

УДК 004.4'242

Анализ применения редакторов онтологий с физической семантикой в педагогической деятельности вуза

Д.А. Филипишин¹, А.В. Григорьев², Е.И. Приходченко³

Донецкий национальный технический университет, г.Донецк

¹domaco@mail.ru, ²grigorievalvl@gmail.com, ³88rapoport88@mail.ru

Аннотация

Рассматриваются популярные методы и средства интерактивных обучающих систем, а также способы использования редактора онтологий как средства CASE- и CALS-технологии. Проводится сравнительный анализ на примере условно доработанного программного продукта до CAD/CAE системы, для описания эффективности обучения будущих программистов и разработчиков, с целью более глубокого понимания парадигмы объектно-ориентированного программирования. Принципиальным отличием предложенного подхода является более приближенный метод использования онтологического инжиниринга в рамках объектно-ориентированного программирования.

Постановка проблемы

Выпускник высшей школы, особенно технической, должен обучаться на протяжении всех жизни, постоянно обновлять свои знания, повышать, развивать и углублять свой уровень компетенций – такое требование выставляют не только Европейские образовательные стандарты, но и Российские [1]. На их решение и нацелена технология интерактивного обучения.

Можно отметить работы ряда специалистов, занимающихся созданием обучающих систем и педагогическими инновациями, таких как Приходченко Е.И., Воробьев Г.Г., Маштабиз Е.И., Шевченко О.И., Ярмухмедова К.Г. [1-7].

С другой стороны, существует такое направление, как онтологический инжиниринг (ОИ), который может рассматриваться как система формальных определений в конкретной предметной области и представленный в виде предлагаемого редактора онтологий с физической семантикой.

Физическая семантика, прежде всего, свойственна задачам САПР, предполагающих наличие CAD/CAM/CAE подсистем.

Онтологический инжиниринг способен упростить технологию интерактивного обучения для учащихся. Проблемой разработки в направлении онтологического инжиниринга занимаются ряд специалистов Боргест Н.М., Гаврилова Т.А., Соловьев В.Д., Доброхотов А.Л., Скобелев П.О., Гартман Н., Клещев А.С., Артемьева И.Л., Черняховская Л.Р., Токарев Д.В., Григорьев А.В. [8-15].

Таким образом, актуальной задачей является не решённая проблема использования онтологического инжиниринга в практике высшей школы.

Цель и задачи исследования

Исходя из поставленной проблемы, рассмотреть применение предлагаемого «усовершенствованного» редактора онтологий на учебных занятиях.

Провести анализ его использования как прикладной технологии (CASE технология):

- в виде графической интеллектуальной системы (CAM);
- умной базы данных (CAD);
- базы знаний для расчета сложных физических процессов и экспертных систем (CAE).

Также следует рассмотреть применение предлагаемого редактора онтологий в машиностроении, на примере САПР механического производства и роботостроения (CALS технология).

Сравнение CASE и CALS технологий

CASE-технология представляет собой совокупность методов и средств проектирования, разработки и сопровождения сложных систем программного обеспечения, поддержанную комплексом программных средств автоматизации. В настоящее время практически ни один серьёзный программный продукт не осуществляется без использования CASE-средств.

Основная цель этой технологии заключается в том, чтобы отделить кодирование программного продукта от его проектирования на всех этапах разработки. В частности, CASE-средства делятся на две большие группы: средства проектирования спецификаций и структуры (не поддерживающие полный жизненный цикл программного продукта), а также средства генерации исходных текстов и

реализации интегрированного окружения поддержки полного жизненного цикла разработки программного продукта.

Чаще всего, понятие «CASE-технология» ассоциируется с UML или семейством стандартов IDEF0-14, позволяющими проектировать практически любой возможный программный продукт, который только может представить разработчик. Реже с интерактивными программами в целом.

Можно сказать, что CASE-технология является САПР программного обеспечения.

Основное отличие CALS-технологии от CASE в том, что первая используется для программирования механических станков, роботов и производства, как в военной, так и в промышленной отрасли. Это отличие не влияет на тот факт, что в любой момент программу по управлению действиями станка, устройства или робота можно перенести в заранее подготовленную программную среду. После чего протестировать процесс работы, проанализировать статистику, сделать экономические выводы и внести правки.

Применение CALS-технологии позволяет существенно сократить объёмы проектных работ, так как описания многих составных частей оборудования, машин и систем, проектировавшихся ранее, хранятся в унифицированных форматах данных сетевых серверов, доступных любому пользователю. На текущий момент считается, что успех на рынке (особенно в век развития нейронных сетей и роботостроения) сложной технической продукции будет немислим вне технологий CALS.

Главной проблемой построения открытых распределённых автоматизированных систем для проектирования и управления в промышленности является обеспечение единообразного описания и интерпретации данных, независимо от места и времени их получения в общей системе, имеющей масштабы вплоть до глобальных. Отсюда становится реальной успешная работа над общим проектом разных коллективов, разделённых во времени и пространстве и использующих разные CAD/CAM/CAE-системы.

Онтологический инжиниринг и редактор онтологий

Определим основные используемые понятия, необходимые для освоения онтологического анализа. Онтология – философское учение об общих категориях и закономерностях бытия, существующее в единстве с теорией познания и логикой [8]. Онтология (в информатике) – это попытка всеобъемлющей и детальной формализации некоторой области знаний с помощью концептуальной модели.

Онтологический анализ – разделение реального мира на составляющие и классы объектов, определение их онтологий или же совокупности фундаментальных свойств, которые определяют их изменения и поведение.

В проектировании онтологий условно можно выделить два направления:

— **весомые или тяжёлые** - связаны с представлением онтологии как формальной системы, основанной на математически точных аксиомах,

— **лёгкие** – связаны с компьютерной лингвистикой и когнитивной наукой (система абстрактных понятий).

Редакторами или конструкторами онтологий называют инструментальные программные средства, созданные специально для проектирования, редактирования и анализа онтологий. Основная функция любого редактора онтологий состоит в поддержке процесса формализации знаний и представлении онтологии как спецификации [8]. В построении онтологий используется достаточно много языков программирования, например: Ontolingua +KIF, OKBC, XOL, OWL. Обычно языки программирования в этой области формальны, они – обобщения языков – фреймов, они кодируют декларативные знания и обычно базируются на логике.

Интерес к разработке онтологий плавно нарастает как со стороны разработчиков интеллектуальных систем, так и со стороны бизнес-аналитиков. Усилия исследователей в основном были направлены на разработку технологических инструментов и примеров. Однако разработка практических онтологий в производстве, проектировании и менеджменте, особенно ИТ-менеджменте, остаётся скорее на уровне «искусства» [4].

На сегодня требуется решить ряд задач для дальнейшего развития онтологического инжиниринга, с целью более гибкого применения его на практике. Прежде всего необходимо решить задачу внедрения в «тяжёлую» онтологию средств объектно-ориентированного программирования, после чего реализовать принципы метапрограммирования [16]. Такой подход позволит выполнять простые расчеты и управлять механизмом наследования наглядно, что может служить отличным инструментом для обучения начинающих программистов.

Решение этих двух задач не позволит решить главную задачу применения редакторов онтологий в обучении и производстве в полной мере без дополнительных инструментов. Для этого, редактор онтологий с физической семантикой потребует каждый раз адаптировать для различных прикладных задач, согласно требуемой сигнатуре.

Рассмотрим предлагаемый редактор онтологий в обучении.

Редактор онтологий с физической семантикой как воркшоп метод в обучении

В эпоху пандемии рабочий процесс наряду с обучением переместился в дистанционный режим. В виду этого интерактивные средства дистанционного обучения получили резкий скачок в развитии и значительный прирост в посещении и использовании. Учителя получили возможность вести личные чаты с родителями, не дожидаясь пока ребёнок приведёт родителя в школу, а также вести всю или как минимум большую часть документации, касающейся учебного процесса. Эти же возможности стали доступны и педагогам в вузах, за исключением наличия курсовых проектов и работ, а также потребности хранить выполненные задания какое-то время, что требует обязательного наличия канала передачи файлов с выполненным заданием.

Программные средства, аналогичные Moodle, позволяют проводить полный жизненный цикл обеспечения обучающихся курсов: создание и регистрация участников, отправка, прием, получение и проверка задания, выставление оценок, а также обратная связь со всеми участниками текущего конкретного курса в случае несогласия с выставленной отметкой, -и всё это в одной программе. Конечно же, любая программная система подвержена тенденциям изменения и устаревания, в виду чего требует наличия минимальных знаний для всех участников и навыков адаптирования к изменениям от версии к версии программного продукта (новый формат хранения данных, новый канал или протокол передачи данных, новый тип шифрования канала передачи и хранения данных и другое).

Рассмотрим вопрос применения интерактивного обучения в высшей школе.

Образование в высшей школе на современном этапе требует обновления, наполнения новыми подходами к подаче изучаемого материала. Технология воркшоп как раз и есть той инновацией, которая в корне изменит весь ход проведения занятий на любой из их форм: лекции, семинарском или практическом занятии, либо на лабораторной работе [1].

Дефиниция «воркшоп» в переводе с английского обозначает «мастерская». Суть технологии состоит в активизации всех студентов, задействованных в учебном процессе, через самостоятельную работу в целом или её отдельный фрагмент, который очень важен в целостной структуре выполняемой работы.

Технология воркшопа состоит из таких методов:

— целостная – метод ретроспекции. Использование этого метода стимулирования познавательной активности обучаемого заключается в особенности преподнесения образовательного материала (на основе ранее изученного материала);

— синектика (мозговой штурм) – метод базируется на групповом решении поставленной проблемы. Применяется широко при преподнесении нового материала, когда дано основное понятие, но ещё не раскрыты все элементы изучаемого явления;

— тематическая – кейс-метод или метод анализа ситуаций. Предназначен для получения знаний по дисциплине, истина в которых плюралистична (сотрудничество студента и преподавателя);

— портфолио – метод, демонстрирующий достижения студента, его приобретенный опыт (ориентирован на личность студента);

— деловые игры – имитация реальности путем проживания ситуаций в роли и дальнейшем применении полученного опыта;

— метод проектов – письменный метод, с целью осмысления материала с разных сторон, в разных контекстах;

— «круглый стол» - метод обучает работе в группе, взаимодействию диады «студент-студент»;

— курсовая – метод развития критического, комбинаторно-логического мышления, радиантного мышления, метод эвристического обучения [2].

Технология воркшопа – это прямой путь к созданию ситуации успеха для студентов, значительному повышению их желания учиться. Искать и использовать новые знания, уметь разбираться в новых системах, а также адаптировать под требуемую предметную область.

Пробуждая интерес к учебе, повышаем и уровень самооценки, самостоятельности в коллективном деле, формируем аналитические, творческие, коммуникативные социальные навыки, расширяем границы практических компетенций, чтобы уметь действовать в новых, необычных ситуациях, сначала учебных, а в будущем – и в жизненных [3].

Следовательно, можно предположить, что при создании интерактивных обучающих систем педагогу или его помощнику не составит проблем разложить курс по составляющим, а затем после короткой беседы радоваться эффективному обучению будущих специалистов. На деле оказывается всё не совсем настолько радужно, а частенько даже наоборот. Информация чаще всего либо вырвана из контекста, что в реальных условиях требует значительной адаптации практически обесценивая полученные знания,

либо вовсе не связана с жизненными требованиями от специалистов текущего профиля на рабочих местах.

Решением этой проблемы призваны выступить программные комплексы относительно нового поколения, основанные на накоплении знаний по необходимой предметной области или концептуальной модели.

Редактор онтологий как САМ-система

Под термином «САМ-система» понимаются как сам процесс компьютеризированной подготовки производства, так и программно-вычислительные комплексы, используемые инженерами-технологами.

При использовании рассматриваемого редактора онтологий, на учебном занятии педагогу доступна возможность значительно сэкономить время и усилия на объяснение составления сложных геометрических фигур, какого-либо составного процесса с помощью постоянно обучаемой базы знаний.

Программный продукт позволяет автоматически генерировать и моделировать необходимую информацию по запросам, аналогично экспертным системам. Например, для рассмотрения некоего сложного шва на платье, относительно самого этого платья, потребовалось бы наличие всех необходимых составляющих, иначе данные получить невозможно. Текущая, условно, САМ онтологическая система способна задать пользователю серию наводящих вопросов для генерации необходимой отсутствующей информации, по которой получить искомое решение или нарисовать графическую фигуру. Это возможно благодаря реализации И-ИЛИ деревьев [17-19]. И-ИЛИ деревья – это один из подходов, используемый для представления пространства поиска решения задач технического творчества и искусственного интеллекта. Представление в виде И-ИЛИ дерева наиболее приемлемо для задач, которые естественным образом разбиваются на взаимонезависимые задачи, например, символическое интегрирование, доказательство теорем, игровые задачи и, в частности, задачи поиска технических решений.

Таким образом, этот тип программ значительно упрощает знакомство школьников с компьютерными технологиями и математикой позволяя каждому из обучаемых самостоятельно «пощупать» каждый из этапов всего необходимого процесса, не теряя возможности поиска необходимого решения и возможности задавать формальные запросы.

Редактор онтологий как CAD-система

CAD система – это автоматизированная система, реализующая информационную технологию выполнения функций проектирования.

В последнее время всё большую популярность набирают интерактивные программы, реализованные в виде видео игр, позволяющие обучающемуся программировать каждый шаг в реальном времени, либо из представленных возможностей, либо исходя из поставленной задачи. Используя подобную программу, пользователь имеет возможность использовать «умную базу данных», которая позволяет получать данные как в формальном, так и программном виде. Например, для обучения будущих программистов навыку понимания сложных алгоритмов, можно представить персонажа в некоем игровом пространстве, где запросы типа «get X±1» для получения координаты слева или справа, можно было бы заменить на «что находится сбоку», а используя диапазон значений – «что находится рядом». Этот же процесс можно эффективно применять для аппроксимации сложных функций, с целью получения графиков, отслеживания поведения сложных систем или получения иной информации.

Редактор онтологий как CAE-система

CAE-система – это общее название для программ и программных пакетов, предназначенных для решения различных инженерных задач: расчётов, анализа и симуляции физических процессов.

Программа подобного типа позволяет педагогу разбирать за учебное время практически любой физический процесс по полочкам, для демонстрации каждой или требуемой его составляющей в рамках темы занятия. Занятие по физике стали бы значительно эффективнее в плане запоминания для любого типа мышления обучающихся, а математические доказательства приобрели бы физический смысл. Например, для расчета нагрева трубы при течении пара под давлением, течение электрического тока по проводнику и т.п.

Онтологический инжиниринг, реализованный в рамках редактора онтологий, позволил бы подойти с обеих сторон физического процесса: формального и практического. Значительно упростил написание инструкций по сборке или, например, покраске изделия составного станка, для чего проектировщику достаточно получить по запросу «синяя деталь» размеры деталей, рассчитать их площадь и вычислить, какое количество синей краски

необходимо конечному пользователю для покраски готового изделия.

Онтологический инжиниринг в машиностроении (CALS-технологии)

Описанные возможности как средства CALS-технологии значительно облегчали бы программирование любой техники человеком. Причем без уточнения специфики, будь то студенческий стенд для научной работы или реально действующая военная техника.

Устройство, программируемое, к примеру, с помощью программных средств Arduino, могло бы запрашивать инструкции по составному типу, самостоятельно размечать фигуры на металлических листах для плазморезов и выполнять подобную работу более интеллектуально. Повсеместно параметры можно было бы получать формально и на родном человеку языке, по признаку или принадлежности, что позволило бы за считанные секунды составлять инструкции по сборке или ремонту, в которых требуется подробное описание всех составных частей.

Внедрение онтологического инжиниринга для сферы роботостроения на шаг бы приблизило человечество к очередной волне дискуссии о вероятной опасности слишком «умных» механических систем, по типу роботов. Само определение «робот», которому стали бы доступны накапливаемые с помощью нейросетей знания, структурируемые в базе знаний по описанию или признаку. Вызывает опасение в виду явной потери контроля над его развитием. Любое неуместное вмешательство со стороны «учителя» может повлиять на работу нейросети механической системы, которая в виду ограничений может не дополучить необходимые знания, а отсутствие ограничений может привести к непредсказуемым последствиям.

Выводы

Разнообразие предметных областей, для которых созданы онтологии по предложенной модели, доказывает универсальность этой модели, её способность описывать отношений понятий и базовые свойства, присутствующие в любой предметной области.

Таким образом, используя редактор онтологий как воркшоп или интерактивную технологию для обучения, педагог может применить множество формальных и практических методов по своему усмотрению для более наглядной демонстрации актуальной темы занятия.

Главной первоначальной задачей предлагаемого редактора «тяжёлых» онтологий с физической семантикой является приобретение обучающимися навыков разделения

поставленной задачи на составляющие её части с формальной и практической точек зрения вне зависимости от предметной области и приобретения практических навыков онтологического инжиниринга.

Принципиальным отличием предложенного подхода использования редактора онтологий является более приближенный метод использования онтологического инжиниринга в рамках объектно-ориентированного программирования.

В качестве перспективного направления данной работы отметим разработку методов и средств реализации подобных систем способных применяться в учебном процессе.

Литература

1. Приходченко, Е. И. Использование технологии воркшоп для повышения креативности будущих инженеров-педагогов / Е. И. Приходченко // журнал Информатика и кибернетика, 2020. - Вып. 2 (20) – С. 65-68
2. Приходченко, Е. И. Применение технологии воркшоп для повышения уровня динамических знаний будущих специалистов / Е. И. Приходченко // журнал Информатика и кибернетика, 2020. - Вып. 3 (21) – С. 62-65
3. Приходченко, Е. И. Технология воркшоп как динамическая система подготовки будущих специалистов / Е. И. Приходченко // журнал Информатика и кибернетика, 2020. - Вып. 4 (22) – С. 63-65
4. Шевченко, О. И. Технологии нестандартного обучения / О. И. Шевченко, М. А. Волков, В. А. Леонов // Педагогика высшей школы – Казань, 2018. – № 3 (13). – С. 17-25.
5. Воробьёв, Г. Г. Школа будущего начинается сегодня / Г. Г. Воробьёв. – М.: Просвещение, 2001. – 174 с.
6. Маштабиц, Е. И. Компьютеризация обучения: проблемы и перспективы / Е. И. Маштабиц, Н. Н. Моисеев. – М., 2013.-195 с.
7. Ярмухмедова, К. Г. Обучение иностранному языку по заочной форме с применением дистанционных технологий / К. Г. Ярмухмедова // Вестник Евразийской академии имени Д. А. Кунаева. – Алматы, 2015. – № 3. – С. 152-157.
8. Онтологический редактор Fluent Editor: учебно-методическое пособие к лабораторным работам / сост.: Н. М. Боргест, А. А. Орлова. – Самара: изд-во Самарского университета, 2017. – 44 с.
9. Онтологическая инженерия, философская онтология: проблемы и перспективы совместного развития / М. В. Заковоротная // Философия, этика, религиоведение, 2013.
10. Боргест, Н. М. Онтологии проектирования от Витрувия до Виттиха / Н. М.

Боргест // Онтология проектирования, 2018. – Т.8. - №4(30). – С. 487-522

11. Гаврилова, Т. А. Визуально-аналитическое мышление и интеллект-карты в онтологическом инжиниринге / Т. А. Гаврилова, Э. В. Страхович // Онтология проектирования, 2020. – Т. 10. - №1(35). - С. 87-99

12. Боргест, Н. М. Онтология проектирования: теоретические основы: учеб. пособие. - Самара: СГАУ, 2010. - 88 с.

13. Харман, Грэм. Объектно-ориентированная онтология: новая «теория всего»: пер. с англ. / Грэм Харман. – М.: Ад Маргинем Пресс, 2021. – 272 с.

14. Григорьев, А. В. Семантика модели предметной области для интеллектуальных САПР. Научные труды Донецкого государственного университета. Серия «Информатика, кибернетика и вычислительная техника», (ИКВТ-2000) выпуск 10. – Донецк: ДонГТУ, 2000. – С.148-154.

15. Онтологии и тезаурусы: модели, инструменты, приложения: учебное пособие / Б. В. Добров, В. В. Иванов, Н. В. Лукашевич, В. Д. Соловьев. – 3-е изд. (эл.) –

Москва: Национальный Открытый Университет «ИНТУИТ»; Ай Пи Ар Медиа, 2020. – 172 с.

16. Краснов, М. М. Применение метапрограммирования шаблонов C++ для решения вычислительных задач // научный сервис в сети Интернет: труды XIX Всероссийской научной конференции (18-23 сентября 2017 г., г. Новороссийск). – М.: ИПМ им. М.В. Келдыша, 2017. – С. 290-305

17. Кручинин, В. В. Использование деревьев И/ИЛИ для генерации вопросов и задач // Вестник Томского государственного университета, 2004. - №284. - С. 183 – 186.

18. Зорин, Ю. А. Интерпретатор языка построения генераторов тестовых заданий на основе деревьев И/ИЛИ // Доклады Томского государственного университета систем управления и радиоэлектроники, 2013. - №1. - С. 75 – 79.

19. Зорин, Ю. А. Использование алгоритмов комбинаторной генерации при построении генераторов тестовых заданий // Дистанционное и виртуальное обучение, 2013. - №6. - С. 54 – 59.

Филипишин Д.А., Григорьев А.В., Приходченко Е.И. Анализ применения редакторов онтологий с физической семантикой в педагогической деятельности вуза. Рассматриваются популярные методы и средства интерактивных обучающих систем, а также способы использования редактора онтологий как средства CASE- и CALS-технологии. Проводится сравнительный анализ на примере условно доработанного программного продукта до CAD/CAE системы, для описания эффективности обучения будущих программистов и разработчиков, с целью более глубокого понимания парадигмы объектно-ориентированного программирования. Принципиальным отличием предложенного подхода является более приближенный метод использования онтологического инжиниринга в рамках объектно-ориентированного программирования.

Ключевые слова: интерактивность, онтологии, редактор онтологий.

Filipishin D.A., Grigoriev A.V., Prikhodchenko K.I. Analysis of the use of ontology editors with physical semantics in the pedagogical activity of the university. Popular methods and tools of interactive learning systems are considered, as well as ways to use the ontology editor as a means of CASE- and CALS-technology. A comparative analysis is carried out on the example of a conditionally modified software product to a CAD/CAE system, to describe the effectiveness of training future programmers and developers, in order to better understand the paradigm of object-oriented programming. The principal difference of the proposed approach is a more approximate method of using ontological engineering in the framework of object-oriented programming.

Keywords: interactivity, ontologies, ontology editor.

Статья поступила в редакцию 14.05.2023
Рекомендована к публикации профессором Зори С.А.