УДК 618.3

ПОСТРОЕНИЕ МОДЕЛИ РАЦИОНАЛЬНОГО ВЫБОРА СИСТЕМ ПРИНЯТИЯ РЕШЕНИЯ

В.И. СУМИН[1]
А.В. ИЛЬНИЦКИЙ[2]

[1)]Воронежский институт Федеральной службы исполнения наказаний РФ

[2)]Воронежский государственный педагогический университет

e-mail:
viktorsumin51@yandex.ru
ilniczki1979@yandex.ru

В настоящей работе рассматривается проблема построения модели рационального выбора систем принятия решения.

Ключевые слова: информационная система, метрика сложности, показатель эффективности.

Сложные информационные процессы разрабатываются с использованием взаимосвязанных библиотек компонентов, для которых такой выбор осуществляется в основном специалистом в этой предметной области, а количественная оценка функционирования таких систем может быть получена только в результате реальной эксплуатации. В связи с этим стоит задача уменьшения влияния специалистов в предметных областях на разработку информационных процессов.

При разработке любой информационной системы (ИС) формируется техническое задание (ТЗ), в котором формируются требования к ИС. Выполнение требований ТЗ может осуществляться разными комбинациями компонентов. Требования ТЗ, компоненты библиотек и программную реализацию можно условно представить в виде четырех абстрактных уровней (A, B, C, D). Первый уровень А соответствует требованиям ТЗ. На втором уровне В находится формализованное описание компонентов (алгоритмов) решения задачи Ei={Ei1, ...,Ein}, которая порождает подмножества О и С наборов общих и специальных компонентов Ei, которые располагаются на уровне D. Каждому элементу множества Oi={Oi1, ... ,Oim} соответствует множество программ ПОi={ПОi1, ... ,ПОim} на уровне Е.

Эти уровни можно построить ориентированным ациклическим графом (A, R) (вершины А – есть функция выбора, дуги R – отношения строгого частичного порядка между функциями).

$(\forall(\alpha \in A)(<\alpha,\alpha>\in R)$ ν a є A)(<a, a> є R),

$(\forall\alpha,b,c \in A)(<a,a>\in R)(<b,c>\in R \rightarrow (<a,c>\in R),$

Компоненты программы на 4 уровне являются на графе потомками искомой вершины и любая из функций – потомков должна быть зафиксирована структурно:

$(\forall a \in r)(\forall b)(<b,a>\in R)(b \in r)$

Приближенная структура системы принятия решений (СПР) осуществляется на основе учета характеристик компонентов и ограничений на стыковку этих компонентов, которые описывают функции связи между требованиями ТЗ и этими компонентами. Индивидуальность компонентов и их программ должна быть обусловлена следующими ограничениями:

- есть определенное количество многоцелевых компонентов, которые могут удовлетворять нескольким требованиям ТЗ;
- есть определенное количество многоцелевых компонентов, которые могут удовлетворять одному требованию ТЗ;
- разработанные программы имеют определенные характеристики (время работы, объем памяти, объем программы, точность вычислений и т.д.), на основе которых вычисляется эффективность работы СПР.

Решение этой задачи потребует больших вычислительных ресурсов. Более оптимальным подходом к решению этой задачи является последовательное применение детерминированного алгоритма и алгоритмов псевдобулевой градиентной оптимизации. Вначале находится нижняя граница оценки целевой функции, а после находится локальный минимум послестохастического поиска.

Приближенный метод решения этой задачи осуществляется на основе двух частей:

- первая часть – на основе детерминированного алгоритма определяется исходное решение задачи;

- вторая часть – находится решение посредством стохастического поиска и локальной оптимизации полученных решений и на основе этого определяется лучший локальный оптимум.

Показатель эффективности определяется на основе набора моделей и метрик оценки качества программного обеспечения для для различных жизненных циклов и позволяет произвести накопление и интеграцию разнородной метрической информации для принятия своевременных производственных решений и заключительной сертификации продукции.

Кратко рассмотрим метрики сложности. Одной из основных целей научно-технической поддержки является уменьшение сложности программного обеспечения. Именно это позволяет снизить трудоемкость проектирования, разработки, испытаний и сопровождения, обеспечить простоту и надежность производимого программного обеспечения на основе теории сложности программ, которая позволяет управлять качеством программного обеспечения и осуществлять контроль ее эталонной сложности в период эксплуатации. Существует множество показателей, с помощью которых определяется сложность программ. Имеется три вида метрик сложности:

- первый вид определяется в виде словарной метрики (метрические соотношения Холстеда, цикломатические меры Мак-Кейба и измерения Тейера);

- второй вид ориентирован на метрику связей, которая позволяет определить сложность отношений между компонентами системы (метрики Уина и Винчестера);

- третий вид ориентирован на семантические метрики, связанные с архитектурным построением программ и их оформлением.

Другая классификация делится на два вида:

- первый вид определяет сложность проектирования по следующим характеристикам: объем программы, количеством переменных, трудоемкостью и длительностью разработки программы, сложность структуры программы, сложность преобразований (алгоритмов), сложность данных и т.д.;

- второй вид определяет сложность программ по следующим характеристикам: временная, программная и информационная сложности, характеризующие эксплуатационные качества программного обеспечения.

Как правило, структура программы является основой для определения сложности программного обеспечения, т.е. мерами, в основе которых лежат топологические характеристики граф-модели программы, которые удовлетворяют подавляющему большинству требований, предъявляемых к показателям: общность применимости, адекватность рассматриваемому свойству, существенность оценки, состоятельность, количественное выражение, воспроизводимость измерений, малая трудоемкость вычислений, возможность автоматизации оценивания.

В основе цикломатической меры Мак-Кейба лежит идея оценки сложности программного обеспечения по числу базисных путей в ее управляющем графе (цикломатическое число $1(G)$ орграфа G с n-вершинами, n-дугами и p-компонентами связности есть величина $1(G) = m - n + p$).

Для определения цикломатической меры Мак-Кейба используется теорема о числе базисных путей в орграфе в котором число этих путей равно его цикломатическому числу, увеличенному на единицу и цикломатической сложностью программного обеспечения P с управляющим графом G называется величина $n(G) = 1(G) + 1 = m - n + 2$).

Эта мера характеризуется:

- простотой вычисления и повторяемостью результата, а также наглядностью и содержательностью интерпретации;

- недостатками: нечувствительность к размеру программного обеспечения и к изменению структуры программного обеспечения, отсутствию корреляции со структурированностью программного обеспечения, а также различия между конструкциями, отсутствие чувствительности к вложенности циклов.

Дж. Майерс предложил интервальную меру сложности в виде интервала [n1, n2], где n1 – цикломатическая мера, а n2 – число отдельных условий плюс единица (оператор DO считается за одно условие, а CASE с n- исходами за n-1 условий).

У. Хансен предложил определить меру сложности программного обеспечения на основе двух характеристик цикломатического числа и числа операторов.

Считаем, что известна топологическая мера Z(G), чувствительная к структурированности программного обеспечения и Z(G) = V(G) (равна цикломатической сложности) для структурированных программ и Z(G) > V(G) для неструктурированных. Цикломатические меры сложности определяют также в виде M(G) = (V(G),C,Q), где C – количество условий, необходимых для покрытия управляющего графа минимальным числом маршрутов, а Q – степень связности структуры графа программы и ее протяженность.

К мере сложности, определяемой через вложенность управляющих конструкций, относят тестирующую меру М и меру Харрисона-Мейджела, которая учитывает уровень вложенности и протяженности программного обеспечения, меру Пивоварского – цикломатическую сложность и глубину вложенности, и меру Мак-Клура – сложность схемы разбиения программного обеспечения на модули с учетом вложенности модулей и их внутренней сложности.

Мера сложности Харрисона-Мейджела определяет для каждой вершины графа сложность разбиения графа на совокупность предикатных вершин. Совокупность предикатных вершин называют приведенной и определяют из первичных сложностей вершин (вычисляются всеми возможными способами), входящих в сферу ее влияния, а также первичную сложность самой предикатной вершины.

Мера Пивоварского учитывает: последовательные и вложенные управляющие конструкции, а также сложность структурированности и неструктурированности программам определяемая отношением

$$N(G) = n(G) + SP,$$

где $n(G)$ – модифицированная цикломатическая сложность;

$V(G)$ – модифицированная цикломатическая сложность, но с одним отличием: оператор

CASE выражении с минимально необходимым числом скобок, описывающим управляющий граф программы.

Меры сложности (узловая сложность) Вудворда, Хедли определяются на основе подсчета топологических характеристик потока управления (число узлов передач управления) т.е. определяется сложность линеаризации программы и чувствительность к структуризации.

Мера сложности Чена определяет число пересечений границ между областями, образуемыми блок – схемой программы (справедливо только к структурированным программам, допускающим лишь последовательное соединение управляющих конструкций). Эта мера для неструкту- рированных программ зависит от условных и безусловных переходов и поэтому можно определить только верхнюю и нижнюю границы меры.

Метрики сложности Джилба позволяют произвести оценку сложности графоориентированных модулей программ по отношению числа переходов в этом модуле к общему числу используемых операторов.

Основываясь на вышеизложенном можно осуществить оценку одной программы или альтернативных ее вариантов что позволит оптимизировать ее разработку, а также контролировать процесс разработки программы от ТЗ до режима опытной эксплуатации.

Литература

1. Чжен Г., Менинг Е., Мети Г. Диагностика неисправностей вычислительных систем. – М.: Мир, 1972. – 232 с.

THE DESIGN OF RATIONAL CHOICE MODEL OF ADOPTION DECISION SYSTEMS

V.I. SUMIN[1]
A.V. ILNITSKIY[2]

[1]Voronezh institute of the Russian Federal Penitentionary Service

[2]Voronezh State Pedagogical University

e-mail: viktorsumin51@yandex.ru ilniczki1979@yandex.ru

The given article describes the problem of the design of rational choice model of adoption decision systems.

Key words: the information system, intricacy metre, the indicator of effectiveness.