

基于 FIPA-ACL 的高层体系结构互操作性扩展^①

岳英超^② 肖田元^③ 范文慧 吴松 马成

(清华大学国家 CIMS 工程研究中心 北京 100084)

摘 要 针对高层体系结构(HLA)仿真中联邦对象模型(FOM)固定机制导致仿真系统交互信息灵活性不足的问题,提出了基于国际组织 FIPA 的智能体通信语言(FIPA-ACL)的 HLA 仿真互操作性扩展方法。该方法利用多智能体通信技术支持基于 FIPA-ACL 的中间件技术,以实现 FOM 固定机制的扩展。它能提高系统的通信性能,有效改进分布式仿真互操作的灵活性与动态性,提高分布式仿真的智能化水平。该方法是对现有标准 HLA 的扩展,具有很好的技术继承性,实现成本低。此方法在典型复杂产品——高速列车的仿真实例中得到了实际应用,从而检验了其可行性和有效性。

关键词 基于国际组织 FIPA 的智能体通信语言(FIPA-ACL),高层体系结构(HLA)分布式仿真,中间件,互操作性扩展

0 引言

目前在分布式仿真领域,高层体系结构(high level architecture, HLA)技术是一种通用的标准,它极大地促进了分布式仿真发展。然而,随着仿真系统规模的不断扩大,其复杂性不断增强,新的需求与新的问题也就逐渐出现。众多学者对此进行了持续研究。本文在继承传统 HLA 功能的基础上,对其进行扩展,重点解决 HLA 互操作灵活性扩展问题。

在 HLA 中,分布式仿真系统称为联邦,联邦中的仿真应用称为联邦成员,联邦成员间的互操作由运行支撑环境(runtime infrastructure, RTI)提供,其中的交互信息由联邦对象模型(federation object model, FOM)提供范式。HLA 中的 FOM 是在联邦建立过程中定义的,经过仿真程序和支撑环境的编译链接后,形成一个可执行的系统。在联邦运行时, FOM 是不能改变的,我们称其为固定机制。这种固定机制限制了联邦成员间通讯内容与互操作的灵活性和动态性,即运行时无法动态修改联邦成员间通讯内容与互操作信息,无法支持联邦成员之间交互数据的动态变化。虽然不同的学者和机构^[1-4]对 FOM 的建立过程、方法进行了持续的研究,但是对如何有效支撑分布式仿真的灵活性与动态性的问

题,缺乏充分的研究。而在众多仿真场景下,这是一个切实的需求。解决该问题的一类方法是提供一种仿真运行时能够随时被触发,进而动态修改并重新装载 FOM 信息的机制,但这种方法的实现要涉及整个分布式仿真支撑系统,成本过高;另一类方法是将消息结构与消息内容相互独立。在对多智能体系统(multi-agent systems, MAS)内智能体(Agent)间通信问题的研究过程中,部分学者^[5-7]注意到利用 HLA/RTI 为多智能体系统提供通信机制的可能性。然而,由于 Agent 的自主性与智能性,运行时固定的 FOM 对于 MAS 内通信的灵活性、动态性无法提供支撑。有学者^[5,6]针对该问题,提出了基于知识检索和操作语言(knowledge query and manipulation language, KQML)层的解决方案。但是,这种方法的主旨在于研究 MAS 中 Agent 间的通信环境,而不是从分布式仿真的角度,因此缺乏对仿真模型重用与互操作问题的关注。本文结合相关研究,提出了基于 Agent 技术国际组织 FIPA(Foundation for Intelligent Physical Agents)提供的智能体通信语言(agent communication language, ACL)(简称为 FIPA-ACL)的 HLA 分布式仿真互操作性扩展方法。该方法基于 ACL 的交互信息结构,可以实现仿真运行中 FOM 固化条件下,交互信息动态、灵活的变化,支持分布

① 863 计划(2009AA110302)和国家自然科学基金(60874066)资助项目。

② 男,1984 年生,博士生;研究方向:分布式仿真,复杂产品协同仿真;E-mail: yyc03@mails.tsinghua.edu.cn

③ 通讯作者, E-mail: xty-dau@tsinghua.edu.cn

(收稿日期:2011-04-26)

式仿真更加灵活的互操作,提升了 HLA 仿真系统的智能化水平。

1 FIPA-ACL

在智能体通讯标准中,早期主要采用 KQML,它基于言语行为(speech act)理论进行定义,可以支持智能体之间的信息交互。但是,KQML 缺乏严格的形式化语义,不利于智能体间的互理解与互操作。因此,FIPA 组织提出了 FIPA-ACL^[8]。

与 KQML 相比,FIPA-ACL 作为一种被广泛接受的标准(PIFA 在 2005 年并入 IEEE 标准组织),具有更加完整和严密的形式化语义描述能力,从而消除了通信中可能含有的任何二义性。它通过显式定义消息通信原语来规范消息结构,实现消息内容与传递语义相互独立。这为克服 HLA 固化 FOM 机制的不足提供了解决途径,从而使提高 HLA 仿真中通信的灵活性和动态性成为可能。

2 基于 FIPA-ACL 的互操作性扩展方法

2.1 基于 FIPA-ACL 的 FOM

为便于讨论,首先对一些概念进行定义。

定义 1 原始联邦成员(primitive federate, PF):严格按照 HLA 标准实现的仿真应用程序。它能够实现与 RTI 的通信,并通过 RTI 与其他联邦成员协同工作,实现分布式仿真系统的运行。

定义 2 扩展联邦成员(extended federate, EF):在原始联邦成员的所有功能的基础上,增加并使用了具备互操作性扩展能力的支撑中间件的原始联邦成员。注:若其未参与灵活性扩展互操作,仍不能称为 EF。

定义 3 联邦成员(federate):指一切满足 HLA 标准要求的分布式仿真子系统。在本文中,是原始联邦成员和扩展联邦成员的并集,即: $Federate = PF \cup EF$ 。简称成员。

定义 4 联邦成员逻辑框架(federate logical frame, FLF):联邦成员中,与联邦相关的创建、加入、订阅、发布、注册、更新/反射数据、发送/接收交互、退出、销毁等整个逻辑过程,称为联邦成员逻辑框架。FLF 是且仅是联邦成员中体现 HLA 标准的部分。

定义 5 ACL 原语对象类/交互类:FOM 中基于 FIPA-ACL 原语定义的对象类和交互类。

定义 6 会话(Conversation):在分布式仿真联邦中,基于同一类型的信息发生的交互,我们定义为一个会话。

关于定义 6 有两点值得注意:第一,其中的信息类型不等价于 FOM 中的对象类/交互类,不同的对象类/交互类对应不同的信息类型,但是相同 ACL 原语对象类/交互类也可以对应不同的信息类型;第二,两个联邦成员之间可以不发生会话,或发生多于一种会话(多种交互信息)。

由定义 5 和定义 6 可知:ACL 原语对象类/交互类以一致的消息结构支持多样的信息类型,同样,订阅/发布 ACL 对象类/交互类的联邦成员之间可能存在多于一种会话。

定义 7 EFJ-R 事件:一个包含未知交互信息的扩展联邦成员(EF)在仿真运行时加入分布式仿真(EF Join At Runtime)的事件。

在 HLA 体系中,RTI 提供了一种基于对象类和交互类进行信息共享的通信机制。本文基于 FIPA-ACL 中的原语来定义 FOM 的对象类和交互类,即 ACL 原语对象类/交互类,每一个原语的参数可以对应定义为对象类的属性或交互类的参数。

值得注意的是,FIPA-ACL 与 RTI 的信息交互理念有所区别:前者针对 MAS 中的通信,消息内包含发送方和接收方信息,由 Agent 平台根据发送方要求将消息传送到接收方;而 RTI 是基于订阅/发布的通信机制,任何信息的发送方并不关心接收方,而任何接收方也无法获知发送方的相关信息。因此,仅从两种体系的通信机制上分析,我们可以将 FIPA-ACL 行为原语中的发送方、接收方等订阅/发布机制的无关属性略去,以简化消息结构。

在 FOM 中,对象类表示持续的状态信息,交互类表示暂时的事件信息,运行时体现为联邦执行数据(federation execution data, FED)文件。图 1 给出了 FED 文件中定义的一部分 ACL 原语对象类,其

```

(objects
  (class ObjectRoot
    (attribute privilegeToDelete reliable timestamp)
    (class RTIprivate reliable receive)
    (class FIPA_ACL_object)
      (attribute performative reliable times:amp)
      (attribute content reliable timestamp)
      (attribute language reliable timestamp)
      (attribute encoding reliable timestamp)
      (attribute ontology reliable times:amp)
      (class inform)
      (class confirm)
      (class not_understood)
      (class request)
      (class cancel)
  )
)

```

图 1 FED 文件中的部分 FIPA-ACL 原语对象类

中略去了 sender、receiver、reply-to 等无关属性。ACL 原语交互类的定义方法相同,不再赘述。此外,仿真人员也可以根据特殊的仿真需求在原语对象类/交互类的基础上添加属性/参数,或是添加继承类。

基于这样的 FED 文件,一方面,因扩展联邦成员中参与交互的信息结构和原语类型都被 FOM 规范化,无论在联邦运行的起始阶段,还是在运行过程中,它都可以加入/退出 HLA 联邦;另一方面,由于 ACL 原语对象类/交互类的统一消息结构中 ontology、content 等属性含有丰富的语义信息,从而在 FOM 固定机制下,支持交互信息的动态变化,支持分布式仿真更加灵活的互操作能力。

2.2 性能分析

本节分析上述方法的性能。为便于定量讨论,所进行的分析基于如下假设:(1)忽略仿真模型的计算性能,仅考虑其通信性能;(2)一个仿真周期内,所有发布/订阅的数据交互且仅交互一次;(3)假定每个参与交互的数据的 Federate 地位均等,且每个 Federate 每次交互的数据量相同,并将其归一化。另外还给出了参数符号及其定义,如表 1 所示。

表 1 符号及其定义

符号	定义
l	参与分布式仿真的物理节点总数
n	参与分布式仿真的 Federate 总数
m	参与分布式仿真的 EF 数目
p	订阅某 Federate 交互数据的 Federates 的平均数
q	某 Federate 订阅的交互数据类型的平均数
r	发布某类型交互数据的平均 Federate 数
τ	Federate 的平均连通率:与某 Federate 交互数据的平均 Federate 数 ($\tau = (p + qr)/2$)
s	EFJ-R 事件发生的次数
Com	一个仿真周期内,仿真系统的通信流量
μ	不发生 EFJ-R 事件下,包含 EF 与只包含 PF 的仿真的通信开销比
φ	EFJ-R 事件发生后较发生前的通信开销比
η	对两个仿真目的相同的系统 a、b,系统 a 通过发生 EFJ-R 事件后得到仿真结果 C_a ;系统 b 则是事先设计完备而后仿真得到的结果 C_b 。则 $\eta = (C_a - C_b)/C_b$ 定义为通信效能代价

若仿真系统 A 只包含原始联邦成员(PF),即 $m = 0$,则一个仿真周期内,接受某 Federate 交互数据的通信流量为

$$\frac{l-1}{l}p \quad (1)$$

一个仿真周期内,发送交互数据给该 Federate 的通信流量为

$$\frac{l-1}{l}qr \quad (2)$$

一个仿真周期内,整个分布式仿真系统的通信流量为

$$Com_{PF} = \frac{n(l-1)}{2l}(p + qr) \quad (3)$$

若仿真系统 B 包含 m 个 EF,则一个仿真周期内,整个分布式仿真系统的通信流量为

$$Com_{EF} = \frac{n(l-1)}{2l}(p + qr) + \frac{m(l-1)}{2l}\left[2m - 2 - \frac{m}{n}(p + qr)\right] \quad (4)$$

其中前一部分为仿真系统 A 的通信流量;后一部分为加入 EF 后增加的通信流量,并把前一部分中计算过的通信流量减除。这样,在不发生 EFJ-R 事件的情况下,包含 EF 与只包含 PF 的仿真的通信开销比为

$$\mu = \frac{Com_{EF}}{Com_{PF}} = 1 + \frac{m(2m - 2 - \frac{m}{n} \times (p + qr))}{n(p + qr)} = 1 + \frac{m(m - 1 - \frac{m}{n} \times \tau)}{n\tau} \quad (5)$$

若在系统 B 运行过程中某时刻,同时发生 s 个 EFJ-R 事件,系统的通信流量变为

$$Com_{EFJ-R} = Com_{EF} + \frac{l-1}{l} \times \left[\frac{(n + s - m)}{n + s} \tau s \times 0.5 + (2m + s - 1)s \right] \quad (6)$$

即在 C_{EF} 基础上增加 s 个 EF 与原系统中的 $n - m$ 个 PF 和 m 个 EF 的通信流量。

EFJ-R 事件发生后,形成的仿真系统 C 与系统 B 的通信开销比为

$$\varphi = \frac{Com_{EFJ-R}}{Com_{EF}} = 1 + \frac{(0.5 \times (n + s - m)/(n + s)) \times \tau s + (2m + s - 1)s}{n\tau + m(m - 1 - (m/n) \times \tau)} \quad (7)$$

通信效能代价为

$$\eta = \frac{n\tau + m(m - 1 - (m/n) \times \tau)}{(n + s)\tau}$$

$$+ \frac{((n-m)/n) \times \tau s + (m-1)s}{(n+s)\tau} - 1 \quad (8)$$

需要指出的是,根据通信效能代价的定义,计算 η 的分母并不是(3)式中的系统 B 的 Com_{PF} ,而是无需发生 EFJ-R 事件也可以完成系统 C 的仿真任务的系统 C' 的 Com_{PF}' ,如(8)式分母所示。

若一个 HLA 仿真系统 E, $l=10, m=10, n=20, p=8, q=4, r=2, s=4$, 则 $\tau=8, Com_{PF}=144, Com_{EF}=189, Com_{EFJ-R}=280.2, Com_{PF}'=172.8, \mu=1.3125, \varphi=1.4825, \eta=0.6215$ 。对于该系统,包含 EF 的通信开销是只包含 PF 的 1.3125 倍;对于包含 EF 的情况,若同时发生 4 个 EFJ-R 事件,通信流量变为原来的 1.4825 倍;通信效能代价为 62.15%。

3 基于 FIPA-ACL 的互操作性扩展方法的优化

从上节性能分析中的例子可以得到 HLA 仿真系统 E 的通信效能代价为 62.15%,这意味着,为了获得互操作性扩展能力,需要付出比原系统多 62.15% 的通信开销。这在节点较多、通信频繁的分布式仿真中会大大增加系统的时间开销。因此,本节我们对基于 FIPA-ACL 的互操作性扩展方法进行优化。

3.1 数据过滤

在 RTI 中,基于订阅/发布的信息交互机制,可以确保消息“按需分配”,不会出现广播机制造成的通信负载过重问题。然而,由于 ACL 原语对象类/交互类具有同一的结构,所有订阅了 ACL 原语对象类/交互类的联邦成员都会收到所有被更新/发送的 ACL 原语对象类/交互类。这使得订阅/发布机制本身对于仿真数据的过滤作用消失,增加了整个分布式系统的通信负担。因此,为了提高通信效率,需要对交互数据进行过滤。RTI 的数据分发管理(data distribution management, DDM)服务可以解决这个问题^[1]。DDM 基于路由空间的区间过滤,对对象类的属性、交互类的参数进行数值过滤。

对于订阅/发布机制,不同研究者的表达方式各有侧重^[9, 10],针对问题特点,做如下形式化定义:

定义 8 订阅: 订阅 s 可以描述为一个三元组: $s = \langle class, attr, cons \rangle$ 。

其中 $class$ 表示订阅的主题,具体到本问题即为订阅的对象类标识; $attr$ 和 $cons$ 表示订阅的内容,具

体来说 $attr$ 表示订阅的对象类属性集合 $attr = \{a_1, a_2, \dots, a_n\}$, 其中的元素为属性句柄和属性名称的属性描述 $a_i = (AHandle_i, AName_i)$; $cons$ 表示订阅的对象类属性约束集合 $cons = \{c_1, c_2, \dots, c_m\}$, 其中的元素为属性句柄、属性值区间和属性类型的约束描述 $c_j = (AHandle_j, AConstraint_j, AType)$ 。

对于 ACL 原语对象类/交互类,本文选取其中的两类属性进行基于 DDM 的数据过滤。第一类属性是 conversation-id, 即会话 ID。第二类属性是 sender。图 2 给出了在 FIPA-ACL 原语对象类增加这两类属性之后的 FED 文件局部图。

```
(objects
(class Object Root:
(attribute privilege ToDelete reliable timestamp)
(class RTIprivate reliable receive)
(class FIPA_ACL_object)
(attribute performative reliable timestamp)
(attribute sender reliable timestamp)
(attribute context reliable timestamp)
(attribute language reliable timestamp)
(attribute encoding reliable timestamp)
(attribute ontology reliable timestamp)
(attribute conversation-id reliable timestamp))
(class inform)
(class confirm)
(class not_understood)
(class request)
(class cancel)
```

图 2 修改后的 FED 文件局部

根据定义 6, 会话 ID 与联邦成员之间交互的信息类型形成一一映射。因此订阅信息的联邦成员利用 DDM 服务, 在路由空间中限定相应的会话 ID, 即可避免消息类型一致, 却内容不相关的仿真数据了。同类型的信息可能有联邦中的多个成员发布, 然而这些发布的信息并不都是该信息的订阅者需要的。因此, 利用 DDM 服务, 限定无关的发送者和接收者, 可以避免冗余信息的通信, 降低系统数据交互的开销。

对于 2.2 节中的 HLA 仿真系统 E, 以其中一个扩展联邦成员 EF_k 为例, 它订阅了 FIPA-ACL 原语对象类。在未优化的情况下, 它按照主题进行订阅, 会接收到所有其他 EF 发布的 FIPA-ACL 原语对象类数据, 造成冗余数据传输; 在优化后的情况下, 它以基于内容的方式进行订阅, 可能的形式如下:

$$s = \langle FIPA_ACL_object, \{ (2, sender), (7, conversation - id) \}, \{ (2, int, [3, 5]), (2, int, 9), (7, int, 6) \} \rangle \quad (9)$$

式(9)表示它订阅 FIPA-ACL 原语对象类数据, 同时需要数据是成员句柄为 3、4、5、9 的 EF 发布的, 且会话 ID 为 6 的。这样能避免冗余数据传输, 降低系统通信开销。

3.2 性能分析与比较

2.2 节中分析了未优化时的通信性能,本节将优化后结果与其进行比较。首先计算优化后性能。

一个仿真周期内,整个分布式仿真系统的通信流量为

$$Com_{PF} = \frac{n(l-1)}{2l}(p+qr) \quad (10)$$

尽管仿真系统包含 m 个 EF,但由于 3.1 节中的数据过滤,系统不再发生冗余数据传输,因此一个仿真周期内,整个分布式仿真系统的通信流量为

$$Com_{EF} = Com_{PF} = \frac{n(l-1)}{2l}(p+qr) \quad (11)$$

即 $\mu = 1$ 。

当发生 EFJ-R 事件,即当携带未知交互信息的扩展联邦成员在仿真运行时加入时,由于信息的未知性,其 conversation-id 也是未知的,同时,其他成员基于 sender 的数据过滤也将该成员排除在外,这将导致其他联邦成员无法接收该成员的仿真数据。为此,对于这样的扩展联邦成员,它更新/发送的 ACL 原语对象类/交互类的 conversation-id 和 sender 属性取值为通配符。则同时发生 s 个 EFJ-R 事件后系统的通信流量变为

$$Com_{EFJ-R} = Com_{EF} + \frac{l-1}{l} \times \left[\frac{(n+s-m)}{n+s} \tau s \times 0.5 + ms \right] \quad (12)$$

即在 Com_{EF} 基础上增加 s 个 EF 与原系统中的 $n-m$ 个 PF 和 m 个 EF 的通信流量。

EFJ-R 事件发生后,形成的仿真系统与原系统的通信开销比为

$$\begin{aligned} \varphi &= \frac{Com_{EFJ-R}}{Com_{EF}} \\ &= 1 + \frac{(0.5 \times (n+s-m)/(n+s)) \times \tau s + ms}{n\tau + m(m-1 - (m/n) \times \tau)} \end{aligned} \quad (13)$$

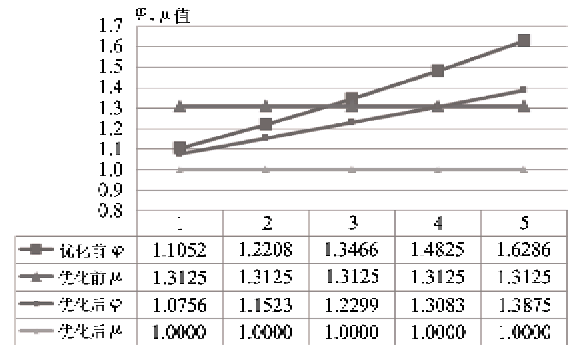
通信效能代价为

$$\begin{aligned} \eta &= -1 + \frac{n\tau + m(m-1 - (m/n) \times \tau)}{(n+s)\tau} \\ &\quad + \frac{(0.5 \times (n+s-m)/(n+s)) \times \tau s + ms}{(n+s)\tau} \end{aligned} \quad (14)$$

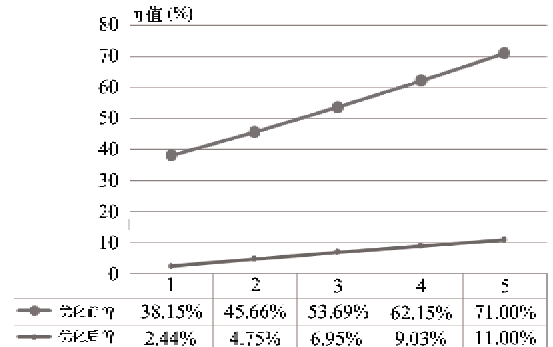
同样,对于仿真系统 E, $l=10, m=10, n=20, p=8, q=4, r=2, s=4$, 则 $\tau=8, Com_{PF}=144, Com_{EF}=144, Com_{EFJ-R}=188.4, Com_{PF}'=172.8, \mu=1, \varphi=1.3083, \eta=0.0903$ 。即对系统 E, 包含 EF 的通

信开销与只包含 PF 的情况相同;对于包含 EF 的情况,若同时发生 4 个 EFJ-R 事件,通信流量变为原来的 1.3083 倍,通信效能代价为 9.03%。

图 3 给出了一组性能优化前后的对比图,图中的 l, m, n, p, q, r 与仿真系统 E 相同,分别计算 EFJ-R 事件数从 1 到 5 的 φ, μ (图 3(a) 中所示) 和 η (图 3(b) 中所示)。从图中可以看出,三个指标反映的优化后通信性能较优化前均有大幅度提升。



(a) 性能优化前后 φ, μ 值对比



(b) 性能优化前后 η 值对比

图 3 性能优化前后对比图

3.3 联邦成员间基于新 FED 文件通信的实现

图 4 是联邦成员间的通信顺序图,此处仅为说明通讯的实现流程,只选取联邦中的两个成员:成员 1、2 都是 EF,由成员 1 发送数据至成员 2。新 FED 文件如图 2 所示。成员 1 的仿真数据首先从仿真模型传递到成员 1 的支撑中间件,根据数据包含的信息类型选择是否进行原语封装,而后经过 RTI 的本地节点驻留模块(local RTI component, LRC)的数据过滤(其中 ACL 原语对象类/交互类实行如 3.1 节描述的强制过滤,普通对象类/交互类进行选择性过滤)并发送给 RTI 的全局模块(RTI gateway, RTIg),并经过 RTIg 发送到成员 2 的 LRC,进而由支撑中间件根据信息类型选择是否进行原语解析,最终把仿真数据发送给成员 2。从而完成一次数据通信。这

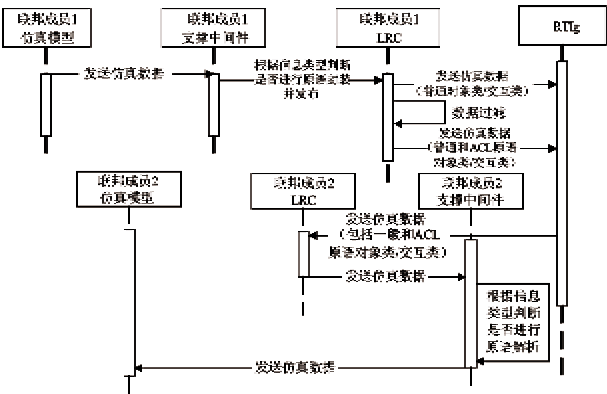


图 4 联邦成员间的通信顺序图

样的通信流程,对于其他扩展联邦成员动态加入联邦并携带未知信息的情况(即发生 EFJ-R 事件时)依然成立。

4 具扩展互操作性的分布式仿真体系结构

在 HLA 仿真中,仿真联邦由众多联邦成员组成,多个联邦成员组成分布式仿真联邦,并由 RTI 提供信息交互支撑功能,实现模型重用和互操作。FOM 作为联邦交互信息的模板,为 RTI 的通信提供数据结构。每一个联邦成员通过 RTI 大使与联邦大使与 RTI 进行通信。联邦成员内部,包括利用不同工具建立的仿真模型,以及支撑仿真模型与 RTI 集成、支持基于 ACL 通信模块的统一中间件。联邦中部分成员依然可以是传统 PF(配备传统适配器的仿真模型,不具备互操作性扩展能力)。

分布式仿真互操作性扩展方法在实现层面需要支撑中间件作为载体。基于 FIPA-ACL 互操作扩展的 HLA 分布式仿真支撑中间件的体系结构如图 5 所示。

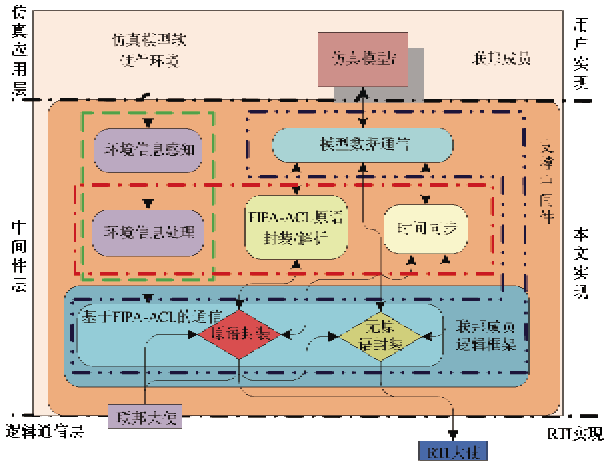


图 5 支持 FIPA-ACL 互操作扩展的中间件体系结构

示,包括环境信息感知、环境信息处理、模型数据通信、FIPA-ACL 原语封装/解析、时间同步、基于 FIPA-ACL 的通信等模块。图中的双点划线区域表示支撑中间件的通信功能部分,虚线区域表示环境信息相关的获取、处理功能部分,点划线区域表示对仿真数据、环境信息进行智能处理功能部分,箭头表示数据流动方向。

联邦成员接收 RTI 传递的仿真数据(包括传统对象类/交互类和 ACL 对象类/交互类规范数据结构的仿真数据),首先通过联邦大使进入联邦成员逻辑框架,经过基于 FIPA-ACL 的通信模块(该模块具有可剪裁性,即对传统的仿真模型,其交互信息已经在 FOM 中完备定义,则可不经过该模块,直接与 RTI 大使与联邦大使进行通信)进行有无原语封装的判断,对于 ACL 对象类/交互类对应的仿真数据,将其传递给 FIPA-ACL 原语封装/解析模块,提取语义信息并根据语义库进行解析。并由时间同步模块协调联邦成员的逻辑时间与仿真模型的仿真时间,使各仿真模型时间逻辑正确,协同推进。最后,经模型数据通信模块与仿真模型与仿真引擎进行数据共享。联邦成员发送仿真数据的流动过程与上文描述的过程相反,最后由 RTI 大使发送到 RTI。同时,环境信息感知模块获取仿真模型所在计算环境的相关信息(CPU 使用率、内存占有率、网络带宽占有率等);环境信息处理模块基于该联邦成员所在节点的环境信息处理得到该节点的繁忙程度,反馈给分布式仿真支撑环境。

基于 ACL 的通信模块与 FIPA-ACL 原语封装/解析模块协同作用,使扩展联邦成员可支持原 FOM 中的信息交互,同时支持 ACL 原语对象类/交互类的通信,突破了传统分布式仿真固化 FOM 对仿真动态性、灵活性的阻碍。

5 应用实例

高速检测列车动车组是目前世界上最先进的高速列车检测设备,是典型的复杂产品,对它的仿真研究,为高速检测列车的设计优化提供分析手段,具有很高的理论和工程价值。本文结合 863 项目“支持 400km/h 高速检测列车研制的协同仿真平台建设和应用”,建立了针对高速列车的分布式仿真系统,并将基于 FIPA-ACL 的 HLA 分布式仿真互操作性扩展方法应用其中,提升了分布式仿真的动态性和灵活性。该系统的体系结构如图 6 所示(仅给出 For-

tran 中间件细节,其他中间件相似)。

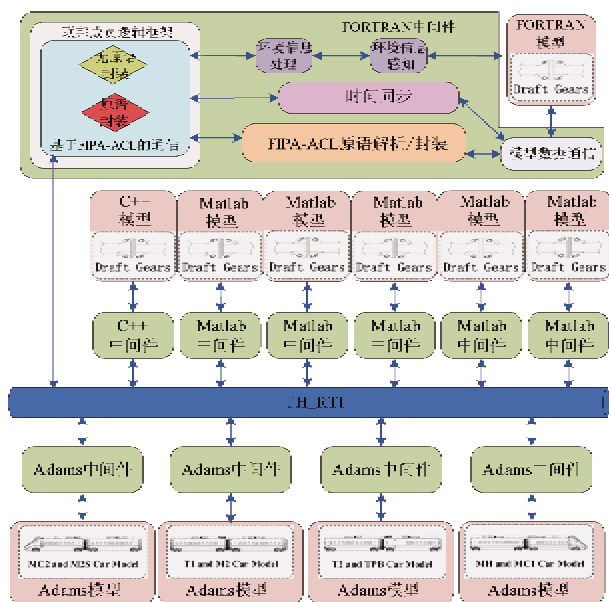


图 6 高速列车分布式仿真系统体系结构

仿真对象为 8 车厢动车组,在建模阶段,我们将仿真对象拆分为 11 个联邦成员:每两个车厢为一个整体建立 ADAMS 运动学模型;车厢间的 7 个勾缓分别建立 1 个 FORTRAN 模型,1 个 C++ 模型和 5 个 MATLAB 模型。按第 4 节的中间件技术,分别针对 ADAMS、ABAQUS、FORTRAN 和 MATLAB 开发分布式仿真支撑中间件,使不同的仿真模型可以方便、高效地与 RTI 集成,并支持灵活性扩展的互操作。

假定在该复杂产品的仿真设计过程中,发现耦合参数需要在线调整(即需要交互数据动态变化),传统方法无法满足仿真需求,需要利用本文的方法实现。分布式仿真的 FED 文件(如图 2 所示)中包含 ACL 原语对象类。具体来说,在仿真中将第二节车厢后部和第三节车厢前部运行参数定义为 ACL 原语对象类(即该联邦中包含两个 EF)。规定仿真时间为 20s,前 5s 该对象类的信息内容只包含车厢的位置信息,运行发现误差较大,不能满足设计要求。于是,后 15s 将信息内容动态改为包含位置信息和速度信息,精确度满足要求。从而实现了交互数据动态变化,即互操作性扩展,证明了本文方法的可行性与有效性。

仿真的场景是列车通过弯道过程,图 7 和图 8 分别反映了第二节车后部速度 Z 方向分量和第一节车前转向架左后轮的脱轨系数在列车行进中的变化情况。仿真过程逼真地刻画了实际物理过程。

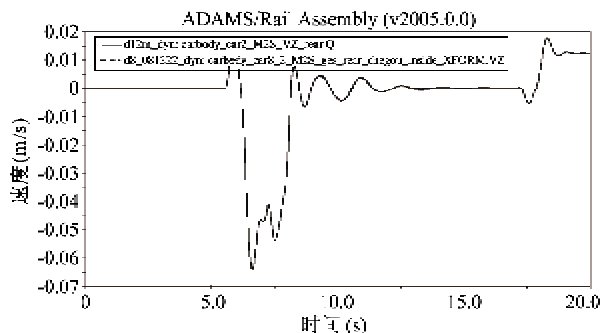


图 7 第二节车后部速度 Z 方向分量

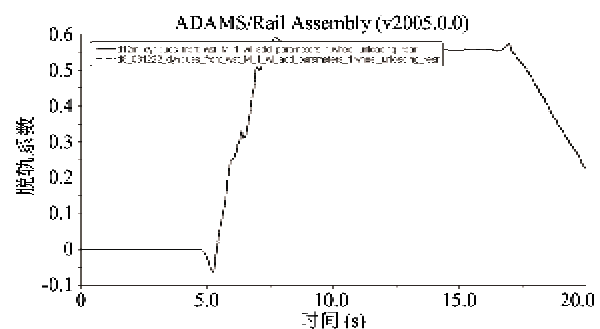


图 8 第一节车前转向架左后轮的脱轨系数

在通信性能方面,前 5s 和后 15s 的仿真时间推进速度相当,即并没有因为 ACL 原语对象类的引入导致冗余数据的交互,3.1 节中数据过滤机制在此发挥了作用。3.2 节的解析分析结果表明:包含两个 EF 的联邦,通过过滤机制的性能优化,通信开销比 μ 由 1.3125 降低到 1。实例仿真的结果支持了解析分析结果。

6 结论

本文提出了基于 FIPA-ACL 的 HLA 仿真互操作性扩展方法,它扩展了 FOM 固定机制,支持联邦成员间交互信息的动态变化,提高了仿真互操作的动态性、灵活性和智能化水平。同时,对该通信技术下的系统通信流量进行了定量分析,提出数据过滤的优化方法,提升了系统通信性能。基于本文的分布式仿真支撑中间件技术,可以方便地对已有模型进行重用,提高了分布式仿真的开发效率。

今后的工作在通信性能方面进行更加深入的研究,考虑寻找合适的作战仿真系统实例,验证发生 EFJ-R 事件时,数据过滤机制对系统通信性能提升的效果。

参考文献

- [1] 王军,姚益平. 基于 HLA 的基础对象模型研究. 系统

- 仿真学报, 2004, (8): 1659-1661
- [2] 王克明, 熊光楞. 一种基于 HLA 适配器的领域模型封装方法. 高技术通讯, 2005, 15(2): 34-38
- [3] Gustavson P, Borah J, Chase T, et al. Guide for Base Object Model (BOM) Use and Implementation: [Technical Report]. SISO Base Object Model Product Development Group, March 31 2006. 21-25
- [4] Paul Gustavson, Jake Borah, Tram Chase, et al. Base Object Model (BOM) Template Specification: [Technical Report]. SISO Base Object Model Product Development Group, 2006. 7-15
- [5] Andersson J, Lof S. HLA as conceptual basis for a multi-agent environment; [Technical Report 8TH-CCG-033]. Pitch Kunskapsutveckling AB, 1999. 1-6
- [6] 尹全军, 杜湘瑜, 张琦等. 基于 HLA 的智能 Agent 仿真环境研究. 系统仿真学报, 2005, 17(4): 875-878
- [7] Wu J, Cao X. Intelligent traffic simulation grid based on the HLA and JADE. *High Performance Computing and Applications*, 2010, 5938: 456-464
- [8] FIPA00001L, FIPA Abstract Architecture Specification. <http://www.fipa.org/>: Foundation for Intelligent Physical Agents, 2002
- [9] Guoli, L. Efficient content-based publish/subscribe for distributed workflow management: [Master Dissertation]. Toronto: Department of Computer Science, University of Toronto, 2005. 9-11
- [10] Tang S, Xiao T, Fan W. A collaborative platform for complex product design with an extended HLA integration architecture. *Simulation Modeling Practice and Theory*, 2010, 18(8): 1048-1068

Interoperability extension for high level architecture based on FIPA-ACL

Yue Yingchao, Xiao Tianyuan, Fan Wenhui, Wu Song, Ma Cheng

(State CIMS Engineering Research Center, Tsinghua University, Beijing 100084)

Abstract

Aiming at the disadvantage of lacking flexible interoperation resulting from the fixed federation object model (FOM) mechanism in the high level architecture (HLA) distributed simulation, this paper proposes a method of interoperation extension for HLA distributed simulation based on FIPA-ACL, the agent communication language (ACL) put forward by the Foundation for Intelligent Physical Agents (FIPA). Taking advantage of communication technology in multi-agent systems, this method supports the FIPA-ACL based middleware technology to extend the current fixed FOM mechanism. It was showed to be efficient in communication in distributed simulation, improving the flexibility and dynamics of interoperation, and promoting intelligence in distributed simulation. It has the good succession to its former technology and the lower cost to implement. This method was put into practice in a typical distributed simulation example of a complex product, the high speed train, and the results demonstrated its feasibility and effectiveness.

Key words: the agent communication language (ACL) put forward by the Foundation for Intelligent Physical Agents (FIPA-ACL), high level architecture (HLA) distributed simulation, middleware, interoperation extension