

基于 UML 的卫星网路由算法仿真系统研究^{*})

孙知信 王汝传 王绍棣 张登银 李 军

(南京邮电学院计算机系 南京210003)

摘 要 本文首先介绍了我们提出的基于时空的路由算法思想,然后用 UML(Unified Modeling Language)对基于该算法的仿真系统(Satellite Network Simulation based on Space Time, STSNS)进行建模。在仿真建模过程中,首先用用例(use case)对仿真系统的需求进行了分析,然后建立了 STSNS 的静态模型和动态模型,并用 UML 中相应的图例来描述。最后给出了 STSNS 的详细实现技术。STSNS 能对基于时空的路由算法进行很好的仿真,仿真的结果表明,基于时空的路由算法能很好地满足卫星网路由的要求。

关键词 路由算法, UML, 仿真建模

Research on the Simulation System of Satellite Network Routing Algorithm Based on UML

SUN Zhi-Xin WANG Ru-Chuan WANG Shao-Di ZHANG Deng-Yin LI Jun

(Department of Computer Science and Technology, NJUPT, Nanjing 210003)

Abstract This paper firstly introduces the idea of routing algorithm based on space time. And then it uses UML (Unified Modeling Language) to process modeling to a simulation system (Satellite Network Simulation based on Space Time, STSNS) based on the algorithm. In the process of the simulation modeling, the paper analyzes the requirements of STSNS using use case model. Afterwards, it creates the static models and dynamic models of STSNS, then using corresponding diagrams of UML to describe the models. Finally, it presents the detailed implementing techniques of STSNS. STSNS can simulate perfectly to the routing algorithm based on space time. The results of the simulation indicate that the algorithm can satisfy the requirements of the satellite routing.

Keywords Routing algorithm, UML, Simulation modeling

1 引言

针对卫星网络所具有的拓扑结构动态变化等主要特点,我们提出了基于时空的动态路由算法^[1],该算法的主要思想是通过每个网络节点对卫星运行网络的拓扑结构的自包含性,根据处理(转发)数据报的特定时刻,实时计算卫星网络的拓扑结构以及到达目的地卫星位置,在此基础上寻找最佳转发路由。

在提出基于时空的路由算法后,我们对设计一个好的路由算法的一般标准作了定性的评判,并得出该路由算法为一个适应卫星网络的较好的路由算法。为了对该路由算法提供更准确的评判,采用仿真的手段以提供定量的分析是必要的。本文首先介绍了我们提出的基于时空的路由算法思想,然后用 UML 对仿真系统进行建模,包括静态建模和动态建模两部分,最后实现了一个该算法的仿真系统 STSNS(Satellite Network Simulation based on Space Time)。STSNS 能对基于时空的路由算法进行了很好的仿真,仿真的结果表明,基于时空的路由算法能很好地满足卫星网路由的要求。

2 基于时空的动态路由算法描述

该路由算法之所称为基于时空的路由算法是因为,当网络节点转发某一特定的数据报时,路由的计算依赖于当时的特定时刻和网络节点当时所处的空间位置,即网络节点根据当时的时刻以及所处的拓扑位置来进行分布式计算以获得该

数据报的转发路由。

基于时空的路由算法主要可分为三步^[1]:

·**初始阶段** 在初始阶段,卫星网络中的各个网络节点(卫星)的初始时刻的主要信息,包括初始相位、所处轨道平面、标识等,向其它所有的网络节点进行发布。发布的方法可以是地面控制站统一向卫星进行广播发布,也可以通过单个卫星的分布式的广播发布来实现。经过初始的信息发布后,每个网络节点(卫星)对于整个的卫星网络的星座结构就有了认知,网络拓扑结构的改变就可以进行预测和计算。

·**计算候选路由** 在路由算法完成初始阶段之后就可以对候选路由进行计算了。当网络节点(卫星)取得数据报的目的出口节点(卫星)的地址后,便根据此刻的源卫星、目的卫星的位置,采用路径延时的度量标准(Metric),对最佳路径进行计算,并得到一个经过优劣排序的候选路径集合。对于候选路径的计算,主要是计算各种源、目节点的路径时延获得,其算法流程及伪代码的实现参见文^[1]。

·**选择转发路由** 前面已经说明该路由算法将主要采用路径延时和负载两个度量标准,在前阶段计算出候选路由集合之后,本阶段就是在这个已经优劣排序的候选路径集合中,根据链路的拥塞情况来选取一条最佳的路径,即从最优的候选路径开始,如果到该路径的下一条链路没有发生阻塞情况,则将该路径的下一跳作为该数据报的下一跳转发出去。如果没有有效的转发路径,则简单地将该数据报丢弃。由于卫星网络多实现为高速的宽带数据网络,网络极度拥塞的情况发生

^{*})863资助项目,项目号2002AA712034。孙知信 博士后,研究方向为计算机网络和安全,软件工程理论。王汝传 教授,博导,研究方向:计算机网络及安全。王绍棣 教授,博导,研究方向:模式识别。张登银 副教授,研究方向:计算机网络。李 军 硕士,研究方向:计算机网络与安全。

的概率将十分小,所以数据报的丢失率也将会很低。链路的拥塞程度可以用发送队列的长度来度量,即当队列长度大于某一阈值的时候便认为该链路拥塞。

3 STSNS 分析与设计

对于本仿真系统的需求分析,可以通过捕捉该仿真系统的用例(Use Case)来获得。所谓用例,简单地说就是从用户的角度上来描述其如何使用该系统的。用例可以采用文字进行描述,也可以通过图形来表示,UML^[2~5]就是一种用于面向对象的系统分析与设计的可视化建模语言,而 Rational Rose 就是一种运用 UML 进行开发的一种高效的工具。

对于这个卫星路由算法的仿真系统,从用户的观点来看,需要系统所作的主要工作有仿真设置和开始仿真两个操作,而仿真设置主要包括两个部分:一是对待仿真的卫星网络的星座模型的参数设置,有轨道参数的设置、卫星参数的设置、星间链路的设置以及星地链路的设置等;另一个就是对在卫星星座模型上进行仿真的数据模型的设置,主要有仿真时间的设置、通信终端的设置以及终端间的会话设置等。这个卫星网络路由算法仿真系统的需求可以用如图1所示的用例图来表示。

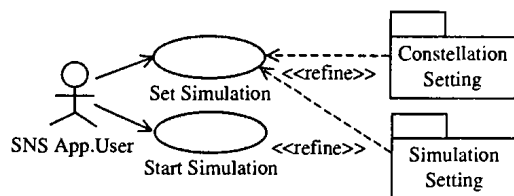


图1 STSNS 的高级用例图

图1是 STSNS 中设置星座参数的主要用例图。

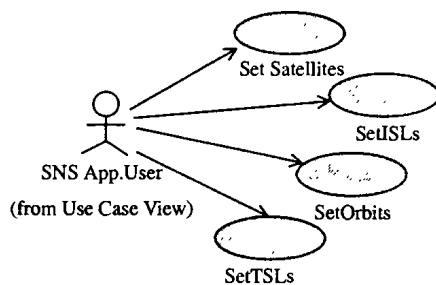


图2 设置星座的主要用例图

图3是 STSNS 设置一个仿真实例参数的主要用例图。

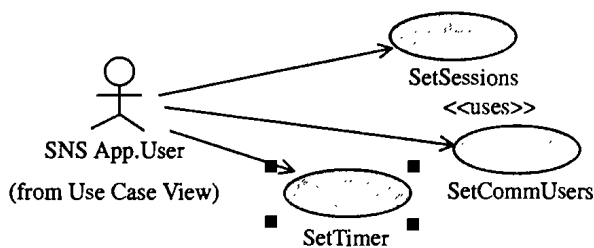


图3 设置仿真的主要用例图

通过捕捉用例获取该仿真系统的需求后,就可以对系统进行分析 and 设计了。在基于面向对象的软件开发思想中,主要的工作是如何从一个系统中抽取出一个个的对象(Object)或者说是类(Class),这些对象通过消息机制互相作用、互相协调,从而共同完成同一个系统功能目标。在 UML 中可以用类

图(Classes View)来表示一个系统中个各类之间的静态联系(或关系,Relations)。

在这个卫星路由算法仿真系统中,要对路由算法进行仿真,首先要建立起卫星网络的星座模型。通过分析不难得出组成一个卫星网络的主要对象有:星座网络、轨道、卫星、星间链路、星地链路、接受发送队列、通信终端等。除此之外,要进行一个仿真还要对一些仿真参数进行设置,所以还有一些仿真要用的对象有:定时器、通信会话、数据报、仿真对象等。在面向对象的系统中有两个重要的特性:继承和多态性。所谓继承就是所有的子类共同拥有基类的特性,而多态则是在继承的基础上,对于同一个消息,不同的子类可以有不同的行为方式。这两个特性在面向对象的系统中具有重要的作用,在这个卫星网络路由算法仿真系统的设计中就用到了这两个特性。由于一个被仿真系统中的所有对象都具有时间驱动性,而且希望所有的仿真对象具有唯一的表示,这样就可以使得所有的对象都派生于同一个基类(在该系统中是 SNS_Object 类)。而对于卫星、星间链路、星地链路和通信终端对数据报都具有处理功能,所以它们可以派生于同一个基类 SNS_Processor。该仿真系统的主要设计类图继承关系如图4所示。

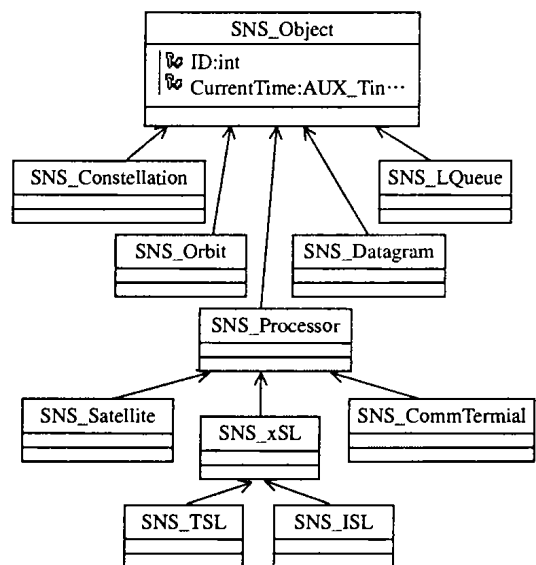


图4 STSNS 中的主要类继承关系

采用类图只能描述系统的静态结构,为了描述系统的动态行为,在 UML 中可以用状态机图、活动图、顺序图以及协作图等来表示,部分 STSNS 的 UML 设计图例如下。图5是仿真系统中用于数据报的传递的协作图。

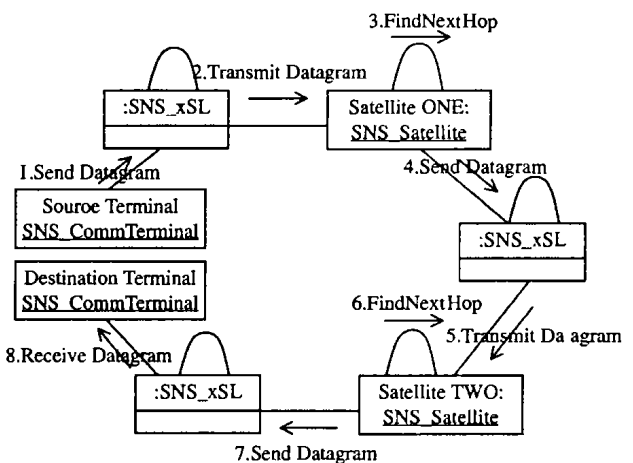


图5 STSNS 中用于数据报的传递的协作图

图6是仿真系统中的星座模型中的主要类图。

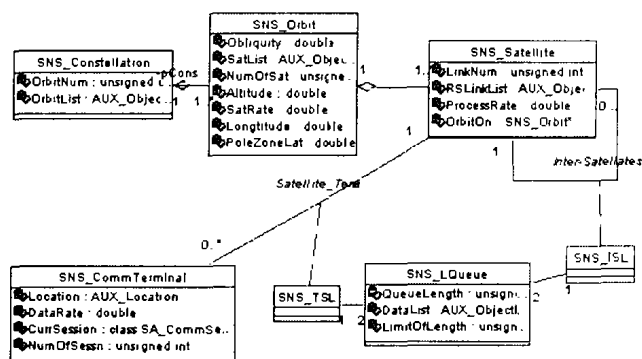


图6 星座模型中的主要类图

对于整个仿真应用的总体结构则可以由图7表示。

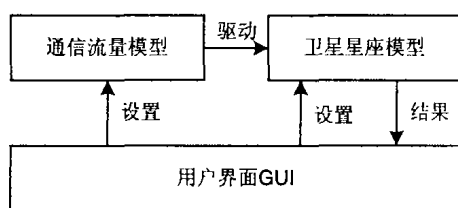


图7 STSNS 的主要框架结构

4 STSNS 的实现

在对卫星网络路由算法仿真系统进行了分析与设计之后,就可以对 STSNS 进行实现了。在统一开发过程(UDP)中,整个软件开发过程是一个迭代的增量过程,螺旋递进地周期性地重复分析、设计、编码、测试的流程直到软件的提交。由于该仿真系统的规模不是很大,经过几次迭代后就完成了。

由于 Microsoft Windows® 平台的友好性,以及 Visual C++®[6~8] 的高效性,所以该仿真系统采用 VC++ 实现。在前述对仿真系统的分析与设计的基础上,通过 Rational Rose 工具的代码产生功能,生成 Visual C++ 的代码框架,然后再用 VC++ 进行开发。在开发的过程中,可以随时采用逆向工程(Reverse Engineer)进行模型更新,以使设计与实现同步。完成后的仿真应用程序的运行界面如图8所示,和一个标准的 Windows 窗口应用程序风格一样。在应用程序界面的左边窗口是对仿真参数进行设置的地方,其中主要包括卫星星座、通信参数和仿真时间的设置等;在右边上面的小窗口则是对仿真结果的图形化的显示,下面的窗口则是仿真结果数据统计。

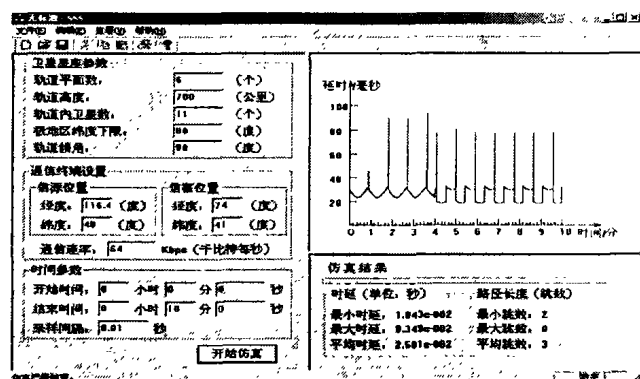


图8 SNS 的运行界面

在图8中所示的仿真中总时长是10分钟,从仿真结果看,大概每40~50秒就会有一次较大的时延,这是由于仿真中采

用的星座模型是铱星系统的模型,平均每45秒钟同一轨道的卫星就会切换一次,而由此造成了传输时延的增加。

5 仿真结果及分析

在仿真系统构建好后,就可以对基于时空的路由算法进行性能仿真了。在仿真试验中,对不同地点的终端之间的通信分别进行了24小时的连续仿真,其中两种的典型仿真结果如图9、图10所示。

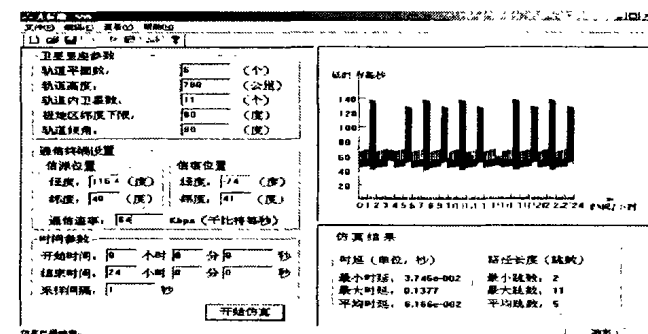


图9 仿真结果之一

图9中所示的是一种典型的仿真结果,在该仿真中通信终端之间的经度跨度较大,因此在地球一个自转周期中有相当一部分的时间段,两个通信终端是处于反向缝的不同侧的。在仿真结果的图中所示的突起部分的周期是2小时左右,这主要是由于轨道间的切换造成的。如图所示,仿真中的轨道数是6个,所以轨道相对地球转动的间隔时长为 $24/(6 \times 2) = 2$ 小时,和仿真结果较为吻合。对于仿真结果中3~4及15~16时间段,传输时延减小的主要原因是由于通信的源端和目的端所通信的卫星处于同一轨道平面中,因此它们之间在那个时间段的传输时延减小了。

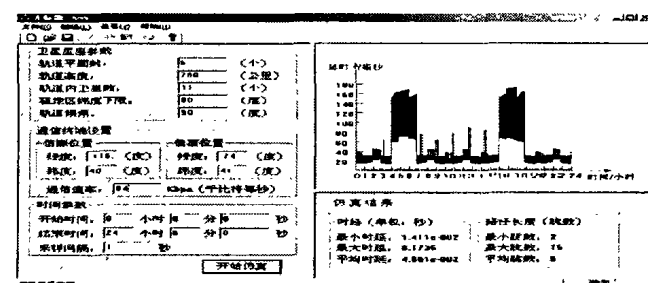


图10 仿真结果之二

图10中所示的是另一种典型的仿真结果,在该仿真中通信终端之间的经度跨度较小,因此在地球一个自转周期中大部分的时间段中,两个通信终端是处于反向缝的同一侧的,所以它们之间的时延主要在20~40ms 之间。对于图中的两处突起,是由于铱星系统不具有反向缝间链路,所以当反向缝处于两个通信终端之间的时候,数据包的传送必须穿越极地区域,这样就造成较大的传输时延,大概在60~160ms 之间。

从仿真结果来看,端到端的通信时延范围为20~170ms 之间,而且由于卫星的周期性运动,传输时延的变化呈现出周期性的变化。而数据报的路径长度跳数则在2~15之间,因此基于时空的路由算法在总体上能够提供较好的通信质量,就有较短的通信时延。

结论 为了对基于时空的路由算法进行定量的评价,本文采用了面向对象的软件工程思想对算法进行了仿真分析与

(下转第83页)

使得在正确性与索引结构的大小这两者的权衡上更加灵活。至于 Index Fabric 也能很好适应这一类查询语句,但是它有一个前提,就是基于 STORED^[6]的,因此效率要受到 XML 数据模型与关系数据库数据模型两者之间如何映射的影响。XISS^[10]能够很好处理正则表达式语句,但是这种将查询语句分解的方式对简单查询语句开销过大,因为比如 A/B 的处理,需要找出所有的 A 和所有的 B 然后进行两两匹配,随着路径长度的增加,这种处理方式的 IO 代价可能要比 Covering Index 大许多,因此 XISS 也只适应正则表达式的处理。当然, XISS 可以在 Covering Index 的基础上做索引,这样可以使得它的 IO 代价与 Covering Index 相差不太大。用数据库存放 XML 数据优点不言而喻,但其缺点也很明显,首先需要额外地增加格式转换,另外,两种数据模型之间不能有效找到一个好的映射,XPath 向 SQL 的转换算法不通用以及原始 XML 文档中的对象层次关系在关系数据库中难以维持都是有待研究的。

XISS^[10]能非常有效支持正则查询,但是 XISS 还有一个缺陷,就是它的 Numbering Schema 仅仅适应于树结构的 XML 文档,因此 XISS 不能适应图结构的数据对象。同时可以看到,XISS 易存放到关系型数据库中,虽然会降低效率,但是数据的维护会变得非常方便。

通过上面总结也可以看到当前 XML 索引结构研究的困境,今后 XML 索引结构的研究必须加强如下两方面:(1)研究适用范围更加广泛的索引结构,能更加全面地支持 XPath 查询语句;(2)目前的索引结构基本上都是静态的,今后应该加强动态索引结构的研究。

参考文献

- 1 Goldman R, Widom J. DataGuides: Enabling query formulation and optimization in semistructured database. In: Twenty-Third Intl. Conf. on Very Large Data Bases, 1997. 436~445
- 2 Milo T, Suciu D. Index structures for path expressions. In: ICDT: 7th Intl. Conf. on Database Theory, 1999
- 3 Kaushik R, Shenoy P, Bohannon P, Gudes E. Exploiting Local Similarity for Indexing Paths in Graph-Structured Data. In: Proc. of ICDE, 2002
- 4 Tatarinov I, et al. Storing and Querying Ordered XML Using a Relational Database System. In: Proc. of SIGMOD, 2002
- 5 Shanmugasundaram J, et al. A General Technique for Querying XML Documents using a Relational Database System. SIGMOD

- Record, Sep. 2001
- 6 Deutsch A, Fernandez M, Suciu D. Storing Semistructured Data with STORED. In: Proc. of SIGMOD Conf. 1999
- 7 Shanmugasundaram J, et al. Relational Databases for Querying XML Documents: Limitations and Opportunities. VLDB, 1999
- 8 Zhang C, et al. On Supporting Containment Queries in Relational Database Management Systems. In SIGMOD 2001
- 9 Knuth D. The Art of Computer Programming. Vol. III, Sorting and Searching. Third Edition. Addison Wesley, Reading, MA, 1998
- 10 Li Q, Moon B. Indexing and Querying XML Data for Regular Path Expressions. In: Proc. of VLDB, 2001
- 11 Shu C, Zografoula V, Zhang D, et al. Efficient Structural Joins on Indexed XML Documents. In: Proc. of VLDB, 2002
- 12 Gionis A, et al. XTRACT: A system for extracting document type descriptors from XML documents. In: Proc. of ACM-SIGMOD 2000 Intl. Conf. on Management of Data, 2000
- 13 Goldman R, Widom J. Approximate DataGuides. In: Proc. of the Workshop on Query Processing for Semistructured Data and Non-Standard Data Formats, Jan. 1999. 436~445
- 14 Nestorov S, Abiteboul S, Motwani R. Extracting schema from semistructured data. SIGMOD Record, 1998, 27(2): 295~305
- 15 Cooper B F, Sample N, Franklin M J, Hjalason G R, Shadmon M. A Fast Index for Semistructured Data. In: Proc. of VLDB, 2001
- 16 Fernandez M F, Morishima A, Suciu D. Efficient Evaluation of XML Middle-ware Queries. In SIGMOD, 2001
- 17 Shanmugasundaram J, et al. Efficiently Publishing Relational Data as XML Documents. In VLDB 2000
- 18 McHugh J, et al. Lore: A Database Management System for Semistructured Data: [Technical Report]. Stanford University Database Group, Feb. 1997
- 19 <http://www.w3.org/TR/XPPath20/>: XML Path Language (XPath) 2.0, W3C Working Draft 15 Nov. 2002
- 20 Papakonstantinou Y, Garcia-Molina H, Widom J. Object Exchange Across Heterogeneous Information Sources. In: Proc. of the Eleventh Intl. Conf. on Data Engineering, Taipei, Taiwan, 1995. 251~260
- 21 Milner R. A Calculus for Communicating Processes, volume 92 of Lecture Notes in Computer Science. Springer Verlag, 1980
- 22 Guttman A. R-trees: A Dynamic Index Structure for Spatial Searching. In: Proc. of SIGMOD, 1984
- 23 Chan C, Garofalakis M, Rastogi R. RE-Tree: An Efficient Index Structure for Regular Expressions. In: Proc. of the 28th VLDB conf. Hong Kong, China, 2002
- 24 Liechti O, Sifer M J, Ichikawa T. Structured graph format: XML metadata for describing web site structure. Computer Networks and ISDN Systems, 1998, 30:11~21

(上接第57页)

设计,并对最终实现的仿真系统 STSNS 作了全面的阐述。在仿真建模过程中,首先用用例(use case)对仿真系统的需求进行了分析,然后建立了 STSNS 的静态模型和动态模型,并用 UML 中相应的图例来描述。最后给出了 STSNS 的详细实现技术。然后,分别采用不同的通信模型对基于时空的路由算法进行了仿真,从仿真结果来看,该算法能够适应卫星网络的特点,对用户提供的通信质量,具有较短的时延。

参考文献

- 1 李军. 卫星网上的动态路由:[南京邮电学院硕士论文]. 2003, 3
- 2 Booch G, Rumbaugh J, Jacobson I. The Unified Modeling

- Language User Guide. Addison Wesley Longman Inc, 1999
- 3 Jacobson I, Booch G, Rumbaugh J. The Unified Software Development Process. Addison Wesley Longman Inc, 1999
- 4 Booch G, Jacobson I, Rumbaugh J. UML Notation Guide (Version 1.1). Rational Corporation, Santa Clara, 1997
- 5 Fowler, Martin, Kendall S. UML Distilled: Applying the Stand Object Modeling Language, Reading, MA: Addison-Wesley, 1997
- 6 Richter J. Programming Applications for Microsoft Windows, Fourth Edition, Microsoft Press, 1999
- 7 Echel B. Thinking in C++, Prentice Hall, 1995
- 8 MSDN Library-July 2001, Microsoft Corp., 2001