

基于 MDA 的 Web 服务合成^{*}

刘 峰 谭庆平 杨艳萍

(国防科技大学计算机学院 长沙 410073)

摘 要 随着已有 Web 服务数量的不断增加,如何利用这些现有的 Web 服务创建新的更复杂 Web 服务成为一项新的研究课题。特别地,利用 MDA 进行 Web 服务合成已经成为研究的热点。本文研究了基于 MDA 的 Web 服务合成问题,针对 Web 合成的动态行为建模部分,提出建立基于 UML 的平台无关 Web 服务动态行为模型的方法,并且给出了平台无关动态行为模型到 Web 服务合成语言 BPEL、WSCI 之间的转换方法。

关键词 Web 服务合成,MDA,动态行为模型

MDA-Based Web Services Composition

LIU Feng TAN Qing-Ping YANG Yan-Ping

(School of Computer Science, National University of Defense Technology, Changsha 410073)

Abstract A number of Web services are now available and it therefore seems natural to reuse existing Web services to create composite Web services. Specially, it is a hot topic to study the problem of Web services composition using MDA. This paper is interested on the problem of Web services semantic composition. It gives a PIM of Web services composition for solving the problem above dynamic-model of Web services composition. Then, we give a method of translating from PIM to BPEL and WSCI.

Keywords Web services composition, MDA, Dynamic behavior model

1 引言

Web 服务被定义为使用标准的 Web 技术与其它服务进行交互的自包含、自描述的模块化应用,它提供了从简单请求到复杂商务处理的功能;一旦 Web 服务被部署,其他的应用(包括 Web 服务)就能够发现并调用所部署的服务^[1]。

目前,已经有越来越多的企业将自己的业务作为 Web 服务发布。然而,一个单独的 Web 服务很可能限制其所拥有的能力,所以工业界和学术界都希望能够通过合成现有的 Web 服务来创造新的 Web 服务。所谓 Web 服务合成,指的是从互联网中选取相对简单、可用的 Web 服务,并将它们组合成新服务的技术^[2,3]。合成后的新服务称为复合服务;用于合成复合服务的子服务称为构件服务。

合成 Web 服务的开发可分成这几个阶段:发现可用的候选 Web 服务,计算它们可能的合成方式,使用特定的服务合成语言描述其合成,将描述文档发布到特定执行引擎供服务使用者调用执行。目前该领域大多数的研究都是针对其中部分问题提出解决方案,如很多组织提出了 Web 服务合成语言(BPML, BPMN, BPEL, BPSS, WSCI),它们是复合服务的描述、执行的平台;还有很多研究者利用 Petri 网^[4,5]、线性逻辑^[6]、状态图^[7]、有限自动机^[8]、图论和接口自动机^[9]进行 Web 服务合成的建模和计算,但是没有一个方法证明是与具体的执行平台无关的。

日前,利用 OMG 组织提出的 MDA(Model Driven Architecture)进行 Web 服务合成成为研究的热点^[10,11]。本文研究了基于 MDA 的 Web 服务合成的方法,重点是 Web 服务合成建立模型,并利用模型转换技术对模型进行转换。通常,

Web 服务合成模型可以分为静态结构模型和动态行为模型^[13]。静态结构模型描述了复合服务的结构和功能,动态行为模型描述了复合服务的合成方式,目前这方面的研究大都集中于静态结构模型方面。本文主要研究如何建立 Web 服务合成的动态行为模型。Gardner^[14]和 Grønmo^[15]曾提出把 UML 活动图扩展为业务流程建模,但没有给出具体进行 Web 服务合成的方法。本文首先利用 UML 活动图建立 Web 服务合成平台无关的动态行为模型,然后通过模型转换,将平台无关模型转换为与特定平台相关的模型,如 BPEL 和 WSCI。

2 基于 MDA 的 Web 服务合成流程

MDA 是一种描述和建立系统的新方法,通过分离系统在特定平台上的功能实现与功能描述,从而实现广泛的应用互操作。它将软件系统的模型分离为平台无关模型 PIM 和特定平台模型 PSM^[12],同时通过转换规则将它们统一起来,以这样的方式来解决需求变更所带来的问题。平台无关模型 PIM 是对工作流程的高层次抽象,其中不包括与实现技术相关的信息,平台相关模型 PSM 是跟特定平台相关的模型。为 Web 服务合成的动态行为建立平台无关的模型,可以为建模人员屏蔽特定平台的相关细节,提高模型的抽象层次,降低建模的复杂度,有助于模型的理解。利用模型转换技术可以将 Web 服务合成动态行为的平台无关模型转换为可执行的 Web 服务合成语言,发布到执行引擎实际运行。图 1 说明了基于 MDA 的 Web 服务合成流程。

基于 MDA 的 Web 服务合成方法主要分 3 个步骤:

(1)用 UML 为复合服务粗略建模,主要反映了用户的需

^{*} 国家高技术研究发展计划(2003AA001023)。刘 峰 硕士研究生;谭庆平 教授,博士生导师;杨艳萍 博士研究生。

求,描述复合服务的基本特征,然后根据复合服务的粗略模型,利用相关算法搜索构件服务。将搜索到的构件服务的 WSDL 文件转化为 UML 类图,并以此对复合服务的模型进行精化。

(2)将复合服务的模型分成两个部分进一步精化:静态结构模型和动态行为模型。静态结构模型是复合服务的接口模型,可以用 UML 类图表示;动态行为模型说明了构件服务的合成方式,可以用 UML 活动图表示。

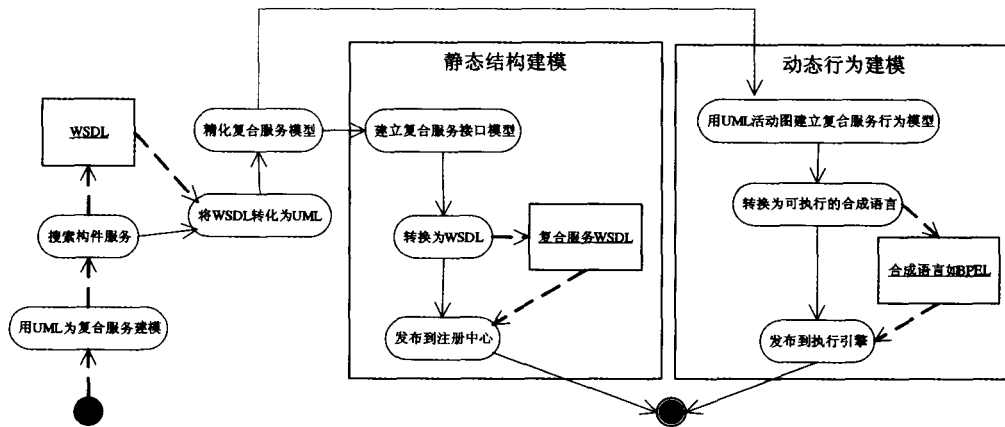


图1 基于MDA的Web服务合成流程

(3)最后将精化后完整的复合服务静态结构模型转换为 WSDL,发布到注册中心;将精化后完整的动态行为模型转换为 Web 服务合成语言(如 BPEL)发布到执行引擎。

3 建立复合服务动态行为模型

从图1所示的Web服务合成流程可以看到,用UML活动图建立复合服务的动态行为模型十分重要。实际上,复合服务的动态行为建模是一种流程建模^[13],目的是解决以什么样的顺序和方式调用不同的构件服务。因此,动态行为建模类似于工作流建模,很多工作流建模中的概念和方法都可以借鉴。一般来说,用UML活动图构建动态行为模型需要考虑3方面的问题:

(1)控制流模式。Barros在工作流建模中定义了控制流模式的概念^[17],Aalst结合工作流建模的研究成果总结了20多种关于控制流方面的模式^[16]。在Web服务合成的动态建模中可以借鉴工作流中控制流模式的概念及很多定义好的工作流控制流模式^[15]。有5种基本的控制流模式对于Web服务合成是必需的:Sequence, Parallel split, Synchronization, Exclusive choice, Simple merge。Hamadi^[4]描述了两种较复杂的控制流模式 Discriminator 和 Selector,它们也应该在Web服务合成中得到支持。上述7种控制流模式对于Web服务合成的动态行为建模十分重要,其它复杂的控制流模式同样会起到帮助。Wohed对UML1.4和UML2.0的活动图支持的控制流模式进行了分析^[20],Wohed和Aalst分别对BPEL和WSCI支持的控制流模式进行了分析^[18,19],得出的结论是:对于Web服务合成必须的5种基本的控制流模式,UML活动图、BPEL和WSCI都支持;Discriminator模式在UML2.0中是支持的,但是在UML1.4、BPEL和WSCI中不支持;Selector模式在UML、BPEL和WSCI中都没有显式的支持,能否通过其它复杂的模式来实现,有待进一步研究。总之,由于UML活动图和当前的Web服务合成语言的能力限制,目前进行Web服务合成的动态行为建模主要采用5种基本的控制流模式。

(2)数据流转换。由于在Web服务合成中需要调用不同的构件服务或者本地的软件构件,因此在不同的构件服务和

软件之间会存在数据流^[21]。数据流中可能存在的问题是数据类型的 mismatch,比如合成中需要一个构件服务的输出作为另外一个构件服务的输入,但是前者的输出与后者的输入在数据类型上是不一致的。为了保证正确性,必须对输出的数据进行转换,然后才作为后续构件服务的输入。Grønmo给出了3种不同的数据流转换方法,其中被广泛采用的是如图2所示的方法^[15],它引入数据转换的动作解决数据类型不匹配的问题。

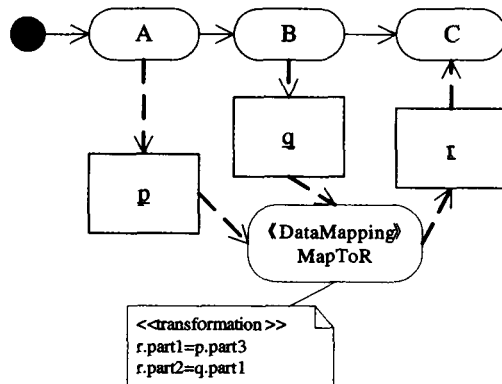


图2 数据流转换方法^[15]

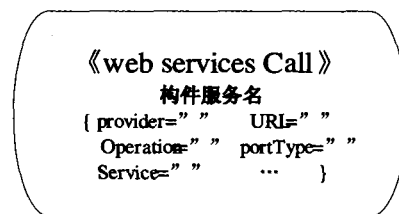


图3 调用构件服务

(3)调用构件服务。Web服务合成实际上是将不同的构件服务组合起来完成特定的功能,形成复合服务。因此,在Web服务合成的动态行为建模中,如何为调用构件服务建模非常重要。调用构件服务需要知道构件服务的输入、输出参

数,这些可以通过数据流转换得到;需要知道调用构件服务的时机和条件,这些可以通过控制流模式进行建模;此外,调用构件服务所需要的其它信息,如构件服务的访问地址等,可以通过 UML 中的扩展机制进行建模,图 3 引入了调用构件服务的构造型解决这个问题。

4 将复合服务动态行为模型转换为 Web 服务合成语言

用 UML 活动图建立平台无关的复合服务动态行为模型后,可以用模型转换的技术将其转换为特定的合成语言。图 4 说明了 MDA 中模型转换的方法^[22]。

MDA 中,元模型是非常重要的概念,它提供了描述模型的语言和规则。MDA 中所有的元模型都是基于一个公共的元元模型,称为元对象设施(MOF)。转换规则描述了从源模型到目标模型的映射规则,指导从源模型(如 PIM)到目标模

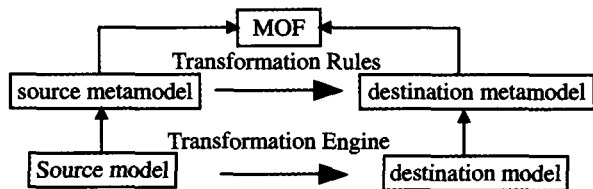
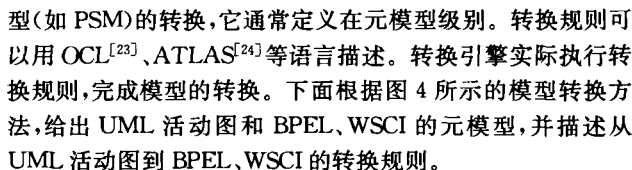


图 4 MDA 中的模型转换

参考 UML2.0 规范^[25],我们给出如图 5 所示简化的 UML 活动图元模型。

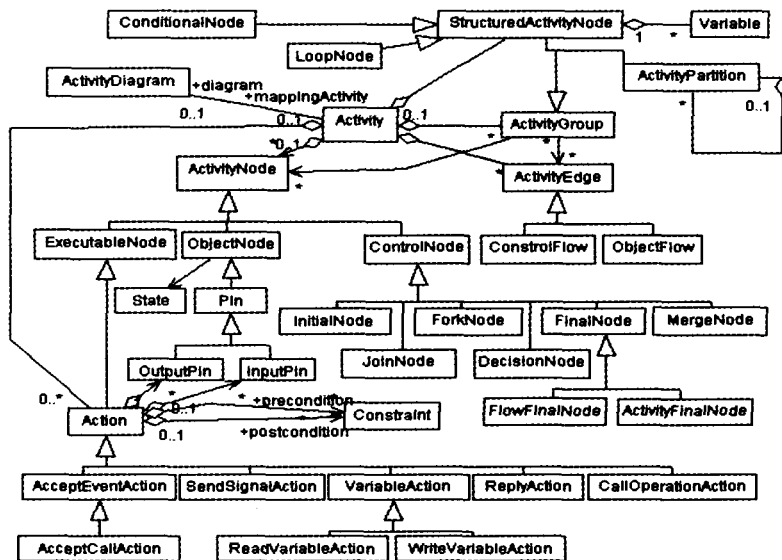


图 5 UML 活动图元模型

图 5 所示的 UML 活动图元模型主要包含了 Activity, ActivityGroup, ActivityPartition, ObjectFlow, ObjectNode, ControlNode, ConditionalNode, CallOperationAction 和 StructuredActivityNode 等基本元素,关于它们的意义及元模型中

其他元素的意义详见 UML2.0 的规范^[23]。

4.1 从 UML 活动图到 BPEL 的转换

参考 BPEL 规范 1.1 版本^[26],我们给出了一个图 6 所示的简化的 BPEL 元模型。

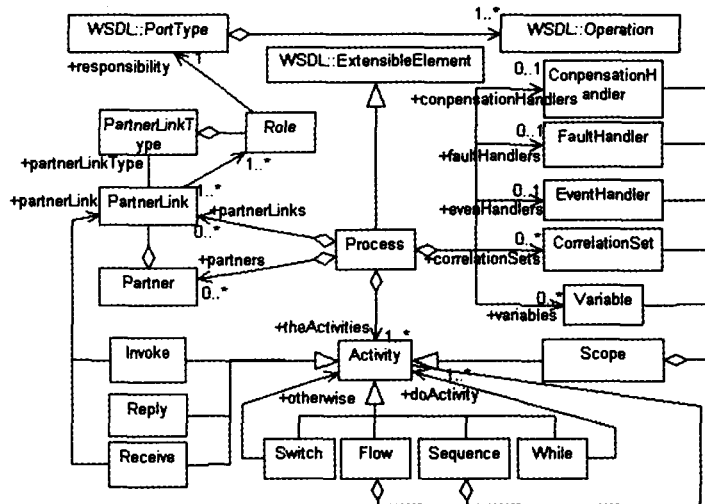


图 6 BPEL 元模型

表2 部分 UML 活动图元模型元素与 WSCI 元模型元素对应关系

UML 2.0	WSCI 1.0
Activity Partition	<interface>
ConditionalNode, DecisionNode	<switch>
StructuredActivityNode	<process>
CallOperationAction	<action>
ControlFlow	<sequence>
StructuredActivityNode, InterruptibleActivityRegion	<context>
AcceptEventAction	<onMessage>
ExceptionHandler	<exception>

```

rule UMLCallOperation2WSCIAction{
From ucp; UML! CallOperationAction
To wa; WSCI! Action
mapsTo ucp(
name←ucp.name,
role←"tns:".concat(ucp.oclAsType(ActivityNode).inGroup→exists
(t, ActivityPartition).oclAsType(ActivityPartition).name),
operation←"tns:".concat(port).concat("/").concat(ucp.operation.
name),
--if there is a correlate attribute within the action
correlate←if ucp→attributes→exists(correlate)
then [UMLCallOperationAction2WSCICorrelate]ucp
endif)
}
rule UMLCallOperation2WSCICorrelate{
From ucp; UML! CallOperationAction(ucp→attributes→exists(cor-
relate))

```

```

To wc; WSCI! Correlate
mapsTo ucp(
correlation←ucp.correlation, --is a tag value
instantiation←if ucp→attributes→exists(instantiation)
then ucp.instantiation --is a tag value
endif)
}

```

图9 UML2WSCI 的转换规则

UML 元模型的 CallOperationAction 元素通过转换规则 UMLCallOperation2WSCIAction 转换为 WSCI 元模型的 Action 元素。如果 UML 元模型的 CallOperationAction 元素的属性中有 correlate, 则元素 CallOperationAction 可以通过规则 UMLCallOperation2WSCICorrelate 转换为 WSCI 元模型的 Correlate 元素。

5 样例

图10说明了从UML活动图到WSCI(左边)和BPEL(右边)转换的例子。图10所示的UML活动图为代理机场订票的用例建模。用户首先发送请求信息,包括时间、目的地等。接受到这些信息后,代理订票系统调用机场提供的Web服务,查询相关的航班,将查询到的信息返回给用户;而后接受用户选定的飞机后,调用机场提供的订票的Web服务,完成订票;最后将机票返回给用户。

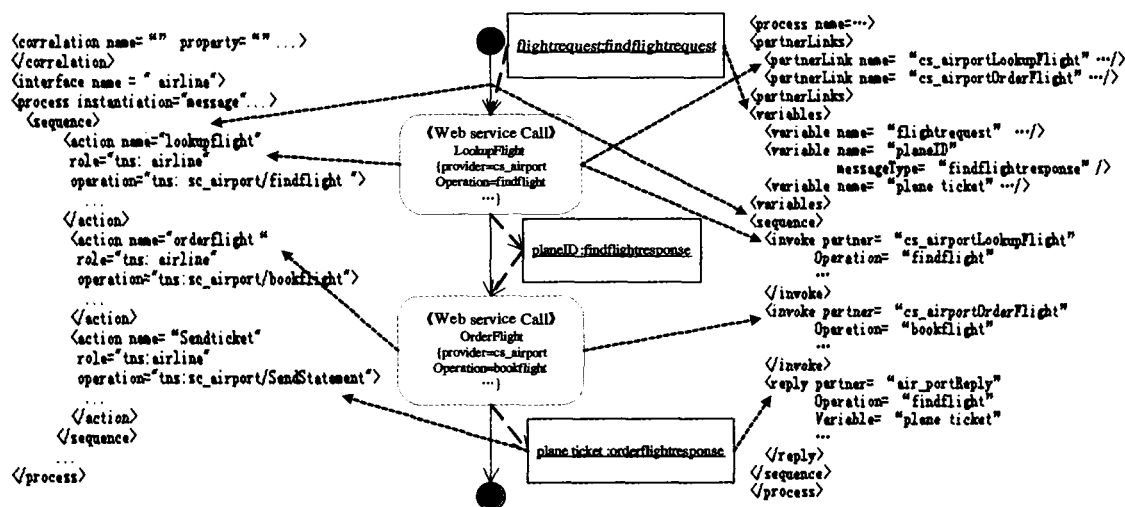


图10 从UML活动图转换为WSCI(左)和BPEL(右)

结束语 本文研究了基于MDA的Web服务合成的方法;针对Web服务合成中动态行为模型部分,说明了用UML活动图建立平台无关的动态行为模型的关键问题和解决方法。最后,本文给出了从基于UML活动图的Web服务合成动态行为模型到Web服务合成语言BPEL和WSCI的转换方法。但是,针对BPEL和WSCI语言的细节,本文提出的建模和转换方法有待进一步完善,例如名字空间的问题、利用UML图中的泳道为角色建模等,这是我们下一步工作需要解决的。同时,从BPEL语言或WSCI语言到UML模型间的逆向转换工程也是我们下一步研究的重点。

参考文献

- 1 Tidwell D. web services: The web's next revolution. <http://www-106.ibm.com/developerworks/webservices/>, 2000
- 2 Benatallah B, Dumas M. Declarative Composition and Peer-to-Peer Provisioning of Dynamic Web Services. In: Proc of the 18th Intl Conf Data Eng. (ICDE. 02), In: IEEE Computer Society, 2002

- 3 Boualem M, Fauvet M-C, Rabbi FA. Towards Patterns of Web Services Composition. Sydney: University of New South Wales, 2001
- 4 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
- 5 Narayanan S, Mellraith S A. Simulation, Verification and Automated Composition of Web Services. In: Proceedings of the eleventh international conference on World Wide Web, 2002. 77~88
- 6 Rao J, Kuungas P, Matskin M. Application of Linear Logicto Web Service Composition. In: Proceedings of the First International Conference on Web Services (ICWS'2003), 2003. 3~9
- 7 Benatallah B, Dumas M, Sheng Q Z, et al. Declarative Composition and Peer-to-Peer Provisioning of Dynamic Web Services. In: Proceedings of the 18th International Conference on Data Engineering (ICDE), 2002
- 8 Berardi D, Calvanese D, De Giacomo G, et al. Automatic Composition of e-Services. In: Proceedings of the First International Conference on Service-Oriented Computing (ICSOC), 2003. 43~58
- 9 Hashemian S V, Mavaddat F. A Graph-Based Approach to Web Services Composition. In: Proceedings of the Symposium on Applications and the Internet (SAINT), 2005
- 10 Bezivin J, Hammoudi S, Lopes D, et al. An Experiment in Map-

- ping Web Services to Implementation Platforms, Atlas Group, INRIA and LINA University of Nantes; [Research Report], March 2004
- 11 Bordbar B, Staikopoulos A. Automated Generation of Metamodels for Web Service Languages. Atlas Group, INRIA and LINA University of Nantes; [Research Report]. 2004
 - 12 MDA Guide Version 1. 0. <http://www.omg.org/cgi-bin/doc?mda-guide>
 - 13 Thöne S, Depke R, Engels G. Process-Oriented, Flexible Composition of Web Services with UML. In: Int Workshop on Conceptual Modeling Approaches for e-Business, A Web Services Perspective (eCOMO 2002), Tampere, Finland, 2002
 - 14 Gardner T. UML Modeling of Automated Business Processes with a Mapping to BPEL4WS. In: 17th European Conference on Object-Oriented Programming (ECOOP), Darmstadt, Germany, 2003
 - 15 Grønmo R, Solheim I. Towards Modeling Web Services Composition in UML. In: The 2nd International Workshop on Web Services: Modeling, Architecture and Infrastructure (WSMAI-2004), Porto, Portugal, 2004
 - 16 v d Aalst W M P. Don't go with the flow; Web Services composition standards exposed. Trends & Controversies Jan/Feb 2003 issue of IEEE Intelligent Systems, 2003
 - 17 van der Aalst W M P, ter Hofstede A H M, Kiepuszewski B, et al. Workflow Patterns. Distributed and Parallel Database, 2003, 14(1): 5~51
 - 18 Wohed P, v d Aalst W M P, Dumas M, t Hofstede A H M. Pattern Based Analysis of BPEL4WS; [FIT-TR-2002-04 Technical Report]. Queensland University of Technology, Australia, QUT 2003
 - 19 v d Aalst W M P, Dumas M, t Hofstede A H M, et al. Pattern Based Analysis of BPML (and WSCI); [FIT-TR-2002-05 Technical Report]. Queensland University of Technology, Australia, QUT 2003
 - 20 Wohed P, v d Aalst W M P, Dumas M, et al. Pattern-based Analysis of UML Activity Diagrams. BETA Working Paper Series, WP 129, Eindhoven University of Technology, 2004
 - 21 Skogan D, Grønmo R, Solheim I. Web Service Composition in UML. 8th International IEEE Enterprise Distributed Object Computing Conference (EDOC 2004), California, 2004
 - 22 Bezivin J, Hammoudi S, Lopes D, et al. An Experiment in Mapping Web Services to Implementation Platforms, Atlas Group, INRIA and LINA University of Nantes; [Research Report]. March 2004
 - 23 OMG. Object Constraint Language Specification. Object Management Group, Adopted Specification, 2003
 - 24 Bezivin J, Breton E, Dupe G, et al. The ATL Transformation-based Model Management framework; [Research Report]. Atlas Group, INRIA and IRIN, September 2003
 - 25 Unified Modeling Language: Superstructure (version 2. 0), <http://www.omg.org/docs/ptc/03-08-02.pdf>
 - 26 BEA, IBM, Microsoft. SAP AG and Siebel Systems; Business Process Execution Language for Web Services, Version 1. 1. May 2003
 - 27 W3C. Web Services Choreography Interface (WSCI) 1. 0. W3CNote, August 2002
 - 28 Grønmo R, Skogan D, Solheim I, et al. Model-driven Web Services Development. The 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE-04), Taipei, Taiwan, 2004
 - 29 Bezivin J, Dupé G, Jouault F, et al. First Experiments with the ATL Model Transformation Language: Transforming XSLT into XQuery. 2nd OOP' SLA Workshop on Generative Techniques in the context of Model Driven Architecture, October 2003

(上接第 108 页)

另外, μ TESLA 并没有考虑 DoS 攻击的问题^[4,6]。恶意节点广播错误数据包, 节点会将这些数据包保存起来等待密钥公布后验证, 这样将可能耗尽节点的资源。

针对上述问题我们提出以下方案来对 SPINS 安全框架协议进行改进。

对于 μ TESLA 协议在基站安全初始化时, 如果在某一节点在中途需要加入网络时, 可以先让该节点只与基站使用 SNEP 协议进行密钥分配和相关认证, 此时无须与其它节点进行数据包的收发, 而在网络中有数据包进行传送时, 而此加入节点正处于该通信链路时, 再进行相关的密钥分配和相关认证工作。利用这种节点加入方式会增加通信过程的开销, 但避免了通信开始前节点加入时的巨大耗费, 这样可以使整个传感器网络的资源耗费降到最低。

另外, 对于 μ TESLA 协议运行时节点加入过程存在的两个问题:

第一: $A \rightarrow S$ 的过程没有进行认证, 所以比较容易受到 DoS 的攻击。

第二: 整个过程没有加密。下面就 μ TESLA 协议的运行中节点加入过程进行改进, 可以使用加密的方法对其进行改进, 加密过程如下:

$$A \rightarrow S: (N_M | R_A), \text{MAC}(K_{am}, N_M | R_A)$$

$$S \rightarrow A: (T_s | K_i | T_i | T_{in} | d) K_{as}, \text{MAC}(K_{am}, N_M | T_s | K_i | T_i | T_{in} | d)$$

其中, K_{as} 为节点与基站之间的通信加密密钥, K_{am} 是通信认证密钥, 都是通过共享的主密钥产生。这样, 上述加密过程对于 $A \rightarrow S$ 进行了认证, 同时对于 $S \rightarrow A$ 过程也进行了加密考虑。理论上该改进过程解决了 μ TESLA 协议运行时节点加入过程存在的问题。

对于 μ TESLA 中另一类 DoS 攻击的问题, 即在传感器网络中存在恶意节点广播错误数据包, 节点会将这些数据包保存起来等待密钥公布后验证, 这样将可能耗尽节点的资源, 对于这样的问题可以使用如下改进方案解决: 首先在整个传感器网络中设置一个鲁棒性很好的初始密钥(非公开)对每个广

播包进行一次预处理, 可以让节点先过滤掉那些纯粹的错误广播包, 避免浪费节点资源去存储它, 当然此方法中初始密钥的选择非常重要。

针对 DoS 攻击, 多级 μ TESLA 协议^[4,7]有较好的特性, 但由于其难于实现并且又会占用更多的节点内存和计算资源, 因此在具体实现时我们可以针对目标应用系统选择使用或者部分使用其中的安全处理机制, 以此来部分解决 SPINS 安全框架协议存在的问题。

总之, SPINS 定义的是一个安全协议框架, 在传感器网络中具体使用时, 很多方面需要进行进一步深入的研究。

结束语 传感器网络的安全问题随着其应用逐渐成为现实而备受关注。文中就传感器网络的安全问题进行了讨论, 对目前一种比较流行的传感器网络安全框架协议——SPINS 进行了深入的研究, 并就其存在的问题提出了一些改进措施, 并进行了相关的理论分析。但对于改进的 SPINS 协议在实际应用中还需考虑诸多问题, 如加密算法的选择、消息认证算法的选择、密钥生成算法、随机数产生器。所有这些都是我们以后的研究方向。

参考文献

- 1 Pottie G J, Kaiser W J. Embedding the internet: wireless integrated network sensors [C]. Communications of the ACM, 2000, 43(5): 51~58
- 2 Wood A D, Stankovic J A. Denial of service in sensor networks. IEEE Computer, 2002, 35(10): 54~62
- 3 Pister K, Hohlt B, Jeong J, Doherty L, Vainio J P. Ivy—A sensor network infrastructure. 2003. <http://www-bsac.eecs.berkeley.edu/projects/ivy>
- 4 Perrig A, Szewczyk R, et al. SPINS: Security protocols for sensor networks. Wireless Networks, 2002, 8(5): 521~534
- 5 Li Dan, Wong K D, Hu Yu Hen, Sayeed A M. Detection, classification, and tracking of targets. Signal Processing Magazine, IEEE, 2002, 19(2): 17~29
- 6 Asada G, Dong M, Lin T S, et al. Wireless Integrated Network Sensor: Low Power System on a Chip. In: Proc. of the European Solid State Circuits Conf. 1998
- 7 Elson J, Estrin D. Time Synchronization for Wireless Sensor Networks. In: Proc. of the Intl Parallel and Distributed Processing Symposium (IPDPS). Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, San Francisco, CA, Apr. 2001