**АНОТАЦІЯ**

Лизун П.Р. Розробка та дослідження моделювання бізнес-процесів систем адміністрування конференцій з використанням Windows Workflow Foundation. Наукова робота магістра за спеціальністю 8.01010401 «Професійна освіта. Комп’ютерні технології». – Луцький національний технічний університет. – Луцьк, 2014. – 101 с. – (на правах рукопису).

Наукова робота магістра присвячена проблемам побудови та обробки моделей бізнес-процесів у системі організації конференції з використанням Windows Workflow Foundation 4.

В першому розділі дипломної роботи розглянуто концепцію бізнес процесів, описано стандарти і методології моделювання бізнес-процесів, висвітлено поняття роботопотоку, проаналізовано основні переваги використання роботопотоків.

В другому розділі розглянуто моделювання потоків робіт та їх опис.

В третьому розділі розроблено веб-додаток системи адміністрування конференцій та проведено огляд існуючих систем адміністрування конференцій, описано функціональні можливості системи, виконано проектування потоків робіт, створено діаграму станів об'єктів, переведено діаграму стану в таблицю станів, складено список змінних стану, реалізовано веб-додаток на основі розроблених моделей, розроблено базу даних конференцій, створено дизайн веб-додатку.

В четвертому розділі розглянуто якість програмного забезпечення та аналіз дослідження сайту.

В п’ятому розділі розглянуто питання охорони праці та безпеки в надзвичайних ситуаціях.

Ключові слова: *бази даних, інформаційні технології, інтернет, інтерфейс, інформаційно–довідкові системи, обладнання, сервер.*

ЗМІСТ

[ВСТУП 6](#_Toc391275564)

[РОЗДІЛ 1](#_Toc391275565) [АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 8](#_Toc391275566)

[1.1. Концепція бізнес процесів 8](#_Toc391275567)

[1.2. Стандарти і методології моделювання бізнес-процесів 9](#_Toc391275568)

[1.3. Поняття роботопотоку 12](#_Toc391275569)

[РОЗДІЛ 2](#_Toc391275570) [ОПИС РІШЕННЯ ЗАГАЛЬНОЇ ПРОБЛЕМИ ТА ОСНОВНИХ МЕТОДІВ ДОСЛІДЖЕННЯ 17](#_Toc391275571)

[2.1. Моделювання потоків робіт та їх опис. Динамічна інформаційна модель 17](#_Toc391275572)

[2.2. Управління потоками робіт 20](#_Toc391275573)

[2.3. Основи роботи Windows Workflow Foundation 25](#_Toc391275574)

[РОЗДІЛ 3](#_Toc391275575) [РОЗРОБКА ВЕБ-ДОДАТКУ СИСТЕМИ АДМІНІСТРУВАННЯ КОНФЕРЕНЦІЙ 47](#_Toc391275576)

[3.1. Огляд існуючих систем адміністрування конференцій 47](#_Toc391275577)

[3.2. Опис і функціональні можливості системи 49](#_Toc391275578)

[3.3. Проектування потоків робіт 50](#_Toc391275579)

[3.4. Створення діаграми станів об'єктів 53](#_Toc391275580)

[3.5. Переведення діаграми стану в таблицю станів 55](#_Toc391275581)

[3.6. Складання списку змінних стану 57](#_Toc391275582)

[3.7. Реалізація веб-додатку на основі розроблених моделей 58](#_Toc391275583)

[3.8. Створення бази даних конференцій 62](#_Toc391275584)

[3.9. Створення дизайну веб-додатку 63](#_Toc391275585)

[РОЗДІЛ 4](#_Toc391275586) [ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА ДОСЛІДЖЕННЯ 65](#_Toc391275587)

[4.1. Оцінка якості програмного забезпечення 65](#_Toc391275588)

[4.2. Еспериментальна оцінка 70](#_Toc391275589)

[РОЗДІЛ 5](#_Toc391275593) [ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ 86](#_Toc391275594)

[ВИСНОВКИ 92](#_Toc391275595)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 94](#_Toc391275596)

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ**

API ([англ.](http://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) Application Programming Interface, API) – Прикладниий програмний інтерфейс;

AS ([англ.](http://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) ActionScript) – Скриптова мова програмуваня, що дозволяє запрограмовувати [Adobe Flash](http://uk.wikipedia.org/wiki/Adobe_Flash)–кліпи та додатки;

CSS ([англ.](http://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) Cascading Style Sheets ) – Каскадні таблиці стилів;

DBI ([англ.](http://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) Data Base Interface) – Інтерфейс баз даних;

NET Framework – програмна технологія, запропонована фірмою [Microsoft](http://uk.wikipedia.org/wiki/Microsoft) як платформа для створення програм та веб-застосувань;

SADT (англ. Structured Analysis and Design Technique) – метод структурного аналізу і проектування;

SDK ([англ.](http://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) software Development Kit) — набір із засобів розробки, утиліт і документації, який дозволяє [програмістам](http://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%96%D1%81%D1%82) створювати [прикладні програми](http://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%B5_%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F) за визначеною [технологією](http://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D1%85%D0%BD%D0%BE%D0%BB%D0%BE%D0%B3%D1%96%D1%8F) або для певної платформи;

UML (англ. Unified Modeling Language) – уніфікована мова моделювання;

WCF – Windows Communication Foundation;

WfMC (англ. Workflow Management Coalition) – міжнародна організація розробників, впроваджувальників, консультантів, дослідників, освітніх установ і користувачів Workflow-рішень;

WFS (англ. Work Flow system) – програмне забезпечення і система технологій управління діловими процесами;

WPF – Windows Presentation Foundation;

WSH (англ. Microsoft Windows Scripting Host) – компонент Microsoft Windows, призначений для запуску сценаріїв на скриптових мовах JScript і VBScript, а також і на інших додатково встановлених мовах (наприклад, Perl);

WWF (англ. Windows Workflow Foundation) – підсистема в складі [.NET Framework](http://uk.wikipedia.org/wiki/.NET_Framework_3.0), орієнтована на візуальне програмування;

НДР – науково-дослідна робота;

ПЕОМ – персональна електронно-обчислювальна машина.

**ВСТУП**

**Актуальність теми.** Використання бізнес-процесів у системі організації конференцій дозволить зробити ефективнішим процес взаємодії між додатком та користувачами, надасть змогу контролювати різноманітні події та зміни.

Але постає інше питання, як ефективніше автоматизувати всі процеси, підвищити контрольованість та передбаченість усіх можливих дій, покращити процес взаємодії між користувачем та додатком, ці питання можна вирішити, якщо використовувати бізнес-процеси, як схему взаємодії.

Організація конференції – діяльність без якої не обходиться робота великої кількості компаній та державних структур, ця робота займає досить багато часу та ресурсів, тому постає проблема автоматизації та прискорення процесу виконання операцій, що проводиться при організації конференції. Вирішенням даної проблеми є веб-додаток, що дозволить покращити ефективність взаємодії з учасниками конференції, а також буде сприяти налагодженню роботи з матеріалами конференції.

**Метою наукової роботи магістра** є дослідження принципів моделювання бізнес-процесів та їх опис з використанням Windows Workflow Foundation та практична реалізація веб-додатку “Організація конференції”.

Досягнення поставленої мети передбачає розв’язання наступних **завдань**:

* вивчити методологію моделювання бізнес-процесів;
* дослідити принципи розробки веб-додатків з використанням Windows workflow foundation;
* здійснити теоретичний аналіз літератури з даної теми;
* провести аналіз бізнес-процесів, що виконуються при організації конференції;
* ознайомитися з принципами опису робочих процесів використовуючи можливості Windows workflow.foundation 4;
* розробити веб-додаток «Організація конференції»;
* здійснити аналіз розробленого додатка на якість роботи та провести статистичний, регресійний аналізи його роботи;
* розглянути питання охорони праці та безпеки в надзвичайних ситуаціях.

**Об’єктом дослідження** роботи є методи моделювання, побудови та обробки моделей бізнес-процесів у системі організації конференції з використанням Windows workflow.foundation 4.

**Предметом досліджень** є використання бізнес-процесів, як схеми взаємодії між користувачем та додатком, для створення модулю “Організація конференції”.

**Методи дослідження:**теоретичний опис та практичне впровадження інформаційної системи.

**Наукова новизна одержаних результатів** полягає в тому, що:

- обгрунтовано та виконано узагальнений опис, аналіз використання бізнес-процесів, як схеми взаємодії між користувачем та додатком;

- використання розробленого веб-додатку для автоматизації роботи, що проводиться при організації конференції;

- запропонована концепція та технологія побудови моделей бізнес-процесів у системі організації конференції на прикладі веб-додатку, що досліджується;

- запропоновано аналіз розробленого додатка на якість роботи та проведення статистичного та регресійного аналізу його роботи.

**Практичне значення одержаних результатів.**

Практична цінність полягає, по-перше, у використанні розробленого веб-додатку для автоматизації роботи, що проводиться при організації конференції. По-друге, можна використовувати запропонований матеріал для вивчення та ознайомлення з відомостями стосовно даної теми.

Результати досліджень по створенню інформаційної технології можуть бути використані у навчальному процесі Луцького НТУ, зокрема на кафедрі КТПН.

**Результати роботи апробовувались:** 6-7 травня 2014 року на ХХХVІІ студентській науково-технічній конференції «Україна сьогодні: інтеграція освіти і науки». Тема доповіді: «Особливості роботи Windows Workflow Foundation».

**РОЗДІЛ 1**

**АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ**

**1.1. Концепція бізнес процесів**

Бізнес-процес – стійка, цілеспрямована сукупність взаємозв'язаних видів діяльності (послідовність робіт), яка за певною технологією перетворить входи у виходи, що представляють цінність для споживача [12].

Існують різні визначення поняття "бізнес-процесс" (workflow). Компанія Giga Group, що спеціалізується на аналізі ринку програмного забезпечення для управління і координації, пропонує наступне визначення:

Бізнес-процесом називаються всі аспекти процесу, що відносяться до його виконання - послідовності етапів бізнес-процесу і їх виконавці, обмін інформацією між етапами, механізм контролю виконання етапів [30].

Концепція бізнес процесів погано лягає на існуючі мови програмування, оскільки втрачається наочність. Частково цю проблему вирішують UML і rad-середовища. Іншою проблемою є те, що частина процесів на підприємстві залежать від людей. Така дія, як, скажімо, збір необхідних документів може бути виконане відразу, або за два тижні, людина може забути, захворіти, або звільнитися. Але, якщо бізнес процес реалізований програмно – немає сенсу тримати в пам'яті програму весь час. Стан процесу має бути збережене в базу даних (або на жорсткий диск) і при настанні такої події, як закінчення збору документів, хвороба або звільнення співробітника процес має бути відновлений, продовжуючи своє виконання. При цьому, якщо людина не виконує дії, процес повинен йому нагадати, а якщо реакція не поступає – повідомити керівництво, або передати завдання іншому співробітникові. Таким чином розробники дуже часто стикаються із завданням автоматизації бізнес процесів [13].

Серед основних причин, що спонукають організації автоматизувати бізнес-процеси, можна виділити необхідність зниження витрат або тривалості виробничого циклу, вимоги, що пред'являються споживачами і державою, впровадження програм управління якістю, злиття компаній, внутрішньо організаційні протиріччя та ін [2].

**1.2. Стандарти і методології моделювання бізнес-процесів**

Моделювання бізнес-процесів – це ефективний засіб пошуку доріг оптимізації діяльності компанії, засіб прогнозування і мінімізації ризиків, що виникають на різних етапах реорганізації підприємства. Цей метод дозволяє дати вартісну оцінку кожному окремому процесу і всім бізнес-процесам організації в сукупності [12].

Методологія створення моделі бізнес-процесу.

Під методологією (нотацією) створення моделі бізнес-процесу розуміється сукупність способів, за допомогою яких об'єкти реального світу і зв'язку між ними представляються у вигляді моделі. Будь-яка методологія включає три основні складові:

* теоретична база;
* опис кроків, необхідних для здобуття заданого результату;
* рекомендації по використанню як окремо, так і у складі групи методик.

Основне в методології – дати користувачеві послідовність кроків, які приводять до заданого результату. Здатність отримувати результат із заданими параметрами і характеризує її ефективність. Найважливішими поняттями будь-якого методу моделювання бізнес-процесів є поняття об'єкту і зв'язку. Кожен об'єкт моделі відображає деякий реальний об'єкт так званої наочної області (організації), люди, документи, машини і устаткування, програмне забезпечення і так далі. Як правило, в рамках одного методу об'єкти моделі, що відображають різну суть реального світу, також є різними [13].

Зв'язки призначені для опису взаємин об'єктів один з одним. До таких взаємин можуть відноситися: послідовність виконання в часі, зв'язок за допомогою потоку інформації, використання іншим об'єктом і так далі. Для кожного об'єкту і зв'язків характерні ряд параметрів, або, як прийнято говорити, атрибутів, що відображають певні характеристики реального об'єкту. Склад атрибутів залежить від типа реального об'єкту організації, що відображується за допомогою моделі. Атрибутами можуть служити такі характеристики, як номер об'єкту, назва, опис, тривалість виконання (для функцій), вартість та ін. На практиці при створенні моделей організації опис атрибутів об'єктів моделі здійснюється за допомогою спеціальних інструментальних засобів моделювання бізнес-процесів. Це дозволяє зробити з простого «опису» бізнес-процесу складнішу «модель», на основі якої виробляють певні обчислення, здійснюють аналіз і оцінку процесу [19].

Основу багатьох сучасних методологій моделювання бізнес-процесів склали методологія SADT (Structured Analysis and Design Technique – метод структурного аналізу і проектування) і алгоритмічні мови, вживані для розробки програмного забезпечення. У стислому вигляді історія розвитку методологій моделювання бізнес-процесів представлена в таблиці. Для наочності паралельно приведена історія розвитку підходів до управління якістю [12]. В даний час для опису, моделювання і аналізу бізнес-процесів використовуються декілька типів методологій. До найбільш поширених типів належать наступні методології: – моделювання бізнес-процесів (Business Process Modeling); – описи потоків робіт (Work Flow Modeling); – описи потоків даних (Data Flow Modeling) [10].

Розглянемо, як змінювалися методології моделювання бізнес-процесів (див. таблицю 1.1) протягом ХХ сторіччя.

Таблиця 1.1.

Методології моделювання бізнес-процесів

|  |  |  |
| --- | --- | --- |
| Період | Методологія моделювання бізнес-процесів | Методологія керування якістю |
| 40-60 рр. | Поява алгоритмічних мов опису | Національні стандарти |
| 60 рр. | Поява методології SADT (структурного аналізу та проектування) | Розвиток стандартів в різних галузях, насамперед в галузі контролю якості продукції |
| 70-80 рр. | Поява методології серії IDEF (IDEF0, IDEF3, IDEF1X), DFD, ERD | Прийняття МС ІСО серії 9000 версії 1988р. |
| 90 рр. | Поява методологій ARIS (архітектура інтегрованих інформаційних систем), UML (універсальна мова моделювання), методології компаній Oracle, Baan, Rational та ін. | Прийняття МС ІСО серії 9000 версії 1994р. (в стандартах закладаються основи процесного підходу) |
| 2000 р. | Прийняття МС ІСО серії 9000 версії 2000р., чітке визначення підходу до керування організацією | |

Ця методологія надає аналітику широкі можливості для опису бізнесу організації на верхньому рівні з акцентом на управління процесами. Нотація дозволяє відображати в моделі процесу зворотні зв'язки різного типу – за інформацією, управлінням, рухом матеріальних ресурсів. Моделі бізнес-процесів призначені для високорівневого опису бізнесу компанії. Їх основна перевага полягає в можливості описувати управління процесами організації.

Методології опису потоків робіт (Work Flow Modeling).

Друга найважливіша методологія опису процесів призначена для опису робочих процесів або, іншими словами, потоків робіт. Вона близька до алгоритмічних методів побудови схем процесів і стандартних засобів створення блок-схем. Основу цієї методології складає побудова моделей процесів за принципом послідовно виконуваних в часі робіт (функцій, операцій) [32].

Методології опису потоків даних (Data Flow Modeling).

Ще одна група методологій, активно використовуваних на практиці, – Data Flow Modeling, призначені для опису потоків даних. Вони дозволяють відобразити послідовність робіт, що виконуються по ходу процесу, і потоки інформації, циркулюючі між цими роботами. Крім того надає можливість описувати потоки документів (документообіг) і матеріальних ресурсів (наприклад, рух матеріалів від однієї роботи до іншої). Ця методологія може ефективно використовуватися для опису процесів при впровадженні процесного підходу до управління організацією, оскільки дозволяє максимально понизити суб'єктивність опису бізнес-процесів. За допомогою схеми процесів виявляють основні потоки даних, що важливе для подальшого створення моделей структури даних і розробки вимог до інформаційної системи організації [19].

Існують також інші методології, пропоновані різними фірмами-виробниками програмних продуктів. На закінчення короткого опису існуючих методологій слід зазначити, що бізнес-процеси підприємства можуть бути представлені за допомогою стандартних блок-схем, які містять деякі додаткові спеціальні графічні об'єкти. Використання цих об'єктів дозволяє зробити блок-схеми процесів наочнішими і зрозумілішими для виконавців.

**1.3. Поняття роботопотоку**

Роботопотік – це набір дій, що координують роботу людей і програм, представлених у вигляді процесу, такого як блок-схема (див. рис. 1.1.), або діаграма станів (див. рис. 1.2.) [38].

Рис. 1.1. Блок-схема

Рис. 1.2. Діаграма станів

Вживаються наступні синоніми: потік робіт, роботопотік, робочий процес, хід робіт, бізнес-процес, схема процесу, сценарій, модель процесу, модель бізнес-процесу, послідовність операцій.

Модель Workflow – колекція дій, а дії представляють собою компоненти, що визначають кроки виконання workflows [32].

Визначення workflows процесу включає одне або декілька дій, кожне з яких включає логічно завершену дію, що входить до складу процесу (див. рис. 1.3). Опціональна інформація може включати правила ініціалізації або завершення дії, його пріоритет по відношенню до інших дій, в разі виникнення конкурентної моделі. Також описується використання даних процесу. Опис дії локалізується в рамках опису процесу. Дія може бути використане кілька разів в рамках одного робочого процесу. Може мати властивість атамарності і при цьому є мінімальною одиницею в рамках процесу, такі дії називаються простими. Також дія може бути контейнером, що містить інші, окремо визначені, дії і визначення процесів і можуть виконуватися як усередині поточного workflows сервісу, так і зовнішніми сервісами. Такі дії називаються композитними.

Рис. 1.3. Роботопотік

За допомогою роботопотоків можуть бути описані наступні типи процесів:

* послідовний процес (Sequential) – перехід від одного кроку в іншій без повернень назад;
* машина станів (State-machine) – перехід з одного стану в інше, можливі і довільні повернення в попередні стани;
* процес керований правилами (Rules-driven) – окремий випадок послідовного процесу, в якому перехід на наступний крок визначається набором правил [34].

Послідовний роботопотік.

Послідовна структура визначає порядок виконання. Є набором послідовно виконуваних дій і у найбільшій мірі відповідають класичнім блок-схемам. Послідовний робочий потік (див. рис. 1.4.) починає роботу, виконуючи першу дочірню дію, що знаходиться в нім, і зазвичай продовжується до тих пір, поки не виконає всі останні дочірні дії. Є декілька ситуацій, коли робочий потік не виконує всіх своїх дій, наприклад коли виникає виключення в процесі виконання робочого потоку [35].

Крок 1

Крок 1

Рис. 1.4. Порядок виконання дій у послідовній структурі

Машина станів.

Тут зовнішні події управляють послідовністю виконання. Кінцеві автоматі дозволяють моделювати переходи між декількома зумовленими станами, що особливо актуально коли конструюються складні програмні системи, наприклад управління бізнес-процесами.

Машина переходів може бути представлена за допомогою орієнтованого графа (див. рис. 1.5), в якому число вузлів дорівнює числу кінцевих станів системи, а число ребер всій сукупності можливих переходів з одного стану в інший. Один із станів вибирається як початковий. З цього стану система може попасти в деякі (усі) інші стани за допомогою послідовності переходів [34].

Подія

Подія

Рис. 1.5. Порядок виконання дій у машині станів

Ці типи до певної міри взаємозамінні, вони можуть бути перетворені один в інший. Проте кожний з них найбільшою мірою підходить лише для певного типу завдань. Наприклад, при великій кількості галужень і виключень послідовна модель може виявитися дуже заплутаною і втратити наочність, що зведе нанівець всі переваги графічного уявлення.

Основні переваги використання роботопотоків.

Останнім часом для управління бізнес процесами у виробництві стала широко використовуватися концепція управління потоками робіт. Робочі процеси привертають велику увагу з боку розробників інформаційних систем, тому що автоматизація виконання робочих процесів, що протікають в різних областях діяльності людини, дозволяє:

* підвищити контрольованість і передбаченість таких процесів;
* прискорити процес взаємодії з користувачами і додатками;
* за допомогою розвинених засобів моніторингу збирати статистику виконання робочих процесів для їх подальшої оптимізації;
* розробляти гнучкі мови декларації маршруту робочих процесів;
* позбавити виконавця від виконання безлічі рутинних типових завдань, а кур'єрів – від необхідності грати роль “транспортного механізму” робочих процесів;
* виконувати ефективну паралельну роботу;
* залучати до етапів виконання процесу лише фахівців з необхідним рівнем кваліфікації за рахунок гнучкої політики ролей користувачів [19].

На даний момент існує цілий ряд систем управління потоками робіт. Одним з головних завдань управління потоками робіт, для опису етапів бізнес процесу, координації і контролю їх виконання, є розділення логіки процесу в цілому від логіки окремих його етапів, яка є прерогативою спеціалізованих додатків для інформаційної підтримки конкретних етапів. Це полегшує окремі модифікації як всього процесу в цілому, так і окремих етапів, а так само полегшує повторне використання додатків в різноманітних процесах.

Завдяки своїй наочності і простоті вони значно прискорюють процес розробки, сама технологія роботопотоків надає абстракції зручні для опису реальних процесів [27].

Тому, робопотоки є дуже зручним засобом формування логіки програми.

**Висновки до першого розділу**

В даному розділі проаналізовано поняття бізнес-процесів. Описано процес моделювання бізнес-процесів. Розглянуо поняття декларативного програмування. Для управління бізнес процесами використовується концепція управління потоками робіт, що дозволяє підвищити контрольованість і передбаченість таких процесів, прискорити процес взаємодії з користувачами і додатками, збирати статистику виконання робочих процесів для їх подальшої оптимізації, розробляти гнучкі мови декларації маршруту робочих процесів, позбавити виконавця від виконання рутинної роботи, розпаралелювати потоки робіт, залучати до етапів виконання процесу лише фахівців з необхідним рівнем кваліфікації за рахунок гнучкої політики ролей користувачів.

**РОЗДІЛ 2**

**ОПИС РІШЕННЯ ЗАГАЛЬНОЇ ПРОБЛЕМИ ТА ОСНОВНИХ МЕТОДІВ ДОСЛІДЖЕННЯ**

**2.1. Моделювання потоків робіт та їх опис. Динамічна інформаційна модель**

Як відомо, логіку роботи програмної системи можна представити у вигляді схеми. Зазвичай схеми використовують на етапі проектування. Для опису логіки роботи програм (алгоритмів) в WF використовуються два основні поняття – activity і workflow.

Activity – це будь-яка дія, яка логічно вважається елементарною (неділимою) або, іншими словами, activity, – це крок алгоритму. З англійського activity переводиться як 1) «діяльність» 2) «активність».

У різних джерелах використовуються наступні синоніми: завдання, функція, операція, крок процесу, елемент роботи.

Workflow – це послідовність або ієрархія активностей (activity). Переводиться як «послідовність виконуваних дій». Синоніми: потік робіт, роботопотік, робочий процес, хід робіт, схема процесу, сценарій, модель процесу, модель бізнес-процесу, послідовність операцій. Якщо activity можна розглядати як команду, специфічну для наочної області, то workflow – це дерево команд [24].

Динамічні моделі описують, що відбувається з інформацією: прикладами таких моделей є діаграми робочих процесів, потоків даних і життєвих циклів об'єктів. Динамічні моделі описують процес обміну інформацією: дані вирушають з одного місця в інше з конкретною метою. Існує декілька підходів до динамічного моделювання:

* моделі робочих процесів;
* моделі потоків даних;
* об'єктні моделі;
* життєві цикли об'єктів;
* варіанти використання;
* діаграми взаємодії об’єктів.

Моделі робочих процесів.

Моделі робочих процесів загострюють увагу на ролі людей і організацій у виконанні роботи, зберігання і обробка інформації грають в них вторинну роль. Модель процесу описує, наприклад, що буде з мандрівником, якщо на відпочинку з ним станеться нещасний випадок. Вона визначає, за які дії відповідає локальний представник на курорті, агент в країні, де знаходь курорт і головний офіс. В результаті стає ясно, хто повинен відповідати за організацію медичної допомоги, за перевезення туриста додому і за інформування родичів. Вона може описувати різні форми, що заповнюються і пересилаються між учасниками, і взагалі не залучати комп'ютерні системи. Модель процесу зазвичай фокусує увагу на ролях, обов'язках і завданнях кожної дійової особи системи (actor), а workflow-модель має справу з документами, що передаються між дійовими особами [27].

Моделі потоків даних.

Моделі потоків даних дуже нагадують попереднього типа, але тут основна увага приділяється інформаційною, а не бізнесу-систем. Ця модель описує сховища даних (data stores), де інформація знаходиться постійно (це може бути база даних в комп'ютері або просто кабінет з теками), процесори, що маніпулюють з цими даними, і потоки даних, передавальні дані від одного процесора іншому. Вона активно використовує статичну інформаційну модель: остання описує, що означають такі концепції, як мандрівник або готель, але нічого не повідомляє про те, де міститься інформація. Навпаки, з моделі потоків даних стає ясно, що інформація про туристичну поїздку знаходиться в базі даних покупок до завершення поїздки і оплати всіх рахунків, після чого резюме цієї інформації передається в маркетингову інформаційну систему, а все інше – в архів [11].

Об'єктні моделі.

Об'єктні моделі містять як динамічний, так і статичний компоненти. Динамічна або поведінкова частина визначення об'єкту зосереджена на тому, що може робити або робив кожен об'єкт, представляючи для цього набір операцій або методів, що описують його дії.

Життєві цикли об'єкту.

Життєві цикли об'єкту (на мові UML це називається лініями життя об'єкту) також загострюють увагу на індивідуальних об'єктах, але дотримуються більш цілісного підходу. Вони описують, що відбувається з об'єктом впродовж його життя: як він створюється, які події з ним відбуваються, як він реагує на ці події і які умови приводять в кінці до його руйнування [30].

Життєві цикли об'єкту дуже корисні для тестування завершеності моделі. Часто спостерігається тенденція до акцентування уваги лише на деяких подіях за рахунок останніх. Поки ми не визначимо, яким чином кожен об'єкт потрапляє в систему і як він віддаляється з неї, повного розуміння не буде.

Варіанти використання.

Варіанти використання (use cases) аналізують виконання специфічних завдань користувача (наприклад, людина, що купила туристичну поїздку, відміняє своє замовлення). Варіант використання нагадує модель процесу, але в загальному випадку загострює увагу на діяльності одного конкретного користувача [31].

Варіанти використання можуть бути корисні як на етапі моделювання ділової активності, так і при описі внутрішньої поведінки інформаційних систем. Одна з небезпек полягає в змішенні двох рівнів. Краще за це не робити, оскільки вони представляють інтерес для різних аудиторій.

Представлений у вигляді варіанту використання діалог призначеного для користувача інтерфейсу описує, якою інформацією користувач обмінюються з системою, бо, як вона представлена на екрані. Це природним чином приводить до реалізації XML, в якій інформаційний вміст відокремлений від особливостей вистави [32].

Діаграми взаємодії об'єктів.

Діаграми взаємодії об'єктів дозволяють проаналізувати обмін повідомленнями між об'єктами на тоншому рівні деталізації, чим модель потоку даних. Діаграми взаємодії об'єктів неоцінимі, якщо потрібно описати взаємодію між різними системами. Вони дозволяють визначити, яка інформація міститься в якому повідомленні. Оскільки ці повідомлення написані на мові XML, діаграми взаємодії об'єктів дають нам контекст, потрібний для початку проектування структури XML кожного індивідуального повідомлення [30].

Вибір підходу до динамічного моделювання.

Моделі потрібні лише як робочі інструменти, що дають можливість при зустрічі з користувачами прийти до угоди з питання про те, як вона повинна працювати. Тому не потрібно розглядати всі моделі. Досить розглянути лише ті, які дійсно дадуть необхідну інформацію про проблему, що розробляється.

**2.2. Управління потоками робіт**

З незапам'ятних часів при вивченні програмування використовувалися блок-схеми. У них є одна безперечна перевага – вони наочно відображують логіку роботи програми. Але за допомогою блок-схем також дуже легко описуються робочі потоки (а бізнес процеси є всього лише набором робочих потоків). Вони досить прості в освоєнні і наочні у використанні, завдяки чому вони популярні і широко використовуються для побудови workflow [12].

В якості інструментальних засобів також використовуються сучасні case-засоби, спеціалізовані і універсальні мови, за допомогою яких можна реалізувати управління потоками робіт. Ось деякі з них. Bp-win (бізнес процеси і вікна) – засіб функціонального моделювання. Bpwin допомагає швидко створювати і аналізувати моделі з метою оптимізації ділових і виробничих процесів [38].

Пакет ARIS Toolset - розрахований на багато користувацьке середовище опису і аналізу робочих процесів підприємств, що підтримує розробку складних гетерогенних інформаційних систем і супроводжує весь цикл розробки (аналіз – проектування – реалізація). Org-master (Система бізнесу-моделювання) – розрахована на багато користувацьке середовище моделювання і організації діяльності підприємства, що підтримує системний і процесний підходи до ведення бізнесу на основі інформаційних моделей.

WFS (Work Flow system) - програмне забезпечення і система технологій управління діловими процесами. Але у ній так само широко використовується UML (Unified Modeling Language) - уніфікована мова моделювання, що є стандартним інструментом для створення програмного забезпечення. UML задовольняє потреби при моделюванні будь-яких систем. Системи описуються набором моделей, що розглядають її з різних точок зору [19].

У UML всі абстракції програмної системи організовуються у вигляді моделей, кожна з яких представляє відносно незалежний аспект системи, що розробляється. Для візуалізації наборів цих абстракцій можна використовувати діаграми. Моделі, які створюються з її допомогою, можуть бути безпосередньо перекладені на будь-яку об'єктно-орієнтовану мову програмування.

Поняття, які зручніше представляти графічно, представляються в UML, ті ж, які краще описувати в текстовому вигляді, виражаються за допомогою мови програмування. UML дозволяє безпосередньо створювати моделі, імітувати поведінку системи і контролювати діяльність системи [28].

Використання діаграм станів для опису потоків робіт.

Діаграма станів відображує стан об'єкту і події що викликають перехід об'єкту з одного стану в інший. Діаграма станів описує якийсь кінцевий автомат, поведінка якого може бути представлена графом. Вершини цього графа – стани, орієнтовані дуги означають перехід з одного стану в інший. Поведінка системи у такому графі моделюється шляхом від початкової вершини до кінцевої [30].

Стани в цьому графі впорядковані, існує наступний і попередній стан. Автомат характеризується наступними властивостями:

Кожен стан має бути визначеним і число станів кінцеве.

Граф не містить ізольованих вершин, кожен перехід сполучає 2 стани, в кожній вершині є перехід в подальшу або в цю ж вершину.

У кожен момент часу автомат знаходиться в єдиному стані.

Автомат не запам'ятовує перехід із стану в стан.

Час переходу на діаграмі станів не враховується і не існує переходу одночасно в 2 стани. Кожна з доріг в графі відповідає певному сценарію.

Стани дії.

В потоці управління, що моделюється діаграмою станів, відбуваються різні події. Ви можете обчислити вираження, внаслідок чого змінюється значення деякого атрибуту або повертається деяке значення (див.   
рис. 2.1.). Також можна виконати операцію над об'єктом, наприклад, послати йому сигнал або навіть створити його або знищити. Всі ці виконувані атомарні обчислення називаються станами дії, оскільки кожне з них є стан системи, що є виконанням деякої дії [37].

Складання кошторису

index := lookup(e) + 7;

Проста дія

вираз

Рис. 2.1. Стани дії

Стани дії не можуть бути піддані декомпозиції. Крім того, вони атомарні. Це означає, що усередині них можуть відбуватися різні події, але виконувана в стані дії робота не може бути перервана. І нарешті, зазвичай передбачається, що тривалість одного стану дії займає невідчутно малий час [38].

Переходи.

Коли дія або діяльність в деякому стані завершується, потік управління відразу переходить в наступний стан дії або діяльності. Для опису цього потоку використовуються переходи, що показують дорогу з одного стану дії або діяльності в інше. У UML перехід представляється простою лінією із стрілкою [41], як показано на рис. 2.2.



Рис. 2.2. Нетригерні переходи

Потік управління повинен десь починатися і закінчуватися (зрозуміло, якщо це не безкінечний потік, в якого є початок, але немає кінця). Як показано на рисунку, ви можете задати як початковий стан (закрашений кружок), так і кінцевий (закрашений кружок усередині кола) [32].

Розгалуження.

Прості послідовні переходи зустрічаються частіше всього, але їх одних недостатньо для моделювання будь-якого потоку управління. Як і в блок-схемі, ви можете включити в модель розгалуження, яке описує різні дороги виконання залежно від значення деякого булевого виразу. Як видно з рисунку 2.3, точка галуження представляється ромбом. У точку галуження може входити рівно один перехід, а виходити – два або більше. Для кожного витікаючого переходу задається булевий вираз, який обчислюється лише один раз при вході в точку галуження. Ні для яких двох витікаючих переходів ці сторожові умови не повинні одночасно набувати значення "істина", інакше потік управління виявиться неоднозначним. Але ці умови повинні покривати всі можливі варіанти, інакше потік зупиниться. (Галуження семантично еквівалентно множинним переходам із сторожовими умовами) [41].



Рис. 2.3. Розгалуження

Для зручності дозволяється використовувати ключове слово else для позначки того з витікаючих переходів, який має бути вибраний у випадку, якщо умови, задані для всіх останніх переходів, не виконані. Реалізувати ітерацію можна, якщо ввести два стани дії – в першому встановлюється значення лічильника, в другому воно збільшується – і точку галуження, обчислення в якій показує, чи слід припинити ітерації.

Розділення і злиття.

Прості послідовні переходи, що гілкуються, в діаграмах діяльності використовуються найчастіше. Проте можна зустріти і паралельні потоки, і це особливо характерно для моделювання бізнес-процесів. У UML для позначення розділення і злиття таких паралельних потоків використовується межа синхронізації, яка зображається у вигляді жирної вертикальної або горизонтальної лінії [47].

Кожен з потоків управління, що паралельно виконується, існує в контексті незалежного активного об'єкту, який, як правило, моделюється або процесом, або обчислювальною ниткою. Розглянемо, наприклад, паралельні потоки, використовувані в пристрої, який імітує людську мову і жестикуляцію. Як показано на рисунку 2.4., точка розділення відповідає розщеплюванню одного потоку управління на два що виконуються паралельно. У цій крапці може існувати рівно один вхідний перехід і два або більш витікаючих. Кожен витікаючий перехід представляє один незалежний потік управління [45]. Після крапки розділення, що асоціюється з кожною дорогою в графі, продовжується паралельне виконання. З концептуальної точки зору мається на увазі дійсний паралелізм, тобто одночасне виконання, але в реальній системі це може як виконуватися, так і не виконуватися (якщо система розміщена лише на одному вузлі). У останньому випадку має місце послідовне виконання з перемиканням між потоками, що дає лише ілюзію дійсного паралелізму.



Рис. 2.4. Розділення і злиття

З рисунка також видно, що точка злиття є механізмом синхронізації декількох паралельних потоків виконання. У цю крапку входять два або більше переходи, а виходить рівно один. У точці злиття паралельні потоки синхронізуються, тобто кожен з них чекає, поки всі інші досягнуть цієї крапки, після чого виконання продовжується в рамках одного потоку [45].

**2.3. Основи роботи Windows Workflow Foundation**

Термін «workflow» досить поширений в IT-індустрії і сприймається насамперед у контексті систем електронного документообігу. Однак, як це часто відбувається в тому випадку, коли спочатку формується ринок, а лише потім узгоджується термінологія, існує кілька різних тлумачень даного поняття, кожне з яких має певне право на існування. Тут ми не будемо їх аналізувати, оскільки, як це стане ясно нижче, у випадку WWF в «workflow» вкладається сенс, дещо відрізняється від прийнятого в бізнес-системах.

У найбільш широкому розумінні термін workflow означає упорядковану послідовність дій (або робіт), покликаних вирішувати якусь завдання. У такому трактуванні досить прозоро визначено суть поняття, проте немає навіть натяку на ту форму, в якій воно постане перед споживачем. Тут цілком можемо прийняти точку зору, напевно, найавторитетнішою в даній галузі міжнародної організації Workflow Management Coalition (WfMC, wfmc.org). Не вдаючись у нюанси, відзначимо, що WfMC передбачає низку вимог до workflow-систем, включаючи наявність спеціального інструментарію для визначення моделей процесів, «движка», який буде їх виконувати, засобів для моніторингу, управління та складання звітності.

При цьому, як правило, мається на увазі, що інструментарій workflow-моделювання має можливості візуального проектування, достатніми для створення складних бізнес-систем фактично без програмування в традиційному розумінні (тобто без написання коду вручну).

Однак якщо розглядати програмування в широкому сенсі - як створення набору інструкцій, що передбачають зворотний зв'язок і можливість повторного використання, то стане ясно, що під це визначення цілком потрапляє моделювання будь-яких робочих процесів, так чи інакше укладаються в моделі workflow. Саме від цього відштовхувалася Microsoft, розробляючи WWF.

Важливо розуміти, що WWF не є ні реалізацією якого стандарту workflow зразок тих, які визначені WfMC, ні інструментом суто ділового застосування. Отже, спроби порівнювати WWF з існуючими на ринку workflow-систем готовими бізнес-продуктами позбавлені всякого сенсу, оскільки це універсальна і гнучка платформа, орієнтована насамперед на [візуальне програмування](http://translate.googleusercontent.com/translate_c?depth=1&hl=uk&prev=/search%3Fq%3D%25D0%25BE%25D0%25BF%25D0%25B8%25D1%2581%2BWindows%2Bworkflow%2Bfoundation&rurl=translate.google.com.ua&sl=ru&u=http://itc.ua/20466&usg=ALkJrhhrOmoxjZL3sehdK7P_sIp98ztGww). І тому конструювання workflow-діаграм за допомогою WWF в тій же мірі програмування, в якій і створення блок-схем.

Відповідно, дуже важлива особливість WWF – універсальність. Оскільки візуальне програмування відрізняється від традиційних підходів до розробки програмних систем тільки зовні, але ніяк не своєю суттю, то WWF практично рівноцінна самій платформі. NET Framework 2.0, на основі якої вона реалізована. Це, втім, жодною мірою не означає, що WWF може повністю замінити традиційні підходи, однак для певного кола завдань (у тому числі, мабуть, і для реалізації workflow-систем в звичайному розумінні) дане рішення дійсно може виявитися кращим.

Архітектура WWF.

Архітектурно Windows Workflow Foundation складається з декількох рівнів (див. таблицю 2.1). На нижньому знаходиться батьківський процес (Host Process). Характерно, що WWF працює не ізольовано, а виконується саме в рамках батьківського процесу, який за допомогою стандартизованих інтерфейсів надає набір сервісів, що визначають функціональність WWF. Схожий підхід застосовується в Microsoft Windows Scripting Host (WSH) для забезпечення доступу до внутрішніх об'єктам.

Взаємодія з батьківським процесом здійснюється за допомогою батьківського рівня (Hosting Layer). Його інтерфейси для ключових процесів рівня виконання WWF повинні реалізувати розробники, інтегруючі WWF в свої додатки. Microsoft також збирається реалізувати Hosting Layer для своїх ключових платформ, зокрема ASP.NET 2.0, SQL Server 2005 і Microsoft Office 12, відповідно, незабаром в їх рамках можна буде використовувати workflow-моделі.

Таблиця 2.1

Архітектурні рівні WWF

|  |  |
| --- | --- |
| Рівень | Завдання |
| Workflow Model | Підтримка визначених типів моделей, дій (activities) і відповідного API |
| Runtime Layer | Виконання моделі та підтримка управління її життєвим циклом |
| Hosting Layer | Інтеграція з батьківським процесом, надання інтерфейсів для ключових процесів WWF, реалізація яких залежить від батьківського процесу |
| Host Process | Забезпечення об'єктів і функціональності, що повністю або частково доступні в WWF |

Ядром технології WWF є рівень виконання (Runtime Layer), який відповідає безпосередньо за workflow-моделювання та містить необхідні для цього служби. Він же управляє життєвим циклом (менеджмент станів і активація) моделей, що дозволяє виконувати значну кількість моделей, що не витрачаючи даремно системні ресурси в тому випадку, якщо частина з них знаходиться в режимі очікування.

На самому верху архітектури WWF – рівень workflow-моделі (Workflow Model Layer). Він відповідає за підтримку різних типів моделей, містить зумовлені дії (activities) і реалізує відповідний API.

Також WWF включає вбудований візуальний редактор, завдяки якому можна конструювати моделі без програмування. Хоча при необхідності вони можуть бути реалізовані виключно за допомогою програмного коду, оскільки фактично представляють собою звичайні класи. Крім того, підтримується можливість зберігання описи моделей в XML-файлах, що дозволить застосовувати різноманітний інструментарій.

Сама технологія WWF спочатку орієнтована на розширення. Окремі елементи workflow-моделі – дії (activities) - виконані у вигляді компонентів, які можуть створюватися сторонніми постачальниками за допомогою Visual Studio і WWF SDK. Крім цього, WWF підтримує динамічне оновлення моделей під час виконання, а також дозволяє вбудовувати редактор моделей у власні додатки.

Використання WWF на практиці.

На поточний момент створювати та виконувати workflow-моделі можна за допомогою відповідного розширення для Visual Studio, доступного на сайті Microsoft (див. рисунок 2.5).

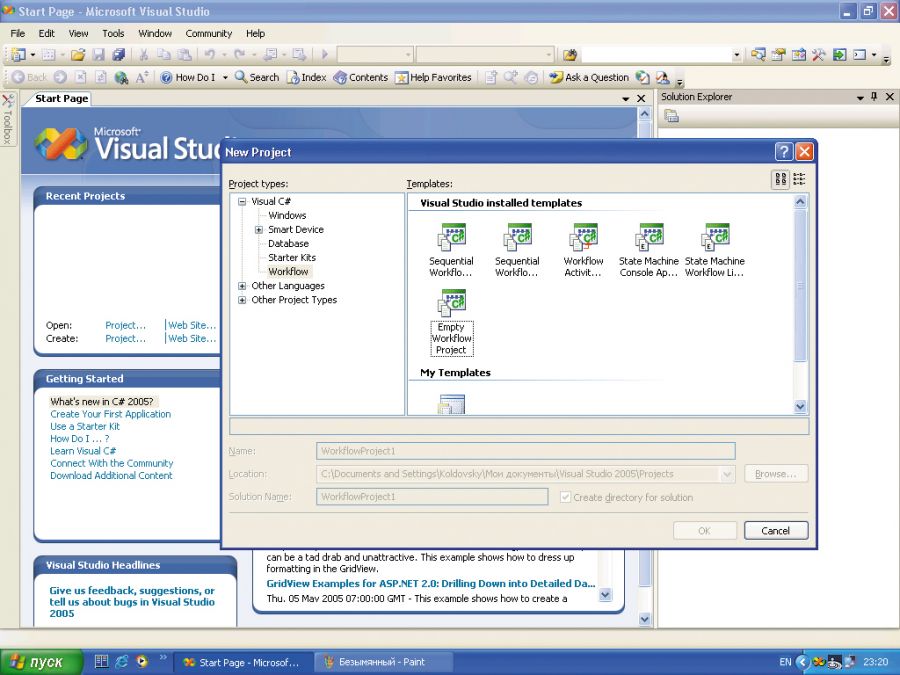


Рис. 2.5. Розширення для Visual Studio

Після встановлення Visual Studio Extensions for Windows Workflow Foundation в середовищі Visual Studio з'являється можливість створювати нові проекти на основі WWF

WWF стандартно підтримує два типи моделей – послідовні (sequential) і кінцеві автомати (state machine). Перші являють собою набір послідовно виконуваних дій і в найбільшій мірі відповідають класичним блок-схемами. Другі дозволяють моделювати переходи між декількома зумовленими станами, що особливо актуально при конструюванні складних програмних систем, наприклад управління бізнес-процесами.

Ці два типи в певній мірі взаємозамінні – послідовна модель може бути перетворена в кінцевий автомат і навпаки. Проте кожен з них найбільшою мірою підходить лише для певного типу завдань. Наприклад, при великій кількості розгалужень і винятків послідовна модель може виявитися занадто заплутаною і втратити наочність, що зведе нанівець всі переваги графічного представлення. Меню Toolbox WWF наведено на рис. 2.6.



Рис. 2.6. Меню Toolbox WWF

В стандартній поставці WWF йде великий набір доступних дій для створення моделей різного рівня складності

Незважаючи на те що стандартно підтримуються тільки два типи моделей, розробники можуть самостійно створювати додаткові. Конструювання нових моделей на основі існуючих не викличе великих складнощів навіть у відносно непідготовленого користувача, що володіє лише загальними навичками опису алгоритмів. Для цього призначений візуальний режим, в якому досить вибирати доступні дії (activities) і вказувати їх властивості та взаємозв'язки. Їх набір досить великий у стандартній поставці та включає можливості управління виконанням, побудови циклів, розпаралелювання, роботу з винятками, під'єднання до джерел даних, зв'язування з Web-службами та ін Створення ж повністю оригінальних моделей вимагає вельми професійної роботи.

Для побудови умов система пропонує візуальний конструктор (див. рис. 2.7), що дозволяє без програмування працювати з бінарними операторами. Якщо ж потрібно створити більш складну умову, то застосовуються традиційні можливості однієї з мов програмування - на рівні методу класу, що реалізовує модель.

Незважаючи на те, що workflow-моделі можна створювати та виконувати самі по собі, повністю їх потенціал розкривається тільки при інтеграції з додатком, що забезпечує об'єкти для маніпуляції. Для того, щоб модель отримала доступ до внутрішнього пристрою батьківського процесу, необхідно реалізувати певні інтерфейси, описані в документації до WWF.

Моделі також підтримують режим налагодження. Незважаючи на певні проблеми у поточній версії, Microsoft обіцяє, що будуть доступні точки переривання і покрокове виконання як в графічному поданні, так і у вигляді програмного коду.

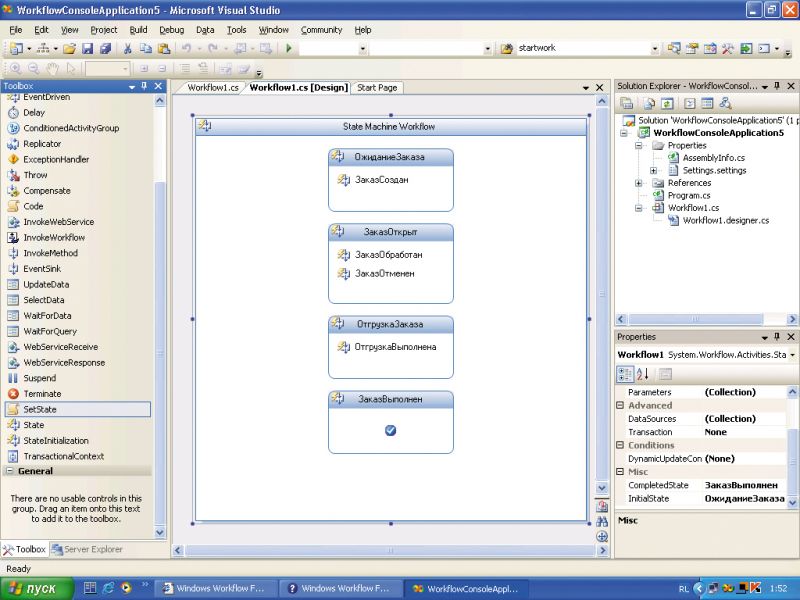


Рис. 2.7. Візуальний конструктор WWF

Моделі кінцевих автоматів добре підходять для вирішення бізнес-завдань.

Навіть таке поверхневий опис дозволяє окреслити деякі сфери застосування WWF. Дана технологія добре підходить для вирішення завдань управління бізнес-процесами та документообігом, оскільки дає можливість наочно представити їх алгоритмику. Це може бути погодження умов контракту та забезпечення його виконання або прийняття замовлень та їх реалізація з урахуванням індивідуальних особливостей і побажань кожного замовника.

Також WWF добре підходить для формалізації взаємодії різних виконуючих пристроїв, і навіть людини і машини. Ця особливість стане в нагоді не тільки в бізнес-системах, але і в побутовій сфері. Приміром, за допомогою WWF можна створити якийсь «конструктор», призначений для управління логікою роботи складових «розумного будинку» - від включення нагрівальних приладів до алгоритму прибирання приміщень за допомогою роботизованого пилососа. Послідовні моделі (див. рис. 2.8) найбільшою мірою схожі з звичайними блок-схемами.

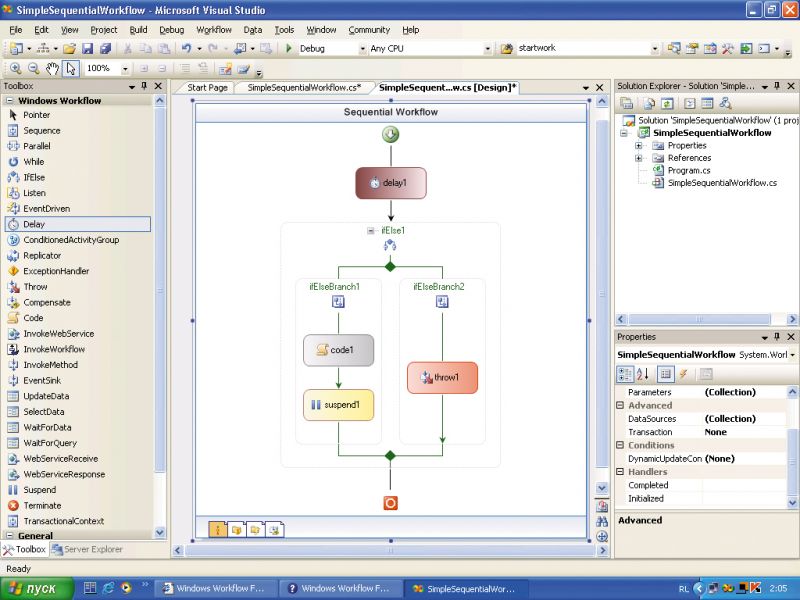


Рис. 2.8. Послідовні моделі WWF

Не виключено, що WWF знайде застосування і в самих звичайних додатках. Наприклад, багатьох лякала необхідність вивчати скриптові мови для того, щоб автоматизувати часто виконувані рутинні операції. Наочність і простота створення моделей в WWF може змінити ставлення користувачів до нескладного програмування і помітно підвищити ефективність їхньої роботи.

Переваги WWF.

Спробуємо виділити деякі переваги технології WWF.

По-перше, використання WWF забезпечує наочність і прозорість програмної логіки. Графічне подання, на відміну від коду, доступно для розуміння не тільки розробникам, а й іншим учасникам процесу створення ПЗ, в тому числі, що особливо важливо, замовникам. WWF дозволяє набагато ефективніше в порівнянні з альтернативними підходами відокремити інтерфейс користувача та бізнес-логіку додатків, спрощуючи колективну розробку, модифікацію рішень та їх портування під нові платформи і різні типи клієнтських пристроїв.

По-друге, наявність середовища виконання WWF та візуального дизайнера моделей у багатьох випадках підніме на якісно новий рівень вбудовані засоби автоматизації / адаптації додатків. На наш погляд, наочне конструювання алгоритмів значно зручніше і доступніше, ніж, приміром, аналогічні рішення на базі скриптів.

По-третє, WWF стає новим стандартним елементом платформи Windows з уніфікованим інтерфейсом і єдиними правилами застосування, що істотно спрощує завдання розробників з просування продуктів на його основі. Приклад показує сама Microsoft, що використовує WWF в майбутній версії Microsoft Office.

По-четверте, слід відзначити, що Microsoft заклала в архітектуру WWF дуже важливу функціональну особливість, якою можуть похвалитися далеко не всі постачальники нинішніх workflow-систем, а саме, можливість динамічно змінювати виконуються моделі.

Нові можливості Windows workflow foundation 4.

В порівнянні з попередніми версіями Windows Workflow Foundation (WF) в .NET Framework, версія 4 змінює декілька принципів розробки. Тепер робочі процеси стало ще простіше створювати, виконувати та підтримувати, а також реалізовувати вузол з новими функціями.

Модель дій робочого процесу.

Тепер дія, а не використання класу SequentialWorkflowActivity або StatemachineWorkflowActivity є базовою одиницею створення робочого процесу. Клас Activity забезпечує базову абстракцію поведінки робочого процесу. Автори дій можуть реалізувати CodeActivity для забезпечення базових призначених для користувача функцій дій або NativeActivity для забезпечення функцій користувача дій, що використовують середу виконання. Activity - це клас, який використовується авторами дій для оголошення нової поведінки за допомогою інших об'єктів NativeActivity, CodeActivity, AsyncCodeActivity або DynamicActivity, незалежно від того, є вони розробкою користувача або включені до бібліотеки «Вбудована бібліотека дій. NET Framework 4».

Параметри складних композитних дій.

Flowchart являє собою нову потужну дію потоку керування, що дозволяє авторам моделювати довільні цикли та умовне розгалудження. Flowchart забезпечує модель програмування на основі подій, яку раніше можна було реалізувати тільки за допомогою дії StateMachineWorkflowActivity. Процедурні робочі процеси отримують додаткові переваги завдяки новим діям управління потоком, які моделюють звичайні структури управління потоком, такі як TryCatch та Switch.

Розширена бібліотека вбудованих дій.

Нові функції бібліотеки дій:

* нові дії керування потоком, такі як DoWhile, Pick, TryCatch, ForEach, Switch та ParallelForEach;
* дії для обробки даних элементів, такі як Assign, та дії коллекції, такі як AddToCollection;
* дії для керування транзакціями, такі як TransactionScope и Compensate;
* нові дії для обміну повідомленнями, такі як SendContent и ReceiveReply.

Явна модель даних дій.

Бібліотека .NET Framework 4 включає нові варіанти збереження та перенесення даних. Дані можливо зберегти в дії за допомогою змінної Variable. При перенесенні даних у дію та з неї для визначення напрямку переміщення даних використовуються спеціальні типи аргументів. Такими типами є InArgument, InOutArgument та OutArgument.

Поліпшені варіанти розміщення на вузлі, зберігання та відстеження.

Бібліотека .NET Framework 4 містить наступні вдосконалення зберігання.

* додані нові параметри для виконання робочих процесів, включаючи WorkflowServiceHost, WorkflowApplication та WorkflowInvoker;
* за допомогою дії Persist дані стану робочого процесу можна зберігати явно;
* вузол може зберегти екземпляр ActivityInstance, не вивантажуючи його;
* робочий процесс може вказувати зони незберігання під час роботи з даними, які не можуть бути збережені, щоб відкласти операцію зберігання до виходу із зони незберігання;
* транзакції можуть бути введені в робочий процес за допомогою TransactionScope;
* виконувати відстеження простіше завдяки TrackingParticipant;
* відстеження та запис у журнал системних подій забезпечується за допомогою EtwTrackingParticipant;
* для поновлення чекающого рабочого процессу тепер використовується об’єкт Bookmark.

Більш зручне розширення середовища проектувания WF Designer.

Новий WF Designer побудован на основі Windows Presentation Foundation (WPF), він забезпечує більш зручну модель для повторного розміщення WF Designer за межамиVisual Studio, а також забезпечує більш прості механізми для створення конструкторів дій користувача.

Сховище примірників робочих процесів SQL.

Бібліотека NET Framework 4 постачається разом зі сховищем примірників робочого процесу SQL, завдяки чому робочі процеси можуть зберігати відомості про стан примірників робочих процесів в базах даних SQL Server 2005 або SQL Server 2008. Ця функція найчастіше реалізується у вигляді класу SqlWorkflowInstanceStore, що є похідним від абстрактного класу InstanceStore платформи зберігання. Компонент сховища примірників робочого процесу SQL складається з постачальника зберігання SQL, що є конкретною реалізацією API зберігання, використовуваної провідним додатком для відправки команд зберігання у сховищі.

Сховище примірників робочого процесу SQL підтримує як самостійно розміщуються робочі процеси, так і служби робочих процесів, які використовують WorkflowApplication або WorkflowServiceHost, а також служби, які розміщуються в WAS з використанням WorkflowServiceHost. Компонент сховища примірників робочого процесу SQL можна налаштувати на роботу з самостійно розміщеними службами програмним чином, скориставшись наданої компонентом об'єктної моделлю. Налаштувати цю функцію для служб, розміщених в WorkflowServiceHost, можна як програмним чином з використанням об'єктної моделі, так і з використанням файлу конфігурації XML.

Компонент сховища примірників робочого процесу SQL (клас SqlWorkflowInstanceStore) не реалізує PersistenceProviderFactory і тому не забезпечує підтримку зберігаємості для стійких служб WCF, що не входять в робочий процес. Він також не реалізує WorkflowPersistenceService і тому не забезпечує підтримку зберігаємості для робочих процесів версії 3.x. Цей компонент підтримує зберігаємість тільки для робочих процесів WF 4.0 і служб робочих процесів. Компонент також не підтримує бази даних, відмінні від SQL Server 2005 і SQL Server 2008.

Налаштування режиму зберігання SQL для робочих процесів і служб робочих процесів.

Далі розглянемо як налаштувати функцію сховища примірників робочих процесів SQL для включення зберігання робочих процесів і служб робочих процесів програмно і за допомогою файлу конфігурації. Перед використанням функції сховища примірників робочих процесів SQL необхідно створити базу даних,яка використовується функцією для збереження примірників робочих процесів.

Програма установки .NET Framework 4 копіює файли скрипта SQL, пов'язані з функцією сховища примірників робочих процесів SQL, в папку % WINDIR% \Microsoft.NET \ Framework \ v4.xxx \ SQL \ EN. Виконайте ці файли скрипта в базі даних SQL Server 2005 або SQL Server 2008, яка повинна використовуватися в сховище примірників робочих процесів SQL для збереження примірників робочихпроцесів.

Спочатку запустіть файл SqlWorkflowInstance StoreSchema.sql, а потім файл SqlWorkflow InstanceStoreLogic.sql.

Також слід зауважити, що для того щоб очистити базу даних зберігаємості для отримання знову підготовленої бази даних, необхідно застосувати сценарії в каталозі %WINDIR%\Microsoft.NET\Framework\ v4.xxx\SQL\EN в наступному порядку: з початку SqlWorkflowInstanceStoreSchema.sql, потім SqlWorkflowInstanceStoreLogic.sql

Якщо база даних зберігаємості не створена, то при спробі вузла зберегти робочі процеси сховище примірників робочих процесів SQL виробляє виняток, подібне до наступного System.Data.SqlClient.SqlException: якщо не вдається знайти збережену процедуру System.Activities. DurableInstancing. CreateLockOwner.

Включення збереження резидентних робочих процесів, які використовують WorkflowApplication.

Можна включити зберіганість для резидентних робочих процесів, які програмно використовують WorkflowApplication за допомогою об'єктної моделі SqlWorkflowInstanceStore. Для цього необхідно виконати наступне.

Додаємо посилання на «System.Activites.DurableInstancing.dll».

Додаємо наступний оператор в початок вихідного файлу після існуючих інструкцій using:

*using System.Activities.DurableInstancing;*

Створюємо об'єкт SqlWorkflowInstanceStore і призначаємо його властивості InstanceStore програми WorkflowApplication, як показано в наступному прикладі коду:

*SqlWorkflowInstanceStore store =*

*new SqlWorkflowInstanceStore ("Server =.\\SQLEXPRESS;*

*Initial Catalog = Persistence;*

*Integrated Security = SSPI");*

*WorkflowApplication wfApp =*

*new WorkflowApplication (new Workflow1 ());*

*wfApp.InstanceStore = store;*

Викличте метод Persist об'єкта WorkflowApplication, щоб зберегти робочий процес, або метод Unload, щоб зберегти і вивантажити робочий процес. Можна також обробляти подію PersistableIdle, викликану об'єктом WorkflowApplication, і повертати відповідний елемент (Persist або Unload) дії PersistableIdleAction.

*wfApp.PersistableIdle = delegate (WorkflowApplicationIdleEventArgs e)*

*{*

*return PersistableIdleAction.Persist;*

*};*

Використання дій обміну повідомленнями.

Дії обміну повідомленнями дозволяють робочим процесам відправляти або отримувати повідомлення WCF. Додаючи дії обміну повідомленнями в робочий процес, можна моделювати шаблони обміну повідомленнями будь-якого рівня складності.

Шаблони обміну повідомленнями.

Існує три основних шаблону обміну повідомленнями:

Датаграма – при використанні шаблону обміну повідомленнями «датаграма», клієнт відправляє повідомлення службі, але служба не відповідає. Такий метод іноді називають “відправити і забути”. У цьому випадку потрібно зовнішнє підтвердження успішної доставки. Повідомлення може бути втрачено при передачі і може не досягти служби. Якщо клієнт успішно відправив повідомлення, це не гарантує, що вилучена служба отримала його. Датаграма – це фундаментальний елемент обміну повідомленнями, на основі якого можна будувати власні шаблони обміну повідомленнями.

Запит-відповідь – при використанні шаблону обміну повідомленнями “запит-відповідь” клієнт відправляє повідомлення службі, служба виконує необхідні обчислення і відправляє відповідь клієнту. Шаблон складається з пар "запит-відповідь". Прикладами викликів "запит-відповідь" є віддалені виклики процедур (RPC) і запити GET веб-браузера. Цей шаблон також називають напівдуплексним.

Дуплекс – при використанні шаблону обміну повідомлення “дуплекс” клієнт і служба можуть відправляти повідомлення один одному в будь-якому порядку. Застосування дуплексного шаблону схоже на розмову по телефону, коли кожне промовлене слово є повідомленням.

Дії обміну повідомленнями дозволяють реалізувати будь-який з цих базових шаблонів обміну повідомленнями, а також будь-які інші шаблони обміну повідомленнями будь-якого рівня складності.

Дії обміну повідомленнями.

Платформа. NET Framework, версія 4 визначає наступні дії обміну повідомленнями:

Send – використовується для відправки повідомлення.

SendReply – використовується для відправки відповіді на отримане повідомлення. Це дія використовується службами робочого процесу при реалізації шаблону обміну повідомленнями «запит-відповідь».

Receive – використовується для отримання повідомлення.

ReceiveReply - використовується для отримання листа у відповідь. Це дія використовується клієнтами служби робочого процесу при реалізації шаблону обміну повідомленнями “запит-відповідь”.

Дії обміну повідомленнями та шаблони обміну повідомленнями.

У шаблоні обміну повідомленнями «датаграма» беруть участь клієнт, що відправляє повідомлення, і служба, яка отримує його. Якщо клієнт – це робочий процес, то слід використовувати дію Send для відправки повідомлення. Для отримання цього повідомлення в робочому процесі слід використовувати дію Receive. Дії Send і Receive мають властивість з ім'ям Content. Ця властивість містить відправляються або одержувані дані. При реалізації шаблону обміну повідомленнями «запит-відповідь» клієнт та служби використовують пари дій. Клієнт використовує дію Send для відправки повідомлення і дію ReceiveReply для отримання відповіді від служби. Ці дві дії пов'язані один з одним властивістю Request. Значення цієї властивості - дія Send, що відправило вихідне повідомлення. Служба також використовує пару пов'язаних дій: Receive і SendReply. Ці дві дії пов'язані властивістю Request. Значення цієї властивості – дія Receive, що отримало вихідне повідомлення. Дії ReceiveReply і SendReply, як і дії Send і Receive, дозволяють відправити примірник Message або тип контракту повідомлення.

Оскільки робочі процеси виконуються протягом довгого часу, дуже важливо, щоб шаблон обміну повідомленнями «дуплекс» також підтримував довготривалі діалоги. Для підтримки довготривалих діалогів клієнти, які ініціюють діалоги, повинні надавати службу з можливістю її повторного виклику пізніше, коли дані стануть доступні. Наприклад, замовлення на покупку подається на схвалення менеджера, але він може бути оброблений через день, тиждень чи навіть рік; робочий процес, керуючий замовленням на купівлю повинен мати можливість продовжити роботу з ним після одержання схвалення. Шаблон обміну повідомленнями «дуплекс» підтримується в робочих процесах, що використовують кореляцію. Для реалізації шаблону «дуплекс» слід використовувати дії Send і Receive. У дії Receive слід ініціалізувати кореляцію з використанням значення спеціального ключа CallbackHandleName. У дії Send задаємо дескриптор взаємозв'язку як значення властивості CorrelatesWith.

Також слід зауважити, що реалізація шаблону «дуплекс» у робочому процесі з використанням кореляції зворотного виклику («стійкий дуплекс») призначена для роботи з довготривалими діалогами. Це не те ж саме, що «дуплекс» WCF з контрактами зворотного виклику, в яких діалоги є короткочасними (на час існування каналу).

Додання посилання на службу.

Коли в додатку робочого процесу використовується середовище Visual Studio 2010 створює для користувача дії обміну повідомленнями, які інкапсулюють звичайні дії Send і ReceiveReply, використовувані в шаблоні обміну повідомленнями «запит-відповідь». Для використання цієї функції клацаємо правою кнопкою миші на клієнтському проекті в середовищі Visual Studio 2010 і вибираємо Додати посилання на службу. Вводимо базову адресу служби в поле адреси і клацаємо «Перейти». Доступні служби відображаються в полі Служби:. Розгортаємо вузол служби для відображення підтримуваних контрактів. Вибираємо контракт, який слід викликати, і в полі Операції буде відображений список доступних операцій. Потім можна вказати простір імен для створюваного дії і натиснути кнопку ОК. З'явиться діалогове вікно з повідомленням, що операція була завершена успішно, а створені для користувача дії з'являться в області елементів після повторного побудови проекту. Для кожної операції, визначеною в контракті служби, існує одна дія. Після повторного побудови проекту користувач зможе перетягувати користувача дії в робочий процес і задавати будь-які необхідні властивості у вікні властивостей.

Шаблони дій обміну повідомленнями.

Для полегшення налаштування шаблону обміну повідомленнями «запит-відповідь» на клієнті і в службі середу Visual Studio 2010 надає два шаблони обміну повідомленнями. Шаблон ReceiveAndSendReply використовується для служби, шаблон SendAndReceiveReply - для клієнта. В обох випадках шаблони додадуть відповідні дії обміну повідомленнями в робочий процес. Для служби шаблон ReceiveAndSendReply додасть дію Receive, за яким слід дію SendReply. Властивість Request автоматично задається для дії Receive. Для клієнта шаблон SendAndReceiveReply додасть дію Send, за яким слід дію ReceiveReply. Властивість Request автоматично задається для дії Send. Щоб використовувати ці шаблони, перетягуємо відповідний шаблон у робочий процес.

Дії обміну повідомленнями і транзакції.

При визовах служби робочого процесу може бути корисно виділити для операції служби транзакції. Щоб зробити це, помістіть дію Receive в дію TransactedReceiveScope. Дія TransactedReceiveScope містить дію Receive і текст. Транзакція, що виділяється службі, залишається зовнішньої протягом виконання тексту TransactedReceiveScope. Транзакція завершує роботу, коли завершується виконання тексту.

Безпека робочих процесів.

Windows Workflow Foundation (WF) об'єднується з кількома різними технологіями, наприклад Microsoft SQL Server і Windows Communication Foundation (WCF). У разі неправильної налаштування взаємодія з цими технологіями може викликати проблеми безпеки в робочому процесі.

Проблеми безпеки SQL.

Якщо використовується велика кількість дочірніх дій, розташувань, закладок, розширень вузла і областей або використовуються закладки з дуже великим обсягом корисних навантажень, може бути вичерпана пам'ять або під час зберігання буде виділено надмірно великий обсяг простору бази даних. Уникнути цього можна за допомогою заходів безпеки на рівні об'єкта і на рівні бази даних.

Якщо використовується SqlWorkflowInstanceStore, сховище примірників має бути захищене.

Конфіденційні дані в сховищі примірників повинні бути зашифровані.

Безпека в SQL.

Сховище примірників робочих процесів SQL використовує наступні ролі безпеки бази даних для забезпечення безпечного доступу

до відомостей про стан примірників у базі даних зберігання.

System.Activities.DurableInstancing.InstanceStoreUsers. Ця роль має доступ на читання та запис для всіх відкритих уявлень і права на виконання збережених процедур, які беруть участь у створенні, завантаження та збереження примірників.

System.Activities.DurableInstancing.InstanceStoreObservers. Ця роль має доступ тільки для читання відкритих уявлень.

System.Activities.DurableInstancing.WorkflowActivationUsers. Ця роль має дозвіл на виконання збережених процедур, які беруть участь у процесі активації екземпляра. Обліковий запис користувача, в якій виконується універсальне провідне додаток (наприклад, служба управління робочого процесу для функції розміщення Windows Server AppFabric), повинна бути додана до цієї ролі бази даних.

Проблеми безпеки WCF.

Кінцеві точки Windows Communication Foundation (WCF), що використовуються в робочих процесах, повинні бути захищені. Авторизацію на рівні вузлів можна реалізувати за допомогою ServiceAuthorizationManager.

Також слід зауважити, що клієнт, який має доступ до даних власного примірника у сховище примірників, також отримує доступ і до всіх інших примірників в цьому сховищі примірників. Сховище примірників не підтримує вказівку прав доступу на рівні екземпляру. Для забезпечення доступу до різних сховищ примірників необхідно створити окремі сховища екземплярів і налаштувати доступ до них різних груп і користувачів.

Відображення інформаційної моделі на Windows workflow foundation 4.

Після побудови моделі потоків робіт, необхідно приступити до процесу розробки робочих процесів. Існує декілька способів їх опису:

“Декларативний” метод – це режим розробки з використанням лише workflow. В даному випадку workflow визначає структуру роботопотоку і бізнес-логіки.

Спосіб з використанням workflow і коди C#/VB, або на будь-якій іншій мові програмування. Тут workflow описує роботопотік, а Code-beside визначає останню логіку програми.

Лише код – описується на мовах програмування (C#/VB), а створення роботопотоку відбувається в конструкторі основного класу [56].

Для опису робочих потоків потрібно вибрати той спосіб, який найкраще підходить для вирішення конкретного завдання і подальшого управління бізнес-процесом.

Windows workflow foundation 4 – це технологія опису роботопотоків[55].

Розглянемо відображення інформаційної моделі на Windows workflow foundation 4.

Представлення типів об'єктів.

В загальному випадку тип об'єкту в інформаційній моделі буде трансльований в тип елементу в workflow. Як ім'я елементу можна використовувати назву типа об'єкту або, якщо потрібно зберегти місце на диску, це ім'я можна скоротити. Часто для тегів елементів застосовуються такі короткі імена.

Це пов'язано не з економією дискового простору, а з тим, що документ XML стає більш читаним: такі теги не дуже відволікають увагу від вмісту. Якщо тип об'єкту входить до складу ієрархії типів, потрібно визначити, на якому рівні ієрархії заснувати елементи XML [57].

Представлення зв'язків.

Для представлення деяких зв'язків моделі використовуються вкладені елементи структури документа XML. Очевидними кандидатами на подібну виставу є зв'язки "вмісту". Елемент може міститися безпосередньо лише в одному елементі, що його породжує, так що інші зв'язки моделі слід встановлювати інакше. На практиці це означає використання якого-небудь різновиду посилань [14].

Представлення властивостей.

Виявивши властивості в своїй інформаційній моделі, ми стикаємося з класичною дилемою: у документі XML представляти його за допомогою атрибуту або вкладеного (породженого) елементу. Розглянемо практичні переваги і недоліки кожного підходу [58].

Баланс всіх чинників залежить від конкретного застосування [33].

**Висновки до другого розділу**

В даному розділі описано процес опису логіки роботи програми, у випадку WF використовуються два поняття – activity і workflow. Існує декілька підходів до динамічного моделювання: моделі робочих процесів, моделі потоків даних, об'єктні моделі, життєві цикли об'єктів, варіанти використання, діаграми взаємодії об’єктів. Для керування потоками робіт використовуються використовуються сучасні case-засоби, спеціалізовані і універсальні мови: Bp-win. Для опису потоків робіт використовуються діаграми станів, які представлені графом.

Проаналізовано поняття workflow, наведено архітектуру WWF, використання якого можливе за допомогою додаткового налаштування для Visual Studio, що дозволяє забезпечити наочність і прозорість програмної логіки, використання WWF підніме на якісно новий рівень вбудовані засоби автоматизації / адаптації додатків, WWF стає новим стандартним елементом платформи Windows з уніфікованим інтерфейсом і єдиними правилами застосування, що істотно спрощує завдання розробників з просування продуктів на його основі, архітектура WWF дає можливість динамічно змінювати виконуються моделі.

В дипломній роботі для опису потоків робіт використовуватимемо технологію Windows workflow foundation 4 разом з Code-beside, тобто структура роботопотоку розроблена як workflow, а логіка розроблена як бібліотека розроблена мовою програмування C#.

**РОЗДІЛ 3**

**РОЗРОБКА ВЕБ-ДОДАТКУ СИСТЕМИ АДМІНІСТРУВАННЯ КОНФЕРЕНЦІЙ**

**3.1. Огляд існуючих систем адміністрування конференцій**

Розвиток ринкових відносин наукових і соціальної сфер, впровадження нових технологій різних галузей, спричиняє за собою попит на програмне забезпечення, яке здатне автоматизувати і прискорити процес виконання різних операцій, зокрема послуг в сфері організацій конференцій, без яких не обходиться діяльність великої кількості компаній та державних структур.

На сьогоднішній день існують різні системи, орієнтовані на управління конференцією. Прикладом є достатньо поширена на сьогоднішній день система "EasyChair".

EasyChair - це система управління конференцією, що надає вільний доступ до основних своїх функцій, яка є гнучкою, простою у використанні і має безліч можливостей пристосуватися до різних моделей конференції. It is currently probably [the most commonly used conference management system.](http://74.125.87.132/translate_c?hl=uk&sl=en&u=http://www.easychair.org/users.cgi&prev=/search%3Fq%3Deasychair.org%26hl%3Duk%26lr%3D%26rls%3Dcom.microsoft:*:IE-Address%26rlz%3D1I7SKPB&rurl=translate.google.com&usg=ALkJrhj23VzYgB1YYItcSlMfBRMRBURkhA) На даний час є однією з найбільш часто використовуваних систем, що забезпечують можливості управління конференцією [22].

EasyChair has been designed to help programme chairs to cope with the complexity of the refereeing process.EasyChair розроблена, щоб допомогти впоратися з труднощами в процесі рецензування. The current version supports Поточна версія підтримує

* management and monitoring of the programme committee; управління та контроль даними;
* sophisticated and flexible management of the access of PC members and referees to papers and conflicts of interests; витончене і гнучке управління доступом ПК членів і рецензентів до документів;
* automatic paper submission; автоматичне додавання статей;
* paper assignment based on the preferences of PC members; статті подаються за заключеням рецензентів;
* list of the latest events; перелік останніх подій;
* sending email to PC members, referees and authors; відправку електронної пошти на комп'ютери учасників, рецензентів та авторів;
* monitoring email; моніторинг електронної пошти;
* online discussion of papers; он-лайн обговорення документів;
* automatic preparation of conference proceedings автоматичну підготовки конференції.

and many other features.Також кFor each of the supported models the users can choose among various options to adjust the system to their requirements.ККористувачі можуть вибирати різні варіанти по адаптації системи до їх потреб. For example, she can choose to use anonymous submissions. Наприклад, вони можуть використовувати анонімні матеріали [20].

OpenConf - ще одна подібна система, що значно полегшує представлення і аналіз процесів для проведення конференцій, семінарів і симпозіумів. The software is flexible enough that OpenConf is also used for journals, grants, and books. Програма є досить гнучкою, розроблена з використанням мови PHP та бази даних MySQL [26].

Conference management system - веб-система, яке підтримує організацію наукових конференцій. It helps the program chair, the conference organizers, the authors and the [reviewers](http://74.125.87.132/translate_c?hl=uk&sl=en&u=http://en.wikipedia.org/wiki/Peer-review&prev=/search%3Fq%3Dconference%2Bmanagement%2Bsystem%26hl%3Duk%26lr%3D%26rls%3Dcom.microsoft:*:IE-Address%26rlz%3D1I7SKPB&rurl=translate.google.com&usg=ALkJrhhY_u7XeuprHQF-CHWM2xdh_FYjVw) in their respective activities. Вона допомагає в роботі організаторам конференції, а також авторам та рецензентам, надає зручність та сприяє прискоренню їх діяльності [24].

A conference management system can be regarded as a domain-specific [content management system](http://74.125.87.132/translate_c?hl=uk&sl=en&u=http://en.wikipedia.org/wiki/Content_management_system&prev=/search%3Fq%3Dconference%2Bmanagement%2Bsystem%26hl%3Duk%26lr%3D%26rls%3Dcom.microsoft:*:IE-Address%26rlz%3D1I7SKPB&rurl=translate.google.com&usg=ALkJrhghRmGlHcwztjhYfHW9WACp2e1vuw) .Conference management system може розглядати в якості домену конкретні системи управління контентом. Similar systems are used today by editors of [scientific journals](http://74.125.87.132/translate_c?hl=uk&sl=en&u=http://en.wikipedia.org/wiki/Scientific_journal&prev=/search%3Fq%3Dconference%2Bmanagement%2Bsystem%26hl%3Duk%26lr%3D%26rls%3Dcom.microsoft:*:IE-Address%26rlz%3D1I7SKPB&rurl=translate.google.com&usg=ALkJrhhxKsV8VMPYK-e7-JEmuPbXtdFLYA) . Подібні системи використовуються сьогодні редактори наукових журналів.

ConfMaster – представляє собою веб-систему для підготовки та управління наукових конференцій та семінарів. ConfMaster підтримує роботу учасників, а також адміністративного персоналу. Вона охоплює весь процес від підготовки конференції та управління. Наприклад, подання документа, призначення рецензента, розгляд, реєстрація учасників, включаючи кредитні картки платежі та завантаження готових документів [26].

Також можливе створення власної системи на базі вільнопоширюваних cms систем, наприклад Drupal.

Усі ці системи підтримують різноманітну функціональність, кожна з яких більш орієнтована на специфічний вид діяльності. Тому я вирішив створити свою систему управління конференцією і реалізувати у ній основні необхідні функції необхідні для підготовки конференції і на цьому прикладі ознайомитися з концепцією управління потоками робіт.

**3.2. Опис і функціональні можливості системи**

В своїй роботі я створив веб-додаток, який автоматизує процеси, що проводяться в ході організації конференції, реалізовує управління потоками робіт, підвищує контрольованість та передбаченість усіх можливих дії, покращує процес взаємодії між організаторами, рецензентами та учасниками конференції.

Основними функціями програми – є забезпечення ефективної взаємодії з учасниками конференції, а також налагодження роботи зі статтею, її рецензуванням та публікацією.

**Вимоги**

1. Модуль повинен надавати можливість користувачам приймати участь у конференції
2. Реалізовувати можливість додавання та зберігання статей
3. Мати можливість рецензування статей
4. Реалізовувати відправку електронної пошти на комп'ютери учасників, рецензентів та авторів;
5. Підтримувати обробку статей включаючи можливості їх прийняття чи відхилення.
6. Підтримка різних прав доступу
7. Інтерфейс модулю повинен забезпечувати її доступність для широкого кола користувачів, мати можливість зміни мови, бути легким та зручним у використанні.

Практична цінність:

Для учасників конференції розроблений модуль надає наступні можливості:

* приймати участь у конференції в ролі слухача чи доповідача;
* додавати нові статті;
* отримувати повідомлення з інформацією стосовно результату рецензування статей;
* подавати заявки на гуртожиток.

Для Рецензентів:

* автоматично повідомляє про надходження нової статті для рецензування;
* дає можливість відмовитися рецензувати дану статтю;
* надає доступ до необхідних ресурсів.

Для організаторів:

* вести облік інформації про учасників конференції, статті, рецензування, публікації;
* автоматично повідомляє про надходження нових статей, надає можливість додати рецензентів до статей;
* повідомляє про відхилення рецензентом статті, чи відмову рецензувати;
* надає інформацію про потребу в гуртожитку учасників конференції, та про відмову у участі;
* надає доступ до ресурсів та даних.

Також в програмі реалізований багато користувацький режим з можливістю розділення прав доступу до функцій програми.

**3.3. Проектування потоків робіт**

*Представлення потоків робіт у вигляді машини станів.*

Потоки робіт зручно представляти у вигляді машини станів, тому що вона моделює поведінку об’єкту, при якій реакції на майбутні події залежать від попередніх подій.

Машина станів – це комп'ютерна програма, яка складається з:

* подій, на які реагує програма;
* станів, в яких програма перебуває між подіями;
* переходів між станами при реагуванні на події;
* дій, що виконуються в процесі переходів;
* змінних, які містять значення, необхідні для виконання дій між подіями [32].

Машини станів найбільш корисні в ситуаціях, в яких поведінка управляється багатьма різними типами подій, а реакція на певну подію залежить від послідовності попередніх подій.

Події, які управляють кінцевими автоматами, можуть бути зовнішніми по відношенню до комп'ютера і виходити від клавіатури, миші, таймера чи мережевої активності, або внутрішніми, витікаючими від інших компонентів прикладної програми або від інших застосувань.

Існує два поширені представлення машин станів:

*Орієнтовані графи.* Еліпси представляють стани, а стрілки між ними - переходи, над якими вказані події і дії.

*Двовимірні таблиці.* Стовпці і рядки представляють події і стани, а комірки містять дії і переходи [33].

Ці представлення еквівалентні, але роблять акцент на різних аспектах проекту. Обидва представлення корисні і використовуються в даній статті.

Розробка керованих подіями програм за допомогою потоків робіт дещо складніше за звичайне процедурне програмування; така розробка вимагає більше дисципліни загалом і більше проектної роботи зокрема. При хорошому виконанні в результаті можна отримати простіший програмний код, меншу тривалість тестування і полегшений супровід. Не дивлячись на це, складність цього підходу виправдана не для всіх керованих подіями програм. Якщо, наприклад, діапазон подій невеликий, або дії, що запускаються подіями, завжди одні і ті ж, додаткові витрати ресурсів на розробку можуть не виправдатися.

*Машини станів і середовище виконання.*

Машини станів управляються подіями і потребують спосіб прив'язки до подій, про які йде мова, в їх середовищі виконання. Такі прив'язки, які називаються обробниками подій, є дуже невеликими фрагментами коду, вставлені в середу виконання так, щоб вони могли виконуватися при настанні конкретної події [36].

Для виконання обробники подій потребують наступну базову інформацію:

* тип події, що настала (наприклад, натиснення кнопки або витік часу таймера);
* контекст події (наприклад, який мережевий запит виконаний);
* розміщення власних змінних і методів машин станів. Для побудови керованих подіями машини станів використовувалася технологія Windows workflow foundation 4, яка описує роботопотік, і мова програмування C#, яка визначає іншу логіку.

*Методична розробка поведінки об’єктів.*

Основні інгредієнти для побудови машини станів - це події, на які вона реагує, і стани, в яких вона чекає між подіями.

Проект повинен передбачати всі можливі події для всіх можливих станів об'єкту:

* чи може подія настати в цьому стані;
* яку дію слід виконати для обробки подій;
* у який стан буде виконаний перехід після настання події;
* які змінні слід запам'ятати між подіями.

Процес проектування зазвичай починається з побудови діаграми переходів на рисунку, на якому стани показані у вигляді еліпсів, а переходи – у вигляді стрілок, що сполучають еліпси, і закінчуються таблицею, в якій події і стани перераховуються відповідно в заголовках рядків і стовпців [54].

У кожній комірці таблиці перераховані дії, які повинні виконуватися, коли в певному стані відбудеться певна подія, або показано, що дана подія не може настати в даному стані.

Зазвичай потрібно декілька ітеративних повторень цієї проектної процедури, щоб отримати граф або таблицю без помилок. Якщо машина станів має багато подій і станів, то процедура може бути достатньо утомливою, тому потрібна певна дисципліна, щоб методично працювати над кожним елементом таблиці в процесі кожної ітерації. Завдяки цьому можна продумати, яка поведінка має бути у всіх можливих ситуаціях. Можна побачити дороги для подальшого удосконалення або деталізації поведінки, можливо, знадобиться більше станів, ніж передбачалося спочатку, а можливо, доведеться перетасувати дії між комірками, щоб визначити правильну поведінку в кожній ситуації. Ця методична процедура проектування потоків робіт виправдана, хоча і утомлива.

**3.4. Створення діаграми станів об'єктів**

Почнемо проектування з розгляду базової поведінки, для прикладу розглянута поведінка об'єкту Article. Після завантаження статті вона поступає на рецензування, де визначається чи буде стаття представлена на конференції.

Таким чином машині станів об'єкту Article доведеться реагувати на такі події:

* AddReviewer – подія яке виникає при додаванні до статті рецензента;
* RefuseToRewiew – відмова рецензента рецензувати дану статтю;
* AgreedReview – подія виникає за згодою рецензента рецензувати статтю;
* ArticleDeviation – відхилення статті;
* AcceptArticle – схвалення статті;
* HasWorkedTimer – події таймера.

Потім слід створити стани, в яких кінцевий автомат чекатиме між подіями.

Стани об'єкту стаття:

* ArticleLoaded – стан в який переходитиме об'єкт з початкового, при завантаженні статті;
* ReviewerIsOffered – рецензент запропонований, в цьому стані об'єкт чекатиме згоди або відмови рецензента дати рецензію даній статті, або ж чекає коли спрацює таймер;
* Reviewing – рецензент дав згоду, очікується рецензія;
* Approval – схвалення і публікація статті, після чого потік завершує свою роботу.
* Deviation – відхилення статті, в цьому стані відправляється повідомлення авторові і потік закінчує свою діяльність.

На рис. 3.1 цей процес відображується у вигляді діаграми станів, на якому стани відображені у вигляді еліпсів, переходи у вигляді стрілок, що сполучають ці еліпси, а події і дії – як написи над стрілками.

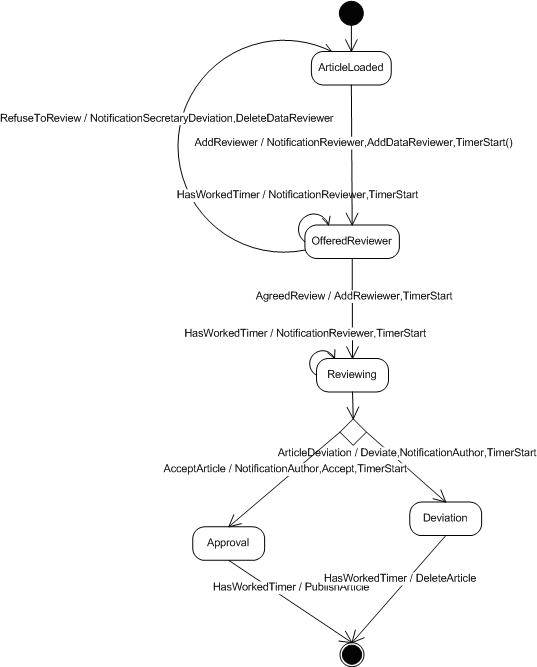


Рис. 3.1. Діаграма станів об'єкту Article

Для кожної події потрібно виконувати деякі дії:

* при настанні події AddReviewer в стані ArticleLoaded, необхідно відправити рецензентові повідомлення з проханням дати рецензію статті або відмовитися її рецензувати;
* при виникненні події RefuseToRewiew в стані ReviewerIsOffered, потрібно буде відправити повідомлення секретарю про відмову рецензента рецензувати дану статтю, і про необхідність запропонувати нового рецензента;
* при настанні події AgreedReview необхідно додати рецензента і запустити таймер;
* подія ArticleDeviation або AcceptArticle в стані Reviewing спричиняє собою відправку сповіщення авторові про те що його стаття відхилена або прийнята;
* при настанні події HasWorkedTimer потрібно буде відправляти нагадування рецензентові про необхідність дати рецензію, а також відхилити або прийняти статтю, залежно від того в якому стані сталася дана подія.

На діаграмі станів ці дії написані після подій, які їх запускають.

**3.5. Переведення діаграми стану в таблицю станів**

Представлення у вигляді графа, показане на рис. 3.1, – хороший спосіб почати проектування роботопотоків, але представлення у вигляді таблиці краще підходить для завершення проектування, оскільки дозволяє наочно відображати всі комбінації подій і станів.

Щоб перевести діаграму станів в таблицю станів, в якості заголовків рядків я обрав назви подій, а як заголовки стовпців - назви станів. Порядок назв може бути довільним.

Потім описуються дії і наступний стан для кожної події у відповідній комірці таблиці, як показано у таблиці 3.1.

*Заповнення таблиці станів.*

Щоб завершити проектування роботопотоку, необхідно продумати, як заповнити кожну порожню комірку в цій таблиці. Таблиця станів об'єкту Article наведена в таблиці 3.1.

Таблиця 3.1.

Таблиця станів потоку робіт

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **ArticleLoaded** | **Reviewer Is Offered** | **Reviewing** | **Approval** | **Deviation** |
| **AddReviewer** | AddDataReviewer(), Notification  Reviewer(), TimerStart() |  |  |  |  |
| **RefuseToReview** |  | Notification  Secretary  Deviation(), DeleteData  Reviewer() |  |  |  |
| **AgreedReview** |  | AddReviewer(), TimerStаrt() |  |  |  |
| **ArticleDeviation** |  |  | Deviate(), TimerStаrt(), Notification  Author() |  |  |
| **AcceptArticle** |  |  | Accept(), TimerStаrt(), Notification  Author() |  |  |
| **HasWorkedTimer** |  | Notification  Reviewer(), TimerStаrt() | Notification  Reviewer(), TimerStаrt() | Publish  Article() | Delete  Article() |

Для кожної комірки необхідно вирішити, чи може дана подія статися в цьому стані, і якщо може, то які дії повинні виконуватися в цій ситуації і яким має бути наступний стан. Це утомлива, але абсолютно необхідна фаза процесу проектування. Порядок заповнення комірок не має значення. Зазвичай до цього кроку в процесі розробки повертаються неодноразово, ще і ще раз продумавши кожну комірку, часто переглядаючи її вміст, причому кожного разу по-новому. Нерідкі також додавання (або видалення) станів по ходу процесу, що викликає подальший перегляд.

Комірки таблиці, що залишилися порожніми (перекреслені), відповідають ситуації “не повинно настати”, тобто в даному стані подібна подія статися не може. Табличне представлення роботопотоку завжди може бути переведене в графічне, оскільки ці представлення еквівалентні один одному.

**3.6. Складання списку змінних стану**

Після складання таблиці і діаграми станів корисно переглянути їх ще раз, щоб скласти список змінних, які автомату потрібно буде запам'ятовувати між подіями, щоб він міг виконувати родинні дії в різних комірках і однозначно ідентифікувати стан в якому він знаходиться.

Роботопотоку об'єкту Article знадобляться змінні станів перераховані в таблиці 3.2.

Таблиця 3.2.

Таблиця змінних

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attributes of object article** | **ArticleLoaded** | **OfferedReviewer** | **Reviewing** | **Approval** | **Deviation** |
| **Author** | ≠0 | ≠0 | ≠0 | ≠0 | ≠0 |
| **NameArticle** | ≠0 | ≠0 | ≠0 | ≠0 | ≠0 |
| **Url** | ≠0 | ≠0 | ≠0 | ≠0 | ≠0 |
| **LoadingDate** | Now | ≠0 | ≠0 | ≠0 | ≠0 |
| **InquiryReviewer** | *0* | ≠0 | ≠0 | ≠0 | ≠0 |
| **EmailReviewer** | *0* | ≠0 | ≠0 | ≠0 | ≠0 |
| **DateInquiryReviewer** | *0* | Now | ≠0 | ≠0 | ≠0 |
| **Reviewer** | *0* | *0* | ≠0 | ≠0 | ≠0 |
| **DateAdditionReviewer** | *0* | *0* | Now | ≠0 | ≠0 |
| **Review** | *0* | *0* | *0* | 1 | -1 |

Стовпці таблиці – це стани, в яких знаходиться об'єкт, рядки позначають значення змінної в даному стані. Комірки із значенням змінної «0» - означають що в даному стані змінна не визначена, значення «≠0» позначає що в змінній зберігається відповідна інформація, а «Now» говорить про значення дати в даному стані, яка відповідає поточній. Коли заповнена таблиця станів і створений список змінних станів всіх об'єктів, етап проектування можна рахувати закінченим.

**3.7. Реалізація веб-додатку на основі розроблених моделей**

Реалізація веб-додатку на основі розроблених моделей складається з наступних етапів:

1. Побудова об'єктної моделі проекту. Об'єктна модель дозволить забезпечити групування змінних і методів в об'єкті. Створення об'єктів дозволить згрупувати змінні і методи в один об'єкт. Наприклад на рис. 3.2 представлено модель реалізовану в окремій бібліотеці класів із зображенням основних об'єктів додатка.

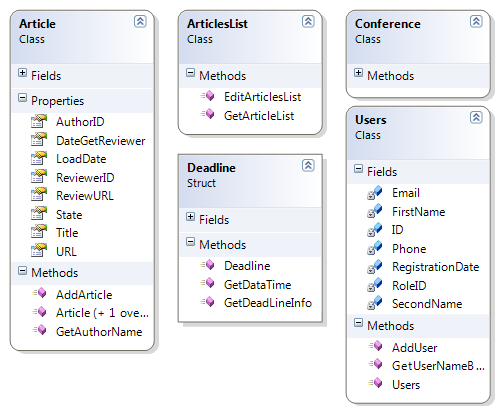


Рис. 3.2. Об’єктна модель

1. Потім необхідно визначити основні властивості об'єктів, які можуть бути змінними або методами, та їх початкові значення, описати події цих об'єктів.
2. На наступному кроці необхідно реалізувати прив'язку (перехоплення) подій об'єктів, налаштування таймерів і перехоплення подій таймерів. Після цього кроку об'єкти можна вважати сконструйованими.
3. Опис потоку робіт використовуючи технологію Windows workflow foundation 4.

На цьому кроці таблицю станів необхідно реалізувати у вигляді workflow, використовуючи імена станів і подій безпосередньо як ідентифікатори, вміст не порожніх комірок таблиці вказуватимуть на функції, які виконають дії для подій за допомогою виклику допоміжних методів (таких як запуск і відміна таймерів), а потім повернуть наступний стан [55].

Розглянемо побудований workflow конференції (див. рис. 3.3).

Як бачимо workflow дуже легко зрозуміти, тому що він наочно відображає весь процес роботи. Також у даному workflow використовуються змінні різних типів (див. рис. 3.4).

Після того, як запускається даний workflow, ми переходимо на послідовний workflow ChangeConferenceState, котрий перевіряє потрібність переводу конференції в інший стан, та якщо потрібно, то переводить у цей стан, та записує в змінну NewConferenceState новий стан. Якщо стан конференції не змінився після виконання послідовного workflow ChangeConferenceState, то в змінну NewConferenceState записується значення -1.

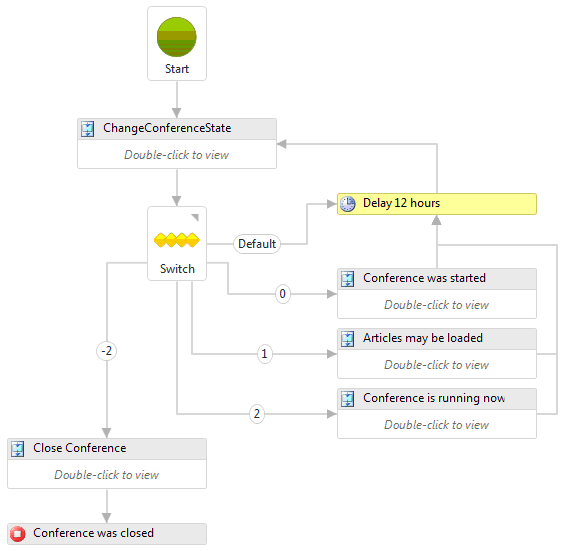


Рис. 3.3. Головний workflow конференції

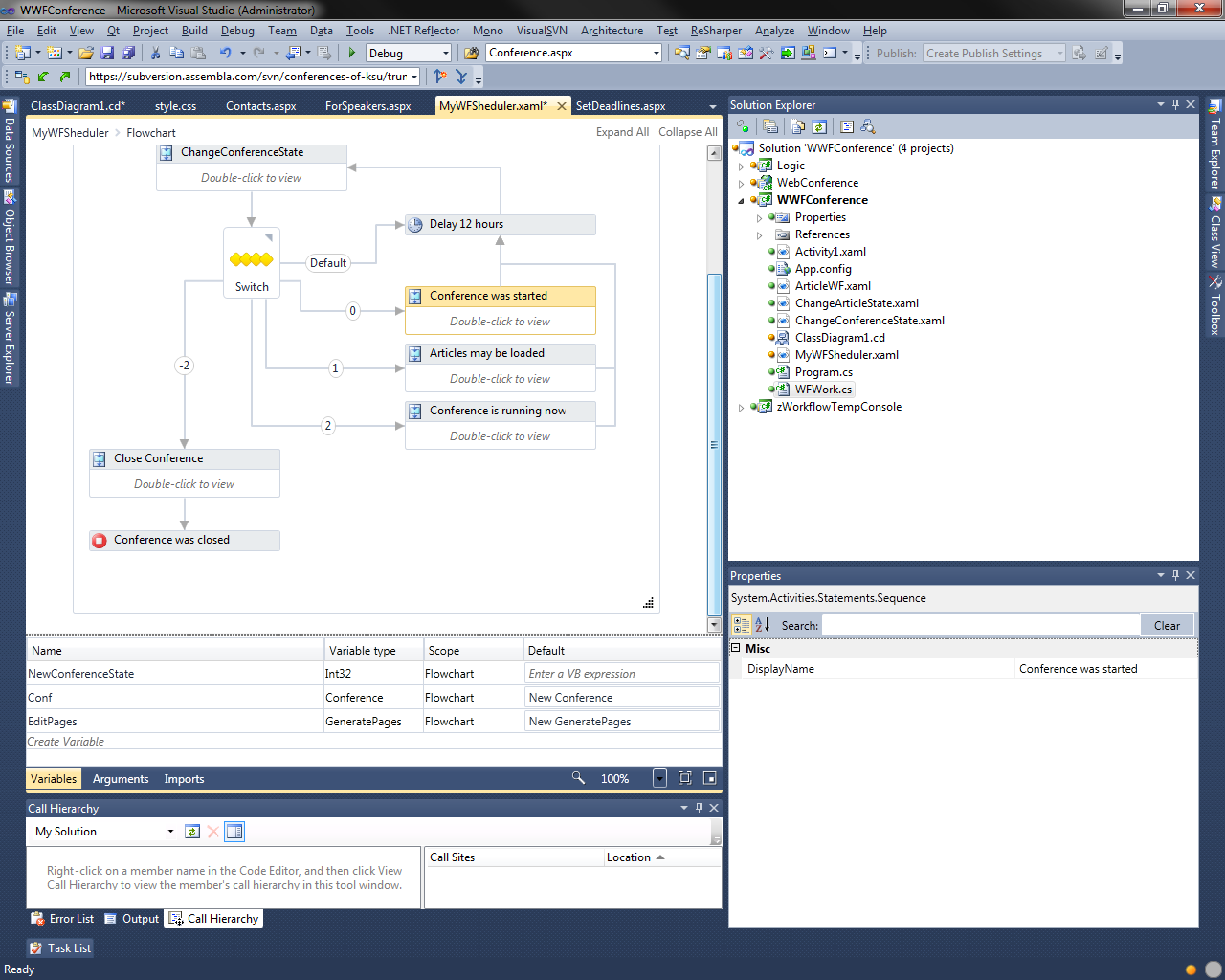


Рис. 3.4. Змінні головного workflow конференції

Далі використовуючи дію вибору switch перевіряємо новий стан. Якщо він дорівнює значенню -2, то це означає, що конференція вже завершилась, та слід видалити усі сторінки сайту та посилання на них з меню, які вже не будуть потрібні. Так, наприклад, після завершення конференції вже будуть не потрібні сторінки реєстрації на цю конференцію, сторінка загрузку статті та інші, тому їх можна видалити. Звичайно крім видалення сторінок будуть редагуватися інші та створюватися нові такі, наприклад, як фотогалерея учасників. В моїй роботі це виконує послідовний workflow Close Conference. Після виконання усіх дій відносно редагування сторінок, головний workflow завершується, вивантажується з пам’яті та видаляється з бази даних призначеної для зберігання екземплярів workflow.

За аналогією будуть виконуватися зміни над сторінками, якщо значення нового стану конференції буде дорівнювати 0, 1 або 2. Але на відміну від попереднього, після завершення виконання змін щодо сторінок сайту, ми переходимо в дію Delay 12 hours. Також ми переходимо в цю дію, якщо у нас не відбувалося зміни стану конференції. Дія Delay 12 hours вивантажує екземпляр головного workflow з пам’яті та зберігає його у базі даних, та через 12 годин екземпляр виконує зворотну дію, тобто він вивантажується з бази даних в тому стані, в котрому туди потрапив та ми знову користуємося нашим.

Завжди можливо перевірити наявність збереженого workflow у базі даних, для цього потрібно відкрити таблицю InstancesTable (див. рис. 3.5) в базі даних вказаної для збереження екземплярів workflow.



Рис. 3.5. Збережені екземпляри workflow

Далі знову потрапляємо в послідовний workflow ChangeConferenceState та переходимо за тим же принципом як і раніше.

**3.8. Створення бази даних конференцій**

Для збергання даних workflow потрібна база даних MS SQL Server версії 2005 або 2008. У роботі використано базу даних MS SQL Server 2008. Тому базу даних конференцій зроблено також на MS SQL Server 2008. Згідно вимогам до розробки веб-додатку слід виділити наступні основні сутності та їх відносини (див. рисунок 3.6.). Таблиця Users містить інформацію щодо користувачів системи, таблиця UsersRoles містить інформацію щодо ролей користувачів, таблиця Artciles містить інформацію щодо прийнятих статей та тезісів, таблиця ArticleStates містить інформацію щодо станів завантажених статей. Таблиця DeadLines містить інформацію щодо важливих дат, що відображають переходи станів конференції. Таблиця ConferencesStates містить інформацію щодо станів конференції. Таблиця Pages містить інформацію щодо створених веб-сторінок.

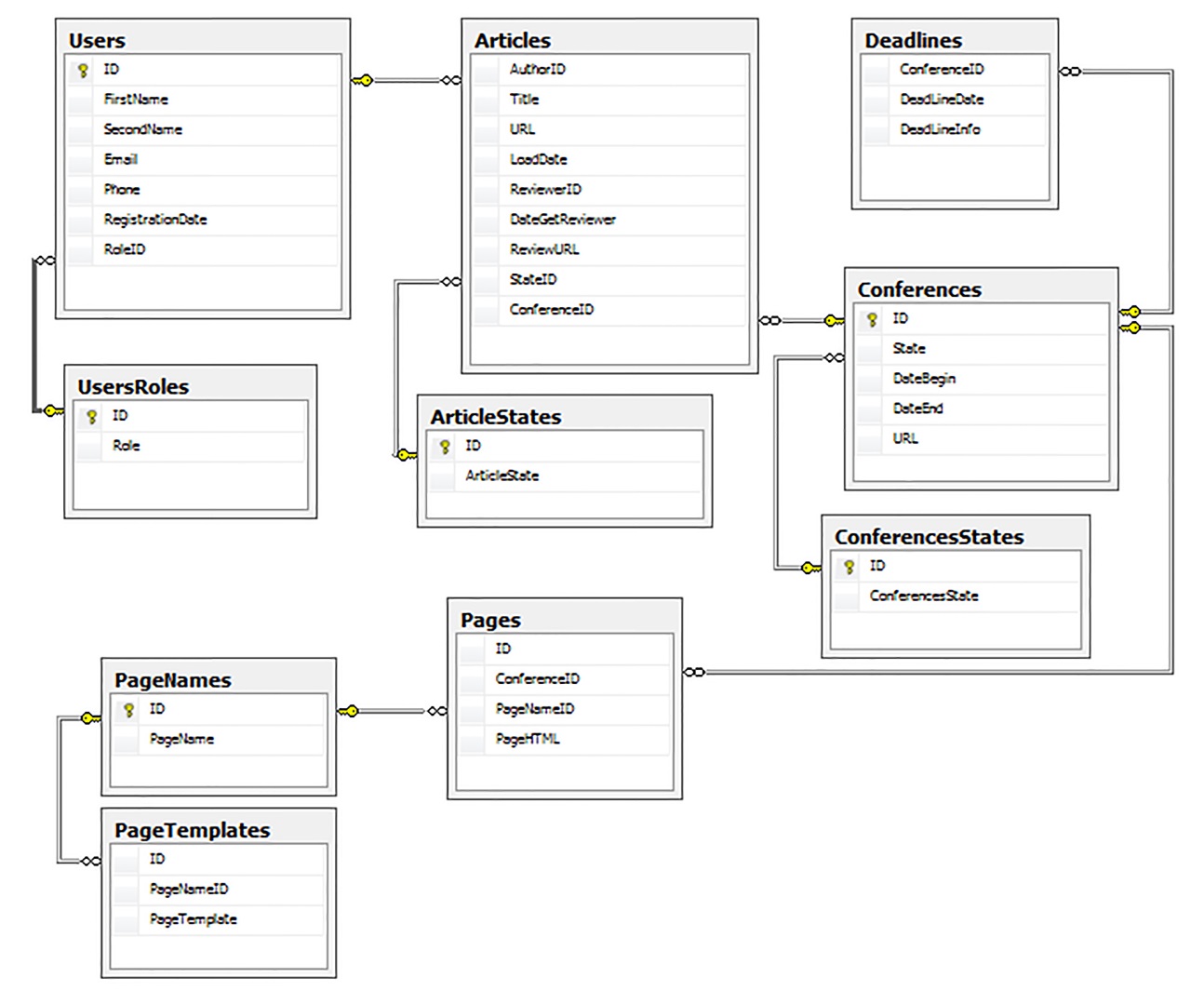


Рис. 3.6. База даних системи. Основні таблиці

Таблиця PageNames містить інформацію щодо назв сторінок. Таблиця PageTemplates містить інформацію щодо шаблонів веб-сторінок.

**3.9. Створення дизайну веб-додатку**

Технологія ASP.NET дуже популярна, широко використовується для створення малих, середніх, великих корпоративних веб-систем, а також для дуже великих корпоративних систем.

ASP.NET – це технологія, яка дозволяє істотно поліпшити процес створення сайтів і порталів, істотно підвищити їх функціональність і безпеку. Ось тільки деякі нові можливості ASP.NET: підтримує розбиття сторінки на частини, дозволяє створювати нові елементи управління, здійснює підтримку шаблонів, дозволяє конфігурувати сторінки, покращує механізм аутентифікації на основі форм, містить безліч елементів управління, які дозволяють прискорити і полегшити створення веб-додатків і інше. C# (Сі-шарп) – об'єктно-орієнтована мова програмування для платформи .NET. Мова заснована на строгій компонентній архітектурі і реалізує передові механізми забезпечення безпеки коду [6]. C# був створений спеціально для технології ASP.NET. В той же час, на C# повністю написана і сама ASP.NET. Виділення і об'єднання кращих ідей сучасних мов програмування робить мову C# не просто сумою їх достоїнств, а мовою програмування нового покоління. Саме тому для створення свого веб-додатку я використовував мову програмування C# і технології ASP.NET та Windows workflow foundation, для створення інтерфейсу і реалізації системи.

Інтерфейс веб-додатку можна побачити на рис. 3.7.



Рис. 3.7. Сторінка відвідувача конференції

Ця сторінка відображає дані, до яких має доступ зареєстрований відвідувач конференції. Звідки він має можливість переглядати та змінювати данні. Також створені подібні сторінки для рецензування та організаторів, що мають змогу переглядати та змінювати необхідні дані тощо.

**Висновки до третього розділу**

В даному розділі описано функціональні можливості системи, зпроектовано потоки робіт, створено діаграми станів об’єктів, переведено діаграму стану в таблицю станів, складено список змінних стану, розроблено базу даних конференції, розроблено дизайн веб-додатку.

**РОЗДІЛ 4**

**ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА ДОСЛІДЖЕННЯ**

**4.1.** [**Оцінка**](http://ua-referat.com/%D0%9E%D1%86%D1%96%D0%BD%D0%BA%D0%B0) **якості програмного забезпечення**

Оцінка якості програмного забезпечення зовсім нова дисципліна. Коли цей напрям отримає достатній розвиток, то будуть розроблені хороші методи оцінки, але в даний час є найбільш суперечливі думки про те, які характеристики програмного забезпечення слід вимірювати. Методологія розробки програмного забезпечення розвивається так швидко, що встановлення окремих оцінок може призвести до вкорінення практики програмування, яка згодом виявиться неправильною.

Боемі, Браун і Лайпоу займалися проблемою обчислення єдиної узагальнюючої міри якості і прийшли до висновку, що це неможливо, тому що входить в протиріччя з окремими характеристиками якості. Керівництво має прийняти рішення про відносну важливість наступних характеристик:

1. Своєчасне виконання.
2. Ефективність використання таких [ресурсів](http://ua-referat.com/%D0%A0%D0%B5%D1%81%D1%83%D1%80%D1%81%D0%B8), як:
   1. процесори;
   2. пам'ять;
   3. [периферійні пристрої](http://ua-referat.com/%D0%9F%D0%B5%D1%80%D0%B8%D1%84%D0%B5%D1%80%D1%96%D0%B9%D0%BD%D1%96_%D0%BF%D1%80%D0%B8%D1%81%D1%82%D1%80%D0%BE%D1%97);
3. Аспекти обслуговування програми, такі як:
   1. зрозумілість;
   2. придатність до модифікації;
   3. зручність перенесення з ПК на ПК.

Важливість характеристик з цього переліку змінюється в залежності від того, в якій організації використовується це програмне забезпечення.

Розробники програмних бібліотек можуть віддати [перевагу](http://ua-referat.com/%D0%9F%D0%B5%D1%80%D0%B5%D0%B2%D0%B0%D0%BB) ефективності зручність перенесення, в той час як творці систем обліку кадрів можуть зосередити свою увагу на модифікованості.

**Метрики Боемі, Брауна і Лайпоу.**

Щоб оцінити якість, необхідно визначити вимірювані характеристики. Боемі, Браун і Лайпоу описали ієрархічне [дерево](http://ua-referat.com/%D0%94%D0%B5%D1%80%D0%B5%D0%B2%D0%BE) характеристик програмного забезпечення, в якому напрям стрілок задає логічне слідування. Так, наприклад, добре підтримувана програма повинна бути добре тестованою, що розуміється і модифікується. Найвищий рівень структури відображає використовувану оцінку якості програмного забезпечення. Боемі, Браун і Лайпоу підкреслюють переваги пакетів програм і вважають, що найбільше значення для них мають відповіді на такі питання.

Як добре (просто, надійно, ефективно) можу я використовувати даний пакет у тому вигляді, як він є?

Наскільки просто його обслуговувати (розібратися в ньому, модифікувати, перевірити ще раз)?

Чи можу я користуватися цим пакетом, якщо зміню обладнання (зручність перенесення)?

Характеристики самого нижнього рівня представляють собою "примітиви", комбінації яких утворюють характеристики середнього рівня. Ці примітиви пропонуються як кількісних метрик, як самих примітивних характеристик, так і характеристик більш високих рівнів.

Боемі, Браун і Лайпоу розробили 51 можливу метрику оцінки примітивних характеристик, а потім провели [порівняння](http://ua-referat.com/%D0%9F%D0%BE%D1%80%D1%96%D0%B2%D0%BD%D1%8F%D0%BD%D0%BD%D1%8F) цих метрик за ступенем їх кореляції з якістю програми. Це детальна і складна схема, яка спирається на практичний досвід, однак, Боемі, Браун і Лайпоу не запропонували чіткої демонстрації її ефективності, надійності чи застосування в різних контекстах. Довгий список понять використовується скоріше як [контрольний](http://ua-referat.com/%D0%9A%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D1%8C) лист для рецензування програми, ніж як керівництво по її складанню.

**Метрики програмного забезпечення Джілба.**

Джілб призводить не претендує на повноту набір метрик програмного забезпечення. Він звертає увагу на те, що кожен додаток вимагає введення власних понять та інструментів; його [книга](http://ua-referat.com/%D0%9A%D0%BD%D0%B8%D0%B3%D0%B0) призначена для введення основних понять, від яких може відштовхнутися користувач.

Серед інших характеристик Джілб згадує надійність програми, яку він визначає як ймовірність того, що дана програма пропрацює певний період часу без логічних збоїв. [Прагматичної](http://ua-referat.com/%D0%9F%D1%80%D0%B0%D0%B3%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%BC) оцінкою програмної надійності є одиниця мінус відношення числа логічних збоїв до загальної кількості запусків.

Відношення кількості правильних даних до всіх даних наводиться Джілбом в якості міри точності (свободи від помилок). Так само, як Боемі, Браун і Лайпоу, Джілб вважає, що точність необхідна для надійності програми. Прецизійність визначається як міра того, наскільки часті помилки, зумовлені однаковими причинами. Джілб оцінює її дробом, у чисельнику якого стоїть число фактичних помилок на вході, а в знаменнику - загальна кількість спостережених помилок, причинами яких з'явилися ці помилки на вході. Так, наприклад, якщо одна помилка викликає протягом певного періоду часу 100 повідомлень про помилки, то прецизійність дорівнює 0.01.

Другий великий категорією, введеної Джілбом, є гнучкість, в яку входять:

* Логічна складність.
* Внутрішня гнучкість.
* Відкритість (адаптованість).
* [Толерантність](http://ua-referat.com/%D0%A2%D0%BE%D0%BB%D0%B5%D1%80%D0%B0%D0%BD%D1%82%D0%BD%D1%96%D1%81%D1%82%D1%8C) (до змін системи).
* Універсальність.
* Зручність перенесення.
* Сумісність.

В якості міри логічної складності Джілб запропонував число логічних "двійкових прийнять рішень". Така оцінка може бути отримана вручну або автоматично. Абсолютна логічна складність задається числом нестандартних виходів з операторів, у яких відбувається прийняття рішень. Джілб припускає, що логічна складність виявиться значимим фактором для передбачення вартості програми.

Крім цих, Джілб наводить ще велика кількість інших метрик, але це довге перерахування швидше будить уяву, ніж приносить користь. Робота Джілба демонструє нові можливості, проте реальне застосування цих ідей на практиці дає вражаючі результати. Більшість характеристик дуже важко отримати; збиває з пантелику і те, що оцінки сильно пов'язані, що утрудняє програмісту передбачення впливу зміни програми на деяку групу характеристик.

**Оцінка складності за Маккейбом.**

Маккейб описує оцінку складності за допомогою теорії графів і демонструє її застосування для управління, [тестування](http://ua-referat.com/%D0%A2%D0%B5%D1%81%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) і контролю за складністю програми. Слід зауважити, що в даному дослідженні Маккейб під складністю програми розумів її логічний складність. У його теорії передбачається, що складність не залежить від розміру, а тільки від структури виборів рішень у програмі.

Маккейб пропонує математичний метод, який дає кількісні [підстави](http://ua-referat.com/%D0%9F%D1%96%D0%B4%D1%81%D1%82%D0%B0%D0%B2%D0%B8) для модулярізаціі і дозволяє виявляти модулі, які буде важко тестувати або обслуговувати.

Згідно з його підходу обчислюється і контролюється число шляхів у програмі. У [математичні](http://ua-referat.com/%D0%9C%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0) передумови входить визначення цикломатичного числа V (G) для графа з n вершинами, e ребрами і p компонентами зв'язності:

V (G) = e - n + p

Маккейб використовує наступну теорему: в сильно пов'язаному   
графі G цикломатичне число дорівнює максимальному числу лінійно-незалежних циклів.

Застосовуючи цю теорему, Маккейб пов'язує з програмою орієнтований граф з одним виходом. Кожній вершині графа відповідає блок коду з послідовним управлінням, а кожній дузі відповідає розгалуження програми. Кожній вершини можна досягти з вхідної вершини і з кожної вершини може бути досягнута вихідна вершина. Цей граф сильно пов'язаний, так як для будь-якої пари вершин існує зв'язуючий їх шлях.

Загальний підхід полягає в оцінці складності програми за допомогою обчислення числа лінійно-незалежних шляхів, цикломатичної складності   
V (G), а також управління розміром програм за допомогою обмеження V (G) і використання V (G) як основи для методології тестування. Маккейб виявив, що розумної верхньою межею для цикломатичної складності є 10. Якщо програмісти переступають цю межу, їм слід або переписати програму, або розбити її на модулі.

Оцінка цикломатичної складності Маккейба корисна при підготовці тестових даних і може дати потрібну інформацію про логічну складності програми. Однак при такій оцінці не приймається до уваги вибір структур даних, алгоритмів, мнемонічних імен змінних чи коментарів, відсутня обговорення таких важливих понять, як зручність перенесення, гнучкість, ефективність. Необхідні додаткові дослідження, щоб прояснити, коли корисно використовувати цикломатичну складність. У розглянутому програмному комплексі циклічна межа складності модуля дорівнює 6, що не перевищує верхню межу складності. Це дозволяє зробити висновок про правильний підхід до написання програмного забезпечення інформаційної системи дипломного проекту.

**Доступність для сприйняття.**

Доступністю для сприйняття програми можна назвати її [психологічну](http://ua-referat.com/%D0%9F%D1%81%D0%B8%D1%85%D0%BE%D0%BB%D0%BE%D0%B3) складність, так як [психологічна](http://ua-referat.com/%D0%9F%D1%81%D0%B8%D1%85%D0%BE%D0%BB%D0%BE%D0%B3) складність пов'язана з тими ж характеристиками програми, які ускладнюють розуміння програми людиною.

Автори роботи "Predicting Software Comdivhensibility" [61] експериментували з 36 професійними програмістами, запропонувавши їм по 25 хвилин вивчати 3 програми, а потім відновити їх за 20 хвилин. Були використані 3 класу задач (інженерні, статичні і не чисельні) і 3 типу структурування (повне, часткове і неструктуровані програми). Було також введено 3 рівня мнемонічності імен змінних.

Результати експерименту показали, що гірше всього відновлюються неструктуровані програми, краще всього - частково структуровані. Рівень мнемонічності імен змінних не вплинув на проведення експерименту.

Важливим висновком цього експерименту було те, що на здатність правильно відтворювати програми вплинули індивідуальні особливості учасників, характеристики програми та рівень їх структурованості.

**4.2. Еспериментальна оцінка**

Для проведення якісного дослідження, перш за все, необхідно знайти достовірні дані. Оглянувши безліч веб-сайтів, було обрано 50 з них, котрі максимально схожі на розроблений сайт. За допомогою он-лайн сервіса http://prlog.ru/ були виміряні такі швидкісні характеристики як:

* час завантаження сторінки;
* найбільший ресурс;
* найменший ресурс;
* найшвидший ресурс;
* найповільніший ресурс;
* загальний час усіх ресурсів.

Інструментами для збору даних стали також додатки для браузеру Mozilla Firefox – Yslow та Page Speed. З їх допомогою було виміряно:

* розмір сторінки;
* кількість HTTP запитів;
* загальну кількість ресурсів;
* загальний розмір усіх ресурсів.

Для визначення коефіцієнта мобільності, тобто значення, що у відсотках показує наскільки сайт оптимізований під пристрої, було використано он-лайн сервіс **W3C OK Checker.**

**Всі дані було занесено в таблицю для подальшого дослідження**(див. Додаток А).

**Аналіз статистичних характеристик.**

Представлені дані відносяться до інтервальної шкали, що сприяє розрахунку статистичних характеристик, оскільки, в основному вони розраховуються для змінних, що відносяться до інтервальної шкали.

В результаті була отримана статистична таблиця, частина якої наведена нижче (табл.4.1), яка містить наступні змінні:

* середнє значення – це арифметичне середнє виміряних значень; воно визначається як сума значень, поділена на їх кількість;
* стандартна помилка середнього значення. В інтервалі шириною, рівній подвоєній стандартній помилці, відкладеному навколо середнього значення, розташовується середнє значення генеральної сукупності з імовірністю приблизно 67%. Стандартна помилка визначається як стандартне відхилення, поділене на квадратний корінь з обсягу вибірки;
* медіана – це точка на шкалі виміряних значень, вище і нижче якої лежить по половині всіх виміряних значень;

Таблиця 4.1.

Статистична таблиця оцінки сайту

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Час завантаження (мс)** | **Кількість HTTP запитів** | **Розмір сторінки (кб)** | **Кількість ресурсів** | **Загальний розмір ресурсів (кб)** |
| Середнє значення | 2804,00 | 25,68 | 207,42 | 27,28 | 284,94 |
| Стд. помилка середнього | 146,172 | 2,921 | 29,021 | 3,096 | 39,335 |
| Медіана | 2654,50 | 20,00 | 142,00 | 23,00 | 182,00 |
| Мода | 2656 | 8 | 40 | 23 | 377 |
| Стд. відхилення | 1033,591 | 20,651 | 205,211 | 21,892 | 278,137 |
| Дисперсія | 1068309,714 | 426,467 | 42111,473 | 479,267 | 77360,302 |
| Асиметрія | 0,986 | 2,767 | 1,598 | 2,579 | 1,667 |
| Стд. помилка асиметрії | 0,337 | 0,337 | 0,337 | 0,337 | 0,337 |
| Ексцес | 1,159 | 10,734 | 2,135 | 8,723 | 2,182 |
| Стд. помилка ексцесу | 0,662 | 0,662 | 0,662 | 0,662 | 0,662 |
| Розмах | 4730 | 120 | 862 | 121 | 1090 |
| Мінімум | 1271 | 6 | 10 | 6 | 11 |
| Максимум | 6001 | 126 | 872 | 127 | 1101 |
| Сума | 140200 | 1284 | 10371 | 1364 | 14247 |

* мода – це значення, яке найбільш часто зустрічається у вибірці. Якщо одна і та ж найбільша частота зустрічається у декількох значень, то вибирається найменше з них;
* стандартне відхилення – це міра розкидання виміряних величин; воно дорівнює квадратному кореню з дисперсії. В інтервалі шириною, рівній подвоєному стандартному відхиленню, який відкладений по обидві сторони від середнього значення, розташовується приблизно 67% всіх значень вибірки, що підкоряється нормальному розподілу;
* дисперсія – це квадрат стандартного відхилення і, отже, ця характеристика також є мірою розкидання виміряних величин. Вона визначається як сума квадратів відхилень всіх виміряних значень від їх середньоарифметичного значення, поділена на кількість вимірювань мінус один;
* коефіцієнт асиметрії – це міра відхилення розподілу частоти від симетричного розподілу, тобто такого, у якого на однаковому віддалені від середнього значення по обидві сторони вибірки даних розташовується однакова кількість значень. Якщо спостереження підкоряються нормальному розподілу, то асиметрія дорівнює нулю;
* коефіцієнт варіації або ексцес вказує, чи є розподіл пологим (при великому значенні коефіцієнта) або крутим. Коефіцієнт варіації дорівнює нулю, якщо спостереження підкоряються нормальному розподілу;
* розмах – це різниця між найбільшим значенням (максимумом) і найменшим значенням (мінімумом);
* мінімум – найменше значення;
* максимум – найбільше значення;
* сума – сума всіх значень.

Проаналізувавши отриману таблицю, можна зробити висновок, що гіпотезу про те, що дані взяті з нормально розподіленої генеральної сукупності, слід відкинути, оскільки значення коефіцієнта асиметрії та коефіцієнта варіації значно відрізняються від нуля. На рисунках наведених нижче, наочно продемонстровано відсутність підпорядкуванню нормальному розподілу.

Нормальний

розподіл

Середнє

значення =2804

Стандартне

відхилення

=1033,591

N =50

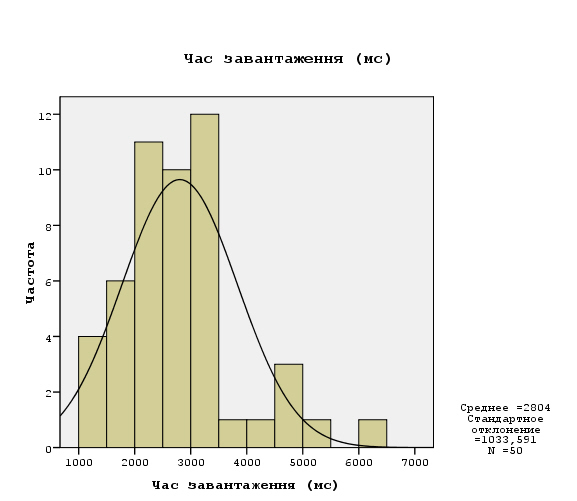


Рис. 4.1. Час завантаження

Нормальний

розподіл

Середнє

значення =25,68

Стандартне

відхилення

=20,651

N =50

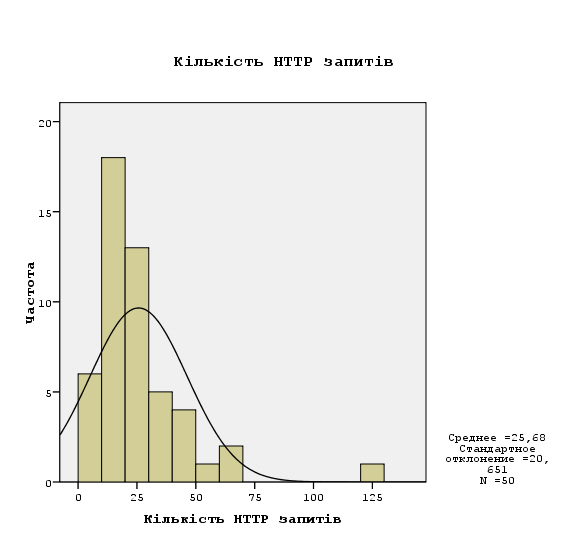


Рис. 4.2. Кількість HTTP запитів

Нормальний

розподіл

Середнє

значення =207,42

Стандартне

відхилення

=205,211

N =50

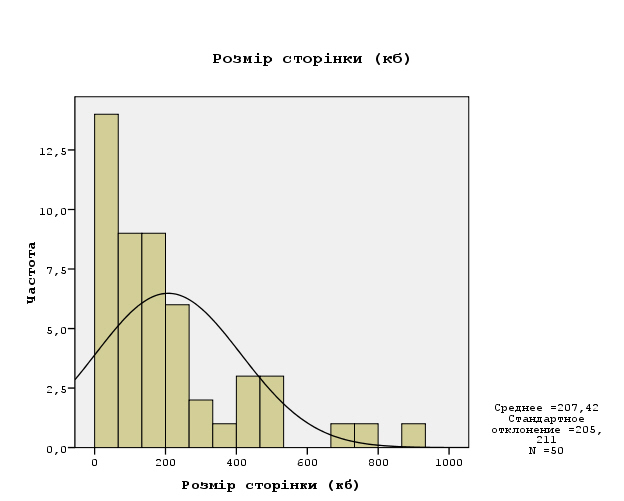


Рис. 4.3. Розмір сторінки

Нормальний

розподіл

Середнє

значення =284,94

Стандартне

відхилення

=278,137

N =50

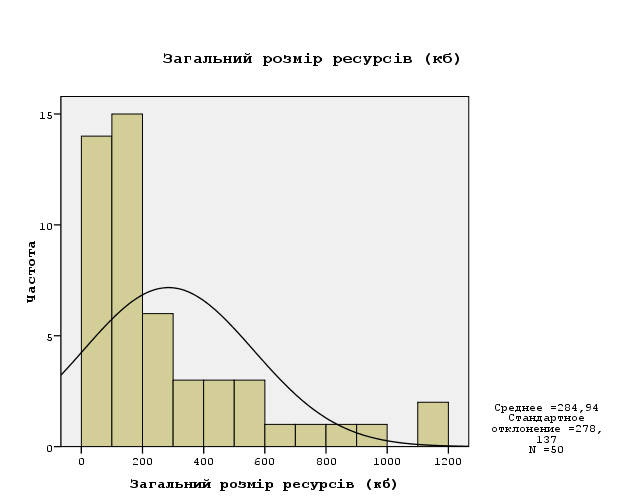


Рис. 4.4. Загальний розмір ресурсів

Нормальний

розподіл

Середнє

значення =42,46

Стандартне

відхилення

=29,747

N =50

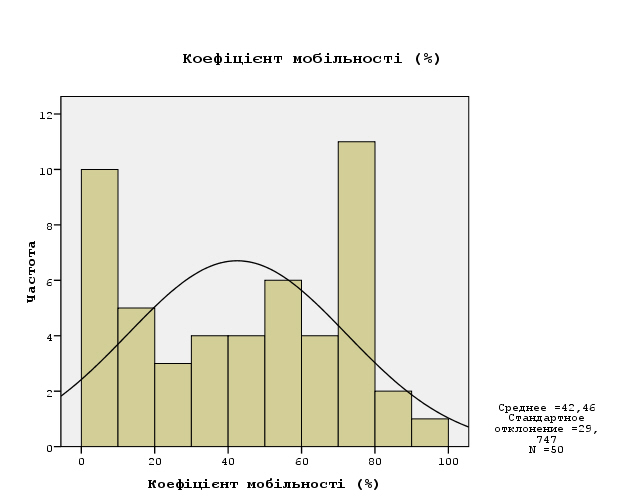


Рис. 4.5. Коефіцієнт мобільності

Отже, для зібраних даних, котрі відносяться до інтервальної шкали і не підпорядковуються нормальному розподілу, в якості основної характеристики використовується медіана або перший та третій квартилі.

Після одержання статистичних результатів, можна виконати порівняння найважливіших характеристик розробленого веб-сайту та досліджуваних аналогів (рис.4.6).

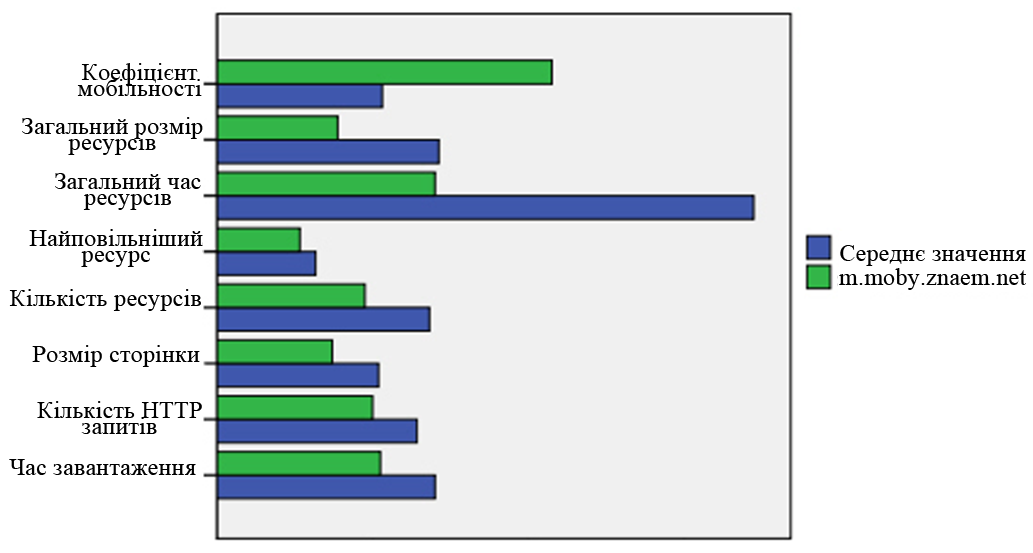


Рис. 4.6. Порівняння характеристик

Вищенаведений рисунок демонструє, що розроблений сайт переважає по всім характеристикам середньостатичний аналог. Найбільша розбіжність по коефіцієнту мобільності та загальному часі ресурсів, де перевага збільшується до 2 разів.

**Множинно-регресійний аналіз.**

Множинно-регресійний аналіз призначений для вивчення взаємозв'язку однієї змінної (залежної) і декількох інших змінних (незалежних) в інтересах передбачення деякого результату або суттєвості впливу тієї чи іншої змінної на передбачуваний результат.

Зв'язок однієї змінної (залежної) y та кількох інших змінних (незалежних) xn висловлюють лінійним рівнянням:

y = b0 + b1 x1 + b2 x2 + ... + bn xn + e,

де, y – залежна змінна;

x1, 2 ... n – незалежні змінні;

b1, 2 ... n – параметри моделі;

e – помилка передбачення.

Змінні, оголошені незалежними, можуть корелювати між собою. Цей факт необхідно обов'язково враховувати при визначенні коефіцієнтів рівняння регресії для того, щоб уникнути помилкових кореляцій.

Для множинного аналізу з декількома незалежними змінними не рекомендується використовувати метод включення всіх змінних. Цей метод відповідний одночасній обробці всіх незалежних змінних, обраних для аналізу, і тому він корисний при використанні тільки в разі простого аналізу з однією незалежною змінною. Для множинного аналізу слід вибирати один з покрокових методів.

При прямому методі незалежні змінні, які мають найбільші коефіцієнти часткової кореляції з залежною змінною покроково додаються в регресійне рівняння. При зворотному методі все починається з результату, що містить всі незалежні змінні і потім виключаються незалежні змінні з найменшими частковими кореляційними коефіцієнтами, поки відповідний регресійний коефіцієнт не виявляється незначущим.

Для проведення множинно-регресійного аналізу, в якості залежної змінної обрано – час завантаження сторінки, а незалежними будуть виступати всі змінні, що залишились.

В результаті проведення аналізу було отримано зведену таблицю моделі (табл.4.2), з якої слідує, що додавання змінних в розрахунок виконується за 3 кроки, тобто змінні найповільніший ресурс, загальний розмір ресурсів та загальний час ресурсів почергово додавалися у рівняння регресії.

Таблиця 4.2.

Зведена таблиця моделі

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Модель** | **R** | **R квадрат** | **Скоректований  R квадрат** | **Стд. помилка оцінки** |
| 1 | 0,722(a) | 0,521 | 0,511 | 722,438 |
| 2 | 0,843(b) | 0,710 | 0,698 | 568,028 |
| 3 | 0,864(c) | 0,747 | 0,731 | 536,523 |

a – предиктори: (константа) Найповільніший ресурс(мс);

b – предиктори: (константа) Найповільніший ресурс (мс), Загальний розмір ресурсів (кб);

c – предиктори: (константа) Найповільніший ресурс (мс), Загальний розмір ресурсів (кб), Загальний час ресурсів (мс).

Проаналізувавши таблицю покрокового додавання коефіцієнтів регресії (табл.4.3) було зроблено висновок, що змінна загальний час ресурсів може бути виключена з рівняння регресії, оскільки її коефіцієнт значимості p = 0,13 > 0,01, тобто він майже не впливає на рівняння.

Таблиця 4.3.

Покрокове додавання коефіцієнтів

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Модель | | Нестандартизовані  коефіцієнти | | Стандарт.  коефіцієнти | t | Знч. |
| B | Стд. помилка | Бета |
| 1 | (Константа) | 1665,807 | 187,642 |  | 8,878 | 0,000 |
|  | Найповільніший ресурс(мс) | 0,898 | 0,124 | 0,722 | 7,232 | 0,000 |
| 2 | (Константа) | 1214,486 | 168,565 |  | 7,205 | 0,000 |
|  | Найповільніший ресурс(мс) | 0,891 | 0,098 | 0,716 | 9,125 | 0,000 |
|  | Загальний розмір ресурсів (кб) | 1,615 | 0,292 | 0,435 | 5,536 | 0,000 |
| 3 | (Константа) | 1177,077 | 159,872 |  | 7,363 | 0,000 |
|  | Найповільніший ресурс(мс) | 0,824 | 0,096 | 0,663 | 8,614 | 0,000 |
|  | Загальний розмір ресурсів (кб) | 1,260 | 0,308 | 0,339 | 4,094 | 0,000 |
|  | Загальний час ресурсів(мс) | 0,016 | 0,006 | 0,221 | 2,585 | 0,013 |

Коефіцієнти β (Бета) – це регресійні коефіцієнти, стандартизовані відповідній області знань. Вони вказують на важливість незалежних змінних, що включені в регресійне рівняння.

Отже, рівняння регресії для прогнозування значення часу завантаження виглядає наступним чином:

**(Час завантаження сторінки)=1177,077 + 0,824·(Найповільніший ресурс) + +1,26·(Загальний розмір ресурсів)**

Важливим моментом є аналіз залишків, тобто відхилень спостережуваних значень від теоретично очікуваних (рис. 4.7).

**Регресія Стандартизований залишок**

Нормальний

розподіл

Середнє

значення =-6,59Е-17

Стандартне

відхилення

=0,969

N =50

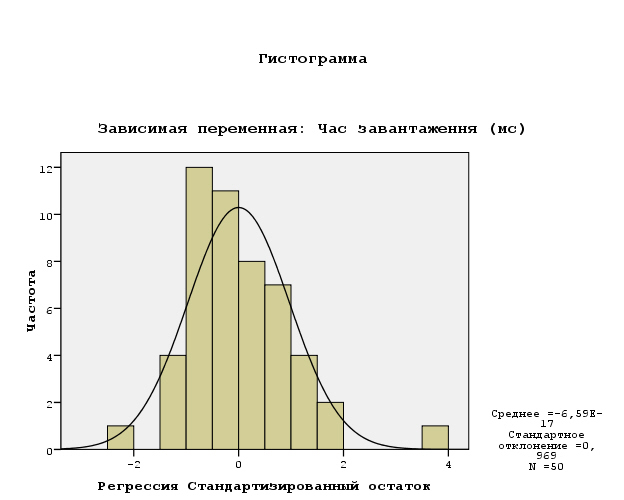


Рис. 4.7. Графік залишків

На вищенаведеному рисунку спостерігається доволі непогана узгодженість залишків з нормальним розподілом, що є позитивним результатом.

**Кластерний аналіз.**

В результаті кластерного аналізу, за допомогою попередньо заданих змінних формуються групи спостережень. Під спостереженнями розуміються окремі характеристики або будь-які інші об’єкти. Члени однієї групи (одного кластера) повинні володіти схожими проявами змінних, а члени різних груп – різними.

В кластерному аналізі важливу ролі відіграє ієрархічний метод. В ньому кожне спостереження створює спочатку свій окремий кластер. На першому кроці два сусідні кластери об’єднуються в один. Цей процес може продовжуватись до тих пір, поки не залишаться тільки 2 кластери.

Після проведення кластерного аналізу, перш за все, наводиться огляд приналежності, з котрого можливо вияснити почерговість побудови кластерів, а також їх оптимальну кількість (табл.4.4).

Таблиця 4.4.

Кроки агломерації

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Етап** | **Кластер об’єднаний з** | | **Коеф.** | **Етап** | **Кластер об’єднаний з** | | **Коеф.** |
| **Кластер 1** | **Кластер 2** | **Кластер 1** | **Кластер 2** |
| 1 | 13 | 19 | 0,000 | 26 | 6 | 15 | 0,090 |
| 2 | 14 | 42 | 0,002 | 27 | 3 | 4 | 0,115 |
| 3 | 12 | 17 | 0,004 | 28 | 25 | 26 | 0,127 |
| 4 | 26 | 37 | 0,007 | 29 | 3 | 8 | 0,127 |
| 5 | 3 | 11 | 0,009 | 30 | 1 | 24 | 0,135 |
| 6 | 7 | 45 | 0,011 | 31 | 28 | 34 | 0,174 |
| 7 | 25 | 41 | 0,013 | 32 | 6 | 18 | 0,182 |
| 8 | 33 | 50 | 0,015 | 33 | 3 | 7 | 0,251 |
| 9 | 18 | 44 | 0,017 | 34 | 13 | 25 | 0,281 |
| 10 | 3 | 12 | 0,018 | 35 | 2 | 28 | 0,293 |
| 11 | 15 | 40 | 0,018 | 36 | 9 | 36 | 0,312 |
| 12 | 8 | 32 | 0,019 | 37 | 1 | 30 | 0,375 |
| 13 | 26 | 29 | 0,026 | 38 | 13 | 20 | 0,399 |
| 14 | 7 | 16 | 0,029 | 39 | 9 | 21 | 0,446 |
| 15 | 4 | 23 | 0,031 | 40 | 1 | 6 | 0,642 |
| 16 | 2 | 22 | 0,033 | 41 | 2 | 49 | 0,694 |
| 17 | 6 | 46 | 0,036 | 42 | 5 | 27 | 0,706 |
| 18 | 20 | 31 | 0,048 | 43 | 3 | 13 | 0,757 |
| 19 | 7 | 33 | 0,048 | 44 | 9 | 47 | ***0,850*** |
| 20 | 21 | 39 | 0,051 | 45 | 1 | 3 | ***1,599*** |
| 21 | 8 | 10 | 0,051 | 46 | 5 | 9 | 3,578 |
| 22 | 1 | 43 | 0,053 | 47 | 1 | 5 | 5,897 |
| 23 | 25 | 35 | 0,061 | 48 | 1 | 2 | 6,353 |
| 24 | 14 | 48 | 0,073 | 49 | 1 | 38 | 18,12 |
| 25 | 9 | 14 | 0,083 |  |  |  |  |

Для визначення оптимальної кількості кластерів вирішальне значення має показник, виведений під заголовком "Коефіцієнт". За цим коефіцієнтом мається на увазі відстань між двома кластерами, визначена на підставі обраної дистанційної міри з урахуванням передбаченого перетворення значень. У даному випадку це квадрат евклідової відстані, визначений з використанням стандартизованих значень. На етапі, де міра відстані між двома кластерами збільшується стрибкоподібно, процес об'єднання в нові кластери необхідно зупинити, тому що в іншому випадку були б об'єднані кластери, що знаходяться на відносно великій відстані один від одного.

У даному випадку – це стрибок з 0,850 до 1,599. Оптимальним вважається число кластерів рівне різниці кількості спостережень (50) і кількості кроків, після якого коефіцієнт збільшується стрибкоподібно (44). Це означає, що після утворення 6 кластерів не потрібно більше робити ніяких наступних об'єднань, а результат з 6 кластерами є оптимальним.

Результати належності сайтів до певних кластерів наводяться в табл.4.5, з якої видно, що розроблений веб-сайт потрапив до кластера номер 3. Всі учасники 3 кластера відмічені в таблиці з початковими даними (див. Додаток А).

На рис.4.8 зображено кількість учасників кожного з кластерів. З нього слідує, що найбільшим виявився кластер під номером 3. Найменшими стали 4 та 6 кластери, які містять по 2 і 1 учасника відповідно.

Таблиця 4.5.

Належність до кластерів

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Спостереження** | | **№**  **класт.** | **Спостереження** | | **№ класт.** |
| 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | sports.ru | 1 | 26 | novostimira.com | 3 |
| 2 | sportbox.ru | 2 | 27 | mob.telegraf.coua | 4 |
| 3 | mobisports.ru | 3 | 28 | wapn.ukrinform.ua | 2 |
| 4 | liveresult.ru | 3 | 29 | focus.ua | 3 |
| 5 | eurosport.ru | 4 | 30 | pda.interfax.com.ua | 1 |
| 6 | football.ua | 1 | 31 | expres.ua | 3 |
| 7 | 0332.ua | 3 | 32 | pda.utro.ru | 3 |
| 8 | ua-football.com | 3 | 33 | gazeta.ua/pda | 3 |

Продовження таблиці 4.5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 |
| 9 | segodnya.ua | 5 | 34 | pda.mk.ru | 2 |
| 10 | kp.ua | 3 | 35 | pda.lenta.ru | 3 |
| 11 | pda.korrespondent.net | 3 | 36 | rbc.ru | 5 |
| 12 | pda.bagnet.org | 3 | 37 | pda.itar-tass.com | 3 |
| 13 | pda.i.ua | 3 | 38 | maximonline.ru | 6 |
| 14 | tochka.net | 5 | 39 | gazeta.ru | 5 |
| 15 | liga.net | 1 | 40 | ixbt.com | 1 |
| 16 | for-ua.com | 3 | 41 | habrahabr.ru | 3 |
| 17 | myscore.ru | 3 | 42 | europaplus.ru | 5 |
| 18 | shakhtar.com | 1 | 43 | rg.ru | 1 |
| 19 | footboom.com | 3 | 44 | tophotels.ru | 1 |
| 20 | metalist.ua | 3 | 45 | rst.ua | 3 |
| 21 | pobeda.od.ua | 5 | 46 | mapia.ua | 1 |
| 22 | 24boxing.com.ua | 2 | 47 | rozetka.ua | 5 |
| 23 | ua-gol.com/ news\_mobile | 3 | 48 | trial-sport.ru | 5 |
| 24 | mail.ru | 1 | 49 | notebook-center.ru | 2 |
| 25 | pda.pravda.com.ua | 3 | 50 | moby.znaem.net | 3 |

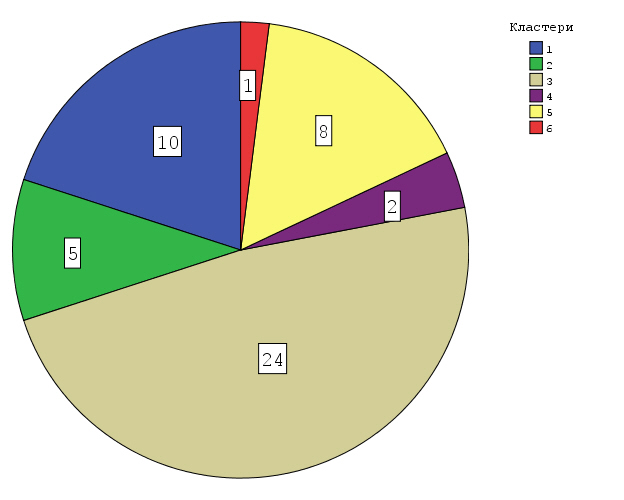


Рис. 4.8. Розмірність кластерів

Розділивши 50 спостережень на 6 груп, необхідно їх порівняти між собою, щоб встановити особливості кожного з кластерів. Для порівняння обрані такі характеристики як: час завантаження сторінки, кількість HTTP запитів, розмір сторінки, кількість ресурсів, найповільніший ресурс, загальний час ресурсів, загальний розмір ресурсів та коефіцієнт мобільності (рис.4.9).

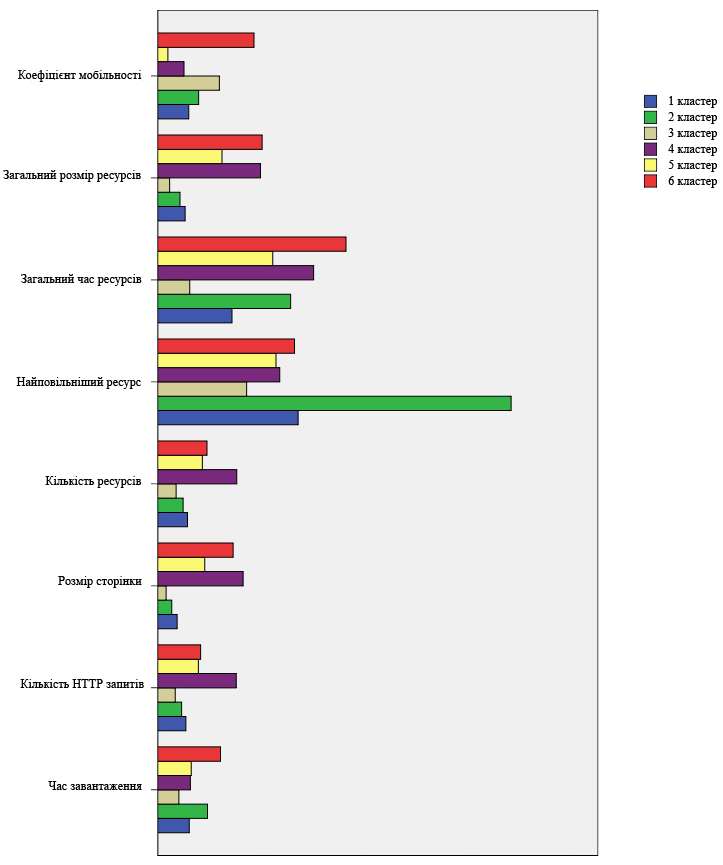


Рис. 4.9. Порівняння кластерів

Проаналізувавши отримані результати, можна зробити висновок, що 1та 5 кластери по всіх параметрах показують середні результати, що в загальному є дуже непогано. Кластер під номером 2 вирізняється найгіршим результатом у спостереженні найповільніший ресурс, та одним з найгірших у – час завантаження та загальний час ресурсів. Кластер, куди потрапив розроблений сайт, тобто 3, показує найкращі результати у всіх спостереженнях, окрім коефіцієнта мобільності, але це пояснюється тим, що кращий у даній категорії 6 кластер містить лише 1 сайт. Кластери під номерами 4 і 6 загалом показують одні з найгірших результатів по всіх параметрам, що є не дуже добре.

Визначивши, що 3 кластер є найкращим, необхідно прослідкувати, наскільки добре туди вписується розроблений веб-сайт (рис. 4.10).

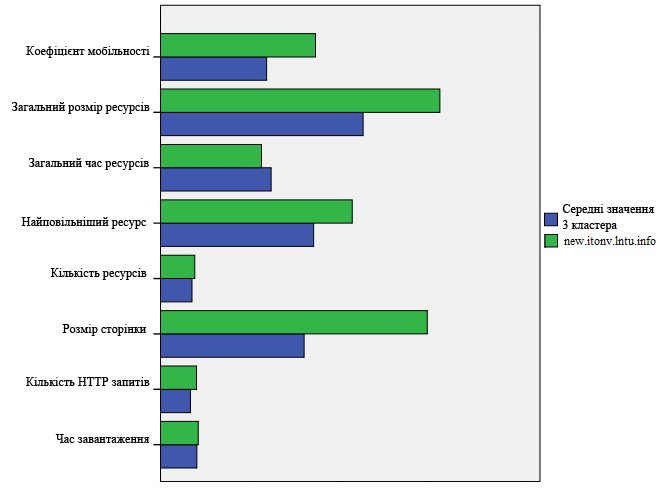


Рис. 4.10. Порівняння характеристик 3 кластера

З вищенаведеного ринку видно, що розроблений веб-сайт в більшості випадків має ідентичні показники з середньостатистичний сайтом 3 кластера. Винятками є два спостереження – загальний розмір ресурсів та розмір сторінки, де він показує дещо гірший результат. Цей факт можна пояснити тим, що в даний кластер потрапила велика кількість сайтів з примітивним дизайном та повною відсутністю зображень на сторінках, що є повною протилежністю до розробленого сайту. Це, звичайно, дає шалену швидкодію, але тим самий наносить збитки привабливості, що зменшує відвідуваність. У спостереженні – загальний час ресурсів, у розробленого сайта є невелика перевага, а коефіцієнт мобільності значно перевищує середньостатистичний.

Отже, після проведення всіх досліджень, можна зробити висновок, що система керування вмістом повністю підходить для створення веб-сайтів. При розумному підході, з її допомогою можна отримати багатофункціональну, яскраву, а головне – швидку версію сайту.

**Висновки до четвертого розділу**

Якість програмного комплексу може бути підвищено. Адміністратор може вибрати принципи керівництва, визначивши, що є основною метою – своєчасна видача результату, ефективне використання ресурсів або надійне обслуговування.

Проведений аналіз статистичних характеристик досліджуваних сайтів. В результаті було встановлено, що зібрані дані не підпорядковуються нормальному розподілу, оскільки значення коефіцієнта асиметрії та коефіцієнта варіації значно відрізняються від нуля.

**РОЗДІЛ 5**

**ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ**

Виконання роботи на протязі тривалого часу завжди супроводжується зниженням працездатності людини та стомленням організму. На це впливає розумова та фізична праця, а також середовище, в якому виконуються робочі операції.

Під час розробки нових освітніх технологій варто приділяти окрему увагу вивченню та вирішенню проблем, пов’язаних із забезпеченням безпечних та комфортних умов праці. Виявлення можливих причин нещасних випадків і розробка заходів, спрямованих на їх усунення – один із ключових факторів, який може забезпечити безпеку праці, зберегти життя та здоров’я працівника.

Саме з метою виявлення проблем та подальшого поліпшення умов праці проводиться аналіз стану охорони праці в навчальній лабораторії.

У даній дипломній роботі було розроблено інформаційну систему підтримки конференцій Луцького НТУ. Створення та тестування системи відбувалось в лабораторії №212А, яка знаходиться на другому поверсі 2-ох поверхової будівлі університету і розрахована на проведення практичних занять із використанням комп’ютерної техніки та проектора.

Ця аудиторія була обрана для виконання роботи тому, що вона безпосередньо підпорядкована кафедрі, оснащена комп’ютерами, на відміну від лекційних кабінетів. Більше того, є змога постійно консультуватись з керівником при виникненні спірних питань, а це є важливим, оскільки розроблювана система буде використовуватись викладачами університету і повинна бути зручна та функціональна.

Загальна площа приміщення становить 40 квадратних метрів, висота –2,8 м, приміщення має два вікна. Кількість робочих місць у лабораторії – 12. Отже, на одного студента в приміщенні припадає 6,6 квадратних метрів робочої площі.

Згідно з п. 2.3. ДСанПіН 3.3.2.007-98 розмір площі для одного робочого місця оператора персонального комп’ютера в аудиторії має складати не менше 6 кв. м. Отже, розмір площі для робочого місця студента в кабінеті відповідає допустимим нормам.

У приміщенні розташовано 12 комп`ютерів. Напруга джерела живлення комп’ютерів у приміщенні – 220В. Щоденно проводиться вологе прибирання.

Провітрювання робочих місць відбувається звичайним метало-пластиковим вікном, яке має два ступеня регулювання: легке провітрювання (вікно відхиляється зверху на 15º) та повне провітрювання (вікно повністю відкривається).

Покриття поверхні стола є матовим з коефіцієнтом підбиття 30%, легко чиститься, кути і передня панель дошки стола заокруглені. Сидіння комфортне, має заокруглені краї, нахиляється по відношенню до горизонталі вперед на 20° і назад на 140°, розмір його – 350x350мм. Висота спинки крісла складає 400–450 мм від поверхні сидіння.

Покриття підлоги є матовим. Коефіцієнт відбиття знаходиться у межах 0,3-0,5. Для внутрішнього оздоблення приміщення використовувались дифузно-відбивні матеріали з коефіцієнтом відбиття від стелі 0,6-0,8, що відповідає ДСанПіН 3.3.2-007-98.

Кабінет не межує з приміщеннями, в яких рівні вібрації або шуму перевищують допустимі граничні значення СН 3233-85, ДСН 3.3.6.039-99 («Санітарні норми виробничого шуму, ультразвуку та інфразвуку»).

Приміщення має природне та штучне освітлення. Природне освітлення здійснюється через світлові прорізи, орієнтовані на північний схід, які забезпечують коефіцієнт природної освітленості (КПО) 3,5%. Для штучного освітлення застосовуються люмінесцентні лампи типу ЛБ, які розміщені на відстані 2,8 метра від підлоги. Робочі місця з ПК розташовані так, що природне світло падає зліва.

У кабінеті є інструкція, яка містить перелік правил та заходів безпеки під час перебування в кабінеті та використання даного комп’ютерного обладнання. На стінах знаходиться план евакуації студентів та інструкція пожежної безпеки, затверджена в 2009 р. Засобом пожежогасіння в даній лабораторії виступає вогнегасник, на якому присутня пломба та який знаходиться біля вхідних дверей.

Перед початком роботи студенти проходять обов’язків інструктаж з техніки безпеки та ставлять підпис, чим підтверджують, що з правилами вони ознайомились та будуть дотримуватись їх. Інструктаж проводиться на початку навчального року завідувачем лабораторіями або викладачем, який відповідальний за аудиторію.

Цілком зрозуміло, що належний рівень безпеки праці можна забезпечити, перш за все, за умови безпечності комп'ютерної техніки. Для того, щоб запобігти виникненню небезпеки при раптовому вимкненні джерел енергії всі пристрої були оснащені спеціальними захисними пристосуваннями.

Рекомендується в подальшому використовувати спеціальні   пристрої безперервного живлення (USР), які можуть забезпечити постійну напругу та надати можливість працювати з комп'ютером при повному вимиканні живлення протягом декількох годин (в залежності від потужності пристрою). За цей час можна повністю завершити роботу на комп'ютері, щоб при його випадковому вимиканні не відбулося втрати інформації. Для серверів локальних мереж і комп'ютерів, що обробляють цінну інформацію, застосування приладів безперервного живлення є практично обов'язковим. Для інших комп'ютерів в якості дешевого замінника можна використати побутові стабілізатори.

Конструкція елементів комп'ютерної техніки повинна забезпечувати захист людини від ураження електричним струмом. Елементи комп'ютерної техніки в даній аудиторії оснащені засобами сигналізації при порушення нормального режиму роботи – засобами автоматичної зупинки та відімкнення від джерел енергії.

Кожне робоче місце обладнане системним блоком, монітором, маніпулятором типу “мишка” та 101-клавішною клавіатурою. Клавіатура виготовлена із матеріалу з високим коефіцієнтом тертя, розміщена під кутом 10°. Конструкція робочого місця користувача забезпечує підтримання оптимальної робочої пози. Конструкція елементів комп'ютерної техніки забезпечує захист людини від ураження електричним струмом. Технічні характеристики та параметри елементів комп'ютерної техніки відповідають антропометричним, фізіологічним, психофізіологічним та психологічним можливостям людини.

В аудиторії використовуються рідкокристалічні TFT монітори з діагоналлю 21.5 дюйма. На сьогоднішній день це найбільш безпечний варіант з точки зору гранично допустимого рівня випромінювання серед всіх можливих. В даних моніторах він не перевищує 1 мкВт/см2  і є допустимим згідно наказу №239 “Про затвердження державних санітарних правил та норм” затвердженого 01.08.96 (поточна редакція 30.10.2007).

Регламентований час роботи аудиторії складає 80 хв. Під час перерви аудиторія провітрюється.

До роботи в кабінеті допускаються лише студенти, які попередньо отримали допуск у викладача або завідуючого лаборанта, закріпленого за комп’ютерним класом. На першому занятті викладач повинен визначити робоче місце для кожного студента та ознайомити їх із правилами безпеки у лабораторії. Студенти повинні підтвердити своїм підписом про ознайомлення, погодження та обов'язкове дотримання всіх правил безпечної поведінки у комп’ютерному класі. Для непередбачуваних ситуацій у аудиторії міститься аптечка першої медичної допомоги.

Детальний аналіз лабораторії показав, що середня температура повітря там становить 22°С, відносна вологість — 54%, рухливість повітря 0.1 м/с. Згідно ГОСТ 12.1.005–88, СН 4088–86) всі ці гігієнічні параметри виробничого середовища знаходяться в нормі.

Найбільш повним офіційним нормативним документом щодо забезпечення охорони праці користувачів ПК є «Державні санітарні правила й норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанШН 3.3.2.007–98.

Організація робочого місця в кабінеті забезпечує відповідність конструкції всіх елементів робочого місця та їх взаємного розташування ергономічним вимогам з урахуванням характеру і особливостей трудової діяльності (ДСанПіН 3.3.2-007-98).

Електрична енергія в певних умовах легко переходить у теплову і це може викликати пожежі. Пожежна небезпека електрообладнання, електронних приладів, ЕОМ пов’язана з використанням займистих матеріалів: гуми, пластмас, лаків, олій.

Джерелами займання можуть бути електричні іскри, дуги, коротке замикання, струмові перевантаження, перегріті опірні поверхні, несправність обладнання. Але потужність і тривалість дії цих джерел займання порівняно малі, тому горіння, як правило, не розвивається. Виникнення пожежі в електронних пристроях можливо, якщо використовуються займисті і матеріали і вироби.

Профілактика коротких замикань передбачає наступні заходи:

1. правильний вибір, монтаж і експлуатація електричних мереж, електрообладнання;
2. правильний вибір конструкції електрообладнання, способу встановлення і класу ізоляції (опір ізоляції згідно з ПУЕ 500кОм);
3. електричний захист електричних мереж, електрообладнання (швидкодіючі реле, автоматичні вимикачі, запобіжники).

Протипожежний захист споруд, людей, які в них перебувають зокрема досягається застосуванням установок автоматичної пожежної сигналізації. Відповідно до ДСТУ 2273-93 “ССБП. Пожежна безпека. Терміни та визначення” під установкою пожежної сигналізації розуміється сукупність технічних засобів, установлених на об'єкті, що захищається, для виявлення пожежі, оброблення, подавання в заданому вигляді повідомлення про пожежу на цьому об'єкті, спеціальної інформації та подавання команд на включення автоматичних установок пожежогасіння та технічних обладнань.

Під час експлуатації комп’ютерної техніки приміщення повинні бути оснащені системою автоматичної пожежної сигналізації з димовими пожежними сповіщувачами та переносними вуглекислотними вогнегасниками з розрахунку 2 шт. на кожні 20 м2 площі.

Отже, проаналізувавши стан охорони праці в лабораторії №212А, було визначено, що вона знаходиться у належному стані, оскільки за тривалий період не зареєстровано жодного нещасного. Студенти та викладачі можуть бути впевнені в безпеці та комфортності роботи. Завідувач лабораторіями та викладачі, закріплені за комп’ютерними кабінетами, завжди проводять інструктаж з техніки безпеки для студентів. Під час інструктажу навчаючі дізнаються про правила поводження в аудиторії в літній та зимовий час та безпосередньо ознайомлюються з правилами експлуатації ЕОМ. Кожен викладач повідомляє про найбільш розповсюджені помилки під час роботи з комп’ютерами.

Хоча стан охорони праці в даній лабораторії в належному стані, проте є пропозиції, які варто розглянути та впровадити для збільшення рівня комфортності роботи та поліпшення рівня безпеки. В якості пропозицій варто розглянути впровадження новітньої системи кондиціонування приміщення та автоматичної системи пожежної сигналізації, встановлення допоміжних ламп на кожне робоче місце для додаткового освітлення в зимовий час. Якщо друга пропозиція є не обов’язковою і лише поліпшить роботу студентам під час занять в зимовий час, то впровадження систем вентиляції та пожежної сигналізації дійсно є важливим та допоможе уникнути замикання техніки.

**Висновки до п’ятого розділу**

В даному розділі було проаналізовано питання пов’язані із охороно праці та безпекою людей в надзвичайних ситуаціях в аудиторії № 212А. В якості пропозицій варто розглянути впровадження новітньої системи кондиціонування приміщення.

**ВИСНОВКИ**

У даній випускній роботі були дослідженні методології моделювання бізнес-процесів, принципи побудови потоків робіт та можливості їх опису використовуючи технологію Windows workflow foundation 4. Були створені моделі поведінки об’єктів, по яким розроблено Microsoft ASP.NET додаток, що надає можливість автоматизувати бізнес процеси, які відбуваються при організації конференції. Веб-додаток сприяє ефективнішій взаємодії з учасниками конференції, а також налагодженню роботи зі статтею, її рецензуванням та публікацією.

В процесі створення даного додатку вивчені принципи моделювання бізнес-процесів, а також використані механізми їх управління. Проаналізовані особливості підходу створення додатку на основі бізнес-процессів, а також в повній мірі використані можливості та переваги використання Windows workflow foundation 4.

У зв’язку з цим зроблено такі висновки: даний підхід добре підходить для вирішення завдань управління бізнес-процесами і документообігом, оскільки дає можливість наочно представити їх алгоритміку за допомогою діаграм та таблиць станів, що значно полегшує процес розуміння і створення структури робочих потоків, механізм управління діями. На мій погляд, такий підхід є достатньо цікавим, цілком може зайняти гідне місце у ряді інструментів, які застосовують сучасні розробники і стати серйозним підґрунтям в руках досвідчених користувачів.

Проведений аналіз статистичних характеристик досліджуваних сайтів. В результаті було встановлено, що зібрані дані не підпорядковуються нормальному розподілу, оскільки значення коефіцієнта асиметрії та коефіцієнта варіації значно відрізняються від нуля.

Проведений множинно-регресійний аналіз, в результаті якого було відсіяні характеристики, які майже не впливають на швидкість завантаження сторінки, а за допомогою важливих характеристик – складене рівняння регресії, що має вигляд:

(Час завантаження сторінки)=1177,077 + 0,824·(Найповільніший ресурс) + + 1,26·(Загальний розмір ресурсів).

Проведений кластерний аналіз. В результаті досліджувані сайти були розбиті на 6 кластерів. Критерями розбиття слугували розмір головної сторінки та швидкість завантаження. Розроблений сайт опинився в кластері під номером 3, що є найпродуктивнішим. Після порівняння характеристик сайтів 3 кластера, було встановлено, що у розробленому сайті чудово поєднується дизайн та швидкодія.

У ході виконання наукової роботи були виконані поставлені завдання і досягнута мета даної випускної роботи.

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. K. Scott Allen. Programming Windows Workflow Foundation: Practical WF Techniques and Examples using XAML and C# / K. Scott Allen. – Packt Publishing Ltd, 2006. – 252 с. – ISBN-10: 1904811213, ISBN-13: 978-1904811213
2. Джесс Либерти. Создание .NET-приложений. Программирование на С#/ Джесс Либерти. – М.: Издательство Символ-Плюс, 2000. – 620 с. – ІSBN 5-93286-038-3, 0-596-00309-9.
3. Бадд Т. Объектно-ориентированное программирование в действии / Тимоти Бадд. – М.: Издательский дом «Вильямс», 2002. – 816 с. – ISBN 5-88782-270-8, 0-201-82419-1.
4. Туан ТАЙ. Платформа .NET. Основы. / Туан ТАЙ, Хонг К. Лэм. – М.: Символ-Плюс, 2002. - 336 с. – ISBN 5-93286-054-5, 0-596-00302-1.
5. Microsoft Corporation. Разработка Web-приложений на Microsoft Visual Basic .NET и Microsoft Visual C# .NET. Учебный курс MCAD/MCSD / Microsoft Corporation; пер. с англ. А. Е. Соловченко; [гл. ред. А. И. Козлов, ред. Ю. П. Леонова]. – М.: Издателъско-торговый дом «Русская Редакция», 2003. – 704 с. – ISBN 5-7502-0221-6.
6. Е. Монахова. Управление потоком работ. / Монахова Е., Бочкарев А., Лукомский А., Майоров А. – М.: Наука, 1984, с.148-152.
7. Винкоп Стефан. Описание и совершенствование бизнес-процессов. Специальное издание. / Стефан Винкоп. – М.: Издательский дом «Вильямс», 1999. – 816 с. – ISBN 978-5-8459-1355-5.
8. Уильям Р.С. Управление бизнес-процессами и моделирование бизнес-процессов / Р.С. Уильям - М.: Весть-Метатехнология, 2000. – 175 с.
9. Рихтер Дж. CLR via C# Программирование на платформе Microsoft .NET Framework 2.0 на языке C#. Мастер-класс./ Дж. Рихтер, пер. с англ. – 2е изд., исправ. – М.: Издательство «Русская Редакция»; – СПб.: Питер, 2008. – 656 с. – ISBN 978-5-7502-0348-2 (“Русская Редакция”), ISBN 978-5-91180-303-2 («Питер»).
10. Asp.Net mania. [Електронний ресурс] / Центр информ. технологий РГБ; авт. Олег Шаститко. – Режим доступу: http://www.aspnetmania.com/Articles/ Article/77.html. – Дата доступу: листопад 2012 р.
11. Дино Эспозито. Знакомство с технологией Microsoft ASP.NET 2.0 AJAX/ Эспозито Дино; пер. с англ. Е. Матвеев. – М.: Издательство «Русская Редакция»; – СПб.: Питер, 2007. – 320 с. – ISBN 978-5-7502-0307-9 (“Русская Редакция”), ISBN 978-5-91180-435-0 («Питер»).
12. Майк Гандерлой. Освоение Microsoft SQL Server 2005. / Майк Гандерлой, Джозеф Джорден, Дейвид Чанц. – М.:Издательство «Вильямс», 2007. – 1104 с. – ISBN 978-5-8459-1208-4, 0-7821-4380-6.
13. Шеер А.-В. Бизнес-процессы. Основные понятия. Теория. Методы. / Август-Вильгельм Шеер; пер. с англ. Михайлова Н.А.; [редактор Каменнова М.С., Громов А.М.]. – М.: Весть-МетаТехнология, 1999. – 320 с. – ISBN 5-88-661-022-7.
14. Шеер А.-В. Моделирование бизнес-процессов. / Август-Вильгельм Шеер; пер. с англ. Михайлова Н.А.; [редактор Каменнова М.С., Громов А.М.]. – М.: Весть-МетаТехнология, 2000. – 232 с. – ISBN 5-88-661-329-5.
15. Каменнова М.С. Моделирование бизнеса. Методология ARIS / Каменнова М.С., Громов А.И., Ферапонтов М.М., Шматалюк А.Е. – М.: Весть-МетаТехнология, 2001. – 157 c.
16. Хаммер М. Реинжиниринг корпорации: Манифест революции в бизнесе / Хаммер М., Чампи Дж. – М.: Изд-во «Манн, Иванов и Фербер», 2006. – 253 c. – ISBN 978-5-902862-54-3.
17. Андерсен Б. Бизнес-процессы. Инструменты совершенствования / Б. Андерсен; пер. с англ. С. В. Ариничева; [науч. ред. Ю. П. Адлер]. – М.: РИА Стандарты и качество, 2003. – 367 c. – ISBN 978-5-94938-058-1,978-5-94938-065.
18. Спицнадель В.Н. Основы системного анализа / В.Н. Спицнадель – СПб.: Бизнес-пресса, 2000. – 297 c. – ISBN 5-8110-0025-1.
19. Робсон М. Реинжиниринг бизнес-процессов: Практическое руководство / Робсон М., Уллах Ф. – М.: ЮНИТИ-ДАНА, 2003. – 384 с.
20. Ян Ван Бон. ИТ Сервис-менеджмент / Ян Ван Бон, Пондман Д. – М.: Van Haren Publishing, 2003. – 212 с. – ISBN (10): 90-77212-94-9; ISBN (13): 978-90-77212-94-3;
21. Гома Х. UML: проектирование систем реального времени, параллельных и распределенных приложений / X. Гома; пер. с англ. – М.: ДМК, 2002. – 172 с. – ISBN 5-94074-101-0.
22. Вендров А.М. Практикум по проектированию программного обеспечения экономических информационных систем: Учебное пособие для вузов / Вендров А.М. – 2-е изд., перераб. и доп. – М.: Финансы и статистика, 2006. – 192с. – ISBN: 5-279-03106-2.
23. Леонтьев Б.К. Microsoft Visio 2002 Professional. Построение проектов, диаграмм и бизнес-схем в операционной системе Microsoft Windows XP / Леонтьев Б.К. – М.: СОЛОН-Пресс, 2002. – 183 с.
24. Боггс М. UML и Rational Rose 2002 / Боггс М., Боггс У.; пер. с англ. И. Афанасьев, И. Дранишников – М.: СОЛОН-Пресс, 2002. – 381 с. – ISBN 5-85582-091-2.
25. Маклаков С.В. BPwin и Erwin. Case-средства разработки информационных систем / С.В. Маклаков – М.: Диалог-МИФИ, 2001. – 174 с. – ISBN: 5-86404-128-9.
26. Калянов Г.Н. Консалтинг при автоматизации предприятий / Г.Н. Калянов– М.: СИНТЕГ, 1997. – 169 с.
27. Ойхман Е.Г. Реинжиниринг бизнеса: Реинжиниринг организаций и информационные технологии/ Ойхман Е.Г., Попов Э.В. – М.: Финансы и статистика, 1997. – 253 с.
28. С. Байдачный. UML. Новые возможности для разработчиков / Байдачный С., Лозинский Ю., Маленко Д. – СПб.: Питер, 2006. – 450 с.
29. Мак-Дональд Мэтью. Microsoft ASP.NET 2.0 с примерами на C# 2005 для профессионалов / Мэтью Мак-Дональд, Марио Шпушта. – М.: Вильямс, 2007. – 1408 с.
30. Сайты и системы на ASP.NET // IntroWebstudiodesign. – Режим доступу: <http://introweb.ru/sale/aspnet/>. – Дата доступу: серпень 2013. – Заголовок з екрану.
31. EasyChair Conference system [Електронний ресурс] // Режим доступу: [http://www.easychair.org](http://www.easychair.org/). – Дата доступу: серпень 2013 року. – Заголовок з екрану.
32. Peer-Review, Abstract and Conference Management // ОpenConf [Електронний ресурс]. – Режим доступу: <http://www.openconf.com/>. – Дата доступу: серпень 2013 року. – Заголовок з екрану.
33. Conference management system // Wikipedia. The Free Encyclopedia. [Електронний ресурс]. – Режим доступу: <http://en.wikipedia.org/wiki/Conference_management_system>. – Дата доступу: серпень 2013 року. – Заголовок з екрану.
34. Introducing Confious – The Conference Nous // Confious – Conference Management System with Intelligence, Power and Style [Електронний ресурс]. – Режим доступу: http://www.confious.com/. – Дата доступу: серпень 2013 року. – Заголовок з екрану.
35. Conference Management System (COMS) // COMS [Електронний ресурс]. – Режим доступу: <http://www.conference-service.com/conference-support/conference-management-system.html>. – Дата доступу: серпень 2013 року. – Заголовок з екрану.
36. Executable UML: A Foundation for Model Driven Architecture./ [Mellor S. et al.]; - МА: Addison-Wesley, 2002. – P. 258.
37. Буч Г. Язык UML. Руководство пользователя./ Буч Г., Рамбо Д., Джекобсон А. – М.: ДМК, 2000. – 320 с. – ISBN: 5-94074-334-X, 0-321-26797-4.
38. UML. SWITCH-Технология. Eclipse./ [Гуров В.С., Мазин М.А., Нарвский А.С., Шалыто А.А.] - 2005. №6(13). - C. 12-17.
39. Шалыто А.А. SWITCH-технология – автоматный подход к созданию программного обеспечения "реактивных" систем / Шалыто А.А., Туккель Н.И. - Программирование. 2001. №5. - с. 45-62.
40. Аалст Вил. Управление потоками работ: модели, методы и системы./ Аалст Вил, Хей Кейс. – М.: Физматлит, 2007. – 120 c. – ISBN 978-5-922107-62-4.
41. Standards // Workflow Management Coalition. [Електронний ресурс]. – Режим доступу: <http://www.wfmc.org/standards/standards.htm>. – Дата доступу: серпень 2013 року. – Заголовок з екрану.
42. Workflow Process Definition Interface [Електронний ресурс] /   
    2013. – Режим доступу: <http://www.wfmc.org/standards/docs/TC-1025_10_xpdl_102502.pdf>. – Дата доступу: березень 2013 року. – Заголовок з екрану.
43. Липаев В. Стандарты в области информационных технологий/ Pcweek. - 1998. - № 24.
44. Нестеренко А.К. Служба управления потоками работ по манипулированию ресурсами репозитория./ Нестеренко А.К., Бездушный А.А., Сысоев Т.М. и др.// – Електронні бібліотеки, 2003, Том 6, Випуск 5. – 460 c.
45. Теория расписаний и вычислительные машины / Под ред. Э.Г.Коффмана. –М.: Наука, 1984. - 336 с.
46. Гудман С. Введение в разработку и анализ алгоритмов / Гудман С., Хидетниеми С. – М.:Мир, 1981. –368 с.
47. Быков С.П. Разработка моделей в системе моделирования / Быков С.П., Храмов А.А. – Москва, МИФИ, 1998. – 110 c.
48. Слепцов А.И. Автоматизация проектирования управляющих систем гибких автоматизированных производств./ Слепцов А.И., Юрасов А.А. [Под ред. В.Н.Малиновского.] – К.: Технiка, 1986. - 110 с.
49. Бауэр В. Введение в теорию конечных автоматов / В.Бауэр –М.: Радио и связь, 1987. – 392 с.
50. Акулич И.Л. Математическое программирование в примерах и задачах./ И.Л. Акулич. – М.: Высшая школа, 1986. - 319 с.
51. Автоматизированное управление технологическими процессами: Учеб-ное пособие/ Под ред. Яковлева В.Б. –Л.: Изд-во Ленингр. Ун-та, 1988. – 224 с.
52. Управление потоком работ. Rondo integrale de universale. [Електронный ресурс] / [Монахова Е., Бочкарев А., Лукомский А., Майоров А.]. – Режим доступу : http://www.russianenterprisesolutions.com/mana/01/051.html. – Дата доступу: березень 2013 року. – Заголовок з екрану.
53. Welcome to BPMI.org [Електронний ресурс] / 2002. – Режим доступу : <http://www.bpmi.org/>.
54. Business Process Modeling Language (BPML) [Електронний ресурс] / 2003. – Режим доступу : <http://xml.coverpages.org/bpml.html>.
55. Robert Shapiro. A Comparison of XPDL, BPML and BPEL4WS./ - МА: Addison-Wesley, 2002. – P. 258.
56. Маклаков С. Инструментальные средства создания корпоративных информационных систем/ - Компьютер Пресс, №7-№9, 1998, - 37 с.
57. Using the XPath provided in XML validation error messages [Електронний ресурс] / М. : 2009 Microsoft Corporation. All rights reserved. – Режим доступа: <http://blogs.msdn.com/denisruc/archive/2005/07/27/> 443989.aspx.
58. Бездушный А.Н. Интегрированная система информационных ресурсов РАН и технология разработки цифровых библиотек. Программирование/ [Бездушный А.Н., Жижченко А.Б., Кулагин М.В., Серебряков В.А.] - V 26, N 4, 2000, - 177-185 с.
59. Марка Д.А. Методология структурного анализа и проектирования SADT (Structured Analysis & Design Technique)./ Марка Д.А., МакГоуэн К. – SofTecb, 1986. – 238 c.
60. XML Support in Microsoft SQL Server 2005 [Електронний ресурс] / 2009, Microsoft Corporation, Режим доступу: http://msdn.microsoft.com/en-us/library/ms345117.aspx. – Дата доступу: серпень 2013 року. – Заголовок з екрану.

61. Predictive analytics [Електронний ресурс] – Режим доступа : URL : http://en.wikipedia.org/wiki/Predictive\_analytics. – Дата доступу: травень 2013. – Загл. с экрана.

1. Брауде Э. Технологии разработки программного обеспечения / Эрик Брауде; перев. с англ. Е. Бочкарева, Д. Солнышкова; [главн. ред. Е. Строганова] – СПб.: Питер, 2004. – 655 с. – ISBN 5-94723-663-X.
2. Гаврилова Т.А. Базы знаний интеллектуальных систем / Т.А. Гаврилова, В.Ф. Хорошевский. – СПб.: Питер, 2001. – 384 с. – ISBN 5-272-00071-4.
3. Дейт, К. Дж. Введение в системы баз данных, 8-е издание / К. Дж Дейт; пер. с англ. К.А. Птицына; [зав. ред. С.Н. Трыгуб] – М.: Издательский дом "Вильямс", 2005. – 1328 с. – ISBN 5-8459-0788-8.
4. Дунаев С.Б. Технологии интернет-программирования / С.Б. Дунаев – СПб.: БХВ-Петербург, 2001. – 480 с. – ISBN 5-94157-086-4.
5. Жидецький В. Ц. Основи охорони праці: Підручник. / В.Ц. Жидецький - 4-те вид., перероб. і доп. - К.: Знання, 2010. - 375 с. + компакт-диск. – ISBN 978-966-346-601-9.
6. Запорожець О. І. Основи охорони праці. Підручник / О. І. Запорожець, О. С. Протоєрейський, Г. М. Франчук, І. М. Боровик – К.: Центр учбової літератури, 2009. – 264 с. – ISBN 978-966-364-934-4.

**ДОДАТКИ**

**Додаток А**

Таблиця з даними для проведення дослідження

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Адреса** | **Час завантаження (мс)** | **Кількість HTTP запитів** | **Розмір сторінки (кб)** | **Кількість ресурсів** | **Найшвидший ресурс (мс)** | **Найповільніший ресурс(мс)** | **Загальний час ресурсів(мс)** | **Найменший ресурс (байт)** | **Найбільший ресурс (байт)** | **Загальний розмір ресурсів (кб)** | **Коефіцієнт мобільності (%)** |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** |
| http://sports.ru | 3337 | 20 | 157 | 23 | 21 | 886 | 9628 | 32 | 43651 | 216 | 21 |
| http://sportbox.ru | 4940 | 29 | 168 | 34 | 4 | 3751 | 24327 | 25 | 93868 | 377 | 38 |
| http://mobisports.ru | 2364 | 26 | 46 | 26 | 123 | 776 | 11828 | 149 | 15143 | 55 | 55 |
| http://liveresult.ru | 2656 | 24 | 40 | 23 | 20 | 795 | 9470 | 32 | 63720 | 96 | 45 |
| http://eurosport.ru | 2783 | 24 | 760 | 24 | 17 | 934 | 5864 | 73 | 60946 | 863 | 50 |
| http://football.ua | 2945 | 28 | 227 | 36 | 90 | 1102 | 14390 | 1 | 58690 | 377 | 0 |
| http://0332.ua | 2293 | 22 | 139 | 23 | 93 | 773 | 5904 | 32 | 39604 | 167 | 77 |
| http://ua-football.com | 2393 | 15 | 101 | 23 | 90 | 968 | 6686 | 1 | 39614 | 151 | 71 |
| http://segodnya.ua | 3369 | 40 | 442 | 48 | 80 | 1082 | 20253 | 32 | 95492 | 563 | 20 |
| http://kp.ua | 2656 | 18 | 93 | 23 | 86 | 850 | 9508 | 32 | 85293 | 222 | 0 |
| http://pda.korrespondent.net | 2270 | 19 | 53 | 20 | 94 | 735 | 6049 | 35 | 39601 | 95 | 57 |
| http://pda.bagnet.org | 2204 | 10 | 28 | 11 | 85 | 1234 | 3960 | 74 | 42090 | 53 | 68 |
| http://pda.i.ua | 1271 | 8 | 10 | 7 | 306 | 498 | 3067 | 43 | 2547 | 11 | 78 |
| http://tochka.net | 3060 | 67 | 443 | 73 | 94 | 1034 | 24696 | 35 | 102973 | 692 | 0 |
| http://liga.net | 3113 | 29 | 215 | 24 | 88 | 1402 | 10992 | 35 | 42090 | 262 | 55 |
| http://for-ua.com | 2162 | 12 | 106 | 16 | 89 | 1084 | 11996 | 32 | 39601 | 154 | 19 |
| http://myscore.ru | 2237 | 8 | 40 | 9 | 92 | 1049 | 4082 | 35 | 66000 | 77 | 58 |
| http://shakhtar.com | 2609 | 32 | 272 | 34 | 96 | 1132 | 17549 | 35 | 47646 | 402 | 14 |
| http://footboom.com | 1272 | 6 | 14 | 6 | 119 | 680 | 2019 | 35 | 21098 | 34 | 70 |
| http://metalist.ua | 1511 | 39 | 182 | 39 | 241 | 896 | 3594 | 243 | 18815 | 198 | 73 |
| http://pobeda.od.ua | 3610 | 13 | 497 | 16 | 22 | 912 | 23368 | 35 | 76921 | 513 | 0 |
| http://24boxing.com.ua | 5037 | 11 | 136 | 11 | 246 | 3538 | 8133 | 1783 | 4148 | 142 | 52 |
| http://ua-gol.com/news\_mobile | 2526 | 8 | 15 | 7 | 127 | 1718 | 4345 | 43 | 3001 | 17 | 74 |
| http://mail.ru | 2987 | 47 | 98 | 51 | 4 | 1698 | 34372 | 32 | 26068 | 115 | 14 |

Продовження додатку А

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** |
| http://pda.pravda.com.ua | 1837 | 20 | 55 | 21 | 95 | 625 | 8056 | 38 | 57728 | 65 | 31 |
| http://novostimira.com | 1563 | 19 | 83 | 18 | 247 | 772 | 6098 | 178 | 8615 | 87 | 62 |
| http://mob.telegraf.com.ua | 3443 | 126 | 872 | 127 | 83 | 1398 | 53667 | 32 | 140360 | 1101 | 0 |
| http://wapn.ukrinform.ua | 4935 | 12 | 54 | 17 | 5 | 4805 | 5603 | 121 | 77746 | 183 | 43 |
| http://focus.ua | 1494 | 10 | 64 | 10 | 89 | 809 | 4927 | 35 | 39601 | 78 | 69 |
| http://pda.interfax.com.ua | 3437 | 9 | 30 | 9 | 93 | 1822 | 4606 | 35 | 39601 | 59 | 75 |
| http://expres.ua | 1384 | 31 | 145 | 32 | 86 | 567 | 4009 | 35 | 39601 | 170 | 74 |
| http://pda.utro.ru | 2474 | 16 | 78 | 20 | 88 | 778 | 10424 | 32 | 42090 | 103 | 0 |
| http://gazeta.ua/pda | 2201 | 16 | 162 | 15 | 92 | 834 | 7971 | 35 | 39601 | 195 | 75 |
| http://pda.mk.ru | 4520 | 29 | 77 | 28 | 95 | 2296 | 16743 | 32 | 39601 | 126 | 44 |
| http://pda.lenta.ru | 1993 | 10 | 33 | 10 | 18 | 666 | 3082 | 35 | 39601 | 70 | 81 |
| http://rbc.ru | 3300 | 66 | 321 | 87 | 100 | 904 | 39626 | 35 | 39601 | 350 | 0 |
| http://pda.itar-tass.com | 1609 | 13 | 97 | 11 | 86 | 725 | 5729 | 35 | 42090 | 152 | 77 |
| http://maximonline.ru | 6001 | 41 | 720 | 47 | 89 | 1307 | 36167 | 2 | 68721 | 998 | 92 |
| http://gazeta.ru | 3447 | 59 | 530 | 50 | 8 | 886 | 14722 | 32 | 42090 | 1100 | 21 |
| http://ixbt.com | 3158 | 25 | 241 | 27 | 95 | 1056 | 7476 | 58 | 95189 | 440 | 44 |
| http://habrahabr.ru | 1725 | 7 | 63 | 8 | 90 | 889 | 2738 | 35 | 39601 | 99 | 74 |
| http://europaplus.ru | 3107 | 31 | 439 | 29 | 86 | 1382 | 30976 | 35 | 48990 | 517 | 0 |
| http://rg.ru | 3104 | 49 | 148 | 50 | 93 | 1611 | 33759 | 32 | 39601 | 181 | 0 |
| http://tophotels.ru | 2653 | 11 | 247 | 12 | 2 | 1105 | 5079 | 32 | 42090 | 268 | 13 |
| http://rst.ua | 2282 | 23 | 118 | 23 | 91 | 613 | 6230 | 32 | 57741 | 194 | 39 |
| http://mapia.ua | 2759 | 19 | 215 | 18 | 98 | 1611 | 4819 | 503 | 81125 | 290 | 60 |
| http://rozetka.ua | 2542 | 17 | 527 | 22 | 54 | 1217 | 6381 | 35 | 61051 | 775 | 0 |
| http://trial-sport.ru | 3188 | 17 | 390 | 16 | 94 | 1630 | 16295 | 35 | 39601 | 405 | 36 |
| http://notebook-center.ru | 4344 | 33 | 232 | 31 | 144 | 2506 | 72222 | 32 | 33371 | 234 | 18 |
| http://new.itonv.lntu.info | 2095 | 20 | 148 | 19 | 128 | 1064 | 5602 | 166 | 14122 | 155 | 86 |