

Aktyvių užklausų kalbų panaudojimas verslo taisyklių realizacijai

Olegas VASILECAS, Algis SAULIS, Valdas SINKEVIČIUS,
Saulius DEREŠKEVIČIUS, Rimgaudas Stanislovas VAIČIULIS (VGTU)

el. paštas: olegas@fm.vtu.lt, asaulis@fm.vtu.lt, valdas@isl.vtu.lt, saulius@cadteam.lt, vaiciulis@vpt.lt

1. Įvadas

Verslo taisyklės yra neatsiejama veiklos dalis. Verslo taisyklės apibrėžia ar apriboja tam tikrus verslo aspektus, nusako verslo struktūrą, valdo verslo procesus [8]. Šios taisyklės atsispindi verslo dokumentuose, jomis grindžiamas verslo informacinių sistemų darbas.

Verslo taisyklės apibrėžia verslo žmonės, jos užrašomos įprasta kalba. Taisyklių analizei gali būti naudojamos įvairios formalios loginio modeliavimo kalbos, pvz., esybių ryšių (ER), vieninga modeliavimo (UML) [4], konceptualių grafų [11]. Toks aprašas gali būti diegiamas kompiuterizuotose sistemose, kur taisyklių veikimas yra automatizuotas. Šiame lygyje taisyklės sudaro įvairūs faktai, apribojimai, veiksmai.

Taisyklėse aprašomi teiginiai skirstomi į struktūrinius ir dinامينius. Mus domina sudėtingesnė dinaminų teiginių grupė, t.y., teiginiai, priklausantys nuo tam tikrų duomenų, kurie kinta laiko bėgyje. Dinaminus teiginius sudaro:

- dinaminiai apribojimai, susiję su leistiniais sistemos perėjimais iš vieno būvio į kitą;
- sprendimo taisyklės, gautos iš kitų žinių, ar atlikus matematinius skaičiavimus;
- reakcijos taisyklės, susijusios su veiksmy inicijaviu, kaip atsaku į įvykius, susijusius su tam tikromis sąlygomis.

Verslo taisyklės, ekvivalentiškos dinaminiam teiginiam, atitinka ECA (event–įvykis, condition–sąlyga, action–veiksmas) paradigmai ir turi šias bendras savybes [12]:

- Verslo taisyklės valdo duomenys, jos aktyvuojamos keičiantis duomenų būsenai. Kiekviena verslo taisyklė turi įvyki, kuris gali būti tiesiogiai aprašytas arba netiesiogiai numatytas. Pastaruoju atveju, netiesioginis įvykis suprantamas kaip atitinkamos duomenų būsenos pasikeitimas.
- Vienos verslo taisyklės turi tiesioginę sąlygą, kitos – ne. Sąlygos nebuvimas visada gali būti pakeistas numatoma sąlyga su pastovia reikšme Tiesa.
- Visi dinaminiai teiginiai numato apibrėžtus veiksmus, sudarančius taisyklių esmę. Taisyklių veiksmai gali būti formuluojami kaip korekciniai veiksmai, arba atitinkamų instrukcijų pateikimas vartotojui.

Šiuo metu dar nėra universalių technologijų verslo taisyklių įdiegimui kompiuterizuotose sistemose. Tai priklauso nuo sistemos tipo, duomenų saugojimo formų, taisyklių valdymo objektų. Šiame straipsnyje yra nagrinėjamos verslo taisyklės, taikomos elektroninių dokumentų valdymo sistemose. Antrame skyriuje yra analizuojami įvairūs elektroninių dokumentų saugojimo būdai. Trečiame skyriuje yra aptariamos ECA modelio realizacijos XML dokumentų sistemose, o ketvirtame – siūlomas realizacijos įdiegimas. Penktame skyriuje pateikiamos išvados.

2. Elektroninių dokumentų saugojimo būdai

Duomenų apimtis žiniatinklio serveriams prieinamuose šaltiniuose ir duomenų bazėse sparčiai auga, todėl būtina turėti efektyvias priemones jų tvarkymui. Perspektyviausias elektroninių dokumentų formatas čia yra XML. Didelės apimties duomenų valdymas siejamas su duomenų bazių valdymo sistemomis (DBVS). Šios sistemos yra skirtos struktūrizuotiems duomenims, tuo tarpu kai XML duomenys dažniausiai tokie nėra. XML tikslas yra atskirti struktūrą ir turinį nuo formos. XML dokumentai integruoja tik struktūrą ir turinį. Dokumentų turinį struktūrizuoja XML žymenys, saugantys papildomą informaciją. Įprastinės DBVS nepritaikytos efektyviam XML duomenų valdymui. Jų saugojimui ir paieškai turi būti taikomi kiti būdai.

XML dokumentų saugojimui, prieigai ir paieškai yra naudojami įvairūs metodai [6]. Šie metodai skirstomi į dvi pagrindines grupes – su duomenų bazėmis ir be jų. Kai duomenų bazės nėra naudojamos, dokumentai yra saugomi įprastuose failuose. Paieška juose vykdoma teksto analizės būdu. Tai lėtas ir neefektyvus būdas, nes XML failai turi būti nuosekliai peržiūrėti.

Naudojant duomenų bazes, dokumentus ar duomenis valdo reliacinės, objektinės reliacinės, ar objektiškai orientuotos DBVS. Šiuo atveju XML dokumentai turi būti transformuoti į atitinkamos DBVS duomenų modelius. Transformacijos vykdomos konceptualiajame lygyje, o fizinė duomenų organizacija – DBVS.

Naudojamos trys pagrindinės XML dokumentų saugojimo strategijos, taikomos su įvairių tipų DBVS.

- Tiesioginis saugojimas. XML dokumentai ir jų žymenys saugomi vienos lentelės stulpeliuose arba duomenų bazės objektuose. Duomenų bazėje jie valdomi kaip dideli dvejetainiai objektai. Tai panašu į XML dokumentų saugojimą atskiruose failuose.
- Išskaidymas ir saugojimas. XML dokumentų komponentai (elementai, atributai, elementų turinys ir t.t.) transformuojami į vienos ar kelių lentelių stulpelius arba į duomenų bazės objektus. Šiuo atveju dokumentai visiškai išardomi.
- Hibridinis metodas. Tai tarpinis būdas tarp pirmųjų dviejų. Išskaidomos tik kai kurios dokumentų dalys. Kitos dalys, tame tarpe žymenys, yra saugomos pilnai.

Tiesioginis saugojimas užtikrina efektyvią prieigą. Dokumentų paieška vykdoma teksto analizės būdu arba indeksų pagalba, tačiau struktūrinė navigacija ir komponentų keitimas atliekami sunkiai. Tuo tarpu išskaidytų dokumentų dalių paieška ir keitimas atliekami lengvai. Vienintelis trūkumas šiuo atveju yra tas, kad papildomas laikas yra gaišamas dokumento atstatymui.

Hibridinis metodas siekia panaudoti pirmųjų metodų privalumus [5]. Sunkiausia šiuo atveju yra nuspręsti, kurias pradinio XML dokumento dalis reikia išskaidyti, o

kurias palikti pradiniu pavidalu. Paprastai, reguliarios struktūros dokumentų dalys yra išskaidomos, o neregulios – paliekamos pradiniame pavidale, išsaugant galimybę lengvai jas keisti.

Efektyvi XML dokumentų valdymo sistema privalo naudoti duomenų modelius, atspindinčius specifines XML savybes. Šie modeliai turėtų remtis atitinkamomis duomenų struktūromis, pvz., XML duomenų tipu, aprašytu [9].

3. ECA modelio realizacija

Aktyvių taisyklių užrašymui ir panaudojimui XML dokumentų sistemose yra siūlomos įvairios technologijos. Jų autoriai [1], [2], [3], [7] siūlo žinomų formatų XSLT, XQuery, XPath modifikacijas, papildytas naujais elementais. Šių technologijų analizė aprašyta darbe [13]. Pagrindinė problema yra ta, kad technologijų autoriai, pristatydami savo sistemas, nepasiūlo visiems prieinamų jų įdiegimo priemonių. Teoriškai patraukliausia atrodo Active XQuery kalba [3]. Ji yra universali ir pakankamai sudėtinga. Sudėtingumas tampa rimta kliūtimi bandant šią kalbą realizuoti.

Verslo taisyklių realizacijai buvo pasirinktas kitas kelias – sukurti savarankišką, paprastą, neuniversalų, bet lengvai realizuojamą aktyvumo palaikymo mechanizmą.

Taisyklių užrašymui pasinaudota XML išplėtimo galimybėmis. Taisyklės aprašinio XML dokumento struktūra grindžiama taisyklių aprašymo kalba SRML (Simple Rule Markup Language) [10]. Taisyklės yra susietos su įvykiais. Taisyklės užrašomos mūsų sukurtu XML plėtiniu, pavadintu ECARML (ECA Rule Markup Language). ECARML sintaksė yra tokia:

```
EVENT NAME = insert|update|delete
RULE ELEMENT = element_name
  RULE:NAME rule_name
  [RULE:PRIORITY integer number]
  [RULE:CONDITION OPERATOR = eq|neq|lt|lte|gt|gte|ex|nex
    CONDITION:CLASS class
    CONDITION:L_VARIABLE NAME=name TYPE=type VALUE=value
    CONDITION:R_VARIABLE NAME=name TYPE=type VALUE=value]
  RULE:ACTION
ACTION:CLASS action_class
ACTION:VARIABLE NAME=name TYPE=type VALUE=value
```

Taisyklių elementai turi šias savybes:

- EVENT susieja taisyklę su įvykiu. Event Name apibrėžia įvykį.
- RULE aprašo taisyklę. Ją sudaro taisyklės vardas (name), prioritetas (priority), sąlyga (condition) ir veiksmas (action). Rule Element nurodo, kokiam XML dokumento elementui taisyklė yra priskirta.
- Rule:Priority nustato prioritetą, naudojamą sprendžiant taisyklių konfliktus.
- Rule:Condition Operator apibrėžia taisyklės operatorių. Jo galimos reikšmės: lygu (eq), nelygu (neq), mažiau (lt), mažiau arba lygu (lte), daugiau (gt), daugiau arba lygu (gte), yra (ex), nėra (nex).
- Condition:Class nurodo java modulį, kuris tikrina taisyklėje nurodytą sąlygą.

- **Condition:L_variable** aprašo taisyklės operatoriaus kairėje pusėje esantį kintamąjį. Jo atributai yra vardas (name), tipas (type) ir reikšmė (value). Galimi tipai: tekstinis (string), loginis (boolean), simbolinis (char), sveiki skaičiai (int, byte, short, long), dešimtainiai skaičiai (float), skaičiai su slankiu kableliu (single, double), java funkcija (function), XQuery užklausa (query).
- **Condition:R_variable** aprašo taisyklės operatoriaus dešinėje pusėje esantį kintamąjį. Jo atributai analogiški.
- **Rule:Action** nurodo taisyklės veiksmą, aprašomą java klase (**Action:Class**), ir veiksmui reikalingus kintamuosius (**Action:Variable**).
- **Action:Variable** aprašomi analogiškai **Condition:Variable**.

Siūloma sistema yra pakankamai adaptyvi, ji leidžia plėsti taisyklių rinkinį nekeičiant vykdomosios programos struktūros. ECARML apibrėžtos taisyklės gali būti lengvai transformuojamos į kitus formatus, tame tarpe ir į Active XQuery.

4. Verslo taisyklių realizacija

ECARML buvo praktiškai panaudota tarptautinio projekto MOCURIS [13] dokumentų valdymo sistemoje. Verslo taisyklės čia tarnauja projekto darbų koordinavimui, dokumentų kokybės kontrolei, reikiamos sąveikos tarp vadovų, vykdytojų ir vartotojų užtikrinimui.

Pavyzdys. Trinant raktinius žodžius iš aprašo elemento *Keywords*, šie žodžiai automatiškai turi būti pašalinti ir iš žodyno (Glossary), jeigu buvo įrašyti jame. Dokumento autorius elektroniniu laišku informuojamas apie įvykdytus pakeitimus.

```
<Event name="delete">
  <Rule element="Keywords">
    <rule:name>Delete_keyword</rule:name>
    <rule:priority>0</rule:priority>
    <rule:condition operator="eq">
      <condition:class>ex3_cond.java</condition:class>
      <condition:l_variable name="keyword" type="function" value=
        "getparameter('keyword')"/>
      <condition:r_variable name="termin" type="query" value= "//
        doc('mocuris.xml')/MOCURIS/Glossary[@module_id
          ={$getparameter('module_id')}] /Termin/term"/>
    </rule:condition>
    <rule:action>
      <action:class>delete_keyword.java</action:class>
      <action:variable name="termin" type="function"
        value="getparameter('keywords')"/>
      <action:variable name="notice" type="string" value="Raktinis
        žodis ištrintas iš aprašo"/>
    </rule:action>
  </Rule>
</Event>
```

Verslo taisyklių praktinei realizacijai sukurtas ECARML interpretatorius. Jį sudaro užklausų apdorojimo blokas, taisyklių procesorius ir programiniai moduliai. Serverio sistemą sudaro interpretatorius ir XML dokumentų bei verslo taisyklių saugyklos. Praktinė sistemos eksploatacija parodė jos pakankamą efektyvumą.

5. Išvados

Straipsnyje nagrinėjamos verslo taisyklės, taikomos elektroninių dokumentų valdymo sistemose. Aptarti šių dokumentų saugojimo metodai ir galimi aktyvumo realizavimo būdai. Verslo taisyklės, ekvivalentiškos dinaminiam teiginiam, atitinka ECA paradigmai ir gali būti aprašytos aktyvių taisyklių pagalba.

Šiuo metu dar nėra universalios aktyvių taisyklių užrašymo kalbos, tenkinančios visus vartotojų poreikius. Perspektyviausia yra Active Xquery, tačiau ji neturi įdiegimo mechanizmo. Autorių siūlomas XML plėtinys ECARML ir jį realizuojantis interpretorius sėkmingai naudojami tarptautinio projekto MOCURIS dokumentų valdymo sistemoje. Šis metodas gali būti taikomas ir kitose panašaus tipo sistemose.

Literatūra

1. 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).
2. J. Bailey, A. Poulouvasilis, P. Wood, An event-condition-action language for XML, in: *Proc. of the 11th Int. WWW Conference*, ACM Press, Hawaii, USA (2002), pp. 486–495.
3. A. Bonifati, D. Braga, A. Campi, S. Ceri, Active XQuery, in: *Proc. of ICDE* (2002).
4. G. Booch, J. Rumbaugh, I. Jacobson, *The Unified Modeling Language User Guide*, Addison-Wesley (2000).
5. K. Böhm, K. Aberer, W. Klas, Building a hybrid database application for structured documents, *Multimedia Tools and Applications*, **8**(1), 65–90 (1999).
6. C. Kanne, G. Moerkotte, Efficient storage of XML data, in: *Proceedings of ICDE*, San Diego, California, IEEE Computer Society (2000).
7. H. Kiyomitsu, A. Takeuchi, K. Tanaka, ActiveWeb: XML-based active rules for web view derivations and access control, *Proc. of the Workshop on ITVE*, **23**(6), 31–39 (2001).
8. R.G. Ross, *The Business Rule Book: Classifying, Defining and Modeling Rules*, 2nd edition, Database Research Group, Boston, MA (1997).
9. D. Scheffner, J.C. Freytag, *The XML Query Execution Engine (XEE)*, Technical Report HUB-IB-158, Humboldt-Universität zu Berlin (2002).
10. *Simple Rule Markup Language (SRML)* (2001). <http://xml.coverpages.org/srml.html>
11. I. Valatkaite, O. Vasilecas, Verslo taisyklių modeliavimas koncepciniais grafais ir jų realizavimas naudojant aktyvių duomenų bazių trigerius, *Liet. matem. rink.*, **42** (spec. nr.), 289–293 (2002).
12. I. Valatkaite, O. Vasilecas, A conceptual graphs approach for business rules modeling, to be published in *Lecture Notes of Computer Science*, Springer-Verlag (2004).
13. O. Vasilecas, A. Saulis, R. Nogis, Aktyvaus XML panaudojimas informacinių sistemų studijų programos rengime, *Liet. matem. rink.*, **43** (spec. nr.), 304–309 (2003).

SUMMARY

O. Vasilecas, A. Saulis, V. Sinkevičius, S. Dereškevičius, R.S. Vaičiulis. Application of active query languages to the implementation of business rules

The paper deals with the problems of business rules implementation into the Web-based document management system. XML document storing, retrieving, querying, and transformation methods are analyzed. Different active methods based on ECA (event-condition-action) model have been discussed. An Active XQuery language is recognized as the most promising, but it lacks implementation means. A specialized XML extension is created and corresponding interpreter is developed to implement business rules into the document management system of the project MOCURIS. The method proposed can be used to solve similar tasks.

Keywords: rule, business rule, e-document, XML, ECA model.