УДК 519.876.5: 621.865.8

А.Н. Горитов, И.В. Колотаев

Синтез управляемых механических устройств с применением экспертной системы*

В статье рассмотрены вопросы автоматического синтеза управляемых механических систем. Предложена методика синтеза и анализа управляемых механических систем. Затронуты вопросы реализации продукционной системы синтеза, в том числе: алгоритма обратного логического вывода и представления знаний в продукционной системе.

Ключевые слова: синтез, экспертная система, управляемые механические системы.

Автоматизация и роботизация является одним из основных элементов повышения производительности труда в машиностроении [1]. Они давно и успешно применяются при крупносерийном производстве. Автоматизация серийного и мелкосерийного производства сталкивается с проблемой выбора средств автоматизации, наиболее полно отвечающих автоматизируемому производству.

В настоящее время для создания управляемых механических систем (УМС) и роботизированных комплексов широко используется агрегатно-модульный подход. Этот подход дает возможность на базе ограниченной группы нормализованных узлов создать специализированную конструкцию УМС, которая наиболее полно удовлетворяет требованиям решения конкретной технологической задачи и не имеет избыточности.

Такой подход в каждом конкретном случае позволяет использовать УМС более дешевые по сравнению с универсальными, сократить время разработки и проектирования специализированных УМС, повысить надежность УМС за счет отработанности входящих в него узлов и агрегатов, удешевить производство за счет снижения номенклатуры деталей и узлов и увеличения объемов их выпуска.

При использовании агрегатно-модульного принципа возрастает роль методов расчета и проектирования, позволяющих обоснованно выбирать как параметры отдельных модулей и агрегатов робота, так и их компоновки, определяющие кинематические и динамические характеристики робота.

Пусть заданы некоторая технологическая операция Т и внешняя среда, в которой эта операция будет выполняться. Необходимо синтезировать УМС, обеспечивающую выполнение операции Т.

Одним из возможных подходов проектирования УМС на основе агрегатно-модульного принципа является использование экспертных систем (ЭС) [1]. Достоинство применения ЭС заключается в возможности применения в уникальных ситуациях, для которых алгоритм заранее неизвестен и формируется по исходным данным в виде цепочки рассуждений из БЗ, причем в условиях неполноты, недостоверности, многозначности исходных данных. Экспертная система позволит быстро подобрать множество вариантов структуры УМС, удовлетворяющих заданным ограничениям.

Экспертная система проектирования УМС имеет следующий состав: база данных (БД), база знаний (БЗ), интерфейс, машину вывода, правила задания неопределенности.

Одной из составляющей ЭС является база знаний (БЗ) – формализованное представление знаний, описание объектов предметной области и их взаимосвязей, действий над объектами, неопределенностей. С помощью механизма вывода из БЗ формирует конкретный алгоритм решения задачи.

Основными факторами, учитываемыми при синтезе УМС, являются выполняемая технологическая операция, количество и тип звеньев, грузоподъемность, точность позициониро-

* Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (код проекта 09-07-99029-р_офи).

вания, типы приводов, зона обслуживания и т.д. Образами будут являться структуры данных из фактов в рабочей памяти, образцами – правила продукции.

Экспертная система обладает следующими свойствами:

- по способу формирования решения ЭС является синтетической, т.е. выполняется не выбор решений из множества известных альтернатив, а генерация нового решения;
- по способу учета временного признака экспертная система является статической;
- по виду используемых данных и знаний ЭС будет неоднозначной: учитывается физическая (неточность измерений) и лингвистическая неопределенность (нечеткость);
- синтез выполняется на основе прямого вывода;
- база знаний задана правилами (объект–свойство–значение).

В экспертных (продукционных) системах широко применяется обратный вывод, то есть вывод, управляемый целевыми условиями [2]. В созданной системе реализован именно этот способ.

Для объектно-ориентированного представления знаний характерно применение механизма наследования атрибутов, когда знание атрибутов передается по иерархии от вышестоящих классов к нижестоящим.

В процессе выполнения работы была спроектирована и реализована система классов (рис. 1), представивших собой как машину логического вывода, так и набор инструментов для представления знаний в ней.

Поясним предназначение каждого класса.

TES_BaseClass – базовый класс для всех элементов машины логического вывода в экспертной системе.

TES_BaseListClass – базовый класс, реализующий функциональность списка элементов (будь то переменные, предикаты, списки предикатов (продукции)).

TES_Predicate – класс, представляющий собой предикат. Так как класс унаследован от базового класса-списка, то предикат может хранить параметры, может быть и без параметров. Параметрами предиката являются экземпляры производных класса базовых переменных.

TES_BaseVariableClass – базовый класс для всех переменных в машине вывода.

TES_Variable – несет смысловую нагрузку неопределённой переменной.

TES_Atom – несёт смысловую нагрузку простой строки.

TES_List – список параметров (как определённых, так и неопределённых).

TES_Struct – класс для представления сложных, структурированных параметров.

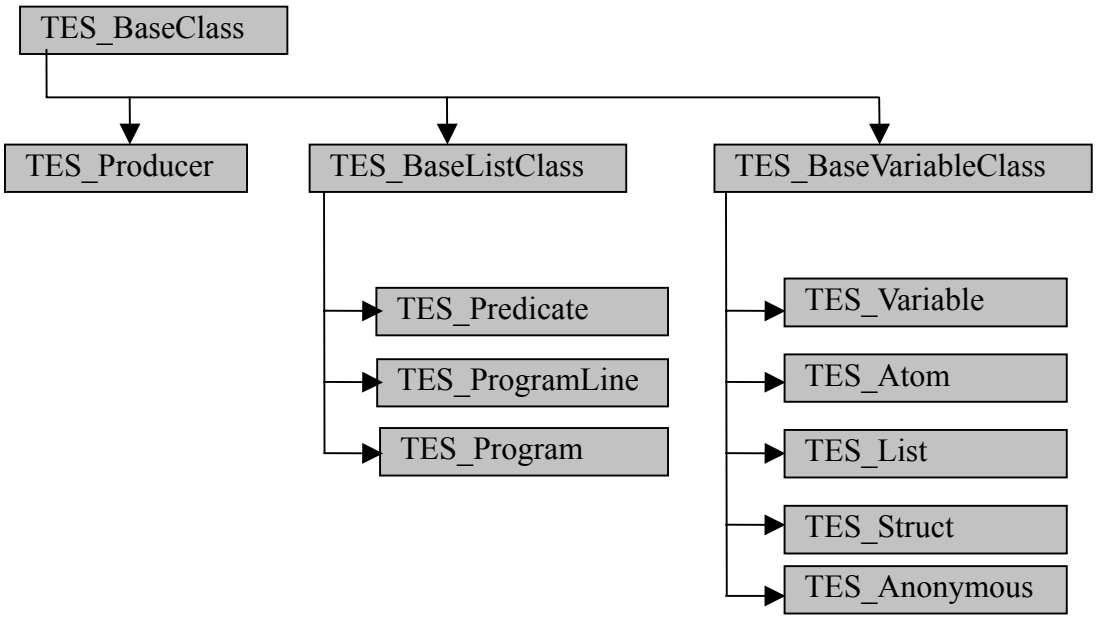

Рис. 1. Иерархия классов

TES_ProgramLine – предназначен для представления отдельных продукций в машине вывода. Включает в себя список предикатов. Так список неких предикатов [A, B, C] в этом классе соответствует правилу A->B^C. Также представляет собой факты – продукции без правой части.

TES_Program – представляет собой программу вывода системы. Программа вывода содержит в себе множество продукций и множество фактов.

TES_Producer – представляет собой машину логического вывода. Содержит в себе программу вывода. Она содержит в себе все механизмы для работы с памятью, хранения, преобразования фактов и правил. В ней реализован механизм сохранения и восстановления состояния. Позволяет подтвердит цель различными способами.

Классы спроектированы так, что факты из базы знаний о предметной области и правила вывода логической машины представлены экземплярами одного и того же класса. Такая унификация упрощает разработку системы, позволяет хранить базу знаний и продукционные правила в одной базе данных, не требуя создания отдельных, отличающихся по интерфейсу и функциональности хранилищ.

В вышеописанной машине вывода применялся алгоритм обратного логического вывода. Этот алгоритм обладает следующими особенностями:

- обратный вывод с использованием стратегии разрешения конфликтного множества продукций перебором в глубину. Эта стратегия по сравнению с перебором в ширину или со стратегией равных цен позволяет получить первый ответ системы быстрее;
- встроена система объяснения – система, позволяющая пользователю системы объяснить, почему система приняла то или иное решение, на каких фактах она основывалась, какими правилами пользовалась;
- благодаря спланированной особым образом системе классов, работу алгоритма можно прервать в любой момент, но при этом сохранить его состояние.

Информация об отдельных модулях робота составила базу фактов системы. Для представления фактов в рабочем пространстве машины вывода использовалась система классов, описанная выше и представленная на рисунке 1. Так, например факт `connect (n,[k,d,p,v,s,rh])` во внутреннем представлении выглядит, как это показано на рисунке 2.

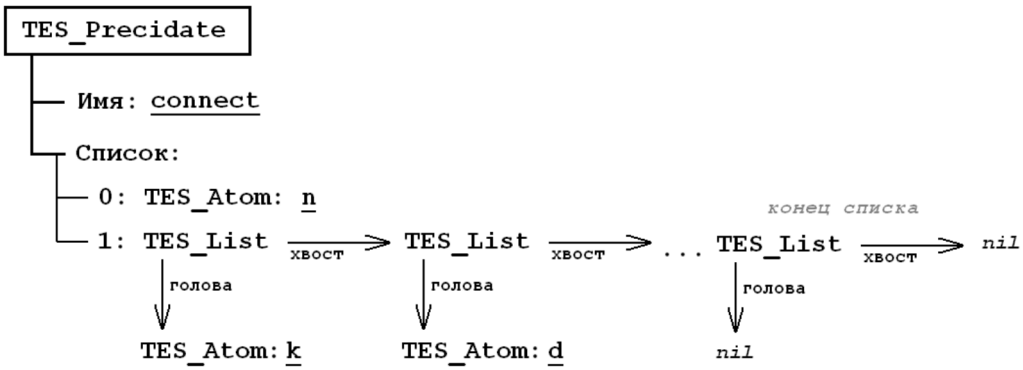

Рис. 2. Представление факта

Факты об отдельных компонентах были представлены следующим образом:

base(t_0).

…

base(t_m).

Информация о соединениях компонентов между собой между собой вошла в базу знаний о роботе и представлена в таком виде:

connect(n,[k,d,p,v,s,rh]).

Также в систему вошёл вариант программы вывода конфигурации робота:

getRobot(X,N,Items):-robot(X), go1(X,N,[],Items).

go1(X,N,In,In):-N>(-1), connect(X,[]).

go1(X,N,In,Out):-N1 is N-1, connect(X,Smeg), member(Y,Smeg), append(In,[Y],In1), go1(Y,N1,In1,Out).

Каждый компонент имеет набор свойств. Он задаётся в следующем виде:

propertySet(component, [property1, property2, ...]),

где component – идентификатор компонента в базе знаний, property1, property2, ... – список идентификаторов свойств, описывающих этот компонент.

В базе хранятся факты о предметной области и факты о наборах свойств (факты о фактах, знания о знаниях). Так, каждое свойство из списка свойств, описывающих компонент, само описывается следующими фактами:

propertySettings(property, shortName, longName) – этот факт описывает наименование свойства:

property – идентификатор свойства;

shortName – краткое описание свойства (также его название);

longName – развёрнутое описание свойства.

propertyType(property, [type1, type2, ...], [name1, name2, ...]) – этот факт описывает внутреннее представление свойства. Свойство может обладать набором полей для описания параметров компонента (так например свойство «Размер» должно обладать тремя полями – длина, ширина и высота):

property – идентификатор свойства;

[type1, type2, ...] – список типов полей свойства;

[name1, name2, ...] – список наименований полей (представляет из себя список строк).

proprtyOrd(property, ord_predicate) – этот факт. если он присутствует, задаёт предикат, при помощи которого можно сравнить компоненты по заданному свойству:

property – идентификатор свойства;

ord_predicate – предикат сравнения.

Выбранный способ описания свойств компонента позволяет легко вводить в базу знаний новые свойства. Более того, пользователь системы в дальнейшем может создавать свои собственные свойства.

В процессе разработки в систему был введён набор РПМ-25 – набор модулей для создания агрегатно-модульных роботов-манипуляторов [3]. Этот робот может включать в свою конфигурацию следующие модули (или компоненты): напольное подвижное основание, подвесное подвижное основание, неподвижное основание, модуль поперечного сдвига, модуль поворота, модуль подъёма, модуль радиального хода, модуль двойного качения, модуль качения, модули рук, модули захватов (одинарный, двойной, с поперечным сдвигом).

Рассмотрим, какие свойства приобрели в базе знаний отдельные модули робота РПМ-25. На примере модуля руки:

propertySet(r1, [nameOfComponent(['Модуль руки']), componentType(['PARAL', 'Paral1']), raumsParam(['3,0.5,0.5']), connectPlace2([3, 0.25, 0.25]), connectPlace0([0.25, 0.25, 0])]).

nameOfComponent — свойство определяет название компонента.

componentType — геометрический тип компонента. Это свойство применимо к системе моделирования РАУМС и указывает какое именно геометрическое тело представляет собой модуль робота.

raumsParam — параметры модуля в системе моделирования РАУМС. Само значение параметров и их количество зависит от типа компонента (предыдущее свойство) и меняется от одного типа к другому. Так, для типа «PARAL», представляющего собой параллелепипед, требуется указать три параметра через запятую – линейные размеры модуля.

connectPlace2 — координаты точки пристыковки следующего компонента структурной схемы робота. Эти координаты задаются в локальной системе координат. В такой системе в начале координат (в точке O) лежит сам компонент (модуль). Преимущества такого подхода — пользователю системы не нужно ломать голову, где окажется точка подстыковки одного компонента к другому в готовой конфигурации. Более того, задать эту точку как-нибудь иначе, чем в локальных координатах, представляется трудноосуществимым.

connectPlace0 — координаты точки пристыковки этого модуля к предыдущему в структурной схеме робота. Эти координаты тоже задаются в локальной системе координат.

Численный эксперимент с набором модулей РПМ-25 показал, что экспертная система обеспечивает синтез всех 36 различных конфигураций, допустимых для этого набора модулей.

Предложенная методика включает синтез вариантов УМС, анализ синтезированных вариантов и отбор синтезированных вариантов по заданным критериям. Это позволяет проектировать и моделировать управляемые механические системы на единой методической и программно-алгоритмической основе.

В настоящее время синтез управляемых механических систем включает подсистему структурного синтеза и базу знаний, в которую занесены знания о промышленном роботе РПМ-25.

Система синтеза предполагает построение базы типовых модулей робота; построение экспертной системы, предлагающей различные варианты структуры робота, по заданным условиям; построение интерфейса с пользователем системы, включающего в себя редактор свойств модулей и компонентов, подсистему визуализации синтезированной структуры робота, структурный редактор, для ручной доводки робота.

Литература

1. Змитрович А.И. Интеллектуальные информационные системы. – Минск : ТетраСистемс, 1997. – 368 с.

2. Павлов С.Н. Интеллектуальные информационные системы : учебное пособие. – Томск : ТМЦДО, 2004. – 328 с.

3. Механика промышленных роботов. В 3 кн. / под ред. К.В. Фролова, Е.И. Воробьева. Кн. 3 : Основы конструирования / Е.И. Воробьев, А.В. Бабич, К.П. Жуков и др. – М. : Высш. шк., 1989. – 384 с.

Горитов Александр Николаевич
Доктор техн. наук, профессор кафедры автоматизированных систем управления ТУСУРа
Тел.: (3822) 70-15-36
E-mail: ang@asu.tusur.ru

Колотаев Илья Владимирович
Аспирант каф. автоматизированных систем управления ТУСУРа
Тел.: (3822) 70-15-36
E-mail: vlilk@ms.tusur.ru

A.N. Goritov, I.V. Kolotayev

Operated mechanical devices synthesis with application of expert system

This paper describes the questions of operated mechanical systems automatic synthesis. The technique of operated mechanical systems synthesis and analysis is offered. Questions of productive synthesis systems realisation are mentioned, including: return(inverse) logic conclusion algorithm and knowledge representation in productive system.

Keywords: synthesis, the expert system, operated mechanical systems.