УДК 658.012.011.56

# ЯЗЫК ОПИСАНИЯ СХЕМ РАСПРЕДЕЛЕННЫХ ПРОЦЕССОВ В АВТОМАТИЗИРОВАННЫХ СИСТЕМАХ С ПРОГРАММНЫМ УПРАВЛЕНИЕМ

**Скатков А.В., Сергеев Г.Г., Юрина М.Н.**

Рассматривается язык спецификаций для описания параллельных процессов в АСУ с программным управлением. Предлагается методика использования языка.

В настоящее время программное обеспечение автоматизированных систем управления определяется как некоторая совокупность программных слоев между техническими средствами и пользователями. Связь между пользователями и программными компонентами является важной задачей, так как описание операций в процессах управления даже при минимальном наборе характеризующих их параметров, является достаточно трудоемким. Эта задача может быть решена с помощью специальных языковых и транслирующих средств.

В настоящей работе предлагается один из вариантов языка спецификаций для автоматизированных производственных систем с программным управлением. При его создании авторы руководствовались следующими базовыми принципами: графическая часть должна отвечать основным требованиям Р-технологии [1]; язык спецификаций строится на базе объектно-ориентированного подхода, обеспечивая возможность иерархического описания объектов управления.

При описании языка будем использовать термины объектно-ориентированного программирования, как это принято в [2].

**Компонент** - это объект, имеющий визуальное представление. **Схема управления** - это компонент, состоящий из совокупности взаимосвязанных компонентов. Таким образом, каждый элемент схемы – объект. Объект может включать в свой состав другие объекты. Для описания структуры объектов используется тип данных, который будем называть **класс**. Объекты, принадлежащие одному и тому же классу, имеют одинаковую структуру. Обратное утверждение неверно.

**Структура программы.** Программа состоит из следующих разделов: описания библиотек, описания типов, раздел описания структуры и раздел операторов.

Программа: <Имя программы>;
Библиотеки: <список библиотек>;
Типы: <список определений типов>;
Структура: <определение схем>;
Оптимизация: <список объектов>;

**Простые типы данных**: строка, число, рисунок (пиктограмма). Описание данных простого типа: Наименование: Тип. Например: Имя: Строка; Длительность: число; Количество: число; Изображение: рисунок.

**Составные типы данных**: фрейм, класс, список, перечислимый тип. Список - это последовательность данных одного типа, разделенных запятой. Под фреймом в языке описания процессов управления будем понимать последовательность данных различных типов. Структура фрейма определяется на уровне описания типов как последовательность пар Имя: Тип, разделенных запятыми. В целом, фрейм заключается в квадратные скобки. Фрейм – это исключительно тип данных. Класс определяется как последовательность Имя_класса (Базовый класс: класс) [фрейм свойств]. При определении значений элемента фрейма свойств в тексте программы вместо типа подставляется значение указанного типа. Приведем примеры объявления и определения списков, фреймов и классов.

Объявление фрейма:

Фрейм ПримерФрейма[Строка Имя; Рисунок Изображение; Число Длительность];

Объявление класса:

ТОбъект()[Строка Имя, Рисунок Пиктограмма].

Отметим, что ТОбъект является базовым классом языка. Все остальные классы порождаются от базового.

**Свойства объектов**. Согласно базовым принципам объектно-ориентированного программирования основными свойствами объектов являются инкапсуляция, наследование и полиморфизм. Инкапсуляция в настоящем языке описания процессов управления проявляется в том, что объект всегда рассматривается как единая структура. Механизм наследования предполагает, что если класс порождается от базового, то в состав фрейма класса-потомка автоматически включаются элементы фрейма класса-предка. Полиморфность объектов, в отличие от таких объектно-ориентированных языков как C++, Object Pascal обеспечивается не перегрузкой методов, а возможностью подстановки вместо объекта-предка объекта-потомка с новыми свойствами. Например, пусть класс ТБлок порожден от класса ТДействие. Если где-либо в объявлении типов в качестве элемента фрейма описан объект типа ТБлок, то при определении ему может быть присвоен фрейм объекта типа ТДействие. Это важное свойство объекта позволяет ввести операцию сопоставления и проекции

объектов, которые будут рассмотрены позже. Кроме того, один и тот же объект может иметь не одно, а множество различных значений.

**Объявление и определение переменных, области видимости переменных**. В настоящем варианте языка описания процессов управления переменными считаются объекты, стоящие после идентификатора типа класса. Если после имени переменной следует фрейм, переменной присваивается значение этого фрейма.

Это значение действительно от места его определения до следующего определения или до конца программы. Значение элементов фрейма указываются в виде последовательности: Имя элемента: Значение элемента.

**Основные классы объектов**. Введем в рассмотрение следующие классы, порождаемые от базового класса ТОбъект:

связь ТСвязь(ТОбъект);

объект управления ТУпрОбъект(ТОбъект);

управляющие воздействия ТДействие(ТОбъект);

ТВход(ТОбъект);

ТВыход(ТОбъект);

ТСхема(ТОбъект)[Список типа ТВход Входы; Список типа ТДействия Действия; Список типа ТУпрОбъект УпрОбъекты; Список типа ТВыход Выходы; Список типа ТСвязь Связи].

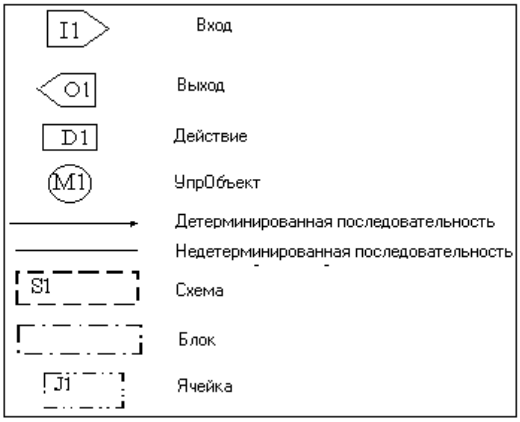
Для уточнения характера связи введем еще два класса:

ТРебро(ТСвязь) [Список типа ТОбъект Вершины];

ТДуга (ТРебро) [Список типа ТОбъект Порядок], где Порядок определен как Фрейм Порядок [ТОбъект Исход; ТОбъект Вход].

Для объединения нескольких действий в блок действий введем класс ТБлок (ТДействие) [Список типа ТДействие Набор].

Определим значение поля «пиктограмма» для экземпляров описанных выше объектов, так как это показано на рисунке 1. С помощью пиктограмм пользователь осуществляет описание схемы управления на графическом уровне. Каждой пиктограмме однозначно соответствует объект программы на языке спецификаций.




**Пример 1.** Рассмотрим описание процесса управления технологическим оборудованием для параллельного изготовления деталей

двух видов. Пусть над деталями вида I1 выполняется фрезеровка D1, после чего они образуют заготовку вида M1. Заготовка M1 является основой для изготовления заготовки вида M2 после выполнения операций фрезеровка D2, шлифовка D3, фрезеровка D4 в произвольной последовательности. Заготовка M2 после шлифовки D7 образует изделие типа O1. Из заготовки M1, над которой дополнительно выполняется действие шлифовка D3, и заготовки M2 после выполнения соединения и сверления D5 производится деталь O2.

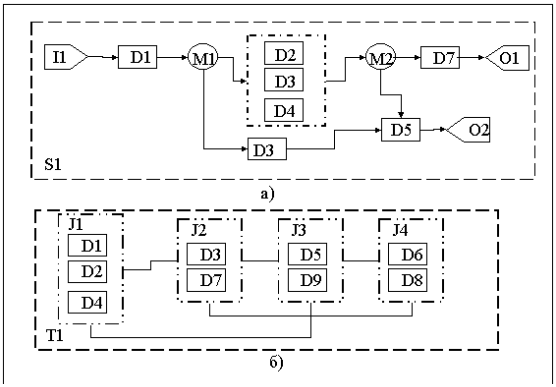



Рисунок 2 - Схема процесса управления и структура АСУ в терминах языка спецификаций

Графически описанная выше схема управления может быть представлена так, как это показано на рисунке 2,а. В терминах языка этот процесс описывается объектом типа TСхема:

```
TСхема S1
[Входы: TВход I1[Имя: 'Дет14'];
Действия: TДействие D1[Имя: 'ФрезК1'], D2[Имя: 'ФрезК1'],
      D3[Имя: 'ШлифК1'], D4[Имя: 'ФрезК2'],
      D5[Имя: 'СверС1'], D3[Имя: 'ШлифК2'];
         TБлок V1[Набор: D2,D3,D4];
Заготовки:
TУпрОбъект M1[Имя: 'Дет14а'], M2[Имя: 'Дет14б'];
Выходы: TВыход O1[Имя:'Дет15'],O2[Имя:'Дет16'];
Связи:          TДуга L1[Вершины:[ Исход:I1, Вход:D1],
   [Исход:D1, Вход:M1],[Исход: M1, Вход:V1],
   [Исход: M1, Вход:D3],
   [Исход:V1, Вход:M2],[ Исход:D3, Вход:D5],
   [Исход:M2, Вход:D5],[ Исход:M2, Вход:D7],
   [Исход:D5, Вход:02]; [Исход:D7, Вход:O1]];
    TРебро R1[Вершины: D2,D3,D4];
];
```

Для описания структуры АСУ введем в рассмотрение следующие типы данных:

Фрейм ФОперация [ТДействие Действие; ТДлительность Длительность].

ТЯчейка (ТБлок) [Список типа ФОперация Операции];

Пусть для реализации схемы управления примера 1 АСУ включает в себя следующие технологические ячейки: фрезерная J1, шлифовальная J2, сверлильная J3 и токарная J4. Попарно между ячейками J1,J2; J2,J3; J3,J4;J1,J3;J2,J4 предусмотрена возможность транспортировки заготовок. На рисунке 2б представлено графическое описание структуры АСУ с точки зрения технолога по управляющей системе. В программе эта структура может быть описана объектом типа ТСхема:

ТСхема Т1

[

Действия:

ТЯчейка J1[Имя: 'Фрезер', Операции: [Действие:D1, Длительность:14],[D2, Длительность:16],[D4, Длительность:70]];

J2[Имя:'Шлиф.', Операции: [Действие:D3, Длительность:65],[ Действие:D7, Длительность:84]];

J3[Имя: 'Сверл.', Операции: [Действие:D5, Длительность:46],[ Действие:D9, Длительность:24]];

J4[Имя: 'Токар.', Операции: [Действие:D6, Длительность:64],[ Действие:D8, Длительность:82]];

Связи:

ТРебро R1[Вершины: D1,D2,D4], R2[Вершины: D3,D7];

ТРебро R3[Вершины: D5,D9], R4[Вершины: D6,D8];

ТРебро R5[Вершины: J1,J2], R6[Вершины: J2,J3];

ТРебро R7[Вершины: J3,J4], R7[Вершины: J1,J3];

ТРебро R5[Вершины: J2,J3];

];

**Операции над объектами.** Операция присваивания. Эта операция записывается так, как это принято в языке Паскаль: ':=', а оператор присваивания соответственно имеет вид Объект:=Выражение. Кроме операции присваивания над объектами могут выполняться операции проекции, сравнения, дизъюнкции с условием и конъюнкции.

Операция конъюнкции обозначается символом '|' и используется для описания альтернативных значений одного и того же объекта. Например: ТСхема С:=С1|С2|С3. Это означает, что схема управления производством изделий С имеет три различных варианта С1, С2 или С3. С помощью операции дизъюнкции & определяются условия выбора альтернативы. При записи условий используются операции '>', '<', '=', '≤', '≥', '≠'. Например,

C:=C1&(C2=C3)|C4&(C2=C5). Для простых типов данных строка и число эти операции имеют такой же смысл, как и в языке Паскаль [2]. Для доступа к элементам фрейма используется имя элемента, заключенное в «обратные» квадратные скобки: C1]Операции]Длительность[[=27. Операции сравнения '>', '<' для объектов не имеют смысла.

Операция проекции '#' над объектами A1 и A2 выполняется согласно следующим правилам. Пусть F1 – множество элементов фрейма свойств объекта A1, F2 – множество элементов фрейма свойств объекта A2. Проекция Z=A1^A2 будет иметь фрейм свойств F1. Если элемент O1∈F1 – объект класса K1, а соответствующий ему элемент O2∈F2 объект класса K2, то результатом проекции будет элемент фрейма, который имеет тип нижнего в дереве иерархии объекта. Если класс K1 – предок класса K2, то результат имеет тип K2 или K1 в противном случае.

Операцию проекции целесообразно использовать для наложения схем управления на структуру АСУ. Для примера 1 результат операции проекции TСхема Z=S1#T1 имеет вид:

TСхема Z [
Входы: TВход I1[Имя: 'Дет14'];
Действия:
TЯчейка J1[Имя: 'Фрезер'; Опреации:[Дейсвтие:D1[Имя 'ФрезК1']; Длительность:14], [Дейсвтие:D2[Имя: 'ФрезК1']; Длительность:16], [D4[Имя: 'ФрезК2'], Длительность:70]]],
J2[Имя: 'Шлиф'; Операции:[Действие: D3[Имя: 'ШлифК1']; Длительность:16], D7[Имя: 'ШлифК2']; Длительность:16]]],
J3[Имя: 'Сверл.'; Операции:[Действие: D5[Имя: 'СверС1']; Длительность:46]]];
TБлок V1[Набор: D2,D3,D4];
Заготовки:
TУпрОбъект M1[Имя: 'Дет14а'], M2[Имя: 'Дет14б'];
Выходы: TВыход O1[Имя:'Дет15'],O2[Имя:'Дет16'];
Связи: TДуга L1[Вершины:[ Исход:I1, Вход:D1],
[Исход:D1, Вход:M1],[Исход: M1, Вход:V1],
[Исход: M1, Вход:D3],
[Исход:V1, Вход:M2],[ Исход:D3, Вход:D5],
[Исход:M2, Вход:D5],[ Исход:M2, Вход:D7],
[Исход:D5, Вход:02]; [Исход:D7, Вход:O1]];
TРебро R1[Вершины: D2,D3,D4]];

Операция объединения объектов обозначается символом '^'. Как правило, она применяется для объединения схем управления производства различных изделий. Например: S:=S1^S2. При выполнении операции

объединения фрейм свойств объекта результата представляет собой объединение множеств элементов фрейма свойств аргументов операции.

При вычислении выражений операции выполняются в порядке их записи слева направо с учетом приоритета. Определим бинарное отношение $\pi(v_1,v_2)$ над операциями языка

$\pi =$

| $v_1$ / $v_2$ | # | ^ | & | \| | < | > | = | ≤ | ≥ | ≠ | := |
|---|---|---|---|---|---|---|---|---|---|---|---|
| # | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| ^ | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| & | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| \| | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| < | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| > | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| = | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| ≤ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| ≥ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| ≠ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| := | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

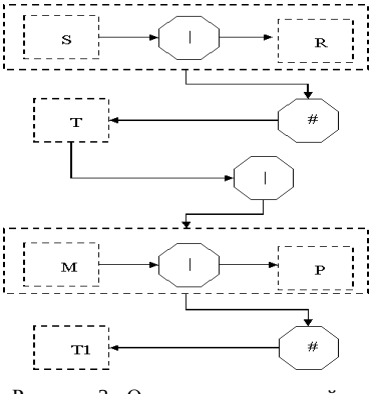
Для изменения порядка операций в выражении могут использоваться круглые скобки, как это принято при записи арифметических операций.

Например: ТСхема W:=(S|R)#T | (M|P)#T1

Операции над схемами могут задаваться в графическом виде так, как это показано на рисунке 3.

После определения схемы управления в проекции на структуру АСУ перечень объектов, подлежащих оптимизации помещается в виде списка в раздел «оптимизация». Например: Оптимизация: Z.

Следует отметить, что типовые схемы управления производством номенклатурных изделий, а также возможные варианты организационной структуры АСУ целесообразно хранить в библиотеках схем. Описание схемы процессов управления выполняется с помощью визуальной среды.




Эффективность работы технолога по системам управления во многом определяется эргономическими характеристиками интерфейсной части программной среды [1, 3]. Следуя рекомендациям стандарта GUI[2], среда

разработчика состоит из палитры визуальных компонентов, формы определения схемы управления и диспетчера объектов.

Создание описания схемы, таким образом, сводится к переносу компонентов из палитры в форму и определения их свойств в диспетчере объектов.

С помощью меню визуальной среды инициируются процессы трансляции программы в формальное представление на языке спецификаций, оптимизации процесса управления, формирования оптимального решения задачи ОКП в частности.

Разработанное авторами программное обеспечение позволяет в значительной степени сократить время описания процессов управления и повысить их наглядность.

## *Библиография*

1 Вельбицкий И.В. Технология программирования / И.В. Вельбицкий. — К.: Техника, 1984. — 279 с.

2 Конопка Р. Создание оригинальных компонент в среде Delphi/ Р. Конопка. — К.: Диалектика, 1996. — 376 с.

3 Слепцов А.И. Автоматизация проектирования управляющих систем гибких автоматизированных производств / А.И. Слепцов, А.А. Юрасов. — К.: Техника, 1986. — 110 с.

***Поступила в редакцию 03.19.2001 г.***