

удаленного рабочего стола; снижение времени на восстановление локальной рабочей станции, выполняющей функции терминального сервера.

Предложенная модель внедрена и успешно эксплуатируется в тестовом режиме на ООО "АСФ"СОЮЗ".

1.1С: Предприятие 8.2. Руководство администратора / А. Алексеев, А. Безбородов и др. – М.: 1С, 2009. – 239 с.

2.Спецификация процессора i5-3570k [Электронный ресурс]. – Режим доступа: [http://ark.intel.com/ru/products/65520/Intel-Core-i5-3570K-Processor-\(6M-Cache-up-to-3_80-GHz\)/](http://ark.intel.com/ru/products/65520/Intel-Core-i5-3570K-Processor-(6M-Cache-up-to-3_80-GHz)/).

3.Спецификация твердотельного накопителя SSD [Электронный ресурс]. – Режим доступа: <http://www.kingston.com/ru/ssd/s/>.

4.Спецификация модуля памяти [Электронный ресурс]. – Режим доступа: <http://www.kingston.com/ru/memory/hyperx/pnp/>.

5.Спецификация системной платы на базе чипсета z68 [Электронный ресурс]. – Режим доступа: http://www.gigabyte.ru/products/select2/mb/socket_1155/intel_z68/.

6.Спецификация сетевого адаптера [Электронный ресурс]. – Режим доступа: <http://www.dlink.ru/ru/products/7/365.html/>.

7.Спецификация накопителя NAS [Электронный ресурс]. – Режим доступа: <http://www.dlink.ru/ru/products/120/1477.html/>.

8.Microsoft Windows Server 2008 R2 / Р.Моримото, М. Ноэл, О.Драуби и др. – М.: ООО "И.Д.Вильямс", 2011. – 1456 с.

Получено 06.07.2012

УДК 004

М.В.ЗБИТНЕВА, канд. техн. наук

Харьковский национальный университет радиозлектроники

ПРИМЕНЕНИЕ ИНТЕЛЛЕКТУАЛЬНЫХ АГЕНТОВ В WINDOWS-ПРИЛОЖЕНИЯХ .NET FRAMEWORK 4.5

Описывается разработанный составной элемент управления `ctlAlarmClock`. Он выполняет функции будильника, который в качестве внутренней структуры использует модель и метод интеллектуального агента. Может применяться для включения в состав любого windows приложения в сфере коммунального хозяйства городов.

Розглядається розроблений складний елемент керування `ctlAlarmClock`. Він виконує функції будильника, який у якості внутрішньої структури використовує модель та метод інтелектуального агента. Може застосовуватися для включення до складу будь-якого windows програмного забезпечення у сфері комунального господарства міст.

It was described composite control `ctlAlarmClock`. It performs functions of alarm which as internal structure is used model and method of intellectual agent. It is possible to embed to any type of windows application in domain of communal services of city.

Ключевые слова: интеллектуальный агент, .Net Framework 4.5, пользовательские элементы управления, Windows Forms, Visual Studio .Net 2011, BackgroundWorker компонент, Timer компонент, Microsoft SQL Server 2012.

Рассмотрим проблему создания windows-приложений с использованием интеллектуальных агентов теории искусственного интеллекта. Связь этой проблемы с важными научными и практическими задачами заключается в построении автоматизированных систем более высокого уровня. Данное направление рассматривают авторы книги «Искусственный интеллект: современный подход» Стюарт Рассел и Питер Норвиг [1]. Оно освещено также в публикациях [2-5], в которых показаны различные сферы применения интеллектуальных агентов, а именно организация работы туристического агентства, создание псевдо-контента веб-сайтов, сбор и обработка информации в интернет, обработка электронной почтовой информации.

Данная статья направлена на создание windows-приложения, которое имеет внутреннюю структуру интеллектуального агента, использует .Net Framework 4.5 (Client Profile) и интегрирует пользовательские элементы управления.

Наследование позволяет создавать эффективные пользовательские элементы управления [6]. С помощью наследования можно создавать элементы управления, которые не только сохраняют все функциональные возможности стандартных элементов управления форм Windows Forms, но и реализуют пользовательские функции. Опишем создание простого наследуемого элемента управления ValueButton. Этот элемент наследует функциональные возможности стандартной кнопки Button Windows Forms и реализует настраиваемое свойство ButtonValue.

Чтобы реализовать элемент управления ValueButton создан проект под названием ValueButtonLib типа «Windows Forms Control Library». Для работы с ValueButton выбран тип UserControl, и заменен на тип Button. Это позволит элементу управления ValueButton унаследовать всю функциональность элемента управления Button. В файле ValueButton.Designer.cs в методе InitializeComponent удалена строка, присваивающая свойство AutoScaleMode, так как это свойство отсутствует в Button.

Один из возможных способов использования наследуемых элементов управления форм Windows Forms – это создание элементов управления, внешне идентичных стандартным элементам управления, но предоставляющих настраиваемые свойства. В данной статье в элемент управления было добавлено свойство ButtonValue в файл ValueButton.cs.

Элементы управления не являются автономными проектами, они включены в контейнер. Чтобы протестировать элемент управления, необходимо поместить его в тестовый проект. В данной статье демонстрируется построение элемента управления и его тестирование на форме Windows Form в приложении типа Windows Application под названием

Test. В приложении Test добавлена ссылка на проект ValueButtonLib.

Элемент управления ValueButton помещен на форму с панели элементов с вкладки ValueButtonLib Components.

В элементе label отображается цифра 3, что подтверждает передачу свойства ButtonValue наследуемого элемента управления элементу label с помощью метода valueButton_Click. Таким образом, элемент управления ValueButton наследует все функциональные возможности стандартной кнопки форм Windows Forms и, кроме того, предоставляет дополнительное настраиваемое свойство.

Перейдем к описанию второго созданного составного элемента управления ctlAlarmClock [7]. Составные элементы управления являются тем средством, с помощью которого можно создавать и многократно использовать пользовательские графические интерфейсы.

Сначала был создан простой составной элемент управления ctlClock. Затем функциональные возможности элемента ctlClock были расширены с помощью наследования.

В проекте ctlClockLib составной компонент ctlClock является наследником класса UserControl, предоставляемого системой. Класс UserControl обеспечивает функциональность, необходимую всем составным элементам управления, и реализует стандартные методы и свойства.

Составной компонент ctlClock содержит два элемента управления Label и Timer.

Поскольку Timer является компонентом, он не имеет визуального представления во время выполнения.

В конструкторе компонентов для компонента timer1 установлено значение свойства Interval равное 1000, что соответствует 1 секунде. Свойство Interval управляет частотой тиканья компонента Timer. Каждый раз, когда компонент timer1 тикает, выполняется код события timer1_Tick. Значение этого свойства – это количество миллисекунд между событиями Tick. Код события timer1_Tick позволяет отобразить текущее время в элементе управления lblDisplay.

В определение метода timer1_Tick добавлено ключевое слово virtual, чтобы сделать его доступным для переопределения.

Созданный элемент управления часы инкапсулирует элемент управления Label и компонент Timer, каждый из которых имеет собственный набор наследуемых свойств. Хотя свойства этих элементов управления не доступны пользователям создаваемого элемента управления, можно создать и предоставить другим приложениям настраиваемые свойства, написав соответствующие блоки кода.

В класс ctlClock добавлены две закрытые переменные colFColor,

colVColor. В них хранятся значения создаваемых свойств. Операторы `get` и `set` соответственно сохраняют и извлекают значение этих свойств под названием `ClockForeColor` и `ClockBackColor`, а также выполняют код, реализующий необходимую функцию.

Элементы управления не являются автономными приложениями, поэтому они включены в контейнер.

Чтобы протестировать элемент управления в таблице свойств тестового контейнера выбиралось свойство `ClockBackColor`, чтобы вывести на экран цветовую палитру. Аналогичные действия со вторым свойством были проведены.

Процесс создания производного класса на основе базового класса называется наследованием. Далее построен наследуемый составной элемент управления `ctlAlarmClock` на базе элемента управления `ctlClock`.

Функциональные возможности элемента `ctlClock` были расширены путем переопределения методов родительского класса и добавления новых методов и свойств.

Для цели создания `ctlAlarmClock` был использован шаблон `Inherited UserControl`.

Добавление свойств в наследуемый элемент управления выполняется точно так же, как и в случае с составным элементом управления. Был использован синтаксис объявления свойства для добавления трех свойств в элемент управления: `AlarmTime`, хранящее значение даты и времени срабатывания будильника, `AlarmSet`, показывающее, установлен ли будильник, и `AlarmSnooze`, демонстрирующее необходимость осуществления повторного звонка будильника через заданный промежуток времени.

Наследуемый элемент управления имеет такой же визуальный интерфейс, как и его родительский элемент управления. Наследуемый элемент управления также наследует все элементы управления, содержащиеся в родительском элементе управления, но их свойства недоступны из наследуемого элемента, если они не были предоставлены явным образом. В графический интерфейс наследуемого составного элемента управления можно добавлять элементы так же, как и в графический интерфейс любого другого составного элемента управления. В визуальный интерфейс будильника добавлен элемент управления надпись `lblAlarm`, который мигает при срабатывании будильника; а также пользовательский элемент управления `ValueButton`, который делает сигнал будильника видимым/невидимым и хранит в свойстве `ButtonValue` интервал времени перезвона будильника.

Свойства и элементы управления позволяют реализовать функциональные возможности будильника в составном элементе управления.

Переопределив метод `timer1_Tick` элемента управления `ctlClock` с использованием оператора `override` и добавив в него дополнительный код, можно расширить функциональные возможности наследуемого элемента управления `ctlAlarmClock`, сохранив в то же время все функциональные возможности элемента `ctlClock`. В методе `timer1_Tick` пользовательский элемент управления добавлен код, в котором текущее время сравнивается со временем срабатывания будильника. Если они совпадают, будильник начинает мигать 1 минуту без повторения или через 3 минуты с повторением.

Оператор `override` указывает, что элемент управления должен использовать этот метод вместо метода, унаследованного от базового элемента управления. При вызове этого метода он вызывает переопределяемый им метод с помощью вызова оператора `base.timer1_Tick`. Благодаря этому все функциональные возможности исходного элемента управления реализуются и в этом элементе. Затем выполняется дополнительный код, реализующий функциональные возможности будильника. При наступлении времени срабатывания будильника появляется мигающий элемент управления "Надпись".

Проверить наследуемый элемент управления можно таким же образом, как был проверен элемент управления базового класса `ctlClock`, а именно в тестовом контейнере пользовательских элементов управления.

Для использования элемента управления необходимо включить его в форму. Как и стандартный составной элемент управления, наследуемый составной элемент управления не может быть автономным и должен быть включен в форму или другой контейнер.

Чтобы добавить пользовательский элемент управления на форму необходимо добавить ссылку на него в эту форму.

Опишем созданное программное обеспечение «Будильник» в терминах интеллектуальных агентов [1] теории искусственного интеллекта. Интеллектуальным агентом является все то, что может рассматриваться как воспринимающее свою среду с помощью датчиков и воздействующее на эту среду с помощью исполнительных механизмов. В структуру агента-модели входят:

- датчики = устройства ввода;
- блок «как сейчас выглядит мир» = входные данные;
- правила «условие-действие» = функции программного приложения «Будильник»;
- исполнительные механизмы = устройства вывода;
- среда = пользователь.

Правила «условие-действие» программного приложения «Будильник» могут быть представлены в виде следующего метода:

1. Ввод времени будильника.
2. Выполнение срабатывания будильника в течение 1 мин. Возможность делать мигание будильника невидимым.
3. Возможность уменьшить время срабатывания будильника или инициировать его повторное срабатывание через 3 мин.
4. Возможность просматривать список незавершенных дел и сроки их исполнения [8, 9]. Используется Microsoft SQL Server 2012 Release Candidate 0.

Описанное программное обеспечение функционирует корректно, выполняет все приведенные правила, применяется для автоматизации процесса напоминания пользователю об определенных событиях. Для автоматизации работы с базами данных в .Net Framework 4 может быть также применен Visual Studio .Net LightSwitch. К перспективам дальнейших решений относится расширение сферы применения интеллектуального агента и его интеграция в Microsoft Visual Studio Connections.

1. Stuart J. Russell and Peter Norvig. Artificial Intelligence: A Modern Approach, Third Edition. Pearson Education, Inc., Prentice hall. 2010. – 1152 p.

2. Збитнева М.В. Интеллектуальный агент для организации работы туристического агентства // Комунальне господарство міст: Наук.-техн. зб. Вип.101. Серія: технічні наука і архітектура. – Харків: ХНАМГ, 2011. – С.488-491.

3. Збитнева М.В. Интеллектуальные агенты для создания псевдо-контента веб-сайтов // Бионика интеллекта, информация, язык, интеллект. – Харьков: ХНУРЭ, 2010. – №1(72). – С.104-108.

4. Дударь З.В., Збитнева М.В., Котенко А.В. Автоматизированные системы управления и приборы автоматики. Обработка электронной почтовой информации пользователя с применением интеллектуальных агентов // Всеукр. межвед. науч.-техн. сб. – 2008. – №142. – С.76-81.

5. Збитнева М.В., Дударь В.В., Деревянко А.Г. Сбор и обработка информации в интернет с использованием интеллектуальных агентов // Бионика интеллекта, информация, язык, интеллект. – Харьков: ХНУРЭ, 2008. – №2(69). – С.61-66.

6. MSDN Library/.NET Development/ .NET Framework 4.5 Developer Preview / .NET Framework Development Guide / Developing Client Applications / Windows Forms / Getting Started with Windows Forms / Windows Forms Controls / Developing Windows Forms Controls at Design Time / Walkthrough: Inheriting from a Windows Forms Control with Visual C#. [Электронный ресурс]. – Режим доступа: [http://msdn.microsoft.com/en-us/library/5h0k2e6x\(VS.110\).aspx](http://msdn.microsoft.com/en-us/library/5h0k2e6x(VS.110).aspx).

7. MSDN Library/.NET Development/ .NET Framework 4.5 Developer Preview / .NET Framework Development Guide / Developing Client Applications / Windows Forms / Getting Started with Windows Forms / Windows Forms Controls / Developing Windows Forms Controls at Design Time / Walkthrough: Authoring a Composite Control with Visual C#. [Электронный ресурс]. – Режим доступа: <http://msdn.microsoft.com/en-us/library/a6h7e207.aspx>.

8. MSDN Library / Development Tools and Languages / Visual Studio 11 Developer Preview / Visual Studio / Accessing Data in Visual Studio / Visual Database Tools / Understanding Visual Database Tools / Database Connections / How to: Connect to a Database from Server

Explorer. [Электронный ресурс]. – Режим доступа: [http://msdn.microsoft.com/en-us/library/0wbea1ae\(VS.110\).aspx](http://msdn.microsoft.com/en-us/library/0wbea1ae(VS.110).aspx).

9.MSDN Library / Servers and Enterprise Development / SQL Server / SQL Server 2012 Release Candidate 0 / Product Documentation / Books Online for SQL Server 2012 Release Candidate 0 (RC 0) / Database Engine / Transact-SQL Reference / Data Definition Language (DDL) Statements (Transact-SQL) / CREATE Statements (Transact-SQL) / CREATE DATABASE (Transact-SQL). [Электронный ресурс]. – Режим доступа: [http://msdn.microsoft.com/en-us/library/ms176061\(SQL.110\).aspx](http://msdn.microsoft.com/en-us/library/ms176061(SQL.110).aspx).

Получено 13.03.2012

УДК 614.8

А.А.ТЕСЛЕНКО, канд. физ.-матем. наук, А.Ю.БУГАЕВ

Национальный университет гражданской защиты Украины, г. Харьков

А.Б.КОСТЕНКО, канд. физ.-матем. наук

Харьковская национальная академия городского хозяйства

ИССЛЕДОВАНИЕ УСТОЙЧИВОСТИ МАТЕМАТИЧЕСКОГО АЛГОРИТМА РАСЧЕТА ПАРАМЕТРОВ АВАРИЙНОГО СЛИВА ОПАСНОГО ВЕЩЕСТВА ИМИТАЦИОННЫМИ МЕТОДАМИ

Рассмотрена проблема устойчивости алгоритма определения расчетных характеристик аварийного слива в зависимости от заданной максимально допустимой ошибки в коэффициенте расхода. Исследования проведены методами имитационного моделирования. Получен опыт применения имитационного моделирования к задачам моделирования объектов, на которых возможно возникновение аварии.

Розглянуто проблему стійкості алгоритму визначення розрахункових характеристик запобіжного клапана залежно від похибки у визначенні тиску в обладнанні. Дослідження проведено методами імітаційного моделювання. Отримано досвід застосування імітаційного моделювання до задач моделювання об'єктів, на яких можливо виникнення аварії з метою її запобігання.

The considered problem of firmness of algorithm of determination of calculation descriptions of safety-valve is depending on an error in the in pressure in equipped. Researches are conducted by simulation techniques. Experience of application of imitation design is got to the tasks designs of objects, on that maybe there is an origin of accident, with the aim of her prevention.

Ключевые слова: аварийный слив, пожарная безопасность технологических процессов, абсолютная ошибка, время опорожнения технологических аппаратов, авария, чрезвычайная ситуация, модель.

В случае возникновения аварии или пожара на производстве (в производственном помещении или на внешней технологической установке) возникает необходимость эвакуации или существенного уменьшения количества пожароопасных жидкостей, газов или паров, а также твердых горючих материалов, которыми заполнены технологические емкости аппараты и коммуникации. С этой целью на производстве устанавливают специальные системы, которые обеспечивают аварийную эвакуацию горючих веществ и материалов. Эвакуация легковоспламе-