УДК 004.414.23

Д. В. ГРОЦЕВ, Т. А. ШМЫГАЛЕВА

ИТЕРАТИВНОЕ ФОРМАЛЬНОЕ МОДЕЛИРОВАНИЕ МАСШТАБИРУЕМОЙ РАСПРЕДЕЛЕННОЙ СИСТЕМЫ

В работе формальный метод Event-B применен для пошагового моделирования масштабируемой распределенной системы. Приведен пример эффективной гиперкубической топологии сети. В нулевой абстрактной модели введены операции масштабирования. В первой детализации они разбиты на две фазы. Иерархическая структура узлов в сети разработана во второй детализации. Модель автоматически верифицирована в среде Rodin.

Формальный метод Event-B. В Event-B модель описывает состояния дискретной системы и переходы между ними. Он основан на классическом методе B [1]. Суть и преимущество метода Event-B заключены в том, что моделирование происходит поэтапно. На каждом шаге модель уточняется новыми свойствами. Разработка поддерживается интегрированной средой Rodin [2, 3], которая автоматически формулирует теоремы о корректности модели. Утверждения доказываются вручную или машиной. Широкий набор теорем, подлежащих доказательству при верификации, дан в [4].

Event-B модель разбивается на статическую часть – контекст и динамическую часть – машину. Контекст с именем *ctx* включает константы c, множества s, список аксиом $P(c, s)$, который ограничивает c и s. Состояние машины с именем *mch* описывается вектором переменных v и ограничивается инвариантами $I(c, s, v)$. Начальное состояние инициализируется событием $R\_I(c, s, v')$, а затем изменяется событиями E. Событие *evt* содержит вектор параметров p. Действие $S(c, s, v, v')$ задает связь между значением переменной v до и v' после совершения события и атомарно выполняется, когда удовлетворено сторожевое условие $H(c, s, v)$.

Масштабируемые распределенные системы. Если в системе существует узел, с которым связаны все остальные, то система называется централизованной, иначе – распределенной системой.

В масштабируемой системе количество узлов может изменяться.

n-мерный тор T^n – есть произведение n окружностей S^1. Сеть с топологией гиперкуба получается из T^n, если S^1 представить в виде двух связанных узлов. Она эффективна тем, что валентность узла и диаметр сети растет со скоростью порядка логарифма от количества узлов сети.

Абстрактная нулевая модель. Масштабирование. Машина *m0* (рис. 1б) имеет только одну переменную масштаба s (inv1). Для гиперкубической топологии масштаб s интерпретируется как размерность сети (рис. 1, *а*).

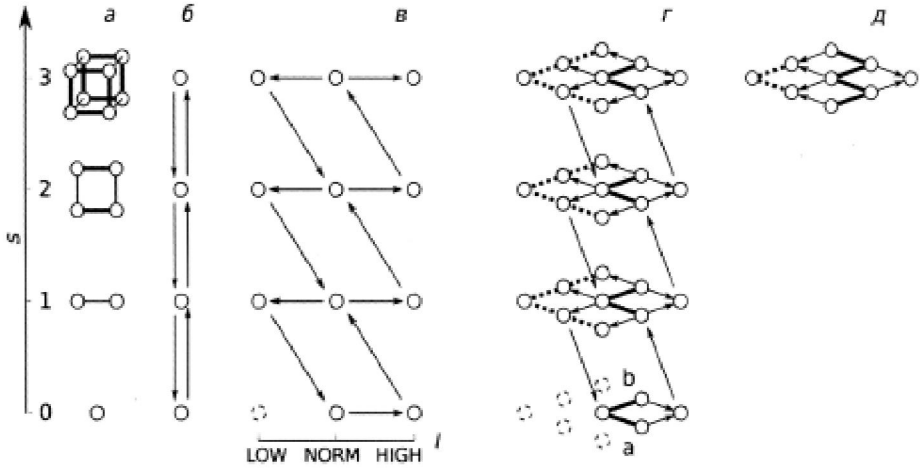

Рис. 1. *а* – гиперкубы; *б* – нулевая модель; *в* – первая детализация; *г* – вторая детализация; *д* – альтернативное разбиение комбинаций *a* и *b*

Событие *scaleup* увеличивает масштаб на единицу. Оно представлено семейством стрелок вверх. Событие *scaledown* уменьшает масштаб на единицу, если тот положителен, и представлено семейством стрелок вниз.

```
machine m0
variables s
invariants
 @inv1 s ∈ ℕ
events
 event INITIALISATION then @act1 s ≔ 0 end
 event scaleup then @act1 s ≔ s + 1 end
 event scaledown where @grd1 s > 0 then @act1 s ≔ s − 1 end
end
```

Первая детализация. Двухфазное масштабирование. В контексте *c1* объявлено множество уровней заполненности *LEVEL* из трех элементов: *LOW* – низкая, *NORM* – нормальная, *HIGH* – высокая заполненность.

```
context c1
constants LOW NORM HIGH
sets LEVEL
axioms
 @axm1 partition(LEVEL, {LOW}, {NORM}, {HIGH})
end
```

В машине *m1* к переменной масштаба *s* (inv1) добавляется переменная степени заполненности *l* (inv2). Допустимые состояния представлены сплошными кружками (рис. 1, *в*), а запрещенное (inv3) – штрихованным. Детализация состояния в машине *m1* заменяем один кружок на три.

Масштабирование разбивается на две фазы для синхронизации распределенного состояния [5]. Первая фаза обнаруживает ненормальную – высокую в событии *overflow* и низкую в событии *underflow* – заполненность. Вторая фаза приводит заполненность к нормальной.

```
machine m1 refines m0 sees c1
variables s l
invariants
 @inv1 s ∈ ℕ
 @inv2 l ∈ LEVEL
 @inv3 ¬ (s=0 ∧ l=LOW)
events
 event INITIALISATION then @act1 s ≔ 0 @act2 l ≔ NORM end
 event scaleup refines scaleup
  where @grd1 l = HIGH
  then @act1 s ≔ s + 1 @act2 l ≔ NORM
 end
 event scaledown refines scaledown
  where @grd1 l = LOW
  then @act1 s ≔ s − 1 @act2 l ≔ NORM
 end
 event overflow
  where @grd1 l = NORM
  then @act1 l ≔ HIGH
 end
 event underflow
  where @grd1 l = NORM @grd2 s > 0
  then @act1 l ≔ LOW
 end
end
```

Проверка $s > 0$ переносится в *underflow* благодаря инварианту (inv3).

Детализация заменяет вертикальные стрелки на диагональные и добавляет стрелки направо *overflow* и налево *underflow* (рис. 1, *в*).

Вторая детализация. Иерархическое масштабирование. Машина *m2* наследует от машины *m1* переменную масштаба *s* (inv1). Но разбивает абстрактную переменную заполненности *l* на две половины *a* (inv2) и *b* (inv3), каждая из которых является машиной предыдущего масштаба. Половины дальше рекурсивно разбиваются, кроме машины нулевого масштаба, в которой половины фиктивны. Узлы в половинах соединены жирными линиями (рис. 1, *а*). Запрещенное состояние разбивается на пять состояний (inv4), показанных штрихованными кружками (рис. 1, *г*).

```
machine m2 refines m1 sees c1
variables s a b
invariants
 @inv1 s ∈ ℕ
 @inv2 a ∈ LEVEL
 @inv3 b ∈ LEVEL
 @inv4 ¬ (s=0 ∧ a=LOW ∧ b=LOW)
 @inv5 l=HIGH ⇔ a=HIGH ∧ b=HIGH
 @inv6 l=LOW ⇔ a=LOW ∨ b=LOW
events
 event INITIALISATION then @act1 s ≔ 0 @act2 a ≔ NORM @act3 b ≔ NORM end
 event scaleup refines scaleup
  where @grd1 a = HIGH @grd2 b = HIGH
  then
   @act1 s ≔ s + 1
   @act2 a,b :| a'↦b' ∈ {NORM↦NORM, NORM↦HIGH, HIGH↦NORM}
 end
 event scaledown refines scaledown
  where @grd1 a=LOW ∨ b=LOW
  then
   @act1 s ≔ s − 1
   @act2 a,b :| a'↦b' ∈ {NORM↦NORM, NORM↦HIGH, HIGH↦NORM}
 end
 event overflow refines overflow
  where @grd1 a↦b ∈ {NORM↦HIGH, HIGH↦NORM}
  then @act1 a ≔ HIGH @act2 b ≔ HIGH
 end
 event underflow refines underflow
  where @grd1 a↦b ∈ {NORM↦NORM, NORM↦HIGH, HIGH↦NORM} @grd2 s > 0
  then @act1 a,b :| a'=LOW ∨ b'=LOW
 end
 event balance
  where @grd1 a↦b ∈ {NORM↦NORM, NORM↦HIGH, HIGH↦NORM}
  then @act1 a,b :| a'↦b' ∈ {NORM↦NORM, NORM↦HIGH, HIGH↦NORM} \ {a↦b}
 end
end
```

Каждая из половин *a* и *b* может находиться в одном из трех состояний, что дает девять комбинаций. Одна из них – это состояние высокой заполненности (inv5). Другие пять комбинаций – это состояния низкой заполненности (inv6). Они объединены штриховой жирной линией (рис. 1, *г*). Остальные три комбинации – это состояния нормальной заполненности. Они объединены сплошной жирной линией.

Далее несколько переходов условно отображаются меньшим числом стрелок. Событие *scaleup* описывает три перехода и отображено одной наклонной стрелкой вверх, *scaledown* описывает пятнадцать переходов и дано одной наклонной стрелкой вниз, *overflow* описывает два перехода и дано двумя стрелками вправо, *underflow* описывает пятнадцать переходов и представлено четырьмя стрелками влево. Новое событие *balance* отвечает за переходы между состояниями нормальной заполненности.

Склеивающие инварианты могут разбивать девять комбинаций на три части другим способом. Например, пять комбинаций отведены для нормальной заполненности и три – для низкой (рис. 1, *д*).

Структура переходов гарантирует, что перед масштабированием обязательно должна произойти синхронизация через события *overflow* или *underflow*. То есть запрещен транзитный переход через несколько масштабов или немедленный возврат к предыдущему значению масштаба.

Заключение. Систему можно смоделировать бесчисленным множеством способов. Состояния системы при взгляде с различных углов зрения предстает в виде разных наборов переменных и ограничений. Метод Event-B предлагает итеративный процесс разработки, когда точка зрения на систему переносится по такой траектории, что, во-первых, не нарушаются уже известные свойства системы, во-вторых, состояние разлагается на более мелкие детали, у которых добавляются новые свойства. Важно, чтобы преобразование не нарушало старых свойств системы. Теоремы об этом автоматически генерирует среда Rodin.

Модель будет использоваться для разработки распределенной коммуникационной среды [6–8], через которую взаимодействуют агенты.

ЛИТЕРАТУРА

1. *Abrial J.R.* The B-Book: Assigning Programs to Meanings. Cambridge University Press, 2005.

2. RODIN Event-B Platform. 2007. http://rodin-b-sharp.sourceforge.net/

3. EU research project IST 511599 RODIN (Rigorous Open Development Environment for Complex Systems). 2007. http://rodin.cs.ncl.ac.uk/

4. *Metayer C., Abrial J.R., Voisin L.* Rodin Deliverable D7: Event B language. Project IST-511599, School of Computing Science, Newcastle University, 2005.

5. *Moss, Elliot.* Nested Transactions: An Approach to Reliable Distributed Computing // The MIT Press. Cambridge, Massachusetts, 1985. P. 31-38.

6. *Lesser V.R., Wileden J.C.* Issues in the Design of Tools for Distributed Software System Development // Software Development Tools. Springer-Verlag, 1980.

7. *Lenat D.B.* On automated scientific theory formation: a case study using the AM program // Machine Intelligence. 1977. V. 9. P. 251-256.

8. *Gelernter D.* Generative Communication in Linda // ACM Transactions on Programming Languages and Systems. 1985. V. 7. P. 80-112.

Резюме

Event-B формалды масштабталған үлестірілмелі жүйелерін қадам бойынша пішіндеу үшін қолданылған. Event-B формалды әдісі сипатталып, масштабталған үлестірілмелі жүйеге анықтама берілген. Желінің тиімді гиперкубтық топологиясына мысал келтіріле отырып, нөлдік абстрактылы пішініне масштабтау операциясы енгізілген. Алғашқы бөлшектеуде олар екі фазаға бөлінген. Желідегі торлардың иерархиялық құрылымы екінші бөлшектеуде жүзеге асқан. Пішін Rodin-ді өңдеу интегралданған ортасында автоматты түрде тексеріледі.

Summary

The paper discusses the application of Event-B formal method to refine the specification of scalable distributed system. Event-B formal method is described. The definition of scalable distributed system is given. Effective hypercube network topology is given for instance. Scale operations are introduced in the abstract model. The operations are divided into two phases in the first refinement. Hierarchy of network nodes is developed in the second refinement. Model is automatically verified by Rodin integrated development environment.

КазНУ им. аль-Фараби, г. Алматы Поступила 6.01.2010г.