



业务目标模型与业务场景模型的语义一致性分析

李宗花^{1a,2}, 叶正伟^{1b}

(1. 淮阴师范学院 a. 计算机科学与技术学院; b. 城市与环境学院, 江苏 淮安 223300;

2. 东南大学 计算机科学与工程学院, 南京 211189)

摘要:多层次多视图模型是在不确定需求环境下进行业务建模的主要方法,不同层次或不同视图模型之间的语义一致性直接影响业务建模的完整性。鉴于此,设计一种业务目标模型与业务场景模型的语义一致性验证方法。分别以范畴模型和扩展Petri网模型代表业务目标模型和业务场景模型,通过定义形式化业务目标模型的紧邻序列和形式化业务场景模型的执行顺序序列,设计这两种模型之间完全语义一致性、部分语义一致性和弱语义一致性的验证条件和验证步骤。Travel Agency业务系统上的一致性验证结果证明了该方法的可行性和有效性。

关键词:业务目标模型;业务场景模型;范畴论;Petri网;语义一致性

开放科学(资源服务)标志码(OSID):



中文引用格式:李宗花,叶正伟.业务目标模型与业务场景模型的语义一致性分析[J].计算机工程,2019,45(12):308-313.

英文引用格式:LI Zonghua, YE Zhengwei. Semantic consistency analysis of business objective model and business scenario model[J]. Computer Engineering, 2019, 45(12): 308-313.

Semantic Consistency Analysis of Business Objective Model and Business Scenario Model

LI Zonghua^{1a,2}, YE Zhengwei^{1b}

(1a. School of Computer Science and Technology; 1b. School of Urban and Environmental Sciences, Huaiyin Normal University, Huai'an, Jiangsu 223300, China;

2. School of Computer Science and Engineering, Southeast University, Nanjing 211189, China)

[Abstract] Multi-level and multi-view modeling is commonly used for business modeling in scenarios without specific requirements, but the semantic inconsistency between models of different levels or views can affect the integrity of business models. To address the problem, the paper proposes a semantic consistency verification method for business objective model and business scenario model. The method represents the two kinds of models respectively with the category model and extended Petri net model, and defines the nearest neighbor sequence of formalized business objective models and execution order sequence of formalized business scenario models. On this basis, the verification conditions and steps of complete, partial and weak semantic consistency between the two kinds of models are designed. Consistency verification results of Travel Agency business system demonstrate the feasibility and effectiveness of the proposed method.

[Key words] business objective model; business scenario model; category theory; Petri net; semantic consistency

DOI: 10.19678/j.issn.1000-3428.0053025

0 概述

在业务建模过程中,需求的不确定性是普遍存在的,如何处理这一问题是需求工程的核心。结合业务目标模型^[1]、业务场景模型^[2]和业务过程模

型^[3-4]等方法的多层次多视图模型,可从系统用户、业务分析人员和软件开发人员等角度建立完整的业务过程模型。业务目标模型用于获取业务和系统的目标,制定实现目标的可选择方案以及确定目标贡献值的大小^[5],因此,目标模型可以识别系统

基金项目:国家自然科学基金(61902141, 41471425);教育部人文社会科学研究青年基金(19YJCZH095);江苏省高校自然科学研究面上项目(18KJB520006);淮安市科技计划项目(HABZ201701)。

作者简介:李宗花(1981—),女,副教授、博士,主研方向为模型驱动开发、形式化方法;叶正伟,教授、博士。

收稿日期:2018-10-30

修回日期:2018-12-29

E-mail: leeleaf@163.com

的目标和暂时性的粗粒度需求,建立业务系统的初始目标模型。业务场景模型从参与者的角度出发,关注实现业务功能的行为过程,从而定义系统的交互式行为模型^[6]。业务目标模型和业务场景模型是相辅相成的,场景交互细节的确定有助于进一步识别系统的业务目标和附加场景,以明确系统的需求。因此,业务目标模型与业务场景模型之间的语义一致性对于构建正确、有效的业务模型具有重要作用。

模型的一致性问题是模型形式化验证的焦点,Petri网作为被广泛使用的形式化模型,适用于描述系统的各种“流”信息^[7],其不但可以验证进程间的行为一致性和语义一致性^[8],还可以验证业务模型的时序约束一致性^[9]。而本体模型则在结构描述上具有很好的优势,其不但可以验证不同演化模型之间的一致性^[10],还可以利用通用本体验证多层本体模型的语义一致性^[11]。在模型转换的研究中,可以从模型语法关系、语义关系和功能性行为3个方面验证源模型与目标模型之间的一致性^[12],或者定义形式化模型转换规则,运用转换合约的逻辑形式化描述来验证和执行模型转换,从而保证源模型与目标模型的一致性^[13]。对于UML(Unified Modeling Language)状态模型,可通过状态机分析其一致性^[14-15]。然而,上述模型一致性验证研究过度依赖一对一的转换规则,忽略了模型之间不能用一对一转换规则表达的相关元素。

本文所述的语义一致性是指在某业务系统中,同一业务活动在业务目标模型和业务场景模型中具有相同的行为。因此,本文基于形式化业务目标模型和形式化业务场景模型,在一对一模型一致性验证的基础上,研究上述两个业务模型之间的协约性问题,进而更加准确地验证模型间的语义一致性。

1 相关模型

1.1 形式化业务目标模型

业务系统的目标模型主要描述不同终端用户和业务协作者的业务目标以及这些业务目标之间的约束关系^[5]。范畴论以抽象方法描述数学结构,从而表示结构之间的相互关系,利用“物件”和“态射”来形式化某个特定的范畴,其结构上的优势可应用于业务目标模型。因此,本文利用范畴论的“物件”表示业务目标模型中的参与对象,利用“态射”描述各业务目标之间的意图关系。

定义1 一个业务目标图可以定义为一个系统^[16],即:

$$G = (A, G_A, G_N, s, t, m_A, m_N)$$

其中:

1) A 表示 Actor,即目标系统中的参与对象,对应业务系统中的业务参与者。

2) G_A 表示射的集合,用箭头符号表示, $\forall a_i, \exists a_0 \in G_A$, 则 $a_0 :: = a_{dc} | a_c | a_{dp}$ 。其中, a_{dc} 代表射的类型为 Decomposition,表示将业务目标结点分解为若干个业务任务结点, a_c 代表射的类型为 Contribution,表示某个任务结点贡献于其他任务结点或者目标结点, a_{dp} 代表射的类型为 Dependency,表示某个任务结点与其他任务结点或目标结点存在依赖关系。

3) G_N 表示结点的集合,用圆形符号表示,对应业务系统中的业务服务或业务活动。 $\forall n_i, \exists n_0 \in G_N, n_0 = (n_{name}, n_{type})$, 则 $n_{type} :: = n_g | n_t | n_{sg} | n_r$, 其中, n_{name} 表示结点名称, n_{type} 表示结点类型, n_g 代表结点类型为 goal,表示业务系统中的功能性目标对应的业务服务概念, n_t 代表结点类型为 task,对应业务的任务概念, n_{sg} 代表结点类型为 soft goal,描述业务系统中非功能性方面的需求, n_r 代表结点类型为 resource,表示业务系统中实现业务目标所需的相关资源。上述结点往往属于某个参与者组件,因此,每个参与者对应一个结点集合,即 $A_i \Rightarrow G_{N_i}$ 。

4) $s: G_N \rightarrow G_A$ 和 $t: G_A \rightarrow G_N$ 分别表示源映射和目标映射, $m_A: G_A \rightarrow G_A$ 表示射字符集的映射, $m_N: G_N \rightarrow G_N$ 表示结点字符集的映射。

定义2 对于任意目标模型系统 $G = (A, G_A, G_N, s, t, m_A, m_N)$, 有 $a_1, a_2, \dots, a_n \in G_A$ 存在射集合 $a_1: s(a_1) \rightarrow t(a_1), a_2: t(a_1) \rightarrow t(a_2), \dots, a_n: t(a_{n-1}) \rightarrow t(a_n)$, 则存在一个目标紧邻序列 $Neighbor-S = \{s(a_1) \rightarrow s(a_2) \rightarrow \dots \rightarrow s(a_n)\}$, 代表目标模型系统中为实现一个任务的动作序列。同时, 存在一个紧邻序列集合 $\Sigma Neighbor-S = \{s(a_1), s(a_2), \dots, s(a_n)\}$, 表示该紧邻序列中所有结点的集合。

1.2 形式化业务场景模型

Petri网作为一个流类型的形式化模型,适合用来描述系统的行为特征,目前已被广泛应用于工作流的形式化任务中。Petri网的建模工具支持模型的自动分析和设计,并且具有可达性,能够进行死锁检测和边界分析等^[16-18],因此,可用Petri网来定义场景流的变化行为。

定义3 业务场景模型被定义为由一个9元组组成的EPN(Extended Petri Nets)模型,具体如下:

$$EPN = \langle IP, OP, ST, BT, IA, OA, M_0, SN, OI \rangle$$

其中, $IP = \{ip_1, ip_2, \dots, ip_n\}$ 表示有限个内部场景流的集合, $OP = \{op_1, op_2, \dots, op_n\}$ 表示有限个外部场景流的集合, $ST = \{st_1, st_2, \dots, st_n\}$ 表示有限个开始、结束和网关元素的集合, st_i 表示业务场景中的某个开始结点、结束结点或网关结点, $BT = \{bt_1, bt_2, \dots, bt_n\}$ 表示有限个责任点元素的集合, $IA \subseteq (IP \times ST) \cup (IP \times BT) \cup (OP \times ST) \cup (OP \times BT)$ 表示一组输入弧, $OA \subseteq (ST \times IP) \cup (ST \times OP) \cup (BT \times IP) \cup (BT \times OP)$ 表示一组输出弧, M_0 表示初始标识, $SN = \langle IP, OP, ST, BT, IA, OA, M_0, SN, OI \rangle_{sub}$ 表示子业务场景模型, OI

表示为业务场景模型中的构件元素,其为包括一系列责任点、网关、内部场景流的集合。

定义 4 对于任意 Petri 网模型 $EPN = \langle IP, OP, ST, BT, IA, OA, M_0, SN, OI \rangle$, 若 $IP(bt_1 \times ip_1) = IP(ip_1 \times bt_2)$ 或者 $OP(bt_1 \times op_1) = IP(op_1 \times bt_2)$, 则责任点 bt_1 和 bt_2 是一个执行顺序对, 表示为 $bt_1 \rightarrow bt_2$, 即 bt_1 在 bt_2 之前实施, 因此, (BT, \rightarrow) 应具有无环性、组合性及连通性的特点。对于多个执行顺序对, 其与开始执行序列对和结束执行序列对可以组成一个执行顺序序列 $P_i = \{st_1 \rightarrow bt_1 \rightarrow bt_2 \rightarrow \dots \rightarrow bt_n \rightarrow st_n\}$ 。

2 语义一致性分析

若一个目标没有涉及任何场景, 则表明该业务目标可能是不正确的, 或者该目标超出了目标模型的描述范围; 如果一个场景对任何目标既不是必须的, 也不没有任何贡献, 则表明该场景设计有误或者目标模型需要改进。因此, 对目标模型与场景模型间的语义一致性问题进行研究是非常有必要的。

2.1 基本思路

文献[12]认为模型验证应该考虑模型整体而不是模型中的个体因素。例如, 目标模型是在源模型的基础上执行模型元素的组合、合并及编织等操作, 这就需要针对不同情形的模型转换操作定义一致性检验方法。因此, 本文定义的业务目标模型与业务场景模型的一致性验证是以这两个模型中的每一个元素可连接且无二义性为前提, 并从完全语义等价、部分语义等价和弱语义等价 3 个方面进行语义验证。

2.2 验证方法

本文首先定义形式化业务目标模型的紧邻序列集合和形式化业务场景模型的执行顺序序列集合, 然后比较这 2 个集合中的部分序列是否完成相同的业务服务, 从而定义业务目标模型与业务场景模型之间的一致性。

定义 5 给定形式化场景模型 $PN_1 = \langle IP, OP, ST, BT, IA, OA, M_0, SN, OI \rangle$, 存在静态事件 $st_1, st_2, \dots, st_n \in ST$, 动态事件 $bt_1, bt_2, \dots, bt_n \in BT$, 子网模型 $sn_1, sn_2, \dots, sn_n \in SN$ 。由于静态事件在 EPN 模型中表示 Petri 网模型的开始、结束和路径选择, 没有实际的动作行为, 因此定义 $\Sigma C_2 = \Sigma BT \cup \Sigma SN$ 表示 PN_1 模型中所有有限个动作的集合。其中, $\Sigma BT = \{bt_1, bt_2, \dots, bt_n\}$ 表示有限个责任点集合, $\Sigma SN = \{sn_1, sn_2, \dots, sn_n\}$ 表示有限个子网模型的集合。

定义 6 给定目标模型 CG_1 和场景模型 PN_1 , 定义 $\Sigma C = \Sigma C_1 \cap \Sigma C_2$, 其中, $\Sigma C_1 = \{ng_1, ng_2, \dots, ng_n, nt_1, nt_2, \dots, nt_n\}$ 表示目标模型 CG_1 中所有结点的集合, ΣC_2 表示场景模型 PN_1 中所有动作结点的集合, ΣC 表示目标模型 CG_1 与场景模型 PN_1 中具有相同行为的结点集合。

2.3 验证步骤

根据定义 5 和定义 6, 在多层次多视图业务模型

中, 业务目标模型 CG_1 与业务场景模型 PN_1 之间的语义一致性验证步骤如下:

步骤 1 利用定义 5 和定义 6 计算目标模型与场景模型具有相同行为的结点集合 ΣC 。

步骤 2 利用定义 2 计算目标模型系统的所有紧邻序列集合 $\Sigma Neighbor-S$, 并通过定义 4 计算出形式化场景模型执行顺序序列集合 ΣP 。

步骤 3 $\exists Neighbor-S_i \in \Sigma Neighbor-S, \exists P_i \in \Sigma P$, 执行 $\Sigma Neighbor-S_i \cap \Sigma C$ 运算并确定有效的紧邻序列。同时, 执行 $\Sigma P_i \cap \Sigma C$ 运算并确定有效的执行顺序序列。iff: $\Sigma Neighbor-S_i \cap \Sigma C = \Sigma Neighbor-S_i$ 且 $\Sigma P_i \cap \Sigma C = \Sigma P_i$, 则转向步骤 4。iff: $\Sigma Neighbor-S_i \cap \Sigma C \neq \emptyset$ 且 $\Sigma P_i \cap \Sigma C \neq \emptyset$, 则转向步骤 5。iff: $\Sigma Neighbor-S_i \cap \Sigma C \neq \emptyset$ 或 $\Sigma P_i \cap \Sigma C \neq \emptyset$, 则表明业务目标模型 CG_1 与业务场景模型 PN_1 之间语义不一致。

步骤 4 iff: $\Sigma Neighbor-S_i = \Sigma P_i$ 且 $Neighbor-S_i = P_i$, 则称业务目标模型 CG_1 与业务场景模型 PN_1 之间保持完全语义一致性。iff: $\Sigma Neighbor-S_i = \Sigma P_i$ 但 $Neighbor-S_i \neq P_i$, 则称业务目标模型 CG_1 与业务场景模型 PN_1 之间保持部分语义一致性。

步骤 5 iff: $\Sigma Neighbor-S_i \cap \Sigma P_i \neq \emptyset$, 则称业务目标模型 CG_1 与业务场景模型 PN_1 之间保持弱语义一致性。

由上述定义可以得到以下内容:

1) 完全语义一致性定义表明, 业务目标模型中的一个紧邻序列与业务场景模型中的一个执行顺序序列, 其结点集合相同, 序列结构相同。完全语义一致性表现为底层的业务场景模型没有对顶层的业务目标模型进行修改, 说明系统的顶层业务目标清晰、完整, 业务目标之间的因果关系明确。然而, 这种完全语义一致性也可能表明业务场景建模的交互细节还不够完整。

2) 部分语义一致性定义表明, 业务目标模型中的紧邻序列与业务场景模型中的执行顺序序列, 其结点集合相同, 但序列结构不同。部分语义一致性表现为底层的场景模型对顶层目标模型中的目标与任务之间的因果关系进行了修改, 说明在顶层业务目标模型中, 部分业务目标及其因果关系不明确, 有待进一步识别。当出现部分语义一致的情况时, 可利用业务场景模型的交互细节进一步识别系统的业务目标, 完善业务目标模型, 从而实现业务目标模型与业务场景模型的完全语义一致。

3) 弱语义一致性定义表明, 业务目标模型中的目标和任务结点与业务场景模型中的责任点存在相互交织的情况。弱语义一致性表现为底层的业务场景模型对业务目标模型中的目标和任务进行进一步完善和细化, 说明在业务目标建模阶段, 系统的业务目标不清晰, 业务目标的实施方案不完整。然而, 这种弱语义一致性也可能表明在业务场景建模阶段, 系统

的业务目标、实现业务目标的方案被大量修改。当出现这种弱语义一致性时,业务场景模型必须将其修改内容反馈至业务目标模型,进一步完善业务目标模型和业务场景模型,以满足更高的一致性要求。

3 验证实例

万维网联盟(World Wide Web Consortium, W3C)对 Travel Agency 业务系统^[19]的住宿服务场景进行了详细描述。首先,出行者向旅游网站提交自己的住宿计划,包括目的地、入住时间和离开时间、住宿酒店标准等信息,然后,旅游网站接收用户的住宿计划并进行评估,找到能够为该用户提供服务的酒店,并将满足条件的信息整理后反馈给用户。用户一旦选择了某一项服务,需要向旅游网站提供个人的信

用卡信息,当相关金融公司验证该支付正确有效以后,旅游网站要求酒店确认订单并发消息通知用户。根据以上业务场景的信息,利用文献[16]的 GSP 业务建模方法得到其形式化业务目标模型和形式化业务场景模型,分别如图1和图2所示。其中,edc和edp分别表示业务目标结点之间的分解和依赖关系。

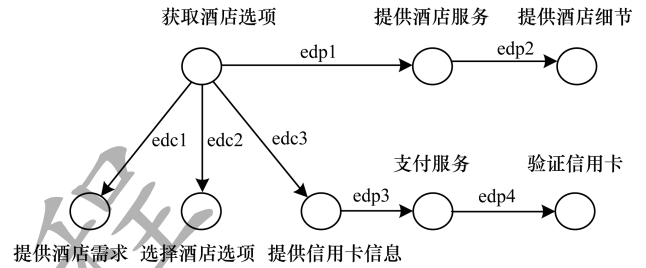


图1 住宿服务的形式化业务目标模型

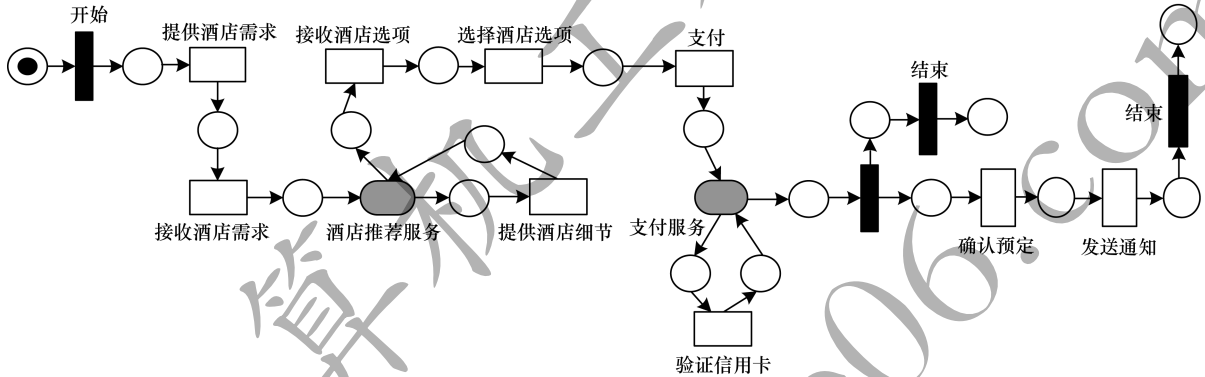


图2 住宿服务的形式化业务场景模型

根据第2节设计的步骤进行实例验证,具体如下:

步骤1 根据图1所示的业务目标形式化模型,得到所有结点集合 $\Sigma C_1 = \{\text{获取酒店选项, 提供酒店服务, 提供酒店细节, 提供酒店需求, 选择酒店选项, 提供信用卡信息, 支付服务, 验证信用卡}\}$ 。根据图2所示的业务场景形式化模型,得到所有结点集合 $\Sigma C_2 = \{\Sigma BT \cup \Sigma SN | BT \in PN_1 \& \& SN \in PN_1\} = \{\text{提供酒店需求, 接收酒店选项, 选择酒店选项, 支付, 接收酒店需求, 酒店推荐服务, 提供酒店细节, 支付服务, 验证信用卡, 确认预定, 发送通知}\}$ 。这2个模型的共有结点集合 $\Sigma C = \Sigma C_1 \cap \Sigma C_2 = \{\text{提供酒店细节, 支付服务, 提供酒店需求, 选择酒店选项, 验证信用卡}\}$ 。

步骤2 获取目标模型系统的紧邻序列集合 $\Sigma Neighbor-S = \{\text{获取酒店选项} \rightarrow \text{提供酒店需求, 获取酒店选项} \rightarrow \text{选择酒店选项, 获取酒店选项} \rightarrow \text{提供酒店服务} \rightarrow \text{提供酒店细节, 获取酒店选项} \rightarrow \text{提供信用卡信息} \rightarrow \text{支付服务} \rightarrow \text{验证信用卡}\}$ 。同时,获取形式化场景模型的执行顺序序列集合 $\Sigma P = \{\text{开始} \rightarrow \text{提供酒店需求} \rightarrow \text{接收酒店需求} \rightarrow \text{酒店推荐服务} \rightarrow \text{提供酒店细节} \rightarrow \text{酒店推荐服务} \rightarrow \text{接收酒店选项} \rightarrow \text{选择酒店选项} \rightarrow \text{支付} \rightarrow \text{支付服务} \rightarrow \text{验证信用卡} \rightarrow \text{支付服务} \rightarrow \text{确认预定} \rightarrow \text{发送通知} \rightarrow \text{结束, 开始} \rightarrow \text{提供酒店需求} \rightarrow \text{接}$

收酒店需求 \rightarrow 酒店推荐服务 \rightarrow 提供酒店细节 \rightarrow 酒店推荐服务 \rightarrow 接收酒店选项 \rightarrow 选择酒店选项 \rightarrow 支付 \rightarrow 支付服务 \rightarrow 验证信用卡 \rightarrow 支付服务 \rightarrow 结束 $\}$ 。

步骤3 执行 $\Sigma Neighbor-S_i \cap \Sigma C$ 运算, 可得:
 $\Sigma Neighbor-S_1 \cap \Sigma C = \{\text{提供酒店需求}\} \neq \emptyset$;
 $\Sigma Neighbor-S_2 \cap \Sigma C = \{\text{选择酒店选项}\} \neq \emptyset$;
 $\Sigma Neighbor-S_3 \cap \Sigma C = \{\text{提供酒店细节}\} \neq \emptyset$;
 $\Sigma Neighbor-S_4 \cap \Sigma C = \{\text{支付服务, 验证信用卡}\} \neq \emptyset$ 。
 上述结果不为空的紧邻序列包括图3所示的4个过程。

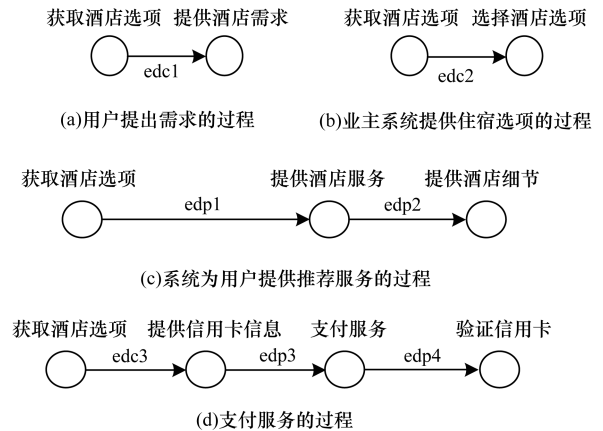


图3 Travel Agency 形式化目标模型的紧邻序列

根据业务分析员的领域知识,图 3(a)和图 3(b)分别表示用户提出自己的住宿计划和业务系统为用户提供的住宿选项,图 3(c)和图 3(d)分别表示 Travel Agency 系统为用户提供了酒店推荐服务和支付服务。根据 Travel Agency 系统的初始需求描述,这 4 条有效的紧邻序列能够完全覆盖该系统需要实现的所有业务功能。

步骤 4 执行 $\Sigma P_i \cap \Sigma C$ 运算,可得: $\Sigma P_1 \cap$

$\Sigma C = \Sigma P_2 \cap \Sigma C = \{\text{提供酒店细节, 支付服务, 提供酒店需求, 选择酒店选项, 验证信用卡}\} \neq \emptyset$ 。形式化场景模型的执行顺序序列如图 4(a)和图 4(b)所示,表示实现酒店推荐服务的业务交互场景,其动作路径完整地描述了为实现住宿推荐服务,所有业务利益相关者的操作。因此,这 2 条动作顺序序列是有效的。同时由上述分析可知, $\Sigma Neighbor-S_i \cap \Sigma C \neq \Sigma Neighbor-S_i \neq \emptyset$ 且 $\Sigma P_i \cap \Sigma C \neq \Sigma P_i \neq \emptyset$, 因此,转向步骤 5。

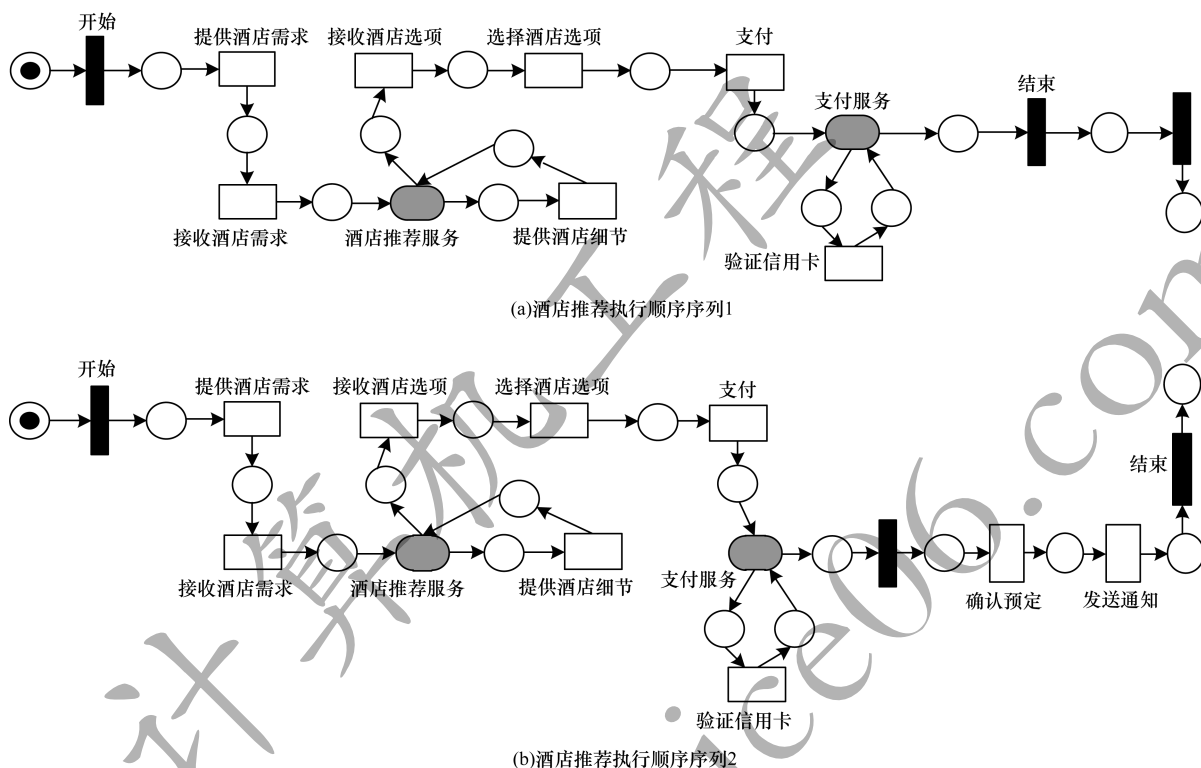


图 4 Travel Agency 形式化场景模型的部分执行顺序序列

步骤 5 执行 $\Sigma Neighbor-S_i \cap \Sigma P_i$ 运算。图 3(a)、图 3(b)所示的紧邻序列与图 4(a)所示的动作顺序序列执行交运算后的结果为: $\{\text{提供酒店需求}\} \neq \emptyset$ 和 $\{\text{选择酒店选项}\} \neq \emptyset$ 。图 3(c)、图 3(d)所示的紧邻序列与图 4(b)所示的动作顺序序列执行交运算后的结果为: $\{\text{提供酒店细节}\} \neq \emptyset$ 和 $\{\text{支付服务, 验证信用卡}\} \neq \emptyset$ 。

由上述形式化验证过程可知,图 1 所示的形式化目标模型和图 2 所示的形式化场景模型满足弱语义一致性条件。从图 2 可以看出,酒店推荐服务和支付服务是子 Petri 网,表明图 1 中提供酒店服务和支付服务这 2 个业务目标需要进一步细化。以支付服务为例,图 5 给出该目标细化后的子目标形式化模型,对应的业务场景形式化模型如图 6 所示,其中,ec 表示业务目标结点之间的贡献关系。可以看出,图 5 存在一条紧邻序列与图 6 的执行序列相等,表明它们的结点序列结构相同,是完全语义一致性的。

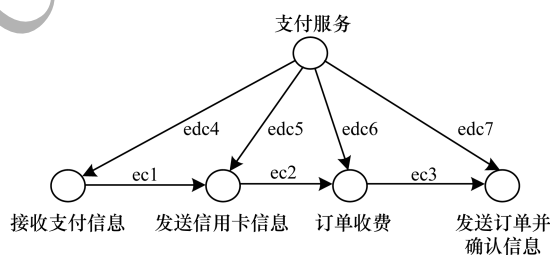


图 5 支付服务细化后的形式化子业务目标模型

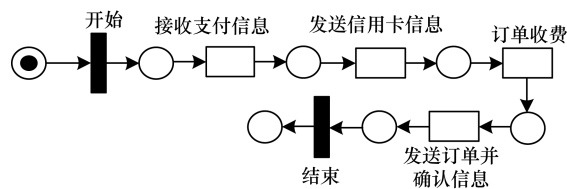


图 6 支付服务细化后的形式化子业务场景模型

图 1 与图 2 的弱语义一致性表明,系统在目标建模阶段对业务目标及业务目标的贡献大小标识不清,业务目标的实施方案定义不准确。因此,可以设计模型反馈机制,将场景模型中识别的业务需求反

馈至顶层目标模型中,使得目标模型更加完善,从而满足更高级别的一致性。无论多视图多层次框架中的业务场景模型如何修改和变化,都可以根据本文定义的一致性验证策略考察业务目标模型与细化后的业务场景模型之间的语义一致性。

4 结束语

针对多层次多视图业务模型中不同层次或不同视图模型之间的动态一致性问题,本文提出一种业务目标模型与业务场景模型语义一致性验证方法。定义形式化业务目标模型与形式化业务场景模型的语法结构,在此基础上提出两者之间满足完全语义一致性、部分语义一致性和弱语义一致性的条件,并设计验证步骤。对 Travel Agency 实例中业务目标模型与业务场景模型之间的一致性进行验证,结果表明,该验证方法不仅能够根据一对一的映射规则分析源模型与目标模型的一致性,而且可以利用不同条件下的语义分析源模型与目标模型之间的语义一致性。但是,本文主要从形式化语义的角度分析一致性验证过程,且采用手工方式验证,同时在提取业务目标模型的紧邻序列和业务场景模型的执行顺序序列时存在提取无效的情况,因此,实现语义一致性验证的自动执行和提高模型一致性验证效率将是下一步研究的重点。

参考文献

- [1] CASTRO V D, MARCOS E, VARA J M. Applying CIM-to-PIM model transformations for the service-oriented development of information systems [J]. Information and Software Technology, 2011, 53 (1): 87-105.
- [2] ZDRAVKOVIC J, ILAYPERUMA T. A model-driven approach for designing e-services using business ontological frameworks [C]//Proceedings of the 14th IEEE International Enterprise Distributed Object Computing Conference. Washington D. C., USA: IEEE Press, 2010: 121-130.
- [3] PAHL C. Semantic model-driven architecting of service-based software systems [J]. Information and Software Technology, 2007, 49 (8): 838-850.
- [4] GRONER G, BÖSKOVIC M, PARREIRAS F S, et al. Modeling and validation of business process families [J]. Information Systems, 2013, 38 (5): 709-726.
- [5] VAN-LAMSWEERDE A. Goal-oriented requirements engineering: a guided tour [C]//Proceedings of the 15th IEEE International Symposium on Requirements Engineering. Washington D. C., USA: IEEE Press, 2001: 249-262.
- [6] ALIX T, ZACHAREWICZ G. Product-service systems scenarios simulation based on G-DEVS/HLA: generalized discrete event specification/high level architecture [J]. Computers in Industry, 2012, 63 (4): 370-378.
- [7] WANG Yuan, FAN Yushun. A method of time constraint workflow model analysis and verification [J]. Journal of Software, 2007, 18 (9): 2153-2161. (in Chinese)
- [8] 王远, 范玉顺. 工作流时序约束模型分析与验证方法 [J]. 软件学报, 2007, 18 (9): 2153-2161.
- [9] BACKER D M, SNOECK M, MONSIEUR G, et al. A scenario-based verification technique to assess the compatibility of collaborative business processes [J]. Data and Knowledge Engineering, 2009, 68 (6): 531-551.
- [10] DU Yanhua, YU Ze. Consistency validation and modification of Guard-Stage-Milestone models under temporal constraints [J]. Systems Engineering - Theory and Practice, 2016, 36 (8): 2108-2126. (in Chinese)
- [11] 杜彦华, 于泽. 时序约束下 Guard-Stage-Milestone 业务模型的一致性验证与异常处理 [J]. 系统工程理论与实践, 2016, 36 (8): 2108-2126.
- [12] HE Wenmin, SHEN Guohua, HUANG Zhiqiu, et al. Ontology-based consistency verification for evolving feature models [J]. Application Research of Computers, 2013, 30 (7): 2072-2076. (in Chinese)
- [13] 何文民, 沈国华, 黄志球, 等. 基于本体的特征模型演化的一致性验证 [J]. 计算机应用研究, 2013, 30 (7): 2072-2076.
- [14] LI Jinhai, MA Yunlei, SUN Lingfang, et al. Research on construction method of multi-layer ontology meta-model based on semantic consistency [J]. Journal of the China Society for Scientific and Technical Information, 2017, 36 (5): 494-502. (in Chinese)
- [15] 李金海, 马云蕾, 孙玲芳, 等. 基于语义一致性的多层本体元模型构建方法研究 [J]. 情报学报, 2017, 36 (5): 494-502.
- [16] CALEGARI D, SZASZ N. Verification of model transformations a survey of the state-of-the-art [J]. Electronic Notes in Theoretical Computer Science, 2013, 292: 5-25.
- [17] BRAGA C, SANTOS C, DA-SILVA V T. Consistency of model transformation contracts [J]. Science of Computer Programming, 2014, 92 (6): 86-104.
- [18] ZHANG Jian, WU Jun, FANG Jinglong. Automatic model transformation and verification method of consistency [J]. Computer Engineering and Design, 2017, 38 (9): 2407-2413. (in Chinese)
- [19] 张建, 吴俊, 方景龙. 模型自动转换与一致性验证方法 [J]. 计算机工程与设计, 2017, 38 (9): 2407-2413.
- [20] LUCAS F J, MOLINA F, TOVAL A. A systematic review of UML model consistency management [J]. Information and Software Technology, 2009, 51 (12): 1631-1645.
- [21] LI Zonghua, ZHOU Xiaofeng, GU Aihua, et al. A complete approach for CIM modelling and model formalizing [J]. Information and Software Technology, 2015, 65: 39-55.
- [22] YOO T, JEONG B, CHO H. A Petri nets based functional validation for services composition [J]. Expert Systems with Applications, 2010, 37 (5): 3768-3776.
- [23] KOSTIN A E. Reachability analysis in T-invariant-less Petri nets [J]. IEEE Transactions on Automatic Control, 2003, 48 (6): 1019-1024.
- [24] ARKIN A, ASKARY S, FORDIN S, et al. Web Service Choreography Interface (WSCI) 1.0 [EB/OL]. [2018-10-06]. <https://www.w3.org/TR/wsci/>.