УДК 004.2, 004.9

ПРИНЦИПЫ ВЕРИФИКАЦИИ ИНФОРМАЦИОННЫХ МОДЕЛЕЙ И АЛГОРИТМОВ

***Павел Юрьевич Елсуков**, канд. техн. наук, ст. науч. сотр.,*
e-mail: elsukov_p@bk.ru,
Институт систем энергетики им. Л. А. Мелентьева
Сибирского отделения Российской академии наук (ИСЭМ СО РАН),
http://isem.irk.ru

Статья раскрывает содержание принципов верификации информационных моделей. Вводится понятие верификации модели. Вводится понятие алгоритма реализации. Показано, что верификация программного обеспечения является основой для верификации информационных моделей. Показано, что алгоритм реализации информационной верифицированной модели обладает большей надежностью в сравнении с обычным алгоритмом. Статья показывает априорные и апостериорные принципы верификации информационных моделей. Показано различие между алгоритмом и алгоритмом реализации.

Ключевые слова: информационные модели; верификация; алгоритм; алгоритм реализации; темпоральная логика.

Введение

DOI: 10.21777/2500-2112-2017-2-81-86

В последнее время широкое применение находят информационные модели и алгоритмы их реализации. При этом возрастает сложность разрабатываемых информационных моделей [1, 2]. Практические задачи требуют все более сложных моделей, а технологии их проектирования не могут обеспечить требуемое качество и надежность [3, 4]. Информационные модели и алгоритмы на их основе могут длительное время сохранять и накапливать ошибки, проявляющиеся после длительной эксплуатации. Это усугубляется как реакция на сложную комбинацию многочисленных факторов, в частности непредсказуемости информационных ситуаций [5, 6] и сложного взаимодействия совокупности процессов. Возникает необходимость верификации информационных моделей подобно верификации программного обеспечения [7, 8]. Особенность этой технологии в том, что верификация информационных моделей в одних случаях является и верификацией алгоритма реализации, в других случаях требуется дополнительная верификация алгоритма, связанная с верификацией модели. Информационные модели часто строятся на проектируемые информационные системы. Поэтому верификация таких информационных моделей обеспечивает верификацию информационных систем. Информационные системы управления часто строятся из взаимодействующих модулей [9]. Ошибки возникают не только в модулях, но и в их взаимодействии. Эти ошибки могут быть критическими. Устранение подобных ошибок требует верификации модели системы и самой модели как реализации. Таким образом, верификация информационных моделей частично решает задачи алгоритма реализации или задачи верификации информационной системы, реализованной с использованием такой модели. Это делает актуальным исследование верификации информационных моделей.

П.Ю. Елсуков

Анализ подходов к верификации

Верификация информационной конструкции (модели, программы, информационной системы) – логическое доказательство того, что данная информационная конструкция удовлетворяет формально определенным требованиям [10]. Термин информационная конструкция [11, 12] – обобщающее понятие объектов информационного поля.

Многие информационные модели переходят из одного состояния в другое. Такие модели называют трансформационными. Состояние может быть формализовано как

вектор значений характеристик состояния. Одна из проблем информационных конструкций моделей в том, что они развиваются и меняются во времени.

Обычная логика относится к стационарным состояниям. Поэтому нельзя верифицировать модели, изменяющиеся с течением времени, так как обычная классическая логика не отражает временных зависимостей. Ограниченность классической логики для выражения свойств динамики (процессов, развивающихся во времени) в том, что высказывания статичны, неизменны во времени; коммутативные статические выражения – не коммутативны во времени. Это исключает применение классической логики для проверки динамических моделей и систем. Для исследования динамических процессов и систем применяют темпоральные логики [13], которые своими корнями уходят в модальные логики [14]. Темпоральные логики используют выражения, истинность которых зависит от временных характеристик: временных интервалов и временных последовательностей.

На рис. 1 показаны принципы действия логических темпоральных операторов. На рис. 1 приведены следующие операторы:

G – «всегда», *Gq* – *q* всегда будет в будущем;
R – «истинно (выполнено) до тех пор, пока не появится утв. 1»;
Q – «истинно после»;
U – «когда-нибудь наступит *R*, а до него все время будет *Q*»;
Between Q and R – «истинно между *Q* и *R*»;
After Q until R – «истинно после *Q* до *R*».

С помощью таких информационных конструкций можно моделировать изменение ситуаций и событий. Темпоральные логики делятся на линейные темпоральные логики (LTL) и ветвящиеся темпоральные логики (CTL) [15].

Формула LTL – это атомарное утверждение (атомарный предикат), или формулы, связанные логическими операторами, или формулы, связанные темпоральными операторами *U*, *X*. Модальных операторов прошлого в LTL нет. Атомарные предикаты – базисные свойства процесса в состояниях. Производные темпоральные формулы в состояниях – это свойства вычисления в будущем, динамика процесса. Последовательность в темпоральной логике можно толковать как бесконечную последовательность состояний дискретной системы, а отношение достижимости – как дискретные переходы системы.

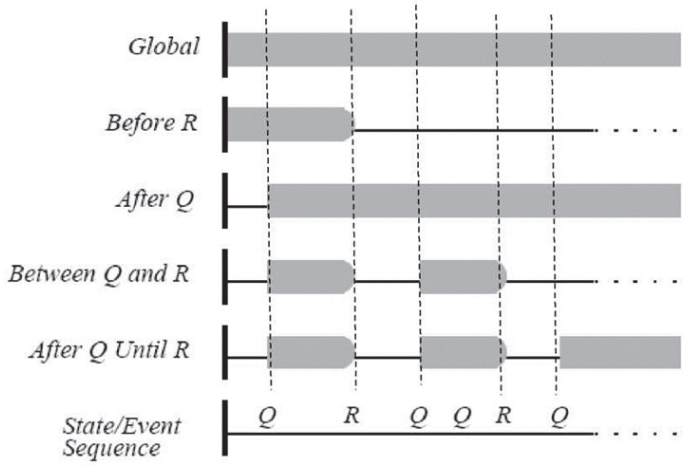

Рис. 1. Принципы логических темпоральных операторов

В последнее время совершен качественный прорыв в области верификации. Разработан метод model checking (проверка модели), основанный на формальных моделях [3, 4, 16, 17]. Метод model checking включает следующие этапы, которые могут быть положены в основу верификации ИС: построение формальной модели системы; построение формального информационного языка требований; выполнение процедуры проверки формальной модели с использованием языка требований.

На рис. 2 приведены две схемы model checking: базовая рекурсивная и развернутая.

Темпоральные логики применяют кванторы пути: **A** – «выполнено для всех путей» по аналогии с квантором общности; **E** – «для некоторого пути», по аналогии с квантором существования. На рис. 2: S0 – начальное состояние; S1 – состояние 1; S2 – состояние 2. Переход S0 → S1 отражает нормальное функционирование, переход S0 → S2 ведет в тупик – ненормальное функционирование. Процесс верификации означает

поиск тупиковых ситуаций и их устранение.

Одной из проблем верификации, особенно для сложных моделей и систем [18], является большое число проверок и вычислений. Верификация информационной модели не самоцель, а средство повышения ее надежности и качественной работы алгоритма или системы которую она моделирует. Это определяет принцип верификации информационной модели не только на основе темпоральной логики, но и на основе системного подхода [19, 20]. При этом фрагменты верификации можно проводить на стадии проектирования информационной модели и на стадии реализации информационной модели в алгоритм или систему. Это повышает надежность результата реализации информационной модели. По архитектуре и проектированию информационные конструкции и информационные модели можно поделить на более устойчивые и менее устойчивые. Более устойчивые – это модели, которые сохраняют состояние устойчивости и работоспособность при широком диапазоне внешних воздействий. Их структура, как правило, логически выверенная и они не имеют «зацикливаний» и паразитных связей.

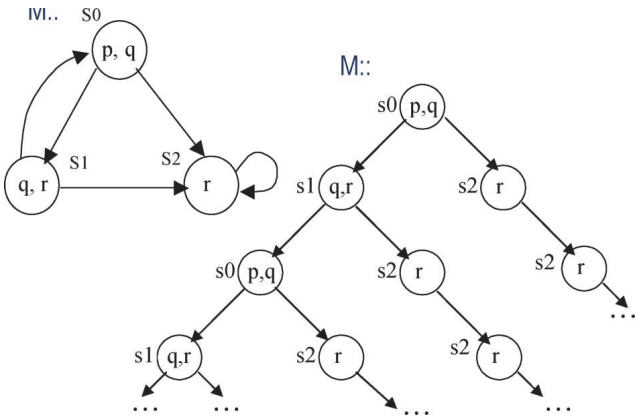

Рис. 2. Схемы проверки

Для создания таких информационных моделей применяют специальные методы проектирования, а для алгоритмов – методы структурного проектирования. К числу методов верификации надежных моделей относится Model Driven Development [21]. В процессе проектирования эти модели включают: спецификацию свойств, разработку модели ИС, разработку тестов, верификацию модели, имитационное моделирование, окончательное формирование проекта. Такая ИС является верифицированной уже на стадии проекта, в отличие от других, которые начинают верифицировать в процессе эксплуатации или после реализации.

Еще один подход к верификации основан на использовании бинарных решающих диаграмм (Binary Decision Diagrams – BBD) [22]. Он применим только тогда, когда на основе декомпозиции ИС можно использовать булевы функции. Использование BDD в алгоритмах верификации позволило увеличить сложность верифицируемых систем (число состояний структуры Крипке) в миллиарды миллиардов раз – с 10^6 до 10^{300}.

Еще один подход к верификации основан на применении принципа композициональности и рекурсивности. Он заключается в выводе о глобальном поведении информационной модели и алгоритма реализации, полагаясь на локальные свойства ее составляющих. Эта идея заимствована из теории фракталов [23]. Термин «фрактал» употребляется не только в математике. Фракталом может называться предмет, обладающий, по крайней мере, одним из указанных ниже свойств:

- обладает нетривиальной, меняющейся структурой на всех масштабах. В этом отличие фрактала от регулярных фигур (таких, как окружность, эллипс, график гладкой функции): если рассмотреть небольшой фрагмент регулярной фигуры в очень крупном масштабе, то он будет похож на фрагмент прямой;
- для фрактала увеличение масштаба не ведет к упрощению структуры, то есть на всех шкалах можно увидеть одинаково сложную картину;
- фрактал является приближенно самоподобным;
- фрактал обладает рекурсивной структурой, которая содержит повторяющиеся компоненты;
- фрактал обладает дробной метрической размерностью или метрической размерностью, превосходящей топологическую.

Многие объекты в природе обладают свойствами фрактала, например: побережья, облака, кроны деревьев, снежинки, кровеносная система, интегральные схемы.

Фрактальный подход применим не ко всем моделям и алгоритмам, а только к тем, которые имеют повторяющуюся структуру и обладают одним из свойств фрактала. Суть фрактального подхода в области верификации состоит в повышении устойчивости информационной модели или алгоритма реализации, логики структуры и исключении неизвестных и математически не описываемых элементов структуры. Многие рекурсивные структуры, построенные на интегральных схемах, являются фрактальными. На рис. 3 показана такая рекурсивная структура.

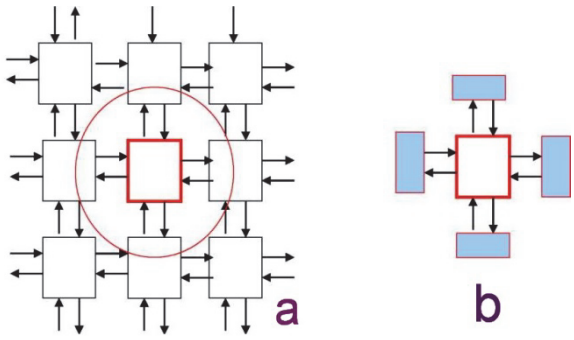

Рис. 3. Компонентная рекурсивная структура

Рекурсивная структура состоит из повторяющихся модулей (рис. 3a). Принцип компонентной верификации модели такой системы заключается в том, что вместо верификации всей системы (рис. 3a) верифицируют отдельный повторяющийся компонент с моделированием всевозможных межкомпоненетных связей (рис. 3b). Эта методика заимствована из САПР, где надежность проверяют на отдельных сложных узлах в первую очередь. Кстати, именно методы САПР служат основой для конструирования интегральных схем, что подчеркивает общность и рекурсивность моделей, схем и систем в этой области.

Обычный метод спецификации требований вытекает из технического задания, написанного на естественном языке. Для информационных систем используют два вида спецификаций: Z-спецификация; B-спецификация.

Еще один принцип верификации строится на модели «от обратного». Воспользуемся топологией состояний. Тупиковую вершину обозначим g (рис. 4) и закрасим ее черным цветом. В формализме темпоральной логики такая модель имеет описание

$$M,s_0 \models EG\ g. \qquad (1)$$

Здесь E – квантор «для некоторого пути», G – оператор темпоральной логики «всегда». Выражение (1) интерпретируется так: для некоторого пути рано или поздно g.

Принцип верификации «от обратного» основан на поиске тупиковых состояний, а не проверке всех или только допустимых.

Дискуссия

Проведенный анализ дает основание ввести понятие алгоритма реализации. Это новое понятие в теории вычислений. Алгоритм реализации обладает свойствами наследования от модели, реализацией которой он является. Наследование распространяется и на верифицируемость. При построении обычного алгоритма строится простая (базисная) логическая структура, которая затем усложняется и модифицируется. Усложненная структура затем подвергается анализу и верификации «с нуля». В отличие от этого алгоритма, алгоритм реализации уже верифицированный на основе верификации модели. В случае рекурсивной или фрактальной структуры верифицируемость такого алгоритма высокая, как и надежность. Соответственно, если на основе верифицированной информационной модели строится информационная система, то она уже по определению будет более надежной, так как частично верифицирована еще до построения и реализации.

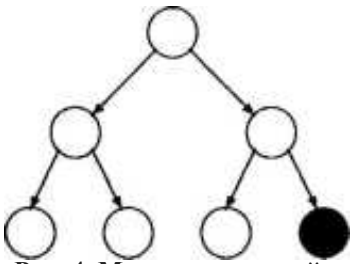
Рис. 4. Модель состояний с одной тупиковой вершиной

Верификация информационных моделей и алгоритмов подразделяется на априорную и апостериорную. Верификация обычных алгоритмов всегда апостериорная. Верификация алгоритмов реализации включает априорную и апостериорную составляю-

щие. При этом чем более верифицирована информационная модель или чем более надежный способ ее построения, тем более верифицируем алгоритм реализации и меньше требуется апостериорной верификации.

Заключение

Верификация информационных моделей повышает надежность алгоритма реализации и устойчивость его работы. Современная верификация информационных моделей во многом опирается на опыт верификации программного обеспечения [10]. Принципиально процесс верификации информационной модели включает два пути. Первый путь включает поиск всех возможных состояний и последующее доказательство правильности модели во всех состояниях. Этот путь включает доказательство того, что эти состояния попадают в область истинности. Второй путь включает поиск тупиковых ситуаций и их устранение. Этот путь включает поиск областей неистинности и доказательство того, что все состояния модели не попадают в область истинности. Первый путь может быть не ограничен. Второй путь всегда ограничен. Это определяет его предпочтительность при анализе. Применение верификации методом Model checking дает инструментарий, с помощью которого можно проверять корректность сложных информационных моделей. Нельзя рекомендовать один путь верификации для всех информационных моделей. В зависимости от типа моделей тот или иной метод дает наибольший эффект.

Литература

1. *Болбаков Р. Г.* Анализ сложности информационных конструкций // Перспективы науки и образования. 2016. № 5. С. 11–14.

2. *TsvetkovV. Ya*. Complexity Index // European Journal of Technology and Design. 2013. Vol. 1. Iss. 1. P. 64–69.

3. *McMillan K. L.* Symbolic model checking. An approach to the state explosion problem: thesis of degree of Doctor of Philosophy in Computer Science. – Carnegie Mellon Universitat, 1992. 214 p.

4. *McMillan K. L.* Using unfoldings to avoid the state explosion problem in the verification of asynchronous circuits // Computer Aided Verification. – Berlin, Heidelberg: Springer, 1993. P. 164–177.

5. *Tsvetkov V. Ya.* Information Situation and Information Position as a Management Tool // European researcher. Series A. 2012. Vol. 36. Iss. 12-1. P. 2166–2170.

6. *Tsvetkov V. Ya.* Dichotomic Assessment of Information Situations and Information Superiority // European researcher. Series A. 2014. Vol. 86. Iss. 11-1. P. 1901–1909.

7. *Гуров В. С., Шалыто А. А., Яминов Б. Р.* Технология верификации автоматных моделей программ без их трансляции во входной язык верификатора. – Таганрог: НИИ МВС ЮФ, 2007.

8. *Вельдер С. Э., Шалыто А. А.* Верификация автоматных моделей методом редуцированного графа переходов // Научно-технический вестник информационных технологий, механики и оптики. 2009. № 6 (64). С. 66–77.

9. *Wand Y., Weber R*. On the deep structure of information systems // Information Systems Journal. 1995. Vol. 5. Iss. 3. P. 203–223.

10. *Синицын С. В., Налютин Н. Ю.* Верификация программного обеспечения. – М.: БИНОМ, 2008. 368 с.

11. *Tsvetkov V. Ya.* Information Constructions // European Journal of Technology and Design. 2014. Vol. 5. Iss. 3. P. 147–152.

12. *Дешко И. П.* Информационное конструирование: монография. – М.: МАКС Пресс, 2016. 64 с.

13. *Clarke E. M., Grumberg O.* Avoiding the state explosion problem in temporal logic model checking // Proceedings of the sixth annual ACM Symposium on Principles of distributed computing. – ACM, 1987. P. 294–303.

14. *Chellas B. F.* Modal logic: an introduction. – Cambridge: Cambridge University Press, 1980. 316 p.

15. *Цветков В. Я.* Применение темпоральной логики для обновления информационных конструкций // Славянский форум. 2015. № 1 (7). С. 286–292.

16. *Кларк Э., Грамберг О., Пелед Д.* Верификация моделей программ: Model checking –

М.: МЦНМО, 2002.

17. *Карпов Ю. Г.* Model Checking. Верификация параллельных и распределенных программных систем. – СПб.: БХВ-Петербург, 2010.

18. *Железняков В. А.* Уровни сложности информационных систем // Славянский форум. 2015. № 3 (9). С. 97–104.

19. *Монахов С. В., Савиных В. П., Цветков В. Я.* Методология анализа и проектирования сложных информационных систем. – М.: Просвещение, 2005. 264 с.

20. *Tsvetkov V. Ya.* Dichotomous Systemic Analysis // Life Science Journal. 2014. Vol. 11. Iss. 6. P. 586–590.

21. *Pastor O. et al.* Model-driven development // Informatik-Spektrum. 2008. Vol. 31. Iss. 5. P. 394–407.

22. *Akers S. B.* Binary decision diagrams // IEEE Transactions on Computers. 1978. Vol. 100. Iss. 6. P. 509–516.

23. *Мандельброт Б.* Фрактальная геометрия природы. – М.: Институт компьютерных исследований, 2002.

Principles of verification of information models and algorithms

***Pavel Yur'evich Elsukov**, Federal State Institution of Science Institute of Energy Systems. LA Melentyeva Siberian Branch of the Russian Academy of Sciences (ESI SB RAS), Irkutsk, Russia*

The article discloses the content of the principles of verification of information models. The article introduces the concept of model verification. The concept of an implementation algorithm is introduced. The article proves that verification of software is the basis for verification of information models. The article shows that the algorithm for implementing the information verified model has greater reliability in comparison with the conventional algorithm. The article shows the a priori and a posteriori principles of verification of information models. The article describes the difference between the algorithm and the implementation algorithm.

Keywords: Information models, verification, algorithm, implementation algorithm, temporal logic.

УДК 528.2/.5 528.8 528.02

СЕТЕЦЕНТРИЧЕСКОЕ УПРАВЛЕНИЕ И КИБЕРФИЗИЧЕСКИЕ СИСТЕМЫ

***Станислав Алексеевич Кудж**, профессор, д-р техн. наук,*
e-mail:
ректор Московского технологического университета (МИРЭА),
***Виктор Яковлевич Цветков**, профессор, д-р техн. наук,*
e-mail: cvj2@mail.ru
Московский технологический университет (МИРЭА)
https://www.mirea.ru

Статья описывает киберфизические системы и киберфизическое управление. Описана связь киберфизического управления с сетецентрическим управлением. Описана эволюция технических систем, которая привела к появлению киберфизических систем. Описаны технологии иерархического и матричного управления как прототипы киберфизического управления. Раскрыто содержание принципов киберфизического управления. Вводится и раскрывается понятие гармонизирующего информационного потока. Раскрывается содержание интеллектуального узла. Раскрывается технология киберфизического управления.

Ключевые слова: управление; интеллектуальное управление; распределенные системы; интеллектуальный узел; гармонизирующий информационный поток; субсидиарное управление; сетецентрическое управление; киберфизическое управление.