

УДК 004

М.В.ЗБИТНЕВА, канд. техн. наук

Харьковский национальный университет радиоэлектроники

## ФОРМЫ WINDOWS .NET FRAMEWORK 4.5 ДЛЯ РЕАЛИЗАЦИИ ИНТЕЛЛЕКТУАЛЬНЫХ АГЕНТОВ

Описывается разработанный составной элемент управления, который 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, який 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, which 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 заданиями 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.

Данное приложение имеет на windows-форме элементы [6, 7]: часы; поля для ввода времени срабатывания, для отображения мигания будильника; кнопка для запуска/остановки срабатывания будильника; кнопка для выполнения функции видимости/невидимости; кнопка для перезвона через заданный промежуток времени, например через 3 минуты; табличную сетку, которая используется для просмотра списка незавершенных дел и сроков их исполнения Microsoft SQL Server 2012

Release Candidate 0 [8, 9].

В следующем проекте к будильнику добавляется функциональность фонового потока для поиска файлов со списком незавершенных дел по указанному пути [10].

Компонент `BackgroundWorker` заменяет аналогичный код из пространства имен `System.Threading` и расширяет его функциональные возможности, однако при необходимости исходное пространство имен `System.Threading` можно сохранить для обеспечения обратной совместимости и использования в будущем.

Формы `Windows Forms` используют модель однопоточного аппарата (STA), так как действуют на основе исходных окон `Win32`, которые наследуют потоковое подразделение. Модель STA подразумевает, что окно может быть создано для любого потока, но не может переключать созданные потоки. Кроме того, все вызовы функций должны выполняться для потока создания. Вне форм `Windows Forms` классы платформы `.NET Framework` используют свободную потоковую модель.

Модель STA требует, чтобы все методы элемента управления, которые необходимо вызывать не из потока создания элемента управления, маршировались в поток создания этого элемента управления и выполнение осуществлялось в нем же. Базовый класс `Control` предоставляет для этих целей несколько методов (`Invoke`, `BeginInvoke` и `EndInvoke`). `Invoke` осуществляет синхронные вызовы метода; `BeginInvoke` осуществляет асинхронные вызовы метода.

При использовании в элементе управления многопоточности для ресурсоемких задач пользовательский интерфейс может остаться работоспособным, если ресурсоемкие вычисления выполняются в фоновом потоке.

В данном проекте `DirectorySearcher` демонстрируется многопоточный элемент управления форм `Windows Forms`, использующий фоновый поток для рекурсивного поиска каталога с файлами, соответствующими заданной строке поиска, а затем выполняется заполнение списка результатами поиска. Ниже приведены ключевые понятия, используемые в проекте:

- `DirectorySearcher` запускает новый поток для выполнения поиска. Поток выполняет метод `ThreadProcedure`, возвращающий вызовы вспомогательного метода `RecurseDirectory` для выполнения фактического поиска и заполнения списка. Заполнение списка требует выполнения вызова между потоками.

- `DirectorySearcher` определяет метод `AddFiles` для добавления файлов в список; однако, `RecurseDirectory` не может напрямую вызвать `AddFiles`, поскольку `AddFiles` может работать только в STA-потоке, соз-

давшем DirectorySearcher.

– Единственный способ для RecurseDirectory вызвать AddFiles – это использование вызова между потоками, другими словами, вызова Invoke или BeginInvoke для маршалинга AddFiles в поток создания DirectorySearcher. RecurseDirectory использует BeginInvoke, что позволяет осуществлять вызовы асинхронным способом.

– Маршалинг метода требует эквивалент указателя функции или обратного вызова. Это делается с помощью делегатов в .NET Framework. BeginInvoke принимает делегат в качестве аргумента. Поэтому DirectorySearcher определяет этот делегат (FileListDelegate), вызывает AddFiles к экземпляру FileListDelegate в его конструкторе и передает этот экземпляр делегата элементу управления BeginInvoke. DirectorySearcher также определяет делегат события, маршалируемый после завершения поиска.

Данный проект выполняет изменение цвета мигания будильника и осуществляет вывод содержимого файла [11] со списком незавершенных дел с использованием компонента BackgroundWorker. Данная функция прерывает любую функцию будильника на время открытия файла и после ее завершения возобновляет прерванную операцию.

При нажатии на кнопку «Загрузить» обработчик событий Click вызывает метод RunWorkerAsync компонента BackgroundWorker для запуска операции загрузки. Кнопка отключается во время загрузки и по окончании загрузки снова становится активной. В поле MessageBox отображается содержимое файла.

Файл загружается в рабочем потоке компонента BackgroundWorker, запускаемом обработчиком событий DoWork. Этот поток запускается, когда код вызывает метод RunWorkerAsync. Событие DoWork на время загрузки файла меняет цвет мигания будильника.

Обработчик событий downloadButton\_Click определяет порядок ожидания завершения асинхронной задачи компонентом BackgroundWorker.

Если предпочтительнее, чтобы приложение лишь реагировало на события и не выполняло никакой работы в основном потоке в ожидании, пока не завершится выполнение фонового потока, необходимо завершить работу обработчика.

Если требуется продолжить выполнение работы в основном потоке, необходимо использовать свойство IsBusy, чтобы определить, по-прежнему ли выполняется поток BackgroundWorker. Индикатор выполнения обновляется в процессе загрузки. Для сохранения отклика пользовательского интерфейса обязательно вызывается метод Application...:DoEvents.

Метод `backgroundWorker1_RunWorkerCompleted` обрабатывает событие `RunWorkerCompleted` и вызывается по завершении фоновой операции. Этот метод сначала проверяет свойство `AsyncCompletedEventArgs.Error`. Если свойство `AsyncCompletedEventArgs.Error` имеет значение `null`, то данный метод отображает содержимое файла. Затем он активирует кнопку загрузки, которая была отключена, когда началась загрузка, и сбрасывает индикатор выполнения.

В следующем проекте происходит демонстрация на форме по таймеру изменения времени срабатывания будильника [12]. В этом проекте применяются атрибуты. Для разработки компонентов и элементов управления, которые правильно взаимодействуют со средой разработки и безошибочно решаются во время выполнения, применяют атрибуты к классам и методам.

В данном проекте используется несколько атрибутов в проекте с пользовательским элементом управления. Элемент управления выполняет запись данных в журнал. После связывания элемента управления с источником данных он отображает значения, отправленные источником данных в элемент управления `DataGridView`. Если значение превышает указанное в свойстве `Threshold`, инициируется событие `ThresholdExceeded`.

`AttributesDemoControl` записывает значения с классом `LogEntry`. Класс `LogEntry` является классом шаблона, то есть он параметризован в типе, который он записывает.

Так как `LogEntry` параметризован произвольным типом, он использует отражение для работы в параметризованном типе. Чтобы функция порогового значения работала, тип параметра `T` должен реализовать интерфейс `IComparable`.

Форма с `AttributesDemoControl` периодически опрашивает счетчик производительности. Каждое значение упаковывается в `LogEntry` соответствующего типа и добавляется в `BindingSource` формы. `AttributesDemoControl` получает значение через свою привязку данных и выводит значение в элементе управления `DataGridView`.

Одна часть данного проекта представляет реализацию `AttributesDemoControl`. Вторая – демонстрирует элемент управления и форму с использованием `AttributesDemoControl`.

Некоторые атрибуты применяются на уровне класса, например атрибут `TypeConverterAttribute` для класса `LogEntry`. Класс `TypeConverterAttribute` указывает, какой тип использовать в качестве преобразователя для объекта, с которым связан этот атрибут.

Атрибуты уровня членов применяются на уровне члена класса. Например, атрибут `AmbientValueAttribute`. Класс `AmbientValueAttribute`

задает значение, которое нужно передать свойству, чтобы это свойство получило значение из другого источника. Это называется окружением.

Опишем созданное программное обеспечение «Будильник» в терминах интеллектуальных агентов [1] теории искусственного интеллекта. 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 минуты. Возможность делать мигание будильника невидимым.
3. Возможность уменьшить время срабатывания будильника или инициировать его повторное срабатывание через 3 минуты.
4. Возможность вывода на экран в фоновом режиме списка незавершенных дел из файла на диске.
5. Возможность поиска файлов со списком незавершенных дел по указанному пути и отображения результата.
6. Возможность сохранения и отображения множества точек времени срабатывания будильника.

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. Випуск 101. Серія: технічні науки 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, №1(72), 2010. – С. 104-108.

4. Дударь З.В., Збитнева М.В., Котенко А.В. Автоматизированные системы управления и приборы автоматики // Всеукр. межвед. науч.-техн. сб. – Вып. 142, 2008. – 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, №2(69), 2008. – 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#. [Електронний ресурс].

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#. [Електронний ресурс].

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. [Електронний ресурс].

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). [Електронний ресурс].

10. 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 / Multithreading in Windows Forms Controls / How to: Use a Background Thread to Search for Files. [Електронний ресурс].

11. 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 / Controls to Use on Windows Forms/ BackgroundWorker Component/ How to: Download a File in the Background. [Електронний ресурс].

12. 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 Custom Windows Forms Controls with the .NET Framework/ Attributes in Windows Forms Controls / How to: Apply Attributes in Windows Forms Controls. [Електронний ресурс].

*Получено 27.11.2012*

УДК 543.3:628.1:663.6

В.О.ГАСВСЬКА, В.А.ШУР, кандидати техн. наук, А.Ю.КАБИШ  
*Харківський національний університет будівництва та архітектури*

### **ОРГАНІЗАЦІЙНО-ТЕХНІЧНА СИСТЕМА УПРАВЛІННЯ БУДІВЕЛЬНИМИ ПРОЕКТАМИ В СФЕРІ ЗАБЕЗПЕЧЕННЯ НАСЕЛЕННЯ ЯКІСНОЮ НЕФАСОВАНОЮ ПИТНОЮ ВОДОЮ**

Розроблена керована організаційно-технічна система «Управління будівельними проектами в сфері постачання населенню нефасованої питної води з підземних джерел» (УБП ПВ), на основі якої можна розробляти й управляти конкретними унікальними проектами в складі цієї системи. Пропонується використовувати її при створенні малим та середнім бізнесом підприємств з виробництва, транспортування і реалізації населенню якісної нефасованої питної води з підземних джерел автомобільним транспортом.