МОДЕЛИ АБДУКТИВНОЙ ЛОГИКИ ПРИ АНАЛИЗЕ НАДЕЖНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

В. Б. Вальковский, Е. Н. Мустафина

Санкт–Петербургский государственный электротехнический университет,
197376, Санкт–Петербург, ул. Профессора Попова, 5
<Vladislav.Valkovsky@dis.uu.se>

УДК 681.3
В. Б. Вальковский, Е. Н. Мустафина. **Модели абдуктивной логики при анализе надежности программного обеспечения** // Труды СПИИРАН, Вып. 2, т. 1. — СПб: СПИИРАН, 2004.
Аннотация. *Предложенный подход дает возможность оценки надежности программных систем при недостаточной информации о надежности модулей и компонент, из которых состоит система.* — Библ. 4 назв.

UDK 681.3
V. B. Valkovsky, E. N. Mustafina. **Abduction models in software reliability analysis** // SPIIRAS Proceedings. Issue 2, vol. 1. — SPb: SPIIRAS, 2004.
Abstract. *Proposed approach allows estimation of software reliability in case of absence of reliability information of its components.* — Bibl. 4 items.

1. Введение

Оценка надежности программных комплексов — сложная и ресурсоемкая задача, особенно на стадии проектирования. Возможность оценить слабые стороны архитектуры системы до ее реализации — на стадии проектирования — могла бы помочь выявить многие особенности проектируемого программного комплекса, например, дать оценку возможности использования данной системы в конкретных условиях или времени выполнения проекта более точно.

В данной статье рассматривается возможность применения математических моделей вероятностной логики для оценки такой надежности. В качестве основы взята модель оценки надежности программного обеспечения, предложенная Диком Хамлетом [1].

В настоящее время при проектировании архитектуры программных систем все более часто используются готовые (off-the-shelf) программные компоненты. Большие программные системы строятся частично или полностью из готовых программных компонентов. На текущий момент состояния отрасли, компоненты (их функциональность и входы/выходы) описываются только словесно, и нет никакой информации, позволяющей оценить надежность компонента численно и, следовательно, сделать вывод о надежности системы включающей в себя тот или иной компонент.

Хамлет [1] предлагает более формальный способ описания программных модулей (компонент), позволяющий сделать вывод о надежности компонента, а также способ вычисления надежности работы всей системы на основании описания компонент, входящих в систему, снабженного разработчиками.

В данной работе предлагается описание возможности проектирования систем и оценки их надежности в рамках неполной информации о компонентах, систем с неизвестными компонентами. Предлагается способ выявления ин-

формации о том, компоненты какой требуемой надежности необходимо включать в систему при заданной надежности всей системы.

2. Существующие модели оценки надежности програмных комплексов

Надежность компонента определим, как вероятность работы компонента без сбоев в рамках проектируемой системы. Хамлет [1–2] предлагает способ оценки надежности систем построенных из компонентов на основе информации о надежности компонентов.

Ключевым моментом модели [1] является разделение области входных данных каждого компонента на «подобласти эквивалентности», характеризующие поведение компонента. Интуитивно понятно, что подобласти должны выбираться так, что поведение компонента «непрерывно», «одинаково» внутри подобласти, таким образом, система предсказуема внутри подобласти [1–2]. Математически это выражено допущением о равномерном распределении дефектов внутри подобласти. Фактически выделение подобластей эквивалентности по входным данным равнозначно делению компонент на более мелкие «элементарные компоненты». Далее оценивается вероятность попадания входных данных компонента в ту или иную область и дается способ описания надежности компонента в каждой подобласти, а не в целом.

Сопоставим каждой i-той подобласти число h из диапазона [0,1] — вероятность попадания входных данных в эту подобласть. Совокупность подобластей входов с вероятностями попадания входных данных в каждую подобласть, называется рабочим профилем компонента (input profile) [1].

Профиль любого компонента в общем случае, возможно оценить эмпирически, однако в случае, если компонент находится внутри системы, очевидно, что профиль компонента зависит от расположения компонента в системе, т.е. от того, как отработали компоненты стоящие раньше. В [1] предлагается способ вычисления преобразования профиля происходящего при проходе данных через компонент (input propagation though the system) [1].

Используя преобразования профилей, информацию о распределении профилей и о надежности компонентов в каждом профиле выводится информация о надежности систем в целом.

Таким образом, можно рассчитать надежность каждого компонента на основании информации о тестировании данного компонента в подобластях и допущения о равномерном распределении дефектов внутри подобласти компонента. Расчет преобразования распределения данных по подобластям, используя информацию об архитектуре системы, позволяет оценить вероятность попадания входных данных компонента в подобласть при конкретном использовании компонента. Недостатком такого подхода является то, что для расчета преобразования рабочих профилей компонент необходимо запускать сам компонент много раз.

Модель построена так, что основная работа ложится на разработчиков компонентов и вычисления, использующие информацию об архитектуре системы, могут быть почти полностью автоматизированы [2].

Труды СПИИРАН. 2004. Вып. 2. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2004. Issue 2. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

3. Надежность програмных комплексов при неполной информации о компонентах — обратная задача

Часто требования к надежности системы определены заранее и требуется выяснить компоненты, какой надежности необходимо использовать, чтобы удовлетворить заданным требованиям надежности системы. Т.е. перед нами обратная задача.

В данном случае надежность одних компонентов системы может быть известна, другие же компоненты (модули) необходимо подобрать или разработать. В таком контексте полученная информация может быть использована при планировании проекта, в частности при выделении времени на тестирование системы.

Попробуем решить задачу, привлекая аппарат абдуктивного логического вывода [3].

Как было отмечено выше, надежность компонента зависит от разбиения входных данных на подобласти, вероятности попадания входных данных в каждую подобласть и от количества тестов успешно прошедших в каждой подобласти, при допущении о равномерном распределении дефектов внутри подобласти [1]. Вероятность попадания входных данных в каждую подобласть для компонентов внутри системы зависит от расположения компонента в системе и также может быть подсчитана [1].

Для простоты будем рассматривать архитектуру, где выход каждой компоненты является входом следующей, т.е. «последовательное соединение» компонент. Модель расширяема для других архитектур, включающих в себя условный переход или цикл, которые не рассматриваются в данной статье.

Надежность механической системы классически определяется как произведение надежностей элементов при условии независимости компонент. В случае с программной системой независимость компонент также можно допустить, однако следует иметь в виду, что распределение входных данных каждого компонента — свойство системы, его можно рассчитать [1].

3.1. Простые примеры

3.1.1. Один компонент

Рассмотрим для простоты систему, состоящую из одного компонента, с двумя подобластями эквивалентности входных данных.

Каждой подобласти сопоставим числа из диапазона $[0,1]$ — вероятность попадания входных данных в эту подобласть: h_1, h_2. Для каждой подобласти возможно также рассчитать другой параметр R_i из $[0,1]$ — критерий надежности (надежность), или вероятность сбоя (о способах вычисления R_i — статья [1]). Обозначим надежность подобластей нашего компонента R_1 и R_2.

Таким образом надежность компонента в целом (на всей области входных данных компонента) определяется как:

$$R = R_1 h_1 + R_2 h_2, \quad (1)$$

то есть, имеется линейная зависимость между надежностью компонента в целом и надежностью его в областях.

Труды СПИИРАН. 2004. Вып. 2. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2004. Issue 2. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

3.1.2. Два компонента

Теперь рассмотрим систему, состоящую из двух независимых программных компонентов соединенных последовательно.

1-й компонент имеет 2 подобласти h_{11}, h_{12};
2-й компонент имеет 2 подобласти h_{21}, h_{22}.

Вероятность попадания данных на вход второго компонента, h_{21}, h_{22}, является в данном случае свойством системы и рассматривается отдельно.

Надежности компонентов в подобластях обозначим R_{11}, R_{12}, R_{21}, R_{22} соответственно.

Надежность системы в данном случае рассчитывается по формуле:

$$R = R_1 R_2 = (R_{11} h_{11} + R_{12} h_{12})(R_{21} h_{21} + R_{22} h_{22}). \quad (2)$$

3.1.3. N компонент

При наличии в системе N компонент соединенных последовательно, где каждый компонент имеет m подобластей входных данных, надежность системы можно рассчитать по формуле:

$$R = \prod_{i=1}^{N} R_i = \prod_{i=1}^{N} \sum_{k=1}^{k=m} R_{ik} h_{ik}. \quad (3)$$

4. Абдуктивный логический вывод, ортонормальные вектора

Для подсчета надежности всей системы, используя традиционные методы [1], надежности всех компонент R_i должны быть известны.

Что делать в случае, если некоторые R_i все-таки неизвестны, как это часто бывает, а требуемая надежность всей системы задана? Например, некоторые компоненты разработчики системы могут купить, а другие им необходимо разработать самим. Какая надежность требуется для этих компонентов, какое время выделить для разработки и тестирования.

Из результатов (3) следует что в условиях максимальной энтропии (в нашем случае — независимость компонент), надежность системы с произвольной архитектурой является полилинейной формой от надежности компонент. Вид полилинейной формы является свойством архитектуры системы.

Полилинейная форма в общем случае является поверхностью L в N–мерном пространстве. Введем в рассмотрение единичный ортонормальный вектор i к поверхности L в точке требуемой надежности. Точка требуемой надежности интерпретируется в данном случае как требуемая финальная надежность R всей системы.

Рассмотрим проекцию данного вектора на интересующую нас секущую плоскость. Проекции ортонормальных векторов считаются при помощи частных производных. Расчет проекций для функции общего вида R приведен ниже.

Труды СПИИРАН. 2004. Вып. 2. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2004. Issue 2. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

$|i| = 1;$

$\frac{\partial}{\partial R_i} R = \text{tg}\ \alpha$

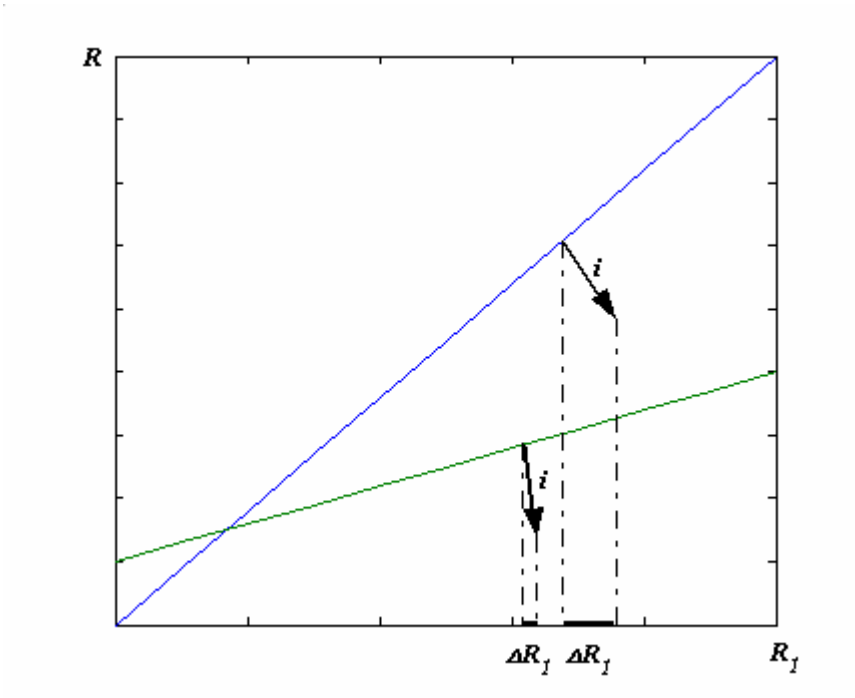
$\alpha = \text{arctg}\ (\frac{\partial}{\partial R_i} R);$

$\sin \alpha = \frac{\Delta Ri}{|i|};$

$\Delta R_i = \sin \alpha |i| = \sin \alpha\alpha$

$\Delta R_i = \sin (\text{arctg}\ (\frac{\partial}{\partial R_i} R))$.

В случае одного неизвестного параметра, задача вырождается в линейную зависимость.



В этом случае:
$R = aR_1 + b;$

$\Delta R_1 = \sin(\text{arctg}\ \frac{\partial}{\partial R_1} R) = \sin a.$

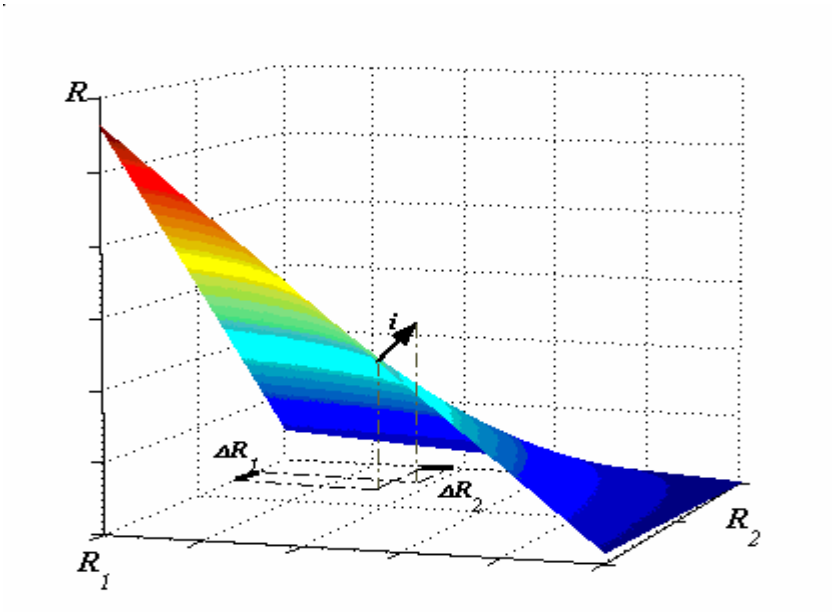
В случае двух и более неизвестных параметров, появляется поверхность:
Расчет проекций при $N = 2$ приведен ниже:

$R = aR_1 + bR_2 + cR_1R_2 + k;$

$\Delta R_1 = \sin \text{arctg}\ (\frac{\partial}{\partial R_1} R) = \sin \text{arctg}\ (a + cR_2);$

$\Delta R_2 = \sin \text{arctg}\ (\frac{\partial}{\partial R_2} R) = \sin \text{arctg}\ (b + cR_1).$

Труды СПИИРАН. 2004. Вып. 2. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2004. Issue 2. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru



В случае многомерной поверхности, расчет производится аналогичным образом и здесь не приводится.

При рассмотрении данной модели допустимо предположение гладкости поверхности L. При таком допущении, проекции ортонормального вектора отражают степень влияния соответствующего фактора надежности компонента — параметра R_i — на надежность всей системы R. Проекцию ΔR_i назовем фактором чувствительности надежности системы. Таким образом, появляется возможность сделать качественные оценки жизнеспособности системы при помощи достаточно простых автоматизированных расчетов еще на стадии ее проектирования, оценивать время выполнения проекта и делать выбор компонентов более обоснованно.

Вычисления требуемых надежностей компонент могут быть произведены также на стадии проектирования при помощи традиционных методов решения систем уравнений, метода Монте-Карло и с использованием технологий программирования в ограничениях (Constraint Logic Programming). Это предмет будущих исследований.

5. Нерешенные вопросы

Выбор «подобластей эквивалентности» в рамках приведенных выше моделей — задача разработчика компонента и нет формальных методов разделения на подобласти. Можно выдавать рекомендации по разбиению, например по условным переходам программы, но в целом этот вопрос — предмет будущих исследований. Возможно, эту проблему можно решить более мелким дроблением областей.

Интересно также рассмотреть, как меняются проекции, если менять точку финальной надежности — параметр R. Возможно, здесь есть область для исследования в направлении более точных оценок.

Модель, предложенную в данной статье, вероятно, возможно применять и для оценок параметров h — вероятностей попадания входных данных в подобласть. Это актуально для компонентов внутри системы, для которых необ-

Труды СПИИРАН. 2004. Вып. 2. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2004. Issue 2. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

ходим расчет преобразования профилей. В этом случае возникает вопрос: возможно ли применять нашу модель и для расчета преобразования профилей.

Важным является рассмотрение поведения систем при значениях вероятностей близких к 1, поскольку значения, существенно отличающиеся от 1, могут быть неинформативны и желательно рассмотреть вопрос реорганизации системы при получении таких значений.

6. Сравнение моделей

Модель Хамлета [1] не применима в случае, если хотя бы один компонент описан не формально или не известны его параметры, мы предлагаем способ решения этой проблемы.

В модели Хамлета — надежность системы вычисляется в конце и разработчик архитектуры должен действовать методом проб и ошибок, в нашей модели — мы в самом начале сужаем круг рассматриваемых кандидатов.

В нашем подходе важнейшей является идея деления области входных данных на подобласти эквивалентности, которая принадлежит Хамлету [1].

7. Литература

[1] *Hamlet D., Mason D., Woit D.* Theory of Software Reliability Based on Components // International Conference on Software Engineering(ICSE), Toronto, Canada, 2001. P. 361–370.

[2] Hamlet D., Random Testing and Subdomain Testing. Encyclopedia of Software Engineering, 2nd ed., book chapter, 2002. P. 132–158.

[3] *Valkovsky V. B., Savvin K. O., Gerasimov M. B.* Abduction Problem in Probabilistic Constraint Logic Programming. Springer-Verlag, Ltd., London, 1999. P. 685–698.

[4] *Hamlet D.* Continuity in Software Systems // Proceedings ISSTA 02, Rome, 2002. P. 196–200.