

## ПРЕДСТАВЛЕНИЕ ПРОЦЕССА РЕШЕНИЯ РАСЧЕТНО-ЛОГИЧЕСКИХ ЗАДАЧ УПРАВЛЕНИЯ С ИСПОЛЬЗОВАНИЕМ СЕТЕВОЙ МОДЕЛИ

к.т.н. Д.Э. Двухглавов, В.А. Затхей  
(представил проф. Б.Н. Судаков)

*Рассматривается структура сетевой модели для представления знаний о процессе решения типовых задач управления.*

Анализ задач, решаемых в различных управляющих органах, позволяет утверждать, что они являются расчетно-логическими задачами. Типовыми задачами данного класса являются: задача определения очередности уничтожения целей по важности и задача отбора огневых источников для воздействия по целям, которые решаются на командных пунктах различных родов войск, задача распознавания типа воздушного судна, решаемая на пунктах управления полётами авиацией. Можно привести задачи в аналогичной постановке, которые решаются в органах управления производством, транспортом и т.д. При этом использование новых технологий обработки информации и возможностей современной вычислительной техники позволяет решать подобные задачи автоматизировано.

Можно выделить следующие типовые этапы решения задач данного класса:

- ввод лицами, принимающими решения (ЛПР), исходных данных;
- поиск и (или) обобщение требуемых данных, хранимых в базе данных (БД);
- расчёт промежуточных данных;
- логическое обобщение решения расчётных задач.

При этом, реализация поиска и обобщения информации в БД трудностей не представляет. Действительно, большая часть информации в мире уже храниться в БД, а ведущие фирмы – разработчики программ обеспечили мощные инструментальные средства для создания программ ввода данных и их использования. Не возникает трудностей и написание программ для проведения расчётов по известным алгоритмам, а также программ для вычисления значений логических функций. Однако, в настоящее время отсутствуют средства, позволяющие представить процесс решения рассматриваемых расчетно-логических задач как единую последовательность операций, позволяющих ЛПР ввести исходные данные и получить интересующий результат путём реализации на ЭВМ

процедур различного характера: процедур поиска, алгоритмических процедур, логических процедур.

Рассмотрим метод, позволяющий представить процесс решения задач рассматриваемого класса с использованием сетевой модели. Модель, которая будет предложена, относится к классу неоднородных функциональных сетевых моделей. Модели данного класса широко используются для представления знаний [1, 2].

Под **сетевой моделью процесса решения задачи** (СМ ПРЗ) будем понимать совокупность вершин и связей между ними, представляющих этапы и последовательность их реализации для решения задач расчетно-логического характера.

На СМ ПРЗ каждому из предложенных типовых этапов поставлены в соответствие вершины различных типов:

- начальные условия (рис. 1, а);
- поисковые вершины (рис. 1, б);
- алгоритмические вершины (рис. 1, в);
- вершины сравнения (рис. 1, г);
- вершины типа И (рис. 1, д);
- вершины типа ИЛИ (рис. 1, е).

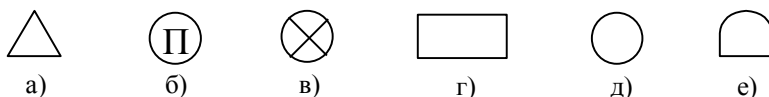


Рис. 1. Типы вершин сетевой модели процесса решения задачи

Вершины первых трёх типов отвечают операциям подготовки логико-аналитической деятельности ЛПР, остальные необходимы для представления процесса логического обобщения числовых данных ЛПР.

Появление на сетевой модели начальных условий определяет необходимость разработки средств для ввода некоторых исходных данных. Поиск и обобщение данных реализуется путём запуска процедур, реализующих запросы к базам данных.

Ввод в структуру сети алгоритмических процедур предполагает необходимость реализации в ходе поиска решения некоторого алгоритма, позволяющего получить некоторые промежуточные результаты, например, подлетное время цели до зоны ответственности соединения ПВО. Алгоритм решения данной задачи предполагает: поиск координат точек многогранника, посредством которого аппроксимируется зона ответственности, в БД; решение геометрической задачи определения расстояния от точки, соответствующей текущему положению цели, до одной из граней многогранника; определение времени по известным расстоянию и скорости движения.

Результатом реализации этапов, которым соответствуют поисковые

и алгоритмические вершины, как правило, являются числовые значения. В тоже время для реализации логико-аналитической деятельности, реализуемой, как правило, с использованием классической логики, используются два логических значения: «истина» и «ложь». Для перехода от числовых к булевым значениям используются вершины сравнения, появление которых также предполагает реализацию некоторых процедур. Отличительной особенностью этих процедур является наличие различных параметров, которые могут отсутствовать в процедурах, реализация которых обуславливается поисковыми и алгоритмическими вершинами, а также булевский тип возвращаемых значений.

Вершины И и ИЛИ появляются на сетевой модели после вершин сравнения и служат для реализации логических рассуждений, представимых с использованием логических формул произвольного вида. Отличительной особенностью является необходимость реализации типовых процедур с переменным числом параметров, число которых определяет количество смежных вершин.

**Отношения**, которым соответствуют связи между вершинами сетевой модели, обозначают тот факт, что реализация этапа решения задачи, которому соответствует вершина, из которой данная связь исходит, является необходимым условием для реализации этапа решения задачи, которой соответствует вершина, в которую входит данная связь.

Такие отношения обладают следующими свойствами:

1. Антирефлексивность – этап решения  $x$  не является необходимым условием реализации самого себя:

$$\neg xRx.$$

2. Антисимметричность – если этап  $x$  является необходимым условием для реализации этапа  $y$ , то  $y$  не может являться необходимым условием для реализации этапа  $x$ :

$$xRy \rightarrow \neg yRx.$$

3. Транзитивность – если этап  $x$  является необходимым условием для реализации этапа  $y$ , а этап  $y$  является необходимым условием для реализации этапа  $z$ , то  $x$  является необходимым условием и для реализации этапа  $z$ :

$$(xRy \wedge yRz) \rightarrow xRz.$$

Приведённые свойства отношений определяют, что связи между вершинами рассматриваемой сетевой модели являются однотипными и направлены от вершин нижних уровней иерархии к вершинам верхних уровней иерархии. При этом под **уровнем иерархии вершины** будем понимать максимальное количество последовательных связей от начальных условий до текущей вершины. Таким образом, на самом нижнем уровне иерархии (нулевом) находятся начальные условия, на первом – вершины, для которых начальные условия являются достаточной информацией для запуска соответствующих процедур, на втором – вершины, которым со-

ответствуют процедуры, запуск которых возможен при наличии значе- ний начальных условий и результатов реализации процедур, соответ- ствующих вершинам первого уровня иерархии и т.д. На самом верхнем уровне иерархии находится единственная вершина, соответствующая главной цели решения задачи. Результат ее означивания соответствует решению логико-расчётной задачи.

Таким образом, формально сетевая модель процесса решения задачи может быть представлена следующим набором элементов:

$$SM = \{V, R\},$$

где  $V$  – множество вершин сетевой модели;  $R$  – множество связей.

Каждая вершина сети может быть описана кортежем вида:

$$V = \{V_i\} = \{ \langle Sem_i, UI_i, NV_i, T_i, Rez_i, Pr Oz_i \rangle \},$$

где  $Sem_i$  – наименование (цель) этапа принятия (поиска) решения;  $UI_i$  – номер уровня иерархии вершины и номер вершины на уровне, необходимые для определения местоположения вершины на сетевой модели;  $NV_i$  – тип вершины, необходимый для реализации графического отображения сетевой модели и для реализации контроля её корректности;  $T_i$  – процедура для реализации этапа принятия решения на ЭВМ;  $Rez_i$  – результат реализации этапа решения задачи, представляющий собой числовое или символьное значение;  $Pr Oz_i$  – процедура, реализуемая для получения результата этапа решения задачи.

Описание процедур реализации этапов решения задач может быть представлено следующим кортежем:

$$Pr Oz_j = \langle K_j, B_j, S_j, TP_j, TR_j, P_j \rangle,$$

где  $K_j$  – команда вызова процедуры, представляющая собой оператор языка программирования, которая позволяет запустить программу из определённого директория на диске или процедуру из библиотеки;  $B_j$  – имя библиотеки процедур, которое заполняется, если значение поля  $K_j$  является именем процедуры;  $S_j$  – семантическое описание (назначение) процедуры;  $TP_j$  – тип процедуры, представляющий собой двоичный признак, определяющий, что список параметров процедуры будет фиксированным (например, для процедуры реализации сравнения двух чисел) либо произвольным (например, для процедур получения результата в вершинах И и ИЛИ);  $TR_j$  – тип возвращаемого процедурой результата;

$P^j$  – набор параметров процедуры (может отсутствовать).

Каждый из параметров описывается следующим кортежем:

$$P^j = \{P_{ij}\} = \{\text{Name}_{ij}, \text{Sem}_{ij}, \text{TPar}_{ij}\},$$

где  $\text{Name}_{ij}$  – имя параметра;  $\text{Sem}_{ij}$  – семантическое описание параметра;  $\text{TPar}_{ij}$  – тип значения параметра.

Связи между вершинами устанавливаются для определения того факта, что вершина нижнего уровня является параметром для реализации процедуры верхнего уровня иерархии. При таком рассмотрении связей сетевой модели они могут быть представлены кортежем следующего вида:

$$R_k = \langle V_k^{is}, V_k^{vh}, P(\text{Pr Oz}(V_k^{vh})) \rangle,$$

где  $V_k^{is}$  – вершина, из которой исходит отношение;  $V_k^{vh}$  – вершина, в которую входит отношение;  $P(\text{Pr Oz}(V_k^{vh}))$  – параметр процедуры, соответствующей вершине  $V_k^{vh}$ , в качестве которого выступает результат реализации процедуры, соответствующей вершине  $V_k^{is}$ .

Использование предложенной сетевой модели позволяет представить процесс решения задач управления, имеющих логико-расчетный характер. К достоинствам такого представления относятся наглядность представления процесса решения задачи, а также уменьшение времени модификации знаний о порядке решения задачи и внесения корректив в процедуры реализации отдельных этапов решения.

## ЛИТЕРАТУРА

1. Ярушек В.Е., Козлов С.А. Метод построения сетевой модели задачи управления организационными системами // Автоматизированные системы управления и приборы автоматики. – Х.: Вища школа. – 1988. – Вып. 86. – С. 18 - 22.
2. Низиенко Б.И., Затхей В.А., Селезнев С.Е. Использование функциональных сетей для означивания начальных условий в открытых экспертных системах // Збірник наукових праць ХВУ. – Х.: ХВУ. – 2000. – Вип. 4(30). – С. 126 - 131.

Поступила 24.07.2002

*ДВУХГЛАВОВ Дмитрий Эдуардович, канд. техн. наук, старший преподаватель кафедры ХВУ. В 1994 году окончил ХВУ. Область научных интересов – использование методов искусственного интеллекта для решения задач управления в военной сфере.*

*ЗАТХЕЙ Владимир Анатольевич, преподаватель кафедры ХВУ. В 1992 году окончил Харьковское ВВАУРЭ, в 1995 году – ХВУ. Область научных интересов – открытые экспертные системы поддержки принятия решений в реальном масштабе времени.*