张赵琪,倪枫,刘文诚,等. 基于场景更新的业务流程迭代演进方法[J]. 智能计算机与应用,2025,15(4):25-34. DOI:10.20169/j.issn.2095-2163.24110601

基于场景更新的业务流程迭代演进方法

张赵琪[1], 倪 枫[1], 刘文诚[2], 刘 姜[1], 陈年年[1], 周兴郡[1]
(1 上海理工大学 管理学院, 上海 200093; 2 上海理工大学 环境与建筑学院, 上海 200093)

摘 要: 新科技浪潮下,复杂多变的应用场景对流程模型的灵活性提出了更高的要求,流程需要不断演进以适应外部发展,而业务流程建模符号(BPMN)由于缺乏语义规范和形式化分析技术,难以支持演进过程中模型的有效性与正确性。针对这一问题,提出一种基于场景更新的业务流程迭代演进方法,旨在通过可执行模型动态调整模型结构,以支持场景更新后演进行为的验证与分析。该方法首先面向场景建立 BPMN 协作模型,并设计演进检查算法识别变更元素;其次,通过形式化映射和检查结果,实现着色 Petri 网(CPN)模型上的更新演进。此外,该方法引入了模块化设计,将复杂 BPMN 模型拆分进行模块化映射;最后,通过一个智能课程推荐系统案例验证了方法的有效性。

关键词: 业务流程; BPMN; 场景更新; 着色 Petri 网; 可执行建模; 迭代演进
中图分类号: TP319 **文献标志码**: A **文章编号**: 2095-2163(2025)04-0025-10

Iterative evolutionary approach to business processes based on scenario updates

ZHANG Zhaoqi[1], NI Feng[1], LIU Wencheng[2], LIU Jiang[1], CHEN Niannian[1], ZHOU Xingjun[1]
(1 Business School, University of Shanghai for Science and Technology, Shanghai 200093, China;
2 School of Environment and Architecture, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: Under the wave of new technology, complex and changing application scenarios put forward higher requirements on the flexibility of process models, and processes need to evolve continuously to adapt to external development, while Business Process Modeling Notation (BPMN) is difficult to support the validity and correctness of models in the process of evolution due to the lack of semantic specification and formal analysis techniques. To address this problem, an iterative business process evolution method based on scenario updating is proposed, aiming at dynamically adjusting the model structure through an executable model to support the validation and analysis of evolutionary behavior after scenario updating. The method firstly builds a BPMN collaboration model for the scenario and designs an evolution checking algorithm to identify the change elements; secondly, it realizes the update evolution on the Colored Petri Net (CPN) model by formalizing the mapping and checking results. In addition, the method introduces a modular design to split the complex BPMN model for modular mapping. Finally, the effectiveness of the method is verified by a case study of an intelligent course recommendation system.

Key words: business process; BPMN; scenario update; Colored Petri Net; executable modeling; iterative evolution

0 引 言

随着新质生产力的发展,技术革新与企业智能化升级对流程的灵活性提出了更高的要求[1],从传统办公自动化到区块链协同平台,从云端大数据处理[2]到物联网建模[3],越来越复杂的应用场景,使得业务流程模型需要不断演进以适应外界需求,在这样的时代背景下,业务流程演进不仅是信息化资源配置的基础,更是应对产业转型升级的关键。

业务流程建模符号(BPMN)经常用于系统早期流程设计,在描绘业务逻辑和过程交互方面有着重要作用[4],然而由于缺乏形式化语义描述,BPMN 很

基金项目: 国家自然科学基金(12371508); 教育部产学合作协同育人项目(220603760210846); 上海市"大学生创新创业训练计划"资助项目(SH2022072)。
作者简介: 张赵琪(2000—),女,硕士研究生,主要研究方向:系统科学,系统分析与集成; 刘文诚(2000—),男,硕士研究生,主要研究方向:系统建模与仿真,大数据分析; 刘 姜(1983—),女,博士,副教授,主要研究方向:复杂系统理论与方法,符号代数计算; 陈年年(2003—),女,本科生,主要研究方向:管理科学,系统科学; 周兴郡(2003—),女,本科生,主要研究方向:管理科学,系统科学。
通信作者: 倪 枫(1982—),男,博士,副教授,主要研究方向:系统科学,系统分析与集成。Email:nifeng@usst.edu.cn。
收稿日期: 2024-11-06

难评估模型的正确性和有效性[5],难以支持模型的动态演进行为验证,进而保证演进的有效性和高效性。为此,许多研究提出建立形式化模型,目前,BPMN 模型的形式化执行语义可以大致分为 2 类。一类是使用纯数学方法(如通信顺序过程(CSP)[6]、π 演算[7]等),另一类是使用 Petri 网的工作流建模语言[8],如 YAWL[9],CPN[10]等。

在模型演进方面,Song 等学者[11]从静态演化和动态演化两个方面去详细梳理了流程演进和过程感知系统。郑伟波等学者[12]从服务层和流程层两个方面分析服务系统演化问题,提出双层依赖关系模型。Ortiz 等学者[13]从 BPMN 片段和全局模型两个角度提出一种管理微服务变更传播的协议,旨在确保这些变更能够无缝地集成到其余微服务的 BPMN 片段和全局 BPMN 模型中。这些分类为深入研究模型演进提供了多维度的视角和框架。

在流程演进模型上,尤殿龙等学者[14]利用互模拟理论,从不同层面研究了业务流程演进判定规则,提出了一种识别受影响服务的策略。文献[15]设计、提出和分析了一种支持业务架构(EA)模型演化的方法,在定义业务架构组件依赖关系基础上,建立了相应的演进模型。Yadav 等学者[16]基于图形分析设计了流程架构演进管理工具,以支持整体演进;但这些方法主要侧重于服务层静态模型的分析,并未涉及具体场景的建模和可执行模型,无法支持动态演进行为的验证。本文从具体场景建模出发,在建立 BPMN 协作模型基础上细化了演进操作定义。

此外,在可执行模型的演进方面,胡强等学者[17]以逻辑 Petri 网为建模工具,面向不同需求场景构建流程结构演进运算,并建立了形式化流程模型。Krishna 等学者[18]探讨了从 BPMN 到 LNT 过程代数和 LTS 形式模型的转换方法,使用过程代数定义了不同类型的演进,并提出了一种检测业务流程演进的方法。文献[19]设计了一种检查算法来识别原始 BPMN 模型和更新后的 BPMN 模型之间的差异,并定义了扩展 Petri 网模型的演化规则来表示 BPMN 模型的变化,避免了模型混淆和令牌的丢失。此外,何海洋等学者[20]提出一种采用高阶 π 演算的构件演化模型,细化了演进行为定义并建立演算规则,实现形式化分析。

本文在可执行模型演进方法的基础上提出了一种场景更新驱动的业务流程迭代演进方法。首先,在定义演进行为与规则的基础上,面向场景构建了 BPMN 协作模型。其次,采用形式化方法将 BPMN 模型映射至 CPN 模型。基于演进检测算法输出,定义了 CPN 模型的演进规则,并成功建立了 CPN 演进模型,从而验证了模型演进的有效性。此外,该方法引入了模块化设计理念,不仅有效降低了业务流程演进的复杂度,同时提高了演进效率。

本文内容主要组织结构如下:第一部分介绍了相关工作;第二部分概述了相关概念与定义;第三部分概述了演进行为与流程演进框架;第四、五部分细述了 BPMN 与 CPN 模型的演进过程;第六部分以智慧选课系统为例进行模型方法的验证;第七部分进行文章小结。

1 概念与定义

本节将重点阐述 BPMN、Petri 网以及着色 Petri 网(CPN)的相关概念。

1.1 BPMN

BPMN(Business Process Model and Notation)是业务流程建模的 OMG 标准,可以用于描述业务流程中的各种事务和决策[21],如图 1 所示。BPMN 核心元素包括事件、活动、网关、数据连接元素、泳道。其中,活动、事件、网关又称为流对象,数据包括数据对象和数据流;连接元素包括序列流、消息流和关联。

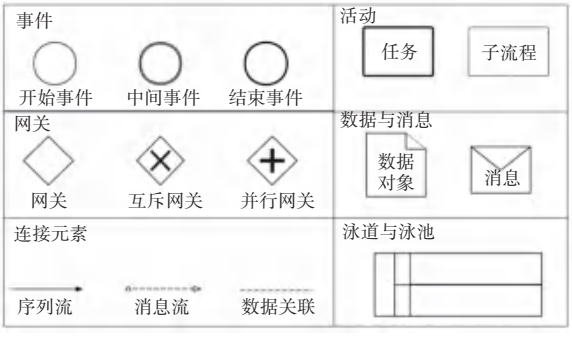


图 1 BPMN 元素

Fig. 1 BPMN element

定义 1 BPMN 协作模型 设一个 BPMN 协作模型为一个三元组 $BP=\{N,F,P_L\}$,三元组中包含如下内容:

(1) N 表示流对象节点集,$N=A\cup G\cup E$ 且 $A\cap G\cap E=\phi$。其中,A 表示活动集合,包括任务和子流程;G 表示网关集合,包括互斥网关、并行网关;E 表示事件集合,包括开始事件、中间事件和结束事件;

(2) F 表示连接元素集,$F=S\cup M$ 且 $S\cap M=\phi$。其中,S 表示序列流,M 表示消息流,序列流可以

表示流对象之间控制结构关系。

(3) $P_L = \{L_1, L_2, \cdots, L_k\}$ 表示泳池，其中 L 表示泳道。

1.2 Petri 网

一个普通 Petri 网(OPN)由库所(Places)、变迁(Transitions)和弧(Arcs)组成，通过令牌(Tokens)在库所与变迁之间的动态流动，能够有效反映系统状态的变化。着色 Petri 网(Colored Petri Net，CPN)是对普通 Petri 网的扩展，能够通过颜色集来区分不同类型的信息，从而提升模型的表达能力，降低模型的复杂度，避免状态空间爆炸等问题[22]。其定义如下：

定义 2　CPN 模型　设一个六元组 $CPN = (\Sigma, P, T, A_r, f, M_0)$。六元组中，包含如下内容：

(1) Σ (color)表示颜色集，且非空有限。

(2) P (place)表示有限非空库所集，$P = \{p_1, p_2, \cdots, p_n\}$，且 $P \neq \phi$ 表示状态或条件。

(3) T (transition)为有限非空变迁集，$T = \{t_1, t_2, \cdots, t_n\}$，且 $T \neq \phi$，表示事件或活动。

(4) $A_r(arc)$ 为有向弧集合，并且只能由库所指向变迁或变迁指向库所，$A \subseteq (P \times T) \cup (T \times P)$ 且 $P \cap T = P \cap A = A \cap T = \phi$。

(5) f (function)表示函数 $f = \{f_C, f_G, f_E\}$，其中 f_C 表示颜色函数；f_G 表示警卫函数；f_E 表示弧表达函数。

(6) M_0 表示初始标识，$M_0: P \rightarrow \{0\} \cup Z^+$ 为初始标识，即为初始时刻所有 place 的 token 的情况。令牌(token)表示对象。

Petri 网基本性质包括有界性、公平性和可达性等。可达性指从某一初始状态出发，能否通过一系列变迁的触发，到达特定状态。有界性则是指在 Petri 网的任意状态中，某个库所包含的标记数量是有限的。具体而言，如果一个 Petri 网是 k 有界的，则在任何状态下，任意库所的标记数不会超过 k。

定义 3　前驱与后继序列集(Pre-set and Post-set)　以节点 n 为标记点，当前节点 n 的前一节点的集合则称为前驱序列集，用 ${}^*n$ 表示，${}^*n = \{y \in P \cup T \mid (y,x) \in N\}$。当前节点 n 的后一节点集合称为后继序列集，用 n^* 表示，$n^* = \{y \in P \cup T \mid (x,y) \in N\}$。

定义 4　子网　设 $PN = (P, T, A_r, M_0)$ 为一个 Petri 网，称 $PN_i = (P_i, T_i, A_{ri}, M_{0i})(i = 1,2,\cdots,k)$ 为 PN 基于 i 模块的子网 PN_i 满足：

(1) $T_i \subset T(i = 1,2,\cdots,k)$；

(2) $P_i \subset P(i = 1,2,\cdots,k)$；

(3) $A_{ri} = \{(P_i \times T_i) \cap (T_i \times P_i)\} \cap A_r, (i = 1,2,\cdots,k)$；

(4) $M_{0i} = \Gamma_{P \rightarrow P_i} M_0$

2　演进行为与基本过程

演进行为是确保演进目标得以实现的基础，其中演进类型判定是业务流程演进的关键环节。流程演进可能涉及单个流程内部节点的增减或修改，或涉及不同模块间的流程结构调整，乃至流程重构或不同主体间的协作演进。基于演进行为判定演进类型，制定相应的演进规则，能够有效提升演进效率并降低成本。

根据演进行为特征，将其划分为原子层演进、结构层演进以及全局演进三个层次。

(1)原子层演进是最基本的演进单元，具有不可再分的特性，构成了演进行为的基础。

(2)结构层演进是在原子层演进基础上，包含了并列、选择等涉及复杂网关的演进操作，进一步拓展了演进的维度。

(3)全局演进从整体视角出发，旨在实现流程的整体优化和升级。

演进过程的关键在于将 BPMN 模型动态演进映射到着色 Petri 网(CPN)模型中，建立 CPN 可执行演进模型。BPMN 与 CPN 模型演进过程框架如图 2 所示。

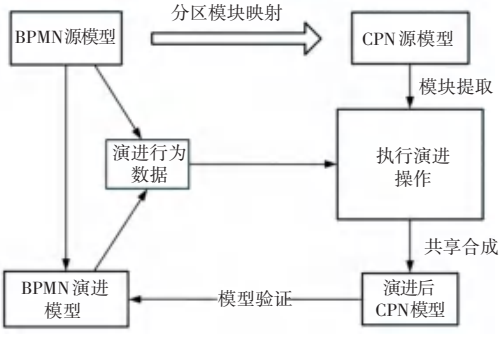


图 2　BPMN 与 CPN 模型演进过程框架

Fig. 2　Basic framework of BPMN and CPN model evolutionary process

该框架主要涵盖 4 个模型：BPMN 源模型、BPMN 演进模型、CPN 源模型和 CPN 演进模型。根据基本框架将模型演进过程分为多个步骤。对于 BPMN 模型，具体描述如下。

(1) 建立场景源模型：面向应用场景，建立 BPMN 协作模型，并分区域模块化处理。

(2)形式化映射：根据映射规则，将 BPMN 模块

映射为相应的 CPN 模型。

(3)场景更新及演进:对于演进后的 BPMN 模型,通过算法检测演进前后 BPMN 模型结构和元素变化,识别演进模块。

对于 CPN 模型,具体描述如下。

(1)可执行模型演进:对 CPN 演进模块,根据识别出的演进数据执行相关演进操作,建立 CPN 演进模型。

(2)验证与反馈:分析 CPN 演进模型状态空间,进行正确性验证。同时通过 BPMN 演进模型进行一致性验证,实现业务流程模型的优化和更新。

3 BPMN 模型演进

3.1 模块划分

BPMN 模型执行路径类似于有向图,其中每个 BPMN 元素都充当有向图的一个节点,能够可视化信息交互场景,建立面向场景的流程模型[21]。对于规模较大的复杂 BPMN 模型需要对其进行分块分析。首先,以池为边界进行区域分割,每个池划分为一个区域;对于每个区域内部,以网关对来划分模块,根据节点类型和网关对分析来确定 BPMN 模块边界[22]。BPMN 模型一个区域内模块划分的伪代码描述详见如下。

算法 1 BPNM 模型区域模块划分

输入 BPMN 源模型

输出 BPMN 区域划分子模块 $BPSM$、划分区域数 Rs、划分模块数 p

Initialize

$BPSM = [\,]$、$Rs = 0$、$p = 0$, $BPOM = \{N, F, P_L\}$

$N = N_1, N_2, \cdots, N_k$

if BPOM is without pool then $Rs = 1$

end if

for each pool in $BPOM$ do

$pool\_modules\_list = [\,]$

for each N in a pool do

if N is a gateway:

if N is first start gateway N_{G1}:

then find all elements from N_1 to $G1$ to create module M_1

$p = p + 1$

add M_1 to $pool\_modules\_list$

else if N is last end gatewayN_{Gt}:

then find all elements from N_{Gt} to N_k to create module Me

$p = p + 1$

add Me to $pool\_modules\_list$

else: // if N_{Gi} is neither a first nor an last gateway

then search for its corresponding gateway node N_{Gi-1} to N_{Gi} create a new module M_i

$p = p + 1$

add M_i to $pool\_modules\_list$

add $pool\_modules\_list$ as a submodule of pool to $BPSM$

$Rs = Rs + 1$

Return $BPSM$, Rs, p

定义 5 模块划分(region division) 设 BPMN 流程 $BPOM = \{N, F, P_L\}$,令每个池作为一个区域 R_s,每个区域内子流程为 $bp = \{M_1, M_2, \cdots, M_i, \cdots, M_p\}$, $i = 2, 3, \cdots, p-1$, $\exists g_{i1}, g_{i2} \in M_i$, $M_i = (g_{i1}, n_1, n_1, \cdots, g_{i2})$,其中 $M_i (i \neq 1, p)$ 表示第 i 个模块(M_1, M_p 表示输入模块和输出模块,不包含网关节点)。图 3 说明了 BPMN 模块划分过程。

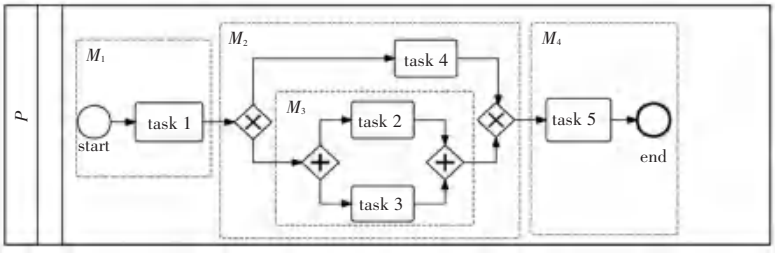


图 3 BPMN 模块划分

Fig. 3 BPMN module division

3.2 BPMN 形式化映射

形式化映射即为 BPMN 模型到 CPN 模型的映射过程,转换后的模型需要具有与原始 BPMN 模型相同的业务逻辑[23]。图 4 展示了从 BPMN 到 CPN

的基本元素的部分映射规则以及元素的基本演进操作，值得注意的是，为保持结构完整，在 CPN 模型中存在一类静默变迁（Silent Transition）[22]，这类特殊变迁不直接影响 token，一般用来补充结构。

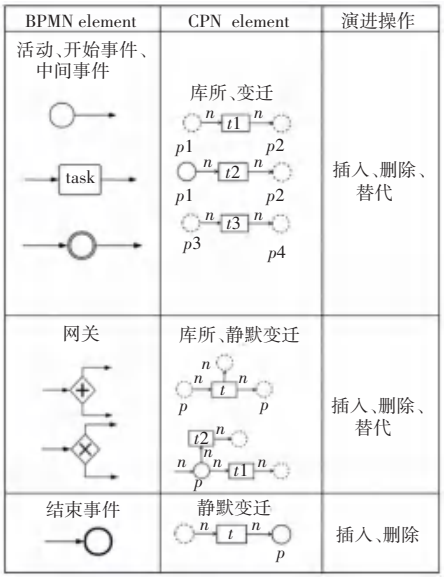


图 4　基本元素映射及相关演进操作

Fig. 4　Basic element mapping and related evolutionary operations

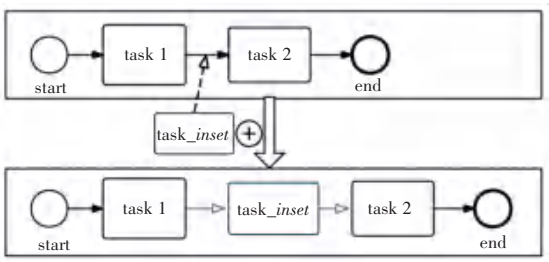
3.3　BPMN 演进过程与检测

在 BPMN 协作模型的演进过程中，通过对各元素及其相互连接关系的精确调整，可以实现既定的演进目标。插入、删除、替代演进示意如图 5 所示。3 种基本的演进操作包括：插入（Insertion）、删除（Deletion）以及替换（Replacement）。这里将展开研究分述如下。

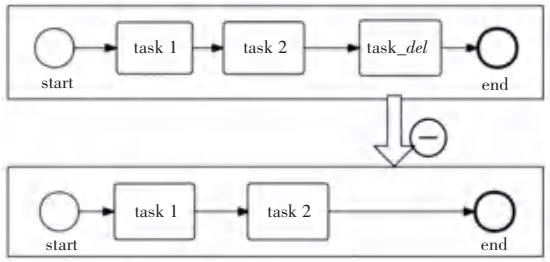
（1）插入演进。指在流程中 2 个相邻节点间植入新的功能片段，并同步更新相应的连接关系，从而形成新的流程，见图 5（a）。

（2）删除演进。是对流程内旧有的功能片段及其连接弧进行移除。见图 5（b）。

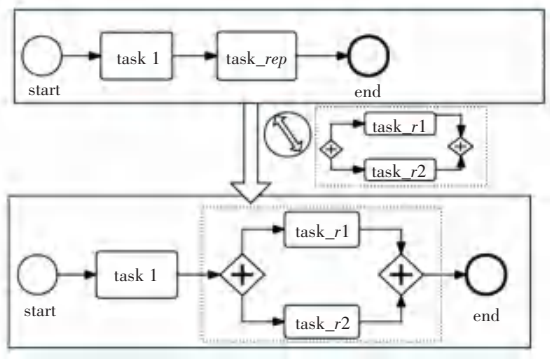
（3）替代演进。是指将既有的功能片段替换为新的活动片段，这一过程可视作删除与插入操作的复合，见图 5（c）。



（a）插入操作



（b）删除操作



（c）替代操作

图 5　插入、删除、替代演进示意图

Fig. 5　Schematic diagram of insertion, deletion and substitution evolution

基于演进过程框架，对演进后的 BPMN 模型与原始模型进行对比分析[18]。算法 2 解释了模型演进检测过程，输入场景源模型 BPO 与演进模型 BPN，最后输出演进操作类型数据。

算法 2　BPMN 演进检测

输入　*BPO*；*BPN*

输出　evolution types data：*ET*

Start：

```
BPO = {B, F, C}, BPN = {B1, F1, C1}, function
Evolution (BPO, BPN, ET)
function Evolution (BPO, BPN, ET):
evolution_type = [ ]
evolution_module_ position = [ ]
    BPO_elements = BPO. elements( )
    BPN_elements = BPN. elements( )
if BPN_elements. is subset( BPO_elements):
      for element in BPN_elements:
         if element not in BPO_elements:
            evolution_type = "Insert"
            evolution_module_ position = element
         Break
elif BPO_elements.  is subset( BPN_elements):
```

```
for element in BPO_elements:
    if element not in BPN_elements:
        evolution_type = "Delete"
        evolution_module_ position = element
    Break
else:
    for element in BPO_elements:
        if element in BPN_elements:
            Continue
        else:
            evolution_type = "Replace"
            evolution_module_ position = (element,
next(iter(BPN_elements - BPO_elements)))
            Break
Return ET
```

4 CPN 模型演进

4.1 CPN 演进模块提取

设 CPN 模型为 $PN=(P,T,\Sigma)$，χ 为提取算子，PN 中存在连续的变更节点集 $NC=(P',T')$，可以通过 χ 算子提取包含 NC 的演进模块 M，得到演进模块集 $PN'=(M_1,M_2,\cdots,M_k)=\chi PN$。CPN 模块如图 6 所示，具体步骤如下：

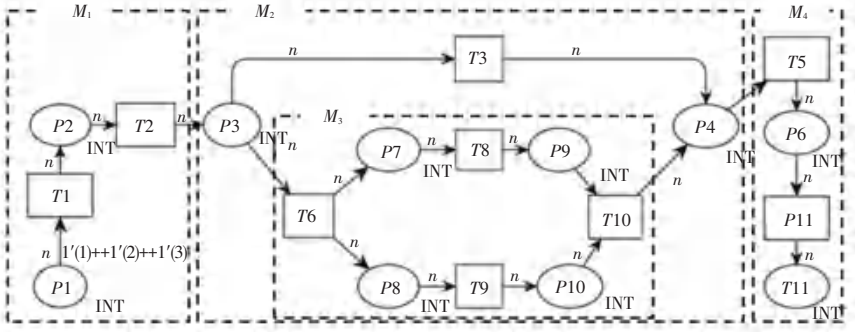


图 6 CPN 模块

Fig. 6 CPN model

（1）确定演进操作中变更活动节点 nc_i，及其标签。

（2）从节点 nc_i 出发，遍历其前驱序列集 ${}^*nc_i$ 和后继序列集 $nc_i{}^*$，将与变更活动节点直接相连的所有变更节点纳入演进模块 M_1 中，$M_1=(nc_1,nc_2,\cdots,nc_t)$。

（3）重复步骤（1）、（2），直到没有新的节点可以被纳入，得到提取的演进模块集 $PN'=(M_1,M_2,\cdots,M_k)$。

4.2 CPN 演进模块演进操作

针对特定业务对象的一系列状态变化或活动执行的子流程称为流程片段（Process Fragments，PF），该片段是一个具有独立功能的可重用构建模块，包含明确的输入、输出、活动和流关系[24]。

定义 6 流程片段（*PF*） 设片段 $PF=(I,O,A,S)$，其中 I 表示输入集合，O 表示输出集合，A 表示活动节点集，S 表示序列流。

根据不同结构特点，可以将模型简单分为顺序、并行、选择等。以插入演进为例，其 Petri 网结构演进示意如图 7 所示。

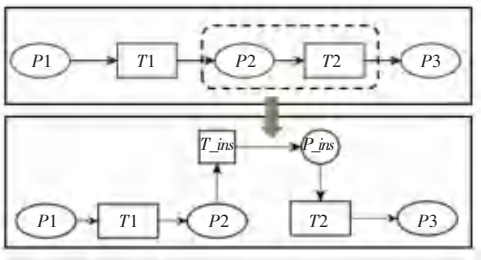
设 $PN=(P,T,A_r,M_0)$，$N1、N2\in PN$，则 Insert Evolution $=(PN,(N1,N2),x,ST)$，其中 x 为待插入片段，ST 为结构类型，$ST=\{Sequence,Choice,Parallel\}$，根据插入节点类型和顺序流可进行以下操作：

（1）库所-变迁插入。若 $N1\in P,N2\in T$，插入活动片段 x 为成对库所变迁集 (P',T')，此时插入演进为顺序插入，表示为：$Seq=(N1\rightarrow(P'\rightarrow T')\rightarrow N2)$，$ins\_seq=(PN,N1_p,x,N2_t,seq)$。

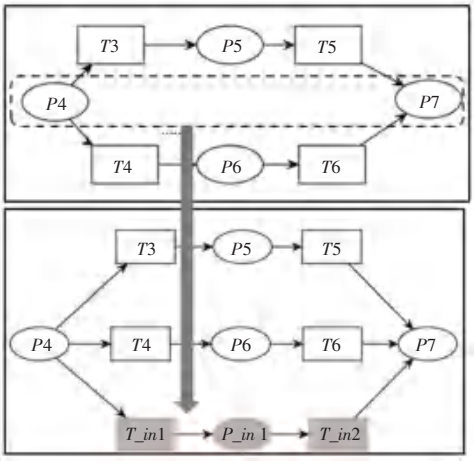
（2）库所-库所插入。若 $N1\neq N2$ 且 $N1,N2\in P,x$ 为变迁库所变迁集 (T_1',P',T_2')，此时插入演进为选择插入，表示为：$Cho=(N1_p\rightarrow(T_1'\rightarrow P'\rightarrow T_2')\rightarrow N2_p)$，$ins\_cho=(PN,N1_p,x,N2_p,Cho)$。

（3）变迁-变迁插入。若 $N1\neq N2$ 且 $N1,N2\in T,x$ 为库所变迁库所集 (P_1',T',P_2')，此时插入演进为并行插入，表示为：$Par=(N1_t\rightarrow(P_1'\rightarrow T'\rightarrow P_2')\rightarrow N2_t)$，$ins\_par=(PN,N1_t,x,N2_t,par)$。

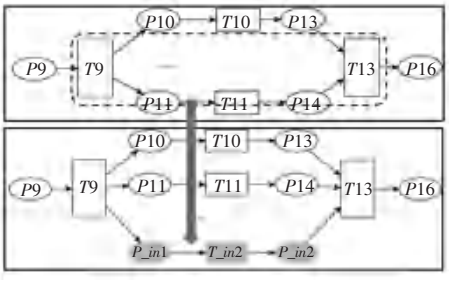
值得注意的是，在讨论演进操作时，通常所选择的流程片段的流向与原流程的顺序流向是一致的。当所选流向与原顺序流向不一致时，称之为逆向演进操作，这种情况容易导致循环嵌套的出现。



(a) 顺序



(b) 选择



(c) 并行

图 7　CPN 结构演进

Fig. 7　CPN structure evolution

4.3　CPN 演进模块合成

定义 7　共享库所　共享库所是一组库所的集合，且具有唯一的标识符，以便在 Petri 网合成过程中能够明确地识别和引用。

设 2 个 Petri 网 $N=(P,T,F)$ 和 $M=(P',T',F')$，其中 P 与 P' 分别为 N,M 中的库所集，若存在库所 $p\in P$ 与 $p'\in P'$，使得在网的合成中，p' 与 p 表示同一实体，即在合成网 NM 中共享相同的标识，且满足：

(1) $\forall t\in T,t'\in T'$，若 $(t,p)\in F$ 或 $(p,t)\in F$，则 $(t,p')\in F'$ 或 $(p',t')\in F$，反之亦然。

(2) 合成网 NM 中，p' 与 p 初始令牌相同，则库所 p' 与 p 称为共享库所。

模块演进完成后的关键步骤是将所有模块有机地连接起来，构建成完整的 CPN 模型。通过共享库所连接模块或者子网，实现网的合成[25]。需要注意的是合成过程中需确保合成后的 Petri 网仍然是可达的，即从起始库所到终止库所仍然存在一条路径。

合成步骤为：设 ψ 为合成算子 $CN_i=(\Sigma_i,P_i,T_i,A_{ri},M_{0i})(i=1,2)$ 为 2 个模块，共享库所 Pd_i，令 $CN=\psi(CN_1,CN_2)=(P,T,A_r,M_0)$，其中，$P=P_1\cup P_2$，$P_1\cap P_2\neq\phi$，$T=T_1\cup T_2(T_1\cap T_2=\phi)$，$A_r=A_{r1}\cup A_{r2}$，可得：

$$M_0(p)=\begin{cases}\max\{M_{01}(p),M_{02}(p)\}, & p\in P_1\cap P_2\\ M_{0i}(p), & p\in P_i-(P_1\cap P_2),i=1,2\end{cases}$$

5　智慧课程推荐系统案例

考虑一个在线网络选课系统，该流程分为 3 个阶段：前期培养计划制定、中期培养计划审核、后期课程选择。在计划制定阶段，学生登录系统后，浏览选课规则，选择课程并生成培养计划。在计划审核阶段，学生将培养计划提交系统审核，获得审核结果。在选课处理阶段，学生根据计划在系统中添加本期课程，同时剔除未开放课程，最终形成个人课程表。BPMN 源模型如图 8 所示，对其进行模块化处理，分为 $\{M_1,M_2,M_3,M_4\}$ 与 $\{N_1\}$ 两个区域，共 5 个模块。经过映射后的 CPN 模型如图 9 所示。

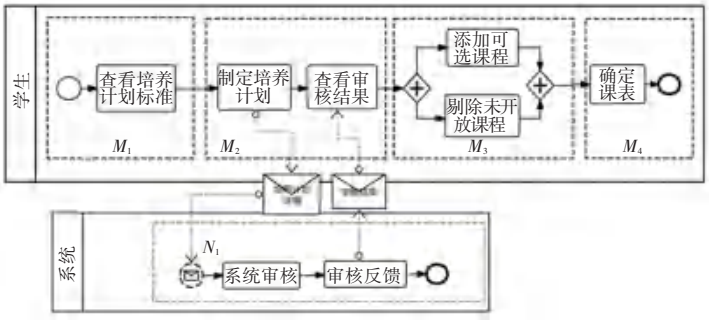


图 8　BPMN 原始模型

Fig. 8　BPMN original model

由于技术升级和新规要求,系统进行了更新,更新演进后的流程模型如图 10 所示。首先,系统添加了人工智能助手,在第一阶段增加了智能化个性化定制课程功能。其次,在审核阶段,增加计划重选,方便学生重新修改计划。选课阶段增设“智能一键选课”功能,以满足学生高峰选课需求。此外,在审核阶段,由原来的系统审核变更为“导师、学院及教务三方审核”。

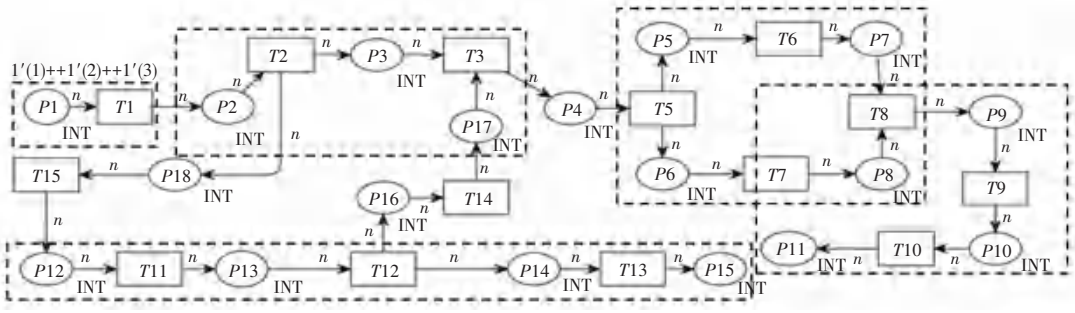


图 9 CPN 源模型

Fig. 9 CPN original model

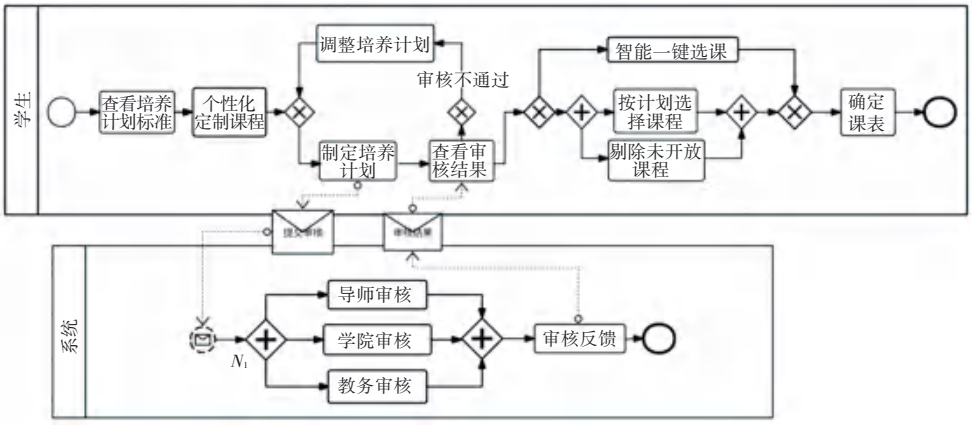


图 10 演进后 BPMN 模型

Fig. 10 Evolved BPMN model

根据算法 2 输出结果,演进模块为 M_1,M_2,M_3,N_1,演进操作分别为顺序插入、选择插入(逆向),选择插入、替代。提取演进模块执行相应演进操作,使用 cpn tools 工具对 CPN 模型进行仿真执行,生成演进后 CPN 模型如图 11 所示,其状态空间报告如图 12 所示。根据该报告可验证 Petri 网模型的基本属性,如可达性、有界性、活性及公平性等,易知模型不存在死锁、活锁,当活动执行终止时,可以得到正确的结果。同时与演进前相比,模型输出库所仍为 home 状态,总是有可能终止,模型具有可达性。

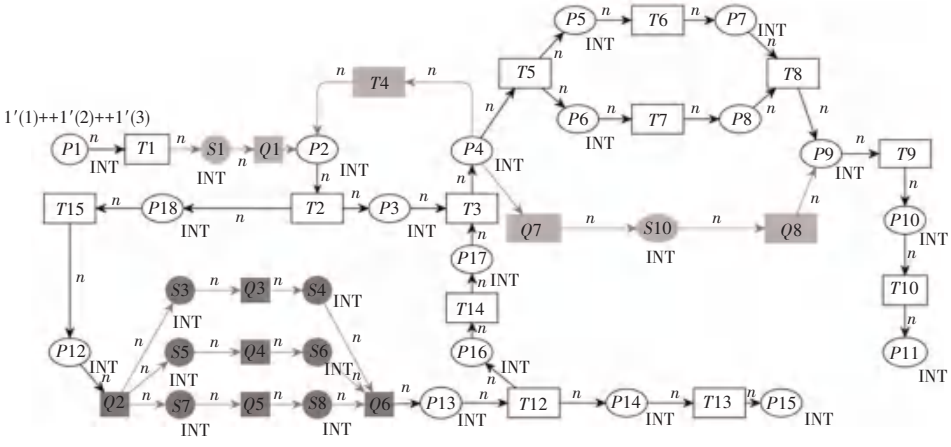


图 11 演进后 CPN 模型

Fig. 11 Evolved CPN model

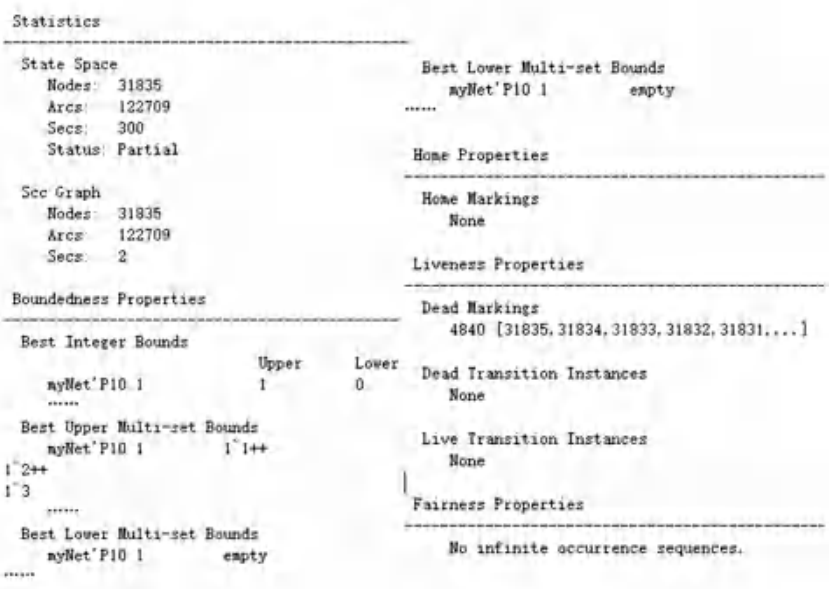


图 12　CPN 演进模型状态空间报告

Fig. 12　State space report of CPN evolutionary model

6　结束语

以人工智能为代表的新技术浪潮正在重塑人们的工作与生活模式，不仅为现代信息系统带来了技术革新，还为架构模型的发展提供了更高维度的空间和更广泛的应用场景，在需求快速变化与技术持续发展的背景下，业务流程模型需要不断演进与优化，以支撑可持续性发展目标。本研究提出的基于场景更新的业务流程迭代演进方法，构建了 BPMN 流程演进与 CPN 模型动态更新的框架，通过模块化分解复杂 BPMN 模型与形式化映射，将静态模型映射为可执行模型，并通过演进检测算法结果进行 CPN 演进行为操作，以构建 CPN 演进模型。最后引入智慧选课系统案例，通过 cpntools 工具，验证了研究方法的可行性。有效降低了模型演进的复杂性和成本，避免了映射重复，提高了系统应对多样化需求场景的适应性和效率。

然而，该方法在演进行为分析方面仍有限制和挑战。目前仅仅对简单演化操作进行探讨，未涉及交、并、补复杂演进行为的研究；在复杂 BPMN 模型向 CPN 模型的映射及演进操作中，主要关注了控制流而忽略了数据流的分析。未来研究将致力于深入探究数据流层面的演进操作。在实际应用中，本方法有望助力企业快速适应市场需求波动与服务场景的演变，提高系统的适应性与有效性。

参考文献

[1] MARLON D，FABIANA F，LIOR L et al. Ai-augmented business process management systems：A research manifesto[J]. ACM Transactions on Management Information Systems，2023，14(1)：11.

[2] DANNIEL S S，NIKOLA T，DARKO E. Big data BPMN workflow resource optimization in the cloud[J]. Parallel Computing，2023，117：103025.

[3] YUSUF K，FLORIAN G，MICHAEL W，et al. BPMNE4IoT：A framework for modeling，executing and monitoring IoT-driven processes[J]. Future Internet，2023，15(3)：90.

[4] 回梦涵，黄凤兰，倪枫，等. 基于 NLP 技术对文本生成 BPMN 图[J]. 智能计算机与应用，2024，14(8)：11-18.

[5] 代飞，赵文卓，杨云，等. BPMN 2.0 编排的形式语义和分析[J]. 软件学报，2018，29(4)：1094-1114.

[6] ZHU Ming，LI Jing，FAN Guodong，et al. Modeling and verification of response time of QoS-aware Web service composition by timed CSP[J]. Procedia Computer Science，2018，141：48-55.

[7] OQUENDO F. π-Calculus for SoS：A novel π-calculus for the formal modeling of software-intensive systems-of-systems[C]// WoTUG Conference on Concurrent and Parallel Systems. Canterbury，UK：University of Kent，2015：541-566.

[8] ABDELKADER B，ABDELKRIM A，DJAMEL M. A new approach for workflow evolution using MDA technology[J]. International Journal of Intelligent Information and Database Systems，2019，12(4)：304.

[9] KAOUTHER M，LATIFA M. Public processes legal issues verification using YAWL[C]// Evaluation of Novel Approaches to Software Engineering. Cham：Springer，2021：289-296.

[10] 严志超，倪枫，刘文诚，等. 基于 CPN 的敏捷开发业务流程建模方法[J]. 智能计算机与应用，2024，14(8)：78-84.

[11] SONG Wei，JACOBSEN H. Static and dynamic process change[J]. IEEE Transactions on Services Computing，2018，11(1)：215-231.

[12] 郑伟波，李志超，刘纪遵，等. 基于关系依赖模型的微服务流程演进分析[J]. 南京大学学报(自然科学)，2022，58(2)：309-319.

[13] ORTIZ J, TORRESV, VALDERAS P. Microservice compositions based on the choreography of BPMN fragments: Facing evolution issues[J]. Computing, 2023, 105: 375-416.
[14] 尤殿龙,申利民,刘芳. Web 服务组合中业务流程演进影响范围判定方法[J]. 计算机集成制造系统,2013,19(7):1704-1714.
[15] SÉRGIO G, KHALED G, ULRIK F. Analysis of enterprise architecture evolution usingmarkov decision processes [C]// International Conference on Advanced Information Systems Engineering. Cham:Springer, 2016: 37-51.
[16] YADAV V, JOSHI R K, LING S. A tool for traceable evolution of process architectures[C]// 2018 IEEE International Conference on Software Architecture Companion (ICSA-C). Piscataway, NJ:IEEE, 2018:101-106.
[17] 胡强,任志考,赵振,等. 基于逻辑 Petri 网的服务流程结构演进研究[J]. 软件学报,2018,29(9):2697-2715.
[18] KRISHNA A, POIZAT P, SALAÜN G. Checking business process evolution[J]. Science of Computer Programming, 2019, 170:1-26.
[19] LI Zonghua, YE Zhengwei. A petri nets evolution method that supports BPMN model changes[J]. Scientific Programming, 2021, 2021: 6610795.
[20] 何海洋,李强,余祥,等. 基于高阶 π 演算的构件演化模型研究[J]. 计算机应用研究,2017,34(1):67-74.
[21] 王新康,倪枫,刘姜,等. 基于扩展 BPMN 的"家园互动"式儿童健康管理系统架构[J]. 智能计算机与应用,2022,12(10):189-199.
[22] 黄凤兰,倪枫,刘姜,等. 基于 HCPN 的复杂 BPMN 协作模型数据流建模与验证[J]. 计算机集成制造系统,2024,30(5):1754-1769.
[23] ANGELINO P R, MELLADO L. Discovering Petri nets including silent transitions. A repairing approach based on structural patterns[J]. Discrete Event Dynamic Systems,2022,32(2):291-315.
[24] 黄凤兰,倪枫,刘姜,等. 基于 ROAD-CPN 业务架构的可执行建模方法[J]. 上海理工大学学报,2023,45(5): 534-542.
[25] DAI Fei, MO Qi, LI Tong, et al. Refactoring business process models with process fragments substitution [J]. Wireless Networks,2020,30:3507-3521.
[26] LUAN Wenjing, QI Liang, ZHAO Zhongying,et al. Logic Petri net synthesis for cooperative systems[J]. IEEE Access, 2019, 7: 161937-161948.
[27] 钟贤欣,倪枫,刘姜,等. 基于 ROADS 的面向场景业务架构建模方法[J]. 上海理工大学学报,2023,45(4):415-424.