

Разработка систем распределенного обучения, подчиненная моделям

В. Е. Кривцов, А. А. Ярославцев

Проблемы распределенного обучения

Развитие систем компьютерного обучения и, в частности, распределенного обучения [1] обусловлено необходимостью интенсификации процессов обучения. К использованию компьютеров в обучении подталкивают, с одной стороны, быстрый рост спроса на образовательные услуги и, с другой, ограниченность ресурсов роста преподавательского корпуса.

Предполагалось, что компьютерные программы будут в состоянии заменить учителей во многих областях их профессиональной деятельности. За счет этого должна существенно возрасти степень индивидуализации обучения, что неизбежно приведет к более высокому качеству знаний при одновременном снижении затрат на обучение.

Эти ожидания отчасти реализовались в современных системах компьютерного обучения. В настоящее время существует довольно много конкретных обучающих систем, нацеленных на школьное, высшее и корпоративное образование, в которых участие преподавателей в процессе обучения, если не сведено к минимуму, то существенно сокращено по сравнению с традиционной моделью обучения. В первую очередь, речь идет об освобождении преподавателей от рутинных функций: ведения дневников, раздачи учебных материалов, учета выполнения заданий и т. п.

Что же касается интеллектуальных видов деятельности преподавателей, то здесь ситуация выглядит далеко не столь обнадеживающей. Проблемы по-прежнему сохраняются:

1. Отсутствие индивидуального подхода к обучению. Индивидуализация рутинных функций управления обучением на самом деле еще не означает, что осуществлен индивидуальный подход к обучению. Индивидуальное компьютерное обучение предполагает наличие моделей обучения, которые были бы в состоянии адаптироваться к индивидуальным психо-физиологическим особенностям обучающихся. Пока таких моделей в обучении нет.

2. Невозможность в полной мере заменить учителя современными программно-техническими средствами. Исследования по созданию искус-

ственного интеллекта нацелены, в частности, на решение данной проблемы, но в настоящее время отсутствуют технические средства, способные ее разрешить.

3. Проблема оценивания уровня знаний. Большинство систем распределенного обучения являются обычными средствами тестирования, в которых применяется примитивный формальный подход для оценивания знаний тестируемых.

4. Отсутствие живого контакта между учеником и учителем.

Многоуровневые синхронные коммуникации

На наш взгляд, современное состояние дел в области разработки средств компьютерного обучения не позволяет надеяться на то, что в ближайшие годы появятся системы, действительно способные заменить учителей в процессах реального обучения. Интенсификация процессов обучения в настоящее время, как и раньше, возможна только за счет повышения качества индивидуальной работы преподавателя с каждым учащимся.

Попытаемся понять, как применение компьютеров и Интернета может реально способствовать решению этой задачи.

В настоящей работе предлагается использовать информационные технологии не для замещения учителя в процессах обучения в целях уменьшения количества учителей, а, наоборот, для нужного наращивания численности преподавательского корпуса.

Этого можно достичь, разрешив обучающимся в определенных обстоятельствах выступать в роли учителей. Можно, например, ввести многоуровневую аттестацию учащихся, разрешив учащемуся уровня n выступать в качестве инструктора для обучающихся, находящихся на более низких уровнях, начиная, скажем, с уровня $(n - 1)$ ¹⁾. Конкретных схем организации таких многоуровневых систем обучения может быть очень много.

Для успеха предлагаемой схемы необходимо снабдить ее средствами живых контактов между преподавателями и обучающимися. В существующих обучающих системах наиболее распространенным средством обмена сообщениями между учащимися и учителями является электронная почта. Реже используются средства мгновенной коммуникации, такие как ICQ, голосовая или видео связь. Практически полностью отсутствуют средства поддержки мгновенных обменов предметной информацией: геометрическими чертежами, математическими формулами, 3D-моделями химических молекул и т. п. Отсутствие средств поддержки живых контактов с преподавателем существенно снижает усвояемость знаний.

Средства поддержки живых контактов в системах обучения с несколькими уровнями «учитель-ученик» далее будем называть системами многоуровневых синхронных коммуникаций (СМСК).

¹⁾ Эта идея несколько лет назад была высказана А. П. Афанасьевым.

Поясним используемый термин «синхронные коммуникации» более подробно.

Коммуникацией называется обмен сообщениями, направленный на достижение определенной цели, участниками которого являются люди²⁾.

В процессе распределенного обучения не всегда имеют место коммуникации. Человек может обучаться во время игры при помощи персонального компьютера или во время тестов, когда локальный или территориально удаленный компьютер сразу выдает результаты тестирования по мере прохождения одного или нескольких тестов. В данном случае коммуникации нет, процесс обучения происходит синхронно.

Существуют системы тестирования, когда ученик, выполнив тест, отправляет его на проверку и получает ответ от компьютера не сразу, а через длительное время. В таких случаях можно говорить об асинхронном обучении.

Асинхронным взаимодействием является передача данных посредством SMS, электронной почты, передача информации в режиме off-line (когда обмен информацией произойдет не сразу) при помощи MS Windows Messenger. Участниками подобных взаимодействий являются люди, поэтому справедливо применить термин асинхронная коммуникация. Асинхронной коммуникацией могут являться опять же тесты, на которые отвечает человек — ученик, и обрабатываемые человеком — учителем. Особенностью асинхронной коммуникации является задержка, которая приводит к разрыву усвоения учеником информации.

Системы, направленные на «живое» участие в образовательном процессе человека в качестве ученика и человека в качестве учителя, относятся к разряду систем синхронных коммуникаций. Обычный телефонный разговор, видеоконференция, компьютерные ролевые игры по Интернет, непосредственное общение через MS Windows Messenger — примеры широко распространенных синхронных коммуникаций. Применение систем распределенного обучения, поддерживающих синхронные коммуникации (работающих в режиме real-time), происходит далеко не везде, хотя очевидно, что пользу применения подобных систем в обучающем процессе трудно переоценить.

СМК может решить основные проблемы распределенного обучения следующим образом:

- Отсутствие индивидуального подхода к обучению восполняется наличием учителя и синхронными коммуникациями между учеником и учителем.
- Проблема невозможности в полной мере заменить учителя современными техническими средствами исключается, поскольку при син-

²⁾ Вообще говоря, термин коммуникация может быть отнесен и к взаимодействию человека с компьютером. Однако в настоящей работе эта трактовка не применяется — термин коммуникация относится только к взаимодействию людей.

хронных коммуникациях присутствует и учитель и ученик, так что задача замены учителя техническими средствами не ставится.

- Проблема оценки уровня знаний не возникает, поскольку всегда в контур обучения может быть включен учитель, который произведет нужную оценку.
- Отсутствие живого контакта между учеником и учителем восстанавливается синхронной передачей информации разного формата, наличием модели «ученик — учитель» в основе системы и усиливается техническими возможностями использования вычислительных способностей компьютерных систем. Наличие средства общения ученика и учителя — основа СМСК распределенного обучения.
- Недостаток учительского состава может быть устранен гибкой настраиваемостью многоуровневой модели коммуникационных соединений между учителем и учеником и адаптацией приложения под постоянно изменяемые требования к ССК. Модель может допускать возможность существования учителей, являющихся одновременно и учениками у более квалифицированных учителей, при этом может возникнуть целый класс СМСК.

Модельный подход к разработке программных систем

Как уже отмечалось выше, СМСК это не конкретная система, а целый класс возможных систем разной архитектуры. Подобные системы могут найти применение не только в распределенном обучении, но также и во многих других областях. Поэтому в разрабатываемую систему хотелось бы заложить максимальную гибкость и способность к быстрой перенастройке.

Для того чтобы разработать СМСК, которая обладала бы такими возможностями, необходимо применить определенный подход, основанный на модельном представлении архитектуры программной системы. Таким подходом является архитектура разработки, подчиненная моделям (MDA, Model Driven Architecture) [2]. Данная архитектура в своей основе опирается на общепризнанные стандарты — UML, MOF, XMI, CWM — и нацелена на решение проблемы переходов между платформами промежуточного уровня (CORBA, J2EE, .Net, Web-Services) и поддержки часто изменяющихся приложений.

Создание MDA преследовало следующие задачи [3]:

- Охватить в одном стандарте такие технологии, как CORBA, J2EE, XML/SOAP, Web-Services, .Net, а также предусмотреть появление новых, пока неизвестных технологий.
- Повысить портируемость моделей, чтобы одна и та же модель могла быть реализована на различных платформах с помощью вспомогательных стандартов.

- Повысить интегрируемость на основе моделей, относящихся к различным предметным областям.

Основной идеей MDA является последовательное применение единого подхода к моделированию. MDA включает:

- единый подход к разработке моделей;
- представление архитектуры программной системы в виде платформо-независимой и платформо-зависимой модели;
- определение способа преобразования платформо-независимой модели в специфическую для конкретной платформы модель.

Единый подход к разработке моделей

При разработке сложных программных систем обычно создают модели, которые отображают систему с разных точек зрения. Для разработки модели необходим язык, имеющий синтаксис и семантику, предназначенные для описания программных систем. В свою очередь, структуру и свойства языка описания программных систем можно представить также в виде модели — метамодели, составленной на языке более высокого уровня абстракции — метаязыке.

Общепризнанным стандартом по организации и работе с метаданными, моделями и метамоделями является спецификация MOF [4]. Эта спецификация представляет собой формальный метод описания метаобъектов, в своей основе использующий синтаксис и семантику унифицированного языка моделирования — UML (Unified Modeling Language).

В концепции представления метаданных MOF выделены четыре метаровня:

- Уровень объектов M0. Объекты являются экземплярами элементов уровня моделей и представляют собой определенные значения (Customer Jane Smith, Account 2989).
- Уровень моделей M1. На уровне моделей находятся элементы — экземпляры метамодели (Class “Customer”, Class “Account”).
- Уровень метамодели M2. Метамодели состоят из элементов, являющихся экземплярами элементов уровня мета-метамодели (UML Class, UML Attribute).
- Уровень мета-метамодели M3. Мета-метамодели содержат элементы, используемые для определения метамодели (MOF Class, MOF Attribute).

Для четырехуровневого представления метаданных MOF характерна независимость элементов более низкого уровня от элементов более высокого. К примеру, появление нового элемента Account 2344 на уровне M0 не может изменить элемент Class “Account” уровня моделей M1, появление нового элемента Class “Customer” на уровне M1 не может изменить элемент UML Class уровня метамодели M2. Таким образом, наличие метамодели позволяет достичь независимости от изменений моделей.

Количество метауровней, на которых приходится создавать модели для описания разрабатываемых программных систем, зависит от целей разработки и наличия или отсутствия уже готовых стандартных моделей. Например, при разработке хранилищ данных не требуется создавать метамодель (уровень M2) — такая метамодель CWM уже существует и стандартизована OMG. Для представления метаданных большинства систем достаточно первых двух метауровней. Более высокие уровни появляются тогда, когда перед разработчиками ставится задача обеспечить будущую легкую интегрируемость системы, ее интероперабельность с другими приложениями и возможность простого и быстрого внесения изменений.

Применительно к задаче создания системы поддержки МСК для целей распределенного обучения можно выделить следующие метауровни: уровень метамодели, уровень модели и уровень объектов МСК распределенного обучения. Для единого описания возможностей многоуровневой системы «ученик — учитель», допускающей существование учителей, являющихся одновременно и учениками у более квалифицированных учителей, необходима MOF-совместимая метамодель стратификации, позволяющая по единому принципу строить разные модели, основными элементами которой являются уровни и объекты, расположенные на уровнях. Необходима также метамодель взаимодействия ученика с учителем, позволяющая строить разные модели, в основе которых лежит элементарное взаимодействие «ученик — учитель».

Платформо-независимые и платформо-зависимые модели

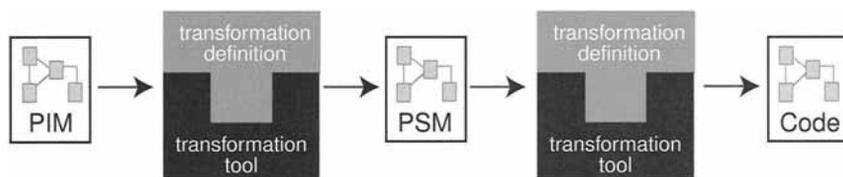
В архитектуре, подчиненной моделям, выделяют три уровня абстракции представления системы в процессе разработки — три типа моделей:

- модель предметной области (CIM, Computational Independent Model);
- платформо-независимая модель (PIM, Platform Independent Model);
- платформо-зависимая модель (PSM, Platform Specific Model).

Модель предметной области предназначена в первую очередь для обеспечения понимания и взаимодействия экспертов предметной области и специалистов по анализу и дизайну разрабатываемой системы на начальном этапе формирования требований и проектирования. Платформо-независимая модель предназначена для более детального представления системы с точки зрения ее реализации, но при этом абстрагируясь от конкретной платформы реализации распределенной системы. Платформо-зависимая модель рассматривает все детали программной системы, учитывая характерные особенности конкретной платформы (CORBA, J2EE, .Net, Web-Services).

Преобразования моделей

Модельный подход рассматривает не только модели, языки и платформы, но и преобразования. Под преобразованием понимается набор



Преобразования моделей

правил и приемов, используемых для превращения одной модели в другую [3]. Обобщенная схема преобразований платформо-независимой модели в платформо-зависимую модель и платформо-зависимой модели в программный код приведена на рисунке выше [2].

При необходимости изменения программной системы изменения вносятся в ее платформо-независимую модель. После этого к ней применяются преобразования, которые ведут к изменению платформо-зависимой модели и программного кода.

Стандарты в области распределенного обучения и в смежных областях

Известными стандартами создания распределенных систем обучения являются спецификации консорциумов IMS и ADL (Advanced Distributed Learning), определяющие основные концепции построения любой обучающей системы: методы обучения, роли участников обучения, их возможные действия по отношению к объектам обучения, а также способы организации и представления ресурсов обучения.

IMS LD

Спецификация IMS Learning Design (LD) Information Model [5] интегрирует концепции языка EML (Educational Modelling Language) и набора стандартов IMS. Одной из важных концепций метамодели распределенной системы обучения является объект обучения (learning object), метаданные для которого определяются спецификацией IEEE LTSC LOM. LD Information Model представляет собой профиль UML для описания метамодели систем распределенного обучения, в основе которой предполагается передвижение обучаемого по курсам.

SCORM

Международный стандарт консорциума ADL SCORM (Sharable Courseware Object Reference Model) [1] определяет формирование электронного содержимого обучающих курсов и предполагает наличие системы управления прохождением по курсам.

В качестве примеров стандартов в смежных областях, представляющих интерес для целей настоящей разработки, следует упомянуть UML

Activity Graph, Workflow Management Facility, SPEM (Software Process Engineering Metamodel).

UML Activity Graph

Метамоделю UML Activity Graph является частью метамодели UML [6] и предназначена для описания последовательности выполнения действий и условий переходов между ними. Последовательность переходов отображается в виде ориентированного графа и может определить логику выполнения любого действия, в частности любое действие, возникающее при синхронных коммуникациях. В качестве языка метамодели используется UML.

Workflow Management Facility

Спецификация Workflow Management Facility [7] основана на стандартах консорциума Workflow Management Coalition и применима для описания бизнес-процессов — сценариев взаимодействия бизнес-объектов. Применение данного стандарта возможно и в распределенном обучении. Перемещение ученика по курсам можно представить как бизнес-процесс и описать в конструкциях Workflow Management Facility.

SPEM

Стандарт SPEM (Software Process Engineering Metamodel) [8] разработан консорциумом OMG и предназначен для описания процессов разработки программных систем. В метамодели SPEM присутствуют средства для описания взаимодействий участников разработки программного обеспечения и артефактов, возникающих в подобных взаимодействиях. Конструкции данной метамодели могут быть адаптированы для создания метамодели распределенного обучения.

Постановка задачи и предлагаемые пути ее решения

Итак, задача состоит в создании системы поддержки многоуровневых синхронных коммуникаций для целей распределенного обучения. К этой системе предъявляются характерные функциональные и нефункциональные требования.

Функциональные требования

Основными функциями такой системы обучения должны являться:

- управление уровнями;
- управление переходами между уровнями;
- управление соединениями между учителем и учеником;
- поддержка синхронных коммуникаций.

1. Управление уровнями. Данная функция должна отслеживать необходимое количество уровней, степень их заполненности, в ней должны

присутствовать возможности создания и уничтожения уровней, управления правами и обязанностями, политикой безопасности и другими вопросами применительно к каждому уровню.

2. Управление переходами между уровнями. Эта функция должна обеспечить переход «ученика»/«учителя» на другой уровень по определенному признаку при выполнении определенных условий (например, это может быть что-то вроде защиты диплома, когда собирается комиссия и решает вопрос о повышении квалификации претендента на переход).

3. Управление соединениями между учителем и учеником. Данная функция должна предоставлять одновременную поддержку многим разнородным связанным между собой конструкциям «ученик — учитель» в разных сочетаниях между «учителями» и «учениками» разных уровней.

4. Поддержка синхронных коммуникаций. Данная функция должна реализовывать механизм обмена данными во время интерактивного взаимодействия, с учетом того, что учитель и ученик физически удалены друг от друга и СМСК может работать в среде разных операционных систем.

Нефункциональные требования

С учетом широкого спектра возможных применений СМСК, основной задачей является разработка системы, обладающей сформулированной выше общей функциональностью и способной к быстрому перенастройкам и изменениям, а также обладающей продвинутыми возможностями интеграции и интероперабельности с существующими программными системами. Такая система с неизбежностью должна разрабатываться по принципам MDA.

Поскольку системы такого рода до сих пор не создавались, вопрос возможности реализации этой схемы остается открытым. Именно в этой части разработки будет аккумулирована ее исследовательская направленность.

Этапы решения задачи

Выбранный подход к разработке системы диктует основные моменты в выполнении работ. Для решения поставленной задачи необходимо:

- Разработать MOF-совместимую метамодель стратификации.
- Разработать MOF-совместимую метамодель взаимодействия ученика с учителем.
- Разработать правила преобразования платформо-независимой модели программной системы в специфическую для конкретной платформы модель.
- Разработать систему, реализующую преобразование модели программной системы в специфическую для конкретной платформы модель.

Подчеркнем, что существующие стандарты в области распределенного обучения, равно как и в смежных областях, не ориентированы на СМСК. Необходимо создать новую метамодель, дополняющую существующие стандарты, которая определит класс многоуровневых систем синхронного обучения.

1. Разработка метамодели стратификации. Данная модель должна быть предназначена для определения специальных конструкций, имеющих соответствующую многоуровневую семантику и позволяющих строить модели в рамках одного домена — домена стратификации — и правил работы с этими конструкциями. Примерами подобных конструкций могут быть следующие: «уровень», «объект», «переход объекта на другой уровень», «критерий перехода объекта на другой уровень», «условие перехода объекта на более высокий (низкий) уровень», «ассоциация между двумя объектами разных уровней». Примером правила работы с подобными конструкциями может быть следующее высказывание: «объект не может быть перемещен на более высокий уровень, пока не выполняются условия перехода».

Для разработки данной метамодели необходимо применить язык UML. Метамодель стратификации должна быть MOF-совместимой.

2. Разработка метамодели взаимодействия ученика с учителем. Данная модель должна быть предназначена для определения специальных конструкций моделирования домена обучения и правил работы с этими конструкциями.

Примерами подобных конструкций могут быть следующие: «ученик», «ученик», «оценка», «роль», «действие», «курс», «учебный материал». Примером правила работы с подобными конструкциями может быть следующий запрет: «ученик не может поставить себе оценку».

Для разработки данной метамодели также, как и для метамодели стратификации, необходимо применить язык UML, она должна быть MOF-совместимой.

3. Разработка правил преобразования модели программной системы в специфическую для конкретной платформы модель. Способ преобразования должен быть определен согласно спецификации MOF и должен позволять в большей степени автоматически преобразовывать платформо-независимую модель программной системы в платформо-зависимую.

4. Разработка системы, реализующей преобразование модели программной системы в специфическую для конкретной платформы модель. Разработка системы, реализующей преобразование модели программной системы в специфическую для конкретной платформы модель, является итоговой задачей. Она должна производиться согласно основным принципам подхода MDA с использованием метамоделей стратификации и обучения, учитывая существующие стандарты и средства в области распределенного обучения, используя аналогичные стандарты в смежных областях.

Литература

1. Electronic resource: www.adlnet.org/index.cfm?fuseaction=ADLTechnologies.
2. Kleppe A., Warmer J., Bast W. MDA explained: the model driven architecture: practice and promise. Addison-Wesley, 2004.

3. *Цимбал А. А., Анишина М. Л.* Технологии создания распределенных систем. Для профессионалов. СПб.: Питер, 2003.
4. MOF Specification, v. 1.4 (www.omg.org/technology/documents/formal/mof.htm).
5. IMS Learning Design Information Model, v. 1.0 (www.imsglobal.org/learningdesign/ldv1p0/imslid_infov1p0.html).
6. Unified Modeling Language Specification, v. 1.5 (www.omg.org/technology/documents/formal/uml.htm).
7. Workflow Management Facility Specification, v. 1.2 (www.omg.org/docs/formal/00-05-02.pdf).
8. Software Process Engineering Metamodel, v. 1.1 (www.omg.org/technology/documents/formal/spem.htm).