

Д.Э. ДВУХГЛАВОВ, канд. техн. наук

ПРИНЦИПЫ ПОСТРОЕНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СИСТЕМ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ

Пропонується будувати програмне забезпечення, що розробляється для вирішення певних задач управління, як алгоритми із логічними блоками. Наявність таких блоків передбачає проведення логічного виведення на знаннях, що є собою правила реалізації окремих етапів виробки рішення (правила відбору альтернатив, правила вибору найкращого варіанту). Використання такого принципу забезпечить збільшення тривалості експлуатації програмного забезпечення систем підтримки прийняття рішень.

Construction of the software which is developed for the decision of the certain tasks of management as algorithms with logic blocks is offered. Presence of such blocks provides realization of a logic conclusion on knowledge which represents rules of realization of separate stages of the decision acceptance (rules of selection of alternatives, rules of a choice of the best variant). Use of such principle will provide increase of use period of the software of decisions acceptance support systems.

Постановка проблеми. Принятие эффективных решений в установленные сроки – основная задача функционирования органов управления различных отраслей производства, транспорта, а также органов управления социальными коллективами. В условиях постоянного возрастания объема информации, подлежащей обработке для принятия эффективных решений, а также сокращения времени на выработку решения (это особенно характерно для систем, функционирующих в реальном масштабе времени) использование средств автоматизации является необходимостью.

По функциональному предназначению система поддержки принятия решений (СППР) – это система, представляющая лицу, принимающему решение, возможности анализа ситуаций, постановки задач, выработки, контроля и оценки вариантов решений, обеспечивающих достижение поставленной цели [1].

В общем случае известны два подхода к построению систем поддержки принятия решений: информационный и когнитивный [2].

Информационный метод базируется на следующих принципах:

1. Носителями знаний о системе, ее целевом предназначении и условиях функционирования является лицо, принимающее решение, и разработчики специального программного обеспечения.

2. ЭВМ является средством решения расчетных задач, описываемых конечным набором числовых параметров, и проверки выполнения простейших логических условий на числовых данных.

3. Процесс управления сводится к решению задач указанного в п. 2 характера, которые формулируются на этапе разработки с учетом всех

возможных условий функционирования системы, заданных директив, норм ее поведения и факторов, влияющих на процесс управления.

Когнитивный (от лат. *cognitio* - знания, познание) метод появился благодаря развитию идей искусственного интеллекта. Его характерные черты:

1. Все или часть знаний, необходимых для принятия решений в широком диапазоне деятельности системы, формализуется и помещается в базу знаний.

2. Наряду с программами решения задач расчетного характера в ЭВМ хранятся программы манипулирования знаниями, позволяющие извлекать из памяти компоненты знаний, релевантные поставленной задаче, и эффективно их использовать при поиске решения.

3. На ЭВМ могут быть возложены функции по формулировке или доопределению условий задач принятия решений, а также по поиску решений в непредвиденных ситуациях.

Использование первого подхода предполагает построение СППР как информационно-справочной или информационно-расчетной системы органа управления. Разработка программного обеспечения таких систем, как правило, выполняется в соответствии с технологией проектирования систем обработки информации на основе баз данных. Современные программисты в достаточной мере вооружены инструментальными средствами разработки подобных систем. Помимо прикладных программ ведения базы данных и манипулирования хранящейся в ней информацией в состав программного обеспечения также включаются модули, реализующие решение расчетных задач по известным алгоритмам, разработка которых не вызывает принципиальных трудностей. Таким образом, основным достоинством СППР, реализованных с использованием информационного метода, является относительная простота их программного обеспечения. Однако при таком подходе лица, принимающие решения, не освобождаются от логико-аналитической деятельности по обработке получаемой информации. Объем такой деятельности, выполняемый операторами, например, при принятии решений в ходе управления воздушным движением, энергетическими установками, может вызвать психофизические перегрузки, стрессы, и даже привести к полной невозможности выполнения функциональных обязанностей.

Представление и использование правил реализации логико-аналитической деятельности возможно при использовании когнитивного подхода при разработке программного обеспечения СППР. Применение данного подхода открывает возможности создания СППР, адаптирующихся к внешним условиям функционирования, способных к самосовершенствованию, самообучению на опыте [2]. Эти свойства определяют перспективность использования интеллектуальных СППР.

Однако анализ практики разработки систем управления позволяет сформулировать тезис, что использование только информационного или только когнитивного подхода к автоматизации при разработке программного

обеспечения СППР невозможно. Дело в том, что для решения части задач управления интеллектуальность не всегда является необходимостью. Например, для расчета расстояния между двумя воздушными суднами (задача, решаемая в процессе управления воздушным движением), использование искусственного интеллекта излишне, это обычная расчетная задача. В тоже время в АСУ воздушным движением желательно учитывать опыт действий и рассуждений диспетчеров в ходе принятия решений как в типовых, так и в нестандартных ситуациях, что невозможно без применения когнитивного подхода.

Таким образом, можно утверждать, что на современном этапе существует объективная необходимость рационального сочетания традиционных и интеллектуальных технологий обработки информации при разработке программного обеспечения СППР, однако методы, позволяющие реализовать такое сочетание, отсутствуют.

Анализ литературы. В известной литературе современный программист может найти общие приемы программирования [3], основы теории построения баз данных и прикладных программ для работы с ними [4]. Широко рассмотрены известные модели знаний и процедуры их обработки [5]. Однако, рекомендаций по реализации алгоритмов совместной обработки знаний и данных при решении прикладных задач, не приводятся. Проблема, сформулированная выше, упоминается в [6], но конкретных предложений для ее разрешения в данном труде не содержится.

Цель статьи. В статье предлагается подход к построению программного обеспечения, сочетающего достоинства информационного и когнитивного подхода к построению СППР, а также особенности обработки информации в процессе выработки решений.

Основной раздел. Рассмотрим предлагаемый подход к построению программного обеспечения на примере разработки интеллектуальной системы планирования автотранспортного предприятия.

Пусть одной из задач, решение которой возлагается на данную систему, заключается в выборе грузового автомобиля для перевозки заданного груза с учетом пожеланий заказчика.

Рассмотрим процесс решения данной задачи в неавтоматизированном варианте. Общую схему решения включает два этапа:

1. Отбор всех машин предприятия, которые могут выполнить перевозку.
2. Выбор машины для выполнения перевозки в соответствии с критериями оптимальности.

Реализация первого этапа предполагает последовательный просмотр перечня машин и рассмотрение каждой машины со следующих позиций:

– подходит ли тип кузова машины (самосвал, цистерна, бортовая машина) для перевозки груза;

– подходит ли грузоподъемность машины и объем кузова для перевозки груза.

Для реализации первой проверки необходимо иметь информацию о типе кузова и его объеме, информацию о том, к какому типу относится груз (сыпучее, жидкость, горючая жидкость и т.д.), а также информацию о соответствии типа груза типу кузова. Для установления соответствия грузоподъемности автомобиля массе груза необходимо определить марку автомобиля, затем узнать по техническому паспорту (или сводной таблице характеристик автомобилей) предельную массу груза, которую можно перевозить; далее следует рассчитать массу груза по его объему, используя табличное значение плотности груза данного типа; сравнить две полученные величины.

Если представить необходимую информацию в базе данных, то для написания программы достаточно использовать принципы информационного подхода. Однако кроме данных очевидных требований, заказчик может дополнительно потребовать, например, чтобы для перевозки была выделена машина марки "Мерседес". Что делать в этом случае? С одной стороны общая схема решения отбора машин не меняется: просмотр всех машин и проверка соответствия заданным условиям, при этом набор условий для проверки изменяется. Модификация программного обеспечения не решит общей проблемы – следующий заказчик может выдвинуть свое новое условие.

Выходом из проблемы может быть разработка средств, позволяющих «наращивать» правила отбора. Речь идет об организации **базы знаний**, в которой будут храниться правила отбора, и разработке процедуры обработки правил (**процедуры логического вывода**), которая будет вызываться как обычная процедура из программного модуля, реализующего просмотр вариантов. Предлагаемый вариант построения алгоритма отбора представлен на рис.1. Список ИСХ, упоминаемый на рисунке, – это исходное множество элементов, список УУО – список элементов, удовлетворяющих условиям отбора.

Аналогичным образом может быть построен алгоритм реализации второго этапа поиска решения. На данном этапе необходимо проанализировать машины, прошедшие отбор на первом этапе, и сравнить их по некоторому заданному критерию, например, по минимуму расхода топлива или по коэффициенту заполнения кузова (по соотношению объема груза к объему кузова).

Принцип решения данной задачи аналогичен принципу решения любой задачи упорядочивания. Необходимо взять первый элемент некоторого списка, сравнить его по заданному критерию со следующим в списке. Если второй элемент превосходит по критерию первый, то этот элемент становится текущим, переносится в списке на первую позицию, а на его место помещается элемент, который был первым до сравнения. Далее анализируется третий элемент в списке, и действия повторяются. В результате на первой

позиции окажется наилучший в соответствии с заданными критериями элемент. В нашем случае на первой позиции окажется наиболее подходящая машина для перевозки груза.

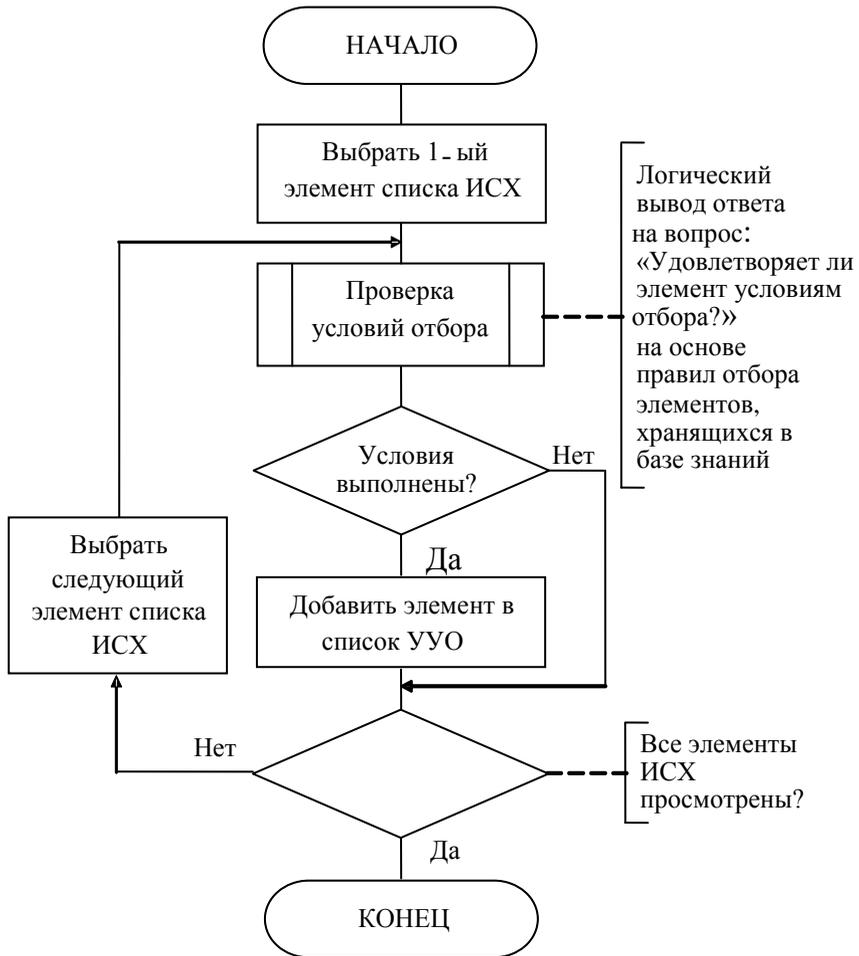


Рис 1. Схема алгоритма отбора

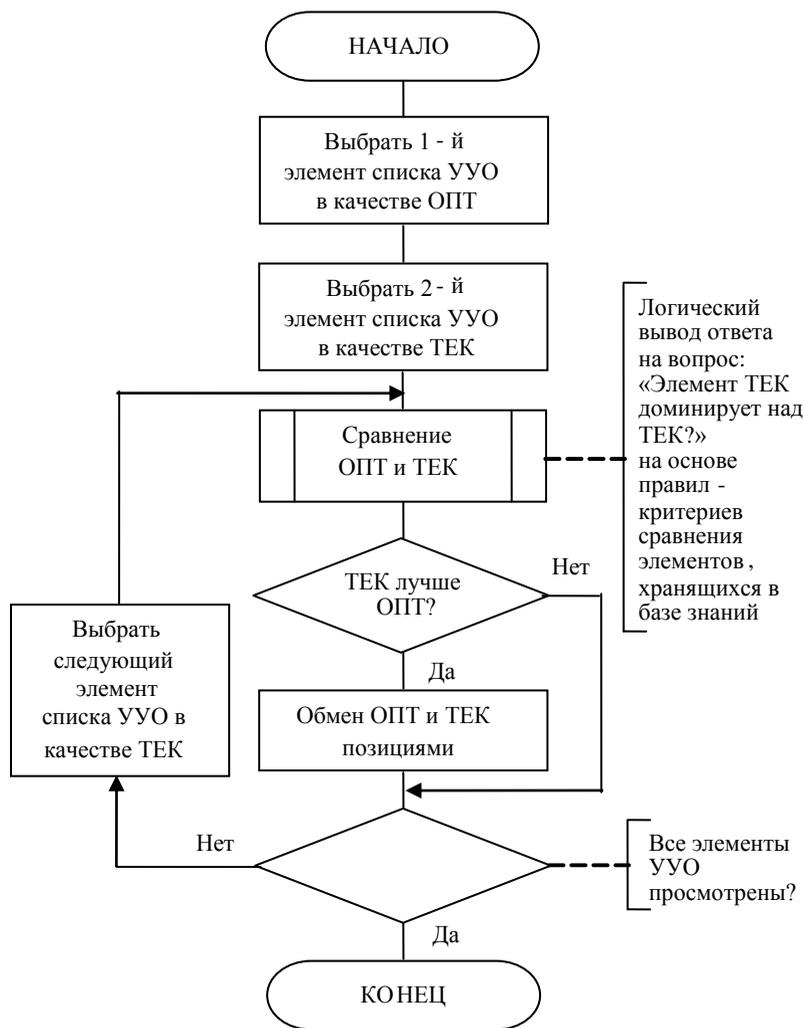


Рис. 2. Схема алгоритма выбора наилучшего элемента

Опять возникает вопрос: как быть, если условия определения оптимальности машины изменяться? Например, будет поставлена задача учета и заполнения кузова, и расхода топлива при выборе оптимальной машины? Выход тот же – обеспечить возможность гибкого наращивания правил сравнения альтернатив. Предлагаемый алгоритм реализации выбора оптимального элемента представлен на рис. 2. Дополнительно введены

следующие обозначения: ОПТ – наилучший элемент, ТЕК – элемент-претендент, анализируемый на текущем шаге.

Рассмотренный подход к решению задачи выбора транспортного средства может быть использован и для решения подобных задач из других предметных областей (выбор медицинского учреждения для лечения, назначение исполнителя для выполнения работы и т.п.). При этом будет меняться состав анализируемых элементов и правил.

Рассмотренный подход решения задачи выработки решения позволяет сформулировать следующие утверждения:

1. Рациональный подход к разработке программного обеспечения, предназначенного для решения задач поддержки принятия решений, заключается в сочетании традиционных и интеллектуальных технологий обработки информации.

2. С использованием традиционных технологий должны быть реализованы известные методы обработки списочных структур, к которым в первую очередь относятся перебор и упорядочивание.

3. Программные модули, реализующие процедуры обработки списков в соответствии с экспертными правилами, меняющимися со временем, должны содержать команды вызова процедуры обработки данных правил. Такие процедуры предлагается называть **процедурами с логическими блоками**.

Рассмотрим более детально процедуру обработки правил.

В рассмотренном примере процессу применения правил отбора или сравнения вариантов предшествовали определенные подготовительные действия. Так, прежде чем проверить соответствие машины грузу по грузоподъемности необходимо рассчитать массу груза, найти в базе данных значение грузоподъемности. Кроме того, в логический блок должна каким-то образом быть передана информация о какой машине и о каком грузе идет речь. Только после выполнения данных действий может быть осуществлена проверка выполнимости правил отбора или сравнения двух альтернатив.

Данные рассуждения позволяют утверждать следующее:

1. Внутри логических блоков реализуется решение расчетно-логических задач приемлемой сложности.

2. Процесс решения задач предполагает реализацию следующих этапов:

- этап подготовки данных для логико-аналитической деятельности;
- логическое обобщение полученной информации.

3. Данные на подготовительном этапе могут быть получены тремя способами:

- путем ввода информации пользователем, вызывающей программой, от внешних источников информации;
- путем поиска информации в базе данных или ином информационном массиве;
- путем расчета значений в соответствии с заданными алгоритмами.

Следует отметить, что подготовительный этап предполагает сочетание указанных действий, например, ввод исходных данных – поиск – расчет.

4. Логико-аналитическая деятельность предполагает:

- сравнение полученных на предварительном этапе данных с эталонными значениями или между собой;
- логическое обобщение результатов сравнения.

Каждому из предложенных этапов поиска ответа на вопрос, а фактически решения несложной расчетно-логической задачи, можно поставить некоторую процедуру (ввода исходных данных, поиска, расчета, сравнения и обобщения), тогда логический вывод ответа на вопрос будет представлять собой упорядоченный вызов соответствующих процедур. В [7] предлагается модель представления правил для реализации логических блоков, в [8] рассматривается структура для их хранения в памяти ЭВМ, в [9] – процедура обработки.

Выводы. Устойчивость систем автоматизации решения задач в условиях постоянной изменчивости условий функционирования, связанной с изменением объектов управления, внешней среды и, соответственно, методов выработки решений – одно из требований к современным СППР. Использование предлагаемого подхода к построению программного обеспечения СППР позволяет удовлетворить данное требование за счет гибкой адаптации программ решения задач к требуемым правилам реализации логико-аналитической деятельности.

Список литературы. 1. ДСТУ 2481-94. Системи оброблення інформації. Інтелектуальні інформаційні технології. – К.: Держстандарт України, 1994. – 72 с. **2.** Ярушек В.Е., Прохоров В.П., Судаков Б.Н., Мишин А.В. Теоретические основы автоматизации процессов выработки решений в системах управления. – Харьков: ХВУ, 1993. – 483 с. **3.** Кнут. Д. Искусство программирования для ЭВМ. – Т.1. – М.: Мир, 1976. – 729 с. **4.** Фаронов В.В., Шумаков В.Н. Разработка баз данных в среде программирования Delphi 4. – М.: Нолидж, 1999. – 458 с. **5.** Бондарев В.Н., Аде Ф.Г. Искусственный интеллект. – Севастополь: СевНГУ, 2002. – 615 с. **6.** Georgiadou E. Software Process and Product Improvement: A Historical Perspective // Кибернетика и системный анализ. – 2003. – №1. – С. 147–166. **7.** Двухглавов Д.Э., Затхей В.А. Представление процесса решения расчетно-логических задач управления с использованием сетевой модели // Системи обробки інформації. – Харків: НАНУ, ПАНМ, ХВУ. – 2002. – Вип. 4 (20). – С. 117–122. **8.** Двухглавов Д.Э., Кулабухов А.М. Представление сетевой модели процесса решения задач управления в памяти ЭВМ // Вестник НТУ "ХПИ". Сб. науч. тр. Тематический выпуск "Системный анализ, управление и информационные технологии". – Харьков: НТУ «ХПИ», 2002. – № 13. – С. 87–90. **9.** Двухглавов Д.Е., Затхей В.А., Карпов С.І. Організація вирішення логіко-розрахункових задач в процесі вироблення рішень // Моделювання та інформаційні технології. Зб. наук. пр. – К.: НАНУ, Інститут проблем моделювання в енергетиці, 2003. – Вип.22. – С. 52–55.

Поступила в редакцию 15.04.2004