

Система МИМ позволяет решить проблему автоматического получения номенклатурных показателей качества для отдельного класса ПС - программ имитационных моделей.

Библиографический список

1. Карповский Е.Я., Бекетова Е.А. Метод определения дифференцированной цикломатической сложности программ имитационных моделей // Программные и аппаратные средства многопроцессорных вычислительных комплексов: Сб. науч. тр. / ИК АН УССР. Киев, 1989. С.112-117.

2. Холостед М.Х. Начало науки о программах. М.: Финансы и статистика, 1981. 128 с.

3. Шрайбер Т. Дж. Моделирование на *GPSS* : Пер. с англ. М.: Машиностроение, 1980. 592 с.

УДК 681.3

А.Г.Ахлямов, В.Г.Пенко

Одесский институт народного хозяйства

МЕТОД АВТОМАТИЗАЦИИ ПОСТРОЕНИЯ ИМИТАЦИОННЫХ МОДЕЛЕЙ

Рассмотрены вопросы автоматизации описания на первых этапах моделей сложных систем. Подход ориентирован на аппарат модифицированных сетей Петри, который служит адекватным средством реализации программ имитационных моделей.

Имитационное моделирование (ИМ) является своеобразным направлением развития информатики. С одной стороны, средства ИМ - специальные программные, т.е., в отличие от универсальных средств, имеют дело с ограниченным кругом проблем и со специфическим подходом к их решению. С другой стороны, круг таких проблем столь велик

Автоматизация научных исследований. Куйбышев, 1990.

и разнообразен, что не дает возможность продвинуть область ИМ до такой степени проработанности, которая позволила бы существенно автоматизировать решение проблем ИМ. В связи с этим для ИМ сейчас наиболее актуальным является разработка технологий, повышающих эффективность программного обеспечения ИМ.

Введение подпрограмм в программировании оказало значительное влияние на уровень его технологичности, позволило не только повысить эффективность, но и существенно расширить содержательный базис программирования. Желая получить подобный эффект в ИМ, попытаемся ввести понятие "подмодель". Очевидно, что основным свойством подмоделей, как и подпрограмм, должна быть способность их к комплексации друг с другом с помощью операций комплексации. С другой стороны, понятие "подмодель" принципиально отличается от понятия "подпрограмма", что, очевидно, связано с непоследовательным характером передачи управления между элементами программы имитационной модели. В определении подмодели следует подчеркнуть, что подмодель соответствует некоторой абстрактно выделяемой подсистеме имитируемой системы. Следовательно, подмодель можно рассматривать как "относительно черный ящик", в том смысле, что о его внутреннем строении у нас информации нет, однако есть некоторая информация о связи подмодели с внешним миром. Связь с внешним миром осуществляется двумя путями: получение управляющей информации (вход подмодели); выдача реакции на управляющую информацию (выход подмодели). Причем часть выхода может использоваться активно, например как вход для другой подмодели, а часть — пассивно, для наблюдения, сбора статистической информации и т.д.

Отличие подмодели от подпрограммы во многом состоит в синхронизации входа и выхода. Вся входная информация поступает на вход подпрограммы одновременно, а вся выходная информация вылается подпрограммой также одновременно и после получения входной информации. Таким образом, формально подпрограмму можно представить как тройку:

$$\langle I, O, F \rangle ,$$

где $I = \langle \{d_{I1}, \dots, d_{In}\}, t_I \rangle$ — вход подпрограммы;

$O = \langle \{d_{O1}, \dots, d_{Om}\}, t_O \rangle$ — выход подпрограммы;

F — правило функционирования подпрограммы:

$$O = F(I) ;$$

d_{Ii} - входные значения; d_{oi} - выходные значения; t_{Ii} - момент входа в подпрограмму; t_{oi} - момент выхода из подпрограммы.

Следует иметь в виду, что моменты входа и выхода относятся к немасштабированной временной шкале, на которой относительно события можно говорить только, произошло оно до или после другого события.

Аналогично подмодель можно рассматривать как тройку:

$\langle I, O, F \rangle$

где $I = \{ \langle d_{I1}, t_{I1} \rangle, \dots, \langle d_{In}, t_{In} \rangle \}$ - вход подмодели;

$O = \{ \langle d_{o1}, t_{o1} \rangle, \dots, \langle d_{om}, t_{om} \rangle \}$ - выход подмодели;

F - правило функционирования подпрограммы; d_{Ii} - входные значения; d_{oi} - выходные значения; t_{Ii} - момент поступления входной информации; t_{oi} - моменты выдачи выходной информации.

Как видим, основное отличие от определения подпрограммы заключается в отсутствии синхронизации между моментами поступления и выдачи входной и выходной информации соответственно.

Отсутствие такой синхронизации приводит к ряду последствий, некоторые из которых следующие:

условность выделения подмодели из модели - многое зависит от точки зрения пользователя на систему, целей его исследований;

невозможность исчерпывающего структурного (графического) представления системы средствами блок-схем программирования;

разнообразие типов взаимосвязей в системах, приводящие к проблемам в полезной классификации подмоделей и определении операций комплексации подмоделей.

В ходе развития ИМ в результате выделения в реальных системах абстрактных и общезначимых концепций и функций появились специальные языковые средства ИМ. Такие средства можно назвать элементарными абстрактными подмоделями (например, блоки в *GPSS*, *SLAM II*, операторы в *SIMULA-2* и т.д.). Модели получались путем соединения таких подмоделей "встык" в тексте программы. Даже в языках ИМ, где одной из форм представления модели служила блок-схема,

не было эффективных средств комплексации элементарных подмоделей.

Авторами был разработан аппарат модифицированных сетей Петри (МСП), который служит адекватным средством реализации программ ИМ, так и технологичным формализмом, позволяющим эффективно пользоваться понятием "подмодель". Подробное описание МСП можно найти в работе [1].

МСП представляют собой традиционные сети Петри, расширенные следующими возможностями: временные переходы; процедурные переходы; прерывающие и освобождающие переходы; предикатные входные и выходные дуги; индивидуализация фишек.

Погружение в среду упрощенной версии языка *PASCAL* делает МСП не только средством моделирования поведения систем, но и эффективным средством сбора и обработки статистической информации и построения планов эксперимента.

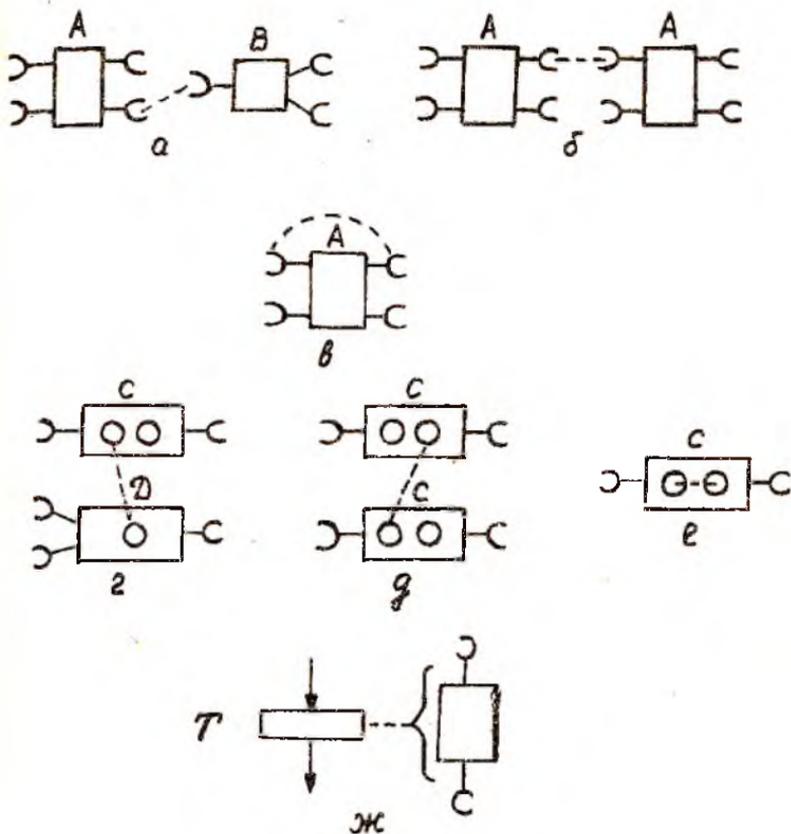
Рассмотрим теперь МСП как средство введения в систему ИМ подмоделей. Нетрудно убедиться, что МСП являются адекватным средством представления подмоделей, определенных в (I). Действительно, МСП решают проблему синхронизации между событиями, происходящими в подмодели. Достаточно в сетевом фрагменте "спрятать" от пользователя "внутренние" элементы и оставить "видимыми" места, куда поступает входная информация, и места, где вырабатывается выходная информация.

Выше мы говорили о разнообразии типов взаимосвязей в системах и, как следствие, многообразии классификаций и операций над подмоделями. При использовании сетевого представления подмодели любая операция комплексации является частным случаем операции обобщенного отождествления элементов сети. Конкретные операции получаются выбором отождествляемых элементов. Рассмотрим набор из трех операций комплексации: стыковка, наложение, иерархия.

Стыковка - отождествление выходных и входных мест подмоделей. Причем отождествляться могут места различных подмоделей, разных экземпляров одной подмодели и даже одного экземпляра одной подмодели (см. рис., поз. а, б, в).

Наложение - элементы множества \mathcal{O} из определения (I) полезно разделить на два подмножества:

$A = \{ \langle d_{a1}, t_{a1} \rangle, \dots, \langle d_{ak}, t_{ak} \rangle \}$ - активные выходные места;
 $P = \{ \langle d_{p1}, t_{p1} \rangle, \dots, \langle d_{pe}, t_{pe} \rangle \}$ - пассивные выходные места.



Р и с. Графическое представление операций комплексации:
 \hookrightarrow - входные места; c - активные входные места; \circ - пассивные входные места; - - - - линия, соединяющая отождествляемые элементы; а - стыковка различных подмоделей А и В; б - стыковка двух экземпляров подмодели А; в - стыковка одного экземпляра подмодели А; г - наложение различных подмоделей С и Д; д - наложение двух экземпляров подмодели С; е - наложение одного экземпляра подмодели С; ж - иерархия: подмодель Е отождествляется с переходом Т

Элементы множества \mathcal{P} характерны тем, что они передают, а точнее разделяют выходную информацию между подмоделями при выполнении операции наложения. Операция наложения получается при отождествлении мест подмоделей, принадлежащих множеству \mathcal{P} . При этом отождествляться могут даже места различных экземпляров одной подмодели. Особый случай – отождествление мест в пределах одного экземпляра подмодели – приводит не к комплексации, а скорее к модификации подмодели, выражающейся в наведении новых связей в подмодели (см. рис. поз. г, д, е).

Иерархия – отождествление перехода МСП с подмоделью (рис., поз. ж). При этом следует учитывать, что в фрагменте МСП, представляющем подмодель, не может быть "висячих" дуг. При выполнении отождествления во всех операциях комплексации должны выполняться ограничения на тип отождествляемых элементов, подобные контролю параметров в программировании.

Для того чтобы описанное определение подмодели реализовать в системе ИМ, необходимо обеспечить формальный способ описания подмоделей, который был бы удобен для их интерактивного использования специалистом предметной области. Для этих целей был спроектирован язык, состоящий из небольшого базиса предложений описания подмоделей. Кроме базиса, имеются средства расширения, дающие пользователю возможность строить более сложные предложения самостоятельно.

В качестве иллюстрации приведем синтаксис базисного предложения:

```

< базисное предложение > ::=
    ЕСЛИ < список предусловий >
        ТО < список результатов >
< список предусловий > ::=
    < предусловие > [ , < список предусловий > ]
< предусловие > ::=
    < входное предусловие > [ И < входной предикат > ]
< список результатов > ::=
    ПРОИСХОДИТ ПРОЦЕСС < процесс > И В РЕЗУЛЬТАТЕ
    [ ЧЕРЕЗ < длительность > ЕДИНИЦ ВРЕМЕНИ ]
    [ ОСВОБОЖДАЕТСЯ ОТ ПРЕРЫВАНИЯ ПРОЦЕСС < процесс > ]
    [ ПРЕРЫВАЕТСЯ ПРОЦЕСС < процесс > ]
    [ ВЫПОЛНЯЕТСЯ ПРОЦЕДУРА < процедура > ]

```

