УДК 681.3

ESTELLE-СПЕЦИФИКАЦИИ СЕТЕВЫХ ПРОТОКОЛОВ ПЕРЕДАЧИ ДАННЫХ

к.т.н. О.Г. Старусев
(представил д.т.н., проф. В.Д. Дмитриенко)

Постановка проблемы. Построение и верификация сетевых протоколов – одна из актуальных проблем современной разработки распределенных и клиент/серверных систем. Не смотря на определенный прогресс в этой области и богатство методов, полученные результаты не очень широко используются при практической деятельности.

На практике используются два крупных класса подходов.

Первый класс – техники формального описания (Formal Description Technique), основанные на модели расширенных конечных автоматов. Это такие языки как Estelle [1], SDL [2] и т.д. Преимущество этих подходов состоит в простоте и выразительности представления. Однако эти преимущества, во-первых, затрудняют формальную верификацию систем, а во-вторых, при увеличении количества состояний существенно увеличивается сама модель.

Второй класс – техники, основанные на временной логике [3], логике высоких порядков [4] и моделях трассовых спецификаций [5]. Этот класс методов не подвержен росту числа состояний в моделях, но плохо понятен большинству специалистов из-за специфичного математического аппарата. К этому классу методов относятся такие методы как LOTOS [6], CSP [7], μCRL [8] и др. Языки выполнимых спецификаций LOTOS и Estelle, являются стандартами ISO, а SDL – стандартом ITU (бывший МККТТ).

Средства построения Estelle-спецификаций описаны в работах [9 – 11].

Анализ литературы. Язык Estelle (Extended Finite State Machine Language) основан на модели расширенного конечного автомата [1]. На сегодняшний день существует стандарт языка [12]. Язык спецификаций представляет собой язык программирования, семантически и синтаксически близкий к языку Паскаль. Спецификация на языке Estelle описывает иерархически структурированную динамическую систему недетерминированных компонент, взаимодействующих между собой при помощи сообщений по двунаправленным связям между портами (точками

взаимодействия). Каждая компонента – экземпляр модуля.

Модуль, в котором не описано ни одного перехода, называется *неактивным*. После инициализации неактивный модуль не выполняет ни одного действия, тогда как его наследники могут их выполнять. Модуль с непустым набором переходов называется *активным*.

Каждый модуль может быть классифицирован как *системный процесс, системная активность, процесс или активность*. Когда не важен класс модуля, специфицированного как системный процесс или системная активность, его называют *системным модулем*. Классификация модулей наряду с их текстуальной вложенностью определяет поведение модулей относительно друг друга и должна удовлетворять следующим требованиям:

– каждый активный модуль должен иметь определенный класс;

– системный модуль не может быть вложен в активный модуль;

– каждый модуль, специфицированный как процесс или активность, должен быть включен в системный модуль;

– процесс может включать в себя описания процессов и активностей, а активность – только описания активностей.

Положение модуля в иерархии модулей спецификации определяется вложенностью его описания в описание модуля-родителя. Количество экземпляров данного модуля может динамически изменяться в процессе выполнения спецификации, но позиция в общей иерархии остается неизменной. Неактивный модуль, охватывающий все прочие модули системы, называется *спецификацией* распределенной системы. Если не существует охватывающего неактивного модуля, то спецификация задается одним системным модулем.

Структура связей между системными модулями описывается в разделе инициализации охватывающего неактивного модуля. Поскольку последний имеет пустой раздел описания переходов, количество системных модулей и структура связей между ними не меняются во время выполнения данной спецификации. Количество экземпляров модулей, находящихся на более низкой ступени иерархии, может изменяться.

Все множество экземпляров модулей, описания которых заключены в описании одного из системных модулей, называется *подсистемой*. В каждый момент времени в процессе выполнения спецификации каждая подсистема есть дерево экземпляров модулей с корнем в экземпляре соответствующего системного модуля. Каждая дуга в дереве экземпляров определяется отношением "родитель–наследник" между модулями.

Поведение системных модулей асинхронно. Внутри подсистем оно синхронизируется экземплярами родительских модулей. Каждый экзем-

пляр модуля предлагает один из своих готовых к выполнению переходов модулю-родителю. Переходы модуля-родителя имеют приоритет над переходами наследников. Его выполнение приостанавливает выполнение переходов всех (не только непосредственных) наследников. Классы "процесс" и "активность" определяют два возможных способа выполнения переходов: параллельное и недетерминированное последовательное. Если модуль-родитель классифицирован как процесс и не имеет переходов, готовых к выполнению, то на следующем такте будут параллельно выполнены все предложенные наследниками переходы. Такт не заканчивается, пока не выполнены все выбранные переходы. Если модуль-родитель классифицирован как активность и не имеет переходов, готовых к выполнению, то на следующем такте выполняется один из предложенных наследниками переходов. Выбор перехода для выполнения осуществляется недетерминировано.

Цель статьи. Целью этой статьи является описание спецификации протокола передачи данных при помощи техники Estelle.

Построение Estelle-спецификации протокола. В качестве объекта Estelle-спецификации был использован один из несложных, но очень часто используемых протоколов – протокол скользящего окна (Sliding Window Protocol, SWP) [13]. Класс содержит несколько протоколов, которые отличаются друг от друга эффективностью, сложностью и размером буфера. Во всех протоколах SWP каждый исходящий кадр содержит последовательных номер в диапазоне [0; max]. Для последовательного номера отводится n бит, поэтому $max = 2^n - 1$. В простых протоколах $n = 1$, а в более сложных n выбирается произвольно.

Суть протоколов SWP в том, что в любой момент времени отправитель поддерживает набор последовательных номеров, соответствующих кадрам, которые ему разрешено посылать (т.н. «посылающее окно»). Получатель поддерживает «принимающее окно», соответствующее набору кадров, которые ему разрешено принять. Окна получателя и отправителя могут иметь разный размер.

Ниже приведена только часть спецификации SWP-модуль передатчика. Спецификация была создана в Estelle Development Toolset.

```
type seq_type = integer;   { seq num type, must be >= 0 }
    user_data_type = ...;
    DTPDU_type = record
             seq: seq_type;
             msg: user_data_type;
           end;
    AKPDU_type = record
             seq: seq_type;
```

```
            end;
  body transmitter for transmitter_head;
    const transmitter_window_size = any integer;
    state SENDING;
    procedure buf_save(s : seq_type; d : user_data_type); primitive;
    procedure buf_free(s : seq_type); primitive;
    function  buf_retrieve(s : seq_type) : user_data_type; primitive;
    function  PDU_DT(s : seq_type; d : user_data_type) : DTPDU_type; primitive;
    var
      Lowest_Unacked : seq_type;
      Highest_Sent : seq_type;
      TWS : integer;
    initialize
      to SENDING  begin
         Lowest_Unacked := 1; Highest_Sent := 0;TWS := transmitter_window_size;
end;
    trans
      from SENDING to same
        when U.Data_Request
          provided Highest_Sent - Lowest_Unacked < TWS
            begin
              Highest_Sent := Highest_Sent + 1; output T.timer_request(Highest_Sent);
              output CT.DT(PDU_DT(Highest_Sent, data)); buf_save(Highest_Sent, data);
            end;
      from SENDING to same
        when CT.AK
          provided (PDU.seq >= Lowest_Unacked) and (PDU.seq <= Highest_Sent)
            var s : seq_type;
            begin
              for s := Lowest_Unacked to PDU.seq do begin
                output T.timer_cancel(s); buf_free(s); end;
              Lowest_Unacked := PDU.seq + 1;
            end;
          provided otherwise
            begin
            end;
      from SENDING to same
        when T.timer_response
          provided (seq >= Lowest_Unacked) and (seq <= Highest_Sent)
            var s : seq_type;
            begin
              for s := seq to Highest_Sent do begin
                output T.timer_cancel(s); output CT.DT(PDU_DT(s, buf_retrieve(s)));
                output T.timer_request(s);
              end;
            end;
  end; { transmr }
```

Выводы. Основным преимуществом Estelle-спецификаций является выразительная простота и возможность трансляции либо в исходный код языка высокого уровня (например, C или C++), либо в другой математический аппарат (например, раскрашенные сети Петри). Это позволит

получить как исходный код, так и дополнительные возможности при моделировании протокола.

В дальнейшем необходимо рассмотрение возможностей верификации построенных спецификаций при помощи «механических» средств доказательства правильности.

ЛИТЕРАТУРА

1. *Turner K.J. Using formal description techniques. – An Introduction to Estelle, Lotos and SDL. – John Wiley & Sons, 1993. – 431 p.*
2. *Doldi L. SDL Illustrated. – Visually design executable models. – NY, TMSO, 2000. – 270 p.*
3. *Boyer R.S., Moore J.S. A Computational Logic Handbook. – NY, Academic Press, 1998. – 518 p.*
4. *Manna Z. Mathematical Theory of Computation. – NY, McGraw-Hill, 1974. – 464 p.*
5. *Hoffman D., Snodgrass R. Trace Specifications: Methodology and Models // IEEE Trans. On Software Engineering. – Sept 1988. – Vol.14, issue 9. – P. 1243 – 1252.*
6. *Eijk P.H.J., Vissers C.A., Diaz M. The formal description technique LOTOS. – North-Holland: Elsevier Science Pub B.V., 1989. – 453 p.*
7. *Хоар Ч.А. Взаимодействующие последовательные процессы. – М.: Мир, 1989. – 264 с.*
8. *Groote J.F., Ponse A. The syntax and semantics of μCRL // A. Ponse, C. Verhoef, S.F.M. van Vlijmen. Algebra of Communicating Processes '94, Workshops in Computing Series. – Springer-Verlag. – 1995. – P. 26 – 62 .*
9. *Algayres B. et al. VESAR: a pragmatic approach to formal specification and verification // Computer Networks and ISDN Systems. – 1993. – Vol. 25, N. 7 – P. 779 – 790.*
10. *Budkowski S. Estelle development toolset (EDT) // Computer Networks and ISDN Systems. – 1992. – Vol. 25, N. 1. – P. 63 – 82.*
11. *Courtiat J. P., de Saqui-Sannes P. ESTIM: an integrated environment for the simulation and verification of OSI protocols specified in Estelle // Computer Networks and ISDN Systems. – 1992. – Vol. 25, N. 1. – P. 83 – 98.*
12. *ISO/IEC 9074:1997. Information technology – Open Systems Interconnection – Estelle: A formal description technique based on an extended state transition model. Amendment 1.*
13. *Таненбаум Э. Компьютерные сети. – С.-Пб.: Питер, 2002. – 848 с.*

Поступила 26.10.2004

СТАРУСЕВ Олег Геннадиевич, *канд. техн. наук, доцент кафедры НТУ "ХПИ". В 1993 году окончил ХПИ. Область научных интересов – формальные методы спецификации и тестирования программного обеспечения.*