



一种基于体系特性的 SaaS 系统评估框架^{*}

张秀伟,何克清,王 健,王 翀

(武汉大学软件工程国家重点实验室 武汉 430072)

摘 要:随着云计算、物联网和软件服务化等新技术的不断发展,在强调深入感知和互联互通互操作的过程中,SaaS 应用系统越来越复杂,如何从体系的视角对这类软件系统的性能指标进行评估和预测成为新的研究方向。首先从体系的定义中抽取体系独有的特征概念对 SaaS 系统进行描述,并根据体系特征间的依赖关系建立了 SaaS 系统评估框架,最后以 SaaS 系统 CloudCRM 为例,并利用文中的方法进行评估,验证了方法的可行性和有效性。

关键词:体系;面向服务架构;软件即服务;体系效能;负涌现

doi: 10.3969/j.issn.1000-0801.2013.12.015

Assessment Architecture of SaaS System Based on Characteristics of System of Systems

Zhang Xiuwei, He Keqing, Wang Jian, Wang Chong

(State Key Lab of Software Engineering, Wuhan University, Wuhan 430072, China)

Abstract: With the development of internet of things, cloud computing and software as a service as well as other new technologies, the distributed software system is becoming more and more complex because of interconnection and interoperability. The software system performance index was assessed and predicted from the perspective of system of systems, which is now a new research field. Firstly, the unique characteristics of the SoS were extracted from different definitions of SoS and an overall assessment framework was established in terms of their internal relationship. Finally, a practical SaaS system was took as a case study to validate the method.

Key words: system of systems, SOA, software as a service, measure of effectiveness, negative emergence

1 引言

近年来,随着计算机通信、智能物联、分布式控制、云计算等信息领域新技术的蓬勃发展和普及应用,原本各自独立的系统个体通过“信息”搭建的“纽带”,将彼此联结成为一个由“独立系统”作为组分构成的大系统,形成了一个关于“系统的系统(system of systems, SoS)”的存在形态^[1]。

以“深入感知、广泛互联和高度智能”为特征的传感网、云计算、软件服务化等新技术的发展,在强调深入感知和互联互通互操作的过程中,导致系统规模越来越大,系统关联性也越来越复杂,影响系统可靠性和效能的不确定性因素也越来越多。在 SOA 和云计算环境下,大型分布式软件系统的应用功能主要通过集成方式实现,而这些功能的组成部分通常是由自治系统提供的,通过集成自治系统的功

^{*} 国家重点基础研究发展计划基金资助项目 (No.2014CB340401, No.2014CB340404), 国家自然科学基金资助项目 (No.61373037, No.61202031, No.61100017), 国家科技支撑计划基金资助项目 (No.2012BAH07B01), 国家云计算示范工程“中小企业管理云应用研发与产业化”项目

能完成其新的应用目标。因此,无论是在大型软件规划开发阶段还是在运行维护阶段,都需要对体系特性进行描述和界定,以确保其组成成分以及构成的体系能够满足目标的要求^[2]。

system of systems 又称为“系统集”或者“元系统(meta-system)”,文中称为“体系”。对于体系而言,其组分系统虽然分布广泛、形式松散、独立工作,但从系统能力来看却更像一个有机整体。信息技术发展在体系中的应用提升了内在组分系统间传递信息的速度和共享信息的程度,增强了独立系统对环境的感知与响应能力,使得多个系统间完全可以通过自主协调,以自同步的方式相互配合,完成个体或单一平台难以完成的任务,缩短了组分系统间原本遥远的物理部署距离和相对较长的交互等待时间,达到了功能上“1+1>2”的整体效果。可以说,云计算和服务计算等新兴技术的发展逐步消除了软件系统孤立存在的形态和独立工作的方式,使分散的组分系统可以实现相互协同、配合工作,使个体聚合成为整体,产生效能更高的体系能力,在这个被压缩的时空纬度下,整体功能的形成使它更像一个具有强大功能的有机整体^[3]。体系的组织管理过程不能按照传统的系统工程方法解决,主要表现在不能进行“自顶向下”的预先设计、不以最优化为目标和负涌现的不确定性。

软件即服务(software as a service, SaaS)作为一种新型的软件交付模式得到越来越广泛的认可,并将成为未来软件服务的重要运营方式。SOA 作为一种分布式体系的体系结构,可以将异构平台上应用程序的不同服务(或称为组件)进行整合,通过这些服务之间定义良好的接口和规范以松耦合方式整合在一起,即将多个已有的服务通过网络将其整合成一个新系统。SOA 的松散耦合和可重用性为 SaaS 系统的开发提供了技术支撑,成为 SaaS 系统开发的核心技术。Sinatra F 将 SaaS 与 SOA 形象地比喻为婚姻与爱情的关系,足以说明二者关系的紧密。关于 SOA 有众多定义, Roy Schulte of Gartner 认为 SOA 必须满足以下几点要求:

- 系统的服务组件可以被增加、替换和修改,且不影响系统运行;
- 服务组件可以分布式运行;
- 服务组件接口必须用标准元数据定义,并能被系统开发者发现;
- 服务组件可以被其他具有相同接口的组件替换;
- 服务组件可以被一个或多个系统使用多次。

可以看出, SOA 架构正好满足了体系的特性要求。因此可以将一个 SaaS 系统作为一个基于 Web 服务组件的体系进行分析研究。首先抽取体系定义中 10 个特征词汇,并依据特征词汇间的依赖关系建立 SaaS 系统评估框架,并从效能、可靠性等多个方面进行分析,最后通过 SaaS 系统 CloudCRM 进行验证。

2 体系定义

尽管目前学术界对体系研究多年,但大多数研究仍集中于国防和军事系统的集成及数据融合过程方面,近年来也开始逐步向企业和政府应用领域扩展。在研究过程中,对体系的概念也存在很多分歧,不同领域的专家学者、实践人员都有不同的认识和理解。为了有效地对其特性进行抽取,在参考文献[4]的基础上进一步完善,对体系的概念进行归纳总结,从中外体系工程专家对体系的定义词汇中抽取其独有特性词汇来描述体系。

- Kotov(1997 年)定义^[5]:体系是大规模并发和分布式系统,它的组件本身也是复杂系统。
- Sage(2001 年)定义^[6]:引起体系问题的现代系统不是单独的,它有 5 个共同特征,分别为个体系统的运行独立性、个体系统的管理独立性、地理分布、涌现行为和演化开发。
- DeLaurentis(2004 年)定义^[7]:体系是一个跨领域异构系统网络集合,如果这些系统之间的交互被独立建模,它们展现的运营和管理独立性、物理分布、涌现和演化行为都不会很明显。
- 美国空军(2005 年)定义^[8]:体系是一个可配置系统,它的组件系统可以增加和删除,每个组件都能以自己的方式提供有用的服务,且接受其他服务的管理,然而它们组合在一起能够表现出协同、超越能力。
- Kuras(2005 年)定义^[9]:为实现个体系统无法达到的目标而集成的系统集合,这些体系中的个体系统通常是为自身目的独立开发,这些组件系统物理结构相异,地理位置不同,通常它的边界脆弱而稳定,系统间通过良好的接口连接,如果其中某个系统发生变化或故障,体系仍可正常工作,但是它提供的整体性能会受影响。
- 张最良研究员(2005 年)定义^[10]:体系是能得到进一步“涌现”性质的关联或联结的独立系统集合。



- Northrop(2006 年)定义^[11]:体系是由一些独立自治系统为满足特定的需求而组合的系统。
- Sheard(2006 年)定义^[12]:体系是一个复杂而有意义的整体;由复杂的独立组件构成,较强的互操作能力使它可以加入不同的配置甚至不同的体系中;由情景复杂度标识,且严重影响了它的行为,使其难以理解;具有模糊或变化的边界,且具有涌现特性。
- 美国国防部(2008 年)定义^[13]:体系被定义为一个系统集,使得独立和实用的系统被集成为一个更大的系统并产生独有的能力。
- White(2008 年)定义^[14]:体系是一个系统集成,可以实现单个系统无法实现的目标,每个系统可以单独管理和运行,可以是地理分布,显出演化开发和涌现行为的特性。
- 胡晓峰教授(2008 年)定义^[15]:系统和体系是两个有时可以互换的概念,它们既有相同之处,也有不同之处,体系是系统的组成,体系内部各系统一般是合作关系,体系具有整体性、信息相关性、耦合松散性和开放性等特点。
- Firesmith (2008 年)定义^[16]:体系由一组相互依赖和交互的组件拼装集成而来,它提供的行为和特性能够满足涉众用户的需求和想法,但这些行为和特性是无法通过单独组件完成的。

- 美国国土安全部(2009 年)定义^[17]:当一组由人、技术和组织机构组成的独立运行的系统连接之后,可以实现紧急应对日常运作、计划事件和重大事故。
- INCOSE(2009 年)定义^[18]:体系又称为元系统,是为达到预期结果而集成的复杂系统,该元系统由多个自治嵌入式复杂系统组成,而这些系统在技术、环境、运行、地理位置和概念框架上有差异。
- Purdue University(2009 年)定义^[19]:体系是为实现一个共同需求,许多独立自治的系统被规模化集成。
- Wikipedia(2009 年)定义^[20]:体系是一个面向任务的专用系统集成,把它们的资源和能力结合在一起得到一个新的更复杂的元系统,元系统与这些系统简单的叠加相比,能提供更多的功能和性能。

3 体系特征抽取与定义

从体系的不同定义词汇中抽取了用来刻画和描述体系独有特性的 10 个词汇:复杂性、演化性、规模、可变性、负涌现、自治性、治理性、异构性、物理分布性和重用性。表 1 列出了上述各种定义对这些特征词汇的涵盖范围。以上 10 类特性作为体系的基本定义词汇对其进行了有效的刻画。因此,分析体系可以通过对 10 类特性进行定性和定量分析来获取体系的整体效能。

定义 1 (重用性)组分系统在体系中的重用程度的度

表 1 不同定义中抽取的体系特性

定义者	特性									
	复杂性	演化性	规模	可变性	负涌现	自治性	治理性	异构性	物理分布性	重用性
美国国防部			✓		✓	✓	✓			✓
国土安全部						✓	✓	✓		✓
美国空军		✓			✓		✓			
普渡大学			✓		✓		✓			
Wikipedia	✓				✓					
INCOSE	✓	✓			✓	✓	✓	✓	✓	
Sage		✓			✓	✓	✓		✓	
Northrop		✓				✓	✓		✓	
White					✓	✓	✓		✓	
Kuras					✓	✓			✓	
Sheard	✓	✓		✓	✓	✓				
Firesmith					✓	✓	✓			✓
Kotov	✓		✓						✓	
Delairemtos		✓			✓	✓	✓	✓	✓	✓
胡晓峰			✓			✓	✓			
张最良					✓	✓				

量,表示为 $R=\{r_i|i \in (1,2,3,4,5)\}$, r_1 表示组分系统完全不可重用, r_2 表示组分系统低可重用, r_3 表示组分系统组件可重用, r_4 表示组分系统单元可重用, r_5 是完全可重用即 100%可重用。

定义 2 (异构性) 组分系统之间的差异程度的度量,表示为 $H=\{h_i|i \in (1,2,3)\}$, h_1 表示高同构系统, h_2 表示中度异构系统,例如硬件和设备差异, h_3 表示高度异构系统,例如应用领域差异(如金融系统与物流系统)。

定义 3 (规模) 系统在合适度量维度中的数额,如组分系统数量、软件代码量、物理维度等,表示为 $S=\{s_i|i \in (1,2,3,4)\}$, s_1 表示小型系统(如电视机), s_2 表示中型系统(如轿车), s_3 表示大型系统(如航母舰队), s_4 表示超大规模系统(如国家电网)。

定义 4 (治理性) 子系统在独立、分散和无协作方式中的治理程度的度量,表示为 $G=\{g_i|i \in (1,2,3,4)\}$, g_1 表示直管系统即完全受控于中心系统控制管理, g_2 表示协管系统即拥有自己的管理体系,但与中心系统目标一致, g_3 表示协作系统即互相自愿协作和共享信息, g_4 是虚拟系统,表示完全独立自治无协作系统。

定义 5 (自治性) 组分系统在独立性、自主性及自足性方面的度量,表示为 $A=\{a_i|i \in (1,2,3)\}$, a_1 表示低自治系统(万维网), a_2 表示中度自治系统(如电视机), a_3 表示高度自治系统(如 SOA 系统)。

定义 6 (演化性) 系统目标和需求随着时间不断变化程度的度量,表示为 $E=\{e_i|i \in (1,2,3,4,5,6)\}$, e_1 表示静态系统即很少更新和维护(如电视机), e_2 表示超低演化率的系统(如卫星), e_3 表示低演化率的系统(如核电站), e_4 表示中等演化率的系统(如汽车), e_5 表示高演化率的系统(如智能手机), e_6 表示超高演化率的系统(如万维网)。

定义 7 (可变性) 系统某一方面同时存在多个变量、版本和配置的程度,表示为 $V=\{v_i|i \in (1,2,3,4,5)\}$, v_1 表示不可变系统或无配置系统, v_2 表示低可变系统(如核电站), v_3 表示中等可变系统(如汽车),同时有豪华版配置和经济实用版配置, v_4 表示高可变系统(如电视机), v_5 表示超高可变系统(如局域网)。

定义 8 (物理分布性) 子系统的地理位置差异程度的度量,表示为 $P=\{p_i|i \in (1,2,3,4,5,6,7)\}$, p_1 表示邻接系统, p_2 表示建筑群分布系统即位于相同建筑群内, p_3 表示城市群系统即位于同一个城市, p_4 表示区域分布群系统即位于相同地区或省份, p_5 表示国家分布群系统即系统位于

某一个国家, p_6 表示全球分布系统, p_7 表示超级分布系统,涵盖了太空范围,如 GPS。

定义 9 (复杂性) 涉众用户对系统的理解、分析和使用的难易程度的度量,表示为 $C=\{c_i|i \in (1,2,3,4)\}$, c_1 表示低复杂系统即简单系统(如 ATM 机), c_2 表示中度复杂系统(如电视机), c_3 表示高度复杂系统(如战斗机), c_4 表示超复杂系统。

定义 10 (负涌现) 由于系统集成而产生的不确定和难于预测的有害的系统行为的度量,表示为 $N=\{n_i|i \in (1,2,3,4)\}$, n_1 表示低负涌现系统, n_2 表示中度负涌现系统, n_3 表示高负涌现系统, n_4 表示无法接受的负涌现系统。

4 SaaS 系统评估框架

SaaS 系统的效能(或性能)度量是对 SaaS 系统内部具有的特征和能力的描述。然而 SaaS 系统的效能往往依赖于其他的特性,如可靠性、演化性、可变性、复杂性及负涌现等。SaaS 系统效能评估是 SaaS 系统设计的重要任务之一, SaaS 系统效能的预测是根据系统中包含的服务的特性推测其整体特性,这是一个由局部到整体、由小到大的典型的综合过程。由于体系特性涉及面很广,在概念的定义上也不尽相同,重点给出了对复杂性、负涌现、可靠性和体系效能 4 个特性的评估方法。利用体系的各类特性及其之间的关联,定性地评估 SaaS 系统的效能,并通过问卷和专家咨询的方式综合分析,建立了体系特性间的依赖关系,形成了基于体系特性的评估框架,具体如图 1 所示,例如 SaaS 系统对 Web 服务的重用会导致系统间异构性增加, SaaS 系统规模增大和治理的松散和自治性的减弱,都会导致

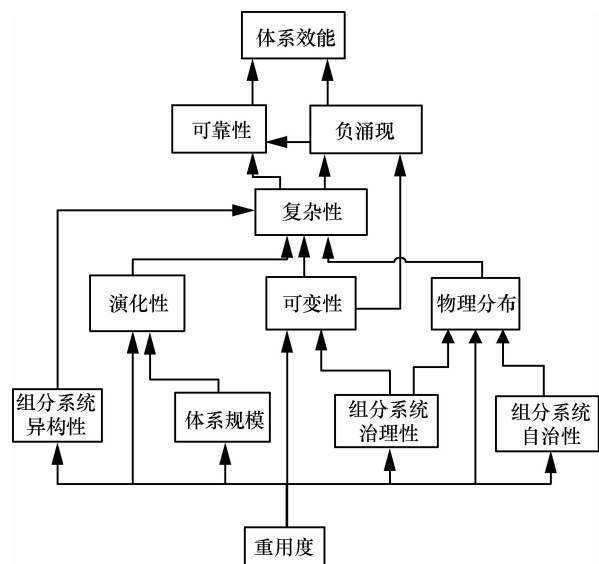


图 1 SaaS 系统特性评估框架



系统复杂性增加,还会导致系统负涌现的增加以及可靠性下降。在进行分析之前首先给出 SaaS 系统的形式化定义。

定义 11 SaaS 系统定义为一个五元组 $SaaS=(Ss,I,O,D,F)$,含义如下:

- Ss 是 SaaS 系统包含服务组件的有限集;
- I 是 SaaS 系统的输入接口集;
- O 是 SaaS 系统的输出接口集;
- D 是 SaaS 系统中服务组件之间的依赖和交互关系, $D \in \{\hat{a}, \oplus, \wedge, \vee, \neg\}$, 其中 $\hat{a}$ 表示服务组件并联, $\oplus$ 表示两个服务组件串联, $\wedge$ 表示服务组件之间有交互, $\vee$ 表示服务组件可以相互替换, $\neg$ 表示服务组件在 SaaS 系统中失效或者不发挥作用;
- F 是 SaaS 系统的涌现行为集, $F=F^+ \cup F^-$, F^+ 是正涌现行为集,也是系统集成的根本目的和目标,是指系统性能的跃升和提高; F^- 是负涌现行为集,系统集成产生的有害且无法预测的行为,可能造成 SaaS 系统的坍塌。

4.1 SaaS 系统复杂性

SaaS 系统的复杂性源于要素与服务组件之间、服务组件与服务组件之间或者服务组件系统与体系之间呈现的不确定性关系。结合 SaaS 系统的特点以及复杂性定义,其复杂性主要取决于各服务组件系统的复杂性构成因素的交互作用,其中每一项的复杂性构成因素又由若干项复杂性因子构成,SaaS 系统的复杂性因素包括以下几项^[21]:

- 实现技术复杂性 C_T ;
- 组织架构复杂性 C_O ;
- 业务内容复杂性 C_C ;
- 信息互操作复杂性 C_I ;
- 实现目标复杂性 C_G ;
- 运行环境复杂性 C_E 。

将 SaaS 系统复杂性表示为 μ 函数,即 $C=\mu(C_T, C_O, C_C, C_I, C_G, C_E)$, $i \in \{1, 2, \dots, k\}$, k 表示 SaaS 系统包含 k 个服务组件系统。

4.2 SaaS 系统负涌现

SaaS 系统集成的过程虽然带来正涌现行为,但同时也带来了一些不确定的负涌现行为。SaaS 系统的正涌现是指系统的性质发生了相变,产生出新的性质。从体系角度来说,就是微观行为通过交互产生出的宏观模式,也就是“体系的形成”。但涌现的效果并不一定都是效能跃升或正涌现,也可能是效能坍塌(即负涌现)。负涌现根据其预测的

难易程度可分为 3 类,具体如下:

- 不利涌现 $N_D(\text{detrimental emergence})$,通过适当的措施可以避免的有害涌现;
- 非预期涌现 $N_U(\text{unintended emergence})$,源于服务组件系统的行为和特性,难以预测;
- 无法预测行为 $N_H(\text{hard-predictive emergence})$,这种行为在 SaaS 系统的需求工程、架构工程和设计工程过程中都完全无法预测,仅在体系运行中才出现。

因此,可将负涌现表示为 η 函数, $N=\eta(N_D(c,v), N_U(c,v), N_H(c,v))$, η 函数是负涌现函数,包括 3 类负涌现 N_D 、 N_U 、 N_H ,参数 c 、 v 分别是体系的复杂性和可变性。

4.3 SaaS 系统可靠性

可靠性预测是根据 SaaS 系统各个服务组件系统的可靠性来推测 SaaS 系统整体的可靠性。可靠性是指在规定条件下、规定时间内,系统不发生失效的概率,是衡量 SaaS 系统质量好坏的最重要指标之一^[21]。可靠性 RE 通常以概率的方式表示,即 $RE \in [0, 1]$ 。在传统系统可靠性预测中,经典的方法有性能参数法、相似产品法、元件计数法和门特卡罗法。现代的方法有上下限法、可靠性精确评估等^[22]。利用串并联的方法对可靠性进行评估。

定义 13 (服务组件串联) SaaS 系统包含的所有服务为达到预期涌现时连接在一起,当任意一个服务失效时,整个系统无法完成目标,则称该 SaaS 系统为服务组件串联,用 $\oplus$ 表示,如 $A \oplus B$ 表示服务 A 与服务 B 串联,由 K 个独立自治的组分系统 $A, B, \dots, K$ 组成的串联体系的可靠性 $RE(A \oplus B \oplus \dots \oplus K)$ 如下:

$$RE(A \oplus B \oplus \dots \oplus K) = \prod_{i=A}^K RE_i, i \in (A, B, \dots, K) \quad (1)$$

RE_i 是第 i 个服务运行的可靠性概率。

定义 14 (服务组件并联) 系统中的服务为达到预期涌现时连接在一起,当任意一个服务失效时,整个 SaaS 系统仍可完成目标,仅仅是任务完成方式和完成效能降低,则称服务组件并联,用 $\hat{a}$ 表示, $A \hat{a} B$ 表示服务 A 与服务 B 并联,由 K 个独立工作的组分系统组成的并联体系的可靠性 $RE(A \hat{a} B \hat{a} \dots \hat{a} K)$ 如下:

$$RE(A \hat{a} B \hat{a} \dots \hat{a} K) = 1 - \prod_{i=A}^K (1 - RE_i), i \in (A, B, \dots, K) \quad (2)$$

RE_i 是第 i ($i=A, B, \dots, K$) 个组分系统可靠运行的概率。

定理 1 设 SaaS 系统由 k 个独立的服务并联而成,第 i 个服务又由 m_i 个原子服务组成,设第 i 个服务的第 j 个

服务的可靠性为 $RE_{ij}, \begin{pmatrix} i=1,2,3,\dots,k \\ j=1,2,\dots,m_i \end{pmatrix}$, 于是第 i 个服务的

可靠性 $RE_i = \prod_{j=1}^k RE_{ij}$, 则系统的可靠性表示为:

$$RE = 1 - \prod_{i=1}^k (1 - RE_i) = 1 - \prod_{i=1}^k (1 - \prod_{j=1}^{m_i} RE_{ij}) \quad (3)$$

定理 2 设 SaaS 系统由 k 个独立的服务组成, 第 i 个服务由 m_i 个独立的原子服务并联而成, 设第 i 个服务的第 j 个原子服务的可靠性为 $RE_{ij}, (i=1,2,\dots,k, j=1,2,\dots,m_i)$,

则第 i 个组分系统的可靠性 $RE_i = 1 - \prod_{j=1}^{m_i} (1 - RE_{ij})$, 则系统的可靠性表示为:

$$RE = \prod_{i=1}^k RE_i = \prod_{i=1}^k (1 - \prod_{j=1}^{m_i} (1 - RE_{ij})) \quad (4)$$

SaaS 系统可分为先串联后并联和先并联后串联, 又因为系统中的服务满足相互独立的假设, 定理 1 和定理 2 就可以用来计算任意方式组合的 SaaS 系统可靠性, 篇幅有限, 定理 1 和定理 2 的证明过程略。

4.4 SaaS 系统效能

SaaS 系统效能评估就是对 SaaS 系统进行综合分析, 把服务的各项性能与任务要求综合比较, 最终得到表示系统优劣程度的结果, 也是系统预期能达到一组特定任务要求程度的度量, 一般用于描述体系完成其任务的总体能力。这种效能定量尺度的效能指标称为系统效能度量 (measures of effectiveness, MoE)。

通过分析各个 SaaS 系统中服务满足任务要求的程度, 根据图 1 评估框架中影响体系效能的因素, 建立体系效能评估模型。其中, 重用性 R 、异构性 H 、规模 S 、治理性 G 、自治性 A 均为影响效能的底层指标, 演化性 E 、可变性 V 、物理分布 P 为影响效能的二级指标, 复杂性 C 、负涌现 N 、可靠性 RE 为一级指标, 首先将底层指标值转化为二级指标相应的映射函数值, 最后将一级指标映射到效能, 函数取值区间是 $[0, 1]$ 的映射, 它将所有的影响指标函数值转化为 $0 \sim 1$ 的一个数值, 该指标是人们对效能的一个主观度量。二级指标值由底层指标决定, 根据效能评估模型, 得到如下映射函数:

$$\begin{aligned} E_i &= \sum_{i=1}^k w_i \sigma(R_i, G_i) \\ V_i &= \sum_{i=1}^k w_i \xi(R_i, G_i) \\ P_i &= \sum_{i=1}^k w_i \rho(G_i, R_i, A_i) \end{aligned} \quad (5)$$

其中, i 为服务组件系统的标号, k 为服务组件系统的数量, w_i 为指标的权值, $\sigma(R_i, G_i)$ 为演化性的映射函数, $\xi(R_i, G_i)$ 为可变性的映射函数, $\rho(G_i, R_i, A_i)$ 为物理分布映射函数, 其中 R_i, G_i, A_i 为一级指标变量。一级指标由二级指标决定, 即:

$$\begin{aligned} C_i &= \sum_{i=1}^k (\lambda_i \varphi(G_i, R_i, A_i) + \mu(C_{Ti}, C_{O_i}, C_{C_i}, C_{L_i}, C_{G_i}, C_{E_i})) \\ N_i &= \sum_{i=1}^k \lambda_i \eta(N_D(c, v), N_U(c, v), N_H(c, v)) \\ RE_i &= \sum_{i=1}^k \lambda_i \delta(N_i, C_i) \end{aligned} \quad (6)$$

其中, i 为服务组件系统的标号, k 为服务组件的数量, λ_i 为指标的权值, $\sigma(R_i, G_i)$ 为演化性的映射函数, $\varphi(G_i, R_i, A_i)$ 和 $\mu(C_{Ti}, C_{O_i}, C_{C_i}, C_{L_i}, C_{G_i}, C_{E_i})$ 为复杂性的映射函数, $\delta(N_i, C_i)$ 为可靠性映射函数, 括号内为对应的变量复用性 N_i 和复杂性 C_i 。得到一级评估指标后, 可得到 SaaS 系统的效能与一级指标映射函数为:

$$MoE = \prod_{i=1}^k (\alpha_i RE(i) + \beta_i N(i) + \gamma_i C(i)) \quad (7)$$

其中, i 表示体系中服务组件系统的标号, k 为服务组件系统的数量, $\alpha_i, \beta_i, \gamma_i$ 分别为可靠性、负涌现和复杂性的权重指数, 为了体现“木桶原理”的短板效应, 效能采用连乘的方式。

5 案例分析

以 SaaS 系统 CloudCRM 为例进行分析, CloudCRM 是一款面向多租户、以客户管理为主的 SaaS 平台, 目的是在企业管理人员、销售人员之间建立一个立体的管理系统, 帮助企业对内、对外实现以“客户为中心”的全方位管理^[24]。CloudCRM 采用 SOA 架构, 通过集成的方式将多个服务组件进行组合集成。系统的服务组件的构成见表 2,

表 2 CloudCRM 的服务组件构成

服务提供者	服务类型	数量	占比
SugarCRM	WSDL	15	27.3%
Yahoo	API	1	1.82%
Google	API	3	5.46%
Kingdee	API	1	1.82%
Baidu	API	1	1.82%
China Weather	API	1	1.82%
Sina SMS	API	1	1.82%
ICTCLAS	components	1	1.82%
SKLSE	WSDL/components	41	74.6%



涵盖了9个服务提供者,其中SKLSE为CloudCRM平台的开发者。

目前该系统提供的不仅是客户关系管理服务,还包括Google地理位置查询服务、RFID服务、百度地图服务、快递查询服务和天气查询服务等,构成了一个以CRM为核心的多功能SaaS平台系统。图2是系统的运行界面。

将CloudCRM间的服务组件连接交互关系进行抽象,如图4所示,图中仅对抽取组合后的服务进行描述。系统由多个并联子系统串联而成,客户关系子系统a包含14个服务组件,快递100子系统b由54个服务组件并联组成,气象服务c包含3个可替换的服务组件,RFID服务d是一个独立的服务组件,地图服务系统d包含2个可替换服

务组件,图形化显示服务与其他服务都有交互。

CloudCRM系统可抽象为6个核心服务组件,服务组件间的交互组合关系如图3所示。通过对6个核心软件服务组件的调用历史统计得到各服务可靠性,见表3,根据定义1和定理2,可得该SaaS系统的可靠性为0.96,系统测试的实际运行可靠性为0.971。二者结果相比稍有偏差。

首先从系统规模分析,SaaS系统CloudCRM包含多个公司提供的服务以及服务器等硬件设施,属于中小规模,

表3 CloudCRM系统可靠性

组件系统	a	b	c	d	e	f	预测值	测试值
可靠性	0.99	0.99	0.97	0.98	0.92	0.98	0.966	0.971



(a)



(b)

图2 CloudCRM系统平台

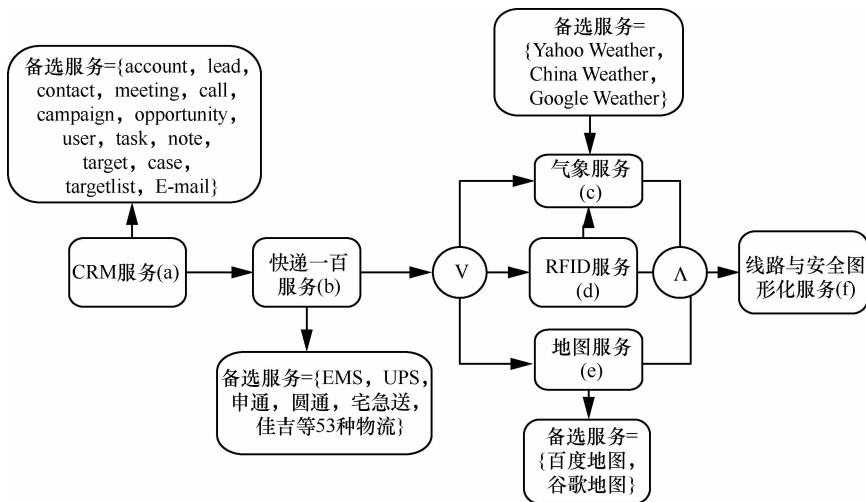


图3 CloudCRM系统服务组件构成

则 $S=s_2$; 它属于 SOA 系统, 属于高度自治, 则 $A=a_3$; 可变性分析, 该系统是面向多租户的可进行个性化定制的系统, 具有较高可变性, 则 $V=v_4$; 异构性方面, SOA 软件服务以及 API 等具有良好的接口, 异构性较小, 则 $H=h_2$; 演化性方面分析, 可依据用户需求增加新的服务内容, 演化性高, 则 $E=e_5$; 在物理分布中采用 SOA 的方式, 包含多个服务不在同一区域, 则分布为 $P=p_5$; 由于系统采用 SOA, 服务重用度很高, 即重用性 $R=r_5$; 在负涌现方面, 在 CRM 服务实现多租户过程中, 系统的处理时延增加, 处理速度下降, 总体来看, 该系统产生的负涌现低, 属于低负涌现系统, 即 $N=n_1$; 由于该应用设计的应用范围相对较窄, 则系统复杂性 $C=c_2$ 。综合各类特性, 将 CloudCRM 的 10 类特性以蜘蛛网图的形式进行定性表示, 如图 4 所示。效能定量评估中权重的取值在不同规模系统中的影响值不同, 但确实是一个复杂的过程, 只尝试进行一些初步探讨, 对具体效能指标和权值获取以及效能值的合并方法需要进一步研究。

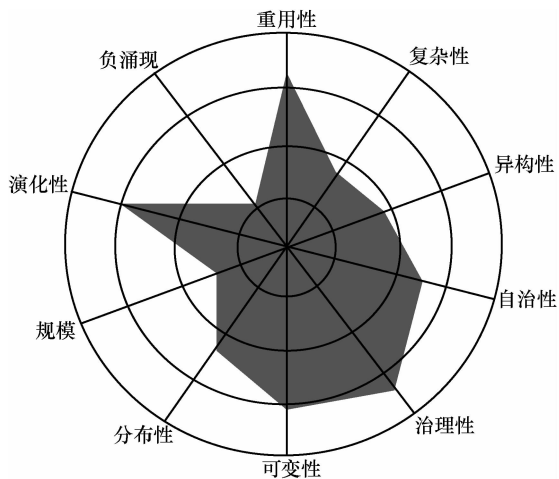


图 4 CloudCRM 系统评估蜘蛛网

6 结束语

从体系的定义词汇中抽取用于刻画体系特性的词汇, 并建立特性评估框架, 最后从体系的角度对 SaaS 软件系统进行分析。因此, 使用体系的理论与方法分析大型软件系统将成为新的发展方向, 对大型软件系统的规划、应用和部署有重要的指导意义。从体系的视角研究软件系统仍然处于初期, SaaS 软件系统为了能有效满足系统业务目标要求, 必须达到按需服务和不断演化的要求, 因此, 下一步的研究重点是结合 SaaS 系统的特点展开细致研究, 尤其是 SaaS 系统效能评估, 包括效能指标和权值获取以及

效能值的合并方法。SaaS 系统可靠性预测方法在系统非常复杂时无法达到满意效果, 尤其是并系统交互增加时, 需要探索新的可靠性预测方法。此外, 需要结合更多的 SaaS 实例进一步分析和验证该方法的有效性。

参考文献

- 1 胡晓峰, 张斌. 体系复杂性与体系工程. 中国电子科学院学报, 2011,6(5):446~450
- 2 卢捍华, 张顺颐, 王亚石等. 系统体系方法与软件系统功能形式分解. 通信学报, 2010,31(12):131~139
- 3 张维明, 修保新. 体系复杂性与体系工程. 中国电子科学院学报, 2011,6(5):452~456
- 4 Firesmith D. Profiling systems using the defining characteristics of systems of systems. <http://www.sei.cmu.edu/reports/10tn001.pdf>, 2011
- 5 Kotov V. Systems-of-systems as communicating structures. <http://www.hpl.hp.com/techreports/97/HPL-97-124.html>, 1997
- 6 Sage A P, Cuppan C D. On the systems engineering and management of systems of systems and federations of systems. Information, Knowledge Systems Management, 2001,2(4):325~345
- 7 Daniel D, Robert K. A system of systems perspective for future public policy. Review of Policy Research, 2004, 21(6):829~837
- 8 United States Air Force. Scientific advisory board report on system-of-systems engineering for Air Force capability development. <https://acc.dau.mil/GetAttachment.aspx?id=150224&pname=file&aid=28788>, 2012
- 9 Kuras M L, White B E. Engineering enterprises using complex system engineering. Proceedings of the First Annual System of Systems (SoS) Engineering Conference, Johnstown, June 2005
- 10 张最良, 李露阳. 体系开发规律和科学途径. 中国科技论坛, 2006(2):106~110
- 11 Northrop L. Ultra-large-scale systems: the software challenge of the future. <http://www.sei.cmu.edu/ulsl/>, 2012
- 12 Sheard S A. Systems of systems necessitates bridging systems engineering and complex systems science. Proceedings of the Second Annual System of Systems Engineering Center of Excellence (SOSECE) Conference, Los Angeles, July 2006
- 13 Office of the Deputy Under Secretary of Defense for Acquisition and Technology. Systems and Software Engineering ODUSD (A&T) SSE. System Engineering Guide for Systems of Systems, Washington DC, USA, August 2008
- 14 White B E. Complex adaptive systems engineering (CASE). Proceedings of the Third System of Systems Conference National Institute of Standards and Technology, Gaithersburg, December 2008
- 15 胡晓峰. 战争复杂性与信息化战争模拟. 系统仿真学报, 2006,8(12):3572~3580



- 16 Firesmith D, Donald G. The method framework for engineering system architectures (MFESA). Proceedings of the Seventh International Conference on Composition-Based Software Systems, Washington DC, USA, 2008
- 17 Department of Homeland Security. The system of systems approach for interoperable communications. http://www.safecomprogram.gov/NR/rdonlyres/FD22B528-18B7-4CB1-AF49-F9626C608290/0/SOSApproachforInteroperableCommunications_02.pdf, 2012
- 18 National Center for Systems of Systems Engineering. What is systems of systems engineering. http://www.eng.odu.edu/ncsose/What_is_SOSE.shtml, 2011
- 19 Purdue University, College of Engineering. System of system (SoS). <https://engineering.purdue.edu/Engr/Research/Initiatives/SoS/>, 2012
- 20 Wikipedia. System of systems. http://en.wikipedia.org/wiki/System_of_systems, 2012
- 21 李慧, 杨乃定, 郭晓. 复杂项目系统复杂性构成研究. 软科学, 2009, 23(2): 75~79
- 22 楼俊钢, 江建慧, 靳昂. 考虑软件不同失效过程偏差的软件可靠性模型. 计算机学报, 2010, 33(7): 1264~1271
- 23 张国春, 胡晓峰. 体系对抗仿真中体系效能分析初探. 系统仿真学报, 2003, 15(12): 1698~1701
- 24 张秀伟, 何克清, 王健等. SaaS 服务超市的构建模型和服务推荐方法. 通信学报, 2011, 32(9A): 158~165
- 25 游光荣, 张英朝. 关于体系与体系工程的若干认识和思考. 军事运筹与系统工程, 2010, 24(2): 14~21

军事运筹与系统工程, 2010, 24(2): 14~21

[作者简介]



张秀伟, 男, 武汉大学博士研究生, 主要研究方向为服务计算、云计算、电子商务。



何克清, 男, 武汉大学教授、博士生导师, 主要研究方向为需求工程、ISO 互操作标准、语义互操作技术、服务计算。

王健, 男, 武汉大学讲师, 主要研究方向为需求工程、Web 服务发现、语义互操作技术。

王翀, 女, 武汉大学讲师, 主要研究方向为语义互操作技术、流程管理。

(收稿日期: 2013-07-30)

(上接第 91 页)

- 7 Yu J, Buyya R, Chen K T. Cost-based scheduling of scientific workflow applications on utility grids. Proceedings of International Conference on e-Science and Grid Computing, Melbourne, 2005: 8~147
- 8 Liu K, Jin H, Chen J, *et al.* A compromised-time-cost scheduling algorithm in SwinDeW-C for instance-intensive cost-constrained workflows on a cloud computing platform. International Journal of High Performance Computing Applications, 2010, 24 (4): 445~456
- 9 李文浩, 王海洋. 事务密集型环境下的工作流自适应调度算法. 计算机工程, 2010, 36(19): 84~86
- 10 晏婧, 吴开贵. 适用于实例密集型云工作流的调度算法. 计算机应用, 2010(11): 2864~2866
- 11 Chiu D K W, Li Q, Karlapalem K. A meta modeling approach to workflow management systems supporting exception handling. Information Systems, 1999, 24(2): 159~184
- 12 罗海滨, 范玉顺, 吴澄. 工作流技术综述. 软件学报, 2000, 11(7): 899~907

[作者简介]



刘伟, 女, 广西大学计算机与电子信息学院硕士研究生, 主要研究方向为云计算、工作流技术。



李陶深, 男, 博士, 广西大学计算机与电子信息学院教授, 主要研究方向为云计算、分布式数据库、无线 mesh 网络、网络与信息安全。

黄汝维, 女, 博士, 广西大学计算机与电子信息学院副教授, 主要研究方向为云计算、信息安全。

(收稿日期: 2013-08-20)