Especificação, Verificação e Testes de Protocolos

Wanderley Lopes de Souza e Stefania Stiubiener

Nesta última década, a utilização e a concepção de técnicas de descrição formal (TDF) para a especificação de protocolos de comunicação, assim como a validação desses protocolos, têm merecido uma atenção especial por parte dos pesquisadores, dos fabricantes de computadores, dos setores de telecomunicações e dos órgãos internacionais de padronização, atuantes em teleinformática. Este artigo apresenta uma revisão das principais TDFs e das principais técnicas de validação, que têm sido empregadas durante o ciclo de desenvolvimento dos protocolos.

1. Introdução

Num sistema de comunicações, onde as interações entre as entidades comunicantes são realizadas através da troca de mensagens, denomina-se **protocolo de comunicação** o conjunto de regras que garante o intercâmbio ordenado dessas mensagens, e denomina-se especificação do protocolo a descrição dessas regras. O termo **validação** refere-se a qualquer atividade que vise aumentar o grau de confiabilidade da entidade **(especificação, projeto ou implementação)** que está sendo analisada.

A fim de facilitar a compreensão dos problemas envolvidos na definição dos protocolos, estabelecer padrões e facilitar o próprio desenvolvimento dos protocolos, especialistas da área equacionaram e estruturaram as funções desejadas, para a comunicação entre sistemas distribuídos, em sete camadas hierarquicamente organizadas, resultando o **Modelo Básico de Referência para Interconexão de Sistemas Abertos** (OSI, "Open System Interconnection")[**1**], esquematizado na **Fig. 1.**

W.L. de Souza é Professor do Departamento de Sistemas e Computação da Universidade Federal da Paraíba, Av. Aprígio Veloso, 882, CEP 58.100, Campina Grande, Pb.
S. Stiubiener é Professora do Departamento de Engenharia Elétrica da Escola Politécnica da USP, Av. Luciano Gualberto, 158 — Travessa 3, CEP 05508, São Paulo, SP.

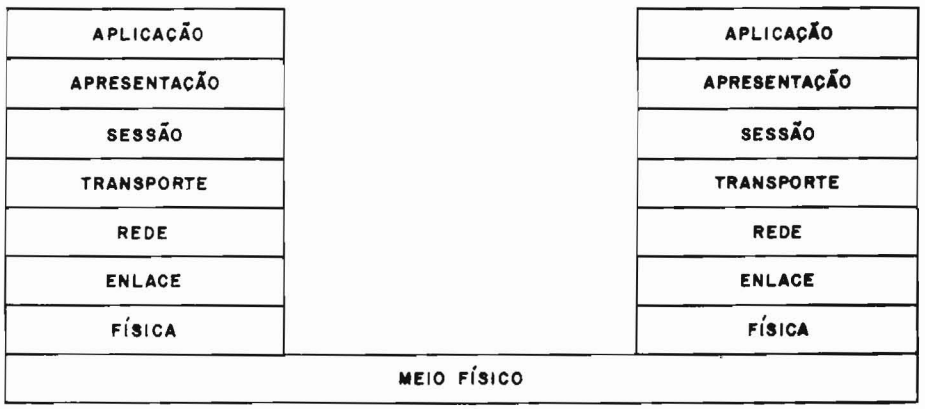

Figura 1. Modelo Básico de Referência OSI/ISO ("Open System Interconnection" / "International Standards Organization").

Na descrição de uma **rede de computadores,** cuja arquitetura obedece ao modelo acima representado, dois conceitos são fundamentais: o conceito de serviço e o conceito de protocolo (ver **Fig. 2**). Utilizando os serviços oferecidos pela camada (N-1), as entidades da camada (N) cooperam entre si, de acordo com o protoloco (N), para fornecer um serviço (N) com mais recursos à camada (N+1).

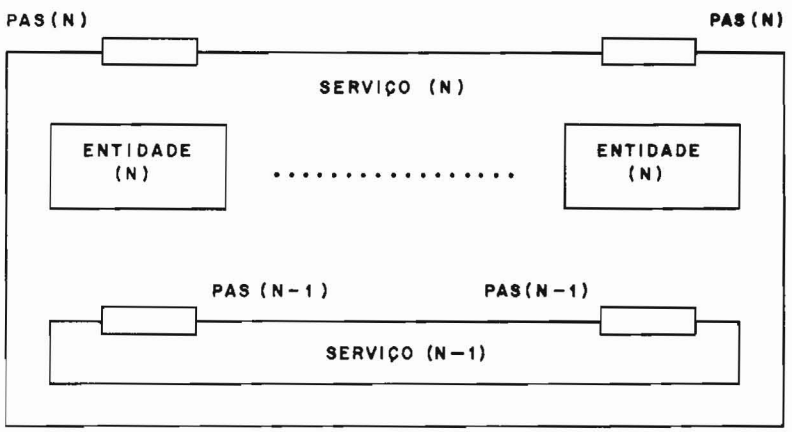

Figura 2. Noções de serviço e protocolo.

A nocão de serviço (N) é abstrata, pois condensa as camadas de 0 a N, omitindo o fluxo de dados entre elas. A especificação desse serviço é a descrição, por um observador externo, do comportamento de uma caixa preta, sujeita às trocas de primitivas de serviço com a camada superior, que são realizadas através dos pontos de acesso ao serviço (N), representados por PAS (N).

Na especificação do protocolo (N) é descrito o comportamento das entidades que se comunicam, se sincronizam e operam de forma concorrente via os pontos de acesso ao serviço (N-1)[**2**].

As descrições de serviços e protocolos têm sido realizadas associando-se uma linguagem natural a representações gráficas e/ou a tabelas de estados. Essa metodologia tem se revelado inadequada e insuficiente, já que essas especificações semi-formais são freqüentemente ambíguas, incompletas e inconsistentes, levando, muitas vezes, a implementações incompatíveis nas diferentes máquinas conectadas à rede. Além disso, tendo-se como referência esse tipo de especificação, as atividades de validação, que podem ser empregadas, são geralmente também semi formais (por exemplo, "walkthroughs"). Tais atividades de validação, por sua vez, têm se revelado incapazes de avaliar todos os possíveis comportamentos dos protocolos e conseqüentemente não podem outorgar um alto grau de confiabilidade à entidade analisada.

Durante o ciclo de desenvolvimento de um protocolo, o emprego de técnicas semi-formais, nas atividades de especificação e/ou validação, pode permitir que erros provenientes da especificação e/ou projeto, que poderiam ser detetados e corrigidos nessas primeiras fases do ciclo, proliferem por todas as implementações, tornando o trabalho de depuração de custo elevado e extremamente difícil. São os casos, entre outros, dos protocolos do CCITT X.21 e X.25, que foram desenvolvidos utilizando técnicas semi-formais e que possuem certos comportamentos inesperados e indesejáveis[**3**].

2. Especificação e Validação de Serviços e Protocolos

Os objetivos principais das especificações formais são: fornecer descrições claras e concisas do sistema que está sendo concebido e suportar uma análise, passo a passo, das diferentes etapas do ciclo de desenvolvimento desse sistema .

Para que os objetivos das especificações formais sejam atingidos, é necessário que a técnica de descrição formal utilizada seja: **expressiva,** isto é, suas construções devem fornecer meios para exprimir comunicação, sincronização e concorrência; **analítica,** ou seja, possua um modelo que permita a validação dos objetos que estão sendo especificados; e **abstrata,** no sentido de independência em relação aos métodos de implementação e no sentido de omissão, em qualquer etapa da especificação, dos detalhes irrelevantes.

Uma vez que a especificação formal de um sistema é a referência para os trabalhos subseqüentes, a validação dessa especificação torna-se fundamental. Na realidade, qualquer esforço posterior de desenvolvimento (e/ou validação) de projetos (e/ou implementações) pode ser desperdiçado, caso a referência seja inválida, ou incompleta, ou ambígua, ou ainda inconsistente com as intenções do especificador.

As atividades de validação podem ser grupadas segundo duas filosofias: verificação e teste.

Utilizando algum tipo de raciocínio lógico, a verificação analisa uma entidade, buscando provar se ela possui ou não determinadas propriedades que lhe são requeridas (por exemplo, ausência de impasses, ausência de mensagens não especificadas, etc).

O teste, freqüentemente aplicado a implementações, busca provar a conformidade de uma entidade em relação à sua especificação, através do funcionamento controlado dessa entidade em seu ambiente e através da observação do seu comportamento.

As propostas para o teste de especificações são mais recentes[**4**]. Num desenvolvimento onde a especificação final de uma entidade é atingida através de refinamentos sucessivos, o teste permite detetar inconsistências entre especificações de um mesmo sistema, realizadas em diferentes níveis de abstração. Obviamente, alguma forma de execução da entidade deve ser prevista, para que o teste possa ser aplicado. Em[**5**]é proposto que essa execução seja simulada e que o teste das especificações seja realizado através de observadores.

Em relação ao Modelo de Referência OSI, as técnicas formais de descrição e validação devem permitir (ver **Fig. 3**): as especificações dos serviços e protocolos das sete camadas OSI, de forma hierárquica, modular e intuitiva; a verificação das especificações; a validação do "design" do protocolo, isto

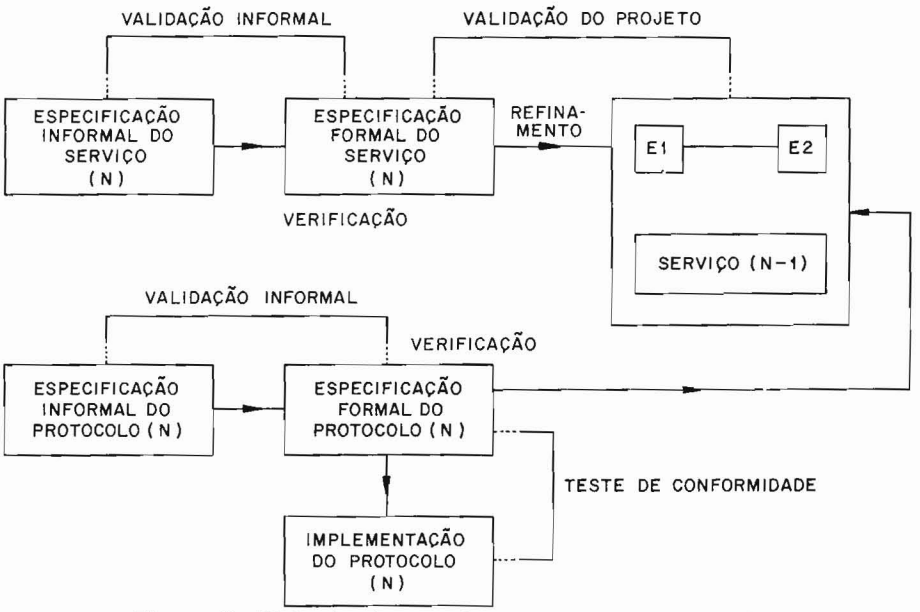

Figura 3. Ciclo de desenvolvimento de um protocolo.

é, verificar se o funcionamento conjunto de duas entidades (N) E_1 e E_2, que operam segundo a especificação do protocolo (N) e se comunicam através do serviço (N-1), satisfaz a especificação do serviço (N); o desenvolvimento de implementações a partir da especificação; e os testes de conformidade das implementações em relação à especificação.

3. Técnicas de Descrição Formal (TDF)

Visando encontrar um ponto de equilíbrio entre abstração, poder de expressão e poder de análise, características nem sempre compatíveis entre si, várias TDFs têm sido empregadas para a especificação de protocolos. Tais técnicas podem ser divididas em três grandes categorias: técnicas baseadas em modelos de transição; técnicas baseadas em linguagens de programação; e técnicas híbridas.

3.1. TDFs Baseadas em Modelos de Transição

A motivação para o uso de TDFs baseadas em modelos de transição para a especificação de protocolos é baseada no fato de que o comportamento de entidades comunicantes, que respondem a certos estímulos externos e internos, pode ser descrito através de máquinas de estados (autômatos).

Tais modelos são eficazes para a descrição dos aspectos de **controle dos protocolos,** durante as fases de inicialização, de estabelecimento e de encerramento da conexão. Para descrever aspectos relativos à **transferência de dados** (por exemplo, parâmetros que acompanham as mensagens, mensagens contendo texto, etc), estados adicionais são necessários (por exemplo, diferentes estados são necessários para representar cada possível número de seqüenciamento que pode acompanhar uma mensagem). Portanto, para protocolos complexos e/ou suposições realistas com respeito ao meio de comunicação (por exemplo, atraso de transmissão aleatório, perda, duplicação e distorção de mensagens, etc), o número excessivo de estados da máquina inviabiliza qualquer tipo de análise (**explosão de estados).**

Os principais modelos de transição, empregados por tais técnicas, são: **máquinas de estados finitas (MEF); gráficos;** e **linguagens formais.**

3.1.1. Modelos Baseados em Máquinas de Estados Finitas

Tais modelos incluem "Pure FSM"[**6**], "Duologue Matrix"[**7**], "Phase Diagram" [**8**],"Perturbation"[**9**] e "Colloquy" [**10**]. Nesses modelos, cada processo é representado por uma MEF. Os estados da MEF correspondem aos estados do processo e as transições entre os estados da MEF correspondem às transições que ocorrem no processo, devido à recepção e/ou transmissão de sinais. O acoplamento entre MEFs corresponde ao meio de transmissão subjacente e pode ser realizado através de filas, conectando as entradas

de uma MEF com às saídas da outra (e vice-versa), ou diretamente (quando o atraso de transmissão é irrelevante), especificando as transições das MEFs que devem ocorrer simultaneamente. O próprio meio de transmissão pode também ser modelado por uma outra MEF.

3.1.2 Modelos Gráficos

As Redes de Petri, linguagens de especificação caracterizadas por expressões gráficas, têm sido bastante utilizadas para descrever o controle de fluxo de informações entre processos concorrentes. Teoricamente, esse modelo é aplicável a uma variedade maior de protocolos do que os modelos MEF. Por exemplo, a especificação dos protocolos que permitem a qualquer mensagem ser transmitida um número arbitrário de vezes pode ser realizada através de Redes de Petri, já que elas podem acumular uma quantidade ilimitada de mensagens em seus nós. Tais protocolos, ainda teoricamente, não podem ser representados através de MEFs[**11**].

Uma rede de Petri é graficamente constituída de **nós**(ou lugares) que representam condições, de **barras** (ou transições) que representam eventos e de arcos direcionados, que podem conectar um nó a uma barra, especificando uma condição de entrada para a ocorrência de um evento, ou podem conectar uma barra a um nó, especificando uma condição de saída após a ocorrência de um evento.

Um estado da rede é representado por uma distribuição de **fichas** (chamada de **marcação**) através dos nós da rede. A partir de uma marcação inicial é possível determinar o conjunto de marcações permissíveis para a rede (e para essa marcação inicial). A qualquer instante a marcação de uma rede é afetada pela ocorrência de eventos. Para que um evento ocorra é necessário que todas as suas condições de entrada sejam satisfeitas (presença de pelo menos uma ficha em cada um dos nós de entrada). A ocorrência de um evento (disparo junto a uma barra) consiste na remoção de uma ficha de cada um dos nós de entrada (em relação à barra), conjuntamente com a adição de uma ficha em cada um dos nós de saída (em relação à essa mesma barra). É possível que, num determinado instante, vários eventos estejam habilitados (não determinismo).

A rede de Petri referente a um protocolo é a combinação das redes de Petri referentes aos processos comunicantes. Essa combinação é obtida, primeiramente, identificando-se, para cada rede, os nós de entrada e saída que devem se comunicar com os nós de saída e entrada, respectivamente, de outras redes envolvidas, e, posteriormente, conectando esses nós, de acordo com as suposições previamente estabelecidas para o meio de comunicação.

3.1.3. Modelos Baseados em Linguagens Formais

A inclusão de linguagens formais nos modelos baseados em transição deve-se

à correspondência existente entre máquinas de estados e **gramáticas formais.** O símbolo inicial da gramática formal corresponde ao estado inicial da máquina de estados, os símbolos não-terminais aos estados, os símbolos terminais ao conjunto de interações (entradas e/ou saídas) e as regras de produção às transições.

Utilizando esse modelo, um protocolo é descrito através de uma gramática formal, que é capaz de gerar todas as possíveis sentenças válidas, relativas à linguagem formal definida por essa gramática. Essa linguagem formal é baseada num alfabeto, que é composto de símbolos, que por sua vez representam as interações do sistema de comunicação. As sentenças, geradas pela gramática formal, descrevem as possíveis seqüências de interações entre os processos comunicantes desse sistema. Essa gramática formal é também chamada de **gramática de ação**[**12**].

Os diversos formatos e campos (freqüentemente complexos) das interações entre os processos comunicantes podem ser também representados nesse modelo, através da definição de uma **gramática de mensagens** para esses formatos e campos. É possível substituir as produções da gramática de mensagens pelos símbolos terminais da gramática de ação, para obter uma representação detalhada do protocolo que está sendo modelado.

3.2. TDFs Baseadas em Linguagens de Programação

Protocolos podem ser descritos, utilizando-se linguagens de programação de alto nível, através de procedimentos contendo processos concorrentes. O modelo básico é constituído de programas para cada processo comunicante. O meio de transmissão subjacente é especificado através de **asserções**, obtidas a partir de suposições feitas a respeito do seu comportamento. A comunicação entre os diversos módulos é realizada através de variáveis compartilhadas, ou através da troca de mensagens[**13**].

Entre as linguagens de programas de alto nível, o emprego de Pascal[**14**], como técnica de descrição formal para a especificação de protocolos, foi uma das primeiras propostas[**15**]. Mais recentemente, tem sido bastante considerado o uso de programação funcional[**16**]e da linguagem Lisp[**17**], o uso de programação lógica[**18**]e da linguagem Prolog[**19**], assim como o uso de programação concorrente[**20**] e da linguagem ADA[**21**].

A grande vantagem desse tipo de modelo, em relação aos modelos de transição, é a sua capacidade de representar certos aspectos relativos às estruturas de dados dos protocolos, presentes principalmente na fase de transmissão de dados. Por exemplo, o número de seqüenciamento, que acompanha cada uma das mensagens contendo dado, pode ser representado por uma simples variável.

Uma outra característica, mas desta vez polêmica, é o grau de abstração das linguagens de programação. Se de um lado, um alto nível de detalhamento

de uma especificação implica numa proximidade desta em relação às possíveis implementações, facilitando sobretudo a tarefa dos implementadores, por outro lado, isso implica também que essa especificação não será suficientemente abstrata e conterá detalhes relativos à fase de implementação, reduzindo o conjunto das possíveis implementações conformes a essa especificação. Em relação ao modelo OSI/ISO, um alto grau de abstração é uma característica necessária para as TDFs (ver Seção 2), já que um dos objetivos é a padronização das especificações formais dos protocolos desse modelo e, portanto, elas devem ser totalmente independentes das implementações.

Ainda dentro dessa categoria, algumas técnicas, empregando notações mais abstratas que as utilizadas pelas linguagens de programação, têm sido exploradas para a especificação de protocolos. Entre elas destacam-se: lógica temporal[**22**], tipos abstratos de dados[**23**], estados delta[**24**]e expressões seqüenciais[**25**].

3.3. TDFs Híbridas

As técnicas que se enquadram nas duas categorias anteriores são eficazes para descrever aspectos dos protocolos relativos ao controle (Modelos de Transição), ou relativos à transferência e estrutura dos dados (Linguagens de Programação), mas não ambos. Vários métodos combinando as características positivas dessas duas categorias têm sido propostos[**26**]-[**28**],[**2**],[**29**],[**30**].

Num modelo híbrido típico, uma máquina de estados é complementada com variáveis de contexto, representando estados adicionais, e procedimentos, que manipulam essas variáveis e influenciam na escolha da transição a ser efetuada, ambos descritos através de uma linguagem de programação de alto nível.

Os estados da máquina original (também denominados de principais) correspondem às subfunções dos protocolos (por exemplo, inicialização, estabelecimento de conexão, encerramento de conexão, transferência de dados, recuperação de erros, etc). Os procedimentos descrevem as ações a serem tomadas, em função das interações de entrada e dos valores das variáveis de contexto. Por exemplo, o valor do número de seqüenciamento de uma mensagem recebida pode ser comparado com o valor de uma variável de contexto, que indica o próximo número de seqüenciamento esperado, para determinar a aceitação ou não dessa mensagem e para determinar o próximo estado da máquina.

3.4. TDFs em Vias de Padronização

Como foi visto na Seção 2, as especificações formais dos serviços e protocolos constituem a referência de base para as outras atividades relacionadas ao ciclo de desenvolvimento de um protocolo de comunicação. Além disso, são os alicerces para a construção de ferramentas que visam a implementação automática, validação e teste dos protocolos.

Preocupados com os problemas oriundos das especificações informais e preocupados com a fixação de normas para o "correto" desenvolvimento dos protocolos, órgãos internacionais de padronização, entre os quais a "International Standards Organization (ISO)", o "Comité Consultatif International Télégraphique et Téléphonique (CCITT)" e o "National Bureau of Standards (NBS)", estão complementando as especificações informais existentes e estão desenvolvendo novos padrões, utilizando técnicas de descrição.

No CCITT, foi aproveitada a "Specification and Description Language (SDL)" (uma técnica originalmente empregada na descrição dos sistema de comutação), para a especificação dos protocolos de comunicação. Essa técnica foi inicialmente orientada para representação gráfica e o modelo utilizado baseia-se numa máquina de estados finita estendida (MEFE)[**31**].

Na ISO foi criado o grupo de trabalho ISO TC97/SC21/WG1 para desenvolver novas técnicas formais para a descrição dos sistemas OSI. Esse grupo foi dividido em três subgrupos: subgrupo A, que definiu os conceitos arquiteturais, que dariam suporte aos trabalhos dos demais subgrupos[**32**]; subgrupo B, que desenvolveu a "Extended State Transition Language (Estelle)"[**33**], uma técnica híbrida cujo modelo é baseado numa MEFE estendida pelas construções da linguagem de programação Pascal[**14**]; e subgrupo C, que desenvolveu a "Language of Temporal Ordering Specification (LOTOS)" [**34**], uma técnica baseada num formalismo que descreve sistemas comunicantes (CCS) [**35**]e complementada com uma linguagem que descreve tipos abstratos de dados (ACT ONE)[**36**].

As duas técnicas desenvolvidas junto à ISO, embora partam de princípios distintos, permitem a construção da especificação de um sistema através da sua divisão em diversos subsistemas (refinamentos sucessivos), para uma posterior aglutinação das pequenas especificações correspondentes às descrições dos subsistemas . Além disso, tais técnicas permitem a especificação de qualquer sistema distribuído, desde que a sua arquitetura seja definida segundo os conceitos estabelecidos pelo subgrupo A.
Os projetistas da linguagem Estelle preocuparam-se em desenvolver uma técnica simples (de fácil aprendizado), utilizando para isso a experiência adquirida no uso das técnicas baseadas em máquinas de estados, e que pudesse ser rapidamente colocada à disposição dos usuários. Como conseqüência, pode ser atualmente encontrada, no cenário internacional, uma variedade razoável de ferramentas (compiladores, simuladores, testadores, etc) que auxiliam o desenvolvimento dos protocolos. Em contrapartida, o grau de abstração da linguagem (inferior àquele da LOTOS) obriga a especificação de pequenos detalhes relativos à implementação.

Os projetistas da linguagem LOTOS preocuparam-se em desenvolver uma técnica que dispusesse de uma sólida base matemática e que utilizasse princípios que tornassem a linguagem abstrata no sentido mais amplo do termo (segundo as características apresentadas na Seção 2). Como conseqüência, um tempo maior para obtenção dos primeiros resultados foi necessá-

rio e somente agora é que começam a surgir as primeiras ferramentas para LOTOS. Embora o aprendizado de LOTOS seja mais demorado, já que os seus conceitos não são tão difundidos e utilizados quanto os conceitos de Estelle, a semântica desta linguagem, formalmente definida através de um conjunto de regras de inferência, aliada aos conceitos e teoremas que definem a equivalência de comportamento entre sistemas, permitem a verificação lógica das especificações realizadas com essa técnica.

4. Técnicas de Verificação Formal (TVF)

A escolha de uma técnica para a verificação de uma entidade normalmente depende da técnica que foi escolhida para a sua especificação formal. As técnicas de verificação também podem ser divididas em três grandes categorias: técnicas baseadas em **exploração de estados**; técnicas baseadas em **provas de programas**; e técnicas **híbridas.**

4.1. TVFs Baseadas em Exploração de Estados

As TVFs associadas às TDFs baseadas em modelos de transição centram-se em técnicas de exploração de estados ("reachability analysis")[**37**],[**38**].

Num sistema constituído por um conjunto de processos que se comunicam através de um meio, um estado do sistema, geralmente denominado global, é definido como sendo uma combinação dos estados dos processos comunicantes juntamente com o estado do meio de comunicação. O estado global inicial é uma combinação dos estados iniciais dos processos, supondo o meio de comunicação vazio (inexistência de mensagens em trânsito).

As TVFs que empregam exploração de estados procuram analisar todas as possíveis seqüências de interação que podem ocorrer entre os processos comunicantes de um sistema, através da geração exaustiva de todos os estados globais que podem ser atingidos a partir do estado global inicial. Desta forma, um diagrama de transição, tendo como raiz o estado global inicial, é gerado. Esse gráfico é denominado de árvore de alcançabilidade.

Tais técnicas são utilizadas, principalmente, para verificar certas propriedades globais dos protocolos[**3**], como, por exemplo, ausência de: estados de impasse ("deadlock") — estado global a partir do qual não é possível a transmissão de novas mensagens, sendo que o meio encontra-se vazio; recepções não especificadas — recepção de uma mensagem, num estado global, não prevista na especificação do protocolo; interações não executáveis — recepções e/ou transmissões que não podem ocorrer em condições normais de operação; estados ambíguos — um estado de um processo que pode coexistir com diferentes estados de outros processos, enquanto o meio está vazio; "overflow" nos canais — ocorre quando o número de mensagens num canal excede a sua capacidade pré-definida; "unbounded growth" — ocorre quando um dos processos transmite mensagens a uma taxa superior à taxa de recepção do processo destino; laços indesejáveis (ou "tempo-blocking") — laços no

diagrama de transição, em que processos podem permanecer em execução, sem que haja uma progressão efetiva do sistema. As TVFs baseadas em exploração de estados são também utilizadas para verificar se o protocolo possui os seguintes fatores de qualidade: vivacidade ("liveness") — existência de um estado fixo ("home state") que pode ser alcançado a partir de qualquer estado global, atingível, por sua vez, a partir do estado global inicial; estabilidade (ou "self-synchronization") — após a sua inicialização, a partir de qualquer estado, o sistema sempre retorna, após um número finito de transições, ao ciclo normal de operação; e término apropriado — o sistema termina no estado final previsto.

O maior inconveniente das TVFs que empregam exploração de estados é o crescimento exponencial da árvore de alcançabilidade em função da complexidade do protocolo. Esse fenômeno, conhecido como explosão de estados impossibilita a geração e a análise de todos os estados globais passíveis de serem atingidos a partir do estado global inicial. Alguns trabalhos, visando a geração automática e análise de árvores de alcançabilidade, estão documentados em [**27**], [**39**], [**8**] e [**9**].

Execução simbólica é uma técnica que busca reduzir o número de estados a serem explorados [**40**]. Nessa técnica, um gráfico de alcançabilidade é construído, sendo que cada nó corresponde a um estado global simbólico. Cada um desses estados representa um conjunto de estados globais e pode ser atingido a partir de um estado global simbólico inicial. Desta forma, os efeitos da explosão de estados, que também podem ocorrer na execução simbólica, são atenuados.

4.1.1. TVFs Empregadas em Redes de Petri

Essas técnicas são freqüentemente baseadas em análise de alcançabilidade. A partir de uma marcação inicial, todas as possíveis marcações da Rede de Petri composta (referente ao sistema especificado) são geradas. Essas marcações e o conjunto de transições entre elas definem uma máquina de estados, chamada de **máquina de fichas**, onde cada estado corresponde a uma marcação. Portanto, a verificação de determinadas propriedades (por exemplo, ausência de impasses, vivacidade, etc) é baseada no exame de cada estado da máquina [**41**].

Se o protocolo é complexo, os problemas para a análise da sua especificação formal, quer ela seja realizada através de Redes de Petri, que ela seja realizada através de um modelo MEF, são semelhantes. Para contornar tais problemas, várias extensões para as Redes de Petri foram propostas, entre elas as **Redes de Petri Temporizadas**. Nesse modelo, restrições "temporais" são impostas aos eventos, o que possibilita a análise da estabilidade do protocolo [**42**].

1.2. TVs Empregadas em Linguagens Formais

sas TVFs freqüentemente utilizam técnicas de "limpeza" da gramática que fine a linguagem formal. A partir de certas propriedades da gramática, s técnicas podem detetar laços nas seqüências de interação, impasses, mino não apropriado, etc. Caso a linguagem formal seja proveniente de a gramática regular, técnicas de exploração de estados podem ser aplica- s à máquina de estados correspondentes a essa gramática.

TVFs associadas às TDFs baseadas em linguagens formais também não) convenientes para verificar aspectos relativos à transferência de dados s protocolos. Nas gramáticas regulares, a parametrização das produções, avés de um número de seqüenciamento, leva a um rápido aumento do nero de produções (análogo à explosão de estados). Em [**12**]é proposto iso de gramáticas livres de contexto, ou ainda o uso de gramáticas sensíveis contexto, para contornar esse problema.

2. TVFs Baseadas em Provas de Programas

sas TVFs envolvem, numa primeira etapa, a formulação de asserções, e devem refletir determinadas propriedades explícitas (por exemplo, especi- ações do serviço) e/ou implícitas (propriedades globais), que são requeridas protocolo que está sendo analisado. Numa segunda etapa é provada correção desse protocolo através da verificação dessas asserções[**15**].

contrário das técnicas baseadas em exploração de estados, as TVFs seadas em provas de programas permitem a verificação de aspectos relati- s à transferência de dados dos protocolos. Entretanto, tais técnicas exigem a certa dose de criatividade e intuição para a formulação e prova das serções, o que impede a automatização completa de todo esse procedi- nto. Outro inconveniente é a escassez de linguagens expressivas, o que culta a especificação formal de asserções que cubram uma grande varieda- de tipos de dados e primitivas funcionais. Isso limita, sobretudo, o emprego provadores automáticos de teoremas para a verificação das asserções.

oricamente, a execução simbólica pode ser utilizada como uma técnica prova. Atrelando as asserções a certos pontos dos programas e associando as asserções a predicados, que representam decisões a respeito dos ximos "caminhos" a serem trilhados, é possível verificá-las através da cução simbólica desses programas[**40**].

3. TVFs Híbridas

a vez que essas TVFs devem verificar especificações que foram construí- utilizando-se TDFs híbridas, elas empregam naturalmente técnicas que enquadram nas duas categorias anteriores: análise de alcançabilidade

na máquina de estados, para a verificação das propriedades globais, e técnicas de prova nas variáveis de contexto e nos procedimentos, para verificar propriedades explícitas (por exemplo, entrega de mensagens na seqüência especificada pelo serviço) [**26**].

Num modelo típico, as asserções são formuladas em termos de predicados relativos aos estados globais do sistema, que refletem as propriedades requeridas ao protocolo. Essas asserções são em seguida analisadas, através de alguma forma de indução. Esse procedimento indutivo verifica se os predicados são asserções "invariantes". Por exemplo, para cada transição t, se o predicado p é válido antes da ocorrência de t, ele deve permanecer válido após sua execução. Isto é, o predicado p é uma invariante do sistema se ele é preservado pelas ações associadas a cada transição t.

5. Teste de Protocolos

O desenvolvimento da área de sistemas distribuídos estimulou a atividade de projeto e implementação de uma grande quantidade de protocolos de comunicação, a maioria deles situada em ambientes OSI; conseqüentemente, surgiu a necessidade de avaliação da qualidade de funcionamento desses produtos, através de programas de teste.

Os testes de protocolos objetivam determinar a correspondência entre a especificação de um protocolo e o funcionamento de sua implementação, abrangendo tanto os aspectos de conformidade (observados em ambiente de teste propriamente dito) como os aspectos de interoperabilidade (observados em ambiente de funcionamento real de sistemas abertos interconectados).

5.1. Teste de Conformidade

Os testes de conformidade têm a finalidade de determinar o grau de coincidência entre a especificação de um protocolo e o comportamento de sua implementação, sem levar em conta aspectos de desempenho, aumentando assim a probabilidade de que diferentes implementações sejam capazes de se comunicar. Isto é feito observando-se o comportamento de cada implementação em relação ao conjunto de testes [**43**].

Os resultados de um teste de conformidade representam não somente diagnósticos do tipo "certa implementação se comporta ou não de acordo com sua especificação formal", mas também implicam em fornecimento de listas de erros e análise de causas e soluções.

A capacidade de deteção de erros de um teste de protocolo depende da arquitetura do teste, da seqüência de testes e do projeto do testador [**44**], elementos que serão examinados adiante.

5.2. Teste de Interoperabilidade

A "conformidade" entre uma implementação e a sua especificação, constatada num teste de conformidade, não garante o funcionamento da entidade testada, no ambiente de comunicação com outras entidades idênticas e parceiras, também aceitas pelo teste de conformidade. Para que isso seja possível é necessário que sejam observados outros aspectos, tais como [**43**]: a coincidência da versão de padrão de protocolo utilizado em todas as implementações dos sistemas que se comunicam; a coincidência de escolha do conjunto de opções e alternativas oferecidas pelo padrão de protocolo, em relação a cada implementação; a adoção de valores comuns de parâmetros; esclarecimento das situações de ambigüidade do protocolo; e valores de temporização.

O teste de funcionamento da implementação de um protocolo, executado em ambiente de interconexão de sistemas abertos, levando em conta os aspectos acima mencionados, é denominado teste de interoperabilidade.

Observa-se que a metodologia de execução deste tipo de teste ainda não está elaborada, existindo a necessidade da definição dos procedimentos de medida da interoperabilidade [**45**].

5.3. Especificação Formal e Teste de Protocolos

Os resultados das pesquisas relacionadas a testes de protocolos conduziram à conclusão de que qualquer tipo de teste de um protocolo deve ser projetado de acordo com sua especificação formal. A principal vantagem dessa conclusão reside no fato de que a utilização da especificação formal permite a obtenção de uma "Seqüência de Execução de Testes", garantia de que o teste programado abrange todas as situações de funcionamento do protocolo.

De acordo com o método de especificação formal utilizado, foram desenvolvidas várias técnicas de testes de protocolos [**46**], conforme descrito a seguir.

(i) Testes baseados na especificação do protocolo em modelos de máquinas de estados finitos. Os métodos de seqüências de teste dentro dessa técnica são: método da **varredura de transições** ("transition tour") que permite que todas as transições de estado do protocolo possam ser percorridas pelo menos uma vez e o método de definição de um **conjunto característico.**

(ii) Testes baseados na especificação do protocolo em linguagem Estelle. Essa técnica apresenta a vantagem de permitir a observação do comportamento do protocolo em relação à mudança de variáveis e parâmetros específicos a diversas mensagens e não somente em relação às mensagens.

(iii) Testes baseados na especificação de protocolos em linguagem LOTOS.

(ii) Testes baseados em técnicas híbridas, ou seja, especificação informal em conjunto com modelos de máquinas de estados finitos.

5.4. Arquiteturas de Teste de Protocolos

Conforme mencionado, a escolha da arquitetura de teste é um fator fundamental, relacionado ao poder de deteção de erros durante a execução de um teste de protocolo.

O método genérico utilizado consiste em aplicar à Implementação sob Teste (IST) de um determinado protocolo, pertencente a um nível de atividade N do Modelo de Interconexão de Sistemas Abertos da ISO, mensagens do tipo daquelas provenientes do nível N-1, N+1 ou da entidade parceira do nível N e observar as respostas e o comportamento da IST.

Existem várias propostas de arquiteturas de testes genéricos, tais como a arquitetura local de teste ou a arquitetura remota de teste (tipo C, D e R) [**47**], de acordo com a posição relativa do gerador de seqüência de mensagens e o elemento julgado dos testes.

5.5. Padronização na Área de Teste de Protocolos

Nos últimos anos, as organizações que atuam na área de padronização em telemática elaboraram documentos preliminares a uma padronização na área de teste de protocolos. Neste sentido, a ISO divulgou em 1986 os documentos: "OSI Conformance Testing Methodology and Framework" [**48**]e "Directory Services — Overview of Concepts, Models and Services" [**49**]. Além disso, o CCITT divulgou em 1987 o documento: "Draft Recommendation X290 on Testing and Verification of Data Communication Protocols" [**43**].

6. Conclusão

Neste artigo, foram mostrados os conceitos de desenvolvimento de protocolos de comunicação de dados, de acordo com a padronização vigente, e mencionadas as etapas significativas desta atividade. Foram apresentadas as técnicas de descrição formal existentes, as de verificação e de testes de protocolos.

A área de especificação formal, verificação e teste de protocolos constitui hoje objeto de interesse de grupos de pesquisa e desenvolvimento, das organizações de usuários (COS — Cooperation of Open Systems e Grupo de Usuários MAP — Manufacturing Automation Protocol), e das organizações internacionais de padronização atuantes na área de telemática. Anualmente, reúnem-se elementos representativos desses segmentos num evento especí-

fico da área, organizado pelo IFIP—WG6.1. Trata-se do "International Symposium on Protocol Specification, Testing and Verification".

No Brasil, observa-se nesta área o crescimento do interesse dos grupos pertecentes às universidades (Grupo de Redes de Computadores da Universidade Federal da Paraíba, da Escola Politécnica da USP e da Universidade Federal do Rio de Janeiro), das indústrias e dos usuários.

Atualmente, a comunidade envolvida enfrenta o desafio da automação das atividades de desenvolvimento de protocolos, ou seja, da elaboração e utilização de ferramentas que possibilitem a verificação lógica do protocolo, a geração automática de implementações a partir da especificação formal e da geração automática de seqüências de testes, assuntos esses que constituem campo aberto de pesquisa a nível nacional e internacional.

Agradecimentos

Este trabalho foi realizado com o apoio do CNPq.

Referências

[1] "Information Processing Systems — Basic Reference Model for Open Systems Interconnection", ISO IS 7498, 1983.

[2] G.V. Bochmann, "A General Transition Model for Protocols and Communication Services", IEEE Transactions on Communications, vol. 28, nº 4, Abril 1980, pp. 643-650.

[3] G.V. Bochmann e C.A. Sunshine, "Formal Methods in Communication Protocol Design", IEEE Transactions on Communications, vol. 28, nº 4, Abril 1980, pp. 624-631.

[4] C. Jard e G.V. Bochmann, "An Approach to Testing Specifications", Anais do ACM SIGSOFT/SIGPLAN Software Engineering Symposium on High-Level Debugging, Pacific Grove, Estados Unidos, 1983, pp. 53-59.

[5] W. Lopes de Souza e E. Ferneda, "Aplicações do Compilador Estelle", Anais do 5º Simpósio Brasileiro sobre Redes de Computadores, São Paulo, SP, 1987, pp. 107-120.

[6] K.A. Bartlett, "A Note on Reliable Full-Duplex Transmission over Half-Duplex Links", CACM, vol. 12, nº 5, 1969, pp. 260-261.

[7] P. Zafiropulo, "Protocol Validation by Duologue Matrix Analysis", IEEE Transactions on Communications, vol. 26, nº 8, Agosto 1978, pp. 1187-1194.

[8] C.H. West, "An Automated Technique of Communications Protocol Validation", IEEE Transactions on Communications, vol. 26, nº 8, Agosto 1978, pp. 1271-1275.

[9] P. Zafiropulo, C.H. West, H. Rudin, D.D. Cowan e D. Brand, "Towards Analyzing and Synthesizing Protocols", IEEE Transactions on Communications, vol. 28, nº 4, Abril 1980, pp. 651-660.

[10] G. Lemoli, "A Theory of Colloquies", Anais do First European Workshop on Computer Networks, Arles, França, 1973, pp. 153-173.

[11] P.M. Merlin, "Specification and Validation of Protocols", IEEE Transactions on Communications, vol. 27, nº 11, Novembro 1979, pp. 1671-1680.

[12] A.Y. Teng e M.T. Liu, "A Formal Model for Automatic Implementation and Logical Validation of Network Communication Protocols", Anais do Computer Networking Symposium, NBS, 1978, pp. 114-123.

[13] N.V. Stenning, "A Data Transfer Protocol", Computer Networks, vol. 1, nº 2, 1976, pp. 99-110.

[14] "Programming Language Pascal", ISO IS 7185, 1ª edição, 1983.

[15] G.V. Bochmann, "Logical Verification and Implementation of Protocols", Anais do 4th Data Communications Symposium, Quebec, Canadá, 1975, pp. 8.15-8.20.

[16] S. Meira, "Applicative Specification of Computer Networks and Protocols", Anais do 4º Simpósio Brasileiro de Redes de Computadores, Recife, 1986, pp. 21-30.

[17] P.H. Winston e B.K.P. Horn, "Lisp", Addison-Wesley Publishing Company, 1981.

[18] W. Lopes de Souza, "Utilização de Prolog para a Especificação e Validação de Protocolos", Anais do V Congresso da Sociedade Brasileira de Computação e XI Conferência Latino-Americana de Informática, Porto Alegre, 1985, pp. 51-59.

[19] W.F. Clocksing e C.S. Mellish, "Programming in Prolog", Springer-Verlag, 1981.

[20] R. Castanet, A. Dupeux e P. Guitton, "ADA, A Well Suited Language for Specification and Implementation of Protocols", Protocol Specification, Testing, and Verification V, North-Holland, 1986, pp. 247-258.

[21] H. Ledgard, "ADA An Introduction — ADA Reference Manual", Springer-Verlag, 1981.

[22] L. Lamport, "Sometime is Sometimes not Never: On the Temporal Logic Program", Anais do 7th Annual ACM Symposium on Principles of Programming Languages, Las Vegas, Estados Unidos, 1980, pp. 174-184.

[23] D.H. Thompson, "Specification and Verification of Communication Protocols in AFFIRM Using State Transition Models", Information Sciences Institute, University of Southern California, Relatório Técnico RR 81-88, 1981.

[24] W.T. Overman, "Verification of Concurrent Systems: Function and Timing", Tese de Doutoramento, University of California Los Angeles, 1981.

[25] S. Schindler, "Algebraic and Model Specification Techniques", Anais de 13th Hawaiian International Conference on System Sciences, 1980.

[26] G.V. Bochmann e J. Gecsei, "A Unified Model for the Specification and Verification of Protocols", Anais do IFIP Congress, 1977, pp. 229-234.

[27] A.A.S. Dantine e J.J. Bremer, "Modeling and Verification of End-to-End Transport Protocols", Computer Networks, vol. 2, nº 4/5, 1978, pp. 381-395.

[28] C.A. Sunshine e Y.K. Dalal, "Connection Management in Transport Protocols", Computer Networks, vol. 2, nº 4/5, 1978, pp. 454-473.

[29] A.A.S. Dantine, "Protocol Representation with Finite State Models", IEEE Transactions on Communications, vol. 28, nº 4, Abril 1980, pp. 632-642.

[30] G.D. Schultz, D.B. Rose, C.H. West e J.P. Gray, "Executable Description and Validation of SNA", IEEE Transactions on Communications, vol. 28, nº 4, Abril 1980, pp. 661-667.

[31] "Specification and Description Language — SDL", Recomendação CCITT Z. 100, 1988.

[32] W. Lopes de Souza, "Utilização dos Conceitos de Módulo, Porta e Canal em Especificações Formais de Serviços, Protocolos e Interface de Comunicação", Anais do 3º Simpósio Brasileiro sobre Redes de Computadores, Rio de Janeiro, 1985, pp. 25.1-25.23.

[33] "Information Processing Systems — Open Systems Interconnection — Estelle — A Formal Description Technique Based on an Extended State Transition Model", ISO IS 9074, 1988.

[34] "Information Processing Systems — Open Systems Interconnection — Lotos — A Formal Description Technique Based on the Temporal Ordering of Observational Behaviour", ISO IS 8807, 1988.

[35] R. Milner, "A Calculus of Communicating Systems", G. Goos e J. Hartmanis (editores), Springer-Verlag, 1980.

[36] H. Ehrig e B. Mhar, "Fundamentals of Algebraic Specification i — Equations and Initial Semantics", Springer — Verlag, 1985.

[37] C.A. Sunshine, "Interprocess Communication Protocols for Computer Networks", Technical Report 105, Digital Systems Laboratory, Stanford University, 1975 (Tese de Doutoramento).

[38] G.V. Bochmann, "Finite State Description of Communication Protocols", Computer Networks, vol. 2, nº 4/5, 1978, pp. 361-372.

[39] H. Rudin, "Automated Protocol Validation: One Chain of Development", Computer Networks, vol. 2, nº 4/5, 1978, pp. 373-380.

[40] D. Brand e W.H. Joyner, "Verification of Protocols Using Symbolic Execution", Computer Networks, vol. 2, nº 4/5, 1978 pp. 351-360.

[41] P.M. Merlin, "A Methodology for the Design and Implementation of Communication Protocols", IEEE Transactions on Communications, vol. 24, nº 6, Junho 1976, pp. 614-621.

[42] P.M. Merlin e D.J. Farber, "Recoverability of Communication Protocols: Implementations of a Theoretical Study", IEEE Transactions on Communications, vol. 24, nº 9, Setembro 1976, pp. 1036-1043.

[43] "Draft Recommendation X.290 on Testing and Verification of Data Communication Protocols", CCITT, Grupo de Estudos VII, Contribuição 187, Julho 1987.

[44] R. Dissouli, "Étude des Méthodes de Test Pour Les Implementations de Protocols de Communications Basées sur les Specifications Formelles", Tese de Doutoramento, Universidade de Montreal, Dezembro 1986.

[45] I. Davidson, "OSI Protocol Testing at the Corporation for Open Systems", Seventh IFIP International Meeting on Protocol Specification, Testing and Verification, Zürich, Maio 1987.

[46] B. Sarikaya, "Protocol Testing: Architectures and Test Sequences", Technical Report, Concordia University, Montreal, Canadá.

[47] B. Sarikaya, "Recent Developments in Protocol Testing", 5º Simpósio Brasileiro de Redes de Computadores, São Paulo, 1987.

[48] "OSI Conformance Testing Methodology and Framework", ISO TC 97/SC21/DP9646/1, Setembro 1986.

[49] "Directory Services-Overview of Concepts, Models and Services", ISO TC 97/SC21/DP9594/1, Setembro 1986.

WANDERLEY LOPES DE SOUZA nasceu em São Paulo (SP) a 14.06.54. Graduou-se em Engenharia Elétrica, modalidade Comunicações, pela Faculdade de Engenharia de Campinas (FEC) da Universidade Estadual de Campinas (UNICAMP), Campinas (SP), em 1976. Obteve o "Diplôme d'Études Aprofondies (DEA)" e o título de "Docteur Ingénieur", modalidade "Traitement du Signal et Télécommunication", pela "Université de Montpellier II", Montpellier (França), respectivamente em 1977 e 1979. De 1980 a 1983, como professor do Departamento de Engenharia Elétrica (DEE) da Universidade Federal da Paraíba (UFPb), ensinou e desenvolveu trabalhos de pesquisa na área de Sistemas Digitais. Em 1984, como professor do Departamento de Sistemas e Computação (DSC) da UFPb, desenvolveu trabalhos na área de Redes de Computadores. De 1985 a 1986, como professor convidado do "Dép. d'Informatique et Recherche Opérationnelle (IRO)", da "Université de Montréal (UdeM)", Montreal (Canadá), desenvolveu pesquisas, a nível de pós-doutoramento, em Especificação e Validação de Protocolos de Comunicação. Atualmente o Dr. Wanderley Lopes de Souza é Professor Adjunto IV do DSC da UFPb e suas áreas de pesquisa incluem Projeto e Desenvolvimento de Sistemas Distribuídos, Técnicas de Descrição Formal, Especificação, Verificação e Teste de Protocolos de Comunicação.

STEFANIA STIUBIENER é formada em Engenharia Eletrônica pelo Instituto Politécnico de Bucareste (Romênia) e obteve o Mestrado (1974) e Doutorado (1981) em Engenharia Elétrica pela Escola Politécnica da Universidade de São Paulo. É professora do Departamento de Engenharia de Eletricidade da EPUSP desde 1971. Participou de vários projetos na área de Comunicação de Dados, através de convênios e projetos junto à Embratel, CPqD-Telebrás, Vasp, Centro Científico IBM Brasil. Desempenhou atividades na comunidade acadêmica, dentro da SBC e LARC. Em 1987 coordenou o "5.º Simpósio Brasileiro de Redes de Computadores", realizado em São Paulo. As suas áreas de interesse atualmente são Projetos de Sistemas de Comunicação de Dados, Especificação, Implementação, Verificação e Testes de Protocolos para Redes de Comunicação de Dados.