

# 语义 Web 服务组合的形式化描述与验证研究<sup>\*</sup>)

程永上 王志坚

(河海大学计算机与信息工程学院 南京 210098)

**摘要** Web 服务组合是 Web 服务的主要研究方向之一,对语义 Web 服务组合进行形式化描述并进行分析和验证是组合服务正确运行的保证。本文首先给出了基于有色 Petri 网的语义 Web 服务形式化模型,形式化描述了 OWL-S 的 ServiceModel 中 8 种基本的 Web 服务组合模式,利用这些组合模式可以构造出满足需求的组合 Web 服务。然后,对服务组合形式化模型的正确性分析和验证进行研究后,给出了组合模型语法正确性、可达性、活性和有界性的算法。最后,通过一个具体的建模实例展示了 Web 服务组合的 Petri 网建模。

**关键词** Web 服务, Web 服务组合, Petri 网, 建模, 验证

## Research on Formal Description and Verification of Semantic Web Service Composition

CHENG Yong-Shang WANG Zhi-Jian

(Computer and Information Engineering College, Hohai University, Nanjing 210098)

**Abstract** Composing Web service is one of the main research directions. Formally describing semantic Web service composition, analyzing and verifying the composition model can guarantee the correctness of the Web service composition. In this paper, firstly, the formal model of the semantic Web service composition based on colored Petri nets is presented, and 8 kinds of basic Web service composition patterns in ServiceModel of OWL-S are provided. Utilizing these patterns can construct the composite Web service of satisfying users' requirements. Then, after the correctness analysis and verification about the Web service composition model are studied, the algorithms on Web service composition syntax correctness, reachability, liveness and boundedness are given. Finally, modeling the Web service composition based on colored Petri nets is applied into a case.

**Keywords** Web service, Web service composition, Petri nets, Model, Verification

## 1 引言

Web 服务提供了一种平台独立、自描述、松耦合、位置透明的软件模块,便于实现不同应用的交互和集成。而语义 Web 的出现给 Web 服务技术带来了新的契机,两者的结合能够实现 Web 服务的机器可理解和处理,使 Web 服务发现、组合、运行和监督等自动化成为了可能。由于单个 Web 服务功能有限,不能满足日趋复杂的实际应用需求,必须将多个可用的自治服务协调工作,产生一个新的增殖的功能更强的复合服务,Web 的潜能才能被激发出来。如今,对于语义 Web 服务组合的研究已成为一个热点,而以透明方式实现服务组合的自动化也成为一挑战。

一般而言,用于组合的各个服务均来自分布于 Web 上的不同组织或者机构的原子服务,使用者很难掌控,设计有误的组合服务一旦付诸实施,造成的损失将会比较大,且随着业务需求的不断增强,Web 服务组合日趋复杂,服务组合出错的可能性也在增大,这些错误将导致资源和时间上的巨大浪费。为了在组合服务执行前发现错误并改正错误,避免执行时因错误而造成重大的损失,必须在组合规划时对组合服务进行正确性验证。达到这种目标的一种途径就是建立 Web 服务组合的形式化描述模型,利用对形式化模型的分析、验证方法来检验 Web 服务组合的正确性。

这里主要有两种类型的组合形式化建模方式:进程代

数<sup>[1]</sup>和 Petri 网<sup>[2]</sup>。进程代数的起源可以追溯到 20 世纪 70 年代,它是一类用代数方法研究通信并发系统的理论的泛称,包括通信系统演算 CCS<sup>[3]</sup>、通信顺序进程 CSP<sup>[4]</sup>、通信进程代数 ACP<sup>[5]</sup>、Pi-演算<sup>[6]</sup>等;而 Petri 网来源于 1962 年德国 Petri 教授的博士论文, Petri 网包括传统经典 Petri 网、高级 Petri 网和扩展 Petri 网等。由于 Petri 网不仅可以刻画系统的结构,而且可以描述系统的动态行为;不仅具有直观的图形表示,而且能够引入许多数学方法对其性质进行分析,并进行正确性验证,因此本文采用 Petri 网来对语义 Web 服务进行形式化描述和分析验证。

本文第 2 节介绍了与 Web 服务组合、建模、分析和验证相关的研究工作;第 3 节给出了语义 Web 服务的形式化描述模型,并对 8 种组合模式进行了基于有色 Petri 网的模型描述;在第 4 节中,分别给出了服务组合模型的语法正确性、可达性、活性和有界性算法,以此来验证组合模型的正确性;第 5 节将 Web 服务组合建模方法应用于实例;最后,对本文进行了总结并指出了下一步工作。

## 2 相关工作

Web 服务组合的相关技术已经成为学术界和工业界的研究重点,出现了许多与此相关的研究成果和文献。

文[7,8]分别建立一个基于 Petri 网和有色 Petri 网的代数模型,将原子服务描述成网模型,任何可以描述成 Petri 网

<sup>\*</sup> 基金资助:国家自然科学基金(60573098),973 项目(2002CB312002),江苏省自然科学基金(BK2006168)。程永上 博士生,副教授,主要研究方向为分布式计算,Web 服务组合等;王志坚 博士,教授,博导,主要研究方向为软件复用,分布式计算等。

模型的 Web 服务,都可以利用该代数运算将原子服务组合成复合 Web 服务。由于 Petri 网具有坚实的数学基础,因此可以定性分析和验证组合服务的正确性。但是这两种方法都没有涉及语义 Web 服务的表示,也没有给出组合服务的正确性验证方法。文[9]把 Web 服务描述成面向消息的基于行为 Petri 网模型,该模型包括消息域和服务过程,既能完成服务间的通信,又给出了服务行为的过程描述,在组合规则的基础上,实现 Web 服务的自动组合。这种方法没有明确描述语义 Web 服务。文[10]将情景演算作为中间语言,对 Web 服务的描述从 DAML-S 转换到 Petri 网,在定义了顺序、选择等 6 种基本组合方式后,由可用 Web 服务的交互关系产生所有可能的组合网,然后在组合网中找出满足目标需求的一种组合,最后利用 Petri 网性质对组合服务进行了分析和验证。这种方法不能精确表达 Web 服务的不同类型数据。

### 3 语义 Web 服务的形式化描述

一般 Petri 网没有数据和分层概念,因此不能精确表达 Web 服务的语义,也不适合刻画大型的系统。有色 Petri 网<sup>[11]</sup>结合了 Petri 网和高级语言的优点,可对复杂系统进行图形化的概念建模,对系统行为进行动态模拟和形式化推理,弥补了一般 Petri 网的不足。所以本文基于有色 Petri 网来表示语义 Web 服务的不同类型数据,使用不同颜色的 token 以及 token 数的变化来表示 OWL-S<sup>[12]</sup>模型中的 IOPE,以此来描述 Web 服务的语义。

对语义 Web 服务的组合问题而言,重要的是如何根据各个不同的服务的过程模型将它们组合在一起,因此,首先要能够准确描述各个过程。Petri 网作为形式化的描述和分析工具,特别适合描述并发和分布式系统,能够对系统进行定性和定量分析。原子过程的操作语义可以利用 Petri 网清晰地描述,原子过程执行的输入 Input 和前置条件 Precondition,由 Petri 网模型前置库中所不同种类 token 表示,输出 Output 由后置库中的 token 表示,原子过程的执行效果 Effect 通过进程发生前后其前置库所和后置库所中的 token 数量的变化表现出来,原子过程的 Petri 网如图 1 所示。



图 1 原子过程模型

语义 Web 服务的组合更需要对组合过程操作进行语义的清晰描述。一个 OWL-S 的组合过程中各子过程是基于控制结构的组合,子过程可以是原子过程或组合过程。原子过程作为组合过程的一个步骤,执行时的数据流将得到明确的描述,组合过程通过各种控制结构规定各子过程之间的数据和功能依赖关系。OWL-S 的 ComposedOf 属性指出了子过程的控制流和数据流,在子过程的有序和有条件执行上进行了约束。

下面依次给出 OWL-S 中的组合过程中的各种控制结构及其对应的 Petri 网语义。

• Sequence 结构。子过程按照顺序,依次执行,如图 2 所示。

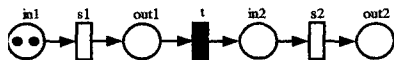


图 2 Sequence 结构

• Split 结构。一组子过程被并发执行,所有子过程被预置安排执行,如图 3 所示。

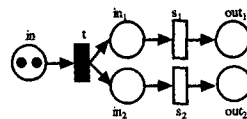


图 3 Split 结构

• Split-join 结构。一组子过程并发执行,当所有子过程执行后,Split-join 结构完成,如图 4 所示。

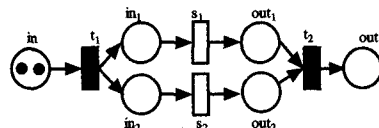


图 4 Split-join 结构

• Choice 结构。任意选择一个子过程执行,任一子过程执行完成都可以说明 Choice 结构完成,如图 5 所示。

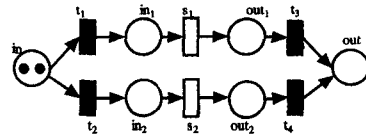


图 5 Choice 结构

• If-then-else 结构。根据 ifcondition 的取值的真或假决定选择哪个子过程执行。 $s_1$  或  $s_2$  有一个完成,则 if-then-else 结构完成。为了表示 ifcondition 的取值对  $s_1$  或  $s_2$  的影响,增加防卫函数来决定分支的选择,如图 6 所示。

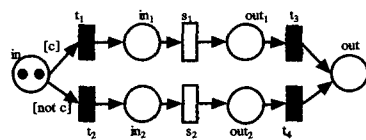


图 6 If-then-else 结构

• Any-order 结构。子过程可以按照任意过程发生,但不能同时发生,所有子过程都被完成,则 any-order 结构完成,如图 7 所示。

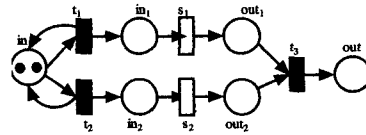


图 7 Any-order 结构

• Repeat-while 结构。首先对条件进行判断,条件为真,执行子过程,直到条件为假,否则退出,增加防卫函数来决定选择分支,如图 8 所示。

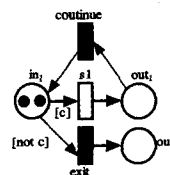


图 8 Repeat-while 结构

• Repeat-until 结构。先执行子过程一次,然后对条件进行判断,条件为假,执行子过程,直到条件为真,增加防卫函数来决定选择分支,如图 9 所示。

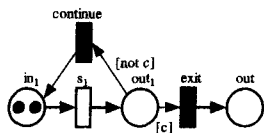


图 9 Repeat-until 结构

通过这 8 种控制结构,就可以将原子过程以及组合过程按照一定的要求组合成新的组合过程。可以用子网变迁表示嵌套的组合过程,子网变迁有自己的内部结构、内部行为和内部状态,这样分层表示结构适合对大系统建模。

#### 4 组合 Web 服务模型的验证

Petri 网具有坚实的数学基础,利用相应的性质和工具可以对组合 Web 服务模型进行分析和验证。

##### 4.1 组合 Web 服务语法正确性验证

利用输入矩阵和输出矩阵来验证 Petri 网模型的语法正确性,利用输出矩阵来判断入口库所设置是否正确,利用输入矩阵验证输出库所设置是否正确,利用输入和输出矩阵来验证是否存在孤立库所和变迁,是否存在不正确的弧。

$A^- [i][j]$  和  $A^+ [i][j]$  是 Petri 网的输入和输出矩阵; $i$  是库所数量,矩阵的行由库所组成; $j$  是变迁数量,矩阵的列由变迁组成; $p_i$  是一个库所; $t_j$  是一个变迁; $A^- [i][j] = 1(0)$  表示有(无)一条从  $p_i$  到  $t_j$  的弧; $A^+ [j][i] = 1(0)$  表示有(无)一条从  $t_j$  到  $p_i$  的弧。

##### Algorithm:

```

1: Void SyntaxCheck (WS_Model)
2: { For (i=0; i<=m; i++)
3:   If ! C(pi) ⊆ Σ
4:     Printf(" pi has Data Type fault");
5: For (i=0; i<=m; i++)
6:   { For (j=0; j<=n; j++)
7:     if A+ [j][i]=0
8:       Outf=true;
9:     For (j=0; j<=n; j++)
10:      if A- [i][j] =0
11:        inf=true;
12:      if Outf && inf;
13:        Printf(" pi is a isolate place"); }
14: For (i=0; i<=m; i++)
15:   { For (j=0; j<=n; j++)
16:     If A- [i][j] =1 && ! (Type(E(pi, tj) ⊆ C(pi) ∧ Type(Var(E(pi, tj)) ⊆ Σ)
17:       Printf("Arc function is error"); }
18: For (j=0; j<=n; j++)
19:   { For (i=0; i<=m; i++)
20:     if A+ [j][i]=0
21:       Outf=true;
22:       For (i=0; i<=n; j++)
23:         if A- [i][j]=0
24:           inf=true;
25:         if Outf && inf
26:           Printf(" tj is an isolate transition"); }
27: For (j=0; j<=n; j++)
28:   { If ! (Type(G(tj))=B ∧ Type(Var(G(tj)) ⊆ Σ)
29:     ,Printf("Guard function of tj is error"); } }

```

##### 4.2 组合 Web 服务死变迁验证

迁移矩阵<sup>[13]</sup>表示由变迁触发状态的变化,可以判断 Petri 网模型是否会有死锁,并且提供解开死锁的办法,可以用来验证 Web 服务组合是否具有死锁性质。下面通过应用迁移矩阵来检验 Web 服务的 Petri 网模型中是否存在死变迁,以此验证服务模型的活性。算法如下。

$M(k) = \{P_1(k), P_2(k), \dots, P_m(k)\}$  表示 Petri 网的标识; $L_{BPj}$  表示迁移矩阵。

```

1: Void deadlockcheck (WS_Model)
2: M(k) = {P1(k), P2(k), ..., Pm(k)};
3: { For (i=1; i<=m; i++)
4:   { M'(pi) = Σ Pj(k). LBPj;
5:   If M'(pi) is integer
6:     M'(pi) = 1; // some transition can fire.
7:   Else M'(pi) = 0; // no transition can fire. }
8: For (i=1; i<=m; i++)
9:   { if M'(pi) = 1
10:    Tfire = Tfire ∪ {ti};
11:   if M'(pi) = 0
12:     printf(" pi is an end marking"); }
13:   Tdead = T - Tfire;
14: If Tdead () null
15:   Printf("There are some dead transitions in Petri net");
16: else Printf("There are no dead transitions in Petri net"); }

```

#### 4.3 组合 Web 服务可达性和有界性验证

关联矩阵和状态方程配合初始状态和目标状态建立方程,以此方程是否有解来判断 Petri 网模型是否可以从初始状态到达目标状态,可以用来验证 Web 服务组合是否具备可达性,具体的算法如下。

```

1: Void ReachableCheck(WS_Model)
2: A[] [] = A+ [] [] - A- [] [];
3: M(0) = {P1(0), P2(0), ..., Pm(0)}; //start situation.
4: M(K) = {P1(K), P2(K), ..., Pm(K)}; //destination situation.
5: M(K) = M(0) + A[] [] X; //state equation.
6: Using Guss elimination method to get the solution of the X vector;
7: If X is a nonnegative integer vector
8:   Printf("M(K) is reachable");
9: For (i=1; i<=K; i++)
10:  M(i) = M(i-1) + (Ai*)T; //get all situation markings.
11: B=0;
12: For (i=1; i<=K; i++)
13:  { If max(M(i)) > B
14:    B = M(i); } //get the max boundedness.
15: Printf("WS_Model's boundedness is b%", B);
16: If B=1
17:   Printf("WS_Model is safe");

```

#### 5 应用

为了保证独立、自治的 Web 服务组合后的正确性,避免运行时因错误而造成重大经济损失,必须对组合 Web 服务进行形式化建模,并进行分析和验证。下面以电子商务中网上订购为例,讨论相应 Web 服务组合的形式化建模。

客户向代理提交订单和网上支付信息,客户代理将订单提交给库存数据查询服务,库存查询服务根据客户订单对仓库中货物信息和数量进行查询,并将查询信息反馈给客户。如果库存货物规格和数量符合客户需求,则将确认信息作为网上支付的前提条件,并将订单信息作为货物配送的依据,如果库存与订单不相符,则转向结束。客户代理将客户支付信息提交给银行支付服务,在库存信息和客户订单相符的条件下,银行支付服务按照规定的价格和数量进行网上支付,并将支付信息反馈给客户。如果网上支付成功,则货物配送服务根据支付确认信息按照订单将货物发送给客户,如果网上支付不成功,则转向结束。

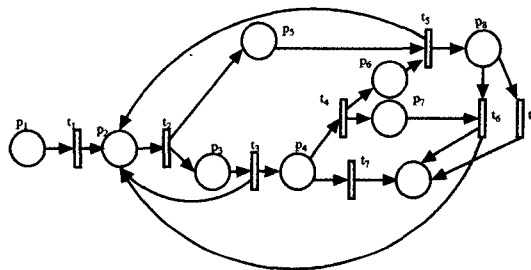


图 10 网上订购服务组合

$P_1$ : 客户的相关信息。  
 $P_2$ : 客户提交给代理的信息。

- P<sub>3</sub>: 客户代理提交的客户订单。  
 P<sub>4</sub>: 查询到的与订单相应的货物库存信息。  
 P<sub>5</sub>: 客户代理提交的客户网上支付信息。  
 P<sub>6</sub>: 将库存与订单相符作为网上支付的条件。  
 P<sub>7</sub>: 订单信息作为货物配送的依据。  
 P<sub>8</sub>: 网上支付成功或失败的信息。  
 t<sub>1</sub>: 客户将相关信息提交给客户代理。  
 t<sub>2</sub>: 客户代理提交客户订单和网上支付信息。  
 t<sub>3</sub>: 查询与订单相应货物的库存信息, 并返回给客户代理。  
 t<sub>4</sub>: 传输库存与订单相符信息。  
 t<sub>5</sub>: 进行网上支付, 并将支付确认信息返回给客户代理。  
 t<sub>6</sub>: 根据网上支付成功信息和订单来配送货物, 并转向结束。  
 t<sub>7</sub>: 订单和库存不符, 转向结束。  
 t<sub>8</sub>: 网上支付不成功, 转向结束。

我们可以利用上一节所给的算法来验证网上订购服务组合系统的语法正确性、可达性、活性和有界性。

**总结和进一步工作** Web 服务组合是 Web 服务的一个重要研究方向, 而 Web 服务形式化建模和验证是保证组合得以正确实施的前提。为了形式化描述语义 Web 服务组合, 本文以有色 Petri 网为基础, 通过使用不同颜色的 token 以及 token 数的变化来表示 Web 服务 OWL-S 模型中的 IOPE, 以此来描述 Web 服务的语义, 并给出了 8 种组合模式的有色 Petri 网模型, 以此进行组合即可形式化描述语义 Web 服务组合模型。然后对 Petri 网模型进行分析, 给出了组合模型语法正确型、可达性、活性和有界性的算法。最后, 以网上购物

系统为例来形式化描述 Web 服务组合模型。下一步的工作是分析组合模型, 验证语义 Web 服务的语义正确性。

## 参考文献

- Baeten J C M. A brief history of process algebra. *Theor. Comput. Sci.*, 2005, 335(2-3): 131~146
- Petri C A. *Kommunikation mit Automaten*; [Ph. D. thesis]. University Bonn, Germany, 1962
- Milner R. *Communicating and Concurrency*. Prentice-Hall, 1989
- Hoare C A R. *Communicating Sequential Processes*. New Jersey: Prentice-Hall, 1985
- Baeten J C M, Weijland W P. *Process algebra*. Cambridge Tracts in Theoretical Computer Science 18, Cambridge University Press, 1990
- Milner R. *Communicating and Mobile System: The Pi Calculus*. Springer-Verlag, 1999
- Hamadi R, Benatallah B. A Petri Net-based Model for Web Service Composition. In: *Proceedings of the Fourteenth Australasian database conference on Database technologies 2003*, Adelaide, Australia, February 2003. 191~200
- 郭玉彬, 杜玉越, 奚建清. Web 服务组合的有色网模型及运算性质. *计算机学报*, 2006, 29(7): 1067~1075
- 钱柱中, 陆桑璐, 谢立. 基于 Petri 网的 Web 服务自动组合研究. *计算机学报*, 2006, 29(7): 1067~1075
- Narayanan S, McIlraith S A. Simulation, verification and automated composition of Web services. In: *Proc. of the 11th Int'l World Wide Web Conf. WWW2002*. Honolulu: ACM, 2002. 77~88
- Jensen K. *Coloured Petri Nets; Basic Concepts, Analysis Methods and Practical*. Volume 1, Basic Concepts, Monographs in Theoretical Computer Science. Berlin, Heidelberg, New York: Springer-Verlag, 1997
- OWL-S: Semantic Markup for Web Services. <http://www.w3.org/Submission/OWL-S>
- Liu Jiang, Itoh Y, Miyazawa I, et al. A Research on Petri Net Properties Using Transitive Matrix. *IEEE Press*, 1999
- Networks (MWCN'02). Stockholm, Sweden, 2002. 140~144
- Gao Zhenguo, Wang Ling, Yang Mei, et al. CNPGSDP: An efficient group-based service discovery protocol for MANETs. *Computer Networks*, 2006. 3165~3182
- Schiele G, Becker C, Rothermel K. Energy-efficient Cluster-based Service Discovery for Ubiquitous Computing. In: *Proc. of the 11th ACM SIGOPS European Workshop*. Leuven, Belgium, 2004
- Sailhan F, Issarny V. Scalable Service Discovery for MANET. In: *Proc. of the 3rd International Conference on Pervasive Computing and Communications*, Hawaii, USA, 2005. 235~244
- Kozat U C, Tassiulas L. Service discovery in mobile Ad hoc networks: an overall perspective on architecture choices and network layer support issues. *Ad Hoc Networks*, 2004, 5(2): 23~44
- Koodli R, Perkins C. Service Discovery in On-Demand Ad Hoc Networks. IETF Internet draft, draft-koodli-manet-servicediscovery-00.txt, October 2002
- Li L, Lamont L. A Lightweight Service Discovery Mechanism for Mobile Ad hoc Pervasive Environment Using Cross-layer Design. In: *Proc. of the 2nd Mobile Peer-to-Peer Computing Workshop, in conjunction with the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom'05)*. Hawaii, USA, 2005. 55~59
- Ververidis C N, Polyzos G C. Extended ZRP: a routing layer based service discovery protocol for mobile Ad hoc networks. In: *Proc. of the Second Annual International Conference on Mobile and Ubiquitous Systems; Networking and Services (MobiQuitous 2005)*. San Diego, California: IEEE Computer Society, 2005. 65~72
- Cheng Liang. Lightweight service advertisement and discovery in mobile Ad hoc networks. In: *Proc. of 14th Annual Wireless and Optical Communications (WOCC 2005)*. Newark, USA, 2005. 46

(上接第 27 页)

- Robinson R, Indulska J. Superstring: A Scalable Service Discovery Protocol for the Wide-area Pervasive Environment. In: *Proc. of the 11th IEEE International Conference on Networks*. Sydney, Australia, 2003. 699~704
- Arabshian K, Schulzrinne H. GloServ: Global service discovery architecture. In: *First Annual International Conference on Mobile and Ubiquitous Systems; Networking and Services (MobiQuitous'04)*. Boston, USA, 2004. 319~325
- Cheshire S. DNS-based Service Discovery. Internet draft, 2006. <http://files.dns-sd.org/draft-cheshire-dns-sd.txt>
- Nidd M. Service Discovery in DEAPspace. *Personal Communications*, 2001, 8(4): 39~45
- Helal S, Desai N, Verma V, et al. Konark - A Service Discovery and Delivery Protocol for Ad-Hoc Networks. In: *Proc. of the Third IEEE Conference on Wireless Communication Networks*. New Orleans, USA, 2003. 2107~2133
- Lee C, Helal A, Desai N, et al. Konark: A System and Protocols for Device Independent, Peer-to-Peer Discovery and Delivery of Mobile Services. *IEEE Transaction Systems, Man and Cybernetics, Part A*, 2003, 33(6): 682~696
- Campo C, Garcia-Rubio C, Marin A, et al. PDP: A lightweight discovery protocol for local-scope interactions in wireless Ad hoc networks. *Computer Networks*, 2006, 50(17): 3264~3283
- Ratsimor O, Chakraborty D, Joshi A, et al. Allia: Alliance-based Service Discovery for Ad Hoc Environments. In: *Proc. of the 2nd International Workshop on Mobile Commerce*. Atlanta, USA, 2002
- Chakraborty D, Joshi A, Yesha Y, et al. GSD: A Novel Group-based Service Discovery Protocol for MANETs. In: *Proc. of the 4th IEEE Conference on Mobile and Wireless Communications*