УДК 004.81:159.935.5

ПРОГРАММНО-АГЕНТНАЯ СРЕДА ДЛЯ ПОДДЕРЖКИ СОВМЕСТНОГО ОБУЧЕНИЯ НА ОСНОВЕ ПЛАТФОРМЫ MICROSOFT .NET

С.Н. Воног, К.А. Жереб, Т.Ю. Кушко

Киевское отделение Московского физико-технического института
03650, Киев, проспект Академика Глушкова, 40,
svonog@gmail.com, zhereb@gmail.com, taras.kushko@mail.ru

Функции совместной работы и интерактивного взаимодействия в реальном режиме времени являются важной частью многих видов человеческой деятельности, в том числе и обучения. Однако, часто они не могут быть реализованы, поскольку в условиях среднестатистического учебного класса недоступны дорогое специальное оборудование и соответствующее программное обеспечение. В данной работе предлагается подход для достижения положительных эффектов, связанных с совместным обучением и взаимодействием всех участников учебного процесса, относительно простым и недорогим способом. Описана упрощенная модель агентной платформы для поддержки приложений совместного обучения под названием Inspirational Classroom Environment (ICE, «Вдохновляющая классная среда»). Эта система написана на языке программирования C# для платформы Microsoft .NET и использует Microsoft Research ConferenceXP в качестве платформы для разработки функций совместной работы и передачи мультимедийного содержимого в реальном режиме времени. Анализируется пример применения системы ICE для обучения детей младшего школьного возраста с помощью ролевых игр, совместного рисования, создания анимационных роликов и персональных обучающих агентов.

Collaboration and interaction features to support real time activities have become an important feature of many practices and education but often can not be used because expensive equipment and sophisticated software are unavailable in standard classroom environments. We propose an approach to achieve collaboration and interaction effects in relatively simple and inexpensive way by development of lightweight agent platform supporting collaborative learning applications called Inspirational Classroom Environment (ICE). The system is written in C# and uses Microsoft .NET platform and ConferenceXP product as the basis for the development of real time collaborative applications and rich multimedia content. A case study is carried out to use the ICE system in learning children by means of collaborative drawing, animations and personal tutor agents.

Введение

Впечатляющие достижения в развитии технологий беспроводных сетей, Интернета и мобильных устройств за последние годы сделали возможным создание учебных классов и заведений с возможностью широкого применения компьютерных устройств для интерактивного обучения. Данные исследований свидетельствуют, что участники процесса обучения способны гораздо лучше усваивать информацию при условии, что они работают в небольших группах [1]. Богатые возможности для совместной работы, предоставляемые компьютерными приложениями [2], способствуют более быстрому обучению, более долгосрочному запоминанию информации, повышению уровня заинтересованности и мотивации [3], [4]. Поэтому особенно важными для многих видов человеческой деятельности, в том числе и обучения, являются функции совместной деятельности и интерактивность в режиме реального времени [5]. Однако, эти функции редко реализуются в современных информационных системах из-за отсутствия возможности применять дорогостоящее оборудование и беспроводные сети, а также сложное и дорогостоящее программное обеспечение в условиях среднестатистического учебного класса.

Основная цель данной работы состоит в том, чтобы показать, что положительные эффекты совместного интерактивного обучения могут быть достигнуты относительно простым и недорогим способом с помощью разработки упрощенной агентной платформы, ориентированной на поддержку процессов совместного обучения. Система, впервые описана в [6], написана на языке программирования C# для платформы Microsoft .NET и использует Microsoft Research ConferenceXP [7] в качестве платформы для разработки функций совместной работы и передачи мультимедийного содержимого в реальном режиме времени.

Архитектура, построенная на основе ConferenceXP, предназначена для работы в широкополосных сетях, предоставляет расширяемую инфраструктуру с широкими возможностями для совместной работы и межличностного взаимодействия. Так, известны примеры внедрения ConferenceXP для обучения компьютерным наукам в университете шт. Вашингтон, США [8] и других университетах США с использованием экспериментальной сетевой инфраструктуры следующего поколения Internet2. Наряду с разработкой технологии ConferenceXP, данный проект Microsoft Research и университета шт. Вашингтон, известный как Learning Experience Project [9], ставил себе цель исследовать возможности дистанционного совместного обучения, а также сделать его реальной и полноценной альтернативой обычному обучению, при условии наличия доступа к технологиям, которые лишь начинают появляться на рынке, таким как: сети высокой пропускной способности, беспроводные устройства, планшетные компьютеры и новейшие возможности операционных систем Microsoft. ConferenceXP позволяет университетам и исследовательским организациям затратить минимум усилий на создание необходимой инфраструктуры и архитектуры

программной системы и сконцентрироваться на исследованиях и разработке прототипов приложений для совместной работы, которые бы улучшили процесс обучения, как в рамках классной аудитории, так и за ее пределами. Однако, насколько известно авторам, Conference XP до настоящего времени не была использована в рамках программных систем, построенных с помощью многоагентного подхода к проектированию ПО.

Данная работа состоит из 2 частей. Первая – состоит в исследовании и разработке упрощенной агентной платформы ICE, реализованной в виде библиотеки классов, предоставляющей возможности для удобного использования, расширения и настройки под нужды конкретного приложения. Визуальные агенты в созданной платформе могут быть более сильно связаны с интерфейсом приложений, использующих функциональность этих агентов, чем традиционные агенты с графическим интерфейсом пользователя. В рассматриваемой разработке графический интерфейс агента может быть частью интерфейса основного приложения.

Вторая – состоит в рассмотрении примера применения построенной агентной платформы для создания интерактивной системы совместного обучения детей дошкольного и младшего школьного возраста [10]. Авторами были разработаны 3 приложения (каждое в виде отдельного агента): 1. «Волшебные чернила». 2. «Персонализированные упражнения и интеллектуальный обучающий агент». 3. «Командное рисование». «Волшебные чернила» – это простой в освоении инструмент для создания анимационных роликов. С его помощью дети могут проще выражать свои мысли, создавая визуальные динамические представления своих идей на планшетных ПК. Интеллектуальный обучающий агент адаптируется к способу изучения учебных материалов ребенком и дает учебные материалы и подсказки в виде различных последовательностей, отличающихся длиной, набором упражнений и их типами, таким образом, чтобы максимально способствовать усвоению материала учащимся. «Командное рисование» предполагает различные виды совместной работы, связанные с рисованием на планшетных ПК. С помощью «Командного рисования» дети могут одновременно вместе рисовать на виртуальном «холсте», создавая новые виды цифрового совместного творчества.

1. Совместное обучение и Microsoft Research ConferenceXP

Любая более-менее серьезная задача человеческой деятельности в современном мире вряд ли может быть полностью разрешена одним человеком. Однако, несмотря на это, в школах детей не учат техникам совместной работы. Это хорошо, поскольку учебный процесс должен гарантировать усвоение определенных знаний и навыков обучающимися. С другой стороны, это плохо, поскольку обучающиеся оказываются не приспособленными к реальным сценариям работы, в которых для того, чтобы достичь результата, приходится интенсивно общаться и сотрудничать с коллегами.

С появлением легких, мобильных планшетных ПК ученики и учителя получают возможность работать над идеями совместно в удобной виртуальной среде. Уроки могут превратиться в коллективные многопользовательские обучающие игры. Например, ученики делятся на команды и принимают участие в сеансе создания совместных рисунков. Задание для каждой команды дается учителем и состоит в том, чтобы нарисовать определенную картину за ограниченный период времени, по истечении которого возможность рисования на экранах планшетных ПК отключается программой. Все ученики имеют различные роли. Это выражается в том, что каждый из них имеет различный набор инструментов для рисования: черная ручка, толстые разноцветные кисти, ластик, емкость с чернилами, … (у команды чернила могут закончиться, если она будет рисовать слишком расточительно и небрежно). Поэтому ученикам приходится вступать в процесс коммуникации для того, чтобы быстро создать совместно красивый рисунок в соответствии с заданием. Таким образом, на практике будут усваиваться техники и методики совместной работы.

ConferenceXP предоставляет расширяемую основу для разработки приложений для совместной работы и проведения конференций. С помощью ConferenceXP пользователи могут взаимодействовать друг с другом внутри виртуального пространства для совместной работы, называемого venue. Обогащая традиционную методику обучения, технология ConferenceXP облегчает создание распределенных приложений для совместной работы с помощью ряда свойств [7]. В данной работе были использованы следующие:

- ConferenceXP является приложением типа peer-to-peer, т.е. данные пересылаются между клиентами напрямую, а центральный сервер не используется;
- для поддержки множественных пользователей и минимизации сетевого трафика ConferenceXP использует технологию многоадресного вещания (multicast);
- ConferenceXP поддерживает традиционнее одноадресное вещание в режиме точка-точка в сетях, которые не поддерживают многоадресное вещание.

ConferenceXP не рекомендуется использовать в беспроводных сетях с высокой степенью потери сетевых пакетов [11]. Однако, эксперименты показали, что система ICE на базе ConferenceXP и упрощенной агентной платформы стабильно работает в беспроводных сетях, развернутых в рамках учебной аудитории. В системе ICE ConferenceXP была использована для разработки агента для активности «Командное рисование».

2. Вдохновляющая Классная Среда (ICE)

Вдохновляющая Классная Среда, а в сущности Inspirational Classroom Environment (ICE) – это решение на основе платформы ConferenceXP и многоагентного подхода для построения системы для совместного обучения и развития творческих способностей детей младшего школьного возраста. Каждый ученик в классе и учитель имеют планшетный ПК с беспроводным сетевым модулем. Все планшетные ПК в классе объединены в

одну беспроводную сеть и через сетевой концентратор подключены к классному серверу, изображение с экрана которого выводится на классную доску с помощью стандартного цифрового проектора (рис. 1).

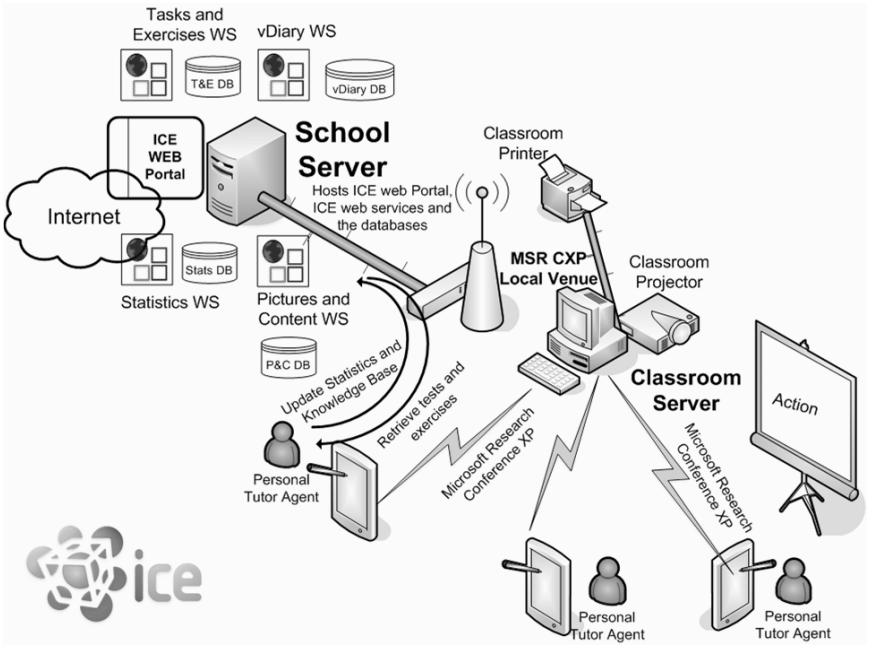

Рис. 1. Физическая диаграмма системы Inspirational Classroom Environment

С точки зрения ПО ICE – распределенная программная система. В системе существуют 3 различных приложения: приложения учителя, серверное приложение, отвечающее за вывод информации на классную доску с помощью проектора и приложение ученика. Рассмотрим активность «Совместное рисование», реализованную в системе ICE. В рамках указанных приложений на каждом ПК выполняются ряд агентов: по 1 агенту на каждую активность системы (рисование, отправка упражнений) и набор служебных агентов. Архитектура агентной платформы и обмен сообщений агентов реализован в соответствии со спецификациями стандарта FIPA [12]. Это позволяет расширять функциональность распределенной системы путем создания новых типов агентов в соответствии со стандартом FIPA. Для хранения информации агенты используют специальные веб-сервисы: статистики упражнений, учебных материалов, картинок и творческих работ, виртуального дневника.

Для реализации функциональности активности «Командное рисование» была использована функциональность программы Classroom Presenter [13], разработанной в университете шт. Вашингтон. Возможности Classroom Presenter были расширены добавлением программных фильтров для идентификации элементов рисунка отдельных учеников, а также для реализации различных инструментов рисования. Каждый элемент рисунка на планшетном ПК в ConferenceXP имеет уникальный идентификатор в системе (GUID). Это позволяет уникально идентифицировать элементы рисунка, созданные каждым учеником на всех клиентских планшетных ПК – ПК учеников и ПК учителя. Для удаления элемента рисунка на все компьютеры, зарегистрированные в системе, отправляется пакет RTStrokeRemove с указанием уникального идентификатора удаляемого элемента рисунка. Помимо глобального уникального идентификатора, каждому элементу рисунка присваивается идентификатор ученика, что позволяет учителю в реальном режиме времени оценивать вклад каждого ученика в совместную работу.

Система написана на языке программирования C# для платформы Microsoft .NET и использует Microsoft Research ConferenceXP [7] в качестве платформы для разработки функций совместной работы и передачи мультимедийного содержимого в реальном режиме времени. В нашей системе Classroom Presenter был реализован в форме агента с графическим интерфейсом пользователя. Все данные хранятся на отдельном сервере баз данных, под управлением СУБД SQL Server 2000. Сохранение и чтение данных производится с помощью специальных веб-сервисов (рис. 1).

3. Агентная система ICE

Решение ICE было разработано в соответствии со спецификациями стандарта FIPA в виде легковесной многоагентной платформы в среде Microsoft .NET Framework для использования в различных приложениях. В последнее время появилось большое количество агентных платформ [14], но в своем большинстве они реализованы на языке Java. Таким образом, первым основным преимуществом ICE является его изначальная ориентация на платформу .NET. Это позволяет использовать множество приложений и библиотек, которые разрабатываются под эту платформу, в частности, ConferenceXP. Другие отличия ICE не настолько очевидны и обусловлены предполагаемым использованием агентной платформы.

Многие известные агентные платформы (такие как JACK [15] и ZEUS [16]) предоставляют серьезную поддержку интеллектуальному поведению агентов. Цель данных агентных платформ заключается в разработке интеллектуальных агентов, которые могут проявлять сложное поведение как сами по себе, так и как часть агентной системы. Другие платформы (такие, как JADE [17], LEAP [18], а также ZEUS) фокусируются на строгой реализации спецификаций стандарта FIPA, которые предоставляют возможность взаимодействия между различными агентными платформами. В таких платформах все внимание смещено к осуществлению взаимодействия внутри агентной системы и они могут быть использованы для создания сообществ агентов, в которых большую роль играет взаимодействие друг с другом. Существует и третье направление в разработке агентных платформ, в котором агенты рассматриваются как технология программирования, способствующая разработке сложных распределенных приложений. Данный тип платформ лучше всего описывается в понятиях спецификаций OMG MASIF [19] (примером является Grasshopper [20]) и наша агентная платформа также соответствует этому направлению.

Основная цель агентной системы ICE – более удобное создание приложений для совместной деятельности, в том числе и для совместного обучения. Наша агентная система сама осуществляет все низкоуровневые технические детали, такие как создание сетевых соединений между агентами, поиск определенных агентов и т.д. С точки зрения разработчика, все агенты расположены в единой среде (которая может физически располагаться на различных компьютерах). Они могут обмениваться сообщениями вне зависимости от содержания и формата передаваемой информации. В то же время, агенты могут использовать определенные сервисы самой платформы при поиске других агентов и при сохранении либо возобновлении своего состояния на сервере. В системе не предусматривается реализация всех возможных протоколов взаимодействия (например, определенных в стандартах FIPA). Наоборот, разработчикам на платформе ICE вместо того, чтобы подстраиваться под существующие протоколы взаимодействия, предлагается самим реализовать специализированные протоколы для своих нужд.

В нашей агентной платформе нет определенных свойств, которые обычно присущи другим платформам (таких как явная поддержка задач, диалогов агентов и даже мобильности), т.к. эти свойства не требуются для наших приложений. Также отсутствует поддержка для любого вида интеллектуального поведения. Но даже без этих свойств, мы можем продемонстрировать, что smart-агенты могут быть реализованы независимо, используя агентную платформу для поддержки сетевого взаимодействия (примером данного подхода является Personalized Tutor Agent). Также это позволяет оставить нашу платформу легковесной и простой для понимания (библиотека классов, написанная на 2000 строках кода на языке C#, включает в себя 10 классов с более чем 50 открытыми методами)

4.1. Структура агентов. Основные сущности, представленные в ICE, – агенты и диспетчера агентов. Диспетчер агентов соответствует контейнеру в FIPA и каждый диспетчер может управлять несколькими агентами. Различные диспетчеры могут располагаться как на разных компьютерах, так и на одном (в различных доменах приложений). Расположение диспетчеров агентов заранее не задано (за исключением основного диспетчера). Более того, оно может динамически изменяться во время работы агентной платформы. У основного диспетчера месторасположение задано заранее и не меняется со временем, т.к. к нему должны обращаться все остальные диспетчера для соединения с платформой. Основное представление архитектуры агента показано на рис. 2.

Агентная платформа ICE реализует свою функциональность с помощью классов: основного диспетчера агентов, диспетчера агентов, агента Agent Management Service (AMS), конфигурации системы, системы графического пользовательского интерфейса (GUI). Далее будет более подробно описана функциональность платформы.

Агентная платформа ICE реализована в виде библиотеки классов, где каждый класс, который наследуется от базового класса Agent, является агентом. Таким образом, у всех агентов реализована общая базовая функциональность, определенная в базовом классе, а именно:

- однозначная идентификация агента во всей платформе. У каждого агента есть свой идентификатор (Agent Identifier, AID), состоящий из трех полей: название агента (agent name), тип агента (agent type) и расположение агента (agent location) (название диспетчера, который отвечает за данного агента). Каждый агент получает свой AID во время создания и сохраняет его неизменным на всей продолжительности своей работы. Уникальность каждого AID обеспечивается диспетчером, который создает его;
- возможность получения сообщений. Реализуется с помощью следующего метода:

public virtual void Receive(Message msg).

Каждая реализация агента может переопределить данный метод для реализации собственного протокола взаимодействия. Базовая реализация данного метода позволяет только регистрировать содержания сообщения для отладочных целей;

- возможность реагировать на соединение с платформой. Реализуется с помощью следующего метода:

public virtual void StartNetwork().

Также есть несколько вспомогательных методов, способствующих выполнению агентами общих задач.

Агенты могут проявлять как *реактивное* (*reactive)*, так и *проактивное (proactive)* поведение, хотя сама агентная платформа слабо обеспечивает возможность реализации поведения агентов. Реактивное поведение реализовано с помощью метода Receive. Есть два основных метода для задания проактивного поведения. Первым является так называемая *network-independent* инициализация, которая реализована в конструкторе

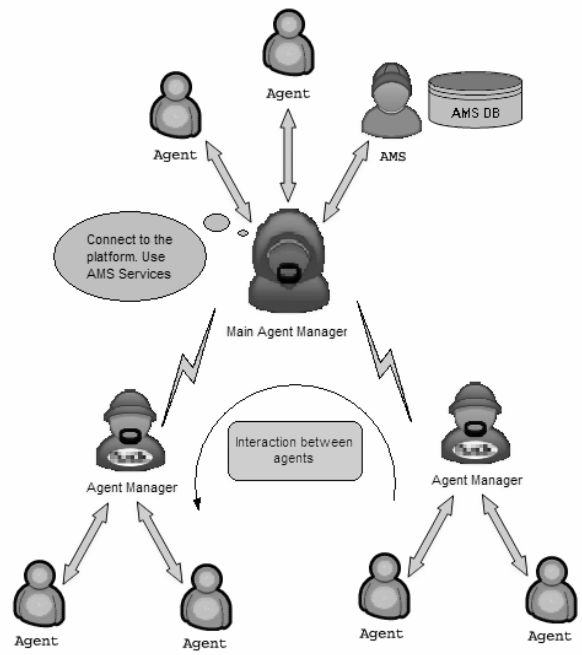

Рис. 2. Архитектура агентов в ICE

определенного подкласса класса Agent и обычно используется для инициализации системы пользовательского интерфейса (GUI). Второй метод может использоваться для *network-dependent* инициализации, что реализуется при переопределении метода StartNetwork и используется для инициализации протоколов взаимодействия с другими агентами и AMS. Для *network-dependent* инициализации необходимо наличие активного соединения с платформой, в то время как для *network-independent* инициализации соединения с платформой не требуется.

4.2 Управление агентами и настройка платформы. Диспетчера агентов являются средой для жизнедеятельности простых агентов и предоставляют им основные сервисы. Они включают в себя:

- *создание агента* − необходимо задать название агента (agent name), тип агента (agent type) и название сборки, которая содержит данный тип агента;
- *получение ссылки на агента* − может быть использовано для прямого взаимодействия между локальными агентами;
- *отправка сообщения* − основной сервис, который предоставляется диспетчерами агентов;
- *предоставление AID агента AMS*.

Все остальные сервисы предоставляются специальными агентами. Один из таких агентов, который является жизненно важным для функционирования агентной платформы, является AMS (agent management service). Физически AMS также является агентом. Но логически было бы правильнее его рассматривать как диспетчера агентов, который содержит всю актуальную информацию о системе в целом. Наш AMS реализует функциональность как AMS, так и DF (directory facilitator), описанные в стандарте FIPA.

Диспетчер агентов, в котором находится AMS, является основным диспетчером. Его расположение неизменно и известно всем диспетчерам (что достигается с помощью системы конфигурирования). Все другие диспетчеры соединяются с основным диспетчером и регистрируются в AMS. После этого они считаются соединенными с платформой. В действительности, AMS не используется для отправки сообщений – при отправке сообщений от одного диспетчера к другому между ними устанавливается прямое соединение. Таким образом, если AMS (или основной диспетчер) недоступны, агенты, которые уже знают друг друга, могут продолжать взаимодействовать. Но для получения информации о новых агентах (необходимой для взаимодействия с ними) без AMS не обойтись. В частности, AMS может использоваться для получения списка всех агентов определенного типа или всех агентов с заданным месторасположением. Агент также может подписаться на рассылку сообщений AMS для получения информации об определенных событиях в агентной системе, таких как появление нового агента.

Для задания параметров, необходимых для работы агентов в системе ICE, используется система конфигурирования. Таким образом, можно изменить конфигурацию агентной платформы без внесения изменения в код. Если у нас есть несколько агентов с графическим пользовательским интерфейсом, каждый из которых находится в отдельной сборке (DLL), тогда все, что нам нужно сделать для создания сложного приложения со всеми возможностями данных агентов, это определить, какие из агентов должны быть загружены. Это можно сделать либо с помощью конфигурационной утилиты в форме отдельного приложения, либо в основном приложении. Таким образом, это улучшает настраиваемость и расширяемость системы ICE.

Конфигурационные данные хранятся в отдельном XML файле. Его можно редактировать с помощью конфигурационной утилиты. Агенты получают доступ к этой информации через специальный класс.

Конфигурационные данные являются общими для диспетчера и для всех агентов в нем. Конфигурационная система хранит всю информацию, необходимую для диспетчеров, а именно: список сборок, содержащих классы агентов; список агентов, которые необходимо загрузить (определяются тип агента и желаемое название агента; если название агента уже используется, то оно генерируется автоматически); расположение основного диспетчера (хост, порт); и предпочтительный локальный порт (если он уже занят, то используется последующий свободный порт). Агенты также могут хранить свою информацию с помощью системы конфигурирования.

У агентов может быть графический пользовательский интерфейс. Он может быть реализован в виде отдельных форм либо элементов управления, которые загружаются в родительское приложение. В ICE, родительское приложение предоставляет набор контейнеров, в которых могут находиться различные элементы управления. Некоторые из контейнеров могут содержать только один тип элементов управления (например, кнопки с рисунком), другие могут содержать элементы управления произвольного типа. Последние часто используют все пространство основного приложения, элементы управления, которые в них загружаются, и представляют собой отдельные приложения. Только один из элементов управления является активным в любой момент времени, и переключение между ними осуществляется с помощью кнопок с изображениями. Различные агенты могут независимо загружать свои элементы управления в основной интерфейс, таким образом, им неизвестно об элементах управления других агентов.

4. Приложения ICE

Нами было разработано три простых приложения (каждое из которых реализовано в виде отдельного агента) для демонстрации потенциала нашей платформы: волшебные чернила (Visual Thinking);персонализированные упражнения и интеллектуальный обучающий агент; командное рисование.

Волшебные чернила. Visual Thinking − средство для создания анимаций, основанным на технологии Microsoft® Ink для планшетных компьютеров [21]. Цель приложения − создать средство для создания анимаций, которое можно было бы просто и интуитивно использовать, и которое позволяло пользователю быстро визуализировать свои собственные мысли посредством создания анимаций, которые, в свою очередь, могли бы использоваться в качестве прототипа пользовательских графических интерфейсов, создания собственных мультипликаций и т.д. Приложение разработано в виде пользовательского элемента Windows, который можно встраивать в любое приложение, использующее .NET Framework. Так как функциональность Visual Thinking разработана для планшетных компьютеров и использует определенную функциональность, доступную только в планшетных компьютерах, то при запуске приложения на обычных компьютерах заметно небольшое отличие в отображении графической информации. Пользовательский интерфейс Visual Thinking позволяет создавать чернильные изображения с помощью технологии cut&paste. Это означает, что вы можете копировать и вставлять кадры из других документов, и это позволяет создавать анимации *так же просто, как и думать о них.*

Как элемент управления для других .NET приложений, Visual Thinking может использоваться так же, как и любой другой элемент управления, и предоставляет несколько простых возможностей для его управления (таких, как Запуск(Play), Пауза(Pause), свойства Time и UseEditor, которые позволяют создавать собственные редакторы анимаций и воспроизведения). Также, элемент управления Visual Thinking предоставляет доступ к объекту Ink, в котором представлен текущий отображаемый кадр, что позволяет использовать любую функциональность технологии Microsoft® Ink, также как если бы вы работали с любым другим объектом Ink. Рекомендуется использовать именно элемент управления Visual Thinking, т.к. он упрощает большинство операций с помощью вызова одной или двух функций; однако, архитектура также позволяет создавать собственные элементы управления со структурой схожей с использованием диаграммы классов технологии Microsoft® Ink. Например, используя StrokePath (для представления изменения объекта класса Stroke на протяжении времени), и InkPath (который представляет собой набор объектов класса Stroke на протяжении времени).

Функциональность Visual Thinking используется в платформе ICE в виде отдельного агента, вследствие чего эта функциональность легко может быть использована в приложении ученика. Для создания анимации необходимо производить действия в определенном порядке (которые многие люди и так делают) и затем изменить рисунок для новой отметки времени. Созданная анимация может быть перенесена не только на любой другой планшетный компьютер, но и на проектор, где анимации (визуализированные мысли) можно просмотреть и обсудить в классе (рис. 3).

Рис. 3**.** Начальный и конечный рисунок в агенте Visual Thinking

Персонализированные упражнения и интеллектуальный обучающий агент. Второе приложение, разработанное в форме агента, − интеллектуальный обучающий агент (Smart Tutor Agent). Он активно взаимодействует с учеником, учителем и сервером, который используется для хранения задач, образовательного материала, статистики и т.д. Необходимость создания данного приложения основана на том факте, что учителя не могут уделять много внимания каждому ученику (они не могут одновременно работать со многими учениками сразу); очень редко они могут позволить себе готовить для каждого ученика по индивидуальному заданию; очень сложно проследить, что происходит в классе и что делает каждый ученик. Проверка тестов и домашних заданий занимает достаточно много учительского времени; ученики не могут получить мгновенный ответ и комментарии от учителя по своей деятельности и тестам сразу после их завершения.

У каждого ученика есть свой собственный интеллектуальный обучающий агент внутри планшетного компьютера − представленный в виде анимированного персонажа. Обучающий агент адаптируется к способу обучения учеником и предоставляет учебные материалы в той последовательности, качестве, количестве, которое наиболее подходит под темп обучения ученика. В качестве дополнительной стимуляции, существо может менять свое состояние, поведение и внешность в зависимости от успехов самого ученика. Агент помогает учителю мотивировать ученика, производит ряд рутинных операций, которые обычно делает учитель. Эти операции включают в себя подборку индивидуальных тестов, обучающего материала и домашних заданий для каждого ученика, сбор полной статистики об успеваемости, а также помощь и ободрение учеников во время выполнения упражнений.

Совместное рисование. Используя виртуальный «холст» и возможность отправки пакетов мультикастом мы создали среду для командного рисования на основе ConferenceXP. В нем используются три типа приложений: приложение учителя; серверное приложение, отвечающее за вывод информации на классную доску с помощью проектора; приложение ученика.

Вначале учитель делит учеников на несколько команд (в нашем случае изображены две команды на рис. 4, 5 и 6). Задача каждой команды состоит в том, чтобы нарисовать определенную картинку (тему называет учитель) за ограниченное время, после которого рисовать на экранах учеников станет невозможно. Каждый из учеников получает различные инструменты для рисования: черную ручку, цветные кисти, ластик, волшебное ведерко красок (если команда рисует не очень экономно или слишком много, то количество краски может исчерпаться). Все инструменты заранее раздаются учителем. Дети рисуют одновременно, создавая новый вид цифрового творчества. Во время рисования ученики общаются друг с другом, чтобы вместе нарисовать красивый рисунок за ограниченное время. Таким образом, они обучаются на практике использовать совместную деятельность.

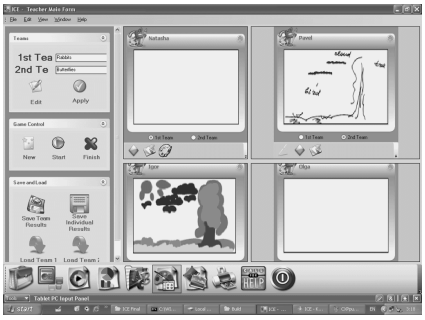

Рис. 4. Интерфейс учителя

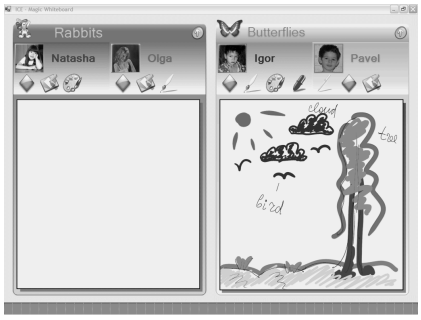

Рис. 5. Интерфейс проектора

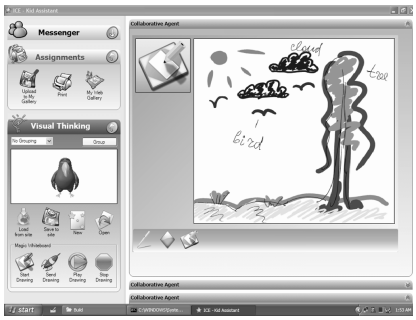

Рис. 6. Интерфейс ученика

Выводы

В данной работе мы представили новую модель интерактивного способа обучения, основанного на платформе Microsoft .NET и технологии ConferenceXP. Основываясь на этой модели, нами была разработана характерная программная среда для образовательных целей, названная Inspirational Classroom Environment (ICE). Полученные результаты свидетельствуют, что эта среда может позволить многим ученикам получить

всесторонний опыт в обучении и увеличить вовлеченность преподавательского состава в процесс обучения. Как было отмечено ранее, эта среда представляет собой технологическую основу для интерактивного совместного обучения в режиме реального времени (Real Time Interactive Collaborative Education). Технологически, мы сможем измерить производительность среды ICE под текущей нагрузкой. Нашим следующим шагом станет ориентация на разработку настраиваемой программной среды, которая поддерживает большое количество основанных на ICE классов.

Воодушевляя учеников новым способом обучения, технология ConferenceXP, расширенная применением системы ICE, является шагом вперед к новому качеству образования. Проект ICE направлен на расширение возможности обучения с помощью масштабируемой, настраиваемой технологии, которая поддерживает обучение в классе. Мы стремимся улучшить технологию ICE, включая добавление поддержки приложений для моделирования и визуализации, виртуальной среды, а также всевозможных игр.

Благодарность. Авторы выражают благодарность за помощь и поддержу компании “Microsoft” (Россия) и Agent R&D Group в МФТИ, а также профессору А.Е. Дорошенко.

1. *Webb N.M., Troper J.D, Fall R.* Constructive activity and learning in collaborative small groups // Journal of Educational Psychology. –1995, № 87(3). – P. 406–423.
2. *Griswold W. et al.* ActiveCampus: Experiments in Community-oriented Ubiquitous Computing // Computer. – 2004, № 37(10). – P. 73–81.
3. *Barak M., Maymon T.* Aspects of teamwork observed in a technological task in junior high schools. // J. of Technology Education. –1998, № 9(2). – P. 1–27.
4. *Van Boxtel C., Van Der Linden J., Kanselaar G.* Collaborative construction of conceptual understanding: interaction processes and learning outcomes emerging from a concept mapping and a poster task // J. of Interactive Learning Research. – 1997, № 8(3/4). – P.341-361.
5. *Hassler V.* Online Collaboration Products. // Computer. – 2004, № 37(11). – P. 106–109.
6. *Doroshenko A., Kushko T., Vonog S., Zhereb K.* A Lightweight Agent System for Collaborative Learning Support // Proc.4-th Int. Conf. ISTA'2005, New Zealand, May 23-25, 2005, Lecture Notes in Informatics. – 2005. – Vol. P–63.– P.49–60.
7. *ConferenceXP***,** http://www.conferencexp.net
8. *University* of Washington Department of Computer Science & Engineering http://www.cs.washington.edu/masters/dl_tech/
9. *The Learning* Experience Project http://research.microsoft.com/research/ pubs/view.aspx?tr_id=746
10. *Microsoft* Crowns Imagine Cup 2004 Champions http://www.microsoft.com/presspass/press/2004/jul04/07-06Champions04PR.asp
11. http://www.conferencexp.net/community/Default.aspx?tabindex=12&tabid=15 #Does_ConferenceXP_work_over_a_wireless_network
12. *FIPA* Abstract Architecture Specification http://www.fipa.org/specs/fipa00001/
13. *University* of Washington. Classroom Presenter Project http://www.cs.washington.edu/education/dl/presenter/
14. http://fipa.org/resources/livesystems.html
15. *Hodgson* A., *Roenquist R., Busetta P.* Specification of Coordinated Agents Behavior (The Simple Team Approach). Agent-Oriented Software Pty., Ltd, Melbourne, Australia, http://www.agent-software.com
16. *Collis J., Ndumu D.* Zeus Technical Manual. Intelligent Systems Research Group. BT Labs. British Telecommunications. 1999. – 33 p.
17. *Bellifemine F., Poggi A., Rimassa G.* Developing multi-agent systems with a FIPA-compliant agent framework // Software - Practice And Experience. – 2001, № 31(2) – P. 103-128
18. *LEAP* http://leap.crm-paris.com/
19. *OMG* Mobile Agent Facility Specification http://www.omg.org/technology/documents/formal/mobile_agent_facility.htm
20. *Grasshopper* http://www.grasshopper.de/
21. *Microsoft* Ink http://msdn.microsoft.com/library/en-us/tpcsdk10/lonestar/ Microsoft.Ink/Microsoft.Ink.asp