

ПРОЦЕСНИЯТ ПОДХОД И ПРОБЛЕМАТИКАТА МУ В ИНФОРМАТИКАТА

Владимир Димитров

Факултет по математика и информатика
Софийски университет „Свети Климент Охридски“
e-mail: cht@fmi.uni-sofia.bg

Резюме: Направен е ретроспективен преглед на възникването и развитието на процесния подход в бизнеса и информатиката. Разглежда се отражението на процесния подход в софтуерната индустрия. Представени са основните му направления. Очертани са основните проблеми свързани с процесната ориентация. Разгледана е появата на Грид технологиите като решение за растящите нужди от научни изчисления. Направен е паралел между Грид за научни изчисления и Грид за бизнеса. Дискутирани са въпросите за стандартизация на Грид технологиите. Подчергана ролята на ориентираната към услуги архитектура в Грид изчисленията. Направена е класификация на свойствата на бизнес процесите в две категории: желани и нежелани свойства. Формулировката се основава на темпоралната логика.

Ключови думи: Процесен подход.

1. Проблематика

През последните години, бизнесът и информатиката се развиват под знака на процесната ориентация [1, 2]. Една от причините за това развитие са неудачите свързани със системния подход. От времето, когато изчислителната мощ на компютрите достигна достатъчно високо ниво, се повдигна въпросът за обединение в единна система на всички самостоятелни приложения, реализиращи или поддържащи функциите на предприятието. Големите разходи, свързани с интеграцията на тези програмни активи, доведоха до идеята за системния подход при разработката на програмното осигуряване и по-точно до концепцията на автоматизираните системи за управление (АСУ) [3]. Тази концепция се базираше на разбирането, че процесите на управление в различните предприятия не се отличават особено много от предприятие в предприятие. За съжаление, това се оказа невярна постановка, огромни средства бяха похарчени за разработката на АСУ, но внедряването се оказа още по-трудно налагаше се да се правят множество доработки.

Ситуацията се усложни през 90-те години на миналия век, когато световната икономика от индустриална се превърна в икономика на знанията и услугите [4]. При индустриалната икономика предприятието е силно свързана система и предприятията

си взаимодействат единствено чрез пазара. В края на миналия век обаче се появиха много нови форми на коопериране между предприятията. Нещо повече, за да оцелее отделното предприятие на високотурбулентния пазар, то трябва да е широко отворено за всевъзможни форми на кооперация с другите предприятия [5]. При този модел на бизнес, най-важното за предприятията стана да развият и кооперират конкурентните си предимства, като заедно с това изнасят (аутсорсинг) дейностите, които не са конкурентни предимства, там където те се извършват на конкурентно ниво. В крайна сметка се оказа, че предприятието е само звено във веригата на доставките и голяма част от неговата функционалност вече не се реализира самостоятелно вътре в него.

На базата на новите икономически теории: вериги на доставките, вериги на стойността, управление на ресурсите на предприятието и т.н., започна да се прилага концепцията на бизнес процесите. Предприятието се организира по бизнес процесите, в които участва, и по задачите от тях, които реализира. Самият термин „бизнес процес“ е доста стар, но се използваше само за организацията на промишленото производство.

Реорганизацията на бизнеса по новите бизнес модели наложи въвеждането на процесния подход в стандартизацията и в разработката на програмното осигуряване. Бизнес процесите не са вече затворени само в рамките на отделното предприятие – последното е част от глобалните процеси и реализира част от техните задачи. Най-често това са задачите, в които предприятието има конкурентни предимства.

Тези изменения в бизнеса доведоха до това, че през 90-те години на миналия век започна интензивна разработка на адекватно програмно осигуряване като SCM (Supply Chain Management) [6], CRM (Customer Relations Management) [7], ERP (Enterprise Resource Planning) [8] и други. Програмното осигуряване има водеща роля в съвременния бизнес. От една страна, то позволява предприятието гъвкаво да се приспособява към турбулентната бизнес среда. Доколкото е гъвкаво програмното осигуряване на предприятието, дотолкова е гъвкаво самото предприятие. От друга страна, се оказва, че единственият ресурс, в който може да се инвестира до „безкрайност“, е програмното осигуряване. За всички останали ресурси, инвестициите имат ниво на насищане, след което по-нататъшните инвестиции са неизгодни. Инвестициите в програмно осигуряване е един от основните начини за постигане на конкурентно предимство.

Процесният подход се оказва сериозен проблем за бизнеса, за организациите по стандартизация и за разработчиците на програмно осигуряване. Много предприятия не можаха да се адаптират към него и излязоха от пазара през 90-те години на миналия век. Организациите по стандартизация, с известно закъснение, в началото на новия век, започнаха да преразглеждат стандартите, свързани с организация и управление, с цел въвеждане на процесния подход. До ден днешен, много организации имат сериозни проблеми с внедряването на процесния подход.

Особено тежка е ситуацията с доставчиците на програмно осигуряване. Едва усвоили обектно-ориентираното програмиране през 90-те години на миналия век и обектно-ориентираното проектиране на новия век и вече навлязоха нови изисквания за разработка на процесно-ориентирано програмно осигуряване. Първата стъпка беше преходът от обектно-ориентирания подход към проектиране и програмиране на компоненти. Идеята е, че само част от програмното осигуряване на глобалния бизнес процес се реализира в дадено предприятие. Такава една част (компонент) е силно свързана група от класове с общ интерфейс. Това решение е развитие на обектно-

ориентирания подход към независимост от платформи и доставчици, но не решава проблемите на процесния подход.

От началото на новия век, процесният подход започна развитието си в две направления. На високо бизнес ниво се работи по стандарта ITIL (IT Infrastructure Library) [9] – библиотека, описваща най-добрите практики за организация на работните потоци в предприятието. В седемте тома на библиотеката е описан целият набор от процеси, необходими за осигуряване на постоянно високо качество на информационно-технологичното обслужване с цел повишаване удовлетвореността на потребителите. Трябва да се отбележи, че процесите не са просто ориентирани към осигуряване на непрекъсната работа на компонентите на информационно-технологичната инфраструктура. В много по-голяма степен те са ориентирани към изискванията на потребителите и клиентите. В крайна сметка, всичките ITIL процеси са за повишаване на конкурентоспособността. В наши дни, дори информационно-технологичните отдели на предприятията не се чувстват в безопасност, тъй като трябва да се опасават от изнасяне на дейностите им извън предприятието.

Реализацията на бизнес нивото се осъществява чрез ориентираната към услуги архитектура (*Service-Oriented Architecture, SOA*), която се поддържа от серията стандарти за Уеб услуги WS-* [10]. Самата архитектура не е някаква новост, но едва със стандартите за уеб услуги (*WebService, WS*) от второто поколение, тя може напълно да реализира възможностите си.

Ожесточената борба между доставчиците на програмно осигуряване доведе до немислими за близкото минало сливания и консорциуми. Това от своя страна реши редица проблеми с прилагането на процесния подход, но това не означава, че технологиите вече са напълно зрели. Преди кризата водещите доставчици на инструменти за разработка на ориентирани към услуги приложения доставяха версии и изменения на всеки три месеца, като не се грижеха особено за обратната съвместимост. Сега темповете са по-бавни – на полугодие по издание. Основният мотив за високите темпове е доставката на нова функционалност, за да може собственото решение да стане основа на стандартизация. Използването на такъв софтуер е като работа в минно поле – не се знае кога и къде ще „гръмне“. Това доведе до създаването на организационни форми от вида на „летящите бригади“ за инсталация и разгръщане на работоспособни среди.

От направения обзор може да се направят следните изводи:

1. Процесната ориентация в ITIL и WS-* (SOA) все още не е достатъчно съзряла и лошо се поддържа от инструментите.
2. Семантиката на бизнес процесите, представяни с ITIL, BPMN, WS-BPEL, WS-CDL и UML диаграмите на дейността е и ще бъде проблем.
3. Разработката на инструменти за поддръжка на бизнес процесите е първостепенна задача за доставчиците им.

2. За научните изчисления и Грид технологиите

В края миналия век, идеята за Грид беше оформена като самостоятелна концепция. В [11] беше добре представена концепцията „компютинга като услуга“ (*computing as a service*). Заедно с това, авторите представиха бъдещия Грид в почти всички области на

информатиката като се започне от взаимодействието с потребителите, като се премине през изискванията за безопасност на информацията и се стигне до възможностите за реализация на програмното осигуряване. Грид идеята вдъхнови както научния свят, така и бизнеса. Нещо повече, това не беше само идея, а имаше и изследователски прототип GlobusToolkit [12]. Концепцията за Грид не подмина темата за научните изчисления. Обикновено, новите концепции в информатиката поставят ударение върху решаване проблемите на бизнеса. Много рядко някоя нова концепция е посветена на решаването на проблемите в научните изчисления, като проблема за дистанционно управление на научния експеримент.

Всички големи доставчици на компютърна техника и програмно осигуряване, като IBM, Oracle и Microsoft, се включиха в работите по Грид инициативите като OGF (*Open Grid Forum*) [13]. Дори, като IBM, се заеха да внедряват Globus Toolkit в бизнеса. Бързо стана ясно, че Грид, така както е представен в [11], скоро няма да го бъде – необходими са много години на изследване и разработка на подходящите за реализацията му стандарти. Още се оказа, че Globus Toolkit не става или е твърде скъпо внедряването му в бизнеса. Това доведе до създаването през 2004 година в рамките на OGF на Enterprise Grid Alliance [14]. Този алианс си постави за задача да внедри Грид технологиите в предприятията в обозримо бъдеще. В крайна сметка, тази инициатива даде много резултати, като се почне от блейд сървърите и се стигне до облачните изчисления.

Междувременно работите по Грид стандартите започнаха да се движат с много бавни темпове. Какво се беше случило? По замисъл, Грид компютингът е услуга. Тази идея е залегнала в OGSA (*Open Grid Service Architecture*) [15]. Това означава, че Грид трябва да бъде реализиран с използване на ориентираната към услуги архитектура и освен това програмното осигуряване, което се изпълнява в Грид среда, трябва да е ориентирано към услуги. Както беше отбелязано по-горе, ориентираната към услуги архитектура върху уеб услуги е все още недостатъчно зряла технология. Така работите се концентрираха върху ориентираната към услуги архитектура, а когато нещо там съзрее достатъчно (например WSDL), то се доработва и за Грид.

Трябва да се подчертае, че бъдещето на Грид е определено в термините на ориентираната към услуги архитектура. Реализацията на Грид трябва да е ориентирана към услуги. Програмното осигуряване, работещо в Грид, трябва да е ориентирано към услуги. С други думи, потребителите на Грид ще асемблират приложенията като процеси от услуги, а няма да програмират на езици от системно ниво. Това означава, че функционалността на ROOT [16] трябва да е налична в Грид среда като услуги, а не, както е сега, като среда за разработка и изпълнение от вида на UCSD Pascal [17] от 70-те години на миналия век.

В действителност, ориентираната към услуги архитектура в Грид все още е далече от реалността. Има някои опити да се приложат тези идеи в научните изчисления по проектите CMS и Alice, но тези доморасли инициативи не е ясно до колко ще намерят реално приложение.

3. Средства за описание на бизнес процеси

Основните средства за описание на бизнес процеси са BPMN, UML диаграмите на дейността, WS-BPEL и WS-CDL. Освен тях има и други частни средства от вида на Agis [18], но те нямат шанс да станат индустриални стандарти.

В този раздел накратко са най-важните характеристики на тези средства, тяхното място и роля.

Бизнес процесите на най-високо ниво могат да се описват с BPMN. Това е графична нотация, която добре се възприема от крайните потребители и е достатъчно прецизна за анализ на бизнеса. BPMN позволява да се прилага постепенна детайлизация на диаграмите, като се започне от схематично очертаване на бизнес процеса и се стигне до генериране на скелета на кода на бизнес процеса. Всички елементи на нотацията могат на стъпки да се детайлизират. Елементите на нотацията добре се възприемат от крайните потребители, тъй като семантиката им е интуитивно свързана с бизнес процесите, а не с реализацията им във вид уеб услуги, например. При детайлизацията може да се посочи начина на реализация на всеки един от елементите. Например, в BPMN нотацията процесите се представят като композиция от задачи. Задачата е елементарна единица работа на определено ниво на абстракция. В даден процес, всички задачи са от едно и също ниво на абстракция. Всяка от задачите може да бъде реализирана като процес, уеб услуга или просто като изпълним код в процеса. Това означава, че задачата може да бъде детайлизирана на следващото ниво на абстракция като бизнес процес (подпроцес).

WS-BPEL е създаден за рекурсивна композиция на уеб услуги в процеси – бизнес процесът композиран с WS-BPEL е уеб услуга. С този език се описва в детайли поредицата от извикване на уеб услугите, които композира в бизнес процес.

WS-BPEL е базиран на XML. Той не е подходящ като език за програмиране, поради което в инструментите се представя с графична нотация, добре възприемана от програмистите. Този подход е използван например в IBM Web Sphere Integration Developer [19]. На нивото на WS-BPEL код, итерации не се предвиждат. На това ниво се пише код. Най-често, скелетът на кода на WS-BPEL процеса е генериран от BPMN спецификация – такъв подход е използван в IBM Web Sphere Business Modeler [20].

UML диаграмите на дейността се използват и като средство за описание на бизнес процеси. Те имат добре специфицирана семантика в метасистемата на UML. От друга страна диаграмите на дейността, освен за описание на бизнес процеси, се използват и за описание на сложни операции в класовете. Тези диаграми не са свързани с бизнес анализа и уеб услугите, поради което BPMN е по-адекватно средство за тази цел. При разработката на BPMN са използвани диаграмите на дейността като концепция за структуриране. Диаграмите на дейността са по-подходящи за проектиране и програмиране, докато BPMN диаграмите – за анализ на бизнеса.

Формалният апарат, върху който се основават BPMN и UML диаграмите на дейността, са мрежите на Петри, но те са разширени до мощта на машината на Тюринг. WS-BPEL ползва за формален апарат процесната алгебра от типа на CSP и π -смятането. Формална спецификация за BPMN и WS-BPEL няма.

Накрая остана WS-CDL. Предимството му е, че има формална спецификация на семантиката му и е утвърден стандарт. Неговото назначение е да описва сътрудни-

чеството на уеб услугите на глобално ниво. Това не е език за програмиране. Хореографията (WS-CDL спецификацията) се използва за проверка на съответствие към нея на дадена реализирана композиция на уеб услуги. Има данни от литературата, че от хореография може да се генерира WS-BPEL код, но тези идеи нямат практическо приложение. По-различно стои въпросът с генериране на WS-BPEL код от BPMN спецификации, тъй като итеративният подход на разработка естествено води до код и това е пътят, избран от повечето доставчици на инструменти. Стандартът WS-CDL остана встрани от приложения.

Свойства на бизнес процесите

Свойствата на бизнес процесите могат най-общо да се определят като желателни и нежелателни. Едно свойство може да бъде желателно за някои проектни изисквания и нежелателно за други. Например, по проектно изискване може да се иска системата да спре след като свърши работата си. Ако системата зацikli, т.е. системата се изпълнява безкрайно, тогава това свойство е нежелателно. От друга страна, ако системата е проектирана да обслужва безкраен процес, тогава спирането ѝ е нежелателно, тъй като тя трябва да работи до безкрайност.

По-долу под система се разбира бизнес процес, за да не се смесва с понятието процес. Системата е набор от паралелно изпълняващи се процеси (нишки или подпроцеси), които използват общи ресурси и се синхронизират за получаването им. Системата има цел – полезната работа, която да върши.

И така от гледна точка на проектирането, системите са такива, които обслужват непрекъснати процеси и се изпълняват до безкрайност, и такива, които след свършване на работата си прекратяват активността си. В първия случай има следните възможности:

1. Системата работи безкрайно и върши полезна работа.
2. Системата работи безкрайно, но от това няма никаква полза.
3. Системата спира при определени проектирани обстоятелства. Най-често това са външни обстоятелства, от които зависи системата.
4. Системата спира поради грешка в проекта.

Във втория случай се предполага, че системата ще работи известно време и ще спре. В този случай има следните възможности:

1. Системата е завършила изпълнението си в желателно състояние, т.е. тя успешно е извършила своята работа или е прекратила работата си поради обстоятелства, независещи от нея, но тя е реагирала на тях адекватно.
2. Системата е завършила работата си в нежелано състояние. Причината за това може да е грешка в проекта, например блокировка.
3. Системата зацikliя. Това е грешка, тъй като се предполага, че след като работи известно време и свърши полезна работа, системата ще спре.

Част от тези ситуации са добре описани във формалните модели, но другите не е възможно еднозначно да бъдат определени – за всеки конкретен случай това трябва да се направи.

Системите за верификация на процеси се изследват: мъртва хватка (блокировка); дивергенция; детерминизъм; зацikliяне (жива хватка или жива блокировка); достижимост; условия представени чрез формули на линейната темпорална логика; и еквивалентност. Всяка от тези ситуации е представена по-долу.

Мъртвата хватка е състояние, при което по-нататъшното изпълнение е невъзможно. В този случай всеки от процесите се опитва да изпълни някакво свое действие. Процесите не могат да съгласуват помежду си кое ще бъде следващото действие (и от кой именно процес) и в резултат на това нищо не се случва. Системата е блокирана, по-нататък нищо не се изпълнява, тъй като процесите изчакват до безкрайност.

Дивергенция има, когато системата обслужва до безкрайност вътрешни събития и не участва в полезни външни събития. Живата хватка е случаят, при който системата работи до безкрайност, но не върши полезна работа – това е вид дивергенция. Системата в това състояние също е безполезна. Тя използва до безкрайност компютърните ресурси и с това е по-лоша от мъртвата хватка. При дивергенцията, системата след известно време започва да работи хаотично.

Системата е детерминистична, когато във всичките ѝ състояния преходът към следващо състояние е еднозначно определен. С други думи, няма състояние, при което изборът на прехода да е неопределен (недетерминистичен). Понякога, на ниво реализация, такава неопределеност може да означава грешка. На ниво спецификация, неопределеност от този вид дава възможност за различни реализации, което поставя спецификацията на по-високо ниво на абстракция.

Безкрайната работа на системата, както беше разгледано по-горе, може да е грешка, а може и да не е.

Във всеки проект има специфични изисквания. Проектантът трябва посочи какво в системата е желателно и какво не е. Това може да се формулира с логически изрази. Съответно, спецификацията на системата трябва да се проверява за достижимост на тези условия.

Формулировката на желателните и нежелателните условия може да се направи особено детайлно със средствата на линейната темпорална логика. По-точно, това се прави с прилагане на модалните оператори. В общия случай системата се проверява върху безкрайни поредици от изпълненията ѝ, като започват проверките от фиксирана позиция. Операторите, които се прилагат, са:

- $\circ\Phi$, формулата Φ трябва да е истинна в следващото състояние;
- $\square\Phi$, формулата Φ трябва да е истинна цялата следваща поредица;
- $\diamond\Phi$, формулата Φ евентуално ще бъде истинна някъде натам в поредицата;
- $\Psi U\Phi$, Φ е в сила в разглежданата позиция, а Ψ трябва да е в сила преди нея;
- $\Psi R\Phi$, Φ трябва да е истинна преди първото състояние, в което Ψ става истинна. Ако Ψ никога не стане истинна, то тогава Φ трябва винаги да е истинна.

С тези формули, системата се изследва много по-прецизно, но това изисква доста добра подготовка на проектанта.

И накрая, при разработката на спецификацията на системата, работата се извършва на итерации. В резултат на това се получава поредица от все по-детайлни спецификации. На края на тази поредица се намира реализацията на системата. На всяка итерация трябва да се проверява дали новата спецификация е еквивалентна на предишната в някакъв смисъл. Еквивалентността може да се определя по следите на изпълнението, по семантиката на аварийните ситуации и по семантиката на дивергенциите. С други думи:

1. Следата на изпълнение на детайлизираната система трябва да бъде детайлизирана следа на специфицираната система.
2. Множеството от нежелателни крайни състояния на детайлизираната система трябва да съдържа детайлизирани нежелателни крайни състояния на специфицираната система.
3. Множеството от дивергенции на детайлизираната система трябва да съдържа детайлизирани дивергенции на специфицираната система.

4. Заключение

В настоящия материал са разгледани основните средства за описание на бизнес процеси, като те са охарактеризирани по назначение и състояние. По-задълбочено представяне на всяко от тях е направено в други материали.

Тук са разгледани и дискутирани свойствата на бизнес процесите, които могат да са обект на специализирано и общо изследване.

Задачата на по-нататъшните изследвания е да избере и представи методология за изследване на бизнес процесите. В този контекст разбирането и описанието на семантиката на представените средства е особено важно за преобразуването на бизнес процесите в еквивалентни моделни формални представяния, които подлежат на формално изследване.

Най-вероятно мащабните работи по тази задача няма да приключат в рамките само на един проект. Резултатите от формалната спецификация на семантиката на средствата за описание на бизнес процеси чрез формални модели имат важно значение за разработка на инструментите по поддръжката им. Процесният подход е водещият подход при решаване проблемите на съвременния бизнес и при разработката на програмни системи.

Благодарности

Изследваният представени в този доклад са по проекта „Съвременни езици, среди и технологии за програмиране и прилагането им при подготовка на софтуерни специалисти“ финансиран от Фонд „Научни изследвания“ по Национален конкурс „Финансиране на фундаментални научни и научно-приложни изследвания в приоритетните области“ – 2012г., ДФНИ-И01/12.

Литература

- [1] Business process orientation, http://en.wikipedia.org/wiki/Business_process_orientation
- [2] Businessprocessorientation. A blog about business process management and process-oriented organizational design, <http://www.processorientation.com/>

- [3] Автоматизированная система управления, http://ru.wikipedia.org/wiki/Автоматизированная_система_управления
- [4] Service economy, http://en.wikipedia.org/wiki/Service_economy
- [5] Laudon, K. J. Laudon. *Management Information Systems*, 11/E, Prentice Hall, 2010, pp. 672.
- [6] Supply chain management, http://en.wikipedia.org/wiki/Supply_chain_management
- [7] Customer relationship management, http://en.wikipedia.org/wiki/Customer_relationship_management
- [8] Enterprise resource planning, http://en.wikipedia.org/wiki/Enterprise_resource_planning
- [9] ITIL, <http://www.itil-officialsite.com>
- [10] Web service, http://en.wikipedia.org/wiki/Web_service
- [11] Foster, I., C. Kesselman. (Editors) *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann; 1998, pp. 675.
- [12] Globus Toolkit, <http://www.globus.org/>
- [13] Open Grid Forum, <http://www.gridforum.org/>
- [14] Enterprise Grid Alliance, <http://www.ggf.org/gf/docs/egadocs.php>
- [15] Open Grid Services Architecture Basic Execution Service, <http://www.ggf.org/documents/GFD.108.pdf>
- [16] ROOT, <http://root.cern.ch>
- [17] UCSD Pascal, http://en.wikipedia.org/wiki/UCSD_Pascal
- [18] IDS Scheer, <http://www.ids-scheer.com/international/en>
- [19] WebSphere Integration Developer, <http://www-01.ibm.com/software/integration/wid/>
- [20] WebSphere Business Modeler Advanced, <http://www-01.ibm.com/software/integration/wbimodeler/advanced/features/>