



(12) 发明专利

(10) 授权公告号 CN 116991373 B

(45) 授权公告日 2024. 01. 05

(21) 申请号 202311253139.2

(22) 申请日 2023.09.27

(65) 同一申请的已公布的文献号
申请公布号 CN 116991373 A

(43) 申请公布日 2023.11.03

(73) 专利权人 杭州华望系统科技有限公司
地址 310030 浙江省杭州市西湖区三墩镇
西园八路3号智汇众创中心E1幢1303-4室

(72) 发明人 刘玉生 熊永琦

(74) 专利代理机构 杭州天勤知识产权代理有限公司 33224
专利代理师 曹兆霞

(51) Int. Cl.
G06F 8/20 (2018.01)

(56) 对比文件

CN 106648617 A, 2017.05.10

CN 101013384 A, 2007.08.08

CN 110175019 A, 2019.08.27

CN 103885816 A, 2014.06.25

CN 116414376 A, 2023.07.11

CN 103618732 A, 2014.03.05

US 9256485 B1, 2016.02.09

US 2015100942 A1, 2015.04.09

D.R. Hild 等.DEVS-DOC: a modeling and simulation environment enabling distributed codesign.IEEE Transactions on Systems, Man, and Cybernetics .2002,全文.
李景峰,李琰,陈平.基于事件序的UML序列图语义规范.长安大学学报(自然科学版).2002,(06),全文.

审查员 任燕

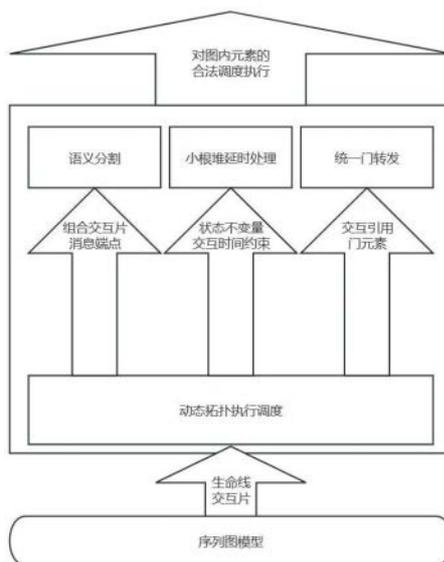
权利要求书2页 说明书7页 附图3页

(54) 发明名称

一种序列图模型的语义解析与仿真系统

(57) 摘要

本发明公开了一种序列图模型的语义解析与仿真系统,包括动态拓扑执行调度模块和基于语义分割的消息同步等待模块,以部分有序的偏序关系形成动态拓扑执行队列作为序列图模型的执行参考,进而实现UML规范对序列图模型执行弱序的约束,避免了部分情况下严格有序时的非必要前序元素的重复执行导致的后续序列饿死的问题。另一方面,本发明基于语义分割的消息同步等待模型实现了包括弱序、循环和并行的操作符在内的更完善的组合交互片元素语义,并实现了对交互图中门元素语义的支持,从而使得用户可以更准确、灵活的实现预期的模型语义。



1. 一种序列图模型的语义解析与仿真系统,其特征在於,包括:

动态拓扑执行调度模块,其用于将执行主体分解到序列图模型的交互片上,并通过动态拓扑执行队列管理序列图模型的交互片,依据动态拓扑执行队列执行任务;其中,每个序列图模型的交互片上设有交互片仿真类,该交互片仿真类存储有交互片的偏序关系和执行状态;所述偏序关系通过前序交互片集合和后序交互片集合实现,依据生命线上两交互片之间偏序关系,将前一交互片加入到后一交互片的前序交互片集合,将将后一交互片加入到前一交互片的后序交互片集合;所述执行状态被保存为第一布尔值和第二布尔值,第一布尔值表明交互片是正常等待执行或不需要再执行,第二布尔值表明交互片是否完成了执行;依据动态拓扑执行队列执行任务包括:当任意的交互片执行完毕后,将执行完毕交互片的后序交互片集合中的所有交互片添加到动态拓扑执行队列中;当组合交互片执行时,将其内部需要执行的交互片加入到动态拓扑执行队列中;对于组合交互片的循环语义,在所有内部交互片执行完且需要循环的情况下,重置内部交互片的执行状态后,将内部交互片重新加入到动态拓扑执行队列中以实现内部语义的循环执行;

基于语义分割的消息同步等待模块,其用于将交互片的执行语义分割为负责主体执行逻辑的主体语义和负责序列图模型执行次序相关执行内容的结束语义,基于主体语义和结束语义实现同步消息的发送交互片的等待行为,基于主体语义和结束语义实现组合交互片的整体执行;

其中,基于主体语义和结束语义实现同步消息的发送交互片的等待行为,包括:针对同步消息的发送交互片,其主体语义为对消息参数进行取值,并将取到的参数值发送给其对应的接收交互片,结束语义是调起后序交互片,发送交互片执行完主体语义后从动态拓扑执行队列中退出并进入等待状态;接收交互片完成执行后通知发送交互片结束等待,将发送交互片以结束语义加入到动态拓扑执行队列中,动态拓扑执行队列再次执行发送交互片且设置完成状态,并调起后序交互片;

基于主体语义和结束语义实现组合交互片的整体执行,包括:针对组合交互片,其主体语义为将所有需要执行的内部交互片添加到动态拓扑执行队列,并将所有不需要执行的内部交互片的状态设置为不需要再执行;为组合交互片的内部交互片的执行元素设置外部包裹交互片字段,实现在内部交互片的执行语义中向外部包裹交互片通知内部交互片的执行已经完成,这种通知是通过将内部交互片的结束语义添加到动态拓扑执行队列实现的;当外部交互片发现内部交互片的执行状态全部为完成执行时,则开始执行外部交互片的结束语义以调起组合交互片的后序交互片。

2. 根据权利要求1所述的序列图模型的语义解析与仿真系统,其特征在於,还包括基于小根堆的延时事件处理模块,其用于通过以超时时间戳为排序依据的延时事件小根堆来实现对延时事件的处理,使得能够处理到序列图模型中的时间约束语义。

3. 根据权利要求2所述的序列图模型的语义解析与仿真系统,其特征在於,所述基于小根堆的延时事件处理模块中,当一个时间约束开始等待时,计算超时时间戳并根据超时时间戳将延时事件插入小根堆中,保证小根堆堆顶始终保存着当前系统中最早到期的时间约束;

在每次从动态拓扑执行队列中取出交互片执行前,检查小根堆并当小根堆不为空且顶部时间约束已经超时,弹出超时的时间约束并将其约束的交互片的主体语义加入到动态

拓扑执行队列中；

当动态拓扑执行队列已经为空时,根据小根堆顶部的超时时间命令当前线程等待到目标时间为止。

4.根据权利要求1所述的序列图模型的语义解析与仿真系统,其特征在于,还包括基于统一门的门消息转发模块,其用于将各类门元素的转发语义抽离成统一门,基于统一门实现消息转发。

5.根据权利要求4所述的序列图模型的语义解析与仿真系统,其特征在于,所述基于统一门的门消息转发模块中,统一门包括两个能够配对且负责消息请求和参数转发的门元素；

为每个门元素维持一个参数暂存区和对配对的门元素的引用,当一个门元素作为消息接收端时,会从消息中读取参数和请求信息并转发到配对的门元素；当一个门元素作为消息的发送端时,会等待配对门元素的请求信息到达,并将请求信息和参数进行转发。

一种序列图模型的语义解析与仿真系统

技术领域

[0001] 本发明属于模型驱动设计技术领域,具体涉及一种序列图模型的语义解析与仿真系统。

背景技术

[0002] 系统建模语言(System Modeling Language, SysML)是一种专门为系统工程应用开发的标准建模语言,它是由对UML2.0的子集进行重用和扩展而来的。在语言结构方面, SysML是由图和元模型组成,图是语法,元模型是语义。

[0003] 在模型驱动设计技术领域中,模型仿真是建模者验证自己模型是否符合设计预期,以及检查工程方案能否满足模型约束的重要手段。在对象管理组织(Object Management Group)的官方文档中,对SysML行为元素中活动所对应的活动图、状态机所对应的状态机图都给出了执行语义的规范和实现样例。基于这些规则和实现样例,业界针对活动图和状态机图这两种图的仿真有成熟且完善的解决方案支持,例如Cameo模拟工具包(Cameo Simulation Toolkit,简称CST)。

[0004] 但是,对于同属行为类型的特化类型的交互以及其所对应的序列图,对象管理组织官方并没有给出除了UML规范之外的更多说明和实现样例。同时,业界现有的解决方案对交互和序列图的支持不仅很有限,在很多时候也不贴合现有规范对序列图语义的约束。例如CST直接以序列图内元素的垂直次序作为元素仿真时的执行次序,不完全满足规范内定义的部分有序。在部分情况下,CST的解决方案将无法按照建模者的意图进行仿真执行次序安排。同时,CST只支持了序列图中的部分元素,在这部分元素中的语义实现也未能完全依照规范进行,并可能导致序列图执行过程互相阻塞,进而无法正确按建模者意图执行仿真过程和验证模型中可能存在和暴露出来的系统问题。

[0005] 序列图作为描述系统中组件间信息流和事件发生顺序的图模型,常用于识别系统中信息组件和事件时序上可能存在的瓶颈和矛盾,最早常见于软件工程和其他需要描述组件间通信和系统内事件时序的领域。但随着系统建模语言被大量地用于工程设计过程,以及计算通信组件在工程领域迅速增长的应用场景,基于模型的系统工程也需要通过对应的图模型来描述这些组件在工程中的通信过程、和事件时序模型。因而承载此类功能的序列图在基于模型的系统工程中正变得愈发重要,对于序列图准确仿真的需求也日益增多。然而业界并无一套成熟、准确的方案来执行序列图的仿真过程,亟需一套准确、规范的仿真系统来处理对序列图仿真的迫切需要。

发明内容

[0006] 鉴于上述,本发明的目的是提供一种序列图模型的语义解析与仿真系统,该系统面向UML规范,提供更准确且完善的序列图模型仿真支持。

[0007] 为实现上述发明目的,实施例提供的一种序列图模型的语义解析与仿真系统,包括:

[0008] 动态拓扑执行调度模块,其用于将执行主体分解到序列图模型的交互片上,并通过动态拓扑执行队列管理序列图模型的交互片,并依据动态拓扑执行队列执行任务;

[0009] 基于语义分割的消息同步等待模块,其用于将交互片的执行语义分割为负责主体执行逻辑的主体语义和负责序列图模型执行次序相关执行内容的结束语义,基于主体语义和结束语义实现同步消息的发送交互片的等待行为、组合交互片的整体执行。

[0010] 优选地,所述动态拓扑执行调度模块中,每个序列图模型的交互片上设有交互片仿真类,该交互片仿真类存储有交互片的偏序关系和执行状态;

[0011] 所述偏序关系通过前序交互片集合和后序交互片集合实现,依据生命线上两交互片之间偏序关系,将前一交互片加入到后一交互片的前序交互片集合,将后一交互片加入到前一交互片的后序交互片集合;

[0012] 所述执行状态被保存为第一布尔值和第二布尔值,第一布尔值标明交互片是正常等待执行或不需要再执行,第二布尔值标明交互片是否完成了执行。

[0013] 优选地,所述动态拓扑执行调度模块中,当任意的交互片执行完毕后,将执行完毕交互片的后序交互片集合中的所有交互片添加到动态拓扑执行队列中。

[0014] 优选地,所述动态拓扑执行调度模块中,当组合交互片执行时,将其内部需要执行的交互片加入到动态拓扑执行队列中;

[0015] 对于组合交互片的循环语义,在所有内部交互片执行完且需要循环的情况下,重置内部交互片的执行状态后,将内部交互片重新加入到动态拓扑执行队列中以实现内部语义的循环执行。

[0016] 优选地,所述基于语义分割的消息同步等待模块中,基于主体语义和结束语义实现同步消息的发送交互片的等待行为,包括:

[0017] 针对同步消息的发送交互片,其主体语义为对消息参数进行取值,并将取到的参数值发送给其对应的接收交互片,结束语义是调起后序交互片,发送交互片执行完主体语义后从动态拓扑执行队列中退出并进入等待状态;接收交互片完成执行后通知发送交互片结束等待,将发送交互片以结束语义加入到动态拓扑执行队列中,动态拓扑执行队列再次执行发送交互片且设置完成状态,并调起后序交互片。

[0018] 优选地,所述基于语义分割的消息同步等待模块中,基于主体语义和结束语义实现组合交互片的整体执行,包括:

[0019] 针对组合交互片,其主体语义为将所有需要执行的内部交互片添加到动态拓扑执行队列,并将所有不需要执行的内部交互片的状态设置为不需要再执行;

[0020] 为组合交互片的内部交互片的执行元素设置外部包裹交互片字段,实现在内部交互片的执行语义中向外部包裹交互片通知内部交互片的执行已经完成,这种通知是通过将内部交互片的结束语义添加到动态拓扑执行队列实现的;当外部交互片发现内部交互片的执行状态全部为完成执行时,则开始执行外部交互片的结束语义以调起组合交互片的后序交互片。

[0021] 优选地,还包括基于小根堆的延时事件处理模块,其用于通过以超时时间戳为排序依据的延时事件小根堆来实现对延时事件的处理,使得能够处理到序列图模型中的时间约束语义。

[0022] 优选地,所述基于小根堆的延时事件处理模块中,当一个时间约束开始等待时,计

算超时时间戳并根据超时时间戳将延时事件插入小根堆中,保证小根堆堆顶始终保存着当前系统中最早到期的时间约束;

[0023] 在每次从动态拓扑执行队列中取出交互片执行前,检查小根堆并当小根堆不为空且顶部时间约束已经超时,弹出超时的时间约束并将其约束的交互片的主体语义加入到动态拓扑执行队列中;

[0024] 当动态拓扑执行队列已经为空时,根据小根堆顶部的超时时间命令当前线程等待到目标时间为止。

[0025] 优选地,所述系统还包括基于统一门的门消息转发模块,其用于将各类门元素的转发语义抽离成统一门,基于统一门实现消息转发。

[0026] 优选地,所述基于统一门的门消息转发模块中,统一门包括两个能够配对且负责消息请求和参数转发的门元素;

[0027] 为每个门元素维持一个参数暂存区和对配对的门元素的引用,当一个门元素作为消息接收端时,会从消息中读取参数和请求信息并转发到配对的门元素;当一个门元素作为消息的发送端时,会等待配对门元素的请求信息到达,并将请求信息和参数进行转发。

[0028] 与现有技术相比,本发明具有的有益效果至少包括:

[0029] 本发明以部分有序的偏序关系形成动态拓扑执行队列作为序列图模型的执行参考,进而实现UML规范对序列图模型执行弱序的约束,避免了部分情况下严格有序时的非必要前序元素的重复执行导致的后续序列饿死的问题。另一方面,本发明基于语义分割的消息同步等待模型实现了包括弱序、循环和并行的操作符在内的更完善的组合交互片元素语义,并实现了对交互图中门元素语义的支持,从而使得用户可以更准确、灵活的实现预期的模型语义。

附图说明

[0030] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图做简单地介绍,显而易见地,下面描述中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前提下,还可以根据这些附图获得其他附图。

[0031] 图1是实施例提供的序列图模型的语义解析与仿真系统的结构示意及流程图;

[0032] 图2是实施例提供的有语义分割时的动态拓扑执行调度过程;

[0033] 图3是实施例提供的没有语义分割时的同步消息死锁问题示意;

[0034] 图4是实施例提供的通过语义分割解决同步消息死锁的示意图;

[0035] 图5是实施例提供的在有语义分割的动态拓扑调度过程中组合交互片调度过程。

具体实施方式

[0036] 为使本发明的目的、技术方案及优点更加清楚明白,以下结合附图及实施例对本发明进行进一步的详细说明。应当理解,此处所描述的具体实施方式仅仅用以解释本发明,并不限定本发明的保护范围。

[0037] 本发明的发明构思为:首先提出了语义分割概念,即将单个序列图模型中交互片的执行语义分割成主体语义和结束语义,这样使得能让单一交互片的语义分段执行,从而

实现同步消息对交互片所调用操作的等待行为。之后,为了使得以组合交互为代表的序列图控制逻辑可以动态地对执行次序做出修正,使用动态拓扑执行队列替代朴素拓扑排序来作为执行排序算法,这样的动态拓扑使得组合交互片可以基于守卫元素实现对其内部元素执行路径的动态控制,并能严格遵守规范对弱序执行的定义。在此之上,为了在单线程下实现对时间约束的高效支持,实现了基于小根堆的延时事件序列,从而提供了对时间约束和状态不变量动态检查的支持。最后,为了支持门语义,提出统一门来实现,通过将任意门的行为抽象为参数的转发,使得能够用同一个门的算法定义处理组合交互和交互使用上的两种不同门的行为。

[0038] 基于以上发明构思,实施例提供的序列图模型的语义解析与仿真系统,如图1所示,语义解析与仿真系统包括动态拓扑执行调度模块、基于语义分割的消息同步等待模块、基于小根堆的延时事件处理模块以及基于统一门的门消息转发模块。下面针对每个模块进行详细说明。

[0039] 实施例中,动态拓扑执行调度模块用于将执行主体分解到序列图模型的交互片上,并通过动态拓扑执行队列管理序列图模型的交互片,并依据动态拓扑执行队列执行任务。具体地,通过动态拓扑执行队列管理当前尚未执行任务,并在恰当时机向动态拓扑执行队列中压入可能就绪的执行任务,使得交互片能以符合UML规范的偏序执行次序来执行交互片包含的语义。

[0040] 不同于当前业界方案中直接将整个序列图模型按照消息的垂直坐标顺序进行执行,本实施例使用构建的动态拓扑执行队列管理当前序列图模型执行过程中所有可能就绪的执行元素,这些执行元素是所有交互片(InteractionFragment)的派生类实例。具体来说,每个序列图模型的交互片上设有交互片仿真类(InteractionFragmentActivarion),该交互片仿真类存储有交互片的偏序关系和执行状态。

[0041] 针对偏序关系,为每个交互片实例设置了两个集合,即前序交互片集合和后序交互片集合,这样偏序关系通过前序交互片集合和后序交互片集合实现。在序列图模型初始化过程中,遍历当前序列图模型中所有生命线(Lifeline),对生命线上的交互片由上至下设置两交互片之间的偏序关系,依据设置的偏序关系将前一交互片加入到后一交互片的前序交互片集合,将后一交互片加入到前一交互片的后序交互片集合。只有当后一交互片执行时保证其前序交互片集合的交互片均已执行或被跳过,即可保证此后一交互片当前的执行满足UML规范,按照生命线有序的条件进行。

[0042] 实施例会维护一个包含交互片的动态拓扑执行队列。在每个执行循环中,将从动态拓扑执行队列中取出可能就绪的交互片,判定其是否已经准备好执行其语义,并执行其相应的语义。在交互片完成执行后,将会把后序交互片也加入动态拓扑执行队列中,从而实现了动态拓扑执行调度。图2展示了在结合后文语义分割后,动态拓扑执行调度的运行过程,在每个循环中,执行队列中队首的交互片被取出,判定其当前需要执行的语义类型,以及语义是否准备就绪。如果就绪的是主体语义,则执行主体语义并在完成后将结束语义插入回执行队列的前端;如果就绪的是结束语义,则执行结束语义,将同步等待事件以及后序交互片插入到执行队列尾端,其中,同步等待事件可以为后文的同步消息接收事件。

[0043] 针对执行状态,为每个交互片设置两个布尔值,即第一布尔值(shouldExecute)和第二布尔值(isFinished),每个交互片的执行状态在实例中被保存为第一布尔值和第二布

尔值,其中,第一布尔值标明交互片是正常等待执行或因各种原因不需要再执行,第二布尔值标明交互片是否完成了执行。两个布尔值的初始值分别设置为正确(true)和错误(false),第一布尔值被设置为正确时,表明交互片处于正常等待执行状态,被设置为错误时,表明交互片处于不需要再执行状态。当第二布尔值被设置为正确时,表明交互片处于完成执行状态,被设置为错误时,表明交互片处于未完成执行状态。当执行一个交互片时,将会把该交互片的第一布尔值置为错误,表明此交互片不应该再被调度执行,但同时本交互片仍会通过第二布尔值设置的错误值阻塞后序交互片的执行,直到它完成执行并将第二布尔值置为正确值为止。

[0044] 当任意的交互片执行完毕后,将该交互片的后序交互片集合中交互片的所有交互片添加到动态拓扑执行队列中。因为在此交互片执行完成前,它的前序交互片集合中必然包含此未完成的交互片而无法执行。此交互片执行完成后,他们则有可能因此进入了可执行的就绪状态。

[0045] 当组合交互片(CombinedFragment)执行时,将其内部需要执行的交互片加入到动态拓扑执行队列中。对于组合交互片的循环语义,在所有内部交互片执行完且需要循环的情况下,重置内部交互片的执行状态后,将内部交互片重新加入到动态拓扑执行队列中以实现内部语义的循环执行。

[0046] 通过将执行主体分解到交互片上并通过单一的动态拓扑执行队列进行管理,使得单条生命线上的执行不需要被另一条生命线上的消息所阻塞,并且使得组合交互片执行过程中仍可穿插其他与其没有偏序关系的交互片的执行过程。从而使得本发明系统可以处理诸如并行(Parallel)语义及和其他逻辑控制语义组合使用时的复杂情况。整体上来说,本发明的执行过程被视为一个可以动态添加交互片的拓扑排序过程。

[0047] 将执行主体拆分到交互片上虽然实现了更灵活、精细的执行次序控制,但也给带来两个技术问题:(1)同步消息的发送交互片如何在等待操作完成期间不阻塞其他交互片的执行;(2)组合交互片如何在知道内部交互片执行完毕,并调起后序交互片。为了解决这两个技术问题,实施例提出了基于语义分割的消息同步等待模块,用于将交互片的执行语义分割为负责主体执行逻辑的主体语义和负责序列图模型执行次序相关执行内容的结束语义,即让单个执行语义能够分离自身的主体语义信息和根据在序列图模型中位置而在结束时需要执行的执行次序相关语义,基于主体语义和结束语义实现同步消息的发送交互片的等待行为、组合交互片的整体执行。

[0048] 具体地,基于主体语义和结束语义实现同步消息的发送交互片的等待行为,包括:

[0049] 针对同步消息的发送交互片,其主体语义为对消息参数进行取值,并将取到的参数值发送给其对应的接收交互片,结束语义与一般交互片一样,是调起后序交互片,发送交互片执行完主体语义后从动态拓扑执行队列中退出并进入等待状态;接收交互片完成执行后通知发送交互片结束等待,将发送交互片以结束语义加入到动态拓扑执行队列中,动态拓扑执行队列再次执行发送交互片且设置完成状态,并调起后序交互片。这样实现了同步消息的等待机制的同时保留了按交互片调度的灵活性。

[0050] 如图3所示,在没有语义分割的情况下,消息发送事件经过占用执行线程同步等待接收事件执行完毕后,进行消息接收事件;消息接收事件经过等待前序发送事件执行完成后开始执行消息发送事件,这样同步消息的发送和接收交互片会由于循环等待而造成死

锁。而如图4所示,在应用了语义分割之后,两个交互片被分别划分为主体语义和结束语义,首尾相接的两条等待关系分别被转换为了消息接收事件的主体语义与消息发送事件的主体语义间的等待关系,和消息发送事件的结束语义与等待消息接收事件的结束语义间的等待关系。这样的转换之后,等待关系不再成环,解决了图3中描述的死锁技术问题,也能得到如图4中的一条合法执行路径。

[0051] 具体地,基于主体语义和结束语义实现组合交互片的整体执行,包括:

[0052] 针对组合交互片,其主体语义为将所有需要执行的内部交互片添加到动态拓扑执行队列,并将所有不需要执行的内部交互片的状态设置为不需要再执行,即将内部交互片的第一布尔值设置为错误值。这样组合交互片通过将其实际执行语义拆散为内部交互片,通过执行队列的调度实现与其他交互片交替执行。

[0053] 为组合交互片的内部交互片的执行元素设置外部包裹交互片(enclosingFragment)字段,实现在内部交互片的执行语义中向外部包裹交互片通知内部交互片的执行已经完成,这种通知是通过将内部交互片的结束语义添加到动态拓扑执行队列实现的;当外部交互片发现内部交互片的执行状态全部为完成执行时,则开始执行外部交互片的结束语义以调起组合交互片的后序交互片。由此,实现了组合交互片在作为一个整体参与偏序关系的同时,内部元素能够在动态拓扑执行调度模块下以交互片的粒度进行调度执行。

[0054] 如图5所示,左侧展示了一个序列图中典型的组合交互片使用场景,其表述的语义是:组合交互片内的所有交互片将首先执行,待全部执行完成后再执行虚线框之下的其他外部交互片。在本发明系统内,组合交互片的主体语义将会首先执行并且调用内部交互片的主体语义,将其加入执行队列,并为自身的结束语义添加对于所有内部交互片结束语义的等待关系。这样随着之后内部交互片的主体语义被调度执行、触发结束语义并最终全部完成执行后。组合交互片的结束语义将会进入就绪状态并被调度执行,调起其所在生命线上的后序外部交互片,从而精确地实现了序列图所描述的语义。

[0055] 尽管在单线程中,可以通过在每次执行元素执行前判定其所有添加的时间约束,但当系统中主要被延时事件所阻塞时,将退化为对时间约束的忙等待,从而不必要地占用大量的CPU事件。为了优化这一过程,系统中还增加有基于小根堆的延时事件处理模块,用于通过以超时时间戳为排序依据的延时事件小根堆来存放所有超时事件,使得能够处理到序列图模型中的时间约束语义。

[0056] 当一个时间约束开始等待时,计算超时时间戳并根据超时时间戳将延时事件插入小根堆中,保证小根堆堆顶始终保存着当前系统中最早到期的时间约束。在每次从动态拓扑执行队列中取出交互片执行前,检查小根堆并当小根堆不为空且顶部时间约束已经超时,弹出该时间段约束并将其约束的交互片的主体语义加入到动态拓扑执行队列中。当动态拓扑执行队列已经为空时,由于此时除了超时事件外,已经不会有其他逻辑执行,则据小根堆顶部的超时时间命令当前线程等待到目标时间为止,避免CPU空转,从而实现高效的延时事件处理。

[0057] 实施例,在系统中还增加了基于统一门的门消息转发模块,用于将各类门元素的转发语义抽离成统一门,基于统一门实现消息转发。

[0058] 在UML规范中,门(Gate)元素会根据其依附元素的不同扮演不同的角色。在组合交

互片上,门分为内外门,负责将组合交互片内部和外部间的消息转发;在交互(Interaction)的边界上,门被视为形式门(Formal Gate),负责将交互内部消息转发到外部和从外部传入消息;在交互引用(InteractionUse)的边界上,门被视为实际门(Actual Gate),负责将消息转发进交互引用(InteractionUse)和从其中将消息转发出来。为了统一门的实现,将这种转发语义抽离成为统一门,该统一门包括两个能够配对且负责消息请求和参数转发的门元素。

[0059] 基于这种抽象定义的统一门,需要通过一套逻辑实现所有情景下门的语义。具体包括:每个门元素维持一个参数暂存区和对配对的门元素的引用,当一个门元素作为消息接收端时,会从消息中读取参数和请求信息并转发到配对的门元素;当一个门元素作为消息的发送端时,会等待配对门元素的请求信息到达,并将请求信息和参数进行转发。通过这样的实现,以上三种门的逻辑均能得以正常运作。最终结合动态拓扑执行调度模块、基于语义分割的消息同步等待模块以及基于小根堆的延时事件处理模块,就完成了对序列图模型仿真所有逻辑功能的较准确和完整的支持。

[0060] 以上所述的具体实施方式对本发明的技术方案和有益效果进行了详细说明,应理解的是以上所述仅为本发明的最优选实施例,并不用于限制本发明,凡在本发明的原则范围内所做的任何修改、补充和等同替换等,均应包含在本发明的保护范围之内。

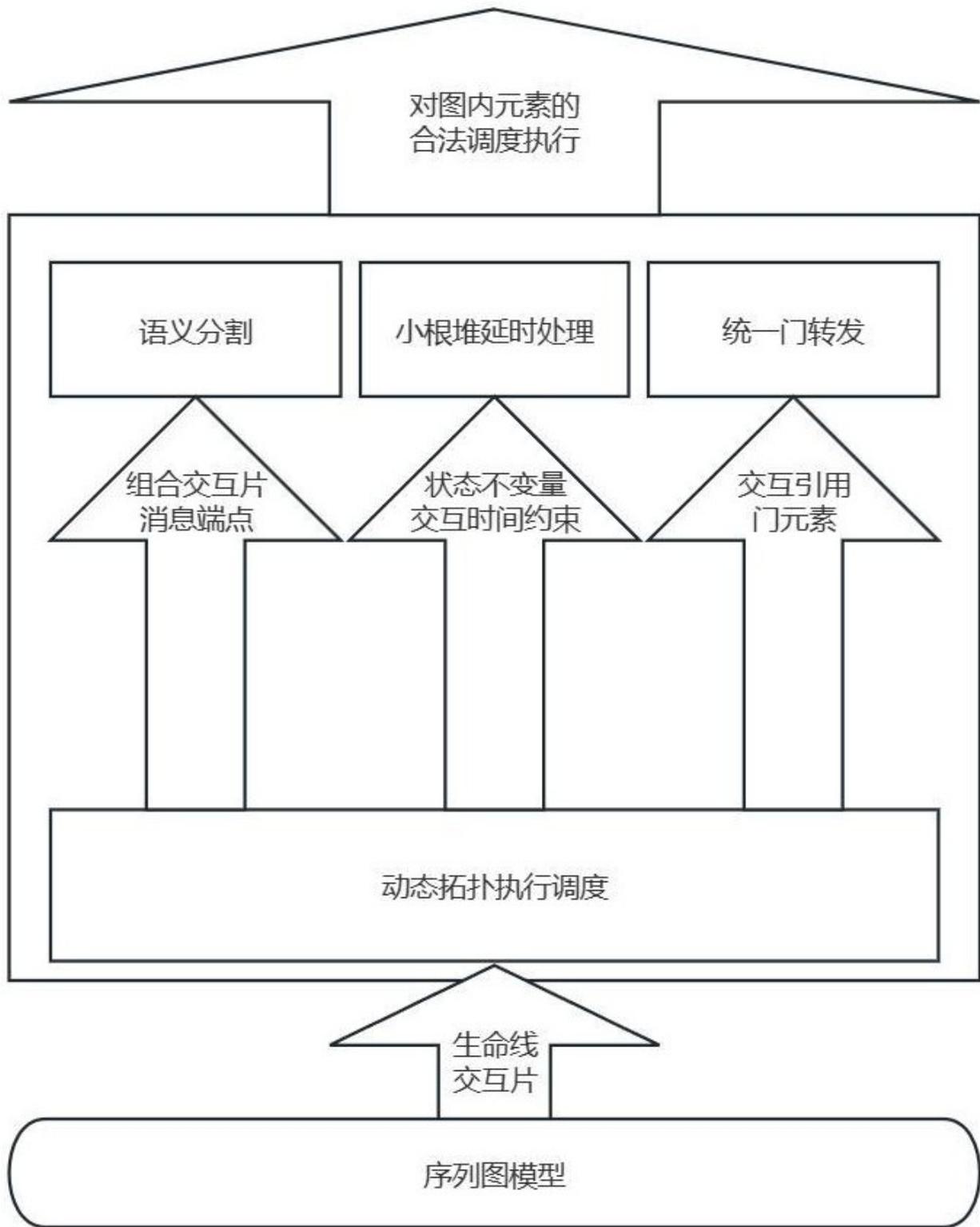


图 1

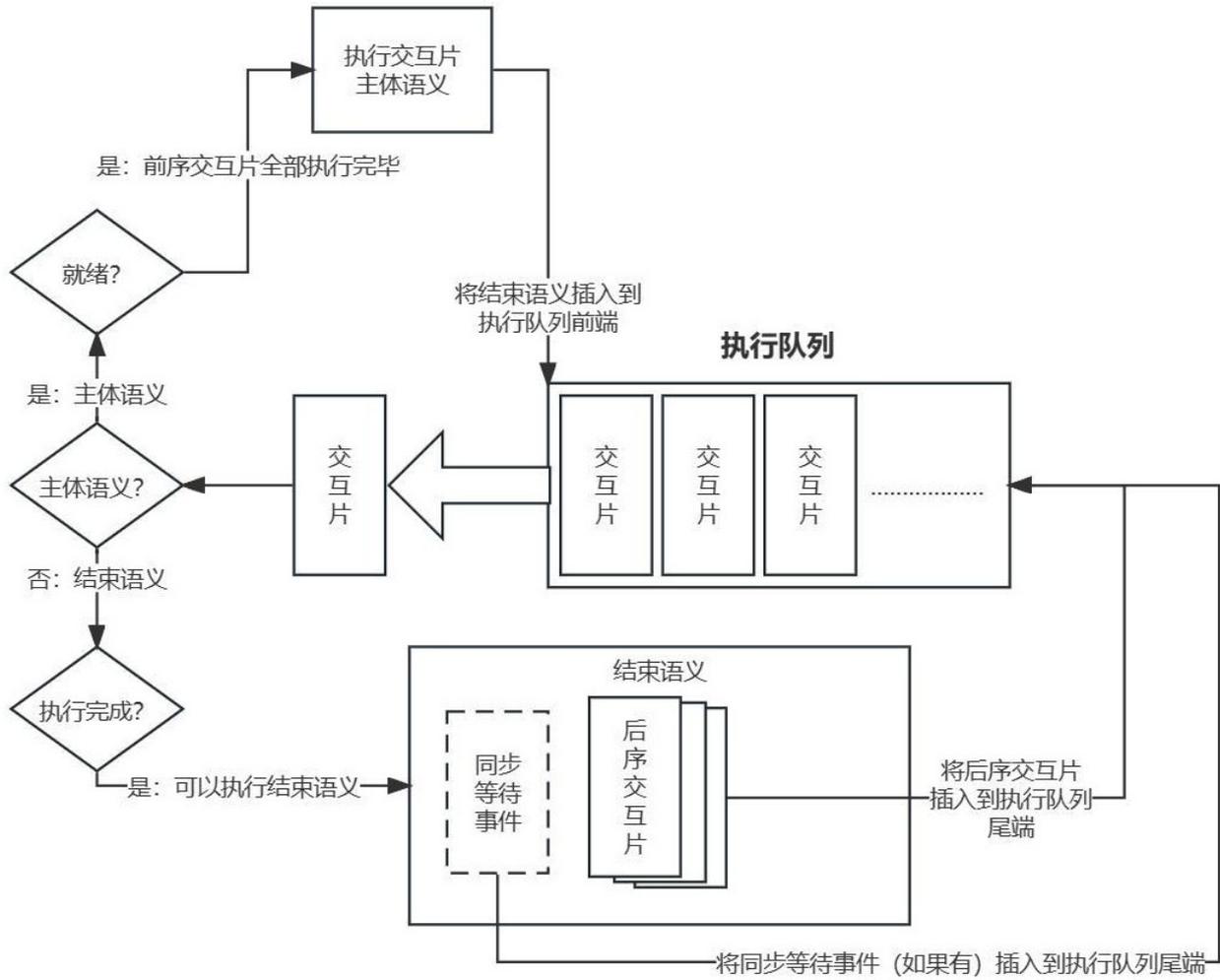


图 2

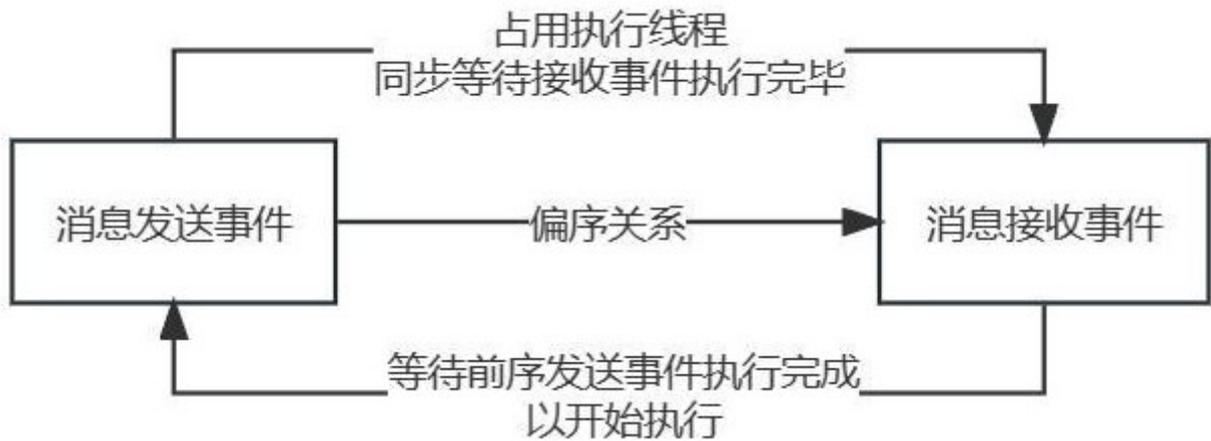


图 3

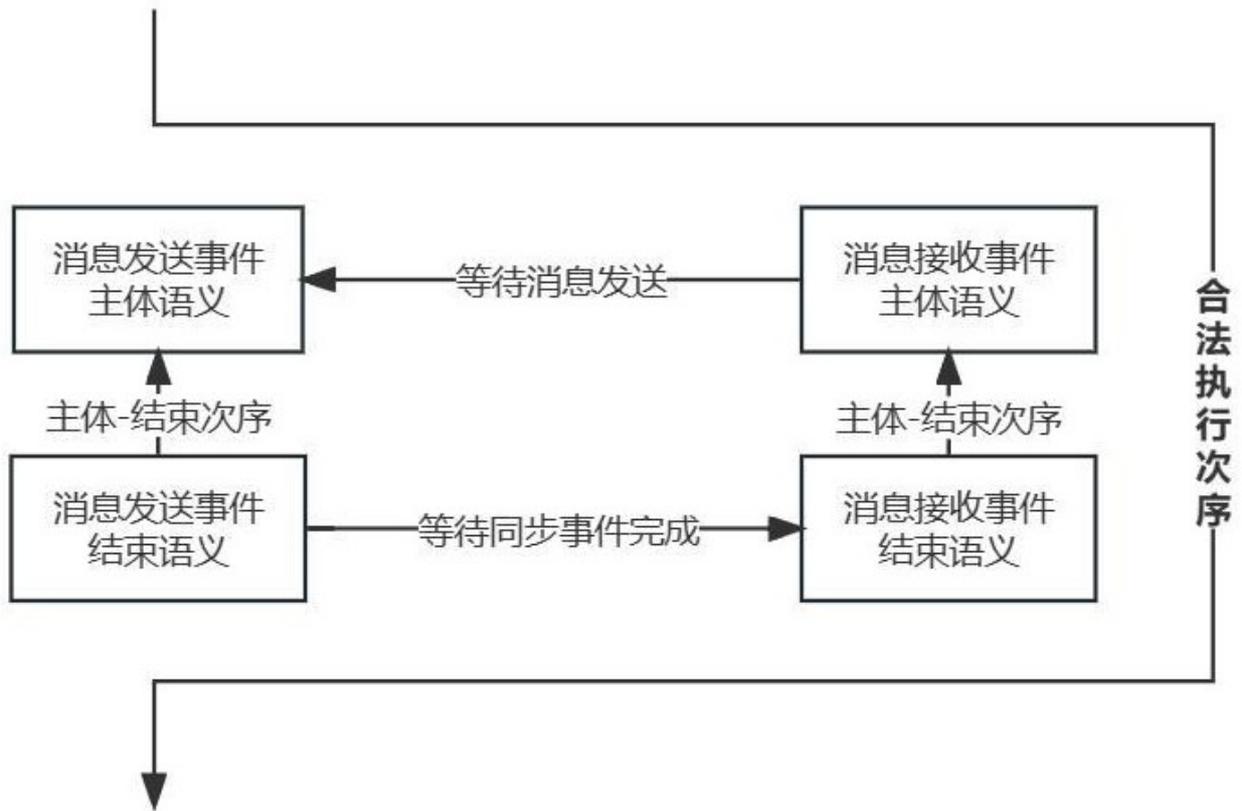


图 4

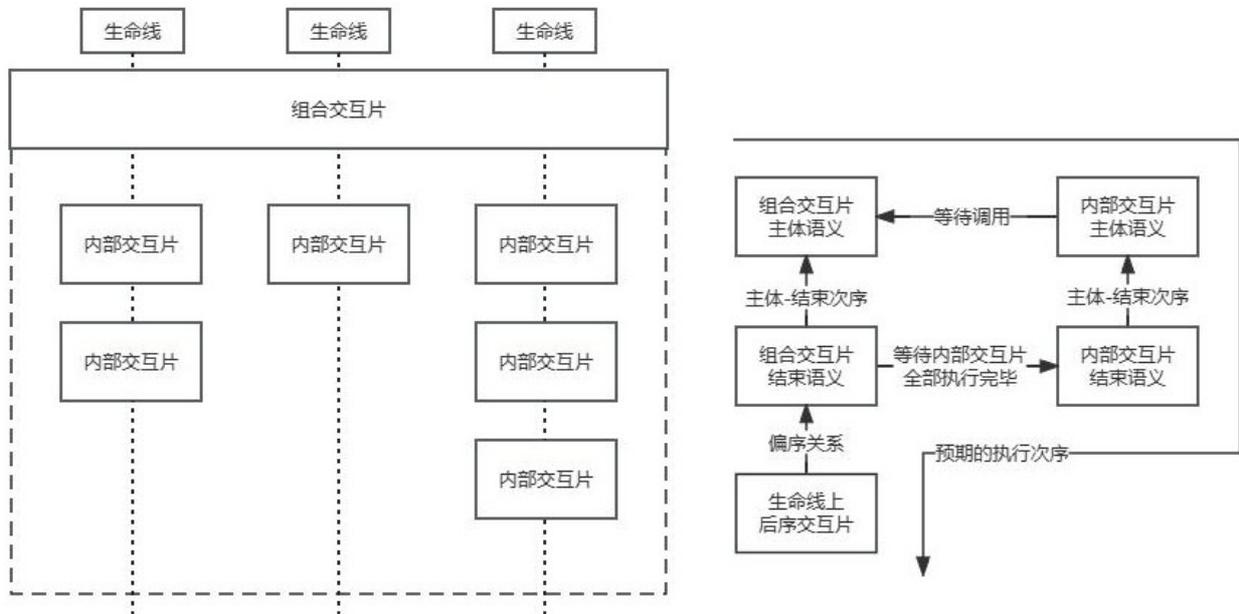


图 5