协议工程概论与进展*)

肖军模　徐越彦　（南京通信工程学院）

摘要 The developmental process of the standard protocol descripted by the formal description techniques (FDT) and directed by the ideas of the software engineering, is known as the "protocol engineering". This is a developmental method of the protocol emerging in recent years. This paper puts emphasis on the basic ideas, history of the protocol engineering, and its main tasks in each phases. Finally, the advances for studying the protocol engineering in our institute is briefly presented.

一、什么是协议工程

在信息社会中，由于信息交换和处理的需要，建立各种类型的计算机网络（局部网、城域网和广域网）已势在必行。从目前的情况看来各厂家所提供的网络通常只包括下三层或下四层协议，运输层以上各层协议则需由用户自己开发。因此不论厂家还是用户都存在通信协议的开发问题。

协议开发的最终产品是可以在不同环境下运行的协议目标代码。早期的开发方法主要是依据协议的非形式描述用手工方式实现的。由于对非形式描述的协议可能存在不同的理解，很难保证协议产品的正确性及其与标准协议之间的一致性。

近几年来，由于协议形式描述技术的日趋成熟，人们用软件工程的思想指导协议的开发，就形成了所谓的协议工程。支持协议工程的一整套软件与工具称为协议开发系统（或环境）。

协议工程通常包含以下几个阶段：

- 协议形式描述，ISO已经提供；
- 协议正确性验证；

*）本课题得到国家自然科学基金资助。

两部分组成：Xgraph和mgraph，二者作为并发进程运行。Xgraph是一个用C书写的独立图形设施，用X窗口系统工作；mgraph是和符号系统有关的部分。SIG通过X实现显示设备的独立性，通过mgraph达到可移植性。该文介绍了SIG的功能、设计与实现、以及进一步的开发计划。

4. 意大利C.N.R. CNUCE研究所G. P. Facouti等人的"图形系统的交互模型"一文介绍一个宜于描述计算机图形系统基准模型（在ISO内部开发的）所定义的构架内部的交互式图形程序的模型。着重讨论其"陈述成份"，其中列出并讨论了来自各种不同方法的问题。随后介绍了由计算机图形基准模型所定义的系统结构。接着描述了作为逻辑设备或交互者内部一组进程的各个交互成份。

限于水平，欠妥之处，请批评指正。

• 协议性能预测分析；
• 协议代码自动生成；
• 协议一致性测试；

图1说明了协议工程各阶段之间的关系。

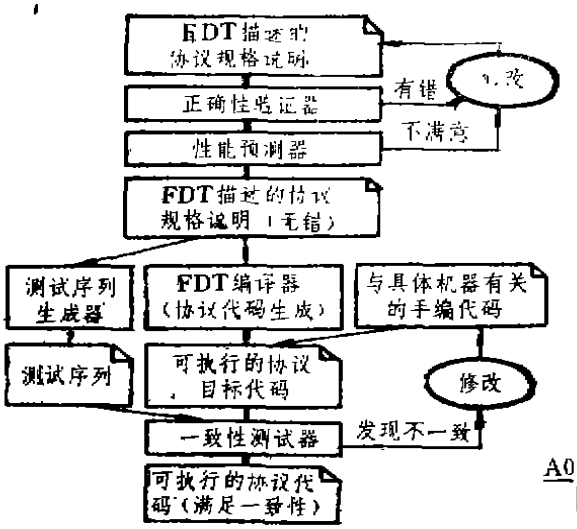


图1 协议工程的各个阶段

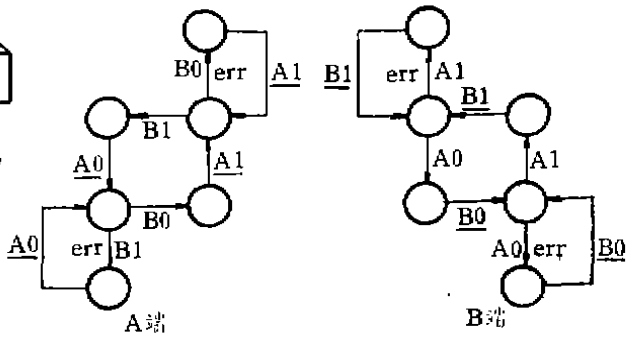
协议开发系统应有支持上述各阶段任务的软件工具。下面先分别介绍协议工程各阶段上的主要任务与方法，然后再介绍我院对协议工程的研究概况。

二、协议工程的形成

二十年前，美国学者W.C. Lynch在ACM通信杂志上发表了题为"在半双工电话线上实现可靠的全双工传输"的文章[1]，首先提出了一个后被广泛引用的简单协议——交替比特协议(Alternating Bit Protocol,简称为AB协议)。他认为在这种协议中必须使用两个比特位作为控制信息(一位验证位和一位交替变化位，该位用作帧编号)才能使AB协议正确运转起来。Lynch在该文最后提出一个猜想，认为这类协议至少需要两位控制信息才能正确运转。他希望能用形式化的方法来证明或否证他的猜想。

Lynch的文章引起了英国国家物理实验室(NPL) K.A. Bartlett等三人的注意。第二年，他们用有穷自动机的方法验证了只用一位控制信息的AB协议仍能正确地运行，否定了Lynch的猜想。图2是他们验证AB协议时所用的自动机。

尽管Lynch和Bartlett所讨论的协议是极其简单的，但却引起了人们对协议形式描述与验证的关注和研究，开始了研究协议形式描述技术的新纪元。在尔后的二十年间，协议的形式描述技术有了很大的发展。大致可以分为以下三个阶段：



A0表示发送A0； B0表示接收B0

图2 验证AB协议的有穷自动机

1. 研究各种形式描述阶段(1968～1979)：这个阶段里人们提出了各种描述协议的模型，主要有以下几种：

1)有穷自动机模型 在这种模型中，自动机的结点表示协议的状态；结点间的连弧表示协议的状态变化，称为协议状态变迁(transition)。协议系统被描述为一组互相通信的自动机。

2)Petri网模型 这也是一种状态变迁模型。协议系统中的一组条件表示为Petri网中的库所(Place)，系统中在某些条件下发生的事件(event)，例如发出一个连接请求信号，用变迁表示。整个协议系统可用一张Petri网描述。

3)形式语言模型 用形式语言的文法(正规文法、上下文无关文法等)描述协议的信息交换过程。在这种模型中，通信原语当作终结符，正确的原语序列作为协议"语言"中的句子。

4)高级语言模型 用现有的程序设计语言(Pascal, PL/1等)描述协议。然后再利用语言的编译器实现协议。

5)**混合模型** 用状态变迁模型（前两种）描述协议的优点是直观，便于协议验证。但协议复杂时，描述协议的图形也相当复杂，产生"状态空间"爆炸的问题。这种模型是面向验证的，不是面向实现的。高级语言模型则是面向实现的，但用这种模型验证协议则是比较困难的。混合模型结合了这两种模型的优点，协议主状态下的变迁仍用图形形式（有穷自动机或Petri网）描述，而协议的其它细节仍用高级语言描述，这样使得协议的描述与验证都得以简化。

此外还有的用时态逻辑和抽象代数语言描述协议。这些方法有益于协议的验证。

上述各种形式描述技术都得了实际应用。

2. 协议及其形式描述的标准化阶段（1979—1985）：人们在验证协议正确性的过程中发现协议实现中的错误主要是由于协议文本本身的描述（用自然语言形式）就存在着二义性或矛盾的地方。要使标准协议是精确的和无二义性的，必须对协议的描述手段进行标准化。这项工作是由ISO/TC97/SC 21完成的，在八十年代先后定义了ESTELLE和LOTOS两种形式描述技术[2,3]。

ESTELLE语言是在PASCAL语言的基础上增加了描述扩展有穷自动机（*EFSM*）的语言成分。EFSM体现了协议各主状态下的输入信息、输出信息以及变迁到下一状态的条件、优先级、时间控制等参数。ESTELLE语言是前面所说的混合模型的一种表示方法。

LOTOS语言[3]描述协议实体间交互时外观行为的时态次序。它用动作表达式描述协议实体的外观行为，动作表达式是由基本动作或动作项通过行为操作符（顺序、选择、并发等）连结而成。LOTOS语言是以CCS（通信系统语言）为基础的，其数据成分采用专用语言ACT ONE描述。

此外，CCITT Q39/Ⅶ研究组在80年代也提出了SDL语言。这个语言与ESTELLE类似，也采用扩展有穷自动机模型。它主要用于并发程控软件，但也用作协议形式描述技术。

由于CCITT和ISO这两个组织的巨大影响，它们提出的开放系统互连参考模型（OSI/RM）已得到了世界各大计算机与数据通信公司及研究单位的承认。从而使得OSI/RM的形式描述技术也得到了公认。ISO已经提供了用ESTELLE和LOTOS描述的运输层、会话层等层次的协议文本，为协议工程打下了基础。

3. 协议工程阶段（1985年以后）：在这一阶段中人们开始研究基于ISO或CCITT建议的形式描述技术（ESTELLE、LOTOS及其它）的协议开发方法。美国学者T. F. Piatkwski在1983年首先提出了"协议工程"的概念[4]。H. Rudin在1985年详细阐述了基于形式描述技术的协议工程的各阶段中的主要任务[5]（见图1）。ISO的ESTE-LLE和LOTOS两种语言已于1988年9月正式形成最后版本，协议工程已由研究阶段转到正式实现的阶段。

进入八十年代以后，世界的一些著名研究所与公司已经开发了基于各种形式描述技术的协议开发系统，比较成功的有以下几个系统：

a. 美国哥伦比亚大学研制的CUPID系统，支持有穷自动机、Petri网和高级语言等描述手段，可以实现协议的验证与协议目标代码的生成。

b. BBN/NBS系统采用的描述技术是基于PASCAL语言和扩展有穷自动机的。该系统可以生成描述协议实体的FSM表格，并验证协议的各种性质和生成部分协议代码。该系统的功能如图3所示。

c. Cadie系统是由瑞典UPPSALA大学研制的通信协议的交互设计环境。这个系统使用的描述技术是ASYL/EFSM语言。该语言也是基于PASCAL和有穷自动机模型。该系统可以从ASYL/EFSM的协议文本生成可执行的协议代码。图4给出了该系统的结构框图。

还有许多协议开发系统，这里不一一介绍了。这些协议开发系统的共同特点是它们

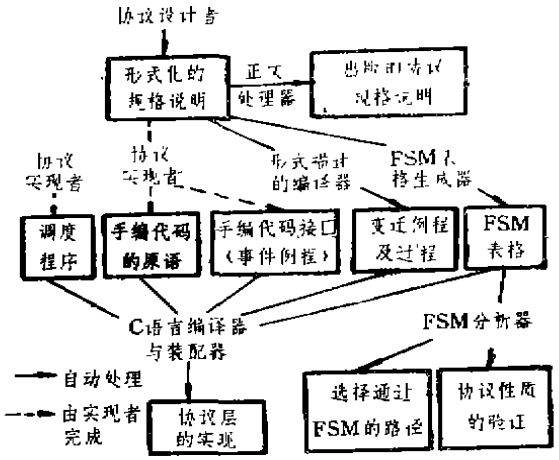


图3 BBN/NBS的协议开发系统

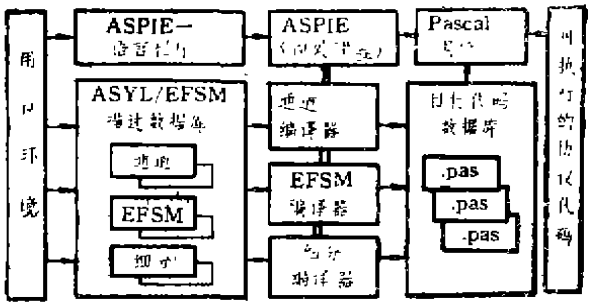


图4 Caddie的编译模型

使用的描述技术是自己定义的，不是标准化组织建议的。但它们和ISO的ESTELLE语言很类似，这些系统为研制基于ISO的形式描述技术的协议开发系统提供了宝贵的经验与方法。

三、协议工程各阶段的主要任务

1. 协议的规格说明

由于ISO已陆续提供用ESTELLE和LOTOS描述的OSI各层协议的标准文本，用户不再需要自己进行规格说明。但需要决定在协议开发系统中选择哪种描述技术。这两种描述技术的特点如下：

- ESTELLE语言是一种过程语言，可以描述协议的细节，用它描述的协议便于实现。
- LOTOS语言描述协议实体的外观行为，不关心实体内部变化。一般说来用LOTOS描述协议的抽象级别较高。该语言中还定义了一套证明行为表达式的公理化系统，因此用LOTOS描述的协议便于验证。

考虑到协议工程后续阶段的需要，协议开发系统最好能同时支持这两种描述技术。从ESTELLE的协议文本产生协议的目标代码；利用LOTOS的协议文本验证协议的正确性。当然需要保证这两种文本的一致性。下面还将看到对ESTELLE的协议文本也可以验证其正确性。

2. 协议的正确性验证

如果协议开发系统支持LOTOS语言，由于LOTOS中定义了对各种语句的验证方法，可以直接利用语言中的验证公理对协议文本进行验证。

如果验证ESTELLE的协议文本，可以采用以下方法：先从ESTELLE的协议文本中抽取协议的EFSM模型（这可以由编译程序完成）；然后利用该模型验证协议的正确性。用自动机验证协议是有算法可依的。还可以把抽取出来的EFSM转换成等价的Petri网模型，从这种模型可以验证协议的更多性质[6]。

3. 协议的性能预测

协议正确性验证可以验证协议的安全性（Safeness）和活性（Liveness）。安全性是指协议中不会发生坏事情（错误），活性是预言协议将会发生好事情（按预期的功能运行）。但正确性验证不能验证协议运行时的性能参数，如时延、吞吐量等。协议性能预测是对将要生成目标代码的协议作性能预测分析，如果发现协议不能满足性能要求，则需对协议文本作相应修改。

在静态下对协议预测分析的主要方法是在描述协议的EFSM模型中注入时延参数、流量参数和通过各条路径的概率参数。然后利用该模型仿真协议的运行，对各种性能指标作统计分析，在此基础上再预测协议的性能。

4. 协议的自动生成

在验证了协议的正确性和预测了协议有满意的性能之后，就可以生成协议的目标代码。下面说明 ESTELLE 编译器的构造方法。一种方法是直接为 ESTELLE 构造编译器，这种编译器翻译 ESTELLE 语言的每个成份，称为全局翻译法。另一种方法是先把 ESTELLE 中的非 PASCAL 成份翻译为相应的PASCAL语句，即先把 ESTELLE的协议文本翻译为 PASCAL 形式，再用PASCAL编译器生成协议的目标代码（参见图5）。这种方法也称为两次编译法或局部翻

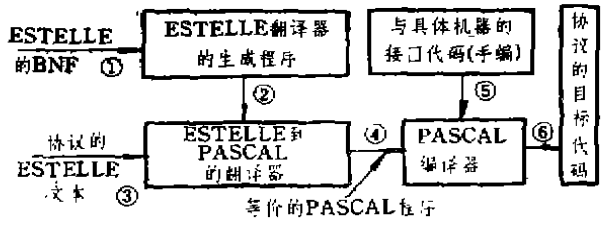


图5 ESTELLE的两次编译过程

译法。这种方法考虑了ESTELLE是在PASCAL基础上扩充EFSM的描述成份这一特点的。ESTELLE 编译器可以采用自动生成技术。一般说来，协议代码的自动生成通常只能生成协议中那些和机器无关的代码，与机器有关部分的代码还需靠手编。手编工作量约占40%左右。

5. 协议的一致性测试

虽然在协议生成之前验证了协议的正确性和预测了协议的性能，但是由于最终运行的目标机各具有自己的特性，必须手编一部分代码和具体操作系统功能相结合，这就很难保证目标协议和其形式描述之间的一致性。因此有必要对目标协议进行一致性测试。

ISO和CCITT两大标准化组织已共同制定了一致性测试的框架和方法学。把一致性测试分为基本互连测试、能力测试、行为测试和一致性解答测试四种类型，其中主要是行为测试。对测试序列的表示方法也作了建议，建议用树与表的结构（TTCN）表示测试序列。

为了提高一致性测试的自动化程度，需解决从协议的形式描述文本自动产生测试序列的方法。一种可行的办法是利用前面介绍的从抽取的EFSM中产生测试序列。由于测试序列是由被测协议的输入原语和响应原语组成的，还需对抽取的EFSM进行“粗化”处理，即把已细分的EFSM再合并为较大的自动机。只要它能反应协议对原语的反应即可。例如，当向某层协议输入一条请求连接原语时，只需给出该层协议的响应信息即可，并不需要知道协议内部是如何处理这些原语的细节。

以上五个阶段构成了完整的协议开发过程，即我们所说的协议工程，它是以ISO的形式描述技术为基础的。

四、 我们的工作进展

我们学院对协议工程的研究在全国是较早的单位之一。83、84年就注意这方面的发展动向，85年对研究生开设了“网络协议的形式描述技术”课程。徐越彦教授指导研究生研究ESTELLE的最早版本，并提出了一种实现方案。86年谢希仁教授指导研究生首先在IBM PC/XT上实现了ESTELLE语言的半自动生成系统。由于机型的限制，该系统未实现ESTELLE语言的并发功能。但该系统的研制为协议工程提供了实际经验。87年又申请了国家自然科学基金，资助协议工程的研究。前面介绍的协议工程各阶段的主要任务与方法，也就是我们要实现的协议工程的总体方案。该项目进展情况如下：

1.已经把原先在PC/XT机器上实现的ESTELLE生成系统移植到微VAX-Ⅱ机上，并利用VAX的操作系统VMS实现了ESTELLE的并发功能，从而实现了较完整的ESTELLE编译器。

2.已经完成了从协议的ESTELLE文本抽取EFSM的研究。由于协议文本中的EFSM含有变迁的控制变量（布尔表达式），对抽取的EFSM需要根据

形式化软件规范技术

刘少英*)（西安交通大学）

摘 要

形式化软件规范技术是增强软件开发过程的科学性和所开发软件的可靠性的一个有效途径。本文通过"教师职称提升系统"和"整数分类系统"实例较系统地介绍了目前国际上流行的两种形式化规范技术VDM和OBJ，并对它们各自的特点和性质进行了比较和分析。

一 引 言

软件开发可分为软件规范说明和软件实现两大部分。软件规范说明的任务是要告诉计算机"做什么"，而软件实现要解决的问题是"怎样做"。因此，软件规范说明在软件开发中是一个极为重要的阶段。如果该阶段所形成的软件规范不能正确地或不能确切地指出"做什么"，那么对后边的软件实现工作将会带来不可估量的灾难。为了解决好这个问题，人们已经提出了许多描述软件规范的方法和语言，如目前最为普遍应用的数据流图[1]，JSD[2]和自然语言描述[3]等，但这些

*) 系西安交通大学的国家公派留学生，在英国曼彻斯特大学计算机系攻读Ph. D.，现已两年多。

在其初态下允许的各变量的初值，驱动该EFSM运转，得到一个确定结构的有穷自动机。这一过程称为"展开EFSM"。展开后的EFSM就成为普通的有穷自动机。

3. 已经完成了从有穷自动机模型中产生测试序列的研究。可以从展开后的EFSM产生测试目标协议的一致性测试序列。主要方法是从自动机的初态出发遍历自动机，所遍历的各弧上的标记（协议原语名）就形成了测试序列。下一步工作就是根据对协议的一致性测试要求，构造实用的测试工具。

4. 在第(2)项工作的基础上，下面准备实现基于有穷自动机方法的协议验证工具。

从总体上看，我们目前进行的研究仍属于基础性研究，要达到实用阶段，还需作出更多的努力。

参 考 文 献

[1] W. C. Lynch, Reliable Full-Duplex Transmision over Helf Duplex Telephone Line, Communications of the ACM, vol. 11, No. 6, June 1968.

[2] ISO/TC97/SC21, ESTELLE——A Formal Description Technique Based on an Extended State Transition Model, ISO9074, 1988. 9.

[3] ISO/TC97/SC21, LOTOS——A Formal Description Technique Based on the Temporal Ordering Behavior, ISO 8807, 1988. 9.

[4] T. F. Piatkowski, Protocol Engineering, Proc. of ICC' 83, Boston, Mass, June 1983.

[5] H. Rudin, An Informal Overview of Formal Protocol Specification, IEEE Communication Magazine, vol. 23, No. 3, March 1985.

[6] 肖军模，一种有效的协议分析工具，（南京通信工程学院）军事通信学报，1987