

ПЛОВДИВСКИ УНИВЕРСИТЕТ "ПАИСИЙ ХИЛЕНДАРСКИ"
ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА
КАТЕДРА "КОМПЮТЪРНИ ТЕХНОЛОГИИ"

НИКОЛАЙ ВЕЛИЧКОВ ПАВЛОВ

**ОБЕКТНО–ОРИЕНТИРАНА РАМКА
ЗА РАЗРАБОТКА НА
РАЗПРЕДЕЛЕНИ БИЗНЕС ПРИЛОЖЕНИЯ**

АВТОРЕФЕРАТ

на дисертационен труд
за присъждане на образователната и научна степен "доктор"
в област 4. Природни науки, математика и информатика,
професионално направление 4.6 Информатика и компютърни
науки, научна специалност 01.01.12 Информатика

Научни ръководители: проф. д-р Асен Рахнев
доц. д-р Николай Кюркчиев

Пловдив, 2011 г.

ПЛОВДИВСКИ УНИВЕРСИТЕТ "ПАИСИЙ ХИЛЕНДАРСКИ"
ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА
КАТЕДРА "КОМПЮТЪРНИ ТЕХНОЛОГИИ"

НИКОЛАЙ ВЕЛИЧКОВ ПАВЛОВ

**ОБЕКТНО–ОРИЕНТИРАНА РАМКА
ЗА РАЗРАБОТКА НА
РАЗПРЕДЕЛЕНИ БИЗНЕС ПРИЛОЖЕНИЯ**

АВТОРЕФЕРАТ

на дисертационен труд
за присъждане на образователната и научна степен "доктор"
в област 4. Природни науки, математика и информатика,
професионално направление 4.6 Информатика и компютърни
науки, научна специалност 01.01.12 Информатика

Научни ръководители: проф. д-р Асен Рахнев
доц. д-р Николай Кюркчиев

Рецензенти: акад. проф. д.т.н. Иван Попчев
проф. д.м.н. Светослав Марков

Пловдив, 2011 г.

Дисертационният труд е обсъден и насочен за защита на разширено заседание на катедра “Компютърни технологии” при Факултета по математика и информатика на ПУ “Паисий Хилендарски”.

Дисертационният труд “Обектно–ориентирана рамка за разработка на разпределени бизнес приложения” съдържа 121 страници. Използваната литература включва 64 източника, от които 61 на латиница. Списъкът на авторските публикации се състои от 5 заглавия.

Защитата на дисертационния труд ще се състои на 04.07.2011 от 13:00 в заседателната зала на новата сграда на ПУ “Паисий Хилендарски”.

Материалите по защитата са на разположение на интересующите се в секретариата на ФМИ, нова сграда на ПУ, каб. 330 всеки работен ден от 8:30 до 17:00 часа.

Съдържание

Обща характеристика на дисертационния труд	4
Актуалност на темата	4
Цели и задачи на дисертационния труд	4
Структура и обем на дисертационния труд	6
Кратко съдържание на дисертационния труд	7
Глава I. Обектно-ориентирани рамки за разработка на приложения	7
Глава II. Авторска справка за разпределени бизнес приложения — Framework for Distributed Business Applications (FDBA)	9
Глава III. Прототип на обектно-ориентирана рамка за разработка на разпределени бизнес приложения — FDBA ...	16
Глава IV. Анализ на инвестиционни проекти	22
Перспективи	28
Апробация	29
Авторска справка	29
Благодарности	30
Публикации по дисертационния труд	30
Литература	31

Обща характеристика на дисертационния труд

Актуалност на темата

През последните години софтуерът с общо предназначение достигна определено ниво на развитие, на което неговата функционалност и приложност са максимизирани и трудно може да се говори за нови възможности. Като следствие софтуерната индустрия се съсредоточава все повече върху разработването на приложения, които задоволяват специфичните нужди на даден отрасъл и дори конкретен клиент.

Същевременно, информационните технологии навлизат във всяко ниво на икономическия живот. Това поставя софтуерните фирми пред задачата да произвеждат постоянно нови решения, ориентирани към съответния бизнес субект, като при това от първите се изисква да доставят решенията си в максимално кратки срокове, но с високо качество. Тази тенденция поставя силно въпроса за използването на вече изградени софтуерни решения с цел понижаване разходите и времето за разработка с гарантиране на добро качество.

Настоящата работа разглежда възможностите за по-ефективното изграждане на софтуер чрез обектно-ориентирана рамка за разработка на разпределени приложения, предназначени за нуждите на малки и средни предприятия в областта на търговията, финансите, застраховането, управлението, частната и държавна администрация. Специално внимание се отделя на средствата за изграждане на разпределени системи, максимизиране на използването на код.

Цели и задачи на дисертационния труд

Основната цел на дисертационния труд е да се разработи модел и архитектура на софтуерна рамка за разработване на разпределени приложения за автоматизиране дейността на малки и средни предприятия и да се реализира прототип на базата на съвременни информационни технологии, чрез който да се разработят и внедрят реални крайни приложения.

Основните задачи са:

1. Да се синтезира модел на софтуерна рамка за разработка на разпределени приложения за малки и средни предприятия. Подзадачи:
 - 1.1. Да се очертаят предимствата от използването на рамки при разработката на софтуер;
 - 1.2. Да се направи преглед на съществуващите технологии и рамки за разработка на разпределени системи и за разработка на бизнес приложения, които да се вземат под внимание при проектирането на рамката, обект на настоящата дисертация;
 - 1.3. Да се формулират основните и допълнителни изисквания към проектираната рамка;
 - 1.4. Да се разработи моделът на рамката, като се опишат:
 - 1.4.1. Характерът на софтуерни приложения за бизнеса;
 - 1.4.2. Основни функционални изисквания;
 - 1.4.3. Интеграцията със системи за създаване на електронни документи;
 - 1.4.4. Модел за разработка и поддръжка на крайни приложения;
 - 1.4.5. Функционалност за автоматично генериране на потребителски интерфейс.
2. Да се проектира архитектура, базирана на стандарти и нови технологии и да се създаде прототип на рамката. Подзадачи:
 - 2.1. Да се направи анализ и избор на съвременни обектно-ориентирани технологии и рамки за разработка на приложения и разпределени системи;
 - 2.2. Да се разработи системната архитектура на рамката;
 - 2.3. Да се класифицират системните услуги, които рамката предлага;
 - 2.4. Да се класифицират приложните услуги на рамката за решаване на стандартни типове задачи;
 - 2.5. Да се реализира прототип на рамката;
 - 2.6. Да се разработят и внедрят реални крайни приложения.
3. Да се създаде приложение за свръхчувствителен анализ на парични потоци за спомагане взимането на инвестиционни решения. Подзадачи:

- 3.1. Получаване на подобрени оценки за положителния корен на алгебрично уравнение спрямо познатите резултати в областта на финансовата математика;
- 3.2. Разработване на софтуерен продукт за спомагане взимането на инвестиционни решения чрез пресмятане на действителната стойност на парични потоци във времето.

Структура и обем на дисертационния труд

Работата се състои от увод, четири глави, заключение, перспективи за развитие, публикации по дисертационния труд и използваната литература.

Обемът на основната част е 110 страници, а използваната литература — 5 страници. Използваната литература включва 64 източника, от които 61 на латиница. Списъкът на авторските публикации се състои от 5 заглавия.

В първа глава са разгледани предимствата и особеностите при използването на рамки за разработка на приложения. Направен е кратък преглед на най-разпространените системни и приложни рамки.

Във втора глава са разгледани специфичните изисквания на бизнес приложенията, средствата за тяхното изпълнение. Описана е работената рамка и начините, чрез които постига целите си. Посочени са нейните текущи недостатъци и насоките за нейното усъвършенстване и разширяване.

В трета глава е представен прототип на обектно-ориентираната рамка. Описани са реални, внедрени приложения, разработени с помощта на рамката. Описан е потребителският интерфейс.

В четвърта глава е описан софтуерен продукт за спомагане взимането на инвестиционни решения и са намерени оценки за единствения положителен корен на алгебрично уравнение.

В заключението е направено обобщение на реализираното по така поставените цели, като се очертават и възможностите за бъдещо развитие.

Кратко съдържание на дисертационния труд

Глава I. Обектно-ориентирани рамки за разработка на приложения

СЪЩНОСТ

Изчислителните възможности и пропускателната способност на мрежите се увеличиха значително през последното десетилетие. При проектирането и реализацията на сложни софтуерни системи се запазва високата цена и нивото на направени грешки. Съществена част от разходите и усилията за разработка произлизат от продължаващото преоткриване и пресъздаване на ключови концепции и компоненти в софтуерната индустрия. От друга страна, нарастващото разнообразие на хардуерните архитектури и многообразието от операционни системи и комуникационни платформи затруднява изграждането функционални, преносими, високопроизводителни и ценово достъпни приложения без допълнителни средства [Fayad, 1997].

Обектно-ориентираните рамки за разработка на приложения са обещаваща технология за многократно използване на доказани софтуерни проекти и реализации с цел намаляване на разходите и подобряване качеството на софтуера. Рамка за разработка на приложения е преизползваемо, “полуготово” приложение, което може да се специализира в производството на потребителски приложения [Johnson, 1988].

Основните предимства на обектно-ориентираните средства за разработка на приложения произлизат от тяхната модулност, преизползваемост, разширяемост и обратен контрол, която те предоставят на разработчиците.

Предимствата, които рамките за разработка на приложения притежават, както и тяхната архитектура, зависят силно от предметната област, за която са разработени. По тази причина една подходяща класификация може да се направи според техния обхват:

- За системна инфраструктура;
- За междуслойна интеграция;
- За фирмени приложения.

Други методи за класификация на рамките за разработка на приложения може да бъде направена според техниката, чрез която се разширяват функционалните им възможности, като най-общо се обособяват два вида: така наречените бели кутии и черни кутии.

СЪВРЕМЕННИ СХВАЩАНИЯ

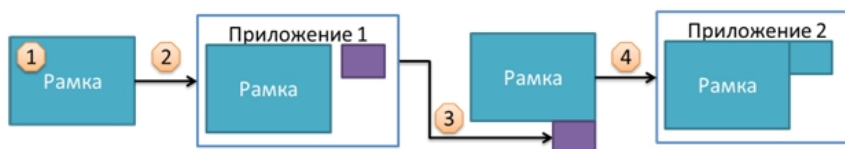
В съвременното изучаване на проблемите на рамките за разработка на приложения се очертават следните насоки за развитие и проблеми:

- Намаляване на усилията за разработка на рамки — с нарастването на броя разработени рамки в различни области и натрупания опит, може да се очаква по-ефективно развитие;
- Преместване на фокуса върху рамки за разработка на приложения в конкретни приложни области — банкиране, медицините, електронен бизнес, застрахователна дейност. [Gamma, 1995; Pree, 1994];
- Рамки “черни кутии”;
- Документация — наличието на добра документация е от изключително важно значение. В много случаи проекти се провалят, защото разработчиците не могат да предадат помежду си информацията за архитектури, алгоритми, подходи [Schmidt, 1996];
- Управление на процеса на разработка на рамките.

ИКОНОМИЧЕСКА ЕФЕКТИВНОСТ И ПРИДОБИВАНЕ

Използването на подходяща рамка за разработка на бизнес приложения ще има следните ползи:

- Съкратено време за разработка;
- Намален риск — веднъж тестван и изпитан, кодът гарантира минимизирането на програмни грешки;
- Сходни архитектури на отделните приложения — създадените приложения се поддържат и изучават по-лесно. Процесът по разработването и поддръжката на собствена рамка са показани на фигура 2.



Фигура 1. Процес на разработване и поддръжка на рамка

Инвестициите в разработването или придобиването на рамка за приложения имат добра възвращаемост, особено за областта на бизнеса и електронния бизнес.

Глава II. Авторска справка за разпределени бизнес приложения — Framework for Distributed Business Applications (FDBA)

Цели

Разработката на приложения за нуждите на бизнеса представлява значителна част от производството на софтуер. Специфичните изисквания на бизнес единиците изискват индивидуален подход и софтуерните пакети с общо предназначение не могат да задоволят напълно нуждите на клиентите. За една фирма е недопустимо да адаптира своя вътрешен начин на работа към функционалността на един програмен продукт, туй като това неминуемо води до затрудняване на дейността на самата фирма.

Динамиката е друга важна характеристика на съвременния бизнес. Фирмите активно се опитват да разширят своята дейност чрез навлизането на нови пазари и чрез предлагането на нов вид услуги на своите клиенти. Тази иновативност носи и информационни рискове за фирмите. По тази причина разработчиците на софтуер трябва да могат предоставят гъвкаво решение, което да позволява бърза адаптация и реализация постоянния поток от нови изисквания.

Това обуславя проектирането и разработката на софтуерна рамка за разработване на бизнес приложения, която да предоставя на производителите на софтуер необходимите средства за изграждане на скелета на едно бизнес приложение, като така те ще могат да се съсредоточат върху специфичните нужди на своя клиент.

Фокусът е насочен към следните области:

- Финанси, счетоводство, финансов мениджмънт;
- Фирмено управление;
- Управление на контрагенти;
- Търговия — класическа и електронна;
- Застраховане;
- Управление на складово стопанство;
- Адвокатски услуги;
- Обслужване на клиенти;

- Държавна администрация;
- Образователни институции — университети, колежи, училища;
- Образователни услуги в малки и средни предприятия.

РЕШЕНИЕ

Проектирана е софтуерна рамка за бизнес приложения със следните характеристики:

- Четирислойна архитектура;
- Изградена системна инфраструктура за двупосочен обмен на данни, който позволява изпращане на съобщения от сървъра към клиента;
- Използване на крайните приложения в локални и глобални мрежи, поддържащи протокола TCP/IP;
- Разработка на приложения без писане на изпълним код;
- Поддръжка на многоезичен потребителски интерфейс;
- Компоненти за визуализация и редактиране на данни на ниво поле;
- Компоненти за визуализация на данни на ниво таблица;
- Компоненти за търсене и филтриране на данни;
- Компоненти за редактиране на данни на ниво таблица;
- Компоненти за защита от неправилен достъп и управление на потребителите на системата;
- Компоненти за разнообразни бизнес задачи:
 - Документооборот;
 - Интеграция с Microsoft Office за създаване на документи;
 - Интеграция с Microsoft Outlook за архивиране и управление на електронна кореспонденция;
 - Изготвяне на отчети;
 - PIVOT справки.

МОДЕЛ

Създаден е модел на рамката за разпределени бизнес приложения FDBA, който се състои от следните основни елементи:

- Софтуерни компоненти;
- Речник на базата данни;
- Речник на приложение;
- Програмни модули.

Софтуерни компоненти

Дефиниран е набор от високо конфигурируеми софтуерни компоненти за изграждане на графичен потребителски интерфейс в две основни групи: визуализация на данни на ниво таблица и визуализация и редактиране на данни на ниво поле от таблица.

Компонентът за визуализация на данни на ниво таблица представлява група от визуални контроли за таблично представяне на данните от дадена таблица или изглед от базата данни, или резултатът от SQL команда. Колоните в табличния изглед се генерират сами, въз основа на метаданните на базата данни и на дефинициите в Речника на базата данни за избрания изглед. Компонентът позволява под таблицата да бъдат изведени контроли за визуализация на стойностите на допълнителни полета за избрания ред. Поддържа се: сортиране по произволна колона в възходящ и низходящ ред; пренареджане на колоните и задаване на размер на всяка от тях; търсене на запис по начален текст от дадена колона; търсене на запис по текст, съдържащ се в произволна колона; изчисляване и визуализация на обобщени резултати.

Компонентите за ниво таблица и визуализация и редактиране на данни на ниво поле от таблица обхващат различни логически и физически типове данни:

- Низ;
- Многоредов текст;
- Форматиран текст;
- Целочислен или реален числен тип;
- Дата и време;
- Период между две дати;
- Продължителност с часове и минути;
- Избор на стойност от фиксиран списък, дефиниран в Речника на базата данни;
- Избор на стойност от списък, съхранен в базата;
- Сума във валута, конвертирана до друга валута и с начисляване на добавени стойности.

Речник на базата данни

Речникът на базата данни разширява информацията, съдържаща се в метаданните на самата база данни. Речникът позволява прецизно дефиниране на логическия смисъл и правила на данните. Подобно ниво на описание не е възможно чрез използването единствено

на метаданните. Например, чрез външни ключове може да се опише релацията между две таблици, но тази информация може да бъде недостатъчна, ако логическата функционалност на софтуерното приложение изисква за определени случаи да се работи с подмножество на тези данни.

Речникът на базата данни съдържа:

- Потребителски имена на таблиците и полетата — това са имената, които крайните потребители на приложението виждат;
- Разширени логически типове на данните — това са логически типове, зад които в самата база данни стои един и същи физически тип. Например, в поле от тип `decimal` в базата данни, приложението може да запише парична стойност, процент, количество. Всеки от логическите типове се характеризира със специфичен интерфейс за въвеждане на данните и правила за валидация. Поддържаните логически типове са: пари, процент, парична стойност в различна валута, парична стойност с добавени стойности, списък от фиксирани стойности — флагове;
- Атрибути — позволяват задаване на специално значение на полетата, което се отразява в потребителския интерфейс. Например, може да се определени, че дадено поле е достъпно за редакция единствено при въвеждане на нов запис, но след това не може да бъде променяно;
- Форматиране — задава се стандартно форматиране на стойността на полето в потребителския интерфейс;
- Стандартен изглед на таблица — позволява дефинирането на един или повече подредби на полетата от таблица за визуализация и редакция в потребителски интерфейс.

Речникът на базата данни позволява на съответните модули от рамката да изградят автоматично необходимия минимален, но достатъчен и правилен потребителски интерфейс, чрез който потребителите да могат да разглеждат, филтрират, търсят, въвеждат и редактират данни. Това улеснява задачата на програмистите, на които се налага да пишат машинен код само когато е нужен сложен потребителски интерфейс за въвеждане на множество взаимосвързани данни в повече от една таблица.

Речникът на приложението

Речникът на приложението описва йерархичната структурата на потребителския интерфейс — менюта, екранни елементи, както и екранното оформление. За последното речникът на приложение разчита на езика XAML и възможностите на Microsoft WPF и Silverlight [Nathan, 2010; MacDonald, 2010; Anderson, 2010; MacDonald, 2010b]. Последното дава възможност външният вид на интерфейса да се изработи от дизайнер, който няма умения на програмист.

Речникът на приложението също така може да съдържа и дефиниции на бизнес логика, когато те са специфични за даден екран на приложението. По същество, той може да допълва или дори отменя дефиниции от речника на базата данни.

За всеки елемент от йерархичната структура, речникът на приложението съдържа следните данни:

- Основни:
 - Наименование;
 - Графичен елемент;
 - Описание;
 - Индекс в ръководството за потребителя;
 - Изпълним модул;
 - Списък със специфични привилегировани операции.
- Параметри към изпълнимия модул — това е допълнителна текстова информация във XML или XAML формат, които дефинират поведението на изпълнимия модул по време на работата на програмата.

Програмни модули

Програмните модули предоставят на разработчиците гъвкави, разширяеми подсистеми за решаване на конкретни проблеми.

За конкретните цели са идентифицирани следните модули:

- Многоезичен интерфейс;
- Управление на потребителите и сигурността;
- Изготвяне и отпечатване на отчети;
- Документооборот;
- Интеграция с Microsoft Office за създаване на документи;
- Интеграция с Microsoft Outlook за архивиране и управление на електронна кореспонденция.

АРХИТЕКТУРА

Рамката предлага четирислойна архитектура, в която са дефинирани следните слоеве:

СУБД

В конкретният случай това е MS SQL Server, версия 2005.

Сървър на приложението

Това е слой, който служи като посредник между потребителския интерфейс (клиент) и базата данни. Той предоставя на клиентските приложения универсален начин за достъп и модификация на данните посредством предварително дефинирани модели на данните, независими от физическата структура на базата данни.

По време на работа на информационните системи, сървърът предава на клиентите информация за настъпили събития и промяна на състоянието на базата данни.

Друга важна характеристика на сървъра на приложението е възможността за изпълнение във фонов режим на външни функции. Тези функции може да са реализирани в базата данни посредством съхранени процедури или като допълнителни двоични модули към приложения сървър.

Функционални обекти

Функционалните обекти са функционалното ядро на рамката. Те осигуряват същинската бизнес-функционалност на приложението. Представяват модули, които се интегрират с приложението клиент и работят в неговия контекст. Притежават собствен потребителски интерфейс, но използват средствата на приложението клиент (лента с бутони) за приемане на команди от потребителя. Разработени са преизползваеми функционални обекти за визуализация, търсене редактиране и отпечатване на данните, както и за реализация на подсистемата за сигурност на системата. Тези обекти са гъвкави и се управляват чрез параметри.

За нуждите на конкретния клиент са разработени функционални обекти за изготвяне на финансовите отчети и експорт към външен финансов софтуер. Функционалните обекти, независимо, че реализират бизнес логиката на приложението, се изпълняват на работните станции. За достъп до данните или други времеемки операции, те разчитат на сървъра на приложението.

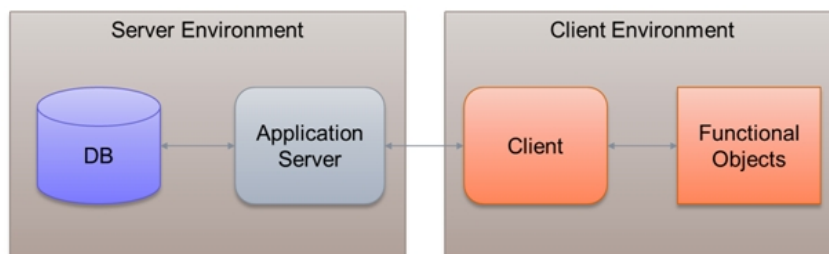
Клиент на приложението

Клиент на приложението е типичен “тънък”, който не съдържа никакъв код, свързан с директен достъп до базата данни или с бизнеслогиката на конкретното приложение. Той служи като среда за изпълнение на функционалните обекти, координиране на техните действия и свързването със сървъра на приложението. Клиентът изгражда основната форма и елементи на потребителския интерфейс на приложението, като приема командите на потребителя и ги предава на активния функционален обект. Клиентът служи и като особен посредник (проху) между функционалните обекти и сървъра на приложението.

Чрез зареждането и изпълнението на функционални обекти, приложението клиент се превръща вече във “дебел” клиент, който съдържа в себе си голяма част от функционалността на системата. По този начин се намаля мрежовия трафик и се подобрява скоростта и мащабируемостта на системата, тъй като се използват оптимално ресурсите на работните станции. Съвременното ниво на развитие на хардуера позволява дори хардуерни системи от нисък ценови клас да притежават значителни процесорна изчислителна мощност, оперативна и физическа памет.

Друга специална функция, която клиентът изпълнява, е да следи за обновени версии на използваните от приложението функционални обекти и да обнови локалните версии. По този начин се минимизират усилията за администриране и обновяване на приложението при наличие на множество работни места.

Архитектурата е илюстрирана на следната фигура:



Фигура 7. Архитектура на рамката

ТЕХНОЛОГИИ

Избрани са следните технологии:

- Microsoft NET Framework като основна библиотека за разработка;
- Windows Communication Foundation (WCF) — използва се за изграждане на необходимата двупосочна връзка между приложението клиент и приложния сървър;
- ADO.NET за връзка на приложния сървър с СУРБД;
- WPF за изграждане на графичен потребителски интерфейс [Nathan, 2010; MacDonald, 2010];
- COM — използва се за интеграция с Microsoft Office и Adobe Acrobat Reader;
- Silverlight — за разработка на многофункционален Интернет-базиран клиент [Anderson, 2010; MacDonald, 2010b; Griffiths, 2010].

Глава III. Прототип на обектно-ориентирана рамка за разработка на разпределени бизнес приложения — FDVA

СЪРВЪР НА ПРИЛОЖЕНИЕ

Сървърът на приложение предоставя на клиентските приложения редица услуги:

- Универсален начин за достъп и модификация на данните;
- Информация за структурата и подредбата на елементите на потребителския интерфейс (менюта, екрани, цветови схеми и шрифтове);
- Предава на клиентите информация за настъпили събития и промяна на състоянието на базата данни.

Друга важна характеристика на сървъра на приложението е възможността за изпълнение във фонов режим на външни функции. Тези функции може да са реализирани в базата данни посредством съхранени процедури или като допълнителни двоични модули към приложния сървър. Поддържат се два режима на изпълнение:

- При поискване от страна на приложението клиент. Този режим позволява прехвърлянето на времеемки операции на приложния сървър. Така може да се повиши производителността

на системата, като подобни операции се изпълняват на сървърна машина, а не на по-слабите откъм изчислителна мощ работни станции;

- Според дефиниран времеви график. Този режим позволява изпълнението на фонові операции на определено време за решаване на конкретни задачи. Например, с такава операция се изготвя дневен експорт на данни към външен счетоводен софтуер.

Сървърът на приложение е реализиран като сервисно приложение (Windows Service Application). Той поддържа готов списък от предварително създадени нишки на изпълнение, чрез които обслужва заявките от клиентските приложения.

Сървърът е проектиран така, че да може да обслужва повече от една крайна информационна система. По този начин се опростява инсталацията и обслужването на системите.

Специално внимание се отделя на подсистемата за изпращане на съобщения към клиентските приложения. Тя следи за състоянието на всяко клиентско приложение, за да се избегне изпращането на съобщения към клиенти, които по технически прилични (например, загуба на връзка по мрежата) не могат своевременно да обработят събитието.

Като допълнение към сървъра е разработен специализиран инструмент за мониторинг и управление на текущото състояние на системата. Инструментът позволява на системните администратори да изпълняват следните операции:

- Да стартират и спират сървъра;
- Да следят натовареността на системата и активността на всеки потребител;
- Да прекратяват достъпа до системата на един или повече потребители;
- Да проследяват изпълнението на външните функции във фонов режим;
- Да управлява конфигурацията.

КЛИЕНТ

Клиент на приложението е типичен “тънък”, който не съдържа никакъв код, свързан с директен достъп до базата данни или с бизнес-логиката на конкретното приложение. Той служи като среда за изпълнение на функционалните обекти, координиране на техните действия и свързването със сървъра на приложението. Клиентът изгражда основната форма и елементи на потребителския интерфейс на приложението, като приема командите на потребителя и ги предава на активния функционален обект. Клиентът служи и като особен посредник (ргоху) между функционалните обекти и сървъра на приложението.

Клиентът съдържа следните визуални елементи:

- Навигационно меню. То е изградено по модел на същото на програмата Microsoft Outlook. Опитът показва, че потребителите познават последната програма и лесно се ориентират в интерфейса на информационните системи, разработени с FDBA;
- Лента с инструменти, реализирана по модел на Microsoft Ribbon. Лентата е динамична и се променя според текущо изпълнявания функционален обект;
- Контейнер на функционални обекти. Контейнерът поддържа стек от функционални обекти, като винаги има само един активен и това е най-горният такъв;
- Лента за допълнителна информация. Нейното съдържание се определя според ролята на активния функционален обект.

С цел подобряване на бързодействието, приложението клиент построява и поддържа временно копие в паметта на текстовете от потребителския интерфейс. Тези текстове се зареждат еднократно при стартиране.

Клиентът е изграден чрез технологията WPF по начин, който да позволява лесно преминаване към Silverlight. Това е постигнато чрез спазване на съответните синтактични изисквания към подмножеството на езика XAML, което се поддържа и от двете платформи.

Специално внимание е отделено на стабилността на системата при интеграция с външни системи [Pavlov, 2010].

ФУНКЦИОНАЛНИ ОБЕКТИ

Функционалните обекти са функционалното ядро на FDBA. Те осигуряват същинската бизнес-функционалност на приложението. Представяват модули, които се интегрират с приложението клиент и работят в неговия контекст. Притежават собствен потребителски интерфейс, но използват средствата на приложението клиент (лента с бутони) за приемане на команди от потребителя. Разработени са преизползваеми функционални обекти за визуализация, търсене редактиране и отпечатване на данните, както и за реализация на подсистемата за сигурност на системата. Тези обекти са гъвкави и се управляват чрез параметри.

Функционалните обекти предоставят програмен интерфейс за обмен на данни помежду си, макар този обмен да се извършва през контекста на приложението клиент. Например, даден функционален обект може да иска да знае какви данни са заредени в предхождащия го такъв, за да филтрира своите резултати по подходящ начин.

Функционалните обекти се реализират като външни модули, които се зареждат динамично по време на изпълнение.

ИНСТРУМЕНТИ ЗА РАЗРАБОТКА

С рамката са проектирани и разработени следните инструментални средства за разработване на приложения чрез самата рамка:

Data Dictionary Manager

Data Dictionary Manager (DDM) е инструмент за създаване и редактиране на речника на базата данни. Притежава следните подсистеми:

- Редактор на речника на базата данни — подсистема за редактиране съдържанието на речника чрез базиран на XML текстови редактор с разширени възможности за оцветяване на текста за по-добро възприятие, използване на шаблони и проверка валидността на съдържанието;
- Система за управление на версиите, описана в следващ параграф.

Application Dictionary Manager

Application Dictionary Manager (ADM) е инструмент за създаване и редактиране речника на приложението. Притежава следните подсистеми:

- Редактор на йерархичната структура на приложението — чрез него се изгражда дървовидната структура на навигационното меню;
- Редактор на основни параметри — чрез него се редактират основните параметри на всеки елемент от йерархичната структура на приложението, като: наименование, графични визуални елементи, изпълним модул, специални права за достъп, индекс в помощната информация;
- Редактор на параметрите към изпълнимия модул — това е базиран на XML текстови редактор с разширени възможности за оцветяване на текста за по-добро възприятие, използване на шаблони и проверка валидността на съдържанието. При XAML параметри се поддържа възможност за визуализация на XAML съдържанието, което ще се генерира;
- Система за управление на версиите, описана в следващ параграф.

Система за управление на версиите

Тази система е вградена в DDM и ADM. Тя предоставя на разработчиците автоматично средства да създават версии на двата речника и да следят за промените и конфликтите. Системата може да прихваща промени и конфликти при сливане на версии.

Text Editor

Text Editor (TE) е инструмент за въвеждане и редактиране на текстовете на потребителския интерфейс на всички езици, които ще се поддържат от крайното приложение. Инструментът поддържа информация за версията на съдържанието. Тя е нужна, за да може автоматично да се определя актуалното съдържание при сливане на версии.

Report Editor

Report Editor (RE) е инструмент за създаване и редактиране на потребителски справки. Инструментът е едновременно част от крайното приложение, както и средство за разработка. Притежава следните подсистеми:

- Редактор за справките към всеки елемент от йерархичната структура на приложението, който позволява:
 - Въвеждане, редактиране и изтриване на справки;
 - Задаване на права на достъп до справките;

- Асоцииране на справки с принтер по подразбиране;
- Асоцииране на справки с конкретен език на потребителя.
- Редактор на заявки към базата данни;
- Редактор на изгледа на справка, който предоставя WYSIWYG визуални инструменти, подобни на Microsoft Word, версия XP.

Чрез инструментите за разработка, рамката се превръща в цялостна платформа.

ЕФЕКТИ

Изследвани са следните ефектите от използване на рамката:

1. Оптимизация на времето за разработка;
2. Оптимизация на качеството на крайния продукт;
3. Оптимизация на цялостните разходи за внедряване.

Изследвани са реални задачи и са сравнени процесите по разработка и внедряване на FDVA спрямо стандартен подход за разработка без използване на специализирана рамка. Чрез FDVA е възможно да се съкратят разходите до 10 пъти.

ВНЕДРЯВАНЕ

С помощта на рамката са разработени и внедрени софтуерни приложения в различни области — застраховане, юридически кантори, софтуерни фирми, образование. Първите версии на някои от тях са внедрени още през 2002 г.

Dutch P&I Navigator

Приложението е предназначено за автоматизация дейността на застрахователен брокер за корабно застраховане. Системата обхваща цялостната дейност на фирмата.

Системата е въведена е в реална експлоатация през 2002 година в холандска фирма. Благодарение на възможностите и характеристиките на рамката, развитието на продукта през всичките тези години е било възможно в кратки срокове, високо качество и приемлива за клиента цена. В края на 2010 г. приложението е внедрено в четири офиса на фирмата клиент и се ползва едновременно от над 60 потребителя.

Distributed e-Testing Cluster

Distributed e-Testing Cluster (DeTC) [Rahneva, 2004; Rahnev, 2005; Pavlov, 2010b] представлява отворена инфраструктура, която предлага услуги за електронно тестване, локализирани върху различни възли, с възможност за частична и автоматично контролирана интеграция в рамките на предварително дефинирани виртуални структури. DeTC е разработен като многослойно приложение [Rahnev, 2005], като значителна част от модулите в DeTC са базирани на рамката за разпределени приложения FDVA.

Capital Budgeting — анализ на инвестиционни потоци

Тази информационна система е предназначен за анализ на инвестиционни проекти, чрез прилагане на различни модели за избор на инвестиционно решение. В нея се прилагат изводите от анализа на финансови потоци за пресмятане на вътрешна норма на възвръщаемост, разгледан в глава IV.

Информационната система е разработена чрез FDVA и може да послужи за демонстрация, че FDVA позволява да бъде разширявана и със сложни изчислителни модули. Повече информация за системата е изложена в глава IV.

Глава IV. Анализ на инвестиционни проекти

Инвестициите са един от най-важните фактори за икономически просперитет на всяка икономическа единица [Oresharski, 1997]. Това поставя проблемите за избор на активи, в които да се инвестира, за избор на източник на капитали за реализация на инвестиционните проекти, за правилно оценяване на възвръщаемостта на проектите в конкретната икономическа среда, за откриване на подценени активи и други [Oresharski, 1997].

ИНФОРМАЦИОННА СИСТЕМА CAPITAL BUDGETING

С помощта на FDVA е разработена информационна система, която може да се използва за извършване на изчисленията, свързани с оценяването на инвестиционни проекти. Системата няма за цел да представлява изчерпателно решение на проблематиката, а да бъде инструмент за подпомагане процеса на взимане на управленски инвестиционни решения.

Информационната система за инвестиционен анализ предлага изчисляване на следните показатели за оценка на инвестиционни проекти:

- Нетна сегашна стойност;
- Нетна бъдеща стойност;
- Вътрешна норма на възвръщаемост;
- Модифицирана вътрешна норма на възвръщаемост;
- Срок на възвръщаемост на инвестициите;
- Коефициент на доходност;
- Анюитетен метод;
- Счетоводна норма на доходност;
- Способност на обслужване на дълга.

Ключова функция на този продукт е възможността за класиране на алтернативни (взаимоизключващи се) проекти. Анализът се осъществява чрез комплексна оценка на описаните методи, като приоритет имат методите, основаващи се на принципа за времевата стойност на парите.

Последните години като водещи показатели за оценка на инвестиционни проекти се налагат нетната сегашна стойност и вътрешната норма на възвръщаемост. Вътрешната норма на възвръщаемост (Internal Rate of Return, или IRR) е онази норма на дисконтиране, която изравнява алгебричната сума на дисконтираните положителни и отрицателни парични потоци на разглежданата инвестиция [Oresharski, 1997; Sharpe, 1985; Mayo, 1988; Mishkin, 1992].

Изчисляването на вътрешна норма на възвръщаемост се извършва чрез прилагането на получените в дисертацията оценки и алгоритъм.

За целите на инвестиционния анализ полиномното уравнение за намиране на вътрешната норма на възвръщаемост е:

$$P(x) = x^n - \sum_{i=1}^n a_i x^{n-i} = 0. \quad (4.5)$$

Различни оценки за единствения корен σ на уравнение (4.5) могат да се намерят в монографиите на Herzberger [Herzberger, 1999], Kyurkchiev [Kyurkchiev, 1998], Sendov, Andreev, Kyurkchiev [Sendov, 1993], както и в някои базови статии, например, Petkovic, M., Petkovic, L. [Petkovic, 1989], Kyurkchiev, Herzberger [Kyurkchiev, 1993] и Hristov, Kyurkchiev [Hristov 1999].

Локализационни оценки за всички положителни корени на даден алгебричен полином могат да се намерят в Сендов [Sendov, 1974] и Кууркчиев [Kuurkchiev, 1991].

Ще се приведат някои нови оценки за единствения корен σ на алгебричен полином.

Още някои известни факти, свързани с името на Deutsch [Deutsch, 1982]:

Теорема А. Нека $A = (a_{ij})$ е $n \times n$ неизродена матрица и нека $x = (x_1, \dots, x_n)^T$ и $y = (y_1, \dots, y_n)^T$ са вектори удовлетворяващи матричните уравнения

$$\begin{aligned} Ax &= Dx, \\ A^T y &= Dy, \end{aligned}$$

за някоя диагонална матрица

$$D = \text{diag}(d_1, d_2, \dots, d_n).$$

Ако x не е Perron вектор на A , то

$$\rho(A) > \frac{y^T Dx}{y^T x}, \quad (4.6)$$

където $\rho(A)$ е спектралния радиус на матрицата A .

Теорема Б. При същите означения като в Теорема А е в сила

$$\rho(A) \geq \prod_{i=1}^n d_i \frac{d_i x_i y_i}{y^T Dx} \geq \frac{y^T Dx}{y^T x}. \quad (4.7)$$

Теорема В. При същите означения като в Теорема А е в сила

$$\rho(A) \geq t - \prod_{i=1}^n (t - d_i) \frac{x_i y_i}{y^T x} \geq \frac{y^T Dx}{y^T x}, \quad (4.8)$$

за всяко

$$t > \rho(A) + \max_{1 \leq i \leq n} (d_i - a_{ii}).$$

Забележка 1.

$$t = \beta + \max_{1 \leq i \leq n} (d_i - a_{ii})$$

е долна оценка за спектралния радиус $\rho(A)$.

Някои от възможните стойности за β са:

- 1) максималната сума от елементи на A (по редове);
- 2) максималната сума от елементи на A (по стълбове);
- 3) $\max_i d_i$.

Основен резултат

Необходимо е да се отбележи, че познатите оценки за положителния корен σ , базиращи се на Теорема А - В са валидни при частния избор на вектора $x = (1, 1, \dots, 1)^T$.

Първата оценка за корена σ , базираща се на Теорема Б с избор на произволен вектор от вида $x_i = \lambda^i$, $i = 1, \dots, n$, $\lambda > 0$ може да се намери в работата на Hristov и Kyurkchiev [Hristov 1999]:

$$\sigma \geq \frac{1}{\lambda} (q(\lambda)) \frac{q(\lambda)}{\lambda q'(\lambda)}, \quad (4.9)$$

където

$$q(\lambda) = \sum_{i=1}^n a_i \lambda^i.$$

Очевидно,

$$0 = P(\sigma) = \sigma^n - \sum_{i=1}^n a_i \sigma^{n-i} = \sigma^n \left(1 - q\left(\frac{1}{\sigma}\right) \right),$$

т.е.

$$q\left(\frac{1}{\sigma}\right) = 1.$$

Забележка 2. От горното равенство веднага може да се съобрази, че ако изберем параметъра λ да бъде близък до величината $\frac{1}{\sigma}$, тогава оценките от тип (4.9) ще бъдат много прецизни.

Ще получим оценка за корена σ , базираща се на Теорема В на Deutsch при произволен положителен вектор x .

В сила е следната

Теорема 1 [Pavlov, 2010c]. Нека

$$a_k \geq 0, \quad k = 2, \dots, n,$$

$$\sum_{i=1}^n a_i x_i \geq \left(\frac{x_1^n}{x_n} \right)^{\frac{1}{n-1}}.$$

При произволни x_1, \dots, x_n , за положителния корен σ на уравнение (4.5) е в сила следната оценка:

$$\begin{aligned}
\sigma \geq & a_1 + \frac{2}{x_1} \sum_{i=2}^n a_i x_i - \left(\frac{1}{x_1} \sum_{i=2}^n a_i x_i \right)^{\frac{x_1}{x_1 + \sum_{i=2}^n \mu_i a_i x_i}} \times \\
& \frac{\frac{x_2}{x_1} \sum_{i=2}^n a_i x_i}{\frac{1}{x_1} \sum_{i=2}^n a_i x_i} \times \\
& \times \left(a_1 + \frac{2}{x_1} \sum_{i=2}^n a_i x_i - \frac{x_1}{x_2} \right)^{\frac{x_2}{x_1 + \sum_{i=2}^n \mu_i a_i x_i}} \times \\
& \frac{\frac{x_n}{x_{n-1}} \sum_{i=n}^n a_i x_i}{\frac{1}{x_{n-1}} \sum_{i=n}^n a_i x_i} \times \\
& \times \dots \times \left(a_1 + \frac{2}{x_1} \sum_{i=2}^n a_i x_i - \frac{x_{n-1}}{x_n} \right)^{\frac{x_n}{x_1 + \sum_{i=2}^n \mu_i a_i x_i}},
\end{aligned} \tag{4.10}$$

където

$$\mu_i = \sum_{k=1}^{i-1} \frac{x_{k+1}}{x_k}, \quad i = 2, 3, \dots, n.$$

Специален избор на параметрите

Векторът $x = (x_1, x_2, \dots, x_n)^T > 0$ може да се избира по произволен начин.

На първо място, ако се избере

$$x_i = \lambda^i, \quad i = 1, 2, \dots, n, \quad \lambda > 0$$

се получава следното

Следствие 1. За произволно положително λ за положителния корен σ на полинома $P(x)$ е в сила следната оценка:

$$\sigma \geq a_1 + 2G - \sqrt[F]{G \left(a_1 + 2G - \frac{1}{\lambda} \right)^{F-1}}, \tag{4.14}$$

където

$$G = \sum_{i=2}^n a_i \lambda^{i-1},$$
$$F = a_1 + \sum_{i=2}^n (i-1) a_i \lambda^i.$$

Отбелязваме, че оценката отдолу за σ е прецизна.

Забележка 3. Очевидно, ако параметърът λ се избере да бъде близък до $\frac{1}{\sigma}$, тогава оценката (4.14) ще бъде по-точна.

Друг възможен специален избор на x_i е следния

$$x_{i_k} = \frac{a_{i_k}^{\mu_k - 1}}{a_{i_1}^{\mu_1} + a_{i_2}^{\mu_2} + \dots + a_{i_m}^{\mu_m}},$$
$$k = 1, 2, \dots, m,$$

където $0 < x_i$ са произволни за $i \neq i_1, i_2, \dots, i_m$, и $\mu_1, \mu_2, \dots, \mu_m$ са числа, подлежащи на допълнителен избор.

Забележка 4. С помощта на разглежданията могат да се получат експлицитни оценки за единствения положителен корен в термините на финансовата математика.

В дисертацията е приведен и интересен числен експеримент.

КОМПЮТЪРНО ПРЕСМЯТАНЕ НА ОЦЕНКАТА

Получената оценка (4.14) се изчислява в разработената информационна система чрез интегрирания във FDBA алгоритъм:

```
Double a[ ];
Double ComputeSigma(double lamdba)
{
  G = 0;
  F = a[1];
  l = lamdba;
  For (i = 2 to Length(a))
  {
    G += a[i] * l;
    l *= l;
    F += (i-1) * a[i] * l;
  }
}
```

```

    Result = a[1] + 2 * G -
        Power(G, 1 / F) *
        Power (a[1] + 2 * G - 1 / lamda, (F-1) / F);
}

Double CalcIRR ( )
{
    a = InitData();
    Result = 1;
    For (i = 1 to 30)
    {
        Result = ComputeSigma(1 / Result);
    }
}

```

Перспективи

Бъдещото развитие на рамката може да се разгледа в следните аспекти:

1. Изграждане на уеб-базиран потребителски интерфейс чрез технологията Microsoft Silverlight;
2. Проектиране и изграждане на приложен програмен интерфейс за интерактивен обмен на данни с външни системи, съответстващ на динамичния характер на рамката и разработените чрез нея приложения;
3. Подобряване на мащабируемостта на приложенията чрез изграждане на възможности за хоризонтално разширяване на ресурсите;
4. Разширяване на приложните услуги на рамката с възможности за едновременно въвеждане на данни в две или повече свързани помежду си таблици от базата данни чрез атомарна операция;
5. Разширяване на приложните услуги на рамката за поддръжка на сложни типове данни, специфични за различни приложни области, като:
 - а. Счетоводни контировки;
 - б. Ръчно изписани формули.

Апробация

Резултати, получени от изследването, са използвани в следните международни, национални и университетски проекти:

- Национален проект към МОН “Създаване на международен център за електронно обучение DeLC”, договор N 107/ВУ-МИ 2005.

Част от резултатите, получени в дисертационния труд, са докладвани на следните национални и международни конференции и семинари:

- Microsoft Academic Days in Central and Eastern Europe, Praha, Czech Republic, 2004;
- 4th .NET Technologies International Conference, Plzen, Czech Republic, 2006;
- Mathematics in Industry Conference, Sofia, Bulgaria, 11-14 July 2010;
- Research and Education in Mathematics, Informatics and their Applications Conference, 2010, Plovdiv, Bulgaria.

Авторска справка

В дисертационната работа е направен преглед на съществуващите софтуерни рамки и технологии за разработка на разпределени приложения. Специално внимание е отделено на съществуващи софтуерните рамки за разработка на разпределени бизнес информационни системи. Изпълнени са задачи 1.2 и 2.1.

Основните приноси на дисертационния труд са:

- I. Анализирани са ефектите от използване на софтуерни рамки при разработка на информационни системи и са получени емпирични резултати;
- II. Проектирани са архитектурата, услугите и функционалността на обектно-ориентирана рамка за изграждане на разпределени бизнес приложения – FDVA;
- III. Създаден е модел на рамка за разпределени бизнес приложения;
- IV. Разработен е прототип на рамката FDVA;
- V. Разработени са и са внедрени в експлоатация пет крайни приложения, чрез което е доказана приложимостта на рамката FDVA;
- VI. Разработена е информационна система за анализ на инвестиционни проекти;

VII. Получени са теоретични оценки за единствения положителен корен на даден алгебричен полином, които могат да се прилагат в областта на финансовата математика за провеждане на свръхчувствителен анализ на инвестиционни проекти.

Връзките между приносите, целите, задачите, мястото на описание в дисертационния труд и направените публикации са следните:

Принос	Цел	Задачи	Параграф	Публикации
I	A	1.1	1.5, 3.3	2, 3
II	A	1.3, 2.2, 2.3, 2.4	2.2	2, 3
III	A	1.4	2.1	2, 3
IV	B	2.5	3.1, 3.2	2, 3, 4
V	B	2.6	3.5	1, 2, 3
VI	B	3.2	4.1	
VII	B	3.1	4.2	5

Благодарности

Издавам най-сърдечна благодарност на научните си ръководители проф. д-р Асен Рахнев и доц. Николай Кюркчиев.

Публикации по дисертационния труд

1. Rahneva O., Rahnev A., Pavlov N., Functional Workflow and Electronic Services In a Distributed Electronic Testing Cluster — DeTC, Proceedings 2nd International Workshop on eServices and eLearning, Otto-von-Guericke Universitaet Magdeburd, 2004, pp 147–157.

2. Rahnev A., Pavlov N., Rahneva O., Architecture & Design of Distributed Electronic Testing Cluster (DeTC) based on Microsoft .NET Framework — IMAPS CS International Conference 2005, September 15-16, 2005, Brno, Czech Republic, pp 417–422.

3. Pavlov N., Rahnev A., Architecture and Design of Customer Support System using Microsoft .NET technologies, .NET Technologies 4th International Conference, May 29 — June 1 2006, Plzen, Czech Republic, Short Papers Proceedings ISBN 80-86943-11-9, pp 21–26.

4. Pavlov N., Custom Surrogate Host for ActiveX In-Process Servers, Proceedings of the Anniversary International Conference REMIA 2010, 10–12 December, 2010, pp 287–294

5. Pavlov N., Valchanov V., Kyurkchiev N., Bounds for the Unique Positive Root of a Polynomial in Mathematics of Finance, Scientific Works of Plovdiv University "Paisii Hilendarski", vol 37, book 3, 2010, pp 101–110.

Литература

- [Anderson, 2010] Anderson, Chris, Pro Business Applications with Silverlight 4, APress, 2010.
- [Deutsch, 1982] Deutsch, E., Lower bounds for the Perron root of a non-negative irreducible matrix, Math. Proc. Cambridge Philos. Soc. v. 92, 1982, pp. 49–54.
- [Fayad, 1997] Mohamen Fayad, Douglas C. Schmidt. Object-oriented frameworks, Communications of the ACM, Special Issue on Object-Oriented Application Frameworks, Vol. 40, No. 10, October 1997.
- [Gamma, 1995] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, Design Patterns: Elements of Reusable Software Architecture, Addison-Wesley, 1995.
- [Griffiths, 2010] Griffiths, Jan, Adams, Matthew, Liberty Jesse, Programming C# 4.0: Building Windows, Web, and RIA Applications for the .NET 4.0 Framework, O'Reilly Media 2010.
- [Herzberger, 1999] Herzberger, J., Einführung in die Finanzmathematik, Wissenschaftsverlag GmbH, Oldenbourg, 1999.
- [Hristov 1999] Hristov, V., N. Kyurkchiev, Bounds end numerical methods for unique positive root of a polynomial, Mathematica Balkanica, New Series, v. 13, (3-4), 1999, pp. 399–408.
- [Johnson, 1988] Ralph Johnson, Brian Foote, Designing Reusable Classes, Journal of Object-Oriented Programming. SIGS, 1, 5 (June/July. 1988), 22-35.
- [Kyurkchiev, 1991] Kyurkchiev, N., A note on a method for localization of the roots of algebraic equations, C. R. Acad. Bulg. Sci. (1991), 5–7.
- [Kyurkchiev, 1993] Kyurkchiev, N., J. Herzberger, On some bounds for polynomial roots obtained when determining the R order of iterative processes, Serdica, v. 19, 1993, pp. 53–58.
- [Kyurkchiev, 1998] Kyurkchiev, N., Initial approximation and root finding methods, WILEY-VCH Verlag Berlin GmbH, Berlin, v. 104, 1998.
- [MacDonald, 2010] MacDonald, Mathew, Pro WPF in C#, Apress, 2010.
- [MacDonald, 2010b] MacDonald, Matthew, Pro Silverlight 4 in C#, Apress, 2010.

- [Mayo, 1988] Mayo, Investments: An introduction, The Dryden Press, 1988.
- [Mishkin, 1992] Mishkin, Fr., The Economics of Money, Banking and Financial Markets, Harper Collins Publishers, 1992.
- [Nathan, 2010] Nathan, Adam, WPF 4 Unleashed, Sams, 2010.
- [Oresharski, 1997] Орешарски, Пл., Анализ и управление на инвестициите, ИК “Люрен”, 1997.
- [Pavlov, 2006] Pavlov N., Rahnev A., Architecture and Design of Customer Support System using Microsoft .NET technologies, .NET Technologies 4th International Conference, May 29 — June 1 2006, Plzen, Czech Republic, Short Papers Proceedings ISBN 80-86943-11-9, pp 21–26.
- [Pavlov, 2010] Pavlov N., Custom Surrogate Host for ActiveX In-Process Servers, Anniversary International Conference REMIA 2010, 10-12 December 2010, Plovdiv, Bulgaria, ISBN 978-954-423-648-9, pp. 287–294.
- [Pavlov, 2010b] Pavlov N., Valchanov V., Kyurkchiev N., Bounds for the Unique Positive Root of a Polynomial in Mathematics of Finance, Scientific Works of Plovdiv University “Paisii Hilendarski”, vol 37, book 3, 2010.
- [Petkovic, 1989] Petkovic, M., L. Petkovic, On the bounds of the R-order of some iterative methods, ZAMM, v. 69, 1989, pp. 197–198.
- [Pree, 1994] Wolfgang Pree, Design Patterns for Object-Oriented Software Development, Addison-Wesley, Reading, MA, 1994.
- [Rahnev, 2005] Rahnev A., Pavlov N., Rahneva O., Architecture & Design of Distributed Electronic Testing Server (DeTC) based on Microsoft .NET Framework — IMAPS CS International Conference 2005, September 15-16, 2005, Brno, Czech Republic, pp 417–422.
- [Rahneva, 2004] Rahneva O., . Rahnev A, N. Pavlov, Functional Workflow and Electronic Services In a Distributed Electronic Testing Cluster — DeTC, Proceedings 2nd International Workshop on eServices and eLearning, Otto-von-Guericke Universitaet Magdeburd, 2004, pp 147–157.
- [Schmidt, 1996] Douglas C Schmidt, Ralph E. Johnson, Mohamed Fayad, Special Issue on Patterns and Pattern Languages, The Communications of ACM, Vol. 39, No. 10, October 1996.
- [Sendov, 1974] Сендов, Бл., Один метод одновременного приближенного вычисления всех положительных корней алгебраического уравнения. Изв. ВУЗ, 1974.
- [Sendov, 1993] Sendov, Bl., A. Andreev, N. Kyurkchiev, Numerical solution of polynomial equations, Elsevier Sci. Publ. North-Holland, 1993.
- [Sharpe, 1985] Sharpe, W., Investments, Prentice-Hall, 1985.