

УДК 004.582

А.С. ПРИГОЖЕВ

Одесский национальный политехнический университет, Украина

ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ ПРИОБРЕТЕНИЯ ЗНАНИЙ ДЛЯ ЭКСПЕРТНОЙ СИСТЕМЫ ПОДДЕРЖКИ ПОЛЬЗОВАТЕЛЯ НА ОСНОВЕ АЛГОРИТМИЧЕСКИХ АЛГЕБР

Статья посвящена построению информационной технологии приобретения знаний для экспертной системы поддержки пользователя на основе алгоритмических алгебр. Рассматриваются вопросы синтеза базы знаний на основе схемы алгоритма. Использование не интерпретированных схем алгоритмической алгебры Дейкстры позволяет настраивать базу знаний на любую решаемую пользователем задачу.

экспертная система, система приобретения знаний, алгоритмическая алгебра, алгебра Дейкстры, алгебра сценариев, сценарии

Введение

Важной составляющей процесса решения задачи с использованием автоматизированных информационных систем (АИС) является правильное управление ею со стороны пользователя. Исторически, первым способом управления АИС являлся командный интерфейс, реализованный в виде пакетной технологии и технологии командной строки [1]. При пакетной технологии пользователь вводит команды в систему в виде пакета, после обработки которого выдаются результаты работы АИС. Недостаток пакетной технологии – сложность обнаружения ошибок в пакете. Для устранения этого недостатка был разработан диалоговый режим взаимодействия пользователя и АИС, при котором пользователь вводит команды последовательно, а система реагирует на них сразу после ввода. Использование командной строки требовало от пользователя знания форматов команд АИС до их использования, что приводило к необходимости написания текстовых руководств по работе с АИС. Однако содержание текста может быть неоднозначно воспринято пользователями и разработчиками системы, что является недостатком такой системы поддержки.

Для решения указанной проблемы в 1973 году фирмой Хегох была разработана графический интерфейс. В своём развитии данный тип интерфейса прошёл две основные стадии - Графический интерфейс

прошёл в своём развитии две основные стадии: простого графического интерфейса и интерфейса WIMP (Windows-Icon-Menu-Pointer), или «чистого» графического интерфейса. На первом этапе развития основными отличиями графического интерфейса были: цветовое выделение определённых символов, возможности изменения размера курсора, зависимость реакции на нажатие клавиши от положения курсора. Как аппаратные средства интерфейса всё чаще стали использоваться манипуляторы типа «мышь», трекбол, которые позволяли быстро перемещать курсор в нужную часть экрана.

На втором этапе развития графического интерфейса появился WIMP-интерфейс (Windows-Icon-Menu-Pointer interface). Взаимодействие с использованием «чистого» WIMP-интерфейса основано на окнах и меню. В терминологии WIMP-интерфейса каждый выполняющийся в системе процесс ассоциируется с окном. Команды АИС выполняются при помощи меню, а данные вводятся посредством текстовых полей. Взаимодействие пользователя на основе «чистого» WIMP-интерфейса основано на объектно-ориентированной методологии. Каждый элемент интерфейса является объектом с некоторыми свойствами. Основными объектами графического интерфейса являются пиктограмма, окно, элемент меню или кнопка, элемент ввода данных. В WIMP-интерфейсе пиктограмма символизирует некий неактивный в настоящее время объект

вычислительной системы: неактивную программу, не открытый файл с данными и т.п. Окно WIMP-интерфейса представляет собой выполняющуюся в данный момент программу или активный объект вычислительной системы. В различных системах с WIMP-интерфейсом в виде окон представляется активная на данный момент в вычислительной системе программа, содержимое жёсткого диска, состояние некоторого устройства и т.д. Такая форма представления частично решила проблему поддержки пользователя, связанную с необходимостью запоминания форматов команд АИС, т.к. теперь параметры команд вводились в некоторую форму, а выполнялись нажатием кнопки.

Однако оба представленных интерфейса обладали существенным недостатком – они не позволяли обеспечить поддержку пользователя при решении определённой задачи с конкретными параметрами, а также обучить методам решения задачи. Поэтому возникает задача интеллектуализации поддержки пользователя при работе с программной системой.

К современным средствам интеллектуализации пользовательских интерфейсов относятся различные типы адаптивных интерфейсов [2], а также технология Microsoft Intellisense [3].

Адаптация интерфейса к пользователю происходит на основании формализации характеристик пользователя и моделирования его деятельности в системе. В [2] предлагается строить модель пользователя для адаптивных систем на основе набора демографических, профессиональных, психофизиологических, психологических характеристик и на основании которых происходит адаптация интерфейса к пользователю. Недостаток – использование ряда качественных, трудно формализуемых параметров, например, «понимание», что делает адаптацию не всегда эффективной.

Поэтому необходимо дальнейшее развитие указанных средств интеллектуализации, в частности применение экспертных систем для поддержки пользователя [4, 5]. Для таких систем является актуальным построение систем приобретения знаний для экспертной сис-

темы поддержки. В данной статье рассматриваются основные принципы, положенные в основу реализации такой системы.

1. Приобретение знаний на основе алгоритма решения задачи

В процессе работы в АИС пользователь решает определённые задачи. На основе проведенного анализа руководств пользователя к различным программным системам, можно утверждать, что действия пользователя в системе могут быть описаны при помощи некоторого алгоритма. В работе [7] описаны алгоритмические стратегии поведения пользователя, и доказано, что с их помощью можно описать любой алгоритм работы в АИС. Рассмотрим далее построение формальных методов синтеза базы знаний с использованием предложенных стратегий.

Входными данными для алгоритма синтеза базы знаний будет являться блок-схема алгоритма, вводимая инженером по знаниям в систему управления базой знаний (СУБЗ). Введём некоторые обозначения. Пусть V – множество вершин схемы алгоритма. В данном подмножестве можно выделить несколько подмножеств.

Подмножество терминальных вершин T , состоящее из двух элементов – вершины начала и вершины конца алгоритма. t_n и t_f .

Подмножество условных вершин A , элементы которого пронумерованы.

Подмножество циклических вершин L , элементы которого пронумерованы.

Подмножество операторных вершин C , элементы которого пронумерованы.

Подмножество вершин «предопределённый процесс» P , элементы которого также пронумерованы. Вершины сопряжения S , к которым относятся те вершины, в которых сходится несколько дуг.

Исходя из приведенных определений, можно утверждать, что для множества T справедливо равенство:

$$T = A \cup L \cup C \cup P \cup S$$

Элементы множества T пронумеруем в порядке

их следования в блок схеме алгоритма. В случае условного оператора вначале нумеруются вершины, которые

Любую структурную схему из алгебры Дейкстры можно представить в виде блок-схемы алгоритма. Обратное утверждение не верно [8]. Поэтому необходимо реализовать проверку на возможность представления блок-схемы в виде некоторой структурной схемы алгебры Дейкстры. Для этого необходимо ввести ограничение на блок-схемы с целью точного их представления структурными схемами. Это ограничение формулируется следующим образом: ни одна из ветвей условной вершины не может выходить за пределы вершин, ограничивающих цикл.

Также в качестве ограничений введём возможность использования исключительно условных вершин при проектировании графа. Тогда для обнаружения циклов в граф схеме можно использовать известные алгоритмы для поиска циклов в графе. Очевидно, что в найденных циклах будут присутствовать одна условная вершина. Для обеспечения однозначности представления граф схемы в виде структурной схемы алгебры Дейкстры будем считать, что в цикле не может быть более одной условной вершины.

С учётом всех приведенных ограничений можно сформулировать следующий алгоритм формирования структурной схемы по блок схеме алгоритма. В приведенном ниже алгоритме i – индекс вершины в граф схеме алгоритма, a – вершина блок схемы алгоритма

1. Поиск циклов в граф схеме алгоритма

2. $i = 0$. Добавить к записи структурной схемы алгебры Дейкстры символ «{»

3. $i = i + 1$. Проверяем тип вершины:

3.1. Если a_i - вершина, не входящая ни в один из циклов, то:

3.1.1. Если a_i - операторная вершина, то необходимо добавить в структурную схему операторную переменную.

3.1.2. Если a_i - условная вершина, то за-

помнить её и занести в список условных вершин. Пройти ребро соответствующее истинной вершине и ребро соответствующее ложной вершине. В полученных списках удалить общие вершины и их добавить в условный оператор в качестве операторных переменных. Если в результате удаления общих вершин один из списков окажется пустым, то заменить его на тождественный оператор E .

3.2. Если a_i - вершина, входящая в ранее обнаруженный цикл, то необходимо определить тип цикла исходя из типа текущей вершины: если вершина – условная, то цикл с предусловием, иначе – с постусловием.

4. Если вершина a_i - терминальная, то заканчиваем алгоритм, иначе переход к п. 3.

Результатом работы данного алгоритма будет структурная схема алгебры Дейкстры, которая далее может быть преобразована в структурную схему алгебры сценариев, из которой возможен синтез базы знаний.

Правила, по которым структурная схема алгебры сценариев и база знаний могут быть получены, используя правила и соотношения, приведенные в [7]. Так как база знаний строится на основе функционально полной системы алгебры сценариев, то в базе знаний также можно представить любой алгоритм работы пользователя

2. Информационная технология для приобретения знаний

Информационная технология включает в себя подготовку и хранение базы знаний о задачах на основе графа. Он включает в себя следующие шаги:

1. Формулировка условий задач. На этом этапе составляются словесные описания задач, решаемых в АИС. Эти описания могут создаваться разработчиком базы знаний из инструкции пользователя системы либо по результатам диалога с пользователями АИС.

2. Разработка алгоритма решения задачи. Разработчик базы знаний, на основе собственного опыта работы с данной АИС, описывает порядок действий

при решении задачи с помощью граф-схемы алгоритма решения.

3. Формальные преобразования алгоритма решения из граф-схемы к структурным схемам алгебры Дейкстры. Этот этап может быть выполнен разработчиком базы знаний вручную либо автоматизировано с помощью соответствующих расширений СУБЗ. Для этого используются преобразования граф-схем в схемы алгебры Дейкстры из [7].

4. Формальные преобразования структурных схем алгоритмов в алгебре Дейкстры к структурным схемам алгебры сценариев, Данный этап может выполняться вручную либо автоматизировано. Для этого в СУБЗ предусматривается транслятор выражений в алгебре Дейкстры в алгебру сценариев.

5. Преобразование структурных схем алгебры сценариев к представлению на основе графа. Данный этап, также как и предыдущий, может выполняться вручную или автоматизировано, используя при этом приведенное в [7]. представление структурных схем алгебры сценариев в виде графа задач.

6. Редактирование и проверка корректности базы знаний. Осуществляется с использованием СУБЗ разработчиком базы знаний.

Заключение

Применение предложенной информационной технологии позволяет автоматизировано синтезировать базу знаний для экспертной системы поддержки пользователя на основе алгоритма. Это позволяет представить знания о решаемых задачах в более формализованном виде на основе схемы алгоритма. Это позволяет упростить процесс подготовки знаний для экспертной системы поддержки пользователя.

Литератуца

1. Денисов Ю.А. Операционные системы: правила работы [Электронный ресурс]. – Режим доступа: http://www.citforum.ru/operating_systems/ois2/index.html
2. Ходаков В.Е., Бень А.П., Дмитриченко В.М. Гипермедиа модель данных, адаптируемая к конечному пользователю // Вестник ХГТУ. – 1999. – № 1 (5). – С. 19–22
3. Клепнин В.Б. Visual FoxPro 9.0. – СПб.: ВHV, 2007. – 1200 с.
4. Рувинская В.М., Пригожев А.С. Разработка планирующей экспертной системы для поддержки работы пользователя с программной системой // Вестник ХГТУ. – 2005. – № 3. – С. 133-137
5. Пригожев А.С. Реализация решателя и базы знаний экспертной системы для планирования действий пользователя при разработке программ // Радиозлектроника и информатика. – 2005. – № 4. – С.110-116.
6. Пригожев А.С. Особенности разработки и применения планирующей экспертной системы для обучения // Искусственный интеллект. – 2005. – № 3. – С. 529 – 540.
7. Пригожев А.С. Информационная технология помощи пользователю // Холодильная техника и технология. – 2007. – № 2. – С. 98-101.
8. Цейтлин Г.Е. Введение в алгоритмику. – К.: Сфера, 1998. – 310 с.

Поступила в редакцию 6.02.2008

Рецензент: д-р техн. наук, проф. Е.В. Малахов, Одесский национальный политехнический университет, Одесса.