

УДК 004.942:519.876.5

И.А. Лысенко, А.А. Смирнов

Кировоградский национальный технический университет, Кировоград

ФОРМАЛИЗАЦИЯ ПРОЦЕССА ПРОЕКТИРОВАНИЯ ТЕСТОВЫХ НАБОРОВ

Рассматриваются вопросы усовершенствования методов создания тестовых наборов для проверки программного обеспечения инфокоммуникационных систем на этапе системного тестирования. Представлена процедура преобразования упорядоченной каскадной таблицы решений в поток управления тестового набора на основе обобщенного метода дерева решений.

Ключевые слова: инфокоммуникационная система, таблица решений, матрица.

Введение

Постановка проблемы в общем виде и анализ литературы. Современные ИКС характеризуются высоким уровнем сложности разрабатываемых компонент, в первую очередь программного обеспечения (ПО) информационной и телекоммуникационной подсистем инфокоммуникационной системы (ИКС), что накладывает определенные требования на реализацию фаз жизненного цикла разработки ПО ИКС, начиная от фазы анализа и завершая фазой принятия ПО.

Одной из определяющих фаз жизненного цикла ПО ИКС, на которой осуществляется контроль качества ПО, является фаза тестирования, которая содержит одну из наиболее важных и трудоемких ступеней – разработку тестовых наборов. Как показали исследования [7], наибольшее количество недостатков разрабатываемого ПО возможно выявить на этапе системного тестирования. Для его выполнения разработаны методы формализованного описания требований к соответствующему ПО. В настоящее время такие информационные технологии рассматриваются, прежде всего, как информационные технологии проектирования тестовых наборов, состоящих из одного или нескольких тестовых случаев. В общем случае их можно рассматривать как совокупность методов проектирования тестовых наборов, проверки корректности тестовых наборов и анализа потоков управления тестовых наборов. Непосредственно под методом проектирования тестовых наборов понимается метод, используемый для создания или выбора множества тестовых случаев. Одним из таких методов, используемых для проектирования тестовых наборов для проверки ПО информационных систем, характеризующихся зависимостью от принятия логических решений, являются таблицы решений.

Для дальнейшего развития аппарата таблиц решений рассмотрим метод их преобразования в поток управления тестового набора. А также формализацию процесса проектирования тестовых наборов в виде определенной последовательности действий.

Изложение основного материала

Преобразование ТР в поток управления

Способы преобразования обычных таблиц решений (ТР) в поток управления можно представить в виде двух категорий: с использованием методов маски и с использованием методов дерева решений.

Основная идея методов маски состоит в отыскании правила ТР, описывающего (распознающего) заданную входную ситуацию [2]. При этом ТР представляется в виде булевых матриц: матрицы масок M и матрицы решений D , элементы которых определяются следующим образом

$$m_{ij} = \begin{cases} 1, & \text{если } u_{ij} \neq \lambda, \\ 0, & \text{в противном случае,} \end{cases}$$
$$d_{ij} = \begin{cases} 1, & \text{если } u_{ij} \neq 1, \\ 0, & \text{в противном случае,} \end{cases}$$

где $i = 1, 2, \dots, n$; $j = 1, 2, \dots, k$, n – количество условий в ТР, k – количество правил в ТР.

Недостатком методов маски является необходимость предварительного формирования значений всех n – условий входной ситуации, что не всегда оказывается действительно необходимым. Дополнительные затраты времени на ненужные для конкретной ситуации проверки условий снижают эффективность методов маски. По этой причине эти методы не нашли широкого применения для обычных ТР, несмотря на существование модификаций – прерываемых методов маски [3] позволяющих в определенных случаях избегать проверок всех условий.

Сравнительный анализ показал, что в общем случае методы дерева решений более эффективны с точки зрения времени преобразования ТР в граф потока управления тестового набора [4].

В методах дерева решений условия проверяются по одному до тех пор, пока не определится, какое из правил ТР следует выбрать. После проверки каждого условия осуществляется переход либо к проверке следующего условия, либо к выполнению указанных в выбранном правиле действий. При этом следующая операция определяется значением ис-

тинности проверяемого условия. Этот процесс можно изобразить в виде бинарного дерева решений, представляющего граф потока управления проектируемого тестового набора.

Общий принцип построения деревьев решений предполагает такую последовательность действий:

1. Выбрать условие C_{i^*} .
2. Создать по исходной ТР две подтаблицы. В одну подтаблицу включить те правила, для которых $(u_{i^*j} = 1) \vee (u_{i^*j} = \lambda)$, в другую те, для которых $(u_{i^*j} = 0) \vee (u_{i^*j} = \lambda)$.

3. Для каждой подтаблицы повторять действия 1 и 2 до тех пор, пока не получится подтаблица состоящая из одного столбца.

Проверка условия, выбранного из ТР, изображается на дереве решений в виде вершины – распознавателя, а множество действий образуют листья этого дерева (т.е. вершины-преобразователи).

В качестве реализации графа потока управления тестового набора на основе дерева решений используем модели деятельности объектов в нотации ООТ. Деятельность – это имеющий некоторую протяженность во времени процесс выполнения действий, которые представляют собой атомарные или мгновенные вычисления (преобразования) [1].

Разработанный нами метод преобразования таблицы решений в упорядоченную каскадную таблицу решений (УКТР) с использованием матрицы следования [6], которая позволяет задать порядок на множестве случаев использования, возможно преобразовать в граф потока управления используя обобщенные методы дерева решений. При этом разработанный метод позволяет обеспечить заданное покрытие требований к информационной подсистеме ИКС в целом и к его ПО, в частности, за счет использования модифицированных УКТР, обеспечивающих формализованное описание тестовых наборов для всех типов случаев использования, а также корректность формализованных описаний тестовых наборов за счет проверки избыточности, противоречивости и полноты ТР из состава УКТР, проверки корректности матрицы следования и ее совместимости с соответствующей ТР.

Применительно к УКТР любой из алгоритмов построения дерева решений должен быть модифицирован таким образом, чтобы учитывались дополнительная информация, заданная в матрице следования. Прежде, чем дать описание такого алгоритма, введем ряд определений, а также построим и обозначим на множестве T (составленном из множества условий C и множества действий A) несколько вспомогательных подмножеств.

Безусловной ТР назовем ТР, в которой либо нет условий, либо все условия несущественны. Безусловно выполнимым действием назовем действие,

которое задано во всех правилах ТР.

Множество безусловно выполнимых действий обозначим \tilde{A} . Очевидно, что $\tilde{A} \subseteq A$, причем равенство имеет место только для безусловной ТР.

Обозначим $C^S = \{C_i | \exists(j)(t_j p C_i)\}$ подмножество связанных условий, а $\overline{C^S} = \{C_i | \neg \exists(j)(t_j p C_i)\}$ или $\overline{C^S} = C \setminus C^S$ подмножество несвязанных условий.

Аналогично обозначим $A^S = \{a_i | \exists(j)(t_j p a_i)\}$ подмножество связанных действий, а $\overline{A^S} = \{a_i | \neg \exists(j)(t_j p a_i)\}$ или $\overline{A^S} = A \setminus A^S$ подмножество несвязанных действий.

В соответствии с приведенными определениями основные этапы алгоритма получения дерева по УКТР можно представить в виде следующей последовательности действий:

1. Сформировать для ТР множества $C, C^S, \overline{C^S}, A, \tilde{A}, A^S, \overline{A^S}$.
2. Пока $\tilde{A} \cap \overline{A^S} \neq 0$ и если $a_i \in \tilde{A} \cap \overline{A^S}$ то:
 - построить вершину, соответствующую a_i ;
 - удалить из матрицы следования i -ую строку и i -й столбец;
 - преформировать $A, \tilde{A}, A^S, \overline{A^S}, C^S, \overline{C^S}$;
 - если $A \neq 0$, то завершить выполнение алгоритма.
3. Если ТР безусловна, то:
 - если $\tilde{A} \neq A$, то выдать сообщение об ошибке в исходных данных;
 - иначе завершить выполнение алгоритма.
4. Если $C^S = C$, то:
 - выдать сообщение об ошибке в исходных данных;
 - завершить выполнение алгоритма.
5. Выбрать $c_j \in \overline{C^S}$, удовлетворяющее заданному критерию оптимальности.
6. Построить вершину, соответствующую c_j .
7. Удалить из матрицы следования j – строку и j – столбец.

8. Образовать из ТР для каждого значения c_j свою подтаблицу.

9. Рекурсивно применить этот же алгоритм для каждой подтаблицы и матрицы следования.

10. Завершить выполнение алгоритма.

На рис. 2 показан пример преобразования УКТР на основе обобщенного метода дерева решений, представленной на рис. 1. Нулевые элементы в матрице следования опущены. В качестве критерия оптимальности алгоритма взят минимум количества вершин на дереве решений, а порядок выбора условий для очередной проверки соответствует алгоритму [5].

C ₁	1	1	0	0	C ₁	0	0	0	0	1	0	0	0	0	0	0
C ₂	1	0	-	-	C ₂	1	0	0	0	0	0	0	0	0	0	0
C ₃	-	1	1	0	C ₃	0	0	0	0	0	0	0	0	0	1	0
a ₁	1	0	1	1	a ₁	0	0	0	0	0	0	0	0	0	0	0
a ₂	1	0	1	1	a ₂	0	0	0	1	0	0	0	0	0	0	0
a ₃	1	1	0	0	a ₃	0	1	1	0	0	0	0	0	0	0	0
a ₄	1	0	0	0	a ₄	0	1	0	0	0	0	0	0	0	0	0
a ₅	1	0	0	0	a ₅	0	0	0	0	0	0	1	0	0	0	0
a ₆	0	1	0	0	a ₆	1	0	0	0	0	0	0	0	0	0	0
a ₇	0	1	0	0	a ₇	0	0	0	0	0	0	0	0	1	0	0
a ₈	0	1	0	0	a ₈	0	0	0	1	0	0	0	0	0	0	0

Рис. 1. Структурная схема организации радиоканала управления с динамическим режимом функционирования

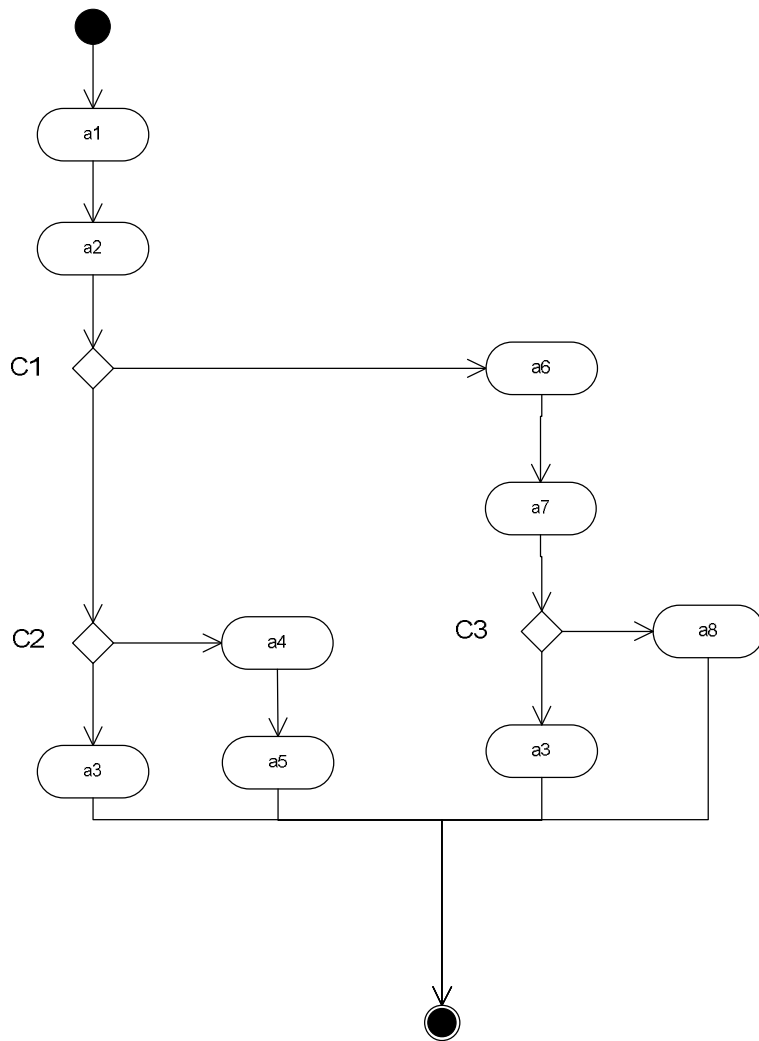


Рис. 2. Пример преобразования УКТР, представленной на рис. 1, на основе обобщенного метода дерева решений

Разработанные предложения по усовершенствованию аппарата классических ТР на основе УКТР для формального представления тестовых наборов [8], разработанные метод проверки корректности УКТР и процедуры преобразования УКТР в поток управления являются основой для проектирования тестовых наборов на основе требований к ПО информационной подсистемы ИКС на системном уровне тестирования.

Новизна метода проектирования тестовых наборов на основе требований к ПО информационной подсистемы ИКС с использованием УКТР заключается:

- в расширении описательных возможностей ТР за счет модификации аппарата ТР в виде упорядоченных каскадных таблиц решений, позволяющих задавать порядок на множестве условий и действий конкретной ТР на основе введения матрицы следования для описания случаев использования с ациклическими сценариями и особых действий для композиции и рекурсии УКТР для описания иерархических случаев использования;

- в комплексной проверке избыточности и противоречивости ТР из состава УКТР на основе использования булевых матриц масок и решений; в проверке полноты ТР на основе выявления неучтенных в ТР ситуаций, описывающих условную часть тестовых случаев, с использованием алгоритма нахождения корней логического уравнения; в проверке корректности матрицы следования и ее совместимости с ТР;

- в разработке процедуры преобразования УКТР в поток управления тестового набора на основе обобщенного метода дерева решений;

- в формализации процесса проектирования тестовых наборов в виде раз-

работки последовательности взаимосвязанных сущностей «требования к ПО на основе модели случаев использования в нотации ООТ → сценарии случаев использования на основе моделей взаимодействия в нотации ООТ → тестовые наборы и тестовые случаи в виде корректных упорядоченных каскадных таблиц решений → потоки управления на основе моделей деятельности в нотации ООТ».

Сущность разрабатываемого метода заключается в следующем (рис. 3):

1. Осуществляется анализ вербального описания требований к ПО информационной подсистемы ИКС на концептуальном уровне. Определяется на-

личие зависимости логики работы информационной подсистемы от принятия логических решений, что в свою очередь определяет необходимость в дальнейшем использования для проектирования тестовых наборов аппарата УКТР.

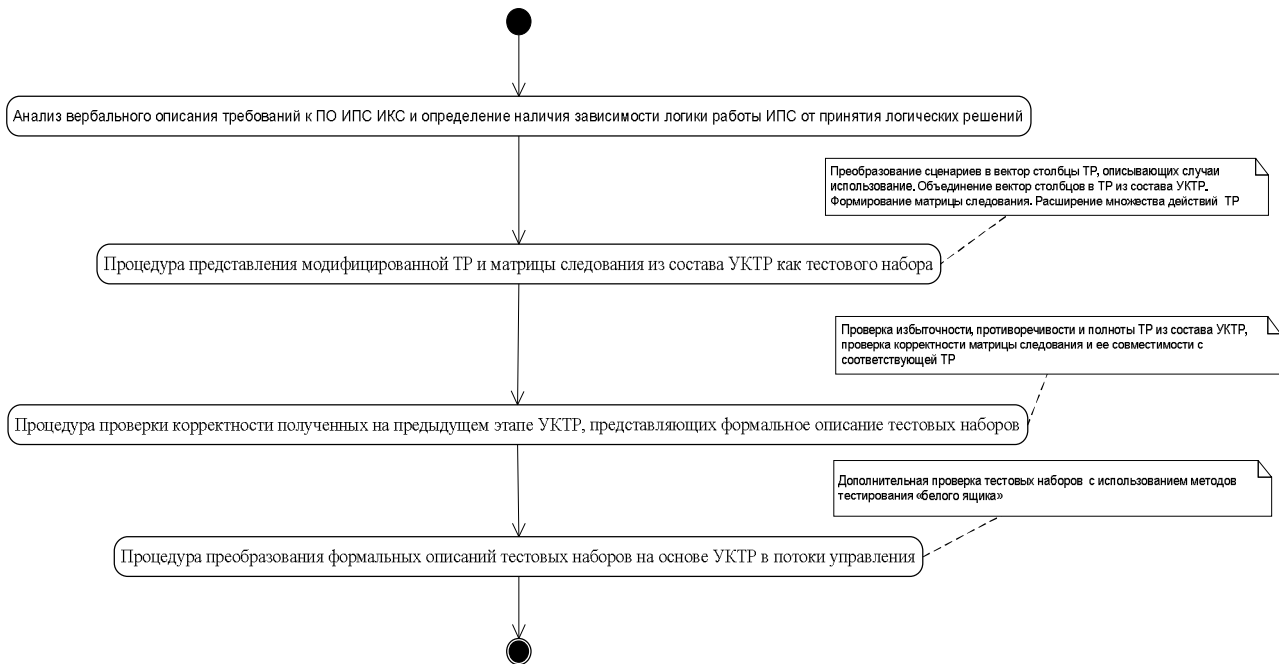


Рис. 3. Структура метода проектирования тестовых наборов на основе требований к ПО информационной подсистемы ИКС с использованием УКТР

2. Выполняется преобразование каждого сценария, представленного моделью взаимодействия $(M_{M_i}, S_{SUC_i}) \in F_{MSUC}$, в совокупность вектор столбцов ТР из состава УКТР задаваемых кортежем $R_j = \langle C_n, A_n, u_j \in U_n, w_j \in W_n \rangle$, т.е. в правила УКТР, которые в свою очередь рассматриваются как тестовый случай для проверки соответствующего сценария. В обращенной к актеру части случаев использования в виде «черного ящика» активности, порождающие сообщения, рассматриваются как действия из множества C_n . Сообщения, поступающие от актера к случаю использования, рассматриваются как условия из множества A_n . Количество вектор столбцов ТР определяется как 2^N , где N мощность множества A_n . Выполняется объединение вектор столбцов реализующих один сценарий случая использования в ТР $T_n = \langle C_n, A_n, U_n, W_n \rangle$. Выполняется формирование матрицы следования $\|l_{ij}\|$, используемой совместно с ТР T_n из состава УКТР, в соответствии с подходами, определенными в [8]. Анализируется наличие отношений расширений, обобщений или включений между элементами из множества $S_{UC}^{Not Im} \subset S_{UC}$ и элементами из множества $S_{UC}^{Im} \subset S_{UC}$. При наличии подобных отношений в зависимости от модели случаев использо-

вания и логики выполнения моделей взаимодействия сценариев случая использования из множества $S_{UC}^{Not Im} \subset S_{UC}$ в соответствующую изоморфную ТР из состава УКТР вводятся особые действия «переход на ТР» или «выполнение ТР», а в случае циклических сценариев $S_{UC}^C \subseteq S_{SUC}$ действие «рекурсивный вызов ТР». Полученная в результате УКТР, состоящая из модифицированной T_n и $\|l_{ij}\|$, рассматривается как тестовый набор для проверки требования, реализованного соответствующим случаем использования $(Reg_i, S_{UC_j}) \in F_{Reg S_{UC}}$.

3. Выполняется проверка корректности полученных на предыдущем этапе УКТР, представляющих формальное описание тестовых наборов для проверки требований к информационной подсистеме ИКС в целом и ее ПО в частности. Проверка корректности включает для каждой УКТР проверку избыточности, противоречивости и полноты ТР из состава УКТР, а также проверку корректности матрицы следования и ее совместимости с соответствующей ТР [6].

4. Выполняется преобразование разработанных и проверенных на корректность тестовых наборов, формально представленных на основе УКТР, в потоки управления на основе деревьев решений, реализованных моделями деятельности объектов в но-

таци ООТ. При необхідності проводиться додаткова перевірка коректності тестових наборів за допомогою перевірки отриманих потоків управління з використанням методів тестування «білого ящика» і з можливістю наступного аналізу додаткового (частичного) тестового покриття на базі аналізу потоку управління.

Метод проектування тестових наборів на основі вимог до ПО інформаційної підсистеми ІКС з використанням УКТР є основою для системного тестування ПО в межах удосконаленої інформаційної технології проектування тестових наборів. При цьому розроблений метод дозволяє забезпечити задане покриття вимог до інформаційної підсистеми ІКС в цілому і до її ПО, зокрема, за рахунок використання модифікованих УКТР, забезпечуючих формалізоване описання тестових наборів для всіх типів випадків використання, а також коректність формалізованих описань тестових наборів за рахунок перевірки надлишковості, суперечливості і повноти ТР з складу УКТР, перевірки коректності матриці слідування і її сумісності з відповідною ТР.

Выводы

Таким чином, розроблений нами метод проектування тестових наборів на основі вимог до ПО інформаційної підсистеми ІКС з використанням упорядкованих каскадних таблиць рішень складає в побудові процедури перетворення упорядкованої каскадної таблиці рішень в потік управління тестовим набором на основі узагальненого методу дерева рішень. А також виявлено, що з допомогою однієї УКТР можна описувати випадки використання за сценаріями, відповідними класичним алгоритмам рішень, т.е. більш широкий клас описуваних випадків використання, ніж в випадку використання звичайної ТР. При цьому забезпечується додаткова можливість аналізу тестового покриття на базі аналізу потоку управління з використанням отриманих дерев рішень без прогону коду ПО.

Також остаточно формалізовано процес проектування тестових наборів в вигляді послідовності сутностей «вимоги до ПО на основі моделі випадків → сценарії на основі моделей взаємодії → тестовий набір в вигляді упорядкованих каскадних таблиць рішень → потоки управління на основі моделей діяльності».

Список литературы

1. Леоненков А.В. Самоучитель UML. / А.В. Леоненков – СПб.: БХВ-Петербург, 2001. – 304 с.
2. Kirk H.W. Use of Decision Tables in Computer Programming. / H.W. Kirk. – Comm. of the ACM, 1965. – V. 8. – №1. – p. 41-43.
3. King P.J.H. Conversion of Decision Tables to Computer Programs by Rule Mask Techniques / P.J.H. King. – Comm. of the ACM, 1966. – V.9. – №11. – p. 796-801.
4. Еремеев А.П. О трансляции таблиц решений. / А.П. Еремеев. // Программирование, 1981. – №1. – с. 48-57.
5. Reinward L.T. Conversion of Limited Entry Decision Tables to Optimal Computer Programs 2: Minimum Average Processing Time / L.T. Reinward, R.M. Soland // Journal ACM, 1966. – V.13. – №3. – p. 339-358.
6. Лысенко И.А. Исследование алгоритма выявления вида неучтенных тестовых случаев в процессе проектирования тестовых наборов / И.А. Лысенко, А.А. Смирнов // Зв'язок. – К.: ДУТ, 2014. – № 2 (108). – С. 153-156.
7. Лысенко И.А. Исследование уровней тестирования программного обеспечения инфотелекоммуникационных систем / И.А. Лысенко, А.А. Смирнов, Е.В. Мелешко // Наука і техніка Повітряних Сил Збройних Сил України. – Вип. 4(17). – X.: ХУПС. – 2014. – С.79-81.
8. Лысенко И.А. Модификация аппарата таблиц решений для формального представления тестовых наборов / И.А. Лысенко, А.А. Смирнов // Збірник тез II МНПК «Інформаційні та телекомунікаційні технології: освіта, наука, практика». Казахстан, м. Алмати. 3-4 грудня 2015 р. – Алмати: КазНІТУ ім.К.І.Сатпаєва. – 2015. – II том. – С. 428-432
9. Лысенко И.А. Исследование процесса разработки программного обеспечения инфотелекоммуникационных систем / И.А. Лысенко, А.А. Смирнов, Л.И. Полищук // Системи озброєння і військова техніка. – X.: ХУПС. – 2014. – Вип. 4(40). – С. 103-106.

Поступила в редколлегию 11.04.2016

Рецензент: д-р техн. наук, ст. научн. сотр. С.Г. Семенов, Национальный технический университет «ХПИ», Харьков.

ФОРМАЛІЗАЦІЯ ПРОЦЕСА ПРОЕКТУВАННЯ ТЕСТОВИХ НАБОРІВ

І.А. Лисенко, О.А. Смірнов

Розглядаються питання вдосконалення методів створення тестових наборів для перевірки програмного забезпечення інфокомунікаційних систем на етапі системного тестування. Представлена процедура перетворення впорядкованої каскадної таблиці рішень у потік керування тестовим набором на основі узагальненого методу дерева рішень.

Ключові слова: інфокомунікаційні системи, таблиця рішень, матриця

FORMALIZATION DESIGN PROCESS TEST SET

I.A. Lysenko, A.A. Smirnov

Questions of improving methods of creating test kits for testing of software systems infocommunication during system testing. Presented conversion procedure ordered cascading flow control solutions in the test set based on generalized method of decision trees.

Keywords: infocommunication system, decision table, matrix