

С.А.Вальдман

АВТОМАТИЗИРОВАННАЯ РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
СИСТЕМЫ КОМПЛЕКСНОГО МОДЕЛИРОВАНИЯ ДИНАМИЧЕСКИХ ОБЪЕКТОВ
(г. Ленинград)

Система комплексного моделирования предназначена для экспериментальных исследований сложных динамических объектов с использованием комплексных моделей, в которых отдельные элементы, не имеющие адекватного описания, представлены в виде физического образа. Программная часть модели исследуемой системы-объекта работает в реальном масштабе времени, т.е. в темпе развития процессов в элементах реальной аппаратуры, включенной в систему моделирования. Высокая динамичность моделируемых процессов обуславливает жесткие требования к быстродействию рабочих программ моделирования и определяет необходимость разработки программного обеспечения на машинно-ориентированных языках с использованием специальных приемов программирования, позволяющих получать высокоэффективные программы. Разработка программ моделирования в связи с этим представляется весьма сложным процессом, требующим значительных затрат времени и ресурсов ЭВМ, привлечения к программированию высококвалифицированных специалистов. Вместе с тем, особый характер задач моделирования как задач исследовательских, обуславливает необходимость реализации различных вариантов моделей, связанных не только с изменением отдельных параметров или начальных условий, но и структур.

Система программирования задач рассматриваемого класса должна удовлетворять противоречивым требованиям высокой эффективности рабочих программ, с одной стороны, а с другой - возможности программирования моделей непосредственно пользователем, не являющимся профессиональным программистом. Система программирования, удовлетворяющая этим требованиям, должна включать в себя непроцедурные средства описания задачи моделирования, позволяющие пользователю описывать точную постановку и необходимые условия решения задачи из фиксированной проблемной области, а не конкретный алгоритм ее решения. По непроцедурному описанию задачи система программирова-

ния, используя модель проблемной области и библиотеку типовых вычислительных и логических модулей, должна построить алгоритм решения и сгенерировать эффективную (в смысле быстродействия) программу. Такие системы, называемые системами автоматизации конструирования программ [I] , включают входные проблемно-ориентированные языки для описания задач, системы анализа программ, написанных на этих языках, и системы, позволяющие оптимизировать полученную по заданным критериям программу.

В основу разработки программных моделей непрерывно-дискретных систем с жесткими ограничениями на эффективность их машинной реализации положен принцип функциональной декомпозиции при составлении спецификаций элементов исследуемой системы с использованием рекуррентной процедуры последовательного уточнения этих спецификаций на модели проблемной области. В результате анализа работ, выполняемых при составлении программных моделей, выделены три этапа перехода от постановки задачи к программе:

задача \longrightarrow спецификация \longrightarrow алгоритм \longrightarrow программа.

Наиболее важной проблемой на первом этапе является создание аппарата формального описания моделируемых систем, который, с одной стороны, предоставляет пользователю удобные изобразительные средства для описания задач, а с другой стороны, делает возможным проведение формального анализа спецификаций и однозначный перевод их в форму алгоритма. На этапе описания задачи моделирования предусмотрена иерархическая процедура описания и формального анализа структуры исходной модели, заключающаяся в декомпозиции описания ее элементов (объектов) и условий их активации (предикатов активации), характеризующих взаимодействие объектов. На каждом уровне описания исходная модель исследуемой системы имеет вид

$$M = (A, R, S, \Phi),$$

где A - множество объектов исходной модели;

R - множество атрибутов объектов (информационных элементов)

S - множество n - местных предикатов, определяющих условия активации объектов;

Φ - бинарное отношение на множестве $S \times A$.

Объектом будем называть пару $A_i (f_i, R_i)$, где $f_i: R_i^+ \rightarrow R_i^-$ - функция, отображающая множество входных атрибутов в выходные, R_i^- - множество атрибутов i -го объекта, $R_i^+ \cup R_i^- = R_i$, $R_i \subseteq R$. Предикат S_j есть отображение $S_j: R^n \rightarrow B$, причем предметными переменными предиката S_j могут быть атрибуты объектов только текуще-

го уровня описания или более высокого. Таким образом, процесс описания модели идет сверху вниз. Конечные элементы последовательности описания задачи - явные определения (модули библиотеки).

Метод программирования, основанный на такой пошаговой процедуре описания задачи, был назван д е д у к т и в н ы м [4] . Иерархическая декомпозиция описания проводится до тех пор, пока все объекты определенного уровня описания не будут идентифицированы.

Идентификация функциональных блоков (объектов) заключается в определении реализуемости описанной таким образом исходной модели с помощью имеющихся функциональных модулей библиотеки. При идентификации функциональных блоков определяется реализуемость описанной таким образом исходной модели. Анализ реализуемости преобразований f_i выполняется путем формального анализа алгоритмических взаимосвязей переменных (атрибутов объектов) на модели проблемной области. При этом для каждого объекта требуется установить, возможны ли вычисления выходных атрибутов по входным для каждого из выходных атрибутов с помощью имеющихся модулей библиотеки. В этих целях необходимо использовать такое представление модели проблемной области, в котором взаимосвязи порядка K ($K \geq 2$) между переменными сокращены только до бинарных.

Модель проблемной области представляется в виде M-графа; предложенного в работе [5] . M-граф определяется как тройка $G = (Z, V, U)$, где $U = \{U_m\}$ является множеством функциональных взаимосвязей, отображающих подмножество $X = \{X_i\} \subset Z$ в подмножество $Y = \{Y_j\} \subset Z$ через $V = \{V_m\}$. $Z = X \cup Y$, V, U представлены узлами и ребрами графа G , в котором U_m устанавливается на единственную комбинацию двух подмножеств: 1) входной тип $X_m V_m$ и 2) выходной тип $V_m Y_m$, где $X_m V_m = V_S(X_S V_m)$, $V_m Y_m = V_S(V_m Y_S)$, $X_S \in X_m \subseteq X \subset Z$, $Y_S \in Y_m \subseteq Y \subset Z$, $V_m \subset V$

представляют вычислительно реализуемую взаимосвязь графа. Таким образом, $U_m \in U = X_m V_m Y_m$ в компактной форме обозначает сложное понятие ребра M-графа. Анализ реализуемости производится на каждом уровне описания исходной модели. Для каждого f_i на графе определяется обратное соответствие для каждой вершины $z_j \in R_i^- \subseteq Y \subset Z$

$$\Gamma^{-1}(z_j) = \bigcup_{k=1}^n z_k^+ \{z_k^+ \in R_i^+ \cap V_k z_j^- = U_k\}.$$

Объект A_i , описанный как $f_i : R_i^+ \rightarrow R_i^-$, будет идентифицирован на модели проблемной области, если $R_i^- \subseteq Y_S \subset Z$, $R_i^+ \subseteq X_S \subset Z$, $\Gamma^{-1}(z_j) \subset X_S \subset Z$ и $(\bigcup_{k=1}^n V_k) \subseteq U_S$.

В таком представлении $V_S \in U$ соответствует программному модулю из библиотеки функциональных модулей, X_S и Y_S - множества входных и выходных переменных модуля, между элементами которых установлены вычислительно реализуемые взаимосвязи V_S . Идентификация логических блоков (условий активизаций объектов) заключается в поиске программных модулей в библиотеке логических модулей, соответствующих элементарным предикативным функциям и подстановке имен предметных переменных. Для практических целей достаточно ограничиться элементарным набором предикативных функций, устанавливающих бинарные отношения между атрибутами объектов, поэтому задача идентификации логических блоков может быть решена тривиальным путем.

Целью второго этапа разработки программных модулей является получение алгоритма. Для этого явные определения, полученные в результате решения задачи идентификации, должны быть упорядочены. В качестве модели выполнения алгоритма предлагается использовать модель, предложенную В.М. Глушковым. В [2] было показано, что при определенных условиях алгоритм может быть выполнен в виде синхронного автомата последовательного действия или управляющего автомата. Управляющий автомат управляет другим синхронным автоматом (операционным автоматом), выполняющим обработку данных. С другой стороны, операционный автомат обеспечивает управляющий автомат соответствующей информацией о результатах промежуточных вычислений. Эта информация представляется в форме логических переменных V_i (значений предикатов, определенных на множестве переменных исходной модели). Тогда модель реализуется с минимальными затратами временных ресурсов ЭВМ.

Таким образом, на втором этапе решается задача преобразования асинхронной модели исходного представления задачи в алгоритм, построенный в соответствии с синхронной последовательной моделью Глушкова. Возможность формализации таких структурных преобразований показана в [3].

Третий этап заключается в переводе результата второго этапа на язык программирования и может быть легко реализован при помощи средств макрогенерации, имеющихся в составе современных операционных систем.

Использование предлагаемого метода автоматизированной разработки программных моделей позволит значительно упростить процесс их составления и уменьшить затраты на разработку за счет более

раннего и менее дорогостоящего обнаружения ошибок, обеспечить высокую эффективность рабочих программ.

Л и т е р а т у р а

1. Мамиконов А.Г., Цвиркун А.Д., Кульба В.В. Автоматизация проектирования АСУ.-М: Энергоиздат, 1981.
2. Глушков В.М., Цейтлин Г.Е., Юценко Е.Л. Алгебра. Языки. Программирование.- Киев: Наукова думка, 1978.
3. Davio M., Thayse A., *Implementation and transformation of algorithms based on automata.* - *Philips Journal of Research*, Vol. 35, No. 2, 1980, pp. 122-144.
4. Bellegarde F., Finance J.P., Huc B., *A type of language for the deductive programming method.* - *Applied Computer Science*, 1979, Vol. 14, pp. 63-72.
5. V. Radhakrishna Mazhy, K.N. Raju. I.A. Bashmakov, *A formal Theory for Automation of Engineering Design.* - *Proc. Int. Conference Interact. Tech. Computer Aided Design, Bologna*, 1978, pp. 473-481.

УДК 681.325

В.В.Кладухин

СОВМЕСТНОЕ ПРОЕКТИРОВАНИЕ ПРОГРАММНО-АППАРАТНОГО УПРАВЛЕНИЯ ОБМЕНОМ ИНФОРМАЦИЕЙ В СИСТЕМАХ АВТОМАТИЗАЦИИ ЭКСПЕРИМЕНТА (г.Свердловск)

Характерной чертой многих систем автоматизации научного эксперимента является распределенное управление процессом сбора информации, которое реализуется путем согласованного распределения функций по программным и аппаратным средствам управления. При этом обеспечить согласование при раздельном проектировании программной и аппаратной компонент трудно в виду многообразия их форм и условий взаимодействия.

Ниже предлагается подход к их совместному проектированию, сущность его состоит в том, что в качестве отправной точки рассматривается единое описание всего процесса управления, из которого получают уже путем его декомпозиции описание программной и аппаратной компонент.