

新型网络体系结构描述语言及其在分布式流媒体服务规约中的应用

杨 鹏¹, 刘 业²

(1. 东南大学计算机科学与工程学院, 江苏南京 210096; 2. 东南大学计算机网络和信息集成教育部重点实验室, 江苏南京 210096)

摘 要: 首先针对传统网络形式化描述语言的局限性, 提出了一种新型网络体系结构描述语言 INSADL, 并详细介绍了其设计思想及主要特色. INSADL 吸收了软件工程领域近年来的主要研究成果和实践经验, 强调将服务和协议两个概念进行明确分离, 能有效支持对网络体系结构的形式化建模. 然后以分布式流媒体服务 (DSMS) 体系结构的形式化规约为例, 对 INSADL 的应用进行了探讨. 通过对 DSMS 体系结构中的实体构件、交互连接件、体系结构配置等的规约和分析, 验证了 INSADL 的形式化建模和体系结构分析能力.

关键词: 网络体系结构; 体系结构描述语言; INSADL; 流媒体服务

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2007) 01-0165-05

A Novel Network Architecture Description Language and Its Application to Specify Distributed Streaming Media Service

YANG Peng¹, LIU Ye²

(1. School of Computer Science and Engineering, Southeast University, Nanjing, Jiangsu 210096, China; 2. Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, Nanjing, Jiangsu 210096, China)

Abstract: Aiming at some limitations of traditional network formal description languages, a novel network architecture description language, named INSADL, is proposed and its design philosophy and major characteristics are introduced. By absorbing the main research results and practical experiences of software engineering, INSADL emphasizes on explicitly separating service from protocol that can effectively support the formal modeling of network architecture. As an example of INSADL's application, how to specify the architecture of distributed streaming media service (DSMS) in INSADL is also introduced and discussed. By describing and analyzing the entity components, interaction connectors and architectural configuration of DSMS architecture, the formal description and architectural analysis capabilities of INSADL are demonstrated and validated.

Key words: network architecture; architecture description language; INSADL; streaming media service

1 引言

形式化描述语言在计算机科学及相关领域均有广泛应用, 它是对系统进行形式化建模的主要工具和载体. 尽管在传统的网络研究中, 人们对形式化描述语言给予了相当的重视, 并提出了 ESTELLE、LOTOS、SDL 等形式化描述语言^[1]. 但总的来说它们多是在上世纪七、八十年代开始制定, 都比较偏重于网络协议的形式化描述、验证、实现和测试等, 却较少关注以整个网络体系结构为描述对象的形式化建模研究, 并且也缺乏有效的网络体系结构建模手段. 因此, 这类主要面向网络协议设计的传统形式化描述语言, 难以胜任对网络体系结构整体的建模需求. 而在与计算机网络研究有密切关系的软件领域, 近年来一个重要研究成果即软件体系结构, 它以构件 (component) 和连接件 (connector) 作为基本建模实体, 将软件系统分解成构件及构件间的交互关系^[2], 为在较高抽象层次上

研究大型软件系统的全局组织和总体结构等提供了有力支持. 关于体系结构描述方法的研究亦是软件体系结构的主要研究内容之一, 当前一般采用体系结构描述语言 ADL^[3]来对软件体系结构进行描述. 国际上较有代表性的 ADL 主要有 UniCon、C2、Rapid、Darwin、ACME、Wright 等, 它们都各有其特色和相应的适用领域.

考虑到 ADL 既能抽象描述体系结构, 又能为形式化推导、分析、验证体系结构的有关性质提供支持, 因此我们在新一代网络体系结构的研究中, 借鉴和吸收了软件体系结构、构件化软件^[4]等的研究思路和方法, 从网络系统分布、并发、交互等本质特征出发, 提出了一种适宜于对面向服务的新一代网络体系结构^[5]进行形式化建模的新型网络体系结构描述语言 INSADL^[6], 并且将 INSADL 在实际网络系统的体系结构建模、描述和分析等方面进行了应用, 验证了 INSADL 的实用性和有效性. 本文一方面介绍了 INSADL 的主要设计思想、目标

收稿日期: 2004-12-22; 修回日期: 2006-11-01

基金项目: 国家自然科学基金 (No. 90604003); 国家 973 重点基础研究发展规划 (No. 2003CB314801)

及特色,另一方面结合我们研究实现的分布式流媒体服务 DSMS,介绍了在 DSMS 的设计和开发过程中,对其体系结构的 INSADL 规约和分析等。

2 INSADL 的设计思想、目标及特色

网络体系结构与软件体系结构虽有许多共性,但也存在许多差异。例如,近年来软件体系结构虽对网络软件系统有一定研究,但基本上都限于体系结构风格(architectural style)这一论题,并且多是简单地用“layered system”来概括,而少有专门深入研究;但网络体系结构决不是仅仅通过软件体系结构中所说的分层风格就能完全表示。因此,作为一种主要针对新一代网络体系结构的形式化描述语言,INSADL 一方面应该借鉴 ADL 的相关研究成果,另一方面又必须从新一代网络体系结构自身的特点出发。

文献[3]提出了一个对软件体系结构建模具有普适意义的 ADL 分类比较框架,INSADL 在设计过程中参照了这一框架,并且还注意吸纳了若干代表 ADL 的精髓,尤其是关于 Wright 的研究。事实上,Wright 虽是一种软件体系结构描述语言,但它所采用的描述软件交互的思想,正是借鉴了传统网络研究领域消息通信的机制^[2]。同时,INSADL 建立在新一代网络体系结构 INSA 参考模型^[5]的研究基础之上,而后的设计初衷就是要顺应下一代网络研究重点正在向服务跃迁的崭新趋势,通过借鉴和吸收软件工程等相关领域近年来的重要研究成果,建立面向服务的新一代网络体系结构。在设计过程中,INSADL 始终以新一代网络体系结构 INSA 参考模型自身的特点作为研究的出发点,并且以 INSA 构件化思想和交互观点作为整个建模的基础。概括起来,INSADL 设计时所考虑的主要目标包括:

能够有效反映计算机网络系统的分布、并发、交互^[7]等本质特性;

既能够刻画网络系统中各种静态特征,又能够刻画各种复杂的动态行为;

支持对 INSA 参考模型中各种建模元素进行形式化描述,并能将各种建模元素装配成实际的网络系统;

既具有网络体系结构的描述能力,又具有一定的推理、分析和验证能力,并适于开发相应的 CASE 工具;

方法本身不局限于 INSA 参考模型,还应支持对其他各种现存网络体系的体系结构进行建模和分析等。

围绕上述目标,INSADL 选取 CSP^[8]为建模理论基础,并采用类 EBNF 对实体构件、端口、交互连接件、角色、网络体系结构配置等建模元素进行了严格定义^[6],这些建模元素及其相互关系如图 1 所示。为支持网络体系结构的复用、配置和演化,INSADL 还明确区分了类型和实例两个概念。相比 ESTELLE、LOTOS、SDL 等传统的网络形式化描述语言而言,INSADL 的特色之一表现在它反映了近年来软件领域内关于构件化软件的研究进展及成果,采用实体构件来对网络各层中各种特定功能实体进行建模抽象。INSADL 的特色之二表现在它支持对网络体系结构组成元素间交互关系的显式建模,这主要通过 INSADL 中的交互连接件来实现。INSADL 严格区分

了实体构件间的水平协议交互和垂直服务交互,这实际上是对协议和服务两个概念从网络体系结构建模方式上的明确分离,这对于解决传统 TCP/IP 参考模型及其实现中模块化程度不高、信息隐藏性不强、服务定制困难等问题有积极的方法论意义。而且,INSADL 中的交互连接件除支持“一对一”的两角色交互关系建模外,还支持“多对多”的多角色交互关系建模,所以无论是传统网络应用中的 C/S 交互方式,还是 P2P 等新兴应用中的对等交互方式,均可用 INSADL 中的交互连接件进行恰当地建模。

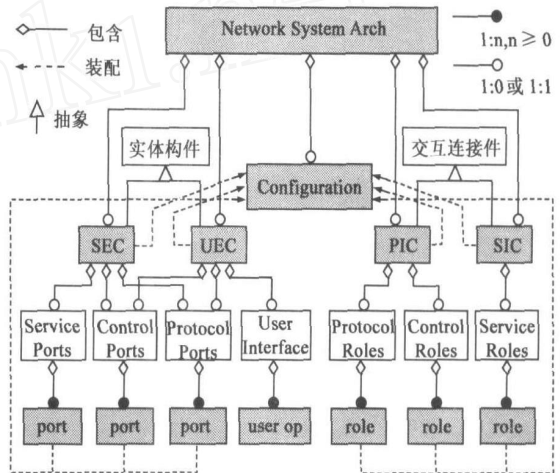


图 1 INSADL 的主要建模元素及其相互关系

INSADL 的特色之三表现在它支持对网络体系结构配置的规约,网络体系结构配置反映的是现实网络系统中实体构件实例和交互连接件实例之间的拓扑结构和装配约束。在 INSADL 中,交互连接件的每个角色描述了参与交互的实体构件的相应端口在该交互中的行为特征,一个实体构件实例能与一个交互连接件实例进行装配连接的前提是:实体构件实例的端口必须与交互连接件实例的相应角色之间满足如下的相容关系。

定义 1:称端口 p 与角色 r 是相容的,并记为 $Compatible(p, r)$,如果满足下述条件:

$$\begin{aligned}
 & (Type(p, inMsg) = Type(r, inMsg)) \quad (Type(p, outMsg) = \\
 & Type(r, outMsg)) \quad ((r, roleProperties \supseteq p, portProperties)) \\
 & ((r, roleConstrains = p, portConstrains) \quad (r, roleCon- \\
 & strains \Rightarrow p, portConstrains)); \\
 & ((p, portSpec) = (r, roleSpec)) \quad (failures(r, roleSpec) \\
 & \subseteq failures(p, portSpec)) \quad (divergences(r, roleSpec) \subseteq di- \\
 & vergences(p, portSpec)).
 \end{aligned}$$

上面 $Type(m)$ 表示消息 m 的类型, $failures(P)$ 和 $divergences(P)$ 表示进程 P 的失败集和分歧集, $failures(P) = \{(s, X) \mid (s \text{ traces}(P)) \quad (X \text{ refusals}(P/s))\}$, $divergences(P) = \{s \mid (s \text{ traces}(P)) \quad ((P/s) = CHAOS_p)\}$ 。采用 INSADL 对网络体系的体系结构进行形式化规约,实际上提高了网络体系结构的抽象级别,从而为基于 INSADL 的体系结构分析与检查奠定基础。INSADL 已被实际应用于对各种网络体系的体系结构进行建模、描述和分析,本文接下来将以分布式流媒体服务

DSMS 体系结构的 INSADL 规约为例介绍这方面的研究工作.

3 分布式流媒体服务体系结构的 INSADL 规约

当前,视频会议、在线点播等分布式流媒体应用业已成为网络中的主要应用模式.我们依据 OMG 音/视频流规范^[9]研究并实现了分布式流媒体服务 DSMS(Distributed Streamingmedia Service),图 2 是基于 DSMS 创建一个简单分布式流媒体应用的示例.在 DSMS 的设计和开发过程中,我们采用 INSADL 对其体系结构进行了建模、描述和分析.这样做的主要原因在于,虽然 OMG 音/视频流规范使用 IDL 对 MMDevice、VDev、StreamEndPoint 和 StreamCtrl 等关键部件的接口进行了定义,但由于 IDL 仅是一种数据类型描述语言,而且也不支持对网络体系结构的建模,所以这种定义存在如下不足:

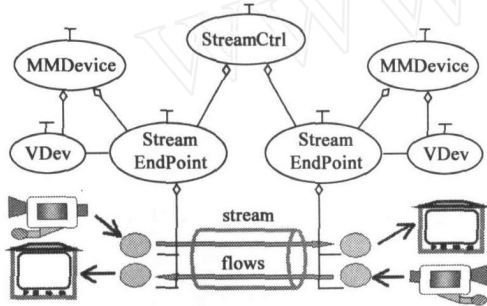


图 2 分布式流媒体服务的应用示例

只能静态描述分布对象的接口定义,而难以刻画分布对象之间复杂的动态交互语义;

不支持对连接件的显式建模,交互逻辑被隐藏于具体实现中,不利于提高系统的适应性和重用性;

难以对各种约束信息、非功能属性以及分布式流媒体服务的体系结构配置等进行描述;

简单的接口信息和大量采用的“自然语言+框图”描述,既不能有效支持基于体系结构的开发方法^[10],也难以对若干重要体系结构性质进行分析和验证等.

INSADL 的优势和特色能有效弥补上述不足.为此,先采用 INSADL 对 DSMS 体系结构中各实体构件类型和交互连接件类型进行规约,主体部分如下(考虑篇幅,进行了适当简化,并略去了端口、角色、属性等定义细节):

```

... .. // 类型及常量定义
ServiceEntityComponent MMDev . SEC
{ ServicePorts{ CreateParty(in StrmCtrl . UEC ,streamQoS ,
  flowSpec inmx ,out VDev . SEC ,StrmEndPoint . SEC outmx)
  { This = down ? inmx up ! outmx }; };
  ControlPorts{ SetVdev(out MMDev . SEC ,StrmEndPoint . SEC ,
  streamQoS ,flowSpec outmy) { This = down ! outmy }; };
  SetSep (out MMDev . SEC ,VDev . SEC ,streamQoS ,
  flowSpec outnz) { This = down ! outnz }; };
  Properties{ Available . Formats :TypeDef Sequence String
  AvFormats = "MIME:audio/wav","MIME:video/mpeg2"; }; };
}; // MMDev . SEC

ServiceEntityComponent StrmEndPoint . SEC

```

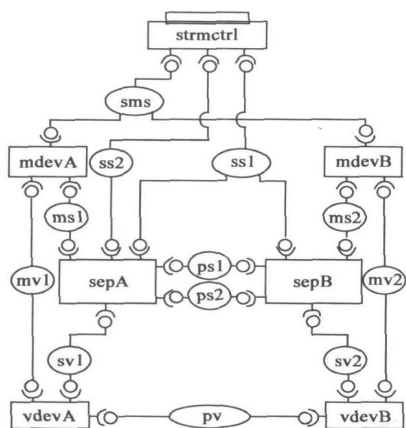
```

{ ServicePorts{ ... // 服务端口 SepParamsSet ,ConctPeerSep ,SepCtrl };
  ProtocolPorts{ ... // 协议端口 PeerConnection ,PeerCtrl };
  ControlPorts{ ... // 控制端口 SepCtrlVdev };
  Properties{ ... // 属性 Flows . Contained }; }; // StrmEndPoint . SEC
ServiceEntityComponent StrmEndPointA . SEC extends
StrmEndPoint . SEC{ ... }; // StrmEndPointA . SEC
ServiceEntityComponent StrmEndPointB . SEC extends
StrmEndPoint . SEC{ ... }; // StrmEndPointB . SEC
UserEntityComponent StrmCtrl . UEC
{ ControlPorts{ ... // 控制端口 MdevsCtrl ,StrmEstablish ,SepCtrl };
  Properties{ ... // 属性 Strm . Status 等 }; }; // StrmCtrl . UEC
ServiceEntityComponent VDev . SEC
{ ServicePorts{ ... // 服务端口 VdevParamsSet ,MediaCtrl };
  ProtocolPorts{ ... // 协议端口 FlowsTrans };
  Properties{ ... // 属性 Flows ,VDev . QoS 等 }; }; // VDev . SEC
ServiceInteractionConnector Mdev . Vdev . SIC
{ ControlRoles{ VdevSetter(out MMDev . SEC ,StrmEndPoint . SEC ,
  streamQoS ,flowSpec mout) { This = down ! mout }; };
  ServiceRoles{ VdevSetee(in MMDev . SEC ,StrmEndPoint . SEC ,
  streamQoS ,flowSpec min) { This = down ? min }; };
  ServiceQue{ This = VdevSetter . down ? mout
  VdevSetee . down ! mout }; }; // Mdev . Vdev . SIC
ServiceInteractionConnector Mdev . Sep . SIC
{ ... // 控制角色 SepSetter、服务角色 SepSetee 等
  ServiceQue{ ... }; }; // Mdev . Sep . SIC
ServiceInteractionConnector Sctrl . Mdevs . SIC
{ ... // 控制角色 SctrlBindRole、服务角色 MdevARole ,MdevBRole
  ServiceQue{ ... }; }; // Sctrl . Mdevs . SIC
ServiceInteractionConnector Sep . Vdev . SIC
{ ... // 控制角色 VdevCtrlr、服务角色 VdevCtrlrlee
  ServiceQue{ ... }; }; // Sep . Vdev . SIC
ServiceInteractionConnector SctrlConctSeps . SIC
{ ... // 控制角色 SctrlConctRole、服务角色 SepARole ,SepBRole
  ServiceQue{ ... }; }; // SctrlConctSeps . SIC
ProtocolInteractionConnector PeerSepsConct . PIC
{ ProtocolRoles{ ... // 协议角色 SepA ,SepB };
  ProtocolQue{ ... }; }; // PeerSepsConct . PIC
ProtocolInteractionConnector PeerSepsCoord . PIC
{ ProtocolRoles{ ... // 协议角色 SepBCaller ,SepBCallee };
  ProtocolQue{ ... }; }; // PeerSepsCoord . PIC
ProtocolInteractionConnector PeerVdevs . PIC
{ ProtocolRoles{ ... // 协议角色 VdevARole ,VdevBRole };
  ProtocolQue{ ... }; }; // PeerVdevs . PIC

```

图 3 DSMS 体系结构中实体构件类型和交互连接件类型的 INSADL 规约

接下来对 DSMS 的体系结构配置进行 INSADL 规约和检查.其中 DSMS 体系结构配置规约的主要任务是用适宜的交互连接件实例,将发生交互的实体构件实例连接装配成一个协调、统一的现实系统.这一步非常重要,因为其合理与否是决定整个系统能否正常运行的关键,并且也是进一步对 DSMS



```

NetworkSystemArch DSMS_Architecture //DSMS 体系结构的 INSADL 规约
{
... //前面关于实体构件类型和交互连接件类型的 INSADL 定义
Configuration | //DSMS 体系结构的配置规约
Instances | strmctrl: StrmCtrl _ UEC; mdevA, mdevB: MDev _ SEC;
sepA: StrmEndPointA _ SEC; sepB: StrmEndPointB _ SEC;
vdevA, vdevB: VDev _ SEC; sms: Sctrl _ Mdevs _ SIC;
mv1, mv2: Mdev _ Vdev _ SIC; ms1, ms2: Mdev _ Sep _ SIC;
sv1, sv2: Sep _ Vdev _ SIC; ss1: SctrlConctSeps _ SIC;
ss2: SctrlCtrlSepA _ SIC; ... | //Instances
Attachments | strmctrl. MdevsCtrl to sms. SctrlBindRole;
strmctrl. StrmEstablish to ss1. SctrlConctRole;
sepA. ConctPeerSep to ss1. SepARole;
sepB. ConctPeerSep to ss1. SepBRole;
strmctrl. SepCtrl to ss2. SepACtrlr; ... | //Attachments
| //Configuration
| //DSMS_Architecture
}
    
```

图 4 DSMS 的体系结构配置图及其对应的 INSADL 规约

图 4 DSMS 的体系结构配置图及其对应的 INSADL 规约

体系结构的性质进行分析和验证的基础。图 4 给出的是 DSMS 的体系结构配置图及其对应的 INSADL 规约。

按照上述分布式流媒体服务 DSMS 的体系结构规约, 可以进行基于 INSADL 的体系结构分析与检查, 主要包括对实体构件规约、交互连接件规约和体系结构配置规约等的分析和检查。其中的一个重点内容是端口-角色相容性检查, 具体实施过程按照定义 1 展开。据定义 1 可以验证, 上述 DSMS 体

系结构的 INSADL 规约中连接装配在一起的各实体构件实例的端口与对应交互连接件实例的角色满足相容关系, 这表明上述体系结构规约不存在配置冲突。进一步地, 还可对采用上述体系结构设计的分布式流媒体服务的系统性质(如死锁等)进行分析。例如, 按照 OMG 音/视频流规范的定义, Point-to-Point 流媒体应用的流建立过程可用图 5 中的 CSP 进程 PtoP-StrmEstablish 表示。

```

PtoPStrmEstablish = strmctrl. MdevsCtrl. down ! x
((mdevA. CreateParty. down ? x mdevA. SetVdev. down ! v1
vdevA. VdevParamsSet. down ? v1 mdevA. SetSep. down ! s1
sepA. SepParamsSet. down ? s1 mdevA. CreateParty. up ! y1)
(mdevB. CreateParty. down ? x mdevB. SetVdev. down ! v2
vdevB. VdevParamsSet. down ? v2 mdevB. SetSep. down ! s2
sepB. SepParamsSet. down ? s2 mdevB. CreateParty. up ! y2))
strmctrl. MdevsCtrl. up ? Concat (y1, y2)
(strmctrl. StrmEstablish. down ! z1 (sepA. ConctPeerSep. down ? z1
sepA. PeerConnection. outdir ! f sepA. PeerConnection. indir ? fb) |
strmctrl. StrmEstablish. down ! z2 (sepB. ConctPeerSep. down ? z2
sepB. PeerConnection. outdir ! f sepB. PeerConnection. indir ? fa))
    
```

消息	消息格式
f, fa, fb	streamQoS_flowSpec
x	sctrl, streamQoS_flowSpec
y1	vdevA, sepA
y2	vdevB, sepB
v1	mdevA, sepA, streamQoS_flowSpec
v2	mdevB, sepB, streamQoS_flowSpec
s1	mdevA, vdevA, streamQoS_flowSpec
s2	mdevB, vdevB, streamQoS_flowSpec
z1	sepB, streamQoS_flowSpec
z2	sepA, streamQoS_flowSpec

图 5 Point-to-Point 流媒体应用中流建立过程 CSP 表示

进程 PtoPStrmEstablish 描述的流建立过程与 DSMS 体系结构的 INSADL 规约一致, 并且依据 CSP 理论还可验证: PtoP-StrmEstablish satr ((traces (PtoPStrmEstablish)) ((tr, refusals (PtoPStrmEstablish/ tr)) failures (PtoPStrmEstablish)) (last (tr) =)), 所以进程 PtoPStrmEstablish 具有无死锁 (deadlock free) 的性质。此外, 由于 INSADL 中的交互连接件支持多角色交互, 因而对于 OMG 音/视频流规范中的 Multipoint 流媒体应用, 只需在前面规约的基础上进行适当扩展即可得到相应体系结构的 INSADL 规约。

自 OMG 音/视频流规范颁布以来已有多种实现, 包括 TAO A/V 流服务^[11]、ESP^[12]、OCTOPUS^[13]等。然而, 在这些系统的设计和实现过程中, 对于其体系结构的描述基本上都采用的是“自然语言+框图”的非形式化方法, 这不仅难以消除描述中的不一致性和歧义性, 而且也难以在设计阶段准确获知系统的性质和确保设计的正确性。本文采用 INSADL 对

DSMS 体系结构进行规约的过程, 实际上也是对 DSMS 的体系结构进行设计、检查和分析的过程, 其好处在于, 可以在系统的设计阶段进行系统配置检查和性质分析, 从而便于开发人员清晰理解系统的组成部分和各部分的功能, 并进而深刻把握系统中各种复杂的交互逻辑, 这对于检查和修正设计中的错误、验证和确保系统的期望性质, 以及指导基于体系结构的系统开发与实现等有重要支持意义。

4 结束语

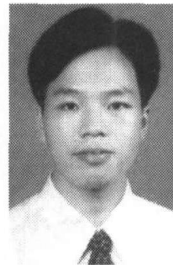
本文针对目前网络领域内所使用的形式化描述语言在网络体系结构建模方面的局限性, 通过吸收软件工程领域近年来的若干重要研究成果, 尤其借鉴了软件体系结构和 ADL 语言的研究思路, 以构件化思想和交互观点为建模基础, 强调将服务和协议两个概念明确分离, 立足于网络系统分布、并发、交互等本质特征, 提出了一种新型网络体系结构描述语言 IN-

SADL,并详细介绍了 INSADL 的主要设计思想、目标及特色。以 INSADL 为基础,结合我们研究实现的分布式流媒体服务 DSMS,介绍了在 DSMS 的设计和开发过程中对其体系结构的 INSADL 规约和分析等,实际验证了 INSADL 的形式化描述和建模分析能力,同时也表明本文所提出的新型网络体系结构描述语言 INSADL 具有较好的理论和应用价值。

参考文献:

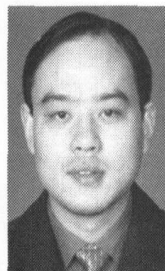
- [1] K J Turner. Using Formal Description Techniques-An Introduction to Estelle LOTOS and SDL[M]. New York:Wihey,1993.
- [2] R Allen,D Garland. A formal basis for architectural connection [J]. ACM Trans on Software Engineering and Methodology, 1997,6(3):213-249.
- [3] N Medvidovic,R N Taylor. A classification and comparison framework for software architecture description languages [J]. IEEE Trans on Software Engineering,2000,26(1):70-93.
- [4] C Szyperski,D Gruntz,S Murer. Component Software:Beyond Object-Oriented Programming,2nd Edition[M]. Boston:Addison-Wesley,2002.
- [5] 杨鹏,吴家皋. 基于交互、面向服务的新一代网络体系结构研究[J]. 电子学报,2005,33(5):804-809.
YANG Peng,WU Jia-gao. Interaction based,service oriented model of network architecture [J]. Acta Electronica Sinica, 2005,33(5):804-809. (in Chinese)
- [6] 杨鹏,刘业. 一种基于交互的网络服务体系结构描述语言 [J]. 东南大学学报,2006,36(5):685-689.
YANG Peng,LIU Ye. Interaction-based network service architecture description language [J]. Journal of Southeast University,2006,36(5):685-689. (in Chinese)
- [7] R Milner. Elements of interaction [J]. Communications of the ACM,1993,36(1):78-89.
- [8] C A R Hoare. Communicating Sequential Processes [M]. New Jersey:Prentice-Hall,1985.
- [9] OMG. Audio/video stream specification,V1.0[OL]. <http://www.omg.org/docs/formal/00-01-03.ps>,2000.
- [10] L Bass,R Kazman. Architecture-based development:technical report CMU/SEI-99-TR-007[R]. Pennsylvania,Pittsburghers:Carnegie Mellon University,1999.
- [11] S Mungee,N Surendran,D C Schmidt. The design and performance of a CORBA audio/video streaming service [A]. In Proc. HICSS-32[C]. Hawaii,1999.
- [12] P H Loosemore,E Sellin,S P Rana. Multimedia stream binding for a pan-European services platform [J]. BT Technology Journal,1999,17(2):103-114.
- [13] C H Zhang,H K Pung. The design and implementation of a Jini/Java-based A/V stream control and management [J]. Multimedia Systems,2003,9(4):315-326.

作者简介:



杨 鹏 男,1975年9月出生于四川南充,2006年4月毕业于东南大学计算机科学与工程学院,获工学博士学位,现为东南大学计算机科学与工程学院讲师,主要从事新一代网络体系结构、分布式计算、形式化理论和技术等方面的研究工作。

E-mail:pengyang@seu.edu.cn



刘 业 男,1977年1月出生于江苏建湖,2006年6月毕业于东南大学计算机科学与工程学院,获工学博士学位,目前在上海贝尔阿尔卡特公司从事3G网络产品的研发工作,研究兴趣主要包括高性能网络、对等网络计算等。