(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 109088741 A
(43)申请公布日 2018.12.25

(21)申请号 201710450435.X

(22)申请日 2017.06.14

(71)申请人 北京航空航天大学
地址 100191 北京市海淀区学院路37号

(72)发明人 胡凯 段张博 白晓敏 陈志东

(51)Int.Cl.
H04L 12/24(2006.01)
H04L 12/26(2006.01)

权利要求书1页 说明书6页 附图2页

(54)发明名称

一种区块链系统形式化建模与验证方法

(57)摘要

本发明提供了一种区块链系统形式化建模与验证方法,包括:步骤1:根据有限状态机利用分层建模的方式建立单个节点的SDL(Specification and Description Language,规范描述语言)模型;步骤2:根据单个节点的SDL模型,利用模块化建模的方式建立区块链系统的SDL模型;步骤3:对整个所述区块链系统的SDL模型进行仿真分析;步骤4:形式化验证所述区块链系统的SDL模型的性质。

CN 109088741 A

1.一种区块链系统形式化建模与验证方法,其特征在于:

步骤1:根据有限状态机利用分层建模的方式建立单个节点的SDL(Specification and Description Language,规范描述语言)模型;

步骤2:根据单个节点的SDL模型,利用模块化建模的方式建立区块链系统的SDL模型;

步骤3:对整个所述区块链系统的SDL模型进行仿真分析;

步骤4:形式化验证所述区块链系统的SDL模型的性质。

2.根据权利要求1所述的一种区块链系统形式化建模与验证方法,其特征在于:在步骤1之前,还包括:

对区块链系统进行环境以及功能分析;

建立区块链系统的有限状态机。

3.根据权利要求1所述的一种区块链系统形式化建模与验证方法,其特征在于:在步骤3中,使用Telelogic Tau中的Simulator(仿真器)对整个所述区块链系统的SDL模型进行仿真分析。

4.根据权利要求3所述的一种区块链系统形式化建模与验证方法,其特征在于:步骤3进一步包括:

(1)对所述区块链系统的SDL模型进行编译,执行“Microsoft RealTimeSimulation(微软实时仿真)”选项,在仿真器中打开编译后的可执行文件;

(2)使用仿真器中的MSC(Message Sequence Chart,消息序列图)功能,生成第一MSC图;

(3)针对区块链系统需求功能,向仿真器输入激励信号的命令脚本;

(4)执行上述命令脚本,产生第二MSC图;

(5)对所述第二MSC图进行分析,判断所述区块链系统的SDL模型是否实现了所述区块链系统需求功能,如果未实现进行调试,并再次仿真,直到所述区块链系统需求功能仿真通过。

5.根据权利要求1所述的一种区块链系统形式化建模与验证方法,其特征在于:在步骤4进一步包括:

使用Telelogic Tau中的Validator(验证器)来对所述区块链系统的SDL模型进行验证,验证的性质包括以下至少一项:可达性,无死锁,无活锁,弱活锁,时间相关活锁,有界性,可恢复性或自同步性,无状态二义性,互斥性,终止或进展,无冗余描述,公平性,完整性,一致性。

6.根据权利要求5所述的一种区块链系统形式化建模与验证方法,其特征在于:

使用所述Validator中的以下遍历算法之一来进行验证:Bit state exploration(位状态遍历),Random walk exploration(随机游走遍历),Tree search exploration(树遍历),Tree walk exploration(树游走遍历),Exhaustive exploration(详尽遍历)。

一种区块链系统形式化建模与验证方法

技术领域

[0001] 本发明涉及区块链验证领域，特别是涉及到区块链节点与网络的建模与验证方法。

背景技术

[0002] 自20世纪90年代以来，互联网的诞生和快速发展解决了信息制造和传输的基本问题。互联网连接了全球的各个国家和地区，让信息能快速进行传播和共享。通过信息互联网，我们进入一个“信息爆炸”的时代，整个互联网的信息呈指数式增长。近年来，随着社会各个行业在信息化和数字化的进程上持续推进，以及大数据技术在各行各业的全面应用和展开，数据也在各个行业的组织和实体之间频繁流动，其中不乏各种保密信息和包含隐私的信息，如果这些信息遭到泄露或者篡改，将会导致非常严重的问题。不仅如此，信息共享的各个利益方之间还存在不信任的问题，然而引入第三方和中心化的数据中心，又会导致信息流动效率低、信息易被人为篡改和易受攻击等新的问题。因此，社会信息化进程对保证信息的安全性和可信性的需求越来越迫切。另外，信息互联网仅解决了信息共享的问题，而不能支持价值转移——某一部分价值在网络中以人可以控制和确认的方式从一个地方转移到另一个地方，而不是复制，当前的互联网并不支持价值转移是许多行业(尤其是金融领域)亟需解决的问题。因此，构建一个安全可信的信息共享和价值转移的协议，能极大促进数字化社会的发展和大数据的全面应用。

[0003] 在这个背景下，区块链(Blockchain)技术逐渐进入人们的视野，并越来越受人推崇，被认为是“21世纪最为创新的应用之一”。区块链最早是在比特币(Bitcoin)系统广泛应用的，比特币是中本聪在2008年发明的一个P2P(Peer-to-peer)电子现金系统。比特币协议的目的是无需第三方的情况下解决电子货币中双重花费(Double-spending)问题。但是，由于比特币是一个自治系统，不为任何一个组织和个人所有，也很难做到监管，因此，各国政府对比特币多是采取打压的态度，比特币也一直被当做类似股票的价值承载物，没有大范围地展开应用。然而，随着近两年来，社会数字化进程的推进，区块链技术脱离了比特币应用的限制，再加上智能合约(Smart Contract)、DAO/DAC(Decentralized Autonomous Organization/Corporation)等概念的引入，区块链技术在各行各业遍地开花。不仅是金融、医疗、能源等领域，还是保险、供应链、法律、存证、版权、社会救助等行业，人们都在积极探索区块链的应用场景、业务模式和实现方式，因为区块链技术能对这些行业带来巨大的创新，甚至是颠覆性的变革。因此，区块链技术作为安全可信的信息共享和价值转移的完美解决方案，正在逐渐成为未来数字社会的基石。

[0004] 从表面上看，区块链是由数据区块(block)链接起来的数据链表，而本质上，区块链是一个去中心化的、公开的、自治的、防止篡改的分布式账本。将交易或数据加上时间戳，通过散列函数(Hash Function)进行散列，通过加密和签名技术保证数据安全和隐私，打包成一个个的数据区块，按时间顺序链接在一起，并在对等式(Peer-to-peer，P2P)网络进行传播，在各个peer节点进行冗余备份，通过共识算法达成全局一致并保证数据不可篡改。通

过这种机制，再加上智能合约这个巨大创新，支持了各种行业的众多应用场景，形成了一个具有公信力的、安全自治的数据库系统。

[0005] 因此，在如此多高安全性要求领域中的应用，使得区块链系统的安全性可信性研究变得十分重要。以太坊在之前一次严重的DAO攻击中损失了约合4500万美元价值的以太币。因此我们将引入形式化验证方法来提高区块链系统的安全性与可靠性。

[0006] 形式化方法是基于数学的特种技术，适合于软件和硬件的描述、开发和验证。将形式化方法用于软件和硬件设计，是期望能够像其它工程学科一样，使用适当的数学分析以提高设计的可靠性和鲁棒性。形式化方法的一个重要研究内容是形式规约(也称形式规范或形式化描述)，它是对程序“做什么”的数学描述，是用具有精确语义的形式语言书写的程序功能描述，它是设计和编制程序的出发点，也是验证程序是否正确的依据。对形式规约通常要讨论其一致性(自身无矛盾)和完备性(是否完全、无遗漏地刻画所要描述的对象)等性质。形式规约的方法主要可分为两类：一类是面向模型的方法也称为系统建模，该方法通过构造系统的计算模型来刻画系统的不同行为特征；另一类是面向性质的方法也称为性质描述，该方法通过定义系统必须满足的一些性质来描述一个系统。

[0007] 形式验证形式化方法的另一重要研究内容是形式验证。形式验证与形式规约之间具有紧密的联系，形式验证就是验证已有的程序(系统)P，是否满足其规约(φ,ψ)的要求(即P(φ,ψ))，它也是形式化方法所要解决的核心问题。传统的验证方法包括模拟和测试，它们都是通过实验的方法对系统进行查错。模拟和测试分别在系统抽象模型和实际系统上进行，一般的方法是在系统的某点给予输入，观察在另一点的输出，这些方法花费很大，而且由于实验所能涵盖的系统行为有限，很难找出所有潜在的错误。基于此，形式验证主要研究如何使用数学方法，严格证明一个程序的正确性(即程序验证)。

[0008] 从某种意义上来说，区块链网络可以被认为是一个高可信度的分布式通信网络。因此，引入形式化方法可以有效地提高用户对区块链系统的安全性的信任度。我们引入了一种基于SDL(Specification and Description Language，规范描述语言)的分层和模块化建模方法来形式化建模区块链系统。通过分层建模来对单个节点进行建模，而通过模块化建模来对整个区块链网络体系进行建模。在此基础上，可以对整个区块链系统进行仿真与形式化验证，以找出系统的逻辑错误。该模式有助于提高开发自动化水平，缩短开发时间，降低编码过程中出错的可能性。传统的区块链实验通常需要部署大量的计算机来满足实验条件。通过区块链形式建模方法，可以在单个计算机上实现多节点块链模拟。此外，我们可以通过建立区块链模型，在给定的区块链网络上研究不同共识算法的一些性质。同时还可以有利于未来区块链标准的制定。

发明内容

[0009] 有鉴于此，本发明引入形式化验证方法来提高区块链系统的安全性与可靠性。

[0010] 一种区块链系统形式化建模与验证方法，包括：

[0011] 步骤1：根据有限状态机利用分层建模的方式建立单个节点的SDL(Specification and Description Language，规范描述语言)模型；

[0012] 步骤2：根据单个节点的SDL模型，利用模块化建模的方式建立区块链系统的SDL模型；

[0013] 步骤3:对整个所述区块链系统的SDL模型进行仿真分析;

[0014] 步骤4:形式化验证所述区块链系统的SDL模型的性质。

[0015] 优选的,在步骤1之前,还包括:

[0016] 对区块链系统进行环境以及功能分析;

[0017] 建立区块链系统的有限状态机。

[0018] 优选的,在步骤3中,使用Telelogic Tau中的Simulator(仿真器)对整个所述区块链系统的SDL模型进行仿真分析。

[0019] 优选的,步骤3进一步包括:

[0020] (1)对所述区块链系统的SDL模型进行编译,执行“Microsoft RealTimeSimulation(微软实时仿真)”选项,在仿真器中打开编译后的可执行文件;

[0021] (2)使用仿真器中的MSC(Message Sequence Chart,消息序列图)功能,生成第一MSC图;

[0022] (3)针对区块链系统需求功能,向仿真器输入激励信号的命令脚本;

[0023] (4)执行上述命令脚本,产生第二MSC图;

[0024] (5)对所述第二MSC图进行分析,判断所述区块链系统的SDL模型是否实现了所述区块链系统需求功能,如果未实现进行调试,并再次仿真,直到所述区块链系统需求功能仿真通过。

[0025] 优选的,在步骤4进一步包括:

[0026] 使用Telelogic Tau中的Validator(验证器)来对所述区块链系统的SDL模型进行验证,验证的性质包括以下至少一项:可达性,无死锁,无活锁,弱活锁,时间相关活锁,有界性,可恢复性或自同步性,无状态二义性,互斥性,终止或进展,无冗余描述,公平性,完整性,一致性。

[0027] 优选的,使用所述Validator中的以下遍历算法之一来进行验证:Bit state exploration(位状态遍历),Random walk exploration(随机游走遍历),Tree search exploration(树遍历),Tree walk exploration(树游走遍历),Exhaustive exploration(详尽遍历)。

附图说明

[0028] 为了更清楚地说明本发明实施例的技术方案,下面将对实施例描述中所需要的附图做简单的介绍,显而易见地,下面描述的附图仅仅是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0029] 图1为本发明的区块链系统形式化建模与验证方法流程;

[0030] 图2为区块链分层架构;

[0031] 图3为共识层SDL模型的一个模型页;

[0032] 图4为一种简单的区块链网络拓扑结构模型。

具体实施方式

[0033] 参考图1,一种区块链系统形式化建模与验证方法,其主要包括以下步骤:

[0034] ●对现有区块链系统进行环境以及功能分析;

[0035] ●建立区块链系统的有限状态机；

[0036] ●根据有限状态机利用分层建模的方式建立单个节点的SDL模型；

[0037] ●根据单个节点模型利用模块化建模的方式建立区块链网络的SDL模型；

[0038] ●对整个区块链系统进行仿真分析；

[0039] ●形式化验证区块链系统模型的性质。

[0040] 区块链系统建模

[0041] 通过区块链系统的形式化规格说明文档，对区块链系统进行模型建立。需要选择一种建模工具和相应的建模语言。在此使用SDL语言，按照上述生成的形式化规格说明文档，对区块链系统进行描述，即完成建模的过程。

[0042] 建模分为两个过程：

[0043] 1、针对SDL语言的特点，结合区块链单个节点系统的属性，提出了基于SDL的区块链分层建模方法，对区块链系统的单个节点进行形式化建模。对于一个单个节点的区块链系统来说，可以由如图2所示的分层架构，对其数据层，网络层，共识层，智能合约层和应用层进行分层建模。

[0044] 数据层负责区块生成和区块链构造、存储。数据层将公开或加密的交易或数据通过哈希函数散列成一棵Merkle树，然后将树根和时间戳以及工作量证明的结果(如果采用PoW共识机制)等字段封装成一个区块，并通过存储前一个区块的哈希值将新生成的区块链接到区块链上。由于数据区块是根据前一个区块的哈希来进行链接的，与传统指针链接相比更安全、可靠。

[0045] 网络层一般是P2P网络，对于私有链来说也可以是一个传统的网络。网络层需具有单播、广播以及组播等功能，并能对数据包进行过滤，同时完成授权或公开的通信节点加入、删除等功能。

[0046] 共识层是区块链模型最核心的层次之一，定义了区块链模型的规则，规定了各个节点之间如何达成一致，如何备份，如何容错，如何做到数据一致性、分区容错性和可用性之间的平衡，公有链比较常用的有PoW(Proof of Work，工作量证明)、PoS(Proof of Stake，权益证明)和DPoS(Delegate Proof of Stake，股权授权证明机制)等，联盟链和私有链一般采用拜占庭容错算法和传统的一致性算法作为其共识，比较常用的共识算法有PBFT(Practical Byzantine Fault Tolerance，拜占庭容错算法)、Paxos、Raft等。

[0047] 智能合约层实现智能合约的编写、编译、运行等机制。包括智能合约语言的规则，智能合约脚本如何编译成指令代码，如何将智能合约指令代码在智能合约虚拟机中安全隔离的运行。

[0048] 应用层则是上层的各种应用，如第一代区块链——电子货币系统，第二代区块链——可编程金融、医疗、供应链等各行各业的应用，以及未来的第三代区块链——可编程社会。

[0049] 其中共识层模型的一个SDL模型页如图3所示。

[0050] 2、在单个节点形式化模型的基础上，对整个区块链网络进行模块化建模，我们可以对各种区块链网络拓扑结构进行验证并分析其特点，一种简单的区块链网络拓扑结构的SDL模型如图4所示。

[0051] 区块链系统实时仿真

[0052] Telelogic Tau(一种SDL工具)中的Simulator仿真器可以对建立好的区块链模型进行仿真。仿真包括两种情况:实时仿真和非实时仿真,区块链具有时效性需要采用实时仿真。实时仿真中,仿真系统中的时间流逝和实际环境保持一致。在采用实时仿真时需要在编译区块链模型时,设置参数为“Microsoft RealTimeSimulation”。

[0053] 在仿真过程中,可以通过仿真器输入各种信号,然后运行;运行的同时可以产生MSC图,MSC可以展现区块链模型的状态转移、信号输入和输出,通过对MSC的观测和分析,可以了解区块链模型在输入激励信号的下的运行情况,判断区块链模型是否实现了相应的功能。

[0054] 因此,对区块链模型进行实时仿真的步骤如下:

[0055] (1)编译建立好的区块链模型,选项为“Microsoft RealTimeSimulation”,在仿真器中打开编译好的可执行文件。

[0056] (2)打开仿真器中的MSC功能,以便在仿真时生成第一MSC图。

[0057] (3)针对某一协议功能,编辑仿真需要输入的激励信号的命令脚本。

[0058] (4)开始仿真,执行命令脚本,产生第二MSC图。

[0059] (5)对第二MSC图进行分析,判断区块链模型是否实现了区块链系统需求功能,如果未实现进行调试,并再次进行仿真,直到该功能仿真通过。

[0060] 区块链系统的模型验证

[0061] 我们将使用SDL工具Telelogic Tau中的Validator(验证器)来对区块链模型进行验证工作,主要验证以下性质:

[0062] (1)区块链系统的一般性质:

[0063] ●可达性(Reachability)。验证区块链系统的各种可能状态之间的可达关系。如果区块链系统的某个状态从初态不可达,则表明区块链系统有错误。如果从状态A到状态B的变迁不可能发生(直接或间接),则从状态A到状态B是不可达的。

[0064] ●没有死锁(Deadlock freeness)。最典型的死锁是区块链系统中各实体都处于这样的一种等待状态,即只有在“某一事件”发生后才能做进一步的动作,但在该状态下,这个“某一事件”却不可能发生。死锁发生时,区块链系统所处的状态称为死锁状态。

[0065] ●没有活锁(Livelock freeness)。活锁是指区块链系统处于无限的死循环中,而没有别的事件可使区块链系统从这一循环中解脱出来。例如,区块链系统无限制地执行超时重发操作,但总是收不到对方的确认信息。状态还是在变化的,不过不能脱离这种死循环状态而已。

[0066] ●弱活锁(Weak livelock)。是指区块链系统处于死循环中,只有当区块链系统交换命令的相对速度达到某一状态时,区块链系统才退出死循环。

[0067] ●时间相关的活锁(Time-dependent livelock)。也称为临时阻塞(Tempo-blocking)。它是指区块链系统处于死循环中,但是当通信双方交换报文的相对速度到达某一状态时,区块链系统可以从死循环中出来。

[0068] ●有界性(Boundedness)。检验区块链系统的某些成分或参数的容量(例如:通道容量、窗口大小)是否有界。有界性是针对区块链系统元素性质和通道性质而言。

[0069] ●可恢复性或自同步性(Recovery from failures)。这是当出现差错后,区块链系统能否在有限的步骤内返回到正常状态(包括初始态)执行。

[0070] ●无状态二义性(State ambiguities freeness)。一个进程在某一时刻只允许具有一个稳定状态。所谓稳定状态是指当通信双方的通道为空时的进程状态。若在某一时刻进程可以有多个稳定状态,则称该进程的状态为二义状态。

[0071] ●互斥性(Mutual exclusion)。互斥性是指有些区块链系统动作不能同时被多个用户执行。例如,多个用户不能同时请求同一资源。

[0072] ●终止或进展(Termination or Progress)。是指区块链系统提供的服务必须在有限时间内完成。终止是针对终止区块链系统(terminating protocols)而言,意思是指区块链系统总是能到达期望的结束状态。而进展则是针对循环区块链系统(cyclic protocols)而言,意思是指区块链系统总是能到达它的初始状态。

[0073] ●无冗余描述(Absence of Overspecification)。区块链系统规范中没有无用的、冗余描述,例如,没有未经实践的报文接收(absence of unexercised message receptions)。

[0074] ●公平性(Fairness)。是指每一个区块链系统实体均应平等地得到运行的机会,无论其它的区块链系统实体想做什么。

[0075] (2)区块链系统的特殊性质:

[0076] ●完整性(Completeness)。是指区块链系统设计考虑了所有可能发生的事件、选项以及服务。检验区块链系统是否能处理所有可能的输入,即是否缺少应用的处理,或有无非期待的接收或输入(即错收)。

[0077] ●一致性(consistency)。是指区块链系统服务行为(或性质)和区块链系统行为(或性质)一致,即区块链系统应该提供用户要求的服务,而无需提供用户没有要求的服务。

[0078] 同时,使用Validator中的几种验证算法Bit state exploration(位状态遍历),Random walk exploration(随机游走遍历),Tree search exploration(树遍历),Tree walk exploration(树游走遍历),Exhaustive exploration(详尽遍历)。来进行验证工作。这几种算法各有优劣各有不同的侧重点,因此需要选择合适的算法。对于区块链系统的验证来说,我们一般选择Bit state exploration。

[0079] 以上所述,仅是本发明的实例,并非对本发明做任何形式上的限制。任何精于本专业的技术人员,在不脱离本发明技术方案范围内,当可利用上述揭示的技术内容做出其他种种的改良或修饰为等同变化的等效实例,但凡是未脱离本发明技术方案内容,依据本发明的技术实质对以上实施所做的任何简单修改、等同变化与修饰,均仍属于本发明技术方案的范围内。

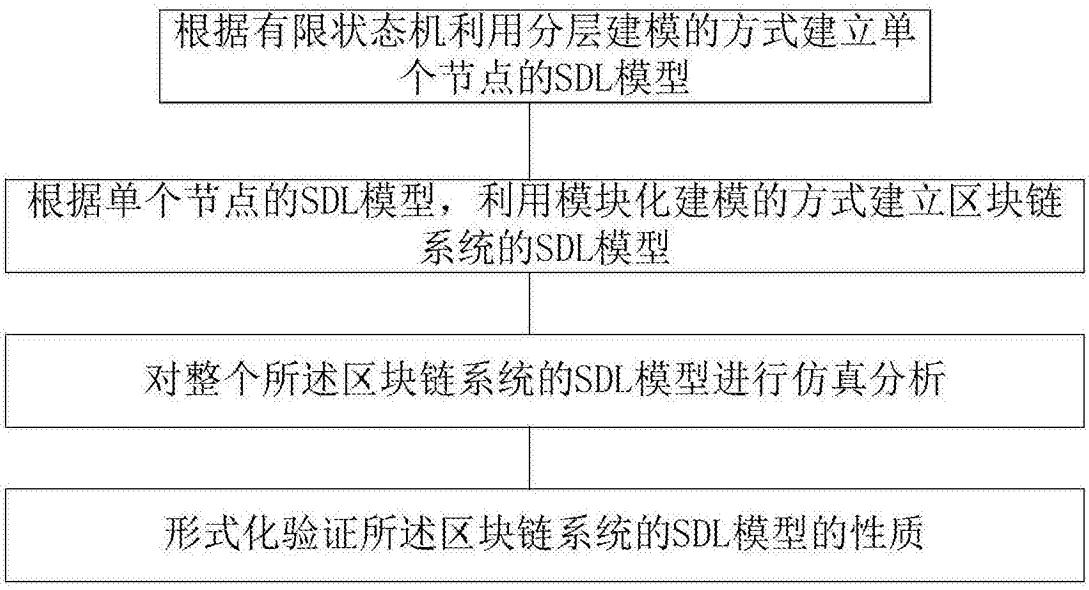
根据有限状态机利用分层建模的方式建立单个节点的SDL模型
根据单个节点的SDL模型，利用模块化建模的方式建立区块链系统的SDL模型
对整个所述区块链系统的SDL模型进行仿真分析
形式化验证所述区块链系统的SDL模型的性质

图1


可编程货币
可编程金融
应用层
智能合约脚本
调用
运行
智能合约层
PoW
PoS
DPoS
……
共识层
P2P网络
传播
验证
网络层
区块体
链结构
时间戳
哈希函数
Merkle树
非对称加密
数据层
区块链系统

图2

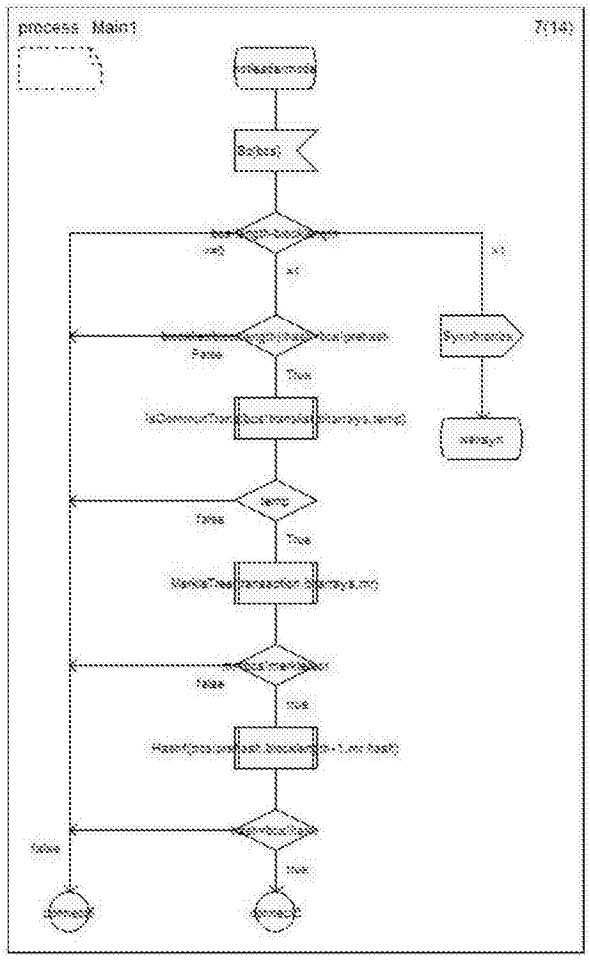
process Main1
Synchronize
False
True
False
True
False
True

图3

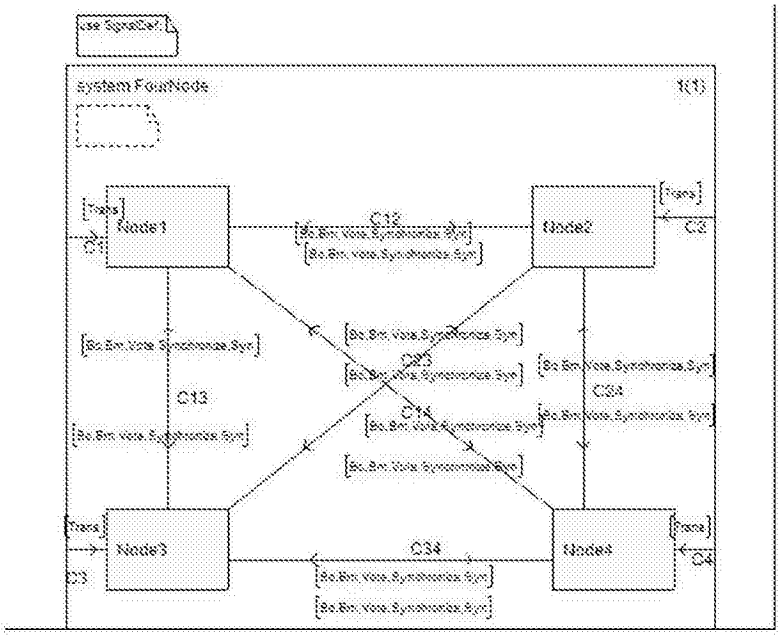
system FourNode
Node1
Node2
Node3
Node4
C12
C13
C14
C23
C24
C34

图4