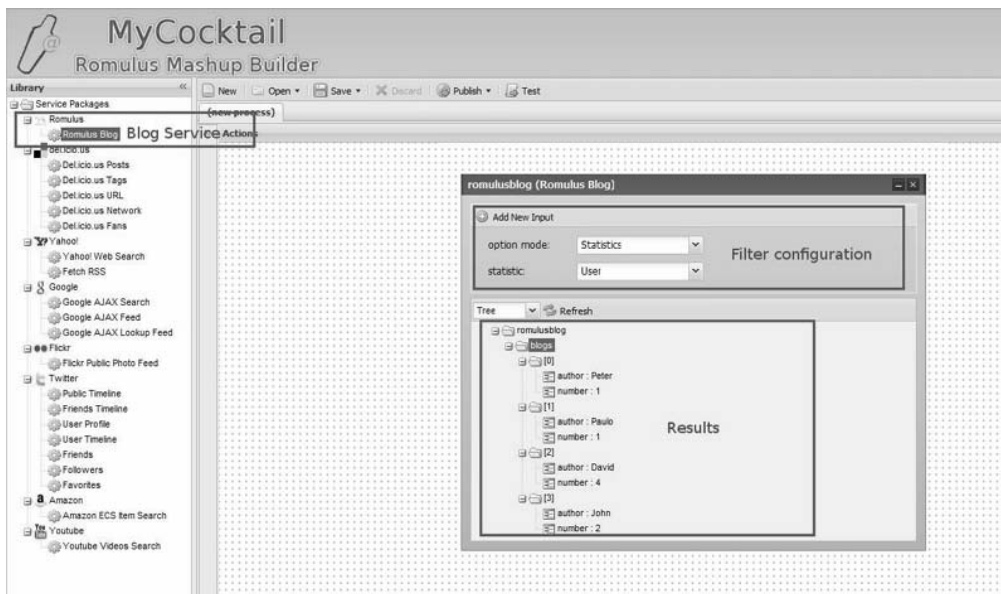
Pentru **mashup de portal**, au fost urmate două abordări pentru a atinge obiectivul de mashup al funcționalității pe care o oferă un portal. Prima abordare face accesibilă funcționalitatea portalului din alte platforme, cum ar fi Facebook sau Netvibes. Toate dezvoltările sunt integrate în Liferay și sunt disponibile integrat pentru orice utilizator. A doua abordare oferă posibilitatea de a construi noi aplicații care combină părți specifice din aplicații existente.

Au fost analizate diferite alternative pentru compoziția din fragmente și, în plus, a fost dezvoltată o bibliotecă JavaScript ușor de utilizat, care să permită exploatarea acestora într-o manieră simplă.

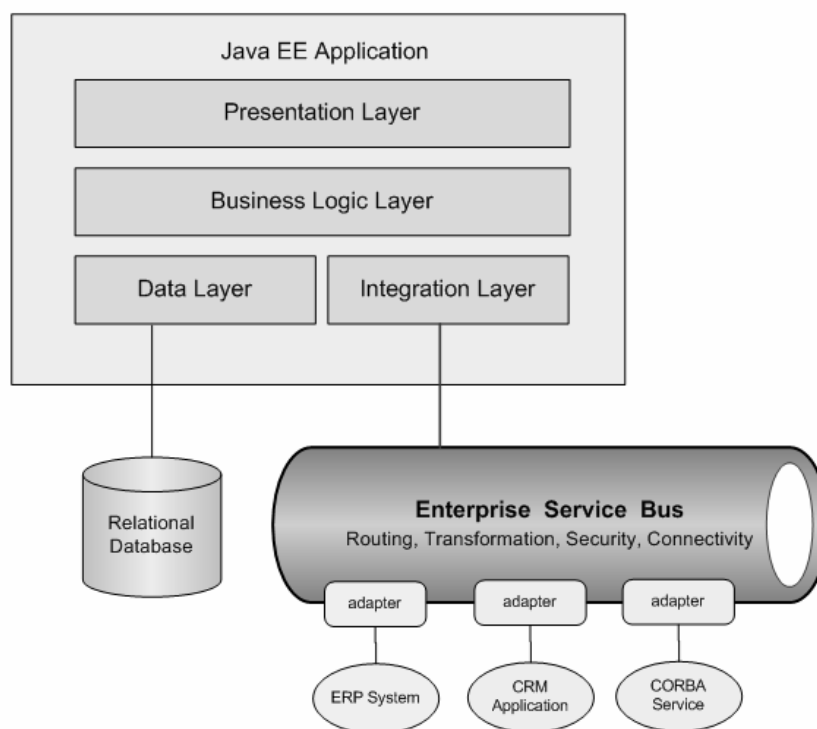


În ceea ce privește integrarea **mashup-urilor pentru servicii Web**, au fost luate în considerare două abordări: utilizarea adnotărilor în interiorul codului de acces la serviciile externe, pe de o parte, și crearea acestor mashup-uri cu un instrument grafic pe de altă parte, **MyCocktail, Romulus Mashup Builder**.

<http://www.ict-romulus.eu/web/mycocktail>

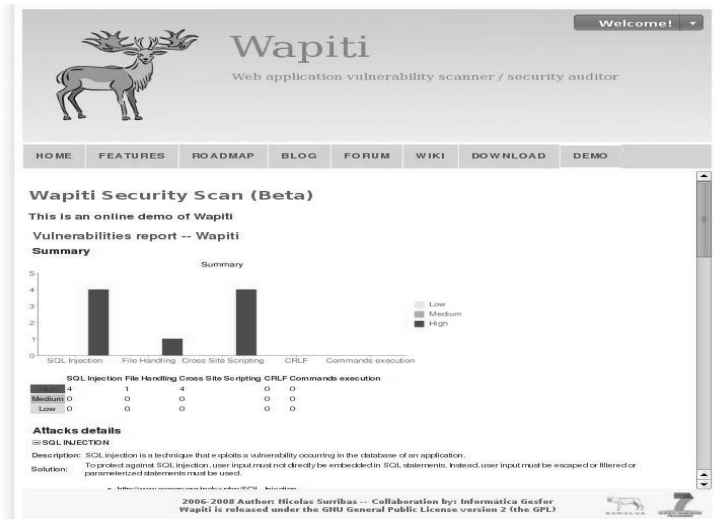


În cele din urmă, pentru integrarea **mashup-urilor pentru întreprinderi**, evoluția de la Web la mashup-uri de întreprindere a fost descrisă și a fost introdusă o referință pentru arhitectura Romulus. Această arhitectură se bazează pe componenta ESB [EM0208], în timp ce mashup-urile de integrare se bazează pe standardul BPEL și pe o extindere a editorului BPEL prevăzute în NetBeans IDE.



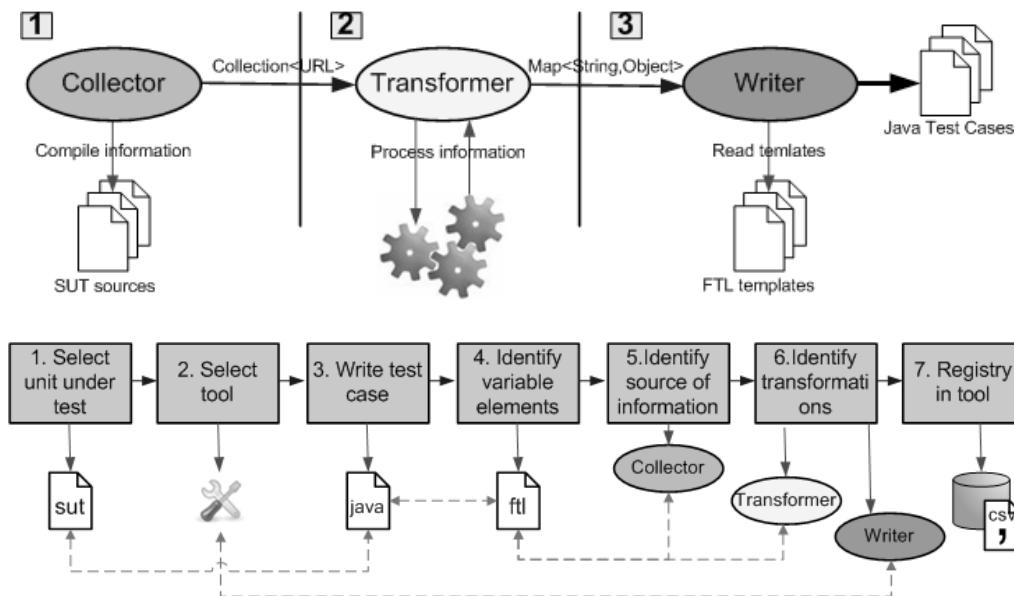
4. Obiective soft pentru calitate

Obiectivele soft, cum ar fi securitatea sau scalabilitatea, nu sunt de obicei luate în considerare de la bun început în procesul de dezvoltare. Obiectivele soft sunt, de obicei, legate de cerințe calitative (nefuncționale) și nu au un criteriu clar pentru satisfacerea lor, nici nu pot fi alocate unui mic subset de elemente din arhitectură. Romulus a realizat cercetări privind securitatea, trasabilitatea, fiabilitatea și performanța soft precum și integrarea lor în procesul de dezvoltare, bazată pe identificarea obiectivelor-aspectelor soft și utilizarea adnotărilor pentru aceste aspecte în



În domeniul securității, activitatea a fost realizată pentru a evalua modul în care MetaFramework ROMA poate îmbunătăți securitatea sa. Un set de adnotări de securitate au fost propuse ca rezultat al cercetărilor efectuate pe mai multe framework-uri **RAD** (Rapid Application Development). De asemenea, mai multe contribuții au fost făcute la instrumentul de securitate Wapiti, un instrument de testare a securității de tip “black box”, inclusiv îmbunătățiri de performanță, refacere de cod pentru extensibilitate ușoară, generator de raport și îmbunătățiri la unele detectoare de atac.

Pentru a asigura calitatea software-ului, a fost dezvoltat **ATP4Romulus**, un instrument de testare pentru aplicații Romulus. Acest instrument permite efectuarea de teste automate ale aplicațiilor generate folosind Roma Framework și efectuează mai multe teste cu privire la diferite aspecte cum ar fi **CRUD, View și i18n**.



5. Aplicații client-side versus server-side

În prezent, aplicațiile se deplasează de la tehnologii centrate server-side la tehnologiile centrate client-side și este necesar să se cerceteze echilibrul corect între client și tehnologiile server în funcție de cerințe. În plus, Web2.0 promovează un rol activ pentru utilizatori, care ar trebui să fie inclus în elaborarea aplicațiilor personalizabile. Romulus a realizat cercetări în această direcție pentru a satisface diferite tipuri de cerințe pentru interfețe. Au fost furnizate mai

multe opțiuni pentru dezvoltatori și utilizatori (cu abilități diferite) pentru a simplifica procesul de dezvoltare de aplicații.

Dezvoltarea de Portlet-uri, folosind limbaje de tip scripting, reduce barierele pentru dezvoltatori diferiți, în timp ce crește productivitatea și se menține accesul deplin la API-urile existente. Este necesară doar crearea unui descriptor de portlet și scrierea logicii în limbajul de scripting ales, asigurând mecanismele pentru integrarea aplicațiilor.

Dincolo de faptul că tehnologiile de scripting dau posibilitatea de a crea aplicații client-side, scrierea de cod va genera vizualizarea aplicației. Cum nu mai este necesară implementarea, productivitatea dezvoltatorilor este crescută.

Aplicațiile pot fi personalizate, prin scrierea de cod, care va pune în aplicare anumite funcții, pentru a răspunde nevoilor clienților. Ca un exemplu, poate fi dezvoltat un formular în care utilizatorul poate să își definească propriile sale funcții de validare complexe, după dorință (validarea chiar unui câmp împotriva altora). Utilizatori calificați (cu permisiuni corespunzătoare) pot actualiza codul JavaScript și îl pot executa atât pe client cât și pe server, cu o singură definiție.

Portlet-uri pot fi aranjate împreună și se comportă în comun, schimbul de informații între ele, oferind, de exemplu, navigarea wiki mai ușoară sau vizualizarea de conținut cu etichete.

6. Demonstratori verticali

Pentru a verifica aplicabilitatea tehnologiilor Romulus în contexte reale și pentru a demonstra modul în care acestea pot îmbunătăți considerabil productivitatea în dezvoltarea Web, au fost creați câțiva demonstratori: Musiteca, Cornelius și Scrooge.

Cornelius este o aplicație Web, ușor de personalizat, creată pentru a gestiona sarcinile specifice oricărei societăți, cum ar fi managementul de proiect, urmărirea realizării sarcinilor, planificarea resurselor umane și a costurilor. Scrisă în Java, bazată de Roma Meta Framework, aplicația utilizează tehnologii elaborate în cadrul consorțiului Romulus.

Cornelius Management de Proiect v1.0

Cornelius este o aplicație ușor adaptabilă pentru a manageria sarcinile uzuale ale oricărei companii, sarcini de tipul management de proiect, monitorizare de cerințe, planificare și costuri de resurse umane.

Este scrisă în Java, pe [Roma Meta Framework](#) dezvoltată în cadrul consorțiului Romulus..

Cornelius este Open Source și sub licența [commercial friendly](#).

Manual Cornelius

Customer	Project	Activity
My customer	My first project	Other
My customer	My first project	Planning
My customer	My first project	Sale activities
My customer	My first project	My first project
My customer	My first project	Teaching
My customer	My first project	Consulting
Other customer	System overview	System overview

Cornelius® Management de Proiect v1.0 -- Realizat de AssetData®

Musiteca este o aplicație de gestionare a conținutului muzical, prin intermediul căreia utilizatorul poate viziona sau asculta piese muzicale.

Scrooge este o aplicație pentru managementul financiar personal și de întreprindere.

Toți demonstratorii sunt dezvoltati folosind Roma Meta Framework și IDE4Romulus, plugin-ul IDE dezvoltat pentru Eclipse. În plus, aceste aplicații combină informații utilizând instrumentele dezvoltate Mashup. Pentru asigurarea calității, ATP4Romulus și Wapiti vor fi utilizate în faza de testare.

În conformitate cu Acordul din Consorțiu, în ceea ce privește **atribuirea drepturilor de proprietate** (intelectuală, de producție, de difuzare) între parteneri privind rezultatele proiectului, toate părțile recunosc că utilizarea de software "open source" (așa cum este definit la <http://www.opensource.org/docs/definition.php>), și/sau eliberarea de elemente ulterioare în termeni de licență asociată cu un astfel de program, în cadrul proiectului, ar putea fi benefică pentru desfășurarea acestuia și ar putea promova utilizarea și difuzarea informațiilor generate care rezultă.

7. Concluzii privind impactul proiectului Romulus

Impactul proiectului Romulus în dezvoltarea europeană de software și servicii Web se bazează pe faptul că majoritatea software-urilor și serviciilor sunt dezvoltate în Europa în Java Enterprise Edition, Java fiind limbajul cel mai popular, iar abilitățile pe care Java Enterprise Edition le crează sunt cele mai solicitate în Europa.

Europa are 1,5 milioane de dezvoltatori Java față de 4,5 milioane la nivel mondial.

Principalul scop a proiectului Romulus a fost îmbunătățirea productivității în dezvoltarea soft-ului Java destinat Web-ului. Datorită impactului software asupra economiei, reducerea costurilor de dezvoltare de software și de îmbunătățire a calității software-ului sunt obiective economice importante din Europa. În plus, Romulus este livrat ca un proiect Open Source, care promovează reducerea costurilor și îmbunătățirea inovării.

De asemenea, Romulus a fost difuzat în cadrul conferințelor, în Workshop-uri, cum ar fi Roma Java Day și în alte câteva comunități de dezvoltatori de open source.

Wapiti, instrumentul de securitate în care Romulus colaborează activ, a fost admis ca un proiect (provizoriu) alfa OWASP și este în prezent în curs de revizuire. OWASP este o comunitate deschisă la nivel mondial, axată pe îmbunătățirea securității pentru software-ul de aplicație.

În plus, proiectul Romulus a contribuit la activitățile de standardizare W3C în domeniile legate de Semantic Web în cadrul grupurilor Semantic Web Deployment Work Group și RDB2RDF Incubator Group și a stabilit conexiuni cu proiectul european OKKAM. Proiectul Romulus a contribuit la promovarea unei noi paradigme Open Source pentru dezvoltarea de sisteme destinate Web utilizând tehnologia Java. Proiectul a realizat cercetări privind proiectarea orientată pe domeniu pentru dezvoltarea de aplicații Java și a oferit un metaframework Java de tip Open Source pentru dezvoltarea de aplicații web.

Proiectul Romulus a avut avantajul comunităților Open Source existente (Roma, Liferay, Wapiti și Déri Pipes) pentru dezvoltarea de proiecte noi Open Source în cadrul Romulus (ATP, IDE4Romulus, MyCocktail, Cornelius și Segovia) și extinderea metaframework-ului Roma prin integrarea de servicii Mashup și portlet-uri.

Proiectul Romulus a dezvoltat **tehnologii de Mashup-uri** (Romulus Data and Services Mashups for Enterprise) - de date, portal, Web și servicii pentru întreprinderi; **tehnologii Web Rich Client**, securitate și fiabilitate (Rich Client Web Technologies, security and reliability) ce au constat în analiza framework-urilor client, compunerea aplicațiilor prin intermediul portlet-urilor slab cuplate, elaborarea tehnologiilor de utilizare a scripting-ului, asigurarea mecanismelor pentru dezvoltarea aplicațiilor folosind tehnologii de scripting și a echilibrului între tehnologii client și server, integrarea obiectivelor soft în ciclul de dezvoltare standard și generarea automată a testelor funcționale precum și evaluarea potențialelor vulnerabilități de securitate; **metaframework-ul Roma** și instrumente de dezvoltare (Roma Metaframework Architecture and Development Tools) și **demonstratorii verticali** (Vertical Demonstrators).

Prezentarea tehnologiilor dezvoltate de proiectul Romulus se realizează în articole

successive, după cum urmează:

- I. Prezentarea generală a proiectului Romulus;
- II. Metodologia Romulus pentru dezvoltarea de aplicații Open Source extensibile;
- III. Tehnologii Rich Client pentru aplicații Web;
- IV. – VII. Tehnologiile de Mashup-uri de date, portal, Web și servicii pentru întreprinderi;
- VIII. Metaframework-ul Roma și integrarea cu tehnologii client;
- IX. Demonstratorii verticali (Cornelius, Scrooge etc).

BIBLIOGRAFIE

[*] Traducerea în limba română a documentației elaborate în cadrul proiectului a fost asigurată de Laura Ciocoiu, Dragoș Barbu, Dragoș Smada, Ionuț Petre, Valentin Răduț, Antonio Cohal sub coordonarea Directorului de proiect dr. ing. Neculai Andrei.

[Java] Java SUN: <http://java.sun.com/javase/downloads/index.jsp> .

[Roma] Roma Framework: <http://romaframework.xwiki.com/xwiki/bin/view/Main/WebHome> (ROMA Metaframework: Architecture and Development Tools) .

[RomaURL] Roma Meta Framework project, <http://www.romaframework.org>

[AawURL] All About Widgets, 2007, <http://www.webopedia.com/DidYouKnow/Internet/2007/widgets.asp> .

[AmdURL] <http://amadig.com/us/amadig.html> .

[BEAAL] Bea AquaLogic Ensemble - <http://fi.bea.com/products/aqualogic/ensemble> .

[BwoURL] <http://www.ibm.com/developerworks/wikis/display/woolf/Mashup> (ROMULUS Data and Services Mashups for Enterprise Integration) .

[EM0208] Enterprise Mashups: Agile development in SOA - <http://blog.corizon.com/2008/02/enterprise-mashups-agile-development-in.html> .

[Abr02] Abrahamsson, P., Salo, O., Ronkainen, J. and Warsta, J. Agile software development methods. VTT Publications, 2002.

[Amb02] Ambler, S. Agile Modeling: Effective Practices for Extreme Programming and the Unified Process. New York, John Wiley & Sons, Inc., 2002.

[Aqu08] Dell'Aquila L., Garulli L., and Maestro G., Roma <Meta> Framework Handbook, 2008.

[Bec99] Beck, K. Extreme programming explained: Embrace change. Reading, Mass., Addison-Wesley, 1999.

[Cro02] Crowston, K., Scozzi, B. Open source software projects as virtual organisations: competency rallying for software development. IEE Proceedings – Software 149(1): 3–17, 2002.

[Fel00] Feller, J. and Fitzgerald, B. A Framework Analysis of the Open Source Software Development Paradigm. 21st Annual International Conference on Information Systems, Brisbane, Australia, 2000.

[Haw00] Hawrysh, S. and Ruprecht, J. Light Methodologies: It's Like Déjà Vu All Over Again. Cutter IT Journal. 13: 4–12, 2000.

[Ken06] Ken Schwaber, 2006, SCRUM Development Process, Advanced Development Methods, <http://jeffsutherland.com/oops/schwabpub.pdf> .

- [Kru00] Kruchten, P. The Rational Unified Process: an Introduction. Addison-Wesley, 2000.
- [Mar98] Martin, R. C. The Process. Object Oriented Analysis and Design with Applications. Addison Wesley: 93–108, 1998.
- [Ric06] Richardson, C. POJOs in Action, Developing Enterprise Applications with Lightweight Frameworks. Manning Publications, 2006.