一种面向SaaS多租户的多层模型

李 响[1] 李 彤[1,2] 谢仲文[1,2] 何 云[1] 成 蕾[1] 韩 煦[1]
(云南大学软件学院 昆明 650091)[1] (云南大学云南省软件工程重点实验室 昆明 650091)[2]

摘 要 SaaS(Software as a Service)伴随云计算而出现,它与传统软件的区别较大。根据SaaS软件的特点,提出支持SaaS软件成熟度的SaaS软件分层元模型,使用形式化方法对每一层进行建模描述。受面向对象Petri网(Object-Oriented Petri Nets,OOPN)和有色Petri网(Colored Petri Nets,CPN)思想的启发,提出面向服务网结构SOP(Service-Oriented Petri Nets)和CSOP(Colored Service-Oriented Petri Nets)。一方面,使用封装的库所元素代表服务,体现了服务对外不可见,且内部结构影响系统运行。另一方面,不同的颜色集代表不同租户请求,突出了SaaS多租户的特点。这不仅为复杂的SaaS软件建模提供了方法,还能够折叠系统变迁,压缩状态空间。最后,以一个客户关系管理(Customer Relationship Management,CRM)SaaS软件系统为例,验证了文中工作的可行性。

关键词 形式化方法,Petri网,SaaS,软件建模
中图法分类号 TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.11.009

Multi-layer Model for SaaS Multi-tenancy

LI Xiang[1] LI Tong[1,2] XIE Zhong-wen[1,2] HE Yun[1] CHENG Lei[1] HAN Xu[1]
(School of Software,Yunnan University,Kunming 650091,China)[1]
(Key Laboratory for Software Engineering of Yunnan Province,Yunnan University,Kunming 650091,China)[2]

Abstract SaaS(Software as a Service) appears with cloud computing. It is much different from traditional software. Based on the characters of SaaS,a hierarchical meta-model of SaaS was proposed,which supports the SaaS maturity model and describes every layer by formalization method. Inspired by object-oriented Petri nets (OOPN) and colored Petri nets(CPN),SOP(Service-Oriented Petri Nets) and CSOP(Colored Service-Oriented Petri Nets) were proposed. On the one hand,encapsulated place is used to represent service,which reflects that services are invisible to outside and inside influences running situation of system. On the other hand,different color sets represent the requests of different tenements,which emphasizes multi-tenancy character of SaaS. The work of this paper not only provides a function for modeling complex SaaS software,but also folds the transition of petri net and encapsulates state space of system. A CRM(Customer Relationship Management) system of SaaS was illustrated to verify the feasibility in the end.

Keywords Formalization method,Petri net,Software as a Service,Software modeling

1 引言

软件即服务(Service as a Software,SaaS)是近年来兴起的一种新的软件模式[1],它与传统的软件模式的区别较大。SaaS模式是指由软件服务商提供基于网络的软件服务,并完全负责维护软件服务所需的软硬件平台等后台工作,而用户则采用租赁的方式租用软件服务供应商提供的软件服务,并通过网络使用这些服务[2]。提到SaaS,人们会想到云计算(Cloud Computing)[3],其将提供的服务分为3个层次:IaaS(Infrastrucure as a Service),PaaS(Platform as a Service),SaaS[4]。而我们经常提到的SaaS实际上是云计算中的SaaS。Salesforce.com作为最成功的SaaS软件提供商,其客户关系管理(CRM)解决方案对SaaS产业产生了深远的影响。目前全世界超过15000家企业租用他们的服务[5]。

SaaS不仅改变了传统软件的商业模式,也改变了传统软件的生命周期[2]。与传统软件相比,SaaS软件最显著的特点主要有:可伸缩性(Saleable)、多租户架构(Multi-Tenant)和可配置性(Configurable)[6]。因此针对传统软件的研究方法和模型不再适用于SaaS。为了方便SaaS的研究,文献[6]根据SaaS软件的成熟度将SaaS分为4个等级:定制开发(多次开

到稿日期:2016-10-31 返修日期:2016-12-23 本文受国家自然科学基金(61379032,61262024,61462092),云南省教育厅科学研究基金(2014Y012)资助。

李 响(1993-),男,硕士生,主要研究方向为领域软件工程、软件演化,E-mail:450507961@qq.com;李 彤(1963-),男,博士,教授,CCF高级会员,主要研究方向为软件工程、信息安全;谢仲文(1982-),男,博士,讲师,CCF会员,主要研究方向为软件需求工程、软件演化等,E-mail:xiezw56@126.com(通信作者)。

发);可配置(一次开发多次部署);单实例,多租户(一次开发一次部署);多实例,多租户(可伸缩)。每一层是在满足前一层的要求的基础上再实现本层的功能。它描述了 SaaS 软件逐步完善的过程。SaaS 软件成熟度模型不仅是业界公认的标准,更成为了 SaaS 开发所遵循的一种规范。

Petri 网是一种描述软件过程活动的形式化工具[11],但基本 Petri 网面对复杂系统时存在状态空间爆炸的问题。有色 Petri 网(CPN)在基本 Petri 网的基础上进行扩展,使用不同颜色集来描述系统状态。它能有效地折叠系统变迁,压缩状态空间。与传统软件相比,SaaS 软件的架构和运行情况十分复杂,尤其是 SaaS 软件中多租户的特点,使得 SaaS 软件难以描述。本文旨在使用有色 Petri 网形式化建模出能够支持 SaaS 软件成熟度的模型。通过使用不同的颜色集代表不同租户请求,折叠了大部分变迁,将 SaaS 多租户复杂的运行情况用可视化的形式化工具有色 Petri 网进行描述。本文借鉴了面向对象 Petri(OOPN)[12] 的思想,分别提出了 CSOP 和 SOP 系统。之后,根据 SaaS 软件的特点,对 SaaS 软件进行分层,提出了 SaaS 软件元模型[13]。

本文的主要贡献如下:

(1)根据 SaaS 软件成熟度模型,提出一种对 SaaS 软件进行分层建模的方法。将 SaaS 软件自顶向下按照不同的粒度分为 7 个不同的层次,体现了 SaaS 软件多租户、可伸缩、可配置的特点。

(2)有色 Petri 网的颜色集可以代表不同租户的请求,面向对象 Petri 网体现了服务的封装性,它对外不可见,内部细节影响系统运行。分别借鉴有色 Petri 网和面向对象 Petri 网的思想,对 Petri 网进行扩展,提出了 CSOP 和 SOP 系统,为 SaaS 软件的建模提供了形式化工具。

(3)本文提出的形式化模型能够在有色 Petri 网工具 CPN 中运行,从而可以模拟 SaaS 软件的运行,分析系统运行时存在的隐患。

本文第 2 节分析了 SaaS 软件的建模思路,提出了 SaaS 软件的元模型;第 3 节根据提出的 SaaS 元模型讨论了具体的形式化建模;第 4 节对 CRM 的 SaaS 软件案例进行了分析;第 5 节对相关的 SaaS 研究工作进行了介绍;最后总结全文。

2　建模思路

根据 SaaS 软件的多租户、可配置、可伸缩这 3 个主要特点,自上而下分析 SaaS 软件的构成。根据 SaaS 软件成熟度模型,不同租户的需求在 SaaS 软件中是可配置的。可配置体现在用户界面、用户管理、数据管理、应用程序管理的可配置[5]。因此所有租户共用相同的软件实例,不同租户的配置使得同一软件实例能够满足多个租户个性化的需求,符合软件成熟度第三级的描述。可伸缩性体现在 SaaS 软件可以横向扩展软件实例。当单一软件实例的负载过高时,势必会影响 SaaS 的服务质量,增加新的软件实例能够有效降低单一软件实例的负载。同时,当 SaaS 软件的用户减少时,多余的软件实例会占用过多的系统资源,因此将租户数量较少的软件实例合并和释放,能有效节省系统资源。这是最高级的 SaaS 软件成熟度。

用户使用 SaaS 软件,通过订阅 SaaS 中的不同服务来实现自己的需求。服务是指服务的组合,本文用功能来区别原子服务。所有的用户功能均由原子服务组合构成。SaaS 软件采用 SOA 的架构,SOA 本身是一个松耦合的架构,服务之间相互独立,可以自由组合[14],这使得 SaaS 软件拥有可配置性。不同原子服务的交互组合实现了某个租户的某种需求,租户通过租用对其进行使用。于是将 SaaS 软件分为 7 层:应用层、租户层、软件实例层、表示层、业务流程层、服务实体层和基础设施层。本文分别从 7 个不同的层次来描述 SaaS 软件,并提出元模型,结构如图 1 所示。

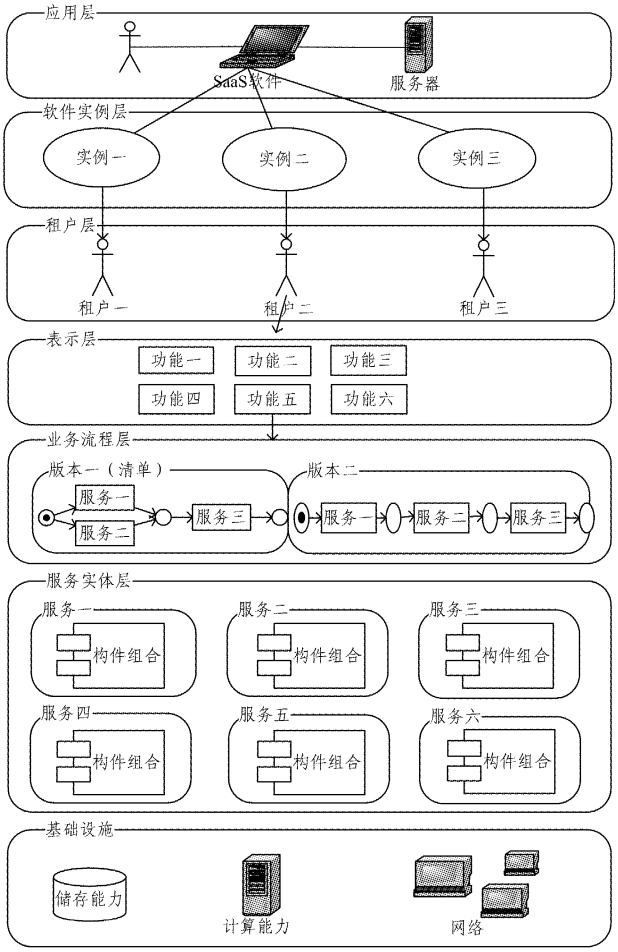

图 1　SaaS 软件元模型

应用层:SaaS 软件的最大粒度,强调 SaaS 软件为用户提供的价值。

软件实例层:根据 SaaS 软件成熟度模型,SaaS 软件使用多实例来解决负载均衡的问题。负载均衡器会评估当前软件实例的负载情况,不同的租户会被负载均衡器分配到不同的实例当中。如果某个 SaaS 软件是单实例,那么在软件实例层存在一个实例即可。

租户层:表示 SaaS 软件中的不同租户。SaaS 会根据不同的用户定制不同的 SaaS 软件,通过配置不同租户的设置来实现界面、数据、服务的可配置化。

表示层:该层主要与用户进行交互,代表用户的界面,承载用户在 SaaS 上租用的所有功能。

业务流程层:每一个功能对应一种服务的组合方式,该层没有具体的服务,只是服务的组合清单,不同的版本产生不同的服务组合清单,以清单的形式来配置服务的组合方式。

服务实体层:该层主要是服务的具体定义,每一个服务的内部对外是不可见的,服务的内部通过构件的组合来实现具体的功能。

基础设施层:该层为 SaaS 软件所运行的硬件环境,抽象为储存能力、计算能力、网络。

多租户的特点体现在租户层,该层拥有多个使用 SaaS 软件的租户,不同的租户会被分配到不同的软件实例中,不同的软件实例描述其不同租户的行为。

可伸缩的特点体现在软件实例层,一个 SaaS 软件可以拥有多个软件实例。多个实例使得不同租户能够根据负载情况被分配到不同的软件实例中。于是将软件实例层单独提出。

可配置体现在表示层,配置租户表示层使不同的租户拥有个性化的用户界面、用户管理、数据管理、应用程序等[15]。每一个表示层功能对应一种具体的服务组合方式。表示层中的功能是抽象的,用户可以随时添加、删除和修改。

该元模型主要描述 level3 和 level4 的 SaaS 成熟度,其也是目前 SaaS 较为流行的两种模式。在 level3 中,SaaS 软件为单实例多租户的模式。元模型的软件实例层中仅有一个软件实例,所有的租户均共享该软件实例。level4 的特点是多实例多租户,软件实例层的实例数目增加为多个,不同的租户通过负载均衡器被分配到不同的软件实例中。

元模型的主要作用是建模 SaaS 软件,它是建立 SaaS 软件模型的模型。针对不同 SaaS 软件,根据元模型逐层建模,将复杂的 SaaS 软件分为不同的层次描述。每一个 SaaS 软件实例被描述在一张有色 Petri 网上。若所描述的软件为 level3的成熟度,则单个的软件实例即为 SaaS 软件本身;若描述的为 level4 成熟度的 SaaS,则多个软件实例组成该 SaaS 软件,它们均有一张有色 Petri 网。不同软件实例的运行情况组成了 SaaS 软件的运行情况。

所建的 SaaS 模型在形式上使用集合描述各个层元素的从属和对应关系,以保证模型的描述清晰、准确,没有二义性,使得模型便于分析。例如,通过能够快速地从模型中得到哪个租户被分配到哪个软件实例,又租用了哪些功能。同时,模型的可视化部分能够模拟软件实例的运行,通过分析 Petri 网的公平性、活性、有界性等性质,能够发现并找出系统存在的问题和隐患。例如,通过分析 Petri 网的活性能够找出系统中存在的死链,这是系统存在的故障,应当及时解决;通过分析 Petri 网中的公平性,能够及时查找出由于设计而产生的租户之间不健康的竞争关系;有界性的分析保证了系统不会处在无限制的运行中。

3 形式化建模

该模型分别在 3 处使用 Petri 网进行描述,它是 SaaS 软件模型的可视化部分:原子服务内部组成结构、业务流程层原子服务的组合方式、软件实例层中各个软件实例不同租户的运行情况。受一种高级 Petri 网——面向对象 Petri 网思想的启发,本文提出一种面向服务的网结构。面向对象 Petri 网的提出是为了解决面向对象的相关问题[16]。面向对象的最重要的一个特征是封装性,对象的内部组成对外部不可见。这恰好也是服务的特点。服务之间是相互独立的,向外部暴露的信息很少,几乎只有输入和输出,外部对服务内部是不可见的;而在服务的内部,每个服务都有自己的组成方式,每个服务的内部组成会影响系统的运行。本文在满足系统描述的完整性的基础上,突出模型的特点,忽略枝节,引入了面向对象 Petri 网的表达描述方式,将服务描述为一个含有输入和输出库所的结构。具体定义如下。

定义 1(面向服务网结构 SOP) SOP 定义为一个扩展的基本 Petri 网。$SOP=\{P,T;F,M_0,I_s\}$,其中:

1)$\{P,T;F\}$为一个基本网系统,P 为网的库所,T 为变迁,F 是定义在 P 到 T 的有向弧。$P\cup T\neq\emptyset$,$P\cap T=\emptyset$,$F\subseteq(P\times T)\cup(T\times P)$。这里的 P 表示服务实体,服务实体均来自于应用层中的服务实体 $P\subseteq S$。同时,P 本身为一个扩展的基本 Petri 网,表示服务的内部结构。

2)M_0 为 SOP 的初始标识。

3)I_s 为 SOP 的唯一表示,对外表示整个网系统。

4)在该网系统中,点火规则和变迁的图形表示与 Petri 网完全相同,不同的是代表服务实体的库所用带有库所的矩形框表示。服务实体上的库所分别代表服务的输入和输出,服务与服务之间通过变迁进行通信和组合。

图 2 给出了 SOP 未细化时的图形表示,未细化的服务实体代表一个库所元素,上述的输入、输出库所是等价的,其中 token 元素可以任意移动。

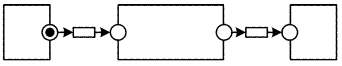

图 2 SOP 的图形表示

5)服务实体可由服务子网来细化,内部为服务实体的具体实现,可由基本 Petri 网进行填充,保证了服务对外的不可见,其他服务只能通过输入、输出库所与该服务交互;另一方面,体现了服务内部结构的完整性,单个服务的内部组成影响整个系统的运行情况。

SOP 网能够细化,细化的服务实体上的输入、输出库所不再等价,其内部发生规则严格遵守 Petri 的点火规则,如图 3 所示。token 不能随意地在输入和输出库所中移动。

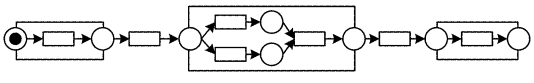

图 3 SOP 的细化

为了使该网结构能够描述软件实例层多租户的运行情况,对该网进行扩展,使用有色 Petri 网中的颜色集来描述不同租户的运行情况,使该网系统能够描述多租户,同时使用该软件实例的状态。具体定义如下。

定义 2(有色面向服务网结构 CSOP) 将 CSOP 定义为一个扩展的有色 Petri 网的八元组 $CSOP=\{P,T;F,C,W,M,I_s,II_s\}$,其中:

1)$\{P,T;F\}$为一个基本网系统,P 为网的库所,T 为变

迁，F 是定义在 P 到 T 的有向弧。$P\cup T\neq\varnothing$，$P\cap T=\varnothing$，$F\subseteq(P\times T)\cup(T\times P)$。这里的 P 表示服务实体，服务实体均来自于应用层中的服务实体 $P\subseteq S$。同时，P 本身为一个扩展的基本 Petri 网，表示服务的内部结构。

2）C 为 SOP 的颜色集合，是一个有限非空集合，用来表示服务中的不同租户。C 中元素的个数与租户数相同，$C=Ir$。

3）$W:F\rightarrow L(C)+$

$I_s:PN\rightarrow L(C)+$

$M:P\rightarrow L(C)$

$L(C)$ 表示定义在颜色集 C 上的一个非负系数线性函数，$L(C)+$ 表示系数不全为 0 的 $L(C)$。W 为 F 到颜色集的权值函数，I_s 为定义在变迁上的映射，M 为库所集到颜色集的映射。

4）CSOP 的图形表示与 SOP 相同。

5）II_s 为该网结构的唯一标识。

多个软件实例之间相互独立。每个软件实例的运行情况的总和组成整个复杂 SaaS 软件系统的运行状态。

根据上文的分层 SaaS 元模型，从上至下描述的粒度越来越小，各层之间存在包含的关系，应用层为 SaaS 的最大粒度，应包含所有层。由于租户被分配到不同的软件实例中，因此软件实例层与租户层为包含关系，每个租户拥有自己的表示层，租户层包含表示层。表示层中的每一个功能均需要业务流程层来描述原子服务的组合关系，因此表示层与业务流程层也是包含关系。对于 SaaS 而言，无论 SaaS 软件的实例如何变化，其底层的原子服务均不会发生变化。因此将服务实体层放在应用层中进行定义。SaaS 软件元模型中每层的具体定义如下。

定义 3（应用层）　将应用层定义为一个三元组，则可以表示为 $A=\{N,I,S\}$，其中：

1）N 为该 SaaS 软件的唯一标识。

2）I 为 SaaS 软件的软件实例层，描述 SaaS 包含的软件实例。

3）S 为 SaaS 软件的服务实体层，包含 SaaS 软件所有的原子服务，所有的功能均由这些服务实体组成。S 由基本 Petri 网表示原子服务的内部结构。

应用层为 SaaS 软件的最大粒度，对应用层的定义可以看作对 SaaS 整体的定义。SaaS 包括应用的名称、软件实例和服务实体。由于不同的租户被分配到不同的软件实例中，因此服务实体层包含租户层。SaaS 软件的可配置性导致租户层包含租户层以下的所有层，即软件实例层包含它以下的所有层。由于 SaaS 中的原子服务不会随着软件实例租户的配置而改变，因此将它单独提到应用层。

定义 4（软件实例层）　将软件实例层定义为一个元组，可以表示为 $I=\{I_i,U,CSOP,F_i\}$，其中：

1）I_i 为实例层的唯一表示，一个 I_i 元素代表一个软件实例。

2）U 为 SaaS 软件的租户，描述租户层。

3）$CSOP$ 为一个有色的面向服务的网结构，用来描述该 SaaS 软件实例的结构，其中不同的颜色代表不同的租户。

4）F_i 为 I_i，U，$CSOP$ 的对应关系，表示不同的租户被分配到不同的 SaaS 软件实例中，并由一个 $COSP$ 表示。$F_i\subseteq I_i\times U\times CSOP$，$I_i$ 与 U 是一对多的关系，I_i 与 $CSOP$ 的为一对一的关系。

定义 5（租户层）　租户层 U 为三元组，表示为 $U=\{I_u,R,F_u\}$，其中：

1）I_u 为租户的唯一标识，I_u 中有多个租户，表示 SaaS 软件中的所有租户。

2）R 为 SaaS 软件为每个用户配置的表示层，不同的租户拥有不同的表示层。

3）F_u 为表示层与租户的对应关系，$F_u\subseteq I_u\times R$，一个租户对应一个表示层，一个表示层可以对应多个租户，为一对多的关系。

定义 6（表示层）　表示层 R 为元组，表示为 $R=\{I_r,SOP,S_r,F_r\}$，其中：

1）I_r 为一个表示层的唯一标识，标识一个表示层。

2）SOP 为一个面向服务网。

3）S_r 为表示层中所有功能的集合。

4）F_r 为 I_r，SOP，S_r 的对应关系，$F_r\subseteq I_r\times SOP\times S_r$，$SOP$ 与 S_r 是一对一的关系，I_r 与 S_r 为一对多的关系。

定义 7（业务流程层）　将业务流程层描述为一个对象服务的网结构 SOP，$SOP=\{P,T;F,M_0,I_s\}$。

业务流程层描述的是服务的组合关系，这种组合关系用面向服务网结构 SOP 进行描述。

定义 8（服务实体层）　将服务实体层描述为一个扩展的基本 Petri 网，表示为 $S=\{P_s,T_s;F_s,M_0,I_s\}$，其中：

1）$(P_s,T_s;F_s)$ 为一个基本网结构。P_s 为网的库所，T_s 为变迁，F_s 为从 P_s 到 T_s 的有向弧。

2）M_0 为该 Petri 网的初始标识。

3）I_s 为该网系统的唯一标识，对外表示整个网系统。

4　案例分析

SaaS 在 CRM 系统上最先取得成功，并且目前多数 SaaS 应用均为 CRM 系统。本文以 CRM 系统为例来验证本文方法的可行性。

该案例是一个简化的 CRM 系统。它共有 4 类租户，每类租户分别租用不同的功能，各个功能分别由原子服务组合而成。租户一、租户三和租户四分别为商家的员工。租户一负责分析市场的商机，为商家寻找客户。租户三负责分析市场，为商家制定市场计划。租户四负责制定市场活动计划。租户二是一位客户，主要使用系统查询市场信息，寻找自己需要的商家。该 CRM 的结构功能图如图 4 所示。

租户大致租用了以下几个功能。

客户推荐：根据其他人的评论为商家推荐合适的客户；由客户信息服务和评价服务组成。

客户管理：为商家管理客户信息，方便联系人的查找、修改、删除；由客户信息服务和市场信息服务组成。

商机分析：根据市场信息和客户信息，为商家绘制报表，分析存在的商机；由客户信息服务、市场信息服务和报表服务组成。

市场信息分享：将查阅到的市场信息分享到第三方社交网络；由市场信息服务和第三方社交网络服务组成。

商家推荐:根据其他人的评论,为用户推荐需要的商家;由商家信息服务和评价服务组成。

潜在客户分析:根据市场情况和各个客户的行为绘制报表,为商家分析潜在的用户;由客户信息服务、市场信息服务、评论服务和报表服务组成。

市场管理:商家查询市场信息、管理市场活动;由客户信息服务、市场信息服务和商家信息服务组成。

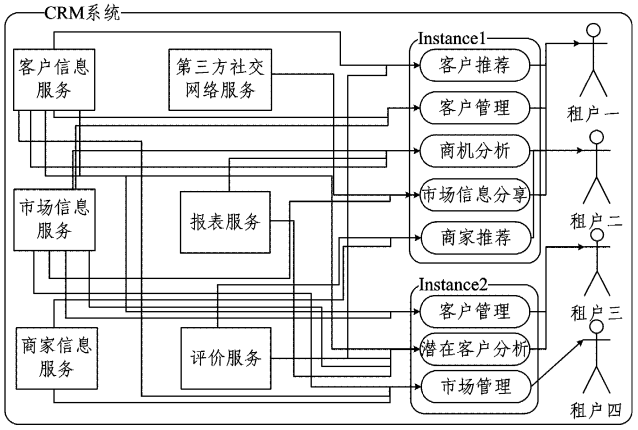

图 4 CRM 系统结构图

根据 SaaS 软件元模型,从下而上建立模型。

(1)服务实体层

为了方便描述,分别对服务的名称进行简写,其中客户信息服务为 CIservice,市场信息服务为 MIservice,商家信息服务为 OIservice,第三方社交软件服务为 SNservice,报表服务为 RFservice,评论服务为 Cservice。服务实体层描述为一个 Petri 网,具体结构如图 5 所示。

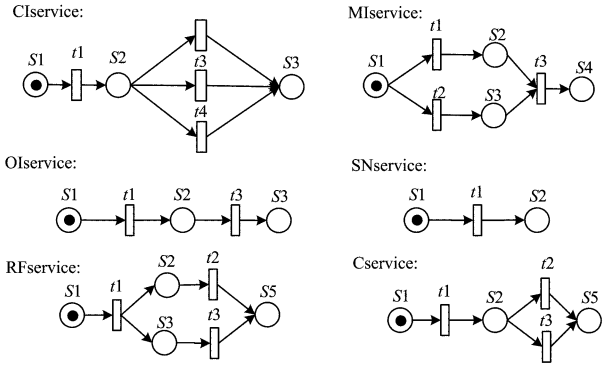

图 5 服务实体层 Petri 网模型

(2)业务流程层

$SOP=\{S1,S2,S3,S4,S5,S6,S7\}$,图形表示如图 6 所示。

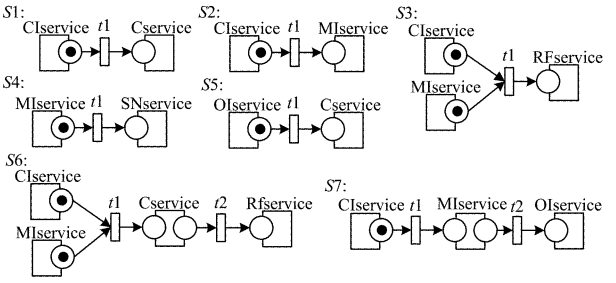

图 6 业务流程层 Petri 网模型

(3)表示层

$R=\{I_r,SOP,S_r,F_r\}$

$I_r=\{r1,r2,r3,r4\}$

$S_r=\{Client\ Recommend,Client\ Manage,Business\ Analyse,Market\ Information\ Share,Organization\ Recommend,Potential\ Client\ Analyse,Market\ Manage\}$

$SOP=\{S1,S2,S3,S4,S5,S6,S7\}$

$F_r=\{(r1,Client\ Recommend,S1),(r1,Client\ Manage,S2),(r1,Business\ Analyse,S3),(r2,Market\ Information\ Share,S4),(r2,Organization\ Recommend,S5),(r3,Client\ Manage,S2),(r3,Potential\ Client\ Analyse,S6),(r4,Market\ Manage,S7)\}$

(4)租户层

$U=\{I_u,R,F_u\}$

$I_u=\{rt1,rt2,rt3,rt4\}$

$R=\{r1,r2,r3,r4\}$

$F_u=\{(rt1,r1),(rt2,r2),(rt3,r3),(rt4,r4)\}$

(5)实例层

$I=\{I_i,U,CSOP,F_i\}$

$I_i=\{Instance1,Instance2\}$

$U=\{rt1,rt2,rt3,rt4\}$

$CSOP=\{c1,c2\}$

$F_i=\{(Instance1,r1,c1),(Instance1,r2,c1),(Instance2,r3,c2)\}$

$c1$:其中颜色集 $R1$ 和 $R2$ 分别代表租户一和租户二的请求。弧上的 $r1$ 和 $r2$ 分别限制了颜色集的流向,它们是弧上的限制函数,颜色集 $R1$ 流经标有 $r1$ 的弧,$R2$ 与 $R1$ 类似,只能流经标有 $r2$ 的弧。两个颜色集均能流经标有 r 的弧。其中,变迁 $t6$,$t8$,$t10$ 上有抑制弧,以保证当服务 *CIservice* 和 *MIserviceR1* 中都有 token 时,只有 $t9$ 变迁有发生权,并使服务按用户所租用的功能正常组合。图形表示如图 7 所示。

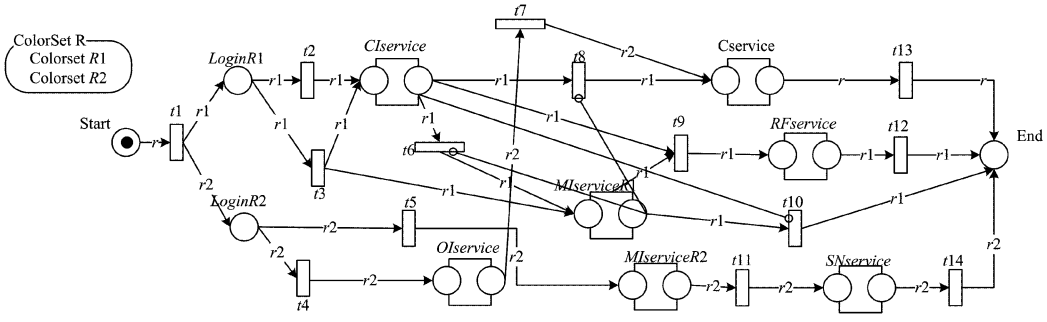

图 7 软件实例一(Petri 网模型 $c1$)

$c2$：与 $c1$ 类似，$c2$ 共有两个颜色集 $R3$ 和 $R4$，分别代表租户三和租户四。$R3$ 只能流经 $r3$ 函数限制的弧，$R4$ 只能流经 $r4$ 函数限制的弧。$R3$ 和 $R4$ 均能通过标有 r 的弧。其中变迁 $t6$，$t9$ 上有抑制弧，与软件实例一类似，用于保证当服务 *CiserviceR3* 和 *Miservice* 中都有 token 时，只有变迁 $t7$ 有发生权。其图形表示如图 8 所示。

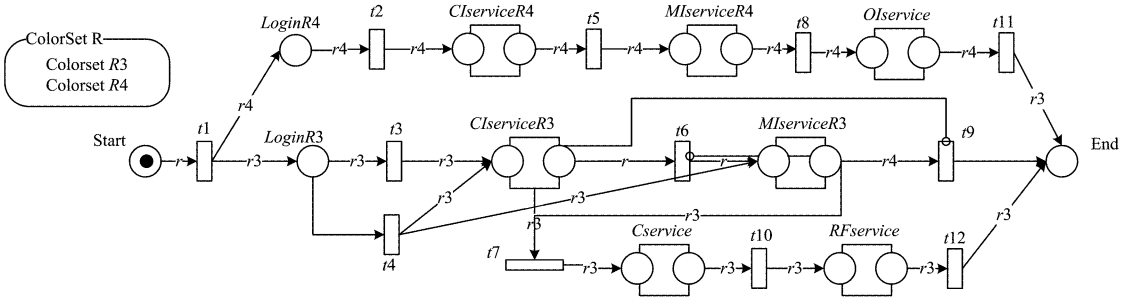

图 8　软件实例二（Petri 网模型 $c2$）

(6)应用层

$A=\{N,I,S\}$

$N=\{CRM\ System\}$

$I=\{Instance1,Instance2\}$

$S=\{CIservice,MIservice,OIservice,SNservice,RFservice,Cservice\}$

分别将两个软件实例模型放入 CPN tool 中进行模拟。其中，所有的库所元素均能容纳复合颜色集 RN，RN 中包含不同租户的颜色集。颜色集上的数字分别代表租户编号。弧上有限制函数 flow1()，flow2，flow3()，flow4()，限制了不同颜色集的流向，只有特定的租户请求才能通过，保证了 SaaS 软件多租户在同一系统中正常运行。所有租户请求均能通过标有 r 的弧，表明有多个租户的请求共用该条弧。限于篇幅，此处未对 Petri 网中的服务进行细化。模拟情况分别如图 9 和图 10 所示。

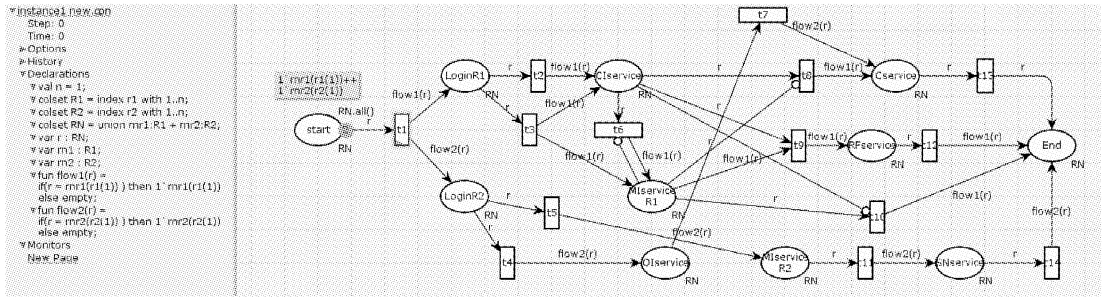

图 9　模拟软件实例 1

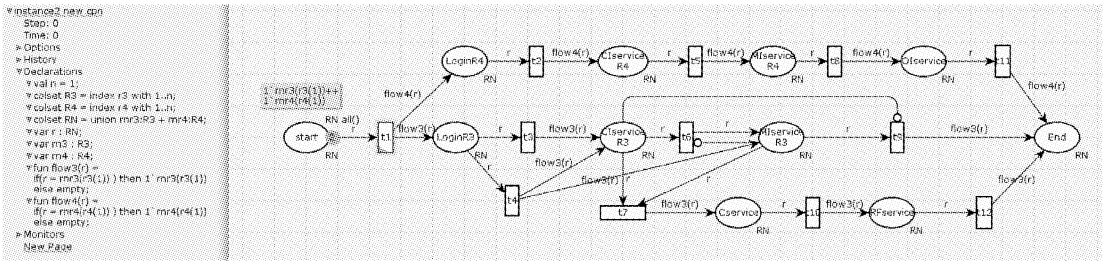

图 10　模拟软件实例 2

软件实例一在 cpntool 中的状态空间分析报告如下：

Statistics

————————————————

State Space

Nodes：56

Arcs：119

Secs：0

Status：Full

Scc Graph

Nodes：56

Arcs：119

Secs：0

软件实例二在 cpntool 中的状态空间分析报告如下：

Statistics

————————————————

State Space

Nodes：48

Arcs：88

Secs：0

Status：Full

Scc Graph

Nodes：48

Arcs：88

Secs：0

从上述两个状态空间报告可以看出，软件实例一共有 56 个节点、119 条弧；软件实例二共有 48 个节点、88 条弧。两个软件实例的模型均为强连通图，表明软件模型具有较好的联通性。

cpntool 中的有界性分析报告分别如下。

软件实例一：

Boundedness Properties

————————————————————

Best Integer Bounds

	Upper	Lower
New_Page'CIservice 1	1	0
New_Page'Cservice 1	2	0
New_Page'End 1	2	0
New_Page'LoginR1 1	1	0
New_Page'LoginR2 1	1	0
New_Page'MIservice_R1 1	1	0
New_Page'MIservice_R2 1	1	0
New_Page'OIservice 1	1	0
New_Page'RFservice 1	1	1
New_Page'SNservice 1	1	1
New_Page'start 1	2	1

软件实例二：

Boundedness Properties

————————————————————

Best Integer Bounds

	Upper	Lower
New_Page'CIservice_R3 1	1	0
New_Page'CIservice_R4 1	1	0
New_Page'Cservic 1	1	0
New_Page'End 1	2	0
New_Page'LoginR3 1	1	0
New_Page'LoginR4 1	1	0
New_Page'MIservice_R3 1	1	0
New_Page'MIservice_R4 1	1	0
New_Page'OIservice 1	1	1
New_Page'RFservice 1	1	1
New_Page'start 1	2	1

以上数据给出了每个库所中拥有最多和最少的 token 数目。可以看出两个软件实例的所有库所中的 token 数目不超过 2 个，因此系统均是有界的，不会无限制地运行。软件实例一的活性分析报告如下：

Liveness Properties

————————————————————

Dead Markings

[56]

Dead Transition Instances

None

Live Transition Instances

None

软件实例二的活性分析报告如下：

Liveness Properties

————————————————————

Dead Markings

[48]

Dead Transition Instances

None

Live Transition Instances

None

从活性分析报告可以看出，两个软件实例仅有一个死标识，均为各自的终点结点，这符合系统的设计；此外再没有死链，这表明该系统设计合理，没有故障。软件实例一和软件实例二的公平性分析报告如下：

Fairness Properties

————————————————————

No infinite occurrence sequences

两个软件实例均没有不公平的序列，表明多个租户之间不存在不健康的竞争关系。通过对该 SaaS 软件进行建模，从不同层次展现了复杂的 CRM 系统，模块之间的从属关系十分清晰，可视化部分的有色 Petri 网能够模拟多租户在各自软件实例中的运行情况。通过分析每个软件实例的运行报告可以得出，该 SaaS 不存在死锁、死循环等隐患和故障。

5 相关工作

SaaS 软件的许多研究工作是使用 Petri 网对 SaaS 软件进行建模展开的。文献[7]提出一个 Petri 网模型，建模了 SaaS 软件的花费情况，为企业选择传统软件还是 SaaS 提供了依据，但并未描述 SaaS 系统的行为。文献[8]使用有色 Petri 网对 SaaS 进行建模，提出了一种预测 SaaS 行为的方法，但没有体现 SaaS 多租户、可伸缩、可配置的特点。还有一些工作使用其他方法对 SaaS 软件进行了建模研究。文献[9]提出了一种支持多租户架构 SaaS 动态演化的方法，为 SaaS 的升级更新提供策略，但并未对所研究系统进行形式化描述。文献[10]提出了一种多租户架构的 SaaS 软件模型，使用数学的方法严格定义了服务质量，反映了 SaaS 在多租户情况下的服务情况，但该模型没有图形表示，不能直观地表现 SaaS 的行为。

针对上述工作中存在的问题，本文提出 SaaS 软件元模型，并使用形式化的方法对其进行定义。元模型支持 SaaS 软件成熟度，充分体现了 SaaS 多租户、可伸缩、可配置的特点。在元模型的软件实例层、业务流程层和服务实体层中使用 Petri 网建模了模块间的关系。图形表示直观地反映了系统内部行为。软件实例层采用有色 Petri 网，不同颜色集表示不同租户的请求，这不仅在一定程度上解决了 Petri 网状态空间爆炸的问题，同时更好地模拟了系统行为。

结束语 本文提出的 SaaS 软件元模型为建模 SaaS 软件提供了框架和方法，不仅体现了 SaaS 软件可配置、可伸缩的

特性,逐步细化的粒度也使得复杂的 SaaS 软件脉络清晰。同时,有色面向服务网结构描述的 SaaS 软件实例解决了多租户的难题,体现了 SaaS 软件成熟度模型的描述。多层 SaaS 软件模型描述 SaaS 软件的结构,Petri 网模型模拟 SaaS 软件的运行情况,不仅能分析出 SaaS 软件系统存在的隐患,还能根据系统运行结果分析服务之间的交互关系,为研究 SaaS 软件的动态演化性奠定了基础。

今后的工作将以现有工作为基础,为 SaaS 软件建立多层模型,获取各个颜色集的可达图。一方面,分析多租户请求服务的行为以及服务之间的交互关系,研究服务行为的相关性和一致性,保证 SaaS 软件动态演化的正确实施。另一方面,文中对 SaaS 软件多租户进行的描述为进一步研究多租户如何影响 SaaS 软件的动态演化提供了方法。

参 考 文 献

[1] ALKALBANI A M,GHAMRY A M,HUSSAIN F K,et al. Sentiment Analysis and Classification for Software as a Service Reviews[C]// Proceedings of the IEEE International Conference on Advanced Information Networking and Applications. 2016:53-58.

[2] CHEN X B,WU Z X. Architecture of software services based on SaaS model supporting multi-terminals and service customization [J]. Journal of Computer Applications, 2010, 30(10): 2754-2757. (in Chinese)
陈小兵,武泽旭. 支持多类终端与服务定制的 SaaS 软件服务架构 [J]. 计算机应用,2010,30(10):2754-2757.

[3] CHEN Q,DENG Q N. Cloud computing and its key techniques [J]. Journal of Computer Applications,2009,29(9):2562-2567. (in Chinese)
陈全,邓倩妮. 云计算及其关键技术 [J]. 计算机应用,2009,29(9):2562-2567.

[4] CAO Y,LUNG C H,AJILA S A. Constraint-Based Multi-Tenant SaaS Deployment Using Feature Modeling and XML Filtering Techniques[C]// Proceedings of the International Workshop on Software Cybernetics. 2015:454-459.

[5] HOU K J,BAI X Y,ZHOU L Z. A multi-tenant system based on constraint combination configuration testing technology [J]. Journal of Computers,2016,39(2):237-252. (in Chinese)
侯可佳,白晓颖,周立柱. 一种基于多约束组合的多租户系统配置测试技术[J]. 计算机学报,2016,39(2):237-252.

[6] CHONG F,CARRARO G. Architecture strategies for catching the long tail [OL]. http://msdn. microsoft. com/en-us/library/aa479069. aspx.

[7] RIBAS M,LIMA A S,SOUZA N,et al. Assessing cloud computing SaaS adoption for enterprise applications using a Petri net MCDM framework[C]// 2014 IEEE/IFIP Network Operations and Management Symposium. IEEE,2014.

[8] GONG Z,YING S,LI L,et al. Performance prediction for saas deployment optimization based on colored petri nets[C]// Proceedings of the International Conference on Mechatronic Sciences,Electric Engineering and Computer. 2014:2689-2693.

[9] GEY F,LANDUYT D V,JOOSEN W,et al. Continuous Evolution of Multi-tenant SaaS Applications:A Customizable Dynamic Adaptation Approach[C]// 2015 IEEE/ACM 7th International Workshop on Principles of Engineering Service-Oriented and Clould Systems. 2015:10-16.

[10] SU W,LIN C,MENG K,et al. Modeling and Analysis of Availability for SaaS Multi-tenant Architecture[C]// Proceedings of the IEEE International Symposium on Service Oriented System Engineering. 2014:365-369.

[11] XIE Z W,LI T,DAI F,et al. Modelling dynamic evolution-oriented software architecture based on Petri net[J]. Computer Applications and Software,2012,29(10):36-39. (in Chinese)
谢仲文,李彤,代飞,等. 基于 Petri 网的面向动态演化的软件体系结构建模 [J]. 计算机应用与软件,2012,29(10):36-39.

[12] FAN Y S,ZHANG J. Object-oriented Petri net method and its application in the software engineering [J]. Journal of Computer Applications,1998,18(5):15-18. (in Chinese)
范玉顺,张军. 面向对象的 Petri 网方法及其在软件工程中的应用 [J]. 计算机应用,1998,18(5):15-18.

[13] XIE Z W,LI X Y,LI T,et al. A demand model into the method of software architecture model [J]. Computer Science, 2014, 41(5):196-203. (in Chinese)
谢仲文,李晓燕,李彤,等. 一种将需求模型转换为软件体系结构模型的方法[J]. 计算机科学,2014,41(5):196-203.

[14] PATERSON D,FOX,et al. SaaS software engineering:cloud computing era of agile development [M]. Tsinghua University Press,2015.

[15] MIETZNER R,METZGER A,LEYMANN F,et al. Variability modeling to support customization and deployment of multi-tenant-aware Software as a Service applications[C]// Proceedings of the The Workshop on Principles of Engineering Service Oriented Systems. 2009:18-25.

[16] PINCI V O,SHAPIRO R M. An integrated software development methodology based on hierarchical colored Petri nets[J]. Proceedings of the Advances in Petri Nets 1991, Papers From the nternational Conference on Applications and Theory of Petri Nets,1990,524(10):227-252.