

Aktyvaus XML panaudojimas informacinių sistemų studijų programos rengime

Olegas VASILECAS, Algis SAULIS, Rimvydas NOGIS (VGTU)

el. paštas: olegas@fm.vtu.lt, asaulis@fm.vtu.lt, mogis@fm.vtu.lt

1. Įvadas

Informacinių sistemų magistrantūros studijų programa rengiama pagal tarptautinį SOC-RATES/ERASMUS projektą MOCURIS (modern curriculum in information systems at master level). Projekte dalyvauja šešių Europos šalių universitetai, jį koordinuoja Vilniaus Gedimino technikos universitetas. Projekto tikslas – pateikti unifikuotą studijų programą, atspindinčią šiuolaikinius informatikos mokslų pasiekimus ir tenkinančią Europos darbo rinkos poreikius.

Studijų programa turi hierarchinę struktūrą. Ji sudaryta iš kursų, kurie savo ruožtu susideda iš modulių. Moduliai yra privalomi ir pasirinktiniai. Modulių gausa leidžia turėti įvairias specializacijas, sudaro galimybę gilinti žinias tam tikrose siaurose srityse. Studijų programa numato reikiamų specialybės žinių branduolį, bet nereglamentuoja visų mokymo aspektų. Įgyvendindamas šią programą, kiekvienas universitetas suderins ją su savo galimybėmis, tradicijomis ir specifiniais studijų proceso reikalavimais.

Studijų programa kuriama remiantis sistemų inžinerijos metodais [6], [7]. Tai sudaro galimybę turėti pakankamai lanksčią programą ir nepažeisti jos darnos keičiant modulius. Šiuo metu yra sukurta programos architektūra, sudaryti reikalavimai kursams, moduliams, specializacijoms, kuriami moduliai. Modulo aprašas turi nusistovėjusią struktūrą, kuri skiriasi įvairiuose universitetuose. MOCURIS projekte sukurta išsami sava modulario aprašo struktūra. Projekto duomenų pateikties modelis aprašytas darbuose [12], [10].

Projekto rengime dalyvauja daug specialistų, dirbančių skirtingose šalyse. Tai apsunkina projekto valdymą ir darbų koordinavimą. Pagrindine ryšio priemone tampa Internetas. Projekto duomenys saugomi reliacinėje duomenų bazėje. Duomenų pateikimo forma gali būti įvairi, priklausanti nuo konkretaus vartotojo poreikių. Tokių duomenų tvarkymui naudojamos struktūrinių dokumentų apdorojimo priemonės, leidžiančios pateikti dokumentą įvairiais pavidalais. Modulių pateikties lankstumą užtikrina XML aprašai ir XSL transformacijos [11].

Šio darbo tikslas yra išplėsti Interneto panaudojimo galimybes studijų programos rengime, automatizuojant kai kurias rankines operacijas bei individualizuojant pateikiamų dokumentų turinį ir formą. Tikslai pasiekiami panaudojus aktyvias XML technologijas.

Antrame skyriuje yra aptariami aktyvaus XML pagrindai. Trečiame skyriuje yra analizuojamos įvairios jo realizacijos, o ketvirtame – konkrečios realizacijos architektūra. Penktame skyriuje pateikiamos išvados.

2. Aktyvus XML

XML leidžia lanksčiai ir efektyviai aprašyti duomenis naudojant žymenis. Šie aprašai yra statiniai ir neleidžia automatiškai valdyti duomenų. Aktyvus XML tai leistų daryti. Jis remiasi aktyviomis taisyklėmis. Panagrinėsime aktyvių taisyklių apibrėžimą ir jų panaudojimą. Aktyvios taisyklės leidžia manipuluoti ir generuoti XML saugyklos turinį atsižvelgiant į XML duomenų pokyčius. Tai galima pavadinti elektroninėmis paslaugomis. Gaunama nauda yra tokia:

- Paslaugos, apibrėžtos aktyvių taisyklių, yra vykdomos automatiškai, įvykus tam tikram įvykiui su XML duomenimis. Atpuola reikmė kurti specialų koordinavimo mechanizmą tolimesnių veiksmų nuoseklumui aprašyti.
- Paslaugų projektavimas ir įdiegimas paprastėja. Jas lengva testuoti, įdieginėti etapais. Paprasta atsisakyti tam tikrų paslaugų. Paslaugų dinamiškumas leidžia lanksčiai prisitaikyti prie aplinkos pokyčių.
- Paslaugos gali būti nuotolinės. Parametrai jų vykdymui perduodami XML formatu (SOAP protokolu). Tai leidžia integruoti aktyviomis taisyklėmis grindžiamas paslaugas ir vykdomas remiantis tradicinėmis technologijomis.
- Egzistuoja mechanizmas taisyklių konfliktams spręsti. Konfliktinės situacijos gali būti modeliuojamos, naudojama prioritetų sistema ir lanksti derybų politika.

Aktyvios XML taisyklės remiasi ECA (event-condition-action) paradigma, kurią sudaro *įvykiai, sąlygos ir veiksmai*:

- Įvykiai yra XML dokumento pokyčiai, tinklalapio publikavimas arba keitimas, darbo sesijos užbaigimas, el. pašto pranešimo gavimas ir pan.
- Sąlygos yra užklausos XML saugyklai, parašytos kuria nors užklausų kalba. Užklausa tikrina XML dokumentų turinį, lygindama jį su tam tikra informacija arba kitu dokumentu. Užklausos rezultatas reiškia „tiesą“, kai gaunamas netuščias XML rezultatas.
- Veiksmai yra vykdomi, įvykus įvykiui ir esant sąlygos rezultatui „tiesa“. Veiksmas – tai naujo XML dokumento kūrimas arba esamo keitimas.

Aktyvios taisyklės leidžia automatiškai sintezuoti XML dokumento turinį, kurti ekrano vaizdus, pateikti personalizuotą informaciją, prižiūrėti susijusią informaciją, tikrinti apribojimų vientisumą. Aktyvaus XML panaudojimą šiuo metu riboja vieningos metodikos ir komercinių programinių produktų nebuvimas.

3. Aktyvių technologijų analizė

Panagrinėsime kelių autorių grupių, šiuo metu pasiekusių ryškiausių rezultatų, siūlomas metodikas.

Ceri grupės [5] siūlomų taisyklių skiriamasis bruožas – taisyklėse apjungiami sąlygos ir veiksmo dalis. Taisyklių užrašymui naudojamos autorių modifikuotos XSLT arba Lorel kalbos. XSLT (XSL transformacijų kalba) [8] naudojama XML dokumentų keitimui iš

vieno formato į kitą ir leidžia užrašyti sąlyginius veiksmus. Kalbos modifikacijos esmė – įvykio užrašymas.

1 pavyzdys. Aktyvi taisyklė – klasifikatorius, valdanti ateinančius el. pašto pranešimus į XML mazgą `Default_service_mail`. Klasifikatorius atrenka pranešimus, kuriuose randa žodį „MOCURIS“, ir kopijuoja juos į mazgą `toProcess`.

```
CREATE RULE MailClassify
PRIORITY: 0
EVENT: <xsl:variable name=M
      select="Default_service_mail/mail" />
      <rule:insert select="$M" />
COND_ACT:<xsl:template match="$M" >
  <xsl:variable name=subject select="subject" />
  <xsl:choose>
  <xsl:when test="contains($subject, "MOCURIS")/>
  <xsl:call-template name=" ProcessIt" >
  <xsl:param name="MProcess" />
  <toProcess>
```

Bailey [3] siūloma metodika griežtai laikosi ECA modelio. Aktyvių taisyklių sintaksė yra tokia:

```
ON <event>
IF <condition>
DO <action>
  <event> ::= INSERT | DELETE <set of nodes>
  <condition> ::= <logical expression>
  <action> ::= INSERT | DELETE <e1> BELOW <e2>
             [BEFORE | AFTER <e3>]
             <e1, e2, e3> ::= <expressions>
```

Įvykiai, sąlygos ir veiksmai čia aprašomi XPath ir XQuery kalbomis. XPath [13] suteikia patogias ir efektyvias sintaksines priemones atskirų XML dokumento dalių nustatymui. XPath remiasi eilučių išraiškomis, kurias naudoja ir kitos XML technologijos. XQuery [14] yra duomenų paieškos XML dokumentuose kalba, ji panaudota naujų dokumento fragmentų sukūrimui. Abiteboul grupės siūloma AXML kalba [1] orientuota į tinklalapio paslaugų panaudojimą duomenų integracijai. Pagrindiniai kalbos objektai yra AXML dokumentai ir tinklalapio paslaugos. Esminis kalbos bruožas – paslaugų kreipinių elementai, turintys pavidalą

`< sc > žymuo`

AXML dokumentai integruoja statinius XML duomenis ir dinامينius, gautus aktyvavus paslaugas. AXML dokumentus galima apdoroti, saugoti, atvaizduoti visais XML įrankiais. Naujus dokumentus galima gauti naudojant XQuery, įvykdžius užklausas su AXML dokumento parametrais. AXML leidžia komponuoti paslaugas, naudojant vienu kreipinių išeią kitų įeigai. Atsiranda galimybė gauti rekursyvines struktūras.

2 pavyzdys. Paslaugos elementas, aprašantis kreipinį į organizacijos el. pašto katalogą. Elemento atributai leidžia kreipinio pasiuntimui naudoti SOAP protokolą.

```
<axml:sc
  endpointURL="http://xyz.com:80/soap/rpcrouter"
```

```

nameSpaceURI="urn:xmethods-email"
methodName="GetEmail" >
<axml:params>
  <axml:param name="PersonName" xpath="../@pname" />
  <axml:param name="CompanyName">VGTU</axml:param>
</axml:params>
</axml:sc>

```

Kiyomitsu ActiveWeb sistema [9] pagrįsta dinaminėmis perkonfigūravimo taisyklėmis. Dinaminės taisyklės traktuojamos kaip metaduomenys, papildantys tinklalapio duomenis. Taisyklės sudaro trys dalys – vartotojo elgesys, istorinė sąlyga ir serverio veiksmas. Taisyklių sintaksė:

vartotojo elgesys: istorinė sąlyga → serverio veiksmas

Taisyklės aprašomos deklaratyviai. Vartotojo elgesys gali būti traktuojamas kaip įvykis, istorinė sąlyga – kaip sąlyga, serverio veiksmas – kaip *veiksmas* ECA modelyje. Vartotojo elgesį nusako jo buvimo vieta, laikas, kviečiami tinklalapiai. Istorinę sąlygą sudaro informacija apie prieigos istoriją. Serverio veiksmas aprašomas kaip vykdomų funkcijų seka. Dinaminio perkonfigūravimo taisyklių įvedimui ir apdorojimui sukurtas taisyklių procesorius yra pasiekiamas per interneto naršyklę.

4. Aktyvių technologijų įvertinimas

Nagrinėtų technologijų palyginimui sudarytoje 1 lentelėje matyti, kad visos technologijos turi trūkumų. Nei viena iš jų nėra pilnai sutvarkyta, universali ir paruošta plačiam vartojimui. Tai tik autorių prototipai, nepasiekę komercinių produktų lygio. Perspektyviausia šiuo požiūriu atrodo Active XQuery kalba [4].

MOCURIS projekte individualizuotam informacijos pateikimui tinkamiausias pasirodė ActiveWeb, o automatizuotam reguliarių pranešimų išsiuntimui el. paštu ir SMS žinutėmis – AXML.

1 lentelė
Technologijų palyginimas

Sistemos autoriai	ECA modelis	Pagrindiniai bruožai	Trūkumai
Ceri ir kiti	Supaprastintas	Naudoja modifikuotas XSLT, XQuery, Lorel kalbas	Kalbų naujų modifikacijų panaudojimas
Bailey ir kiti	Pilnas	Naudoja XPath, XQuery; įdiegta konfliktų analizė	Veiksmas – tik XML dokumentų kūrimas ir keitimas
Abuteboul ir kiti	Bus įdiegtas vėliau	Duomenų integracija, paslaugų aprašymas	Nėra ECA taisyklių
Kiyomitsu ir kiti	Supaprastintas	Taisyklių procesorius reaguoja tik į užklausą	Veiksmas – tik ekrano turinio, stiliaus perkonfigūravimas

Aktyvaus XML realizacijai perspektyviausia yra lygiavertė architektūra (peer-to-peer) [2]. Lygiavertis mazgas – tai svetainė, turinti dokumentų saugyklą, užklausų procesorių, protokolų adapterių, paslaugų ir valdymo modulius. Valdymo modulis koordinuoja paslaugų aktyvavimą. Lokaliai apibrėžtos paslaugos vykdomos vietoje, o kitos – kituose mazguose arba serveriuose. Lygiavertis mazgas yra universalus, jis gali vykdyti kliento ir serverio funkcijas, palaikydamas ryšį SOAP protokolu.

5. Išvados

Straipsnyje nagrinėjamos informacinių sistemų studijų programos rengimo automatizavimo problemos. Automatizavimu siekiama gerinti projekto valdymą, bendradarbiavimą tarp projekto dalyvių, užtikrinti reikiamos informacijos savalaikį pateikimą.

Išanalizuoti bendri reikalavimai aktyvaus XML kalboms, naudojančioms ECA taisykles. Kalbų analizė parodė, kad šiuo metu dar nėra universalios kalbos, galinčios tenkinti visus vartotojų poreikius. Atskirų autorių siūlomos kalbos yra prototipų lygio ir turi trūkumų.

Darbo rezultatai realizuoti veikiančiame sistemos prototipe, kuriame panaudotos dviejų technologijų idėjos.

Literatūra

- [1] S. Abiteboul, O. Benjelloun, T. Milo, *A Data-Centric Perspective on Web Services*, Technical Report 212, INRIA (2001).
- [2] S. Abiteboul, O. Benjelloun, T. Milo, I. Manolescu, R. Weber, Active XML: peer-to-peer data and Web services integration, in: *Proc. of VLDB*, Hong Kong (2002).
- [3] J. Bailey, A. Poulovassilis, P. Wood, An event-condition-action language for XML, in: *Proc. of the 11 th Int. WWW Conference*, ACM Press, Hawaii, USA (2002), pp. 486–495.
- [4] A. Bonifati, D. Braga, A. Campi, S. Ceri, Active XQuery, in: *Proc. of ICDE* (2002).
- [5] A. Bonifati, S. Ceri, S. Paraboschi, Active rules for XML: a new paradigm for e-services, *The VLDB Journal*, **10**, 39–47 (2001).
- [6] A. Čaplinskas, O. Vasilecas, Modern curriculum in information systems: a case study, *Informacinės technologijos ir valdymas*, **1**(22), 22–27 (2002).
- [7] A. Čaplinskas, Curricula engineering: application of systems engineering methods to the development of university curricula. *Informacinės technologijos ir valdymas*, **1**(22), 10–21 (2002).
- [8] *Extensible stylesheet language transformations (XSLT) specification (Version 1.0)*. W3C Recommendation (1999), <http://www.w3.org/TR/WD-xsl>
- [9] H. Kiyomitsu, A. Takeuchi, K. Tanaka, ActiveWeb: XML-based active rules for web view derivations and access control, in: *Proc. of the Workshop on ITVE*, IEEE Press, **23**(6), 31–39 (2001).
- [10] K. Lapin, A. Saulis, O. Vasilecas, MOCURIS projekto mokymo medžiagos pateikties modelis, naudojantis reliacines ir XML technologijas, in: *Informacinės technologijos 2003*, Konferencijos pranešimų medžiaga, KTU, Kaunas (2003), pp. VI-29–VI-35 .
- [11] D. Martin, M. Birbeck, M. Kay, B. Loesgen, *Professional XML*, Wrox Press (2001).
- [12] O. Vasilecas, K. Lapin, Informacinių sistemų mokymo medžiagos pateikties, naudojant XML, modelis, *Liet. matem. rink.*, **42** (spec.nr.), 294–298 (2002).
- [13] *XML path language (XPath) specification*. W3C Recommendation (1999), <http://www.w3.org/TR/xpath>
- [14] *XQuery: A Query Language for XML*. W3C Working Draft (2001), <http://www.w3.org/TR/2001/WD-xquery>

Active XML: application to the development of information systems curriculum

O. Vasilecas, A. Saulis, R. Nogis

The paper deals with the problems of expanding the system of information systems curriculum development in the field of Web application. The main aim is to enhance the project coordination and control, to personalize presentation of information. Different active XML methods based on ECA (event-condition-action) model have been analyzed. All the methods discussed have drawbacks and limitations, and can be treated as prototypes of future universal system. Peer-to-peer architecture is recognized as the most suitable for active system. The results of investigation are implemented in a prototype that uses ideas of few methods.