

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

УДК 681.324.06

В.В. Краснопрошин, Г. Шаках, А.Н. Вальвачев

ИНТЕГРАЦИЯ РАСПРЕДЕЛЕННЫХ ЭКСПЕРТНЫХ ЗНАНИЙ:
ПРОБЛЕМЫ И РЕШЕНИЕ

Рассматриваются вопросы интеграции фрагментарных экспертных знаний фреймового типа для построения предметных областей. Предложены методы и средства анализа задачи, формирования онтологии, консолидации и публикации эксплицитных знаний. Дано описание соответствующего инструментария, приведен пример его использования для решения практической задачи.

Введение

Глобализация мировой экономики вызвала радикальную трансформацию среды существования человека и всех организованных им структур. Преобладавшие ранее материальные процессы заменяются информационными, знания признаны главным инструментом производства, совершенствуются стиль и методы управления [1]. Появляются компании с высокой степенью независимости подразделений, территориально распределенным персоналом, динамичными бизнес-процессами (так называемые «горизонтальные» компании) [2]. Во многих из них инновации, знание и обучение признаны парадигмой выживаемости, поэтому в литературе их называют «learning organizations» («обучающиеся организации», далее - компании). Этот тип компаний получил широкое распространение в области среднего и крупного бизнеса. Соответственно стал актуальным вопрос о поиске методов и технологий для получения новых знаний и эффективных способов их применения [3].

Используемые в современных задачах знания обладают свойствами актуальности, полноты и точности. Источники знаний разбросаны, как правило, по сотням исследовательских организаций в различных странах мира. Непосредственными носителями знаний являются эксперты. Природа из знаний может носить как эксплицитный (явный), так и имплицитный (скрытый) характер. В целом знания фрагментарны, имеют произвольную структуру и различный формат.

В этих условиях извлечение экспертных знаний, их консолидация, обработка и публикация являются достаточно серьезной проблемой, которая широко обсуждается в научной литературе [2 - 4].

Традиционно вопросы обработки знаний решаются с помощью информационных систем (ИС). В крупных компаниях для построения ИС применяют интеллектуальные инструментальные средства. Прежде всего, это CASE-технологии, реализованные в CASE-средах Erwin, Rational Rose, S-Designer и др. Они ориентированы на постоянные источники знаний и требуют использования реляционных баз данных. Высокая стоимость и технологическая сложность приводят к их нерентабельности в «горизонтальных» компаниях.

Технологии компонентно-базированного проектирования, в частности Microsoft .NET и SunSoft JavaBeans, эффективны при строительстве систем из компонентов одного типа (программных), поэтому использование информационных компонентов вызывает проблемы. Ротация источников знаний различной структуры требует корректировки и повторной компиляции программ, что постоянно увеличивает стоимость проекта.

Технологии добычи знаний data mining не отвечают требованиям актуальности, точности и полноты, что в принципе исключает их применение для решения данного класса задач.

В результате можно сделать вывод, что свойства знаний не соответствуют характеристикам имеющихся пакетов построения ИС. Требуются новые методы и средства, которые учитывают свойства знаний и специфику источников их происхождения.

В статье рассматривается один из подходов к решению проблем получения и обработки распределенных экспертных знаний на основе специализированных информационных систем KSIS (от англ. Knowledge Support Information System).

Представлены решения на основе композиционного подхода, главный принцип которого заключается в корректной композиции универсальных программных компонентов и независимых распределенных информационных компонентов. Этот подход включает в себя прямую и обратную фазы. Прямая фаза осуществляет анализ задачи, обратная фаза выполняет синтез релевантной предметной области (ПО). Полученные результаты позволяют говорить о полезности данного подхода при создании систем интеграции распределенных экспертных знаний.

1. Постановка задачи

Прежде всего дадим определения терминам, которые будем использовать в работе.

Знание будем трактовать как логически полный ограниченный набор сведений, необходимый для решения задачи.

ПО – это набор консолидированных знаний, необходимый и достаточный для решения задачи.

Публикация – способ представления ПО в форме целевой системы, оптимальной для решения задачи и соответствующей подготовке распределенного персонала.

KSIS – система программных и информационных компонентов, предназначенных для реализации всех этапов обработки знаний: от анализа задачи до публикации ПО.

Задача построения KSIS предполагает наличие требований к системе и исследование свойств среды функционирования целевой системы.

Требования определены списком операций, которые должна выполнять система:

- построение модели задачи и модели адекватной ПО;
- доставка модели ПО эксперту;
- формирование экспертом ПО;
- доставка ПО от эксперта разработчикам KSIS;
- публикация ПО.

Главные требования к разработке KSIS – компонентная архитектура и минимизация времени подготовки ПО, ее публикации и доставки результата персоналу.

Свойства среды функционирования KSIS заданы реалиями формирующейся в настоящее время глобальной информационной среды:

- KSIS получает знания из источников новейшей информации;
- источниками служат инновационные компании, исследовательские, учебные и научно-производственные организации и др.;
- источники территориально распределены, их природа динамична и изменчива;
- знания находятся у экспертов, в базах данных, выставлены в Интернете (в данной работе рассматривается только первый случай, так как автоматические методы добычи знаний из текста и баз данных находятся на стадии исследований);
- знания экспертов соответствуют фреймовой модели, представлены в документах произвольной структуры и формата.

Практическая реализация KSIS неизбежно встречается с проблемами методологического, технологического и психологического плана, перечислим лишь некоторые из них:

«проблема понимания» - крайне сложно найти и формализовать оптимальный способ объяснения удаленным экспертам, какие именно знания от них требуются;

«проблема посредника» - не отработаны универсальные методы построения динамического посредника для доставки модели задачи эксперту и инкапсуляции формализованных экспертных знаний в динамическую модель ПО;

«проблема консолидации» - методы консолидации распределенных знаний различной структуры и формата в единую ПО требуют комбинирования технологически сложных операций различных инструментальных систем, что само по себе является нетривиальной задачей;

«проблема публикации» - существующие технологии публикации ПО ориентированы на постоянные источники и жестко связаны с консервативными структурами баз данных, что усложняет процесс автоматизации построения целевой системы.

Задачу построения KSIS можно сформулировать следующим образом.

Пусть имеется задача P , решение которой требует новых знаний $Z(P)$. Знания носят фрагментарный характер и распределены в k источниках: $Z(P) = (Z^1(p), Z^2(p), \dots, Z^k(p))$. Источниками знаний служат эксперты. Знания $Z^i(p)$ представлены в виде документов произвольной структуры и произвольного формата. Требуется разработать методы и средства интеграции $Z^1(p), Z^2(p), \dots, Z^k(p)$ в ПО $Z(P)$ и публикации $Z(P)$.

В решении общей задачи можно выделить следующие этапы:

- 1) построение модели ПО $M(Z(P)) = f(M(Z^i(p)))$;
- 2) построение средства доставки модели $M(Z^i(p))$ эксперту и возврата ПО $Z^i(p)$;
- 3) консолидация фрагментов в ПО $Z(P) = f(Z^i(p))$;
- 4) публикация $Z(P)$.

Требования к решению: независимость программных и информационных компонентов, возможность их интеграции в процессе разработки KSIS.

2. Модель предметной области

В основе современных методов моделирования ПО лежит онтологический подход [3, 4]. Онтология – это модель ПО, использующая все доступные средства представления знаний, релевантных для данной области. Онтологическая модель содержит описание концептуально-логической модели ПО в терминах сущностей, отношений между ними и преобразования сущностей, которое выполняется в процессе решения задачи. Онтологический анализ начинается с декомпозиции задачи, основанной, как правило, на системном подходе.

Любая решаемая задача P может интерпретироваться как система во всей совокупности составляющих ее компонентов, связей и отношений, включая отношения с окружающей средой. Принцип системности предполагает разбиение сложной системы на более простые подсистемы, следовательно, моделирование начинается с декомпозиции задачи на эквивалентную совокупность более простых подзадач: $P = (P_1, P_2, \dots, P_m)$. В данном случае применима схема пространственной декомпозиции, которая формирует структуры подзадач, не связанных на одном уровне общими полями данных. Фактически это граф, корнем которого служит задача, узлами – подзадачи, дуги показывают отношения между ними. Выделив подзадачи максимально возможного уровня вложения, опишем каждую из вершин кортежем $P_i = (V_{ij}, T_{ij})$, где V_{ij} – термин-идентификатор подзадачи, T_{ij} – аннотация, содержащая определение термина, цель подзадачи, требования к решению и любую другую полезную информацию.

В результате итерационного процесса формируется модель задачи $M(P)$ в виде древовидной структуры, количество узлов, ребер и уровней которой зависит от полноты постановки задачи и накопленного опыта (рис. 1).

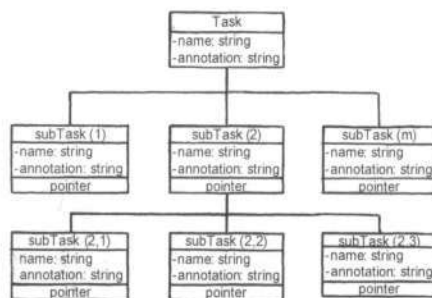


Рис. 1. Пример модели задачи

Таким образом, в модели $M(P)$ мы имеем словарь терминов, используемых при описании характеристик объектов и процессов, имеющих отношение к задаче (V); определения этих терминов (T) и классификацию логических взаимосвязей между терминами (pointer).

Описание модели $M(P)$ соответствует определению онтологии, что дает право использовать $M(P)$ в качестве предварительной модели онтологии, релевантной поставленной задаче.

Абстрактная модель $M(P)$, выраженная в терминах задачи, в конечном счете должна быть представлена в форме информационной модели $M'(P)$, которая будет обладать свойствами изоморфности, мобильности и динамичности (под динамичностью здесь понимается свойство инкапсуляции произвольного количества экспертных знаний). Создать такую модель можно с помощью существующих языков OIL, RDF и RDFS, OML, DAML+OIL. Однако для реализации модели $M'(P)$ они избыточны и технологически замкнуты.

В полной мере свойствам модели $M'(P)$ отвечает метаязык eXtensible Markup Language (XML), предложенный в рекомендациях консорциума W3C в 1998 г. XML - это не язык разметки, но на его основе можно создавать языки разметки для решения специфических задач. Реализуя эту возможность, авторы разработали проблемно-ориентированный язык SDP (Subject Domain Platform). Он учитывает требования к KSIS, свойства среды их функционирования и изначально предназначен для строительства ПО из независимых распределенных информационных компонентов, объединенных только логикой общей задачи.

На языке SDP корень графа $M(P)$ описывается тегом `<project>` с элементами `<title>`, `<description>`. Узлы любой степени вложенности с именами идентификаторов и аннотацией формирует рекурсивный тег `<item>`. Отличительной особенностью языка SDP является наличие тегов группы `<contentLocation>`, которые связывают модель $M'(P)$ с произвольно структурированными экспертными знаниями, представленными в виде документов различных типов. Значение параметров `<contentLocation>` может указывать как относительный, так и полный путь к файлу знаний или URL, в дополнении он может иметь атрибут `contentType`, описывающий характер данных.

Для быстрого создания моделей на языке SDP разработан редактор SPD Workshop. Он интегрирует независимые программные компоненты редактирования (SPDeditor), сегментации (SPDfr), генерации кода модели онтологии (SDPgo) и публикации ПО (SPDpubl) в виде целевой системы.

В качестве примера приведем сгенерированный редактором фрагмент кода модели онтологии «Экстренные хирургические операции», которая используется в системе повышения квалификации врачей:

```
<spd code/gff/h><version 1.0>
<project>
<title>Экстренные хирургические операции</title>
<description>Опыт 2003 г.</description>
<item>
<title>Хирургическое лечение острого аппендицита</title>
<description>Эксперт: проф. Конкин П.И. Сфера: перитонит. Пациенты: 50-75 лет. Пожелание: Объем материала: не ограничен. Срок: до 15 ноября 2003 г.</description>
<item>
<title>Описание АФТ-технологии</title>
<description>Текст любого объема, фотографии, анимация. Желателен аудиокомментарий.</description>
<contentLocation><путь к файлу знаний>
</item>
<item>
<title>Техника выполнения операции</title>
<description>Видеофильм. Желателен Ваш комментарий для каждого этапа.</description>
<contentLocation><путь к файлу знаний>
</item>
<item>
<title>Результаты применения АФТ-технологии.</title>
<description>Текст, диаграммы, аудиокментарий.</description>
```

<contentLocaton><путь к файлу знаний>
</item>

Фактически модель онтологии $M'(P)$ на языке SDP представляет собой формальное унифицированное описание задачи. Если в решении участвуют несколько экспертов, модель сегментируется на независимые фрагменты: $M'(P) = M'_1(p), M'_2(p), \dots, M'_n(p)$. Существенно, что каждый фрагмент онтологии имеет все необходимые опорные точки (элементы языка) для последующей обработки в процессе консолидации фрагментов в ПО $Z(P)$.

3. Средства доставки знаний

Назначение средств доставки - перемещение модели $M'(p)$ из компании к удаленному эксперту, формирование ПО и возвращение результата назад. Проблема здесь в том, что эксперт должен понять задачу, перевести требуемые от него знания из имплицитной формы в эксплицитную и «разместить» их в модели.

Авторы предлагают комплексное решение этих задач, в основе которого лежит модифицированная идея посредника [5]. Посредник – это программно-информационный компонент, состоящий из области идентификации SPDmail, области онтологии Model, области редактора онтологии Editor и области хранения экспертных знаний Knowledge (рис.2).

SPDmail(i)		
Model	Editor	Knowledge
$M^i(p)$	SPDeditor	$Z_i(p) = \emptyset$

Рис. 2. Структура посредника

Посредник формируется по команде редактора Export сразу после формирования модели и отсылается по электронной почте экспертам. При отправке область Knowledge пуста.

Получив SPDmail, эксперт запускает SPDeditor, который визуализирует только ту часть онтологии, которая содержит условие задачи, требования к решению и точки подключения файлов (параметры contentLocaton) с формализованными экспертными знаниями. Уяснив задачу, эксперт указывает в визуализированной части модели пути к требуемым материалам, тем самым инкапсулируя собственные знания в ПО $Z_i(p)$.

После окончания работы эксперту достаточно активизировать режим Export, чтобы посредник автоматически заполнил область Knowledge указанными файлами. Затем эксперт управляет SPDmail по электронной почте разработчикам KSIS.

Это простое и эффективное решение «проблемы понимания» и «проблемы посредника» апробировано авторами на десятках экспертов и в 90% случаев имело успех.

4. Консолидация фрагментарных знаний

Унифицированное представление доставленных посредниками информационных ресурсов требует решения проблемы консолидации [5]. В данном случае консолидация заключается в нормализации содержимого областей Knowledge неупорядоченного множества источников. Эта задача не имеет общего решения, так как посредники могут включать раритетные структуры в форматах неизвестного типа. В данной работе алгоритм консолидации основан на известных типах файлов, которые используются подавляющим большинством экспертов (SGML, HTML, DOC).

В качестве консолидирующего требуется формат, который позволяет визуализировать знания как в локальных программах, так и в Интернет-приложениях, поэтому авторы использовали формат HTML. Для специальных случаев предусмотрена консолидация в формате CHM. При формировании из $Z(P)$ программных модулей применяется компонент конвертора, кото-

рый формирует ПО без изменения оригинального формата, что предполагает дальнейшую обработку в соответствующих формату программных средах.

Идея алгоритма заключается в поэтапной обработке посредников, которая осуществляется программными компонентами интеграции, конвертации и консолидации (рис.3).

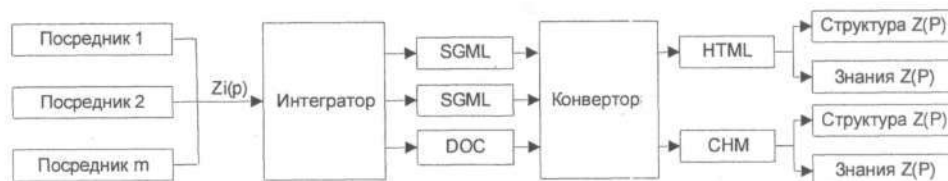


Рис. 3. Схема консолидации

Интегратор на основании значений параметров $\langle project \rangle \langle title \rangle$ создает упорядоченное множество посредников, относящихся к задаче P . Затем из области Knowledge каждого посредника он последовательно формирует ПО $Z(P)$ фреймового типа. Она полномасштабна, но неприменима по причине разнородности данных и отсутствия средств доступа к ним. Эту проблему решает конвертор, который преобразовывает файлы всех типов в консолидирующий формат HTML, не затрагивая оригинальные форматы файлов анимации, звукового сопровождения, видеофильмов и т.д. На стадии консолидации из ПО выделяются два информационных компонента: управляющая и содержательная части, которые будут служить исходным материалом для компонента публикации. Таким образом «проблема консолидации» решается для подавляющего большинства используемых экспертами типов файлов.

5. Публикация предметной области

Цель публикации – представление ПО в оптимальной для распространения и использования форме в минимальные сроки при минимальных затратах. Форма публикации зависит от специфики решаемой задачи. В большинстве случаев используются две формы: локальное приложение и сетевое. Предметные области с большим количеством видеоматериала, звука и анимации публикуются как локальные приложения на CD и рассылаются всем заинтересованным лицам. Если количество текста и графики относительно невелико, то публикуется web-приложение.

Традиционно публикацией занимаются инженер знаний и программист. Это дорогостоящий и долговременный процесс. Минимизация затрат требует отказа от стадии программирования и замены ее процессом автоматической генерации исполнительного кода приложения. Для решения этой проблемы авторы разработали компонент публикации (рис. 4), использующий в качестве входной информации результат работы конвертора (управляющую и содержательную части ПО).



Рис. 4. Схема публикации

В процессе публикации независимо от формы приложения компонент SPDpubl обрабатывает HTML- или CHL-составляющие ПО и формирует данные для навигационной части приложения (меню) и содержательной части (формализованные знания). Результат проецируется на один из двух типов программных компонентов, которые формируют EXE-файл локальной

программы или web-приложение. Существенно, что целевое приложение имеет большое количество сервисов.

6. Практика применения

Разработанные модели и алгоритмы объединены в единую технологию, с помощью которой был создан ряд прикладных систем в различных отраслях знаний [6]. Примером может служить система повышения квалификации хирургов, где были представлены новые технологии проведения операций.

Для формирования модели онтологии были выбраны семь часто проводимых операций, построены онтологии, сформированы посредники SPDmail и по предварительной договоренности разосланы по электронной почте семи экспертам. После консолидации ответов объем ПО составил 240 мегабайт, включая текст, слайды размером 1620x2800 и видеофрагменты. Большой объем данных определил форму публикации как локальное приложение. В результате была сгенерирована программа, которая имеет Explorer-подобный интерфейс, интуитивно понятную систему навигации и множество сервисов, вызываемых командами меню. Сервисы позволяют получить различные разрезы содержательной части, формируя тематические выборки материала. Например, по выборке ключевых слов из ПО можно выделить соответствующий текст, слайд-шоу или видеофильм, что позволяет взглянуть на проблему с совершенно разных сторон.

Инструментарий KSIS использовался также для консолидации библиотек программных компонентов, сформированных удаленными разработчиками на основании онтологии заказчика. В результате компиляции результата консолидированных ПО в среде Delphi был создан ряд прикладных систем, в частности Orthopedische Casu stiek и Atlas van de Paradontaal [6, 7].

Заключение

В статье описаны результаты решения проблемы интеграции распределенных экспертных знаний. В основу моделирования знаний положен онтологический подход, программная часть реализует корректную композицию программных и информационных компонентов.

Основные результаты работы сводятся к следующему:

- разработан метод построения динамических моделей ПО;
- решена проблема доставки модели ПО распределенным экспертам на основе универсальной программы-посредника;
- предложен способ формирования ПО на базе консолидированных экспертных знаний;
- разработан метод автоматизации процесса публикации ПО в форме локальных и web-приложений;
- описан инструментарий, реализующий разработанные методы;
- полученные результаты апробированы в процессе разработки ряда прикладных систем, нашедших достаточно широкое применение на практике.

За рамками статьи остался вопрос уточнения представленных экспертами материалов. Например, идеально созданная технология сборки автомобиля может «работать» в одной стране (положительное мнение независимых экспертов) и «не работать» в другой (отрицательное мнение фирм-продавцов машин). При решении реальных задач оперативное получение и сопряжение мнений двух «участвующих» сторон значительно повысит достоверность результата.

Список литературы

1. Skobelev P.O. Holonic Systems Simulation //Proc. of the 2nd International Conference «Complex Systems: Control and Modelling Problems», Samara, June 20-23, 2000. - P. 73-79.
2. Сенге П. Пятая дисциплина: искусство и практика самообучающейся организации: Пер. с англ. – М.: ЗАО «Олимп-Бизнес», 1999. – 408 с.

3. Jarrar M., Verlinden R., Meersman R. Ontology-based Customer Complaint Management// Proceedings of the workshop on regulatory ontologies and the modeling of complaint regulations. LNCS, November 2003. - V. 2889. - Springer. - P. 594-606.

4. Spyns P., Meersman R., Jarrar M. Data modelling versus Ontology engineering// SIGMOD Record: Special Issue on Semantic Web, Database Management and Information Systems. December 2002. - № 31(4). - P. 7-12.

5. Kalinichenko L., Missikoff M., Skvortsov N. Ontological Modeling //Proceedings of the 5th Russian Conference on Digital Libraries RCDL2003. St.Petersburg, Russia, 2003.

6. Krasnoproshin V., Obratsov V., Vissia H. Decision-Making by Precedence: Modeling, Technology and Applications // Proceedings of International Conference on Modeling and Simulation in Technical and Social Sciences (MS'2002), Girona, Spain, 25-27 June 2002. - P. 267-277.

7. Krasnoproshin V., Valvachev A., Vissia H. Unstructured Knowledge Synthesis for Decision-Making Problems // Proceedings of the Seventh International Conference PRIP'2003. - V.1. – Minsk, 2003. - P. 145-149.

Поступила 12.01.04

*Белорусский государственный университет
Минск, пр. Ф.Скорины, 4
e-mail: kras@bsu.by*

V.V. Krasnoproshin, G. Shakah, A.N. Valvachev

**INTEGRATION OF DESTRIIBUTED EXPERT KNOWLEDGE:
PROBLEMS AND SOLUTION**

Integration of fragmented frame based expert knowledge for constructing subject domains is under study. Methods and means of problem analysis, ontology formation, consolidation and publication of explicit knowledge are proposed. A description of the corresponding tools is given with an example of their use for solving a practical problem.