



ГОСУДАРСТВЕННЫЙ НАУЧНЫЙ ЦЕНТР РОССИЙСКОЙ ФЕДЕРАЦИИ
ИНСТИТУТ ФИЗИКИ ВЫСОКИХ ЭНЕРГИЙ

ИФВЭ 98-38
ОАПнЭС

Т.В. Егошина

**ФОРМИРОВАНИЕ ЗНАНИЙ
О ТЕХНОЛОГИИ ОБРАБОТКИ ДАННЫХ
В ИНФОРМАЦИОННЫХ СИСТЕМАХ**

Направлено в *“Приборы и системы управления”*

Протвино 1998

Аннотация

Егошина Т.В. Формирование знаний о технологии обработки данных в информационных системах: Препринт ИФВЭ 98-38. – Протвино, 1998. – 9 с., 3 рис., библиогр.: 6.

Описывается метод формализованного отображения в ЭВМ профессиональных знаний разработчиков информационных систем (ИС). Предлагается способ автоматизированного формирования знаний экспертов для анализа работы ИС и принятия решений на этапе эксплуатации системы.

Abstract

Egoshina T.V. Knowledge Formation on Data Handling Technology in Information Systems: IHEP Preprint 98-38. – Protvino, 1998. – p. 9, figs. 3, refs.: 6.

The method of formalized translation of professional knowledge of information systems (IS) developers in computers is described. A way of automatic experts' knowledge formation to analyze the IS work and decision making during system service is proposed.

Введение

Работа современных информационных систем (ИС), представляющих собой комплекс программных, информационных и технических средств, связана с множеством процессов сбора, хранения, обработки, передачи и использования данных. Действия человека, участвующего в работе такой информационной системы, требуют знания не просто перечня входных и выходных документов, программных модулей, таблиц или файлов базы данных (БД), но и знания внутренних взаимосвязей всех системных объектов. Иными словами, от человека требуется знание и соблюдение определенной технологии работы ИС. В связи с этим возникает необходимость решения, по крайней мере, двух проблем: 1) автоматизированного построения описания технологических процессов обработки данных, 2) использования полученного описания для анализа работы ИС и принятия решений на этапе эксплуатации системы [1].

Хотя в последнее время на рынке инструментальных средств появились мощные программные продукты (*Delphi*, *Oracle CASE* и др.), в силу экономического фактора немногие предприятия могут позволить себе выполнять разработку информационных систем с использованием *Oracle CASE* и СУБД *Oracle*. Кроме того, на предприятиях и в организациях до сих пор функционирует много информационных систем, разработанных в среде desktop СУБД, таких как *Clipper*, *FoxBase*, *FoxPro*, *Access* и др. на ПЭВМ с процессорами 286/386/486. Эффективность использования информационной системы зависит от того, насколько оперативно удастся переделать, заменить или адаптировать систему к новым условиям работы. Возможным путем решения проблемы является формализованное отображение в ЭВМ профессиональных знаний разработчиков и автоматизация формирования знаний экспертов.

1. Построение описания технологических процессов в информационных системах

Одной из основных проблем определенного круга работающих ИС является понимание и знание информационных потоков в системе. Построение описаний путей прохождения данных в работающих ИС можно рассматривать и как составление инструкций для персонала, участвующего в эксплуатации информационной системы или получающего результаты ее работы. Под технологией работы информационной системы будем понимать взаимосвязанную по входам и выходам последовательность технологических операций обработки данных, выполнение которых приводит к достижению требуемого результата — бесбойному функционированию ИС. В результате выполнения некоторой технологической операции могут быть получены компоненты, являющиеся промежуточными, используемыми в качестве входа следующей технологической операции.

Любая сложная информационная система содержит несколько точек ввода и вывода информации из ЭВМ. И, конечно, ЭВМ или комплекс ЭВМ является центральным звеном организации технологии информационных процессов. Кроме того, любая сложная ИС — это совокупность информационно-согласованных подсистем. Информационная согласованность означает, что работа подсистем обеспечивается информационно-согласованными программами, т.е. результаты работы одних программ будут исходными данными для других.

Технологический информационный процесс определяет действия, их последовательность, исполнителей, программы, средства и ресурсы, необходимые для выполнения этих действий.

2. Разбиение информационных систем на технологические компоненты

Для информационных систем важно знать, какие данные откуда берутся, куда попадают и с помощью чего. Поэтому разбиение информационных систем на технологические компоненты происходит по пути следования данных. Основная задача при этом состоит в выделении стабильных составляющих информационной системы, разбиении технологии работы системы на компоненты, их объявлении и описании информационных связей компонентов с помощью правил.

Изучение большого круга информационных систем, работающих на основе desktop СУБД, позволило выявить несколько классов объектов. Каждый класс в свою очередь допускает несколько типов объектов. Понятие *типа* введено для объектов, сходных по описанию, и в тип вынесен перечень основных правил. Таким образом, с типом связано описание обобщенных правил. Метод обработки объектов одного класса не зависит от типа. А это значит, что *класс* определяет фундаментальный метод работы с объектами нескольких типов. Множество, объединяющее близко расположенные объекты разных типов с одним и тем же именем, но из разных классов, будем называть *кластером*. Кластеры имеют большое значение

при выборе имени нового объекта в правилах перехода. При обобщенном описании правил возникает проблема именования следующего объекта, имя которого нельзя назвать точно, иначе не получится обобщенного правила в типе. Поэтому новое имя объекта образуется по умолчанию на основании имени кластера.

Можно назвать, например, такие классы объектов: класс объектов “Программы” разных типов, класс объектов “Базы данных” разных типов, класс объектов “Рабочие места” разных типов, класс объектов “Принтеры” разных типов и др. Причем класс “Программы” включает в себя не только программные модули, но экранные формы и отчеты.

3. Взаимосвязь компонентов информационных систем

Как уже указывалось, в информационных системах основное связывание объектов происходит по пути движения данных.

При графическом изображении связи каждый объект будет представляться узлом, а ссылки между объектами — дугами, направленными от начального объекта к конечному.

Граф на рис. 1 показывает направление движения информации от первого объекта (object_1) ко второму (object_2). Запишем переход информации из object_1 в object_2 в виде

$$\text{object}_1 \longrightarrow \text{object}_2$$

Тогда для каждого класса объектов графически можно изобразить все этапы обработки данных в информационной системе.



Рис. 1. Граф связи объектов.

Объекты — это конкретные компоненты технологии работы системы. Правила — некие состояния информации во время ее движения от объекта к объекту. Переход из одного состояния в другое можно записать в виде

$$r_1 \longrightarrow o.r_2$$

Здесь o — объект, r_1 и r_2 — правила. При отсутствии r_2 в правой части выражения считается, что состояние информации не меняется; при отсутствии o в правой части происходит движение информации в рамках одного объекта.

При переходе информации в конкретный кластер немаловажным становится однозначное определение имени объекта. Имя объекта может быть составным `typename$objectname`, включающим указание типа объекта `typename` и собственное имя объекта `objectname`. Связи, описанные для определенного типа объектов одного класса, верны для любого объекта данного типа из этого класса.

Предложенный способ записи правил перехода служит отображением структурных и динамических свойств объекта автоматизации.

4. Возможные условия решения проблемы

Итогом разработки средств формирования технологических процессов обработки данных должны стать инструментальные средства для администратора информационной системы, с помощью которых он может получать сведения и осуществлять анализ структуры информационной базы и технологии работы системы. В качестве конечного программного продукта, с которым предстоит работать администратору, выступает интерпретатор запросов, осуществляющий поиск ответа на запрос в некотором описании информационной системы. Приведем примеры возможных запросов администратора в неформализованном виде: в какую таблицу БД заносится информация из заданного входного документа? Или, каков путь (трасса) следования порции информации из конкретной таблицы БД?

Для создания базы данных интерпретатора — описания структуры информационной системы и технологии ее работы, которую можно назвать технологической базой знаний информационной системы, — необходимо разработать язык описания технологических процессов обработки информации и компилятор ЯОТП, позволяющий преобразовывать формализованное описание работы информационной системы в объектную программу для построения сети фреймов [2].

На рис. 2 приводится схема работы инструментальных средств формирования технологических процессов обработки данных в ИС.



Рис. 2. Схема работы средств формирования технологических процессов обработки данных.

5. Уточненная постановка задачи

Итак, создание инструментальных средств формирования знаний о технологии обработки данных в информационных системах включает:

1. Разработку грамматики, синтаксиса и компилятора ЯОТП.
2. Создание интерпретатора запросов пользователей.

6. Язык описания технологических процессов обработки информации

При разработке языка описания технологических процессов обработки информации учитывалось, что он должен быть удобен и требовал бы при его использовании минимальных усилий не только от квалифицированных программистов, но и от конечных пользователей. С другой стороны, ЯОТП должен быть прост для реализации на стандартных средствах ЭВМ и не должен предъявлять жестких требований к другим ресурсам вычислительных систем.

Грамматика ЯОТП относится к классу контекстно-свободных грамматик (КС-грамматик) и формализует большую часть правил, предназначенных для описания синтаксической структуры языка.

Грамматика ЯОТП включает следующие виды операторов:

1. *Операторы описания типа объекта*

TYPE <идентификатор типа> [; комментарий]

2. *Операторы описания объекта*

ОБЪЕКТ <имя объекта> TYPE <идентификатор типа>
[REF <идентификатор типа>] [; комментарий]

3. *Операторы перехода*

<имя правила> —> <имя нового правила> [; комментарий]

Ключевые слова TYPE, ОБЪЕКТ, REF (ссылка на другой тип объекта) являются зарезервированными и не могут быть использованы в качестве имен. В идентификаторах типов и объектов, в именах правил могут использоваться любые буквы русского (что, безусловно, удобно для русскоязычных пользователей) и латинского алфавитов, цифры и специальные символы (–, _). Знак —> (“стрелка”) используется в записи правил для обозначения перехода информации из одного состояния в другое.

7. Компилятор ЯОТП

В основе всех процессов обработки языков лежит теория автоматов. Процесс компиляции рассматривается как взаимодействие небольших процессов. Согласно

упрощенной модели компиляция осуществляется тремя последовательно соединенными блоками: лексическим блоком, синтаксическим блоком и генератором выхода. Эти три блока имеют доступ к общему набору таблиц, куда можно помещать долговременную или глобальную информацию о программе. Лексический блок разбивает цепочку символов на слова (лексемы), из которых она состоит. Синтаксический блок переводит последовательность лексем, построенную лексическим блоком, в последовательность атомов, которая более непосредственно отражает порядок выполнения операций.

Немаловажным вопросом при построении компилятора является определение, что компилятор должен выдавать в качестве выхода. Поэтому на первом этапе построения компилятора должен быть получен ответ на этот вопрос. Так как для получения сведений по технологии информационной системы был выбран фреймовый метод представления знаний, то на выходе компилятора ЯОТП будут получены функции языка LISP и процедуры языка FRL (Frame Representation Language) [3,4] для генерации фреймов, содержащих описание технологии функционирования ИС.

Разработка компилятора на языке С позволила отказаться от жестких требований к вычислительной технике. При построении компилятора ЯОТП использованы сервисные программы построения компиляторов, такие как генератор синтаксических анализаторов BISON — прототип сервисной программы YACC [5], и генератор программ лексического анализа FLEX — прототип генератора лексических анализаторов LEX [6].

Компилятор ЯОТП реализует так называемый LALR(1)–разбор, являющийся модификацией одного из основных методов разбора “снизу вверх” — LR(k)–разбора.

8. Интерпретатор запросов

Следующим этапом автоматизированного формирования технологических процессов обработки данных в информационных системах является использование администратором ИС скомпилированной ЯОТП–программы для построения сети фреймов и получение с помощью интерпретатора сведений по структуре информационной базы и технологии работы ИС.

Анализ существующих программных средств показал, что в наибольшей степени для решения задачи получения ответов на запросы подходят методы представления и извлечения знаний, используемые в интеллектуальных системах. Был выбран фреймовый подход, связанный с попытками совместить строгость формальных систем с простотой и гибкостью семантических сетей.

Наиболее известным, развитым и доступным средством оказался интерпретатор языка FRL, который является открытой системой с точки зрения механизмов обработки объектов, что позволяет при переходе от одной проблемной области к другой дополнять его необходимыми процедурами. Язык FRL реализован как библиотека над языком LISP. В FRL введено понятие поименованного списка с ассоциативным доступом — списка, к которому возможен доступ по первому элементу — имени *фрейма*. Все подструктуры фрейма представлены такими списками, вложенными

друг в друга. Соответствующие подструктуры фрейма называются: *слот*, *аспект*, *данное*, *комментарий*, *сообщение*.

Модель технологии работающей информационной системы образована системой фреймов и правил перехода. Рассматриваемая предметная область представляется сетью фреймов объектов — компонентов. Для каждого объекта справедливы правила перехода

$$r_1 \longrightarrow o.r_2$$

Здесь $o(object)$ соответствует фрейму, а $r_1(ule)$ и $r_2(ule)$ — слотам фреймов. Если в выражении отсутствует правое o , то используется имя текущего фрейма. Если в записи выражения отсутствует r_2 , то используется имя слота, соответствующего значению r_1 , причем r_1 в выражении относится к текущему фрейму.

На рис. 3 схематично показано соответствие между объектами и слотами фреймов.

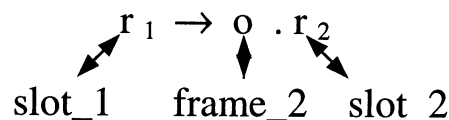


Рис. 3. Соответствие между объектами и слотами фреймов.

Полученное после компиляции описание технологии информационной системы используется для построения сети фреймов, которое представляет собой граф, образованный наложением сети фреймов объектов-компонентов, связей группирования и правил перехода. Соответственно вершины сети могут отображать объект, тип или правило.

Сеть фреймов обеспечивает накопление и использование знаний и опыта разработчиков информационной системы, выступающих в качестве экспертов. Сети на фреймах в FRL реализуются посредством указания имен фреймов в качестве значений терминалов. В процессоре FRL имеется ряд встроенных механизмов, называемых механизмами наследования, обеспечивающих работу с сетью. В системе FRL возможно несколько типов наследования:

- наследование по умолчанию, когда запрос на данные из аспекта \$VALUE дополняется запросом в аспект \$DEFAULT;
- наследование по АКО-иерархии, когда запрос автоматически распространяется на фреймы, лежащие выше по АКО-иерархии;
- косвенное наследование, при котором запрос дополняется данными из специально указанного фрейма.

Текущая ситуация запроса описывается подграфом активных в настоящий момент узлов и связей сети фреймов.

Основные типы запросов следующие:

1. Получить конечную точку (или конечное значение) следования заданной порции информации.
2. Получить путь (трассу) следования заданной порции информации.
3. Выдать набор исходных данных, достаточных для получения требуемого результата в заданном пункте (объекте).
4. Получить конечный пункт или трассу пути следования информации при неполном задании исходных данных.
5. Получить значения функций при определении трассы следования информации.

Рассмотрение вопроса вывода решений проводится применительно к способу представления технологии информационной системы. Формирование правильного по смыслу ответа достигается путем механизмов сопоставления фреймов, построенных из запроса, с фреймами из сети и получения в результате этого фреймов, в которых все слоты заполнены. Поиск фрейма в системе делается с помощью ключей, выделенных из запроса. После обработки найденного фрейма через слот АКО осуществляется переход на более высокий уровень иерархии фреймов и вновь применяются слоты нового фрейма для заполнения слотов фрейма, найденного по запросу. В ходе сопоставления может возникнуть конфликт информации, для его устранения используется комментарий.

Заключение

Созданные инструментальные средства, дающие возможность разработчику формально описывать технологию работы ИС, а администратору получать интенсивные сведения о структуре базы данных и технологии работы системы, являются по существу средствами построения экспертных систем в области информатики.

Разработанное математическое и программное обеспечение апробировано для автоматизированной системы учета и анализа заболеваемости на основе сбора данных с предприятий города.

Автор выражает искреннюю признательность А.Д.Начинкину и Л.А.Егошину за плодотворные обсуждения проблемы. Автор благодарит В.В.Байдуна и П.Л.Чернова за предоставленную возможность программирования с использованием языка FRL.

Список литературы

- [1] Егошина Т.В. Формализация описания технологических процессов в АСУ: Препринт ИФВЭ 96-43. — Протвино, 1996.
- [2] Minsky M. A Framework for Representing Knowledge // AI Memo 306, MIT, June 1977.

- [3] Roberts R.B., Goldstein I.P. The FRL Manual // AI Memo 409, MIT, September 1977.
- [4] Байдун В.В., Бунин А.И., Кружилов С.И., Сергиевский А.Е., Чернов П.Л. /Под редакцией Семеновой Е.Т. Языки и системы представления знаний (язык программирования ФРЛ). — М.: Изд-во МЭИ, 1994.
- [5] Антонов В.Г., Володин Д.В., Давидов М.И., Шульгейфер О.Е. Диалоговая единая мобильная операционная система ДЕМОС. Описание языка. Часть 3. Описание языка YACC. — Калинин, НПО "Центрпрограммсистем", 1985.
- [6] Антонов В.Г., Давидов М.И. Диалоговая единая мобильная операционная система ДЕМОС. LEX – генератор программ лексического анализа. — Калинин, НПО "Центрпрограммсистем", 1985.

Рукопись поступила 28 мая 1998 г.

Т.В. Егошина.

Формирование знаний о технологии обработки данных в информационных системах.

Оригинал-макет подготовлен с помощью системы \LaTeX .

Редактор Н.В.Ежела.

Технический редактор Н.В.Орлова.

Подписано к печати 2.06.98. Формат $60 \times 84/8$. Офсетная печать.

Печ.л. 1,12. Уч.-изд.л. 0,86. Тираж 150. Заказ 176. Индекс 3649.

ЛР №020498 17.04.97.

ГНЦ РФ Институт физики высоких энергий
142284, Протвино Московской обл.

