

基于 BPMN 的电力业务行为模型生成方法

李新家¹, 张崇超², 刘梦佳¹, 陈世晖², 喻伟¹, 李燕超²

(1. 国网江苏省电力有限公司, 江苏 南京 210024;

2. 国网思极网安科技(北京)有限公司, 北京 102200)

摘要:作为新型电力系统的重要组成部分,用电信息采集系统需要与电网多个业务系统交互并提供计量结算基础数据,导致业务生态较为复杂,难以实现对业务行为的高效建模及分析。为此,提出一种面向用电信息采集系统的业务行为模型自动化生成和形式化方法。首先,利用自然语言处理技术对业务行为相关描述进行分析并提取关键词,生成中间模型,然后,利用 BPMN 对中间模型进行解析,转化生成基于 BPMN 的业务行为模型。进一步,结合扩展 Petri 网模型,定义了 BPMN 模型形式化映射规则,实现对 BPMN 业务行为模型的形式化分析。实验结果表明,该方法能够在合理的时间、内存和 CPU 开销下,实现业务行为模型的自动化生成和形式化分析,为有效理解、分析和改进用电信息采集系统整体业务布局提供重要支撑。

关键词:新型电力系统;业务行为;自动化生成;形式化;BPMN 模型

中图分类号:TP711

文献标识码:A

文章编号:1673-629X(2025)03-0202-08

doi:10.20165/j.cnki.ISSN1673-629X.2024.0349

A Method for Generating Power Business Behavior Models Based on BPMN

LI Xin-jia¹, ZHANG Chong-chao², LIU Meng-jia¹, CHEN Shi-hui², YU Wei¹, LI Yan-chao²

(1. State Grid Jiangsu Electric Power Co., Ltd., Nanjing 210024, China;

2. State Grid Cyber Security Technology (Beijing) Co., Ltd., Beijing 102200, China)

Abstract:The electric data collection system is an important component of new power system. It needs to interact with multiple professional business systems within the smart grid and provide basic measurement and settlement data, resulting in a complex business ecosystem and difficulty in achieving efficient modeling and analysis of business behavior. To address the issue, we propose an effective method to automatically generate and formalize business behavior models of the electric data collection system. It first uses the NLP technology to analyze the related descriptions of business behavior, to extract relevant keywords which are used to form the intermediate model. Secondly, the BPMN model is adopted to parse the intermediate model and then generate a BPMN-based business behavior model. Furthermore, by combining the extended Petri net model, a formal mapping rule for the BPMN model was defined to achieve a formal analysis of the BPMN business behavior model. The experimental results show that the proposed method can achieve automated generation and formal analysis of business behavior models with reasonable time, memory, and CPU overhead, providing important support for effectively understanding, analyzing, and improving the overall business layout of the electric data collection system.

Key words:new power system; business behavior; automated generation; formalization; BPMN model

0 引言

用电信息采集系统^[1]是新型电力系统的重要组成部分,其是计量结算数据的唯一来源,为其它业务系统提供基础数据,需要与电网内多个业务系统交互,导致其业务生态较为复杂,从而很难实现对业务行为的高效建模及分析。因此,研究从业务相关描述,如:系

统日志、文本描述或流量等,自动提取业务关键词并建立业务行为模型的方法成为了必要,对有效理解、分析和改进整体业务布局具有重要意义。

业务行为模型是一种用于描述和分析业务行为的设计或体系结构^[2],旨在明确各个参与方之间的行为模型和特征。BPMN(Business Process Model and No-

收稿日期:2024-08-14

修回日期:2024-12-17

基金项目:国家电网有限公司科技项目(5700-202318300A-1-1-ZN)

作者简介:李新家(1967-),男,正高级工程师,研究方向为电力自动控制工程、电力大数据及安全、新型电力系统业务建模技术等;通信作者:张崇超(1982-),男,硕士,研究方向为计量采集、电力大数据及安全。

tation)^[3]作为一种可视化的业务建模语言,常用于各类业务模型的建模。程航宇等人^[4]对 BPMN 在业务建模中的相关应用进行了研究,提出了一种业务流程一体化建模的元模型。陈虎^[5]对 BPMN 到业务流程语言的映射进行了研究,以 XML 的格式对 BPMN 进行表达。但是,BPMN 并不能对语义、模型的正确性和一致性进行验证。为此,汪玉泉等人^[6]提出了基于对齐的 BPMN 2.0 模型符合性检测算法,能够计算带有复杂模式的 BPMN 2.0 模型与日志之间的契合度。为了实现业务模型的自动化生成,Sintoris 等人^[7]使用 NLP(Natural Language Processing) 技术设计 BPMN 生成算法,从业务流程的自然语言描述中提取生成 BPMN 模型。另外,李宗花等人^[8]提出一种针对 BPMN 模型的形式化方法,支持对 BPMN 模型正确性的验证。

目前关于电力系统业务行为模型自动化生成的研究成果较少,尤其缺乏对电力系统业务模型进行形式化映射及分析的方法。为此,该文提出一种面向用电信息采集系统的基于 BPMN 和扩展 Petri 网的电力业务行为模型生成方法(business behavior models based on BPMN and Extended Petri Nets, BEPN)。首先,利用自然语言处理技术对电力业务行为相关的文本描述、系统日志进行分析并提取关键词,生成中间模型;然后,利用 BPMN 模型对中间模型进行解析,转化生成基于 BPMN 的业务行为模型。进一步,结合扩展 Petri 网模型^[9],定义了 BPMN 模型形式化映射规则,实现对 BPMN 模型的形式化及分析。

1 基于 BPMN 的业务行为序列描述

1.1 BPMN 模型介绍

BPMN 提供了一套符号和规则^[10],用于表示业务行为序列中的事件、活动、网关等元素。利用 BPMN 对业务行为序列进行描述、分析和优化,解决了业务行为序列难以直观呈现和标准化的问题。近年来,随着业务序列复杂性的增加,BPMN 作为一种业务建模语言被广泛使用^[11-13]。

1.2 业务行为序列的 BPMN 描述

将用电信息采集系统业务行为模型描述为: $NPM-BPMN = (name, AC, GW, EV, SF, LA)$, 其中: name 是业务行为模型的名称; AC 表示该模型的 Activity 元素,也即业务行为模型中的具体业务活动。例如数据提供方准备数据、数据请求方发起数据请求等。 $AC = (Task, Subprocess)$, 其中 Task 表示行为模型中的任务, Subprocess 代表子 NPM-BPMN 模型; GW 表示行为模型中的 Gateway 元素集合,用于控制业务行为序列的分支和聚合。 $GW = (ParallelFork, ParallelJoin, Ex-$

clusiveMerge, ExclusiveDecision, Inclusive), 其中 Parallel 网关包括 fork 和 join 类型, Exclusive 网关包括 merge 和 decision 类型, Inclusive 是指包容性网关; EV 是行为模型中的事件,例如数据提供方准备好共享数据的事件、数据请求方提交数据请求的事件等。 $EV = (StartEvent, IntermediateEvent, EndEvent)$, 其中 StartEvent 表示开始事件, IntermediateEvent 表示中间事件, EndEvent 表示结束事件; SF 即 Sequence Flow, 为行为模型顺序流, $SF = \{sf_1, sf_2, \dots, sf_n\}$; LA 表示为行为模型的参与者, $LA = \{la_1, la_2, \dots, la_n\}$, 用于对业务参与者进行逻辑分组。例如可以将电力用户、数据提供方、数据请求方和数据共享平台分别划分到不同的甬道中,以显示它们的不同角色和职责。

下面通过一个例子说明利用 BPMN 模型对电力业务行为序列的描述,业务名称为“电力公司 A 共享用电负荷数据给电力公司 B”, 对该业务行为序列的描述为: 首先, 电力公司 A 收集用电负荷数据。然后, 电力公司 B 提交对电力公司 A 用电负荷数据的数据请求。电力公司 A 收到了数据请求, 并进行权限的验证。如果验证通过, 电力公司 A 允许数据共享, 否则, 拒绝共享。电力公司 A 将用电负荷数据集传输给电力公司 B。

该过程涉及到数据提供方(电力公司 A)和数据请求方(电力公司 B)的协作。图 1 给出基于该业务行为序列描述的 BPMN 图。

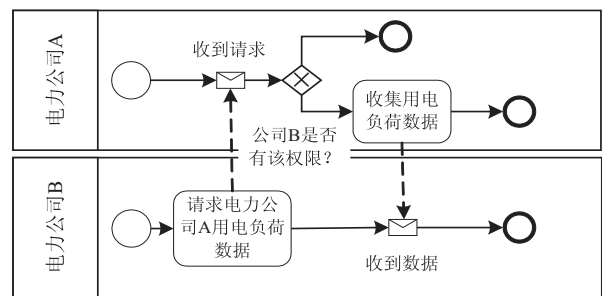


图 1 电力公司共享用电负荷数据 BPMN 图

2 基于自然语言处理的业务行为模型自动化生成方法

为了解决由业务流程相关描述高效生成业务模型的问题,近年提出了一些支持自动化生成业务模型的方法^[14-17],即:利用 NLP 技术,使用指代消解^[18]和主谓宾提取^[19]等方法提取业务流程主体,进一步将其映射为业务模型中的元素,从而生成业务模型。针对用电信息采集系统业务行为相关描述(业务系统日志、业务文本描述)的特点,本节提出一种基于自然语言处理的业务行为模型自动化生成方法,其原理如图 2 所示。

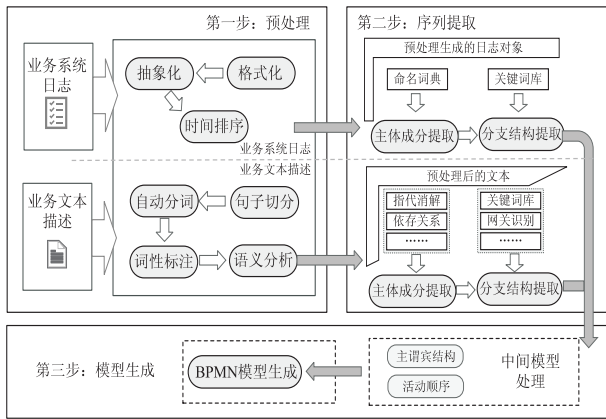


图 2 BPMN 模型生成方法

2.1 预处理

针对电力系统日志的预处理包括三个步骤:格式化、抽象化和时间排序,如图 3 所示。对于电力系统日志 PSLog,首先根据格式化标准将系统日志进行格式化,得到格式化列表 $FL = \{fl_1, fl_2, \dots, fl_n\}$ 。接着将其转换为对象,进行抽象化表达,得到 $OL = \{Ol_1, Ol_2, \dots, Ol_n\}$,其中 $Ol_i = \text{abstract}(fl_i, OA)$,OA 表示对象结构。最后根据调用时间进行排序,得到 $SOL = \{Sol_1, Sol_2, \dots, Sol_n\}$ 。

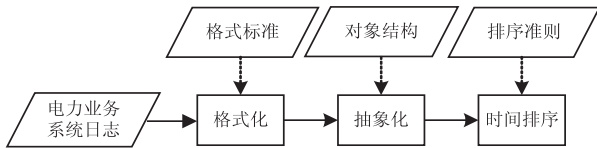


图 3 针对业务系统日志的预处理

针对电力业务文本描述的预处理由 NLP 技术完成,包括四个步骤:句子切分、自动分词、词性标注和语义分析,如图 4 所示。对于电力业务文本描述 PBNLD,首先通过句子切分,将其按照句号、问号、感叹号等标点符号进行分割,得到多个独立的句子 $IS = \{IS_1, IS_2, \dots, IS_n\}$ 。随后对每个句子进行分词、词性标注和语义分析的操作,得到 $OS = \{OS_1, OS_2, \dots, OS_n\}$,其中 $OS_i = \text{handle}(IS_i, h_1, h_2, h_3)$,表示对独立句子 IS_i 进行自动分词、词性标注和语义分析三种处理后得到的对象。

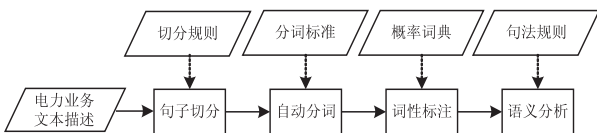


图 4 针对业务文本描述的预处理

2.2 针对业务行为的序列提取

针对业务行为的序列提取基于预处理结果 SOL 或 OS。首先对主体成分进行提取,随后提取分支结构,如图 5 所示。

主体成分提取需要依据 SOL 或 OS 来提取主语 Subject 和主谓宾结构 SVO。当输入为 OS 时,将具有

主语依存关系且同时具有名词词性标注的词汇列为主语。对于具有主语依存关系且同时具有代词词性标注的词汇,使用指代消解技术寻找其指代词汇,将指代词汇列为主语。之后,通过语义分析寻找主语对应主谓宾结构,并将其列入业务行为描述主谓宾列表 BD-List。当输入为 SOL 时,选择当前日志对象所记录的服务调用者或发起者,根据命名词典寻找其实际词汇,列为主语。接着,选择当前日志对象所记录服务,列为活动内容,并将结果列入 BD-List。最终,BD-List 表示为 $\{SVO_1, SVO_2, \dots, SVO_n\}$ 。

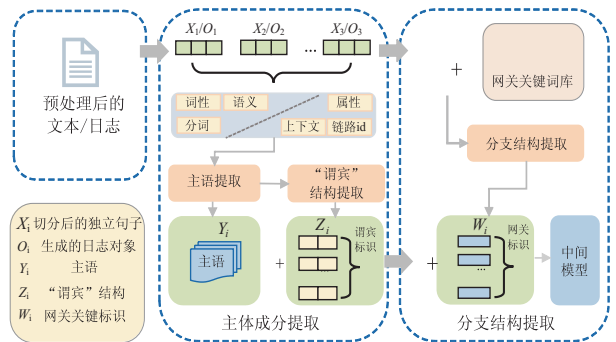


图 5 提取方法

分支结构提取则需要依据预设的网关关键词库来识别条件、并行和默认网关。若检索到条件、并行或默认网关关键字,则添加对应网关标记至电力系统业务行为描述主谓宾列表;若未发现预设的网关关键字,则视为简单活动,不做处理。完成该项提取后,形成带有网关标记的中间模型 $IM = \{SVO_1(\text{mark}_x), SVO_2(\text{mark}_y), \dots, SVO_n(\text{mark}_z)\}$,其中 $\text{mark}_x, \text{mark}_y, \text{mark}_z \in \{\text{Condition}, \text{Parallel}, \text{Default}\}$ 。

2.3 业务行为模型生成

为了生成业务行为模型,需要解析中间模型,随后利用 BPMN 模型,生成基于 BPMN 的业务行为模型。其具体步骤为:

- (1)对标记为默认网关的主谓宾结构进行处理。若上一个主谓宾结构被标记为条件网关,则该主谓宾结构添加为条件网关的另一个选项,标记为条件网关;若上一个主谓宾结构被标记为并行网关,则该主谓宾结构添加为并行网关的并行选项,标记为并行网关;
- (2)遍历主谓宾列表,对只有条件流的主谓宾结构进行处理,在其后添加默认流;
- (3)将业务行为描述主谓宾列表的“谓宾”结构、完整主语、标识分别添加至中间模型的 Activity、Who、Condition 属性,形成二维列表;
- (4)遍历列表,按顺序添加序号 Order,将连着的相同网关结构视为同顺序,按照形如 1a,1b,1c...的顺序添加;
- (5)使用 bpmn-python 库提供的功能,将生成的

中间模型转换为 BPMN 模型,并以 XML 的格式导出。

下面通过一个实际案例,介绍业务模型自动化生成的具体应用。其业务描述如下:

电力供应商将电力需求和分配计划传输到变电站。变电站接收来自电力供应商的数据。如果电力供应商身份正确,变电站就根据电力供应商提供的计划进行电力变压和分配;否则,变电站就丢弃该消息。变电站将实时的电力分配和负荷情况数据报告给电力供

应商。电力供应商收到数据。

首先对该文本进行预处理,通过句子切分将其划分为独立的句子,对每个句子进行分词、词性和语义分析的操作。接着通过提取模块提取主体成分和分支结构,以生成中间模型。生成的中间模型如表 1 所示。

基于该中间模型利用 BPMN Python 库即可生成 BPMN 模型,如图 6 所示。

表 1 基于示例文本的中间模型

Order	Activity	Condition	Who	Terminated
1	将电力需求和分配计划传输到变电站		电力供应商	
2	接收来自电力供应商的数据		变电站	
3a	根据电力供应商提供的计划进行电力变压和分配	如果供应商身份正确	变电站	
3b	丢弃该信息	否则	变电站	yes
4	将实时的电力分配和负荷情况数据报告给电力供应商		变电站	yes
5	收到数据		电力供应商	

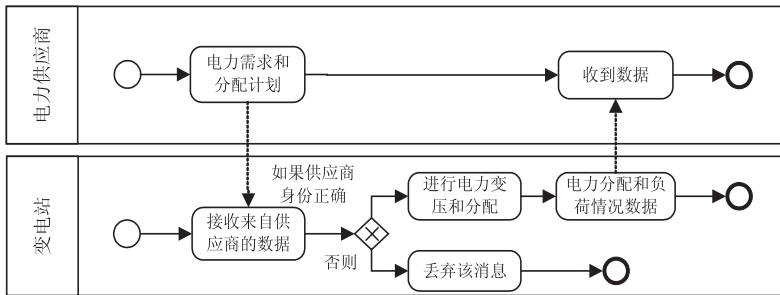


图 6 基于示例文本的 BPMN 模型

3 基于扩展 Petri 网的业务行为模型形式化方法

将扩展 Petri 网用于业务行为模型的形式化,通过映射算法将业务行为模型转换为扩展 Petri 网模型,能够对业务行为模型的正确性进行分析和验证^[20-22]。本节提出一种基于扩展 Petri 网的业务行为模型形式化方法,即:利用扩展 Petri 网对生成的 BPMN 模型进行形式化,以验证业务行为模型的正确性。

3.1 形式化映射规则

扩展 Petri 网元模型可以描述为一个六元组: $EPetri = (PL, TR, IA, OA, CO, SP)$,其中 PL 表示条件集合;TR 表示事件集合; $IA \subseteq PL \times TR$,表示一组输入弧; $OA \subseteq TR \times PL$,表示一组输出弧;CO 表示容器集合; $SP = (PL, TR, IA, OA, CO, SP)$,表示子 EPetri 模型。

对 BPMN 模型进行形式化就是通过一系列映射规则将 BPMN 图转换为扩展 Petri 网。由于同种类型元素映射规则的相似性,这里同种类型规则只举一个例子进行表示。

考虑到一个 BPMN 模型通常只有一个开始事件,

该事件在映射为扩展 Petri 网模型时需要转换为一个带有初始条件和后置条件的静态事件,并且该初始条件需要带有标识符。BPMN 模型中开始事件的映射规则可以表示为:

$$\text{StartEvent} \rightarrow \text{SilentTransition} \quad (1)$$

$$(\text{PrePlace}(\text{source}_{\text{token}}), \text{PostPlace})$$

式中, StartEvent 表示开始事件, SilentTransition (PrePlace(source_{token}), PostPlace) 表示封装带有初始标识符的前置条件和后置条件的静态事件。

相比之下, BPMN 模型的结束事件与中间事件在触发机制上有所不同。结束事件通常较为稳定,不依赖于其他事件的触发。因此,在映射为扩展 Petri 网模型时,结束事件应被转换为带有前置条件和后置条件的静态事件。中间事件在 BPMN 模型中常依赖于其他事件的触发,具有更多的变化性。因此,在映射为扩展 Petri 网模型时,中间事件应被转换为带有前置条件和后置条件的动态事件。这两种事件映射规则可类似式 1 形式表示。

BPMN 模型中,任务是业务行为序列中的具体操作或活动,常依赖于其他事件和活动,具有更多的变化

性。因此,在将 BPMN 模型映射为扩展 Petri 网模型时,任务被表示为带有前置条件和后置条件的动态事件。任务的映射规则为:

$$\text{Task} \rightarrow \text{BehaviorTransition} \quad (2)$$

$$(\text{PrePlace}, \text{PostPlace})$$

式中,任务 Task 被转换为封装有前置条件和后置条件的动态事件。

BPMN 模型的并行网关分为 fork 和 join 类型。其中 fork 类型只有一个前置条件,它在满足前置条件后将同时进入多个分支,每个分支对应一个后置条件。而 join 类型具有多个前置条件,它在满足所有前置条件时将合并为一个流程,该流程对应一个后置条件。其中,fork 类型网关的映射规则定义为:

$$\text{ParallelFork} \rightarrow \text{SilentTransition} \quad (3)$$

$$(\text{PrePlace}, \text{map}(\text{PostPlace}))$$

式中,ParallelFork 表示平行 fork 类型的网关,SilentTransition(PrePlace, map(PostPlace)) 表示为封装有一个前置条件和后置条件集合的静态事件,表示 fork 类型网关被映射为扩展 Petri 网模型的带有一个前置条件和多个后置条件的静态事件。

BPMN 模型的排他网关分为 merge 和 decision 类型。其中,merge 类型具有多个前置条件,它在满足任意一个前置条件时将合并为一个流程,该流程对应一个后置条件。而 decision 类型只有一个前置条件,它在满足前置条件后会进入多个分支中唯一的一个分支。BPMN 模型的包容网关在满足任意一个前置条件后都将进入后续的多个分支中一个或多个分支执行。这三种类型网关均可如式 3 形式表示。

3.2 形式化映射算法

考虑到 BPMN 模型不同类型元素之间的差异性,将同一类型元素映射为一个扩展 Petri 网模型,映射得到的不同结果再进行合并,因此,针对不同类型元素,本节分别给出其代表性元素的映射算法。根据规则(1),将开始事件映射为带有一个初始条件和一个后置条件的静态事件,具体实现如算法 1 所示。

算法 1:开始事件映射

输入:BPMN 开始事件集合

输出:扩展 Petri 网模型 EPetri

1: let se_map //获取开始事件集合

2: EPetri = ()

3: foreach se ∈ se_map do //遍历事件

4: create Place p1, p2 //创建条件

5: set_source_token(p1) //p1 为前置条件并设置初始标记

6: EPetri.PL ← place_merge(EPetri.PL, p1, p2) // p1, p2 并入条件集合

7: create Transition t ← Transition(silent, se) //创建静

态事件 t, silent 为静态事件标记

8: EPetri.TR ← transition_merge(EPetri.TR, t) //事件 t 并入事件集合 TR

9: EPetri.IA ← arc_merge(EPetri.IA, (p1, t)) //(p1, t) 表示弧,并入输入弧集合 IA

10: EPetri.OA ← arc_merge(EPetri.OA, (t, p2)) //(t, p2) 并入输出弧集合 OA

11: end for

12: return EPetri

BPMN 模型中的任务与中间事件类似,都依赖于其他事件和活动。根据规则(2),任务映射算法的具体实现如算法 2 所示。

算法 2:任务映射

输入:BPMN 任务集合

输出:扩展 Petri 网模型 EPetri

1: let task_map //获取任务集合

2: EPetri = ()

3: foreach task ∈ task_map do

4: create Place p1, p2

5: EPetri.PL ← place_merge(EPetri.PL, p1, p2)

6: create Transition t ← Transition(behavior, task) //创建动态事件 t, behavior 为动态事件标记

7: EPetri.TR ← transition_merge(EPetri.TR, t) //事件 t 并入事件集合 TR

8: EPetri.IA ← arc_merge(EPetri.IA, (p1, t)) //并入输入弧集合 IA

9: EPetri.OA ← arc_merge(EPetri.OA, (t, p2)) //并入输出弧集合 OA

10: end for

11: return EPetri

网关作为 BPMN 模型中起分支或决策作用的节点,其结构比事件和活动更复杂,其形式化往往会映射出更复杂的结构。考虑到每种网关各有特点,选择 ParallelFork 类型进行说明。在 ParallelFork 类型网关形式化的过程中,将其映射为带有一个前置条件和多个后置条件的静态事件。根据规则(3),ParallelFork 网关映射算法的具体实现如算法 3 所示。

算法 3:ParallelFork 网关映射

输入:BPMN ParallelFork 网关集合

输出:扩展 Petri 网模型 EPetri

1: let pf_map, sf_map //获取 ParallelFork 网关集合和顺序流集合

2: EPetri = ()

3: foreach pf ∈ pf_map do //遍历网关

4: create transition t ← Transition(silent, pf)

5: EPetri.TR ← transition_merge(EPetri.TR, t)

6: foreach sf ∈ sf_map do //遍历顺序流

7: if pf is target of sf then //若顺序流 sf 指向网关 pf

8: let p ← seq_make_target(sf) //根据函数 seq_make_target 和 sf 创建条件 p, 标记为 target

```

9:   EPetri.PL ← place_merge(EPetri.PL, p)
10:  EPetri.IA ← arc_merge(EPetri.IA, (p, t)) //(p,
t)并入输入弧集合
11:  else pf is source of sf then //若顺序流 sf 由网关 pf 指
向其它节点
12:  let p ← seq_make_source(sf) //根据 sf 创建事件 p,
标记为 source
13:  EPetri.PL ← place_merge(EPetri.PL, p)
14:  EPetri.OA ← arc_merge(EPetri.OA, (t, p)) //(t,
p)并入输出弧集合
15:  end if
16:  end for

```

```

17: end for
18: return EPetri

```

由以上算法可以看出,网关结构的映射比较复杂,这是因为每种网关对应扩展 Petri 网结构不同,其触发后续路径的条件也不同。

3.3 形式化映射执行

采用 EMF (Eclipse Modeling Framework)^[23] 构建 BPMN 和扩展 Petri 网的元模型,并使用 ATL (ATLAS Transformation Language) 模型转换技术^[24] 完成模型形式化映射的执行。图 7 给出了 BPMN 模型形式化映射执行的流程。

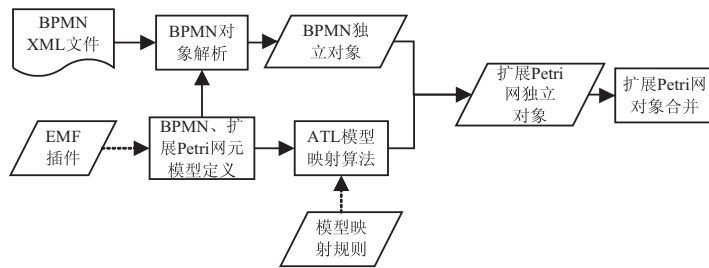


图 7 BPMN 模型形式化映射流程

考虑到生成的 BPMN 模型以 XML 文件的格式承载,使用 BPMN 模型解析器对 BPMN 模型进行解析,根据 BPMN 元素的类型生成 BPMN 独立对象。随后使用模型形式化映射算法将 BPMN 独立对象映射为扩展 Petri 网独立对象。最后使用扩展 Petri 网模型合并器即可将扩展 Petri 网独立对象合并为扩展 Petri 网对象。

根据映射算法将图 6 所示模型映射为扩展 Petri 网模型,映射后的扩展 Petri 网如图 8 所示。PNML (Petri Net Markup Language)^[25] 作为一种基于 XML 的用于各种 Petri 网的交换格式,广泛用于高级 Petri 网^[26] 的交换。因此,映射得到的扩展 Petri 网模型以 PNML 的格式存储。利用 ePNK^[27] 插件可以对 PNML 的扩展 Petri 网模型进行正确性分析与检验,从而可以检验 BPMN 模型的正确性。

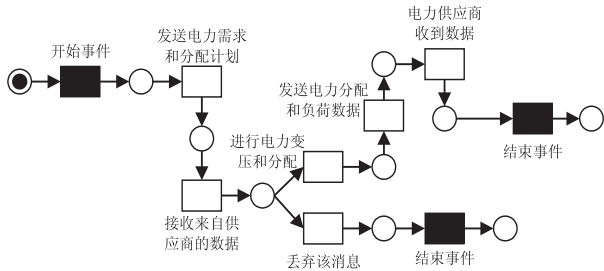


图 8 映射得到的扩展 Petri 网模型

4 实验结果与分析

4.1 实验设置

实验在一台操作系统为 Windows11 的计算机上进

行,其硬件配置为:RAM 为 16 GB,CPU 为 Intel® Core™ i7-13700H 2.4 GHz。实验数据集为各类电力业务行为描述(包括日志和文本),分别在业务行为描述数量为 2 000、4 000、8 000、12 000、16 000 和 20 000 的数据集下测试方法执行时间、内存和 CPU 开销,实验数据取得运行 20 次所得平均值。

4.2 实验结果及分析

图 9 给出基于自然语言处理的业务行为模型自动化生成方法各个阶段执行时间对比。可见,预处理和 BPMN 模型生成的时间开销较高,这是因为二者分别涉及自然语言处理和模型生成。另外,随着业务行为描述数量的增加,算法各个阶段所需的时间也在逐渐增加。

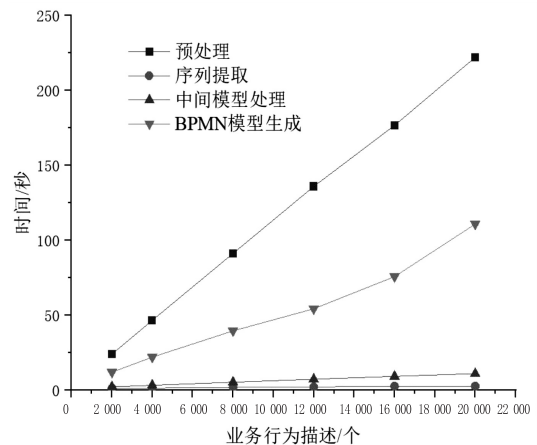


图 9 业务行为模型生成各阶段执行时间

图 10 给出业务行为模型生成方法执行总时间。可见,随着业务行为描述数量的增加,模型生成的时间

开销接近线性增长,且生成业务行为模型平均只需 17.3 ms 左右的时间开销。

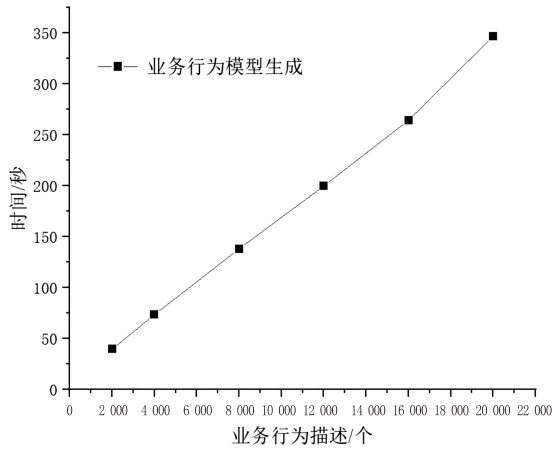


图 10 业务行为模型生成各阶段执行时间

图 11 给出基于扩展 Petri 网的业务行为模型形式化方法执行时间。可以发现,所提出的业务行为模型形式化算法即基于扩展 Petri 网对 BPMN 模型进行形式化,业务行为描述的平均执行时间开销约为 1 ms。

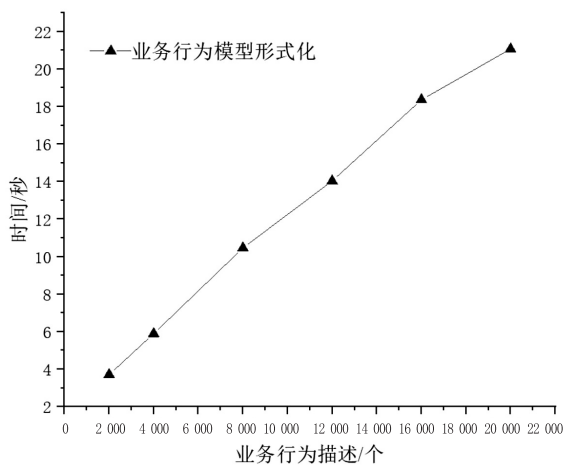


图 11 业务行为模型形式化执行时间

表 2 不同业务行为描述数量下内存和 CPU 占用情况

业务行为描述/个	业务行为模型生成		业务行为模型形式化	
	内存/MB	CPU/%	内存/MB	CPU/%
2 000	6.248	53.54	0.297 1	27.81
4 000	7.977	54.61	0.322 8	38.46
8 000	17.308	56.37	0.325	46.83
12 000	21.652	57.23	0.295	49.93
16 000	25.264	59	0.33	49.73
20 000	29.424	58.92	0.32	49.52

表 2 给出了在不同业务行为描述数量下,业务行为模型生成和业务行为模型形式化两个阶段的内存和 CPU 占用情况。前者的内存占用量和 CPU 占用率均高于后者,这是因为业务行为模型生成算法涉及业务行为文本描述、系统日志、行为序列、中间模型和

BPMN 模型的转换,较为复杂。另外,随着业务行为描述数量的增加,两种算法在 CPU 占用率上均呈先快后慢的趋势,不同的是前者增长较缓,后者在业务行为描述数量为 12 000 时趋于稳定。在内存占用方面,业务行为模型生成算法稳步增加,而业务行为模型形式化算法则趋于稳定不变。

通过上述实验结果及分析,可见该方法在较少时间、内存和 CPU 的开销下,即可生成符合实际场景的电力业务行为模型。

5 结束语

用电信息采集系统是新型电力系统的重要组成部分,是计量结算数据的唯一来源,为其它业务系统提供各项基础数据。其业务生态较为复杂,再叠加多元化影响,从而很难实现对业务行为的高效建模及分析。该文面向用电信息采集系统,提出一种基于 BPMN 和扩展 Petri 网的电力业务行为模型生成方法。首先利用自然语言处理技术对业务行为相关的文本描述或系统日志进行分析并提取关键词,生成中间模型。然后利用 BPMN 模型对中间模型进行解析,转化生成基于 BPMN 的业务行为模型。进一步,结合扩展 Petri 网模型,定义了 BPMN 模型形式化映射规则,实现对 BPMN 模型的形式化及分析。实验分析表明,该方法能够实现业务行为模型的自动化生成和形式化分析,可为大规模业务行为序列自动化分析和监控,尤其是对具有数据交互复杂特点的大量业务进行理解、分析和改进整体布局具有重要意义。当然,在实践中业务行为序列的规模可能很大且逻辑更复杂,这将导致生成和形式化业务模型的时空开销都很高。因此,下一步将重点研究如何减少限制性约束,以提高 BPMN 模型形式化的效率和准确性。

参考文献:

- [1] 胡江溢,祝恩国,杜新纲,等.用电信息采集系统应用现状及发展趋势[J].电力系统自动化,2014,38(2):131-135.
- [2] LARA MACHADO P,VAN DE VEN M,AYSOLMAZ B,et al. Methods that bridge business models and business processes:a synthesis of the literature[J]. Business Process Management Journal,2023,29(8):48-74.
- [3] LOPES T, GUERREIRO S. Assessing business process models: a literature review on techniques for BPMN testing and formal verification[J]. Business Process Management Journal, 2023,29(8):133-162.
- [4] 程航宇,康国胜,刘建勋,等.基于 BPMN 的业务流程建模元素扩展机制[J].计算机集成制造系统. doi:10.13196/j.cims.2023.0088.
- [5] 陈 虎.基于 BPMN 的建模研究与应用[D].鞍山:辽宁科

- 技大学,2014.
- [6] 汪玉泉,闻立杰,闫志强. 基于对齐的 BPMN 2.0 模型符合性检测算法[J]. 计算机研究与发展,2017,54(9):1920-1930.
- [7] SINTORIS K, VERGIDIS K. Extracting business process models using natural language processing (NLP) techniques [C]//2017 IEEE 19th conference on business informatics (CBI). Geneva:IEEE,2017:135-139.
- [8] 李宗花,周晓峰,吴克力,等. 基于扩展 Petri 网模型的 BPMN 形式化[J]. 计算机科学,2016,43(11):40-48.
- [9] CASSANDRAS C G, LAFORTUNE S, CASSANDRAS C G, et al. Petri nets [J]. Introduction to Discrete Event Systems, 2021,1(1):259-302.
- [10] CHINOSI M, TROMBETTA A. BPMN: an introduction to the standard[J]. Computer Standards & Interfaces,2012,34(1):124-134.
- [11] MURSYADA A. The role of business process modeling notation in process improvement; a critical review[J]. Advanced Qualitative Research,2024,2(1):41-56.
- [12] MOREIRA S, MAMEDE H S, SANTOS A. Business process automation in SMEs; a systematic literature review [J]. IEEE Access,2024,12(1):75832-75864.
- [13] CHOUDHARY R, RIAZ N. A business process re-engineering approach to transform business process simulation to BPMN model[J]. Plo S One,2023,18(3):1-25.
- [14] BELLAN P, DRAGONI M, GHIDINI C. Extracting business process entities and relations from text using pre-trained language models and in-context learning [C]//International conference on enterprise design, operations, and computing. Bozen-Bolzano; Springer International Publishing,2022:182-199.
- [15] LICARDO J T. A method for extracting BPMN models from textual descriptions using natural language processing [D]. Pula: University of Pula,2023.
- [16] SHING L, WOLLABER A, CHIKKAGOUDAR S, et al. Extracting workflows from natural language documents; a first step [C]//International conference on business process management. Sydney: Springer International Publishing, 2018: 294-300.
- [17] REBMANN A, VAN DER AA H. Extracting semantic process information from the natural language in event logs [C]//International conference on advanced information systems engineering. Melbourne: Springer International Publishing, 2021: 57-74.
- [18] SUKTHANKER R, PORIA S, CAMBRIA E, et al. Anaphora and coreference resolution; a review [J]. Information Fusion, 2020,59(1):139-162.
- [19] QUARESMA P, FINATTO M J B. Information extraction from historical texts; a case study [C]//DHandNLP@ PROPOR 2020. Evora: PROPOR,2020:49-56.
- [20] NOULAMO T, TANYI E, NKENLIFACK M, et al. Formalization method of the UML statechart by transformation toward Petri Nets [J]. IAENG International Journal of Computer Science,2018,45(4):505-513.
- [21] SHAILESH T, NAYAK A, PRASAD D. Transformation of sequence diagram to timed Petri net using atlas transformation language metamodel approach [J]. Journal of Software: Evolution and Process,2022,34(1):1-22.
- [22] HUANG E, MCGINNIS L F, MITCHELL S W. Verifying SysML activity diagrams using formal transformation to Petri nets [J]. Systems Engineering,2020,23(1):118-135.
- [23] STEINBERG D, BUDINSKY F, MERKS E, et al. EMF: eclipse modeling framework [M]. [s. l.]: Pearson Education, 2008: 10-56.
- [24] JOUAULT F, ALLILAIRE F, BÉZIVIN J, et al. ATL: a model transformation tool [J]. Science of Computer Programming, 2008,72(1-2):31-39.
- [25] HILLAH L M, KORDON F, PETRUCCI L, et al. PNML framework: an extendable reference implementation of the Petri Net Markup Language [C]//Applications and theory of petri nets: 31st international conference. Braga: Springer, 2010: 318-327.
- [26] DING Z, YANG R. Modeling and analysis for mobile computing systems based on petri nets; a survey [J]. IEEE Access, 2018,6(1):68038-68056.
- [27] KINDLER E. The ePNK: an extensible Petri net tool for PNML [C]//Applications and theory of petri nets; 32nd international conference, PETRI NETS 2011. Newcastle: Springer, 2011: 318-327.