维普资讯 http://www.cqvip.com

一种基于 RSL 的协议形式化描述技术的研究*)

赵 静 屈玉贵 赵保华
(中国科学技术大学计算机系 合肥230027)

Research on Formal Description Techniques of Protocols Based on RSL

ZHAO Jing QU Yu-Gui ZHAO Bao-Hua
(Department of Computer, University of Science and Technology of China, Hefei 230027)

Abstract This paper presents a formal technique of protocols based on RSL, which can efficiently connect the formal description, validation, implementation and testing of protocols. It mainly discusses the formal description of protocols based on RSL, and also provides an algorithm about this. Finally, taking the protocol AB as an example, a formal description based on RSL is given.

Keywords Protocol engineering, RSL, Formal description technique

1 引言

随着计算机网络和分布式系统的发展,特别是开放型异构的互联,协议本身的复杂性越来越高,为了使通信协议的开发过程能理论化和规范化地进行,协议工程学[1]已应运而生。它主要包括协议的形式化描述、协议验证、协议实现和协议测试等阶段,其中形式化描述技术(FDT)贯穿于整个协议开发过程的始终,是协议工程学的基础。

协议的形式化描述技术主要有模型技术以及形式描述语言(FDL),模型技术常用的有有限状态机模型(FSM),Petri网模型,时态逻辑 TL,通信进程演算 CCS 等;形式描述语言中影响大的有 ISO 组织公布的 ESTELLE[2]和 LOTOS[3],以及 ITU-T 组织颁布的 SDL,其中 ESTELLE 和 SDL 都是基于扩展有限状态机模型,LOTOS 基于通信进程演算 CCS。以上几种 FDT 在当前的协议工程领域得到了广泛的应用,但它们都存在着不同程度的缺陷,如当进行大型复杂协议的构造时,FSM 和 Petri 网模型将面临状态爆炸的问题;另外,这些 FDT 在描述协议的某些性质或在协议工程的某些阶段可能工作得较好,但在其它方面和阶段的表现则存在明显的缺陷甚至无法发挥作用。因此,可以说它们都没有能够真正实现形式化的协议工程。

我们的研究将要实现一种基于 RSL[4]的协议形式化技术,它能有效地连接协议形式化描述、协议验证、协议实现和协议测试等阶段。RSL(RAISE Specification Language)是一种"广谱"的语言,既可以用于书写抽象级别较高的规范,也可以用于书写易于转换到程序语言代码的抽象级别较低的规范。我们在 RSL 规范语言的基础上充分利用其优点,并在某些方面作适当的扩充,以便能有效地描述协议的各种性质。在描述一个具体协议时,可以根据其各主要功能的特点分别选择合适的模型作为基础,这样将有效地克服基于单一模型的缺陷。用基于 RSL 形式化描述的协议规范,其公理部分可用来帮助进行协议验证和协议测试;如果将此初始规范利用 RSL 的精化(refinement)关系不断精化,可以得到一个易于转化为程序语言代码的规范,从而有助于协议的实现;如果所有的规范、精化关系的证明以及各种文档都整理清楚,就为今后协议的维护和扩展打下良好基础。由此可见,采用基于 RSL 的协议形式化技术能够实现形式化的协议工程。

本文主要讨论基于 RSL 的协议形式化描述,并给出了 AB 协议的描述实例。

2 基于 RSL 的协议形式化描述

RAISE(Rigorous Approach to Industrial Software Engineering),即工业软件工程的严格方法,是一个1985年至1990年的 ESPRIT 项目,它是在一个光谱的规范语言 RSL 的基础上提供一系列工具和转换技术,形成了一种开发软件的严格方法。我们的研究要对 RSL 规范语言进行有效的扩充,从而形成基于 RSL 的协议形式化描述语言。

基于 RSL 形式化描述的协议规范由一组层次分明、结构清晰的模块组成,通过模块可以将协议规范分解为易于理解和可重用的单元。简单的模块定义有如下形式:

```
id =
  extend id1,…,idn with
  class
    declarations
  end
```

扩展子句(extend)用于刻画多个模块间的层次依赖关系,能够突出相关性并且屏蔽细节。

说明部分(declarations)可以定义为一个5元组 D=(T, V, C, W, A),其中:T 为一组类型说明,以 type 为起始关键字,主要有三种类型:固有类型,抽象类型和复合类型;V 为一组变量说明,以 variable 为起始关键字;C 为一组信道说明,以 channel 为起始关键字,in 是从信道中输入值,out 是向信道中输出值;W 为一组值的定义,每个定义都给出了值的名字及其类型,以 value 为起始关键字;A 为一组公理的说明,公理可以用来刻画值的名字所代表的属性,也可以用来表述约束规则,以 axiom 为起始关键字,每条公理的一般形式为右

*)国家自然科学基金重大研究计划项目(90104010),国家863计划项目(2001AA112062)。赵 静 硕士生,研究方向为软件工程,通信软件与协议工程。屈玉贵 副教授,研究方向为计算机体系结构,通信软件与协议工程。赵保华 教授,博导,研究方向为软件工程,通信软件与协议工程。

维普资讯 http://www.cqvip.com

部可带条件的恒等式。为了在证明中引用方便，公理可以被命名，其名字由方括号括起并置于公理的前面。

基于 RSL 的协议形式化描述可采用如下的算法：

(1)阅读协议文本，提出协议需要描述的功能。

(2)根据需要，可以将要描述的功能按一定的方式，如按层次、类别、阶段、运行方式等，划分成若干个模块。

(3)为了使描述较为完整，对于复杂协议，如有需要可为某些模块分别选择合适的模型作为基础。

(4)如有必要可适当增加一些辅助模块，用于说明全局函数，共享变量等。

(5)使用基于 RSL 的规范语言描述各个模块。

3 AB 协议的基于 RSL 的形式化描述

AB(Alternating Bit)协议是最早的端—端通讯协议之一，它的协议机制和功能的非形式化描述在此不再赘述，下面根据前述的算法对其进行基于 RSL 的形式化描述：

(1)协议需要描述的主要功能有：发方协议实体的功能，收方协议实体的功能以及系统的功能。

(2)将需要描述的功能划分为三个模块：SEND，RECEIVE 和 SYSTEM，其中 SEND 模块主要描述发方协议实体的内部行为，与上层发方用户的交互(通过抽象信道 cha)，与下层网络层的交互(通过抽象信道 ch)；RECEIVE 模块主要描述收方协议实体的内部行为，与上层收方用户的交互(通过抽象信道 chb)，与下层网络层的交互(通过抽象信道 ch)；SYSTEM 模块主要描述整个 AB 协议系统的行为和性质。

(3)从描述之方便和完整的角度出发，为这三个模块均选择了 CCS 模型。

(4)根据需要，增加了一个辅助模块 ADDITION 用于全局资源等的说明。

(5)以下为 AB 协议的基于 RSL 的形式化描述。

```
ADDITION=
  class
    type
      TA,TB,TC (抽象数据类型)
    variable
      s_seq: Nat:=0,(发端报文序列号)
      r_seq: Nat:=0,(收端报文序列号)
    channel
      cha:TA,
      chb:TB,
      ch :TC
    value
      increase : Nat→Nat (报文序列号增加函数)
    axiom forall x: Nat·
      increase(x) ≡ (x+1) mod 2
  end
  SEND =
  extend ADDITION with
  class
    value
      transa: TA → TC,(类型转换函数)
      overtime : Unit → Bool, (判断是否超时函数)
      seq :TC → Nat,(从报文中提取序列号函数)
      send : Unit → in cha,ch out ch Unit,(发送函数)
      csend : Unit → in ch out ch Unit (重发函数)
    axiom
      send() ≡ let udata:=cha? in ch!transa(udata) end;csend(),
(;表示顺序执行)
      csend() ≡ if overtime() then ch!transa(udata);csend()
        else
          let ack:=ch? in (?表示接受,!表示发送)
            if (seq(ack)=s_seq) then increase(s_seq);send()
end
          end
        end
end
RECEIVE =
  extend ADDITION with
  class
    value
      transb :TC → TB,(类型转换函数)
      ackout :Nat → TC,(发出确认报文函数)
      istrue :TC → Bool,(判断收到的报文是否正确的函数)
      receive :Unit → in ch out ch,chb Unit (接收函数)
    axiom
      receive() ≡ let m=ch? in
        if istrue(m) then
        ch!ackout(r_seq);chb!transb(m);increase(r_seq);receive
()
        else receive() end
      end
end
SYSTEM =
  extend SEND,RECEIVE with
  class
    value
      system :Unit → in cha,ch out ch,chb Unit
    axiom forall x:TC,i:Nat·
      system() ≡ send()‖receive(),(‖表示并行执行)
      [ch_ack]
      x:=ch?⫛ch!ackout(i) ≡ x:=ackout(i) (⫛为内锁连接符)
end
```

由上述可见，同其它协议形式化描述语言相比，该形式化描述结构清晰可读性好，并且公理部分(axiom)刻画协议性质的能力较强。以 SYSTEM 模块 axiom 部分的[ch_ack]公理为例，其中的'⫛'为内锁连接符，它不会遗漏任何可能的通信，此公理以简洁的形式表达了确认报文不会丢失、不会出错并且不会重复的协议机制。

如果将上述形式化描述中的各个模块逐步精化，将会使抽象级别较高的规范转化为抽象级别较低的规范，并最终可以转化为程序语言实现代码；另一方面，将各个模块的另一方面，将各模块的公理部分组织和利用起来，可有效地帮助进行协议验证和协议测试。因此，基于 RSL 的协议形式化描述技术，将会有力地促进协议工程形式化的发展。

结论 协议的形式化描述技术 FDT 是协议开发的基础，本文初步讨论了基于 RSL 的协议形式化描述，今后的工作要进一步扩充 RSL 规范语言，一方面使之更加适合于描述协议的各种性质，另一方面将协议的形式化描述与协议验证、协议实现和协议测试等综合起来考虑，较好地实现整个协议开发过程的形式化。

参 考 文 献

1 龚正虎. 计算机网络协议工程. 北京：国防科技大学出版社，1993

2 ISO. ESTELLE. A Formal Description Technique Based on an Extended State Transition Model[S]. DIS 9074,1987

3 ISO. LOTOS. A Formal Description Technique Based on Temporal Ordering Observational Behavior[S]. DIS 8807,1987

4 The RAISE Language Group. The RAISE Specification Language, BCS Practitioner Series. Prentice Hall,1992