

Aktyviųjų duomenų bazės taisyklių aktyvumo analizė ir vertinimas

Juozas LAUČIUS, Arūnas RIBIKAUSKAS, Justas TRINKŪNAS

Vilniaus Gedimino technikos universitetas

Saulėtekio al. 11, LT-10223 Vilnius

el. paštas: juozas@fm.vgtu.lt; arunas@fm.vgtu.lt; justas@isl.vgtu.lt

Santrauka. Šiuolaikinių informacinių sistemų viena iš būtinų sudėtinių dalių yra aktyvi duomenų bazių valdymo sistema, kurioje yra realizuojamos verslo taisyklės. Paprastai dinaminės verslo taisyklės yra realizuojamos kaip trigeriai arba procedūros, o statinės verslo taisyklės – duomenų bazės struktūromis. Realizavus sudėtingą taisyklėmis grįstą sistemą, dėl didelio taisyklių skaičiaus ir sudėtingos jų tarpusavio sąveikos susiduriama su sistemos valdymo ir analizės sunkumais. Šiame straipsnyje pateikiama aktyviųjų taisyklių aktyvumo samprata, o aktyvumo vertinimui kaip vieną iš galimų būdų siūlome panaudoti modifikuotą funkcinių taškų metodą.

Raktiniai žodžiai: ECA taisyklės, taisyklių aktyvumo vertinimas, funkcinių taškų metodas.

1. Įvadas

Pastaruosiu metu informacinėms sistemoms realizuoti vis plačiau naudojamos aktyviosios duomenų bazės. Jų privalumas yra tas, kad verslo logika, kuri įprastai yra integruota taikomosiose programose, gali būti iš dalies arba visiškai perkeliama į aktyviųjų duomenų bazių serverį. Aktyvusis tokių duomenų bazių valdymo sistemų (DBVS) komponentas yra aktyviosios taisyklės (dar vadinamos ECA taisyklėmis) (angl. *ECA rules*), sudarytos iš trijų pagrindinių dalių: įvykio, sąlygos, veiksmo (angl. *Event, Condition, Action*). ECA taisyklės veikia taip: kai pasirodo įvykis, tikrinama sąlyga ir, jeigu ji tenkinama, vykdomas veiksmas. ECA taisyklėmis galima realizuoti daugumą kompiuterizuojamų verslo taisyklių [5].

Reliacinėse DBVS ECA taisyklės gali būti užrašomos kaip trigeriai ar kaip saugomos procedūros. Trigeriai yra orientuoti įvykių fiksavimui ir reakcijai į juos, tuo tarpu saugomos procedūros daugiau tinka ECA taisyklėms, neturinčioms įvykio dalies, užrašyti.

Naudojant trigerius bei saugomas procedūras, reliacinėse DBVS galima realizuoti sudėtingas, įvairių tipų verslo taisykles. Didelis taisyklių skaičius apsunkina programų sistemų analizę ir valdymą. Metodų, kaip analizuoti ir įvertinti atskirų taisyklių savybes, nėra daug. Dažniausiai yra siūlomi instrumentai, vaizdžiai pateikiantys ECA taisyklių sistemos pagrindines savybes.

Mokslinėje literatūroje dažniausiai nagrinėjama tokia ECA taisyklių ir jų sistemos savybė kaip baigtinumas [1], [4], suprantamas kaip įvykio sužadintas taisyklės vykdymas, lemiantis kaskadinį ir rekursinį taisyklių sistemos apdorojimą.

Baigtinumas yra svarbiausia ir labiausiai pageidaujama savybė. Be kokybinių charakteristikų dažnai nagrinėjama ir tokia kiekybinė charakteristika, kaip sudėtingumas. Ši savybė yra svarbi, prognozuojant klaidų sistemoje tikimybę ir aktyviosios DBVS serverio apkrovą. Tačiau viena svarbi ECA taisyklių savybė – aktyvumas – dažnai užmirštama. Šiame straipsnyje pabandydysime aptarti ECA taisyklės aktyvumo sampratą bei aktyvumo kiekybinio vertinimo galimybes.

2. ECA taisyklės aktyvumas

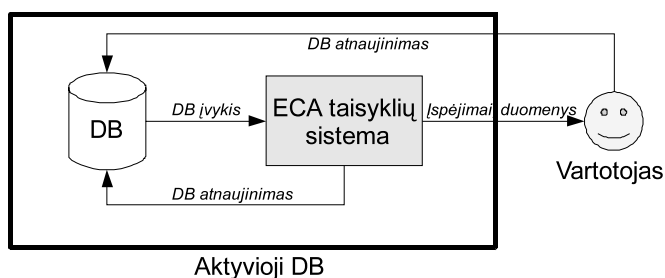
Paprastai ECA taisyklės sudarytos iš šių pagrindinių dalių:

1. Įvykis (angl. *Event*), sužadinantis ECA taisyklės vykdymą.
2. Sąlyga (angl. *Condition*), kurią patenkinus, vykdomas sužadintoje ECA taisyklėje numatytas veiksmas.
3. Veiksmas (angl. *Action*) – imperatyvioji taisyklės dalis, leidžianti duomenų bazės (DB) sistemai būti aktyvia.

Pasirodžius įvykiui, tikrinama sąlyga ir, jeigu sąlyga yra tenkinama, vykdomas veiksmas. Veiksmai gali atnaujinti DB ar taisyklių rinkinio struktūrą, sukelti tam tikrą elgseną DB viduje ar išorėje, informuoti vartotoją ar administratorių apie tam tikrą situaciją, nutraukti transakciją [9]. Veiksmas gali būti vykdomas, kaip priedas prie (angl. *in addition to*) operacijos, kuri sukėlė atitinkamą įvykį, arba vietoj (angl. *instead of*) jos [6]. Bendroju atveju ECA taisyklių sistemos vieta aktyviosiose DB gali būti pavaizduota 1 pav.

Paveiksle matyti, kad ECA taisyklių aktyvumas yra dvejopas – taisyklės gali atnaujinti DB būseną, pakeisti joje esančius duomenis ir tuo pačiu gali sukelti naujus įvykius, sužadinančius kitas taisykles arba pateikti aktyviosios DB sistemos vartotojui duomenis, pranešimą, kurie gali sukelti DB atnaujinimus iš vartotojo pusės.

ECA taisyklių poveikis, jei toks yra, į DB esančius duomenis yra aiškus ir nedviprasmiškas. Pokyčiai dažniausiai perkelia DB į kitą aktualią būseną, kuri atspindi realaus pasaulio būseną dalykinėje aplinkoje. Tuo tarpu taisyklių poveikis į išorę, vartotojui, nebūtinai sulaukia atsako ir baigiasi DB būsenos atnaujinimu. ECA taisyklių poveikį išorės vartotojui bendroju atveju prognozuoti sunku – neaiškus vartotojas (tai gali būti ir gyvas darbuotojas, tai gali būti išorinės informacinės sistemos ar kitos taikomosios programos), neaiškus ir jo atsakas aktyviajai DBVS.



1 pav. ECA taisyklių veikimas aktyviojoje DBVS.

Sudėtingų ECA taisyklių sistemų nagrinėjimas reikalingas tam, kad būtų lengviau suprasti pavienių taisyklių ir visos sistemos veikimą, lokalizuoti ir taisyti klaidas, pasitaikančias atskirose ECA taisyklėse. Klaidos taisyklėse, tiesiogiai veikiančiose DB duomenis, yra ypač skausmingos, nes sugadintų duomenų atstatymas yra labai sudėtingas, o kartais neįmanomas. Tuo tarpu klaidingos informacijos pateikimas į išorę, nors yra taip pat svarbus, bet gali ir nepaveikti bazėje saugomų duomenų, o taip pat gali likti nepastebėtas. Todėl aktyviają DB sistemą palaikančiam administratoriui pasidaro ypač svarbu, kiek aktyviai viena ar kita ECA sistemos taisyklė tiesiogiai keičia DB būseną, t.y. kiek ji yra aktyvi.

ECA taisyklės, kuriomis yra generuojamos ataskaitos ar jų elementai, niekaip neįtakoja DB būsenos, todėl tokios taisyklės nėra aktyvios. Galima išskirti atskiras taisyklių sistemos dalis, atsakingas už analogišką informacinės sistemos funkcionalumą, ir visos jos bus pasyvios. Programų sistemos vystytojams klaidos tokiose taisyklėse mažiau svarbios, nes jos nesugadina bazėje esančių duomenų.

Tolesniuose skyreliuose siūlomas aktyvumo kiekybinio vertinimo metodas. Atsiradus klaidai, gadinančiai duomenis bazėje, kiekybinis įvertis leistų pirmiausia atkreipti dėmesį į aktyvesnes ECA taisykles.

3. Taisyklės, taisyklių sistemos struktūra ir aktyvumas

Formali taisyklių elgsenos analizė gali palengvinti sistemos veikimo suvokimą. Nesudėtinga suprojektuoti atskirą aktyvią taisyklę, kai jos įvykis, sąlyga ir veiksmas yra aiškiai apibrėžti. Tačiau kur kas sunkiau yra suvokti bendrą aktyvių taisyklių elgseną, nes jų tarpusavio sąveika yra pakankamai subtili.

Panagrinėkime atskiros taisyklės aktyvumą. Nors aktyviai veikianti komponentė yra taisyklės veiksmo (angl. *Action*) dalyje, tačiau taisyklės aktyvumas priklauso ir nuo taisyklės vykdymą lemiančių įvykių (angl. *Event*) bei sąlygos (angl. *Condition*) dalių. Apibendrinus vienos taisyklės aktyvumą galima būtų užrašyti taip:

$$A_{ECA} = A_E \cdot A_C \cdot A_A;$$

čia A_{ECA} – visos taisyklės aktyvumas, A_E , A_C – aktyvumo koeficientai, kuriuos lemia taisyklės įvykių bei veiksmo dalys, A_A – taisyklės veiksmo dalies aktyvumas. Sudėtingiausias yra dydžio A_A skaičiavimas, jį atliekant galima naudotis funkcinių taškų metodu, aptartu kitame skyrelyje.

A_E yra dydis, parodantis, kaip dažnai sistemoje nutinka įvykis, iššaukiantis taisyklės vykdymą. Paprasčiausiu atveju tai gali būti statistinis dydis, parodantis, kiek kartų per nustatytą laiko tarpą sistemoje nutinka tai, kas gali sužadinti taisyklės vykdymą (įvykių dažnis). Jei įvykis yra kažkurio duomenų vieneto atnaujinimas (pvz., naujo dokumento registracija), tai tokio įvykių dažnį galima prognozuoti iš anksto, kuriant sistemą. Priklausomai nuo konkrečios DBVS, kurioje sistema realizuota, kai kurie statistiniai rodikliai, reikalingi įvykių dažnumui skaičiuoti, gal būti aptikti sisteminiame duomenų bazės kataloge. Tuo atveju, kai įvykis yra sudėtinis, daug kas priklauso nuo elementariųjų įvykių derinimo būdo, panaudojant loginius ir/arba laiko operatorius [8].

Tipiniai sudėtiniai įvykiai ir jų vertinimas:

- Konjunkcija – kai pasirodo du įvykiai E_1 ir E_2 . Sudėtinio įvykio pasirodymo dažnis skaičiuojamas priklausomai nuo įvykių jungimo būdo. Tarkime, sudėtinį įvykį sudaro E_1 ir E_2 įvykiai. Tada sudėtinio įvykio dažnis yra ne didesnis nei $\min(A_{E_1}, A_{E_2})$, kur A_{E_1}, A_{E_2} yra elementariųjų įvykių E_1 ir E_2 aktyvumo koeficientai.
- Disjunkcija – kai pasirodo arba įvykis E_1 , arba E_2 . Pasirodymo dažnis yra $A_{E_1} + A_{E_2}$.
- Seka – E_1 pasirodo prieš E_2 . Tiksliai apskaičiuoti galima tik žinant įvykių pasiskirstymo pobūdį, tačiau sudėtinio įvykio dažnis yra ne didesnis nei $\min(A_{E_1}, A_{E_2})$.
- Pasikartojimas – įvykio E pasirodymas n kartų sužadina vieną sudėtinį įvykį. Pasirodymo dažnis yra A_E/n .

Tuo atveju, kai taisyklė yra iškviečiama kitos taisyklės ar naudojama kaip paprasta duomenų bazėje saugoma procedūra, A_E reiktų prilyginti 1.

Taisyklės aktyvumo A_C dalis atspindi tikimybę, kad ECA taisyklės sąlyga bus patenkinta. Jei taisyklė yra besąlyginė, tai $A_C = 1$. Priešingu atveju $0 < A_C < 1$. Konkrečiai taisyklei A_C skaičiavimas gali būti palengvintas, jei žinoma jos sąlyga. Pavyzdžiui, naudojama DB kintamojo X prilyginimo konkrečiai reikšmei Y sąlyga $X = Y$. Tarkim, X su vienoda tikimybe gali įgyti n reikšmių, tada $A_C = 1/n$. Duomenų bazių teorijoje analogiškas yra išrenkamumo koeficientas (apdorojimo metu sąlygą tenkinančių ir visų įrašų skaičiaus santykis). Tikslus koeficientas nežinomas, tačiau jo statistinė vertė gali būti saugoma sisteminame DB kataloge ir panaudojama prognozei.

Taisyklių sistemoje privalu atsižvelgti ir į taisyklių tarpusavio santykį. Viena ECA taisyklė gali iššaukti kitos taisyklės vykdymą. Tai gali būti tiek tiesioginis saugomos procedūros iškvietimas, tiek įvykis (tarkim, duomenų atnaujinimo operacija), sužadantis naujos ECA taisyklės vykdymą. Dėl šių ryšių iškviečiančios taisyklės aktyvumas santykinai išauga. Analizuoti sistemoje esančios ECA taisyklės aktyvumą siūloma, pasitelkus grafa. Grafo viršūnėse atidedamos ECA taisyklės, viršūnės sujungiamos iškvietimo briaunomis. Iškviečiančios ECA taisyklės aktyvumas galėtų būti skaičiuojamas taip:

$$A_{ECA} = A_E \cdot A_C \cdot (A_A + k_2 \cdot A_{E_2} \cdot A_{C_2} \cdot A_{A_2} + k_3 \cdot A_{E_3} \cdot A_{C_3} \cdot A_{A_3} + \dots),$$

kur $A_{E_2} \cdot A_{C_2} \cdot A_{A_2}$, $A_{E_3} \cdot A_{C_3} \cdot A_{A_3}$ – iškviečiamųjų ECA taisyklių aktyvumas, o koeficientai k_1, k_2 parodo, ar taisyklė yra iškviečiama tik iš nagrinėjamos ECA taisyklės. Koeficientas $k = 1$, jei iškvietimas besąlyginis, o kitų iškvietimo šaltinių nėra, priešingu atveju $k < 1$.

4. Taisyklių aktyvumo vertinimas, naudojant funkcinių taškų metoda

Šiame skyriuje nagrinėsime vieną iš galimų būdų ECA taisyklių veiksmo dalies aktyvumo A_A vertinimui – modifikuotą funkcinių taškų metoda.

IFPUG (angl. *International Function Point User Group*) tarptautinė organizacija [7] sukūrė metodą, kuriuo remiantis programinis kodas skaidomas į elementus, o priklausomai nuo elemento tipo, priskiriamas atitinkamas taškų skaičius [10]. Gauta taškų

suma dauginama iš išlyginamojo faktoriaus reikšmės. Išlyginimo faktoriaus reikšme įvertinama bendra programos svarba, programavimo metodas, išorinių bibliotekų ir programos vidinių funkcijų ir procedūrų pakartotinio panaudojimo atvejai. Tokiu būdu gauname reikšmę, kuri vadinama funkcinių taškų skaičiumi. Ši metrika leidžia pamatuoti programos sudėtingumą ir dydį. Funkcinių taškų metodas yra aprašytas ISO 20926 standarte.

Funkcinis taškas – matavimo vienetas, naudojamas informacinės sistemos funkcionalumo kiekiui išreikšti [3].

Vienas iš pagrindinių funkcinių taškų naudojimo kaip matavimo vieneto privalumų yra tas, kad funkcinius taškus galima pakankamai nesudėtingai apskaičiuoti. Be to, naudojant funkcinius taškus galima įvertinti sistemos kūrimo produktyvumą [2].

Algoritmas, kuriam yra taikomas funkcinių taškų metodas, turi tenkinti šiuos pagrindinius reikalavimus: a) algoritmas turi būti sudarytas iš žingsnių, pagrįstų matematinėmis formulėmis ir loginiais operatoriais, realizuojančiais verslo taisykles; b) algoritmas turi būti užbaigtas, išsprendžiamas, įvykdomas.

SQL (angl. *Structured Query Language*) – struktūrizuota užklausų kalba, naudojama daugelyje šiuolaikinių reliacinių DBVS duomenų atnaujinimui. Šią kalbą sudaro sakiniai, skirti atlikti operacijas su duomenimis (išrinkti, įterpti, ištrinti, atnaujinti) duomenų bazėje, duomenų struktūroms apibrėžti, duomenų prieigai duomenų bazėje apibrėžti.

SQL kalba užrašytos taisyklės tenkina aukščiau išvardintus reikalavimus, todėl ECA taisyklėms, užrašytoms šiomis kalbomis, gali būti naudojamas funkcinių taškų metodas. Su duomenų bazėje atnaujinimu nesusiję algoritmo elementai neturėtų būti vertinami (duomenų apdorojimas kompiuterio atmintyje, įvedimas ir išvedimas į išorę).

1 lentelėje pateikiame siūlomas funkcinių taškų neišlygintas reikšmes, kurias susumavę gausime taisyklės veiksmo dalies aktyvumo A_A įvertį.

Pateiktas aktyvumo vertinimo metodas leidžia reitinguoti aktyviausias taisykles pagal jų poveikį duomenų bazei. Sistemos vystytojai ir administratoriai, siekdami

1 lentelė. Funkcinių taškų neišlygintos reikšmės

Funkcinis taškas	Neišlygintas taškų skaičius
Duomenų išrinkimo sakiniai	0
Duomenų atnaujinimo užklausa (Insert, Update, Delete sakiniai)	1, 5, 7 – priklausomai nuo sakinio apimties ir sudėtingumo
Teisių vartotojui suteikimo, atšaukimo operacija (Grant, Revoke sakiniai)	1
Duomenų struktūros modifikavimo sakiny (Create, Alter, Drop sakiniai)	5, 10, 15 – priklausomai nuo sakinio apimties ir sudėtingumo
Ciklas n žingsnių	Aktyvumo koeficientas n .
Vykdomų sakinių atšaukimo operacija Rollback	Sukauptą taškų skaičių dauginti iš 0.
Vykdomų sakinių įtvirtinimo operacija Commit	Sukauptą taškų skaičių dauginti iš 1.

užtikrinti, kad nebus gadinami duomenys bazėje, aktyvesnėms taisyklėms turėtų skirti didesnę dėmesį.

5. Išvados

Atlikus aktyviųjų duomenų bazės taisyklių analizę matyti, kad aktyvumo sampratai skiriamas nepakankamas dėmesys. Siūloma patikslinti aktyvumo sąvoką, kaip aktyvumą suprantant tik poveikį į DB – mus supančio pasaulio dalykinės srities atspindį.

Pasiūlytas taisyklės aktyvumo vertinimo metodas, atsižvelgiant į taisyklės struktūrą ir jos vietą taisyklių sistemoje. Išanalizavus funkcinių taškų metodą matyti, kad jis yra tinkamas SQL kalbos sakiniams vertinti, ir todėl gali būti panaudotas taisyklių aktyvumo kiekybiniam vertinimui.

Literatūra

1. A. Aiken. Static analysis techniques for predicting the behavior of active database rules. *ACM Transactions on Database Systems*, 20(1), 1995.
2. A.J. Albrecht, J.E. Gaffney. Software engineering. *IEEE Transactions on*, Vol. SE-9(6):639–648, 1983.
3. A.J. Albrecht. Measuring application development productivity. In: *Proceedings of the Joint SHARE, GUIDE, and IBM Application Development Symposium*, Monterey, California, IBM Corporation, 83–92, 1979.
4. E. Baralis, J. Widom. Better static rule analysis for active database systems. *ACM Transactions on Database Systems*, 25(3), 2000.
5. F. Bry, M. Eckert, P.L. Patranjan, I. Romanenko. Realizing business processes with ECA rules: Benefits, challenges, limits. Principles and practice of semantic web reasoning. *LNCS*, 4187, Springer Berlin, 48–62, 2006.
6. T. Hong. A survey of active database systems. In: *Proceedings of 3rd International Workshop Rules in Database Systems (RIDS '97)*, *LNCS*, 1312, Skoevde, Sweden, 1997.
7. *International Function Point User Group*. <http://www.ifpug.org> [2009 05 19]
8. U. Jaeger, J.K. Obermaier. Parallel event detection in active database systems: The heart of the matter. In: *Proceedings of the 2nd International Workshop on Active, Real-Time, and Temporal Database Systems*, *LNCS*, 1553, Springer, 159–175, 1998.
9. N.W. Paton, O. Diaz. Active database systems. *ACM Computing Surveys*, 31(1):63–103, 1999.
10. R. Tichenor, N. Tichenor, B. Charles. Measure size, complexity of algorithms using function points. *CrossTalk*, 12–15, 2001.

SUMMARY

J. Laučius, A. Ribikauskas, J. Trinkūnas. Analysis and evaluation of ADBMS rules activity

Article presents analysis and evaluation of ECA rules activity. Most of active rules are implemented by triggers and functions. Sophisticated rules based systems have several problems including difficulty to control and analyse the system. We propose to use Functional Size Measurement Method to evaluate activity of rules in ADBMS. A function point is a unit of measurement to express the amount of business functionality an information system provides to a user. Paper provides major reasons and benefits for using this method.

Keywords: ECA rules, function point method, evaluation of rules activity.