3. *Шпаковский Г. И., Серикова Н. В.* Программирование для многопроцессорных систем в стандарте MPI: Пособие Мн.: БГУ, 2002. 323 с.

ПРОБЛЕМА ИЗВЛЕЧЕНИЯ ЗНАНИЙ ДЛЯ СОЗДАНИЯ ТЕСТОВ В ОБУЧАЮЩИХ СИСТЕМАХ

Гази Шаках

Системы компьютерного обучения (СКО) применяются в органах государственного и местного управления, в системе высшего и среднего образования и на корпоративном уровне. Наибольшие темпы развития имеет корпоративное обучение, что вызвано резким увеличением зависимости эффективности работы предприятия от темпов и уровня обучения персонала. Важнейшей частью СКО являются модули тестирования, с помощью которых выполняется постоянный мониторинг персонала с целью выявления узких мест или определения реальных результатов переподготовки. До настоящего времени эти средства разрабатывают группы, состоящие из эксперта, инженера знаний и программиста с использованием дорогостоящих программных средств. Это долговременный, трудоемкий, дорогостоящий и психологически сложный процесс (Рис. 1), разработанный в 1990-х годах.

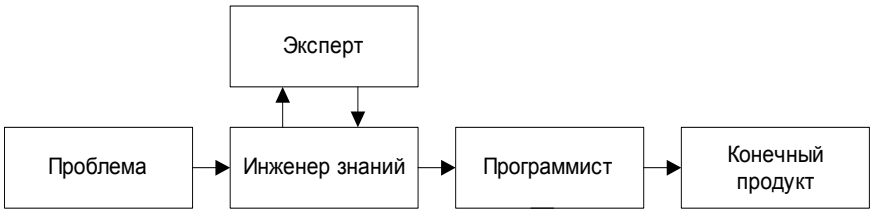

Рис.1. Традиционный процесс разработки тестовых систем

Характерные особенности этого процесса:

- сложность подбора психологически совместимой группы разработчиков;
- большие затраты времени и высокая стоимость разработки;
- необъективность тестов, так как эксперт, как правило, один;
- внести изменения в базу знаний или создать новую могут только разработчики.

В последние годы появилась потребность одновременного обучения и регулярного мониторинга уровня знаний посредством тестовых систем практически всего персонала корпораций часто в количестве десятков тысяч человек. Очевидно, что традиционный способ не соответствует новым условиям, так как за время создания теста тема может просто устареть. Требуется разработать новую технологию, которая обеспечит дос-

туп к распределенным в Интернет экспертам, извлечение, структуризацию и доставку их знаний, а так же автоматическое построение тестовых систем (Рис. 2).

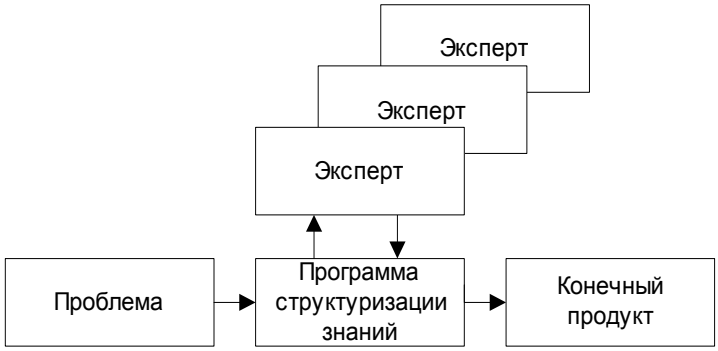

Рис. 2. Автоматизированный процесс разработки тестовых систем

В этом случае отпадает необходимость создания группы разработчиков, время создания и стоимость сокращаются до минимума, объективность тестирования гарантируется за счет работы с несколькими экспертами по одной и той же проблеме и быстрого изменения содержания тестов.

В данной работе представлен один из вариантов автоматизации извлечения экспертных знаний и их использования, основанный на многоагентной методике FIPA[1]. В контексте данной статьи под агентом будем понимать механизм инкапсуляции и обмена распределенными знаниями и функциями. Каждый агент – это процесс, обладающий определенной частью знаний об объекте проектирования и возможностью обмениваться этими знаниями с остальными агентами. В зависимости от типа, агент может поддерживать и интерфейс с пользователем. Под многоагентной системой будем понимать многокомпонентную систему, состоящую из агентов различного назначения. Соответственно решение общей задачи (рис. 2) будет состоять из комплексного решения двух подзадач:

- создание агента Knowledge, имитирующего инженера знаний,
- создание агента Builder, заменяющего деятельность программиста.

Пространство существования агентов может быть различным: локальный компьютер и Internet.

Работа агента основана на имитации деятельности инженера знаний, в результате которой формируется база данных, соответствующая конкретной предметной области. Эта задача относится к проблеме создания систем искусственного интеллекта и тесно связана с изучением человеческой системы переработки информации. Представление знаний основано на четырех моделях: логической, семантической, продукционной и фреймовой [2]. Поскольку традиционному процессу тестирования свой-

ственна иерархичность [3], целостный образ знаний можно сформировать в виде одной фреймовой системы, имеющей иерархическую структуру. В общем случае фрейм имеет следующий вид:

{<имя фрейма> <имя слота> <значение слота>…}

Согласно [4] модель тестовой системы должен включать слоты:

- текст вопроса;
- набор возможных ответов;
- показатель правильности ответов;
- тип интерфейса для презентации;
- обратная связь на уровне вопроса, которая представлена обучаемому независимо от ответа, и специфическая обратная связь для каждого из возможных ответов.

Очевидно, что такая структура соответствует традиционной схеме создания тестов (Рис. 1) и не позволяет решить поставленную задачу (Рис. 2), поэтому мы расширим список четырьмя новыми процедурами:

- процедура диалога с экспертом;
- тип вопроса (для формирования мультимедиа-вопросов из компонентов различного происхождения);
- процедура построения эталонов состояний обучаемого;
- процедура диалога с агентом компиляции тестовой программы.

Эти компоненты не противоречат общим требованиям к программам тестирования [4], так как реализуют служебные функции (доставка, организация диалога, подключение элементов вопроса, созданных в Dreamweaver и т. д.).

Для машинного представления такого рода структур обычно применяют объектно-ориентированные языки. Например, в системах Delphi и Jawa для поддержки сложных динамических структур введены объектные типы данных, с помощью которых одновременно описываются данные и операции над ними.

В данном случае агенты Knowledge и Builder были реализованы средствами языка Jawa. Агент Knowledge после доставки к эксперту, начинает с ним диалог, выполняет структуризацию его знаний по данной проблеме, заполняет слоты описанного выше фрейма и формирует соответствующую базу знаний. Затем он отсылает ее в центр тестирования и сообщает о новой базе знаний агенту Builder. Последний анализирует БЗ, на основе ее характеристик формирует интерфейс, объединяет в стандартной структуре всю необходимую для проведения тестирования информацию и формирует конечный продукт – готовую к эксплуатации тестовую программу Test (Рис. 3).

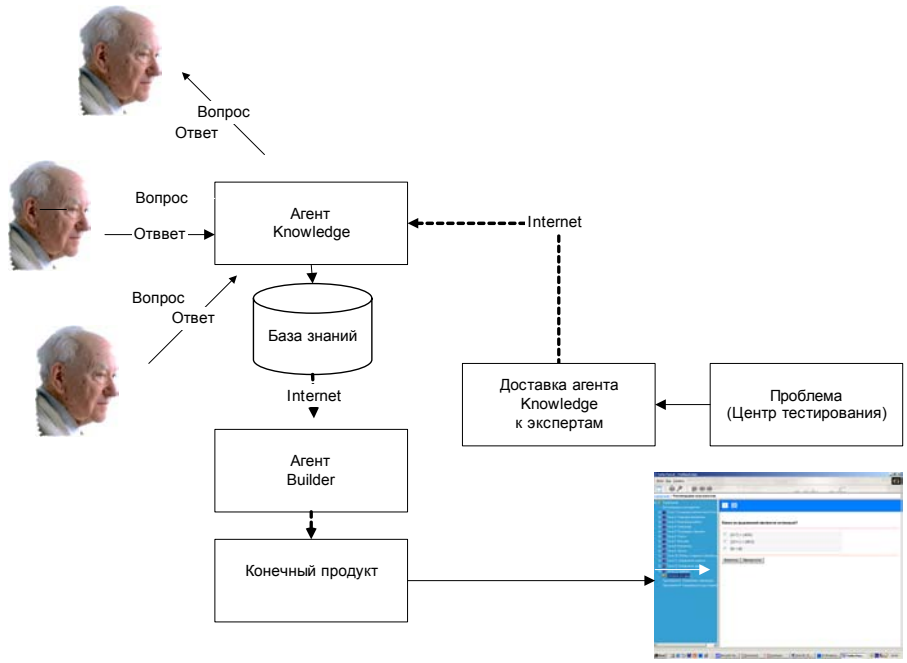

Рис. 3. Схема взаимодействия агентов

Выводы

Показаны преимущества фреймовой модели представления знаний и способа ее реализации средствами объектно-ориентированного программирования. Опробована возможность автоматизации процесса извлечения экспертных знаний в прямом диалоге программы-агента с экспертом, без участия инженера знаний и программиста. Применение многоагентной технологии позволило при необходимости разделить в пространстве процессы создания БЗ и ее эксплуатации, что теоретически позволяет использовать знания любого эксперта, доступного в Internet.

Литература

1. *Johnson W L & Shaw* E 1997 Using Agents to Overcome Deficiencies in Web-Based Courseware. 8th World Conference on Artificial Intelligence in Education, Kobe, Japan, 18 August 1997, pp. 70–77.
2. *Tschang F. T. and Senta T. Della* E 2001 Access to Knowledge: New Information Technologies and the Emergence of the Virtual University. Amsterdam: Elsevier Science and International Association of Universities, 167–206.
3. *Synnes K, Lachapelle S, Parnes P & Schefström* D 1998 Distributed education using the mStar environment. Journal of Universal Computer Science 4 (10): 803–823.
4. *Specht M. & Oppermann R.* 1999 ACE – Adaptive Courseware Environment. The New Review of Hypermedia and Multimedia 4: 141–161.