

ИНФОРМАЦИОННЫЕ, ИЗМЕРИТЕЛЬНЫЕ, ЭКСПЕРТНЫЕ, ОБУЧАЮЩИЕ СИСТЕМЫ

УДК 681.5:622.276:622.279

СЕМИОТИЧЕСКИЕ СИСТЕМЫ КАК СРЕДСТВО ОПИСАНИЯ МОДЕЛЕЙ ПРЕДМЕТНОЙ ОБЛАСТИ УПРАВЛЕНИЯ, ПРОЦЕССОВ ОЦЕНКИ СИТУАЦИЙ И ИНТЕЛЛЕКТУАЛЬНОЙ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ

А.А. Башлыков
(ЗАО "ВНИИСТ-Нефтегазпроект")

Введение

Системы интеллектуальной поддержки принятия решений, основанные на знаниях, в определенном смысле, моделируют интеллектуальную деятельность человека и, в частности, логику его рассуждений при решении задач оперативной поддержки принятия решений – модель оперативной деятельности [1]. В упрощенной форме логические построения экспертов, строящих такую модель, сводятся к следующей схеме: из одной или нескольких посылок (которые считаются истинными) следует сделать "логически верное" заключение (вывод, следствие). Очевидно, для этого необходимо, чтобы и посылки и заключение были представлены на понятном языке, адекватно отражающем предметную область, в которой проводится вывод. С другой стороны, если формализовать всё множество моделей интеллектуальной деятельности людей-управленцев в рамках некоторого математического аппарата и построить средства автоматического решения задач в созданном мультимодельном множестве, то появляется возможность построения некоторого прикладного базиса средств проектирования и реализации систем интеллектуальной информационной поддержки принятия управляющих решений.

Рассмотрению такого математического аппарата – семиотических моделей, систем и семиотических баз знаний посвящена настоящая статья.

1. От формальных систем к семиотическим системам

Известно, что в обычной жизни наш естественный язык общения, в математике, например, это язык определенных формул и т. п. Наличие же языка предполагает, во-первых, наличие алфавита (словаря), отображающего в символьной форме весь набор базовых понятий (элементов), некоторой предметной области и, во-вторых, набор синтаксических правил, на основе которых, пользуясь алфавитом, можно построить определенные правильно построенные выражения, описывающие взаимосвязи, существующие в предметной области.

Логические выражения, построенные в данном языке, могут быть истинными или ложными. Некоторые из этих выражений являются всегда истинными.

Такие выражения объявляются *аксиомами* (или *постулатами*). Они составляют ту базовую систему посылок, исходя из которой, пользуясь определенными правилами вывода, можно получить заключения в виде новых выражений, также являющихся истинными.

Если перечисленные условия выполняются, то говорят, что система удовлетворяет требованиям *формальной теории* [2]. Ее будем называть *формальной системой* (*FS*). Система, построенная на основе формальной теории, называется также *аксиоматической системой*.

Формальная теория должна, таким образом, удовлетворять следующему определению: всякая формальная теория, определяющая некоторую аксиоматическую систему вида

$$FS = (T, P, A, R),$$

которая характеризуется:

- наличием алфавита (словаря) T ,
- множеством синтаксических правил построения выражений P ,
- множеством аксиом, лежащих в основе теории A ,
- множеством правил вывода R .

Исчисление высказываний (ИВ) и исчисление предикатов (ИП) являются классическими примерами аксиоматических систем. Эти *FS* хорошо исследованы и имеют прекрасно разработанные модели логического вывода (поиска решений для поставленных задач) – главной метапроцедуры в интеллектуальных системах, например, основанной на "принципе резолюций" [3–6]. Поэтому всё, что может и гарантирует каждая из этих систем, гарантируется и для прикладных *FS* как моделей конкретных предметных областей. В частности, это гарантии непротиворечивости вывода, алгоритмической разрешимости (для исчисления высказываний) и полурешимости (для исчисления предикатов первого порядка).

FS имеют и недостатки, которые стимулируют поиск других форм описания предметных областей. Главный недостаток *FS* – это их "закрытость", негибкость в разнородных явлениях предметного мира. Модификация и расширение здесь всегда связаны с перестройкой всей *FS*, что для практических систем сложно, трудоемко и не всегда позволяет получить желаемый результат. В них очень сложно и накладно учитывать происходящие в конкретной предметной

среде изменения. Поэтому FS как модели представления знаний используются в тех предметных областях, которые хорошо локализируются и мало зависят от внешних факторов.

Поспелов Д.А. в работе [7] показал, что моделирующие способности FS могут быть расширены, путем:

- объединения всех FS некоторой предметной области управления;
- управлением каждым из задающих множеств базовых элементов, синтаксических правил, аксиом и правил вывода для каждой FS .

Это позволило ему ввести понятие семиотической модели как средства интеграции FS , изменения синтаксиса и семантики FS .

Определение 1. Семиотическая модель (SM) представляется выражением вида

$$SM = \{F, S, \alpha(T), \beta(P), \gamma(A), \sigma(R)\},$$

где $FS = \{T, P, A, R\}$ – формальная система, элементами которой являются:

T – множество термов;

P – множество правил построения правильно построенных формул;

A – множество аксиом;

R – множество правил вывода в формальной системе;

$\alpha(T)$ – правила изменения элементов T ;

$\beta(P)$ – правила изменения элементов P ;

$\gamma(A)$ – правила изменения элементов A ;

$\sigma(R)$ – правила изменения элементов R .

В этой же работе показано, что если создать средства, позволяющие фиксировать такты работы семиотической модели, то на каждом из этих тактов она работает как некоторая формальная система с фиксированными значениями T, P, A и R , и можно сказать, что семиотическая система меняет свои состояния, описывая некоторую траекторию. С другой стороны, если рассматривать SM как множество FS , описывающих предметную область и множество отношений между элементами этого множества, то можно утверждать, что можно построить процедуру решения задач в такой мультимодельной среде. Исходя из вышесказанного, введем определение семиотической системы.

Определение 2. Семиотическая система есть выражение вида

$$SS = \{FS, SM, \Xi(FS), \varphi(FS, SM), \alpha(T), \beta(P), \gamma(A), \sigma(R), \varepsilon(\Xi), \mu(Q)\},$$

где $\Xi(FS)$ – множество отношений между FS ; $\varphi(FS)$ – множество правил вывода в семиотической модели, определенное на множестве (FS, Ξ, SM) ; $\varepsilon(\Xi)$ – правила изменения элементов Ξ ; $\mu(\varphi)$ – правила изменения элементов φ .

Функционирование семиотической системы теперь можно представить в виде чередования шагов моделирования процессов поиска решений в рамках некоторой последовательности поиска в связанной совокупности формальных систем (если она существует).

Поспеловым Д.А. показано, что если правила взаимосвязи формальных систем в предметной области управления зафиксировать, или смоделировать в виде некоторой формальной системы (например, задать алгоритмически), то семиотическая система превращается в композицию множеств формальных систем, вкладываемых в некоторую общую формальную систему. Такая композиция могла бы моделировать некоторую сложную реальность, но не могла бы моделировать открытые системы.

Формализм семиотической системы позволяет описывать модель области управления сложными техническими системами в виде связанной совокупности моделей всех ее технологических компонент и моделировать технологические процессы, происходящие в любой сложной системе. Кроме того, на семиотической модели можно решать задачи интеллектуальной поддержки принятия решений.

Семиотическая модель представляет собой совокупность формальных моделей и "протокол" взаимодействия между этими моделями – способы перехода от одного описания объекта к другому (способы изменения или задания свойств формальной системы).

Кроме того, необходимо заметить, что при определении семиотической системы, как множества формальных систем M и множества переходов E между ними, в качестве M может быть использована практически любая математическая модель.

Как уже упоминалось выше, одной из главных причин появления понятия семиотической модели явилась необходимость описания сложных объектов, которые не поддаются описанию в рамках какого-либо одного языка представления знаний. В этом случае либо, согласно определению 1, семиотическая модель создается как некое расширение формальной модели, либо, согласно определению 2, как совокупность нескольких различных моделей и средств взаимодействия (перехода) между этими моделями.

2. Организация взаимодействия формальных моделей в семиотической системе

В случае описания предметной области в виде совокупности нескольких формальных моделей встает вопрос об организации взаимодействия между этими моделями.

Схематически один из способов взаимодействия моделей в рамках семиотической сети может быть представлен на рис. 1.

Для каждой модели M_i семиотической сети существует $2(N - 1)$ правил перехода, определяющих взаимодействие M_i с другими $(N - 1)$ моделями сети, где N – общее число моделей.

Как указывалось выше, одним из основных преимуществ семиотической модели как мультимодельной среды является ее открытость, т. е. возможность безболезненного встраивания новых моделей (новых типов моделей) в логическую модель – семиотическую сеть с целью расширения ее описательных или

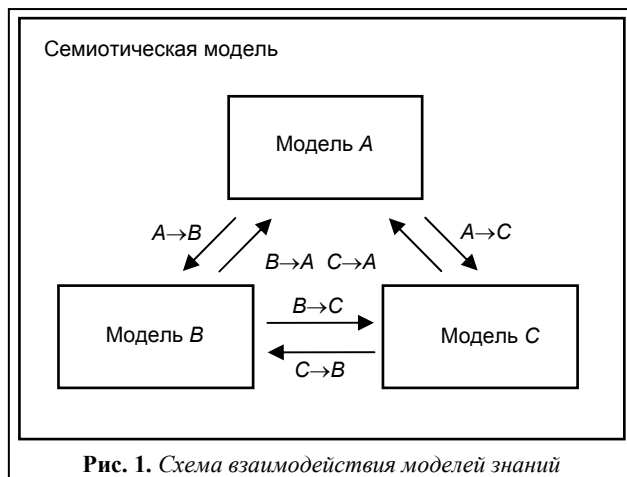


Рис. 1. Схема взаимодействия моделей знаний

вычислительных возможностей. Как следует из вышесказанного, в случае введения в семиотическую сеть новой модели (модели нового типа) потребуется определить дополнительно $(N - 1)$ правил перехода для описания всех возможных типов взаимодействия между всеми моделями сети.

Очевидно, что с ростом числа входящих в семиотическую сеть моделей (т. е. с ростом возможностей сети) будет происходить лавинообразный рост количества требуемых дополнительных определений. В случае большого числа моделей N (даже порядка 3–4) такой подход может вызвать значительные затруднения. То есть рост описательных возможностей сети сдерживается сложностью процедуры определения этих возможностей.

Данная проблема становится особенно актуальной, если рассмотреть ее с точки зрения программной реализации. Открытость и расширяемость программной реализации семиотической модели подразумевает:

- модульность программной системы, т. е. реализацию отдельных моделей в виде независимых программных модулей – библиотек динамической компоновки;
- возможность встраивания новых программных модулей без внесения каких-либо изменений (включая перекомпиляцию и перекомпоновку) в ранее созданные программные модули.

Рассмотренный выше подход к организации взаимодействия модели семиотической сети не соответствует этим принципам. Во-первых, определение новой модели требует описания правил взаимодействия со всеми существующими моделями (т. е. каждая модель должна "знать" о существовании всех остальных), что нарушает принцип независимости отдельных моделей.

Интерфейс взаимодействия между моделями

В данной работе предлагается рассмотреть альтернативный подход к организации взаимодействия между моделями, ориентированный в первую очередь на возможность программной реализации. Данный подход предполагает создание некоторого единого (общего) интерфейса взаимодействия между моделями семиотической сети. Схематически данный способ представлен на рис. 2.

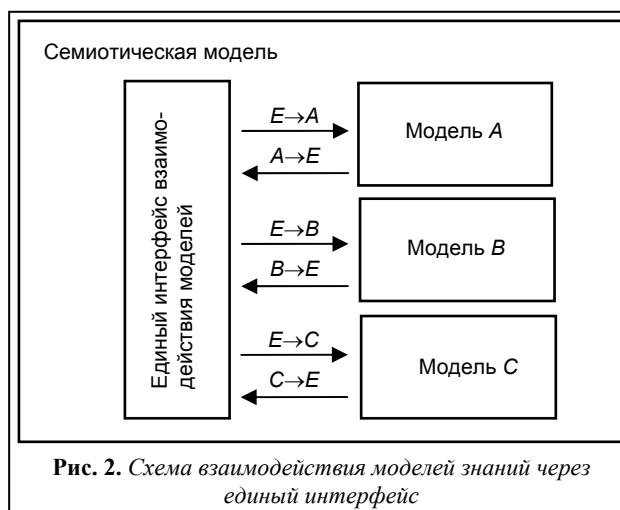


Рис. 2. Схема взаимодействия моделей знаний через единый интерфейс

Использование такого подхода позволяет решить две проблемы:

- во-первых, при определении новой модели (модели нового типа) количество требуемых описаний правил взаимодействия равно 2 вместо $2(N - 1)$, так как требуется определить лишь правила взаимодействия (перехода) между моделью и единым интерфейсом;

- во-вторых, таким образом, достигается независимость моделей (каждая модель "знает" только о едином интерфейсе и ничего не "знает" о наличии других моделей).

Во всех предыдущих рассуждениях использовались довольно абстрактные описания как самих математических моделей, составляющих семиотическую сеть, так и правил перехода между ними. Рассмотрим их более конкретно с точки зрения программной организации и реализации семиотической сети.

С точки зрения семиотической модели любая математическая модель, входящая в нее, представляет собой некий черный ящик, который содержит описание некоторой проблемной области и процедуры вывода на этом описании. Способы описания (и вывода) различны и зависят от типа математической модели. Например, семантические сети и продукционные системы оперируют совершенно разными методами и понятиями для описания той или иной проблемной области.

Тем не менее существуют некоторые базовые (общие) принципы, присущие всем (или, по крайней мере, очень широкому классу) моделям. Эти общие (характерные для всех) принципы и должны составить единый интерфейс (набор интерфейсов) взаимодействия моделей в рамках семиотической сети.

3. Модели знаний как компоненты семиотической модели

С точки зрения семиотической модели любая математическая модель, входящая в нее как некоторый черный ящик, предназначенный для выполнения некоторых функций. Какие это функции и каким обра-

зом они выполняются, определяется самой математической моделью (семиотическая модель об этом ничего не "знает").

Любая модель семиотической сети взаимодействует с другими моделями сети посредством обмена переменными. Каждая модель имеет определенный набор (массив) входных и выходных переменных. Входные и выходные переменные – это единственное средство взаимодействия моделей друг с другом.

С этой точки зрения, модель представляет собой объект, инкапсулирующий некоторые данные и методы работы с этими данными. Все объекты-модели наследуются (в смысле объектно-ориентированной технологии) от некоторого базового класса, который содержит только самую общую (абстрактную, характерную для всех типов моделей) информацию и методы работы с ней. При введении в семиотическую модель модели нового типа создается новый класс (описывающий эту модель), наследуемый от базового класса.

Базовый класс содержит только самую общую (характерную для всех типов моделей) информацию о модели, т. е. информацию о входных и выходных переменных модели.

Аналогично тому, как были введены интерфейсы предоставляющие единый подход для работы с разнотипными переменными (переменными разнотипных моделей) должны быть определены интерфейсы, определяющие единый подход для работы с разнотипными моделями.

Этапы создания и применения семиотической модели

1. Этап программной реализации предполагает:
 - программную реализацию математических моделей, составляющих семиотическую модель;
 - регистрацию созданных моделей.
2. Этап создания (конструирования) семиотической модели предполагает:
 - вставку созданных моделей в семиотическую модель;
 - настройки взаимодействия между отдельными моделями путем указания "протокола" передачи переменных между моделями.
3. Этап решения задач интеллектуальной поддержки принятия решений (работы) в семиотической модели.

Этап программной реализации состоит в программной реализации отдельных математических моделей, которые помимо реализации вычислительных функций, непосредственно связанных с функциональностью модели, также поддерживает описанный выше набор программных интерфейсов (т. е. "умеют" предоставить информацию о своих входных и выходных переменных).

Этап создания (конструирования) семиотической модели. После того как создана программная реализация для требуемых типов моделей семиотической сети, проблемная область моделируется экспертом

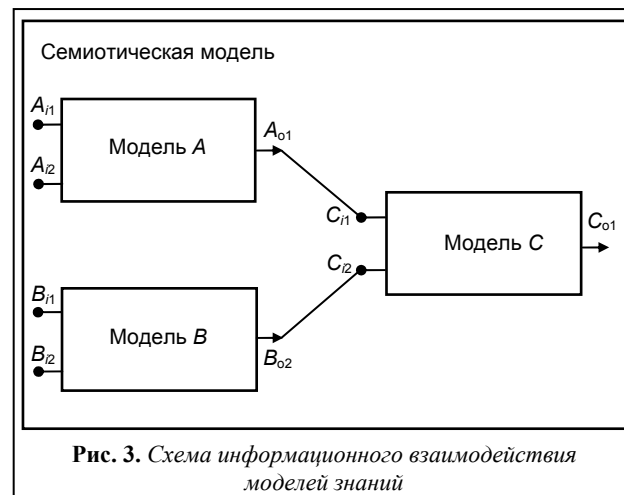


Рис. 3. Схема информационного взаимодействия моделей знаний

(специалистом по знаниям) с использованием средств, предоставляемых моделями, входящими в семиотическую сеть. Данный этап предполагает:

- определение типов моделей, необходимых для моделирования данной проблемной области (т. е. в определении типов моделей, которые будут составлять семиотическую сеть);
- определение входных и выходных переменных каждой модели семиотической сети;
- создание описания проблемной области (или ее части) на языках моделей семиотической сети;
- задание соответствия между входными и выходными переменными моделей, составляющих семиотическую сеть.

Конечным результатом этапа является семиотическая сеть, состоящая из набора разнотипных математических моделей, связанных между собой. Помимо связей, которые существуют между входными и выходными переменными моделей внутри семиотической сети, существуют также внешние входные и выходные параметры всей сети.

Входные параметры сети (не отдельной модели), т. е. те параметры, которые не рассчитываются в процессе работы сети, служат источником данных для сети (могут быть получены из базы данных). Множество входных переменных сети есть объединение всех входных переменных всех моделей сети, которые не связаны с выходными переменными моделей той же сети.

Множество выходных переменных сети есть объединение всех выходных переменных всех моделей сети. В схеме, приведенной на рис. 3, входными переменными сети являются A_{11} , A_{12} , B_{11} , B_{12} , а выходными – A_{01} , B_{01} , C_{01} .

Алгоритм взаимодействия моделей знаний в семиотической сети

Итак, семиотическая сеть SS состоит из связанных между собой моделей знаний, каждая из которых характеризуется набором входных и выходных параметров, а также внутренним алгоритмом функционирования каждой модели (поиска решения задач), преобразующим входные параметры в выходные. Поми-

мо этого, каждый входной "порт" модели знаний содержит некоторый внутренний флаг, характеризующий наличие входных данных. В момент запуска сети флаги всех портов сброшены. В момент поступления данных на порт флаг порта активизируется. Модель запускается на выполнение в тот момент, когда активизированы флаги всех входных портов модели. После того как модель "отработала", флаги всех ее входных портов сбрасываются. Результатом работы модели является вектор выходных параметров, которые в свою очередь могут быть входными для других моделей сети.

При передаче разнотипных переменных между входами и выходами моделей сети при необходимости происходит автоматическое преобразование переменных к нужному (требуемому моделью) типу на основании информации о своем типе предоставляемой переменной с помощью описанных выше интерфейсов.

Такое функционирование семиотической сети во многом схоже с функционированием сети Петри [8].

4. Логико-вычислительные средства для автоматизации процессов поддержки принятия решений

Известно, что основными факторами любого управления, присущими всем системам, в том числе и семиотическим, являются:

- информация о состоянии объекта управления I ;
- нештатная ситуация, требующая принятия решений $S(I)$;
- цель управления $Z(I)$;
- SM – локальные и глобальная семиотические модели знаний (область поиска решений);
- Φ -множество правил вывода (методов поиска решений);
- воздействия на объект управления – управляющие решения (РЕШ).

Взаимосвязь между этими факторами определяется выражением вида

$$\text{РЕШ} = \varphi(S(I), SM, \Xi, Z(I)).$$

Рассматриваемый класс сложных технологических объектов характеризуется тем, что "решающим элементом" в контурах принятия решений является человек. Это он в настоящее время реализует алгоритм оценки состояния, выявления и анализа аномальных состояний (ситуаций), диагностики, прогнозирования и оперативного поиска управляющих решений, осуществляя, кроме того, согласование решений с лицами оперативного персонала своего уровня и с вышестоящими инстанциями.

Логико-математические процессы автоматизации должны реализовать модельные отображения вида

$$\varphi: (S(I), SM, \Xi, Z(I)) \rightarrow \text{РЕШ}.$$

Для этого SS должна уметь "воспринимать и понимать" входную информацию. Последнее достигается установлением связи между объектом управления и человеком, с одной стороны, и системными средствами распознавания, классификации ситуаций (аномальных состояний, требующих вмешательства в процесс управления) и планирования – выполнения вычислений – с другой. Эта связь носит условно-событийный характер, определяемый схемой, приведенной на рис. 4.

Каждое событие трактуется в SS как ситуация управления. В ее описании обязательно присутствуют условия наступления определяющего ее события и информация, позволяющая относить каждую из ситуаций к некоторому классу ситуаций, для которых существует постановка задач, а возможно, и алгоритм поиска решений. Будем различать пять классов ситуаций:

- 1) ситуации срабатывания защит, однозначно связывающие условия наступления событий с постановкой задачи и алгоритмом выдачи управляющих команд;
- 2) ситуации регулирования, однозначно связывающие условия наступления событий с постановкой задачи и алгоритмом поиска ее решения;
- 3) типовая ситуация принятия решений, однозначно связывающая условия наступления событий с постановкой задачи и алгоритмом поиска ее решения;
- 4) проблемная ситуация принятия решений, однозначно связывающая условия наступления событий с постановкой задачи, существование алгоритма поиска решения которой необходимо доказать;
- 5) экспертная ситуация принятия решений, задающая гипотетические условия наступления событий и связи с возможными постановками задач, существование алгоритмов поиска решений которых необходимо доказать.

Первые два из этих классов ситуаций относятся к задачам, решаемым в контурах технологических защит и автоматического регулирования, остальные – в контурах оперативного принятия решений. При этом для задач, решаемых в первых трех классах ситуаций, связь между событием, задачей и алгоритмом описывается на основе информации, изложенной в регламентных материалах по эксплуатации оборудования, при проектировании SS .

В четвертом классе задач связь между событием и

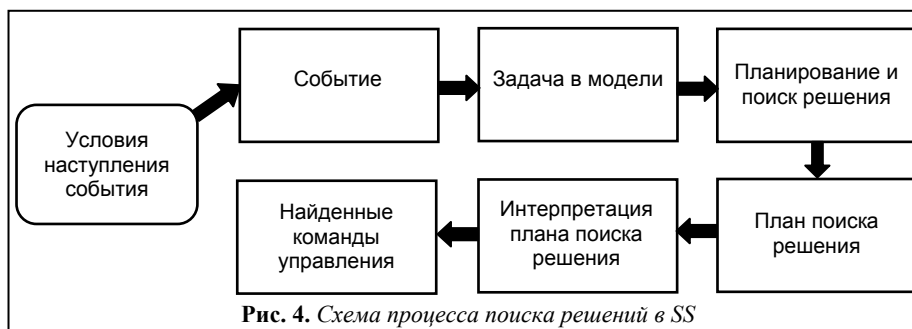


Рис. 4. Схема процесса поиска решений в SS

задачей может быть описана экспертом только после появления опыта эксплуатации конкретного вида оборудования. Только опыт позволяет описать условия наступления событий и возникающие при этом постановки задач поиска решений. Выяснение существования алгоритма поиска решения и его построение являются функцией SS .

Пятый класс ситуаций введен для обеспечения моделирования гипотетических условий наступления событий на технологическом объекте, которые не предусмотрены регламентными материалами и пока не встречались в практике управления.

Экспертные ситуации в основном характерны для предаварийных и аварийных состояний объекта управления, особенно в режимах взаимного наложения предаварийных и аварийных условий наступления событий. Моделирование таких условий, с одной стороны, позволяет определить новый круг задач, который необходимо решать SS , с другой стороны, позволяет проверить полноту модели знаний системы.

Рассмотренная цепочка преобразований (см. рис. 4) описывается на языке, основными понятиями которого являются ситуации, классы ситуаций, модели, задачи на модели, управление поиском в модели, состояния объекта управления, команды управления. В качестве правил вывода в языке задаются правила, определяющие соответствие между ситуациями и классами ситуаций, классами ситуаций и задачами, задачами и моделями, правилами планирования и вычисления в моделях. Правила установления соответствия определяют классы постановок задач в базе знаний. Правила планирования и вычислений в базе знаний определяют классы методов решения возникающих задач. Именно эти правила позволяют организовать в SS поиск ответов на вопросы:

какие процессы вычислений для некоторого класса задач разрешимы в модели?

существует ли процедура планирования, строящая для любого разрешимого процесса в базе знаний или их совокупности алгоритм выполнения этого процесса?

существует ли алгоритм, позволяющий для любого процесса в модели или их совокупности указать, разрешим ли он?

Формализуем цепочку преобразований (см. рис. 4). Пусть X – множество наблюдаемых переменных, характеризующих текущее состояние объекта управления, а $X\uparrow$ (переменная X вышла за верхнюю уставку) и $X\downarrow$ (переменная X вышла за нижнюю уставку) – множества элементов условно-постоянной информации, характеризующих соответственно верхнюю и нижнюю допустимые границы изменения переменных из X .

Пусть $s(x, x\uparrow, x\downarrow, \Delta t)$, $x \in X$, $x\uparrow \in X\uparrow$, $x\downarrow \in X\downarrow$ – состояние объекта управления на некотором временном интервале Δt . Тогда для каждого временного интервала множество X можно разбить на классы эквивалентности, задав отношения $P(x, x\uparrow, x\downarrow)$, определяющие вы-

ходы значений переменных множества X за верхние и нижние допустимые границы, т. е. условия наступления события. Множество отношений $\{P(x, x\uparrow, x\downarrow)\}$, определяющих аномальные состояния объекта управления, будем рассматривать как ситуации управления в классе задач стабилизации параметров объекта. Содержательно ситуация несет для человека информацию о том, что какие-то параметры вышли из допустимых границ и требуют нахождения и реализации некоторых действий (команды управления). Последние должны позволить перевести объект в новое состояние, в котором контролируемые параметры находились бы в норме. Пусть такой переход возможен только за счет оперативного воздействия одной или совокупности команд управления из множества R . Тогда для каждого класса ситуаций управления, определенного в SS отношением $P(x, x\uparrow, x\downarrow)$, можно сформулировать задачу, которую необходимо решить.

Постановку задачи опишем следующим утверждением:

"По условиям задачи $\{x, x\uparrow, x\downarrow\}$ для переменных множества X , удовлетворяющих этим условиям, найти команды из R , также удовлетворяющие этим условиям".

Для того чтобы каждая из поставленных в SS задач была разрешима, необходимо обеспечить истинность рассмотренного выше утверждения. Если это осуществимо для каждого из определенных в SS классов задач, то присущие им аналогичные утверждения будем называть теоремами о существовании решения задач.

Процесс планирования вычислений и поиска решений будем рассматривать как процесс доказательства теоремы существования. Схема вывода, полученная в процессе доказательства, будет являться планом поиска решений. Если задачу в базе знаний описать предикатом вида $Q(x, x\uparrow, x\downarrow, r)$, $r \in R$, то теорема существования решения может быть записана в виде

$$(\forall x)(P(x, x\uparrow, x\downarrow) \rightarrow (\exists r)(Q(x, x\uparrow, x\downarrow, r))).$$

Существование решения этой задачи необходимо доказать, опираясь на базу знаний.

Процесс доказательства состоит в выяснении существования логического следования теоремы из утверждений, описывающих модель оперативного мышления, путем последовательного применения правил вывода.

Структура процесса планирования вычислений представляет собой последовательность шагов и приведена на рис. 5.

Эта последовательность шагов полностью применима только к четвертому и пятому классам ситуаций. Для ситуаций, относящихся к первым трем классам, шаги построения планов доказательства выполняются только на этапе проектирования. На этапе функционирования сразу же за шагом распознавания и классификации следует шаг интерпретации.

Естественным требованием к базе знаний при этом является наличие в ней, помимо информации из мо-

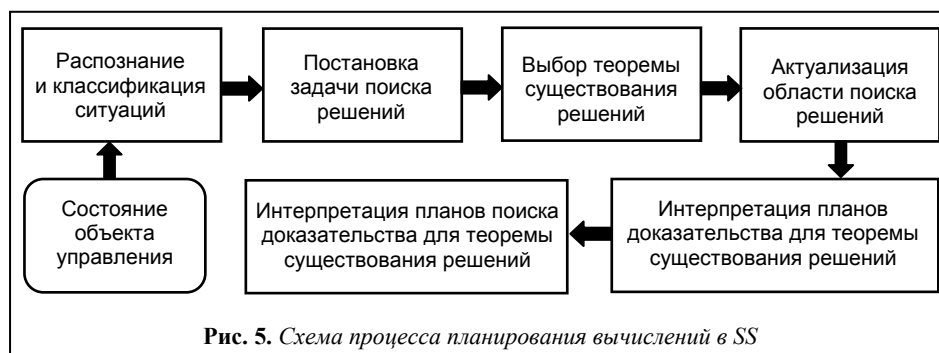


Рис. 5. Схема процесса планирования вычислений в SS

плана доказательства в декларативном представлении;
 – поиск и построение локального плана доказательства в процедурном представлении;
 – поиск и построение глобального плана доказательства в декларативно-процедурном представлении.

Если существуют локальные планы доказательства теорем существования

решений в декларативном и процедурном представлении информации в базе знаний, то существует и глобальный план доказательства.

решений в декларативном и процедурном представлении информации в базе знаний, то существует и глобальный план доказательства.

<Нештатная ситуация → Задача в модели>

зависит от способа описания информации в базе знаний и применяемых правил вывода.

Для реализации вышеописанного алгоритма в SS организована двухуровневая схема планирования вычислений.

Существующие подходы к описанию базы знаний ориентированы на декларативное или процедурное описание. Первое из них содержит экспертные сведения о способах поиска управляющих команд в пространстве состояний; второе – сведения о совокупности подзадач, композиция которых может привести к решению любой из описанных в SS задач.

На верхнем уровне (глобальное планирование) выполняется последовательность шагов, приведенная на рис. 6.

На нижнем уровне – локальное планирование осуществляется последовательностью шагов, представленных на рис. 7.

Интерпретация локальных и глобальных планов доказательства теорем существования решений для поставленных задач позволяет построить вычисли-

Наличие двух типов описаний позволяет рассматривать базу знаний как композицию двух областей, поиск решений в которых ведется разными методами. В этом смысле уместно говорить о планировании вычислений для построения планов доказательства теорем существования решений в каждой из этих областей (локальное планирование). Однако в SS существуют и такие классы задач, для решения которых необходимо использовать планирование вычислений как в области декларативных описаний, так и в области процедурных описаний (глобальное планирование).

Заметим, что теоремы существования решений явно не содержат указаний на область поиска доказательства, поэтому необходим метод глобального планирования. Этот метод позволяет актуализировать те или иные области локального поиска доказательства теоремы существования решений в зависимости от постановки задачи и полноты описания базы знаний. В этом смысле задача доказательства теоремы существования решений распадается на три подзадачи:

– поиск и построение локального

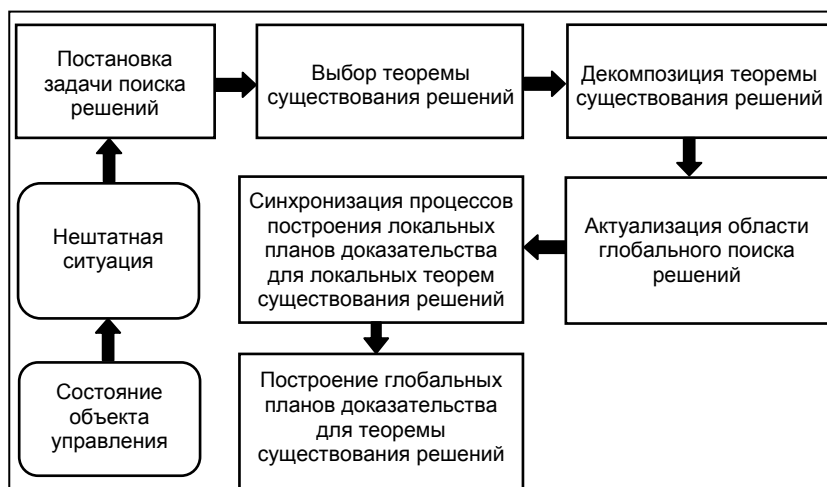


Рис. 6. Схема глобального планирования процесса поиска решений

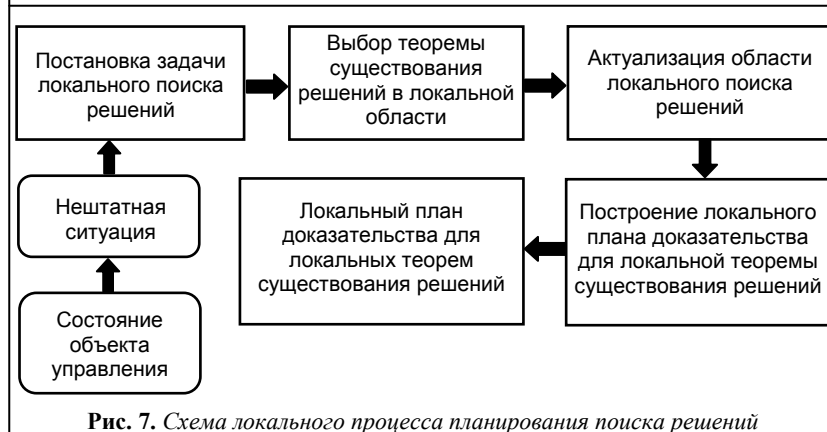


Рис. 7. Схема локального процесса планирования поиска решений

тельную последовательность, состоящую из правил вывода.

Элементы множества правил вывода $\{ПВ\}$ разбиты в SS на два подмножества. Принадлежность правил к тому или иному подмножеству определяется в зависимости от того, как выбираются и как применяются эти правила в процессе доказательства теорем о существовании решений.

Первое подмножество содержит правила общего вывода в логике предикатов первого порядка; второе подмножество – правила логического вывода, описанные как элементы продукционной системы. Действительно, в логике предикатов нет ограничений на применимость правил вывода. К любой уже выведенной формуле можно применять любое правило вывода, если данная формула допускает применение этого правила. В продукционных системах имеются дополнительные условия на применимость того или иного правила вывода. Эти условия применимости могут меняться в процессе функционирования продукционной системы в зависимости от получения той или иной информации во время вывода.

Наличие ПВ первого множества позволяет находить в SS все возможные команды управления за сравнительно длительный интервал времени. Поэтому найденные команды применяют для целей тестирования и проверки на полноту баз знаний. ПВ второго множества позволяют в темпе с процессом управления находить достаточное для управления количество команд управления.

Доказательство теорем существования решений для всех возможных постановок задач в базе знаний не является самоцелью, а служит основой для автоматизации процесса дедуктивного синтеза программы поиска решений из набора подпрограмм, находящихся в базе знаний. При этом путь от анализа входной информации к получению управляющих решений выглядит так, как он представлен на рис. 8.

Если в SS описаны правила распознавания и класси-

фикации ситуаций и постановки задач поиска решений в базе знаний, то специальный программный комплекс – планировщик вычислительного процесса – автоматически осуществит выбор теоремы существования решения, соответствующей задаче, построит план доказательства этой теоремы и синтезирует на основе данного плана программу получения управляющих решений. Далее данная программа настраивается на текущие значения параметров, характеризующих состояние объекта управления, и по ним рассчитывает регулирующие воздействия.

Приведенная на рис. 8 схема планирования вычислений характерна для всех рассматриваемых классов ситуаций, с той лишь разницей, что пути организации вычислительного процесса доказательства будут различными для задач, определяемых разными классами ситуаций. Одной из целей организации планирования

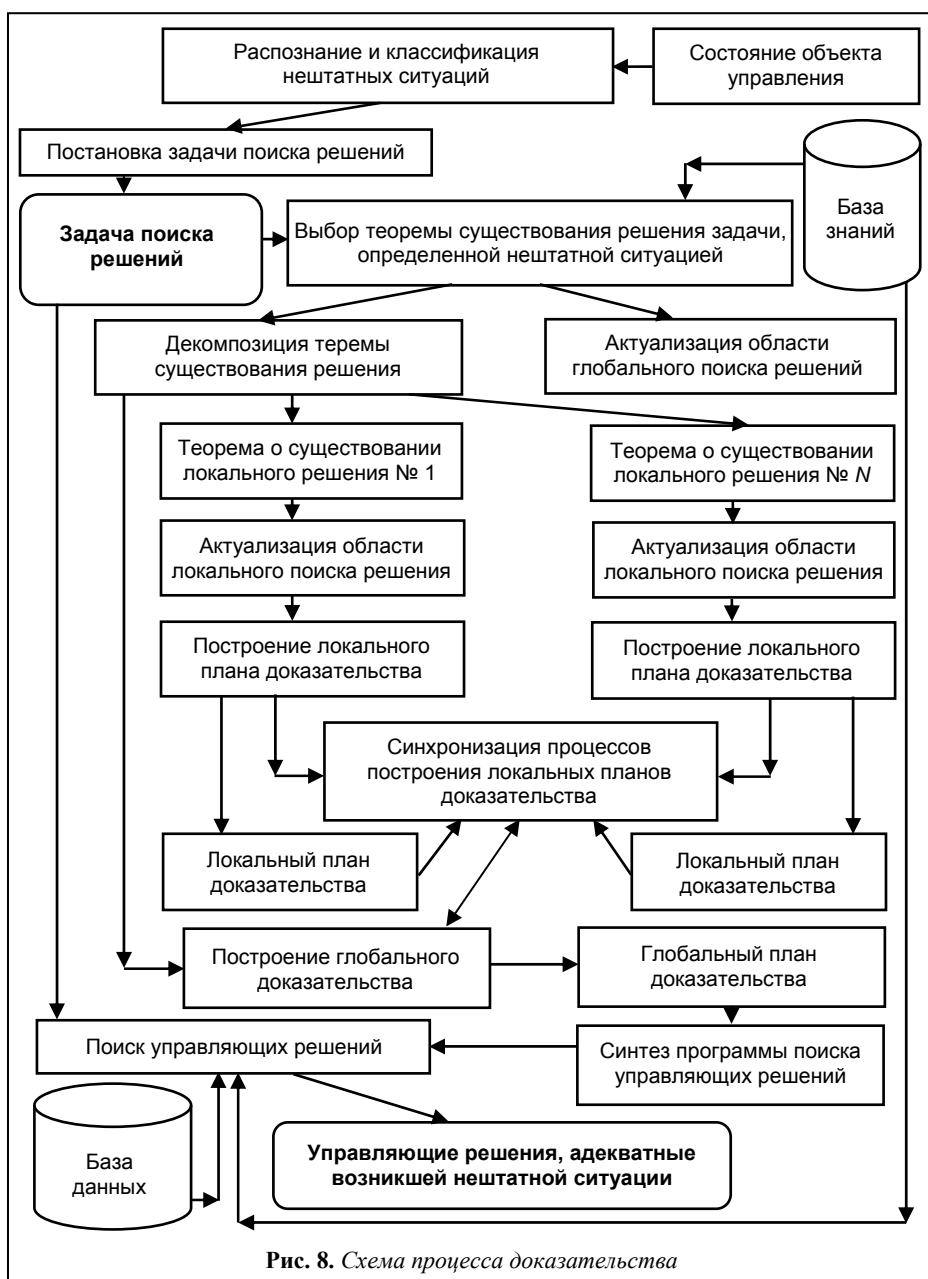


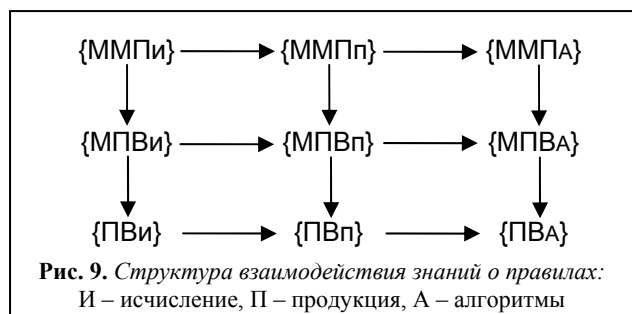
Рис. 8. Схема процесса доказательства

вычислений в *SS* при ее внедрении и эксплуатации является выявление и перевод как можно большего числа проблемных и экспертных ситуаций на уровень типовых ситуаций управления и защит. Наличие такого свойства в перспективной системе позволит в период адаптации к объекту управления эволюционно пройти от информационной стадии развития к управляюще-информационной стадии в каждом из контуров управления.

В процессе адаптации *SS* должна уметь пополнять свои знания о проблемной области управления и уточнять имеющиеся правила вывода и условия их применения.

Для этого в системе имеются средства введения метаправил вывода (МПВ). МПВ рассматривают в *SS* как информацию (правила над правилами), определяющую способ использования правил более низкого уровня. При этом не ограничиваются введением одного уровня метаправил. Для описания междууровневой и внутриуровневой синхронизации и координации процесса построения глобального плана доказательства теорем существования решения введены метаметаправила (ММП). Содержательно метаправила МПВ позволяют изменять правила вывода ПВ или переводить их из одного класса правил в другой. Например, введение условий применимости переводит правило из класса, относящегося к логическому исчислению, в класс, принадлежащий продукциям, а дополнительное уточнение этих правил – в класс, принадлежащий алгоритмам (программам, на базе которых строится синтез программ).

Структура взаимодействия знаний о правилах приведена на рис. 9.



5. Концепция информационной системы интеллектуальной поддержки принятия решений со средствами автоматической оценки ситуаций, их диагностики и поиска управляющих решений

Учитывая множество функций, возлагаемых на системы интеллектуальной поддержки принятия решений семиотического типа, существующие подходы к автоматизации процессов принятия решений, а также наличие лиц, принимающих решения (ЛПР), рассмотрим концепцию систем человеко-машинного управления, умеющих автоматически решать задачи оценки текущей аномальной ситуации, ее диагностики, прогнозирования направления развития аномалии,

поиска управляющих решений, адекватных возникающим аномалиям.

Такие системы, обладающие "собственными знаниями" и позволяющие автоматически или путем общения с человеком находить управляющие решения, адекватные возникающим проблемным (нештатным) ситуациям, или вырабатывать и обосновывать логические факты, не заложенные априори, вести диалог с ЛПР, будем относить в дальнейшем к классу систем принятия решений с интеллектуальными механизмами информационной поддержки оператора-технолога (СПРИНТ).

В описанном выше смысле рассматриваемый класс систем интеллектуальной поддержки оператора-технолога может быть отнесен к классу систем автоматизации процессов принятия решений [9].

В настоящее время не существует законченной общей теории систем подобного класса, поэтому и нет общей модели таких систем, обладающих знаниями в области управления и способными его использовать для организации целенаправленного поведения системы управления с целью достижения заданных критериев качества.

Вместе с этим, анализ задач, которые должны решаться в системах типа СПРИНТ, показывает, что формальный аппарат, описывающий процессы распознавания ситуаций, выработки и принятия решений в быстроменяющейся обстановке с элементами неопределенности, должен быть чрезвычайно "гибким". Процессы выработки и принятия решений базируются не только на количественных характеристиках, но и на факторах, не всегда имеющих количественные меры (психологические, моральные и т. д.).

Поэтому подготовку информации для принятия решений следует рассматривать как творческий акт выбора из совокупности возможных решений, в котором количественные факторы сочетаются с эвристическими способностями, заложенными в ЭВМ, формирующей решения. То есть решения формируются на основе двух составляющих сторон выработки и принятия решений – формальной и творческой. Особое внимание нужно уделить самой процедуре принятия решений, т. е. важно знать, какие составляющие процесса выработки и принятия решений должны контролироваться ЛПР и какие могут быть выполнены ЭВМ.

Важное место занимает проблема коммуникации человека и машины. Эта проблема имеет две стороны – удовлетворение информационных потребностей по имеющейся в системе информации и участие в процедуре выработки и принятия решений. Естественным требованием к средствам представления информации является их информативность и удобовоспринимаемость, к языку общения – приближенность к языку профессиональной лексики и его "слэнгам". Форма общения должна быть диалоговой.

Проблема обучения или адаптации разработанной системы к предметной области управления требует разработки специальной процедуры, позволяющей

выдавать информацию, представленную формально (алгоритмически) и неформально (экспертно). Такая процедура должна носить человеко-машинный характер и быть применимой для большого класса объектов.

Проблема проектирования и генерации разных версий алгоритмического, информационного, модельного и программного обеспечения систем принятия решений требует разработки специальной человеко-машинной технологии проектирования в рамках этого класса систем.

Учитывая характер деятельности систем типа СПРИНТ, основанный на способности адаптации и построении целенаправленного поведения, будем различать в системе два вида информации – информацию, обеспечивающую целенаправленное поведение системы СПРИНТ путем организации процессов распознавания, поиска и принятия решений, и информацию, которая является элементами переработки со стороны вышеназванных процессов.

Первый тип информации назовем знаниями системы о предметной области управления. К ним относятся модели, задачи, алгоритмы. Второй тип информации назовем данными о состоянии системы, объекта и среды управления. К ним относятся параметры, характеризующие текущее и историческое состояние системы, объекта и среды управления и области определения этих параметров.

Анализ, проведенный в работах [9–11], позволил положить в основу концепции систем типа СПРИНТ следующие понятия:

1) глобальная логическая модель знаний (ГЛМЗ) как коллектив моделей и способов их использования для организации процессов целенаправленного распознавания аномальных ситуаций, их диагностики, прогнозирования, выработки и принятия управляющих решений;

2) локальная логическая модель знаний (ЛЛМЗ);

3) глобальная логическая модель данных (ГЛМД) как область интерпретации глобальной и локальной логических моделей знаний, которая рассматривается в виде структурированного и упорядоченного динамического множества атрибутов, характеризующих параметры системы, объекта и среды управления;

4) множество решателей задач в рамках каждой формальной модели FS как средство организации локального поиска решения для каждой проблемной ситуации, описываемой данной моделью;

5) глобальный координатор решения задач $\phi(SS)$ в ГЛМЗ;

6) глобальный интерпретатор результатов решения задач в ГЛМД;

7) коллектив системных аналитиков-экспертов, которые, используя средства коммуникации, могут определить и описать элементы глобальной логической модели знаний и область ее интерпретации в объеме, достаточном для решения любой проблемной ситуации или задачи принятия решений.

Отношения коммуникации и интерпретации между этими понятиями позволили построить концептуальную модель СПРИНТ, приведенную на рис. 10.

Эта модель по способу описания и представления информации о проблемной области управления сходна с моделью многоуровневого представления информации в системах управления базами данных, что позволило определить направление реализации системы в целом. При этом на учет были взяты следующие характеристики модели:

- экспертность – как основа для формирования целей управления, гипотез управления, моделей поиска решений, правил поиска и принятия решений;

- ассоциативность – как основа для автоматического накопления, обобщения информации и адаптации системы на предметную область управления;

- многоальтернативность – как основа для отбора всех возможных путей поиска решений;

- семиотичность – как основа для разработки механизмов интеграции разнородной информации об объекте и среде управления;

- коммуникабельность – как основа для реализации диалоговых средств общения системы с ЛПР;

- виртуальность – как основа для отображения глобальности информации, которая характеризуется территориальной разобщенностью и многоуровневостью источников получения, хранения и потребления информации;

- алгоритмическая разрешимость – как основа для реализации модели СПРИНТ в вычислительных средах средств компьютеризации.

Полученная модель обладает рядом новых свойств, например, коллектив экспертов – это одновременно и средство решения возникающих перед системой задач, и методология конструирования и реализации таких систем. Такой коллектив обладает как формальными, так и неформальными знаниями о предметной области управления. При этом каждый эксперт выполняет в коллективе определенную функцию, что позволяет легко сконструировать представление и использование знаний, давая при этом эксперту возможность как использовать (создавать) определенную локальную логическую модель знаний, так и иметь доступ к глобальной модели данных. Наличие экспертов позволяет помимо процедур распознавания, выработки и принятия решений строить:

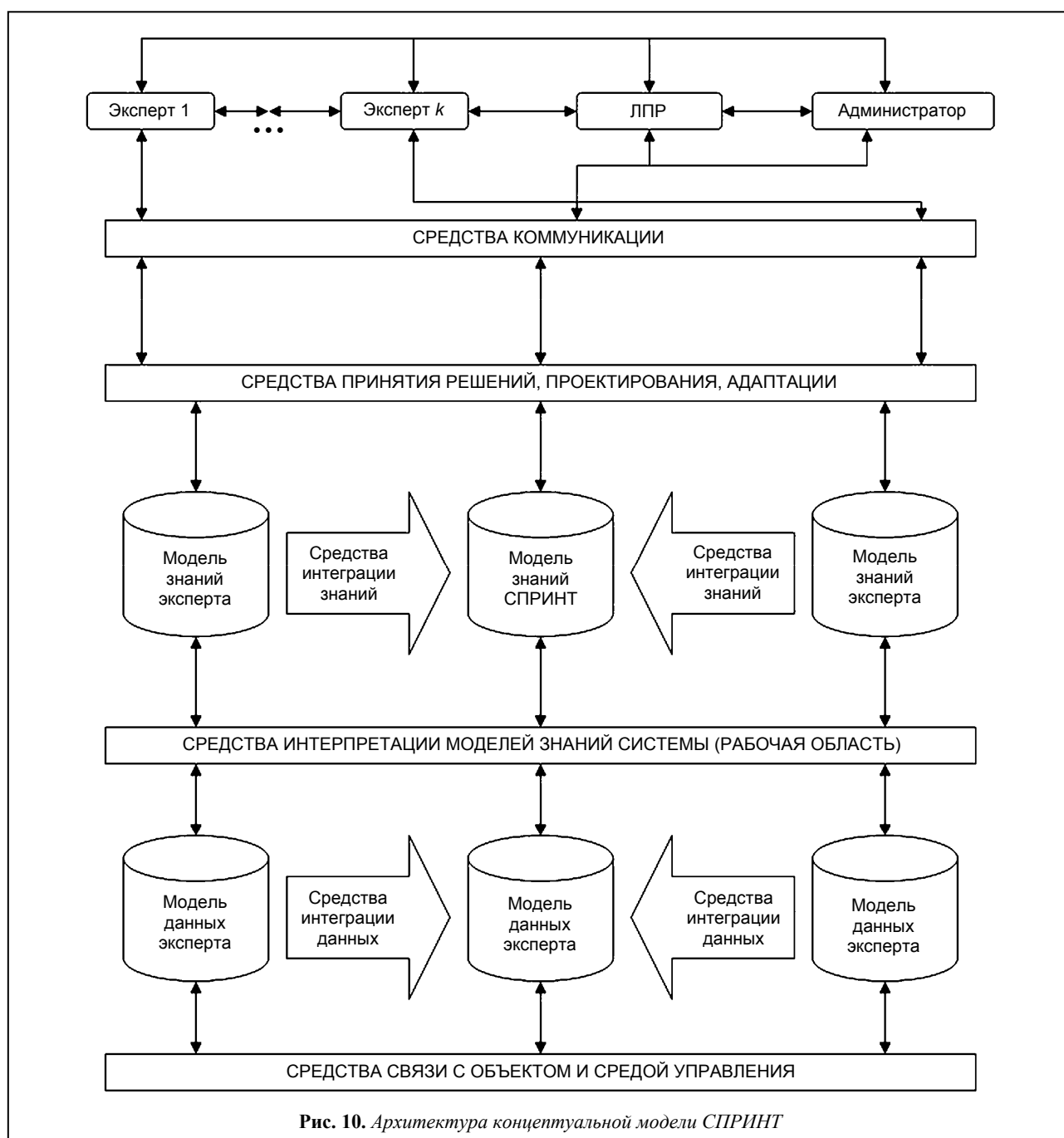
- процедуры выявления противоречивых знаний, используя динамически меняющиеся группы других экспертов;

- создавать экспертные модели выработки решений для разных классов проблемных ситуаций;

- моделировать любые комбинации централизованного и децентрализованного принятия решений;

- достигать большей общности, что позволяет реализовать различные методы решения задач.

Кроме того, в таком коллективе есть возможность формализации связи между экспертами и построения на этой основе стандартных средств коммуникации.



Эти средства позволяют организовать различные режимы взаимодействия – от явного обращения одного эксперта к другому до неявного обращения, когда адресат определяется по реализуемой им функции.

Наличие коллектива алгоритмов, задач, моделей и знание способа их использования в тех или иных ситуациях выработки и принятия решений позволяют разработать единые средства описания такой информации и организации их автоматического использования системой. В качестве таких средств используются средства лингвистической и программной поддержки. К ним относятся языки определения логической модели знаний (ЯОЗ) и манипулирования элементами

логической модели знаний (ЯМЗ) и их программные процессоры.

В основе ЯОЗ используются два формализма:

$$\text{ЯОЗ1} = \{W, M, \alpha(T), \beta(P), \gamma(A), \sigma(R)\},$$

где ЯОЗ1 – язык описания локальной модели знаний семиотической модели;

$W = \{pT, pP, pA\}$ – множество правил описания формальных локальных моделей знаний, элементами которой являются:

pT – правила описания множества термов модели знаний;

pP – правила описания множества правил построения правильно построенных формул модели знаний;

pA – правила описания множества аксиом;
 M – множество локальных моделей знаний, элементов семиотической модели.

Правила изменения формализма описания локальной модели знаний включают:

$\alpha(T)$ – правила изменения элементов T ;

$\beta(P)$ – правила изменения элементов P ;

$\gamma(A)$ – правила изменения элементов A ;

$\sigma(RL)$ – правила изменения правил вывода RL в локальной модели знаний RL ;

$\eta(R)$ – правила изменения правил вывода R в глобальной модели знаний (семиотической модели).

$$ЯОЗ2 = \{M, \Xi, SM, \varphi(FS, SM), \varepsilon(\Xi), \mu(\varphi)\},$$

где M – множество локальных моделей знаний; Ξ – множество отношений между M ; $Q(M)$ – множество правил вывода в сети M (глобальной семиотической модели знаний), определенное на множестве Ξ .

Правила изменения формализма описания глобальной семиотической модели включают:

$\varepsilon(\Xi)$ – правила изменения элементов Ξ ;

$\mu(\varphi)$ – правила изменения элементов Q .

Объединяя ЯОЗ1 и ЯОЗ2, получаем язык описания знаний для семиотической модели:

$$ЯОЗ = \langle ЯОЗ1 \cup ЯОЗ2 \rangle.$$

Определение знаний предполагает введение новых типов информации, таких, как утверждение, зависимость, модель. Манипулирование знаниями основывается на планировании деятельности системы путем использования механизма целеполагания.

Язык манипулирования знаниями (ЯМЗ) содержит множество правил вывода (поиска решений), что позволяет организовывать решение задач на моделях знаний, описанных в семиотической модели.

В основе языка используются два формализма. Формализм вывода в локальной модели знаний определяется выражением вида

$$ЯМЗ1 = \langle M, \varphi_i, РЕШ_i \rangle,$$

где φ_i – множество правил вывода в локальных моделях знаний; $РЕШ_i$ – множество локальных (одномоделных) решений.

Формализм вывода в глобальной модели знаний определяется выражением

$$ЯМЗ2 = \langle M, \Xi, SM, \varphi, РЕШ \rangle,$$

где φ – множество правил вывода в глобальной модели знаний – семиотической модели (SM); $РЕШ$ – множество решений, найденных в глобальной модели знаний (семиотической модели); $РЕШ_i \in РЕШ$.

Объединяя ЯМЗ1 и ЯМЗ2, получаем язык манипулирования знаниями для семиотической модели:

$$ЯМЗ = \langle ЯМЗ1 \cup ЯМЗ2 \rangle.$$

Под манипулированием знаниями понимается способ (метод) автоматического поиска решений в глобальной базе знаний средствами φ для каждого из

классов ситуаций S , возникающих на объекте управления:

$$РЕШ = \varphi(S, M, \Xi, SM, Z),$$

где S – множество классов нештатных ситуаций, возникающих на энергоблоке и требующих поиска решений, адекватных возникшей ситуации (постановка задачи поиска решений);

M, SM – локальные и глобальная модели знаний (область поиска решений);

Z – цель управления;

φ – множество правил вывода (методов поиска решений);

$РЕШ$ – множество найденных решений.

Средства ЯМЗ представляется набором методов задач для поиска решений и включают следующие правила вывода (операторы):

– транзитивное замыкание графа (сети) [9];

– управление разметкой в сетях Петри [8];

– резолюция (семантическая, линейная, гиперрезолюция...) в логике предикатов первого порядка [4, 5];

– прямая волна в вычислительных моделях, представленных двудольным графом [5];

– обратная волна в вычислительных моделях, представленных двудольным графом [13];

– прямая и обратная волны в вычислительных моделях, представленных двудольным графом [14];

– раскраска графа в семантических сетях [15].

ЯМЗ реализуется средствами семиотического решателя задач [9, 10, 14]. Многообразие логических моделей данных (МД) в СУБД является объективной закономерностью развития теории и практики информационных систем, отражающее разнообразие элементов данных и их свойств и отношений в реальных предметных областях. Это предопределено предшествующим развитием языков и методов определения множеств объектов, их свойств и операций над ними. Каждая МД определяется семантикой языка определения данных (ЯОД) и языка манипулирования данными (ЯМД). Принято различать иерархические, сетевые, реляционные МД, а также МД, основанные на бинарных отношениях, семантических сетях, на понятии расширенных множеств и т. д. [16, 17].

Классы моделей данных существенно различаются выразительностью средств представления свойств и отношений элементов, уровнем предусмотренных операций над базой данных (БД) и эффективностью реализации таких операций при существующих средствах обработки данных.

Различие МД обычно оказывается принципиальным, так что выбор той или иной МД при создании БД и решении конкретных задач может дать решающие преимущества.

Существование различных МД привело на практике к разработке многоуровневой архитектуры СУБД. Эта структура сочетает единственную концептуальную модель с произвольным числом внутренних и внешних МД.

Всё это порождает проблему отображения одной модели в другую. В случае системы СПРИНТ это – отображение ГЛМЗ в ГЛМД и создание средств интерпретации логической модели данных в модели знаний.

Рассмотренная концепция хорошо согласуется с природой сложных человеко-машинных систем принятия решений. Она позволяет, используя знания:

- экспертов-управленцев, строить модели распознавания состояний объекта и среды управления, классификации состояния, целеполагания, формирования гипотез, выработки и принятия управляющих решений;
- экспертов-постановщиков локальных задач управления, строить функционально-полный коллектив вычислительных алгоритмов, характеризующий конкретную область управления;
- экспертов-программистов вычислительных алгоритмов, "наполнить" программную систему конкретным процедурным содержанием;
- экспертов-конструкторов систем принятия решений проектировать и генерировать программное обеспечение системы и организовывать ее проблемную ориентацию.

Резюмируя сказанное, можно заключить, что рассмотренная концепция полностью отвечает проблемам человеко-машинного управления сложными технологическими объектами, методам и теории построения больших управляющих программных комплексов.

Выводы

Системы семиотического типа позволяют интегрировать в себе опыт работ по созданию систем интеллектуальной поддержки принятия решений во всех областях человеко-машинного управления.

ЛИТЕРАТУРА

1. Поспелов Д.А. Моделирование рассуждений. Опыт анализа мыслительных актов. – М.: Радио и связь, 1989. – 182 с.
2. Смальян Раймонд. Теория формальных систем / пер. с англ.: под ред. Н.А. Шанина. – М.: Наука, 1981.
3. Ситуационные советующие системы с нечеткой логикой / А.Н. Мелихов [и др.]. – М.: Наука, 1990. – 272 с.
4. Нильсон Н. Искусственный интеллект: методы поиска решений / пер. с англ. – М.: Мир, 1973. – 270 с.
5. Ковальски Р. Логика в решении проблем. – М.: Наука, 1990.
6. Чень Ч., Ли Р. Математическая логика и автоматическое доказательство теорем / пер. с англ.: под ред. С.Ю. Маслова. – М.: Наука, 1983. – 360 с.
7. Ефимов Е.И., Поспелов Д.А. Семиотические модели в задачах планирования для систем искусственного интеллекта // Изв. АН СССР. Техническая кибернетика. – 1977. – № 5. – С. 60–68.
8. Питерсон Дж. Теория сетей Петри и моделирование систем. – 1984. – 264 с.
9. Баилюков А.А., Еремеев А.П. Экспертные системы поддержки принятия решений / под ред. А.Ф. Дьякова. – М.: МЭИ, 1994. – 2134 с.
10. Баилюков А.А. Проектирование систем принятия решений в энергетике. – М.: Энергоатомиздат, 1986.
11. Попов Э.В. Алгоритмические основы интеллектуальных роботов и искусственного интеллекта. – М.: Наука, 1976.
12. Зыков А.А. Основы теории графов. – М.: Вузовская книга, 2004. – 664 с.
13. Кахро М., Калья А., Тыгу Э. Инструментальная система программирования ЕС ЭВМ ПРИЗ. – М.: Финансы и статистика, 1988. – 181 с.
14. Тыгу Э.Х. Концептуальное программирование. – М.: Наука, 1984.
15. Вагин В.Н., Еремеев А.П. Параллелизм в продукционных моделях представления знаний // Изв. РАН. Техническая кибернетика. – 1994. – № 1. – С. 48–55.
16. Date C.J., Darwen Hugh, Lorentzos Nikos. Temporal Data and the Relational Model. – Morgan-Kaufmann Publishers, 2002.
17. Дейт К.Дж. Введение в системы баз данных / пер. с англ. – М.: Вильямс, 2005.