

工业复杂软件系统的成长性构造框架与实现

于景洋^{1,2}, 宁德军², 毛建华¹

(1. 上海大学 通信与信息工程学院, 上海 200444; 2. 中国科学院 上海高等研究院, 上海 201210)

摘要: 工业领域中智能制造和智慧城市的快速发展, 促使各种具有边界开放、持续演化、自适应性等特征的复杂软件系统的出现; 基于还原论和单一组织的传统软件开发方法不再适合发展需求, 也不能满足跨组织的社会化协作和资源共享等基本需求; 提出一种基于容器云和跨组织资源共享的复杂软件系统成长性构造框架, 通过容器技术、多租户技术、信息资源目录技术实现社会化协作开发和复杂软件系统成长性构造, 并且该框架在一个工业研究案例中的应用实践证明了其在复杂软件系统的快速构建和演化方面具有重要意义。

关键词: 智慧园区; 复杂软件系统; 容器技术; 海云分形架构

Growing Construction Framework and Implementation of Industrial Complex Software Systems

Yu Jingyang^{1,2}, Ning Dejun², Mao Jianhua¹

(1. School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China;

2. Shanghai Advanced Research Institute, Chinese Academy of Sciences, Shanghai 201210, China)

Abstract: The rapid development of intelligent manufacturing and smart cities in the industrial field has led to the emergence of various complex software systems with features such as open borders, continuous evolution, and adaptability. Traditional software development methods based on reductionism and a single organization are no longer suitable for development needs, nor can they meet the basic needs of social collaboration and resource sharing across organizations. This paper proposes a complex software system growing construction framework based on container cloud and cross-organization resource sharing. Through container technology, multi-tenant technology and information resource directory technology, social collaborative development and complex software system growth construction are realized. And the application of the framework in an industrial research case proves its importance in the rapid construction and evolution of complex software systems.

Keywords: smart park; complex software system; container technology; sea-cloud fractal architecture

0 引言

随着信息技术的快速发展以及各行业智能化的变革, 催生出工业领域各种大型的复杂软件系统。需求的频繁变更和应用的快速部署使得传统软件工程方法越来越难以适应复杂软件系统的开发与维护。敏捷开发方法通过使用快速迭代、持续集成和过程自动化技术等方法, 缩短了客户反馈周期, 从一定程度上缓解了复杂软件系统 (complex software systems, CSS) 开发和面临的挑战, 但在复杂软件系统的成长性构造和适应性演化方面, 仍然缺少必要的技术和方法。复杂软件系统的成员异构性使得系统本身封闭性较强, 各系统之间协同困难, 其复杂性、重量级等固有特性导致系统无法快速构建和复制。

复杂软件系统^[1]是指由大量局部自治软件系统持续集成、相互耦合关联而成的大型软件系统。系统要素之间的耦合交互关系动态变化且日趋复杂, 整个系统的行为难以通过各自治软件系统特征的简单叠加加以刻画。复杂软件

系统呈现以下典型特征:

1) 边界开放性: 复杂软件系统是在人一机一物三元融合世界, 通过各种主体的融合和社会化协作, 实现不断地自我创新和演化, 这决定了复杂软件系统必须是边界开放的, 必须能够不断融合新的主体的加入。

2) 行为涌现性: 涌现现象的本质特征是由小到大、由简入繁。复杂软件系统是基于一个个能够完成简单行为的独立软件、微服务或单元级信息物理系统 (cyber physical systems, CPS), 通过相互协作, 完成各种复杂的智能场景。

3) 自适应性: 自适应性是指复杂软件系统面对复杂的环境, 通过感知—分析—决策—执行的自适应环, 不断地进行自我调节动态地逼近目标的能力。

4) 持续演化性: 持续演化性是指复杂软件系统能够在与环境和其它复杂软件系统进行交互作用的过程中, 通过“感知—分析—决策—执行”的数据智能环^[2], 不断地“学习”或者“积累经验”, 并基于学习结果和经验改变自身结构和行为模式, 从而实现演变和进化的能力。

区别于传统的软件开发, 边界开放性和行为涌现性决定了复杂软件系统往往是在大量已经存在的自治系统基础上“成长”而来, 使得复杂软件系统的开发者不仅要关注

收稿日期: 2018-11-26; 修回日期: 2018-12-12。

作者简介: 于景洋(1993-), 男, 吉林集安人, 硕士研究生, 主要从事智能故障诊断、工业数据智能等方向的研究。

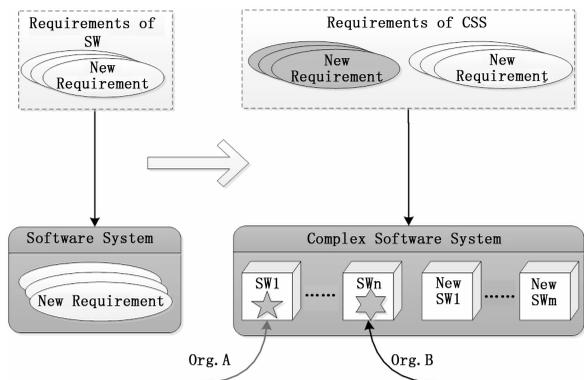


图 1 复杂软件系统的成长性构造

复杂软件系统层面新的需求, 还要兼顾到现有软件和服务的现状和它们之间的协作, 并且这些软件和服务很有可能来自于不同的组织。例如, 在当前智慧城市环境下的智能复杂软件系统的开发场景中, 很多都需要调用来自第三方的地图服务、图像识别服务、语音服务等。因此, 在复杂软件系统的开发过程中实现软件和服务资源的跨组织共享和协作, 面向某一行业实现软件、服务的共享和汇聚, 将大大加速该行业的复杂软件系统的开发能力和速度。

本文共分为六部分: 第一部分为引言, 阐述本文研究的相关背景并提出研究问题; 第二部分为国内外相关研究进展; 第三部分介绍了基于容器云和海云分形架构的复杂软件系统成长性构造框架; 第四部分介绍了面向工业领域的复杂软件系统成长性构造实践; 第五部分给出了一个该框架在智慧园区领域的实践案例。

1 相关研究

复杂软件系统研究一直是工程界研究热点问题。随着应用场景的转换和自身复杂性的增加, 软件需要平滑扩展的能力和适应自身环境变化的能力, 这就意味着它需要具备横向、纵向扩展的能力以及根据这些变化来调整自身行为的能力^[3]。因此, 支持大规模系统的有效开发、灵活部署和持续演化的软件开发方法和技术逐渐成为研究热点, 例如超大规模系统的软件工程^[4], 软件在线演化使能技术^[5]、面向复杂系统的“系统联盟”观点^[6]。工业领域中各种嵌入式监控软件、工业路由器和各类传感器相互互联, 协同完成计算、通信、分析等功能, 是支撑工业互联网的高效运转的数字中枢^[7]。因此, 如何驱动复杂软件系统的构造, 是软件领域研究者和工业实践者当前面临的重大挑战。王怀民^[1]等人认为, 复杂系统具有成员异质、边界开放、行为涌现、持续演化等特性。通过对复杂软件系统的内涵、形成特征和基本性质进行研究, 提出了复杂软件系统的“成长性构造”和“适应性演化”法则, 并认为与离线演化技术相比, 在线演化技术更有意义, 它能够使系统在结构修改和功能调整期间持续提供服务^[8]。汤磊^[9]等人结合自身的实际经验和复杂软件系统的特点和基本性质, 对“成长性构造”与“演化规则”进行了探讨。复杂系统的构造离

不开服务的支撑, 在面向服务的计算中, 服务是指对资源进行封装的自治、平台独立的实体, 其可以被描述、发布、发现和松散绑定。通过服务的抽象与封装, 快速、廉价、可互操作、可演化和大规模的分布式应用开发成为可能^[10]。服务发现和服务动态组合实质上就是建立动态连接的一种手段。更进一步, 在复杂软件系统中, 可以从系统架构的层次来看待动态连接问题。例如, 动态体系结构技术^[11-12]通过引入软件运行时体系结构模型, 并且将构件和连接子都建模为一阶实体, 通过运行时体系结构的变化来实现软件的新陈代谢和动态连接。

综上所述, 复杂软件系统具有“成长性构造”和“适应性演化”两条规律。针对成长性构造法则, 思考当前工业领域内复杂软件系统难以扩展, 企业强调本地数据安全性, 需要高效部署私有云等实际问题, 提出一种面向工业领域的复杂系统成长性构造框架。该框架使用容器云架构, 以轻量化容器技术作为突破口, 希望解决复杂软件系统的解耦、扩展、自适应、快速构建等问题。

2 基于容器云和海云分形架构的 CSS 成长性构造框架

应对复杂软件系统开发的挑战, 本文提出了基于容器云和海云分形架构的 CSS 成长性构造框架, 如图 2 所示。该框架的核心部分包括容器云操作系统和公共服务资源。

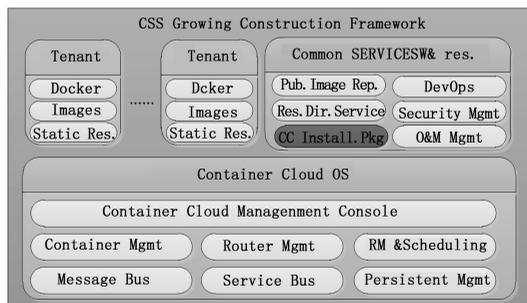


图 2 复杂软件系统成长性构造框架

2.1 容器云操作系统

容器云操作系统为容器云提供分布式操作系统, 它统一管理组成容器云的分布式服务器集群, 屏蔽底层管理细节, 诸如容错、调度、通信等, 让开发人员认为分布式服务器集群在逻辑上是一台服务器。它和单机 Linux 一样要解决五大类操作系统必需的功能, 即资源分配、进程管理、任务调度、进程间通信和文件系统, 可分别由 Docker、Mesos、Marathon/Chronos、RabbitMQ 和 HDFS/Ceph 来解决。

容器云操作系统包括容器云管理控制台、容器管理、路由管理、资源管理、资源编排、消息总线、服务总线、持久化管理等。其中, 容器云管理控制台负责管理云平台的硬件资源、服务器上部署安装的软件资源、租户信息、域名信息等; 容器管理基于 Docker Hub 进行构建, 可以提供高性能可伸缩的容器应用管理, 支持用 Docker 容器进行

应用生命周期管理；路由管理由 Nginx 和 Bamboo 构成，可以将终端用户请求路由到对应的服务器实例，并且提供应用动态注册以及负载均衡的功能；资源管理与调度是整个容器云操作系统的核心，采用 Mesos 作为其分布式资源管理框架，主要功能是管理应用实例的生命周期，汇报应用的运行状态；消息总线负责分布式系统中所有功能模块之间消息的通信与传输，采用 RabbitMQ，该模块需要支持 pub/sub 模式；服务总线基于 Marathon 进行构建，负责不同子系统之间的远程调用，能够为开发人员和运维人员提供统一的管理接口；持久化管理采用主流的分布式存储框架 Ceph，可以保证数据的高可用性与高可靠性。各个模块的组合可以充分保证系统提供动态满足应用需求的能力，保证其具有很好的弹性、灵活性以及可扩展性。

2.2 公共服务资源

公共服务资源包括公共镜像库、资源目录服务、容器云安装包、Devops、安全管理、运行维护管理等。

公共镜像库包含所有预定义的镜像文件、安装包和配置文件，并提供类似于 Dockerhub 的功能，能够自动化测试、开源应用、公共注册外，还提供了 restful API，让其他公有或私有镜像库可以下载或上传 Docker 镜像资源。

资源目录服务作为 CSS 成长性构造框架中的统一目录服务，提供数据目录、服务目录等服务，为复杂软件系统及其成员组件间实现标准化的数据交换和开放服务调用。复杂软件系统中的各成员自己决定要开放的数据资源和服务资源，使用标准的元数据将相关资源登记到数据目录和服务目录中，使得其它的复杂软件系统和成员可以通过目录服务，发现和使用开放的数据资源和服务资源，最终实现基于标准方式完成数据交换和服务整合。

容器云安装包是容器云管理控制台的安装文件，分形租户能够在控制台自动化设置容器云管理模块，从公共镜像库中下载好相关镜像后，分形租户能够获得和公有云完全相同的容器云能力。

DevOps 提供复杂软件系统的开发测试能力，通过提供的类似 Github 的协作平台，使能开发者可以协同定义设计复杂软件系统。此外，它还还为复杂软件系统及其包含的各种主体提供微服务架构支持，使其能够以微服务的方式封装各自的业务领域组件，并通过云服务使其自动化部署、智能化管理和功能交付变得更加简单。

安全管理服务为复杂软件系统成长性构造框架以及在它上面开发的各种复杂软件系统提供统一认证、授权和加密服务，确保数据和服务被安全的使用。

运维管理服务复杂软件系统成长性构造框架以及在上面开发的各种复杂软件系统提供统一监控、日志分析服务，确保整个环境的可靠运行。

2.3 海云分形架构

为了更快、更方便地满足企业基于私有云开发复杂软件系统的需求，海云协同计算分形架构技术结合容器技术和多租户技术，实现了容器云服务从云端到边缘端的快速

复制能力，并且边缘端拥有进一步向其子节点复制的能力。当某个复杂软件系统的开发组织希望建立私有容器云支撑系统的运行，同时还希望能够尽可能多地获取第三方分享的应用、数据和服务时，CSS 开发者可以通过下面指定的操作，完成海云分形架构的分形过程。

1) 租户创建：在 CSS 成长性构造框架中的容器云管理控制台中建立一个新的租户；

2) 私有 repository 搭建：私有云使用者可以选择安装本地镜像库服务和目录服务，其需要将公有 repository 中相应的资源信息下载到本地；

3) 云安装包的下载与执行：在 CSS 成长性构造框架中的容器云管理控制台中将云安装包下载到租户本地进行安装并运行；

4) 私有云硬件资源配置：安装包运行成功后，该租户可以进入其管理界面添加该私有云需要的服务器资源，此时该租户所在的服务器与其它新添加的服务器会打通无密码登陆，便于后续组件的安装；

5) 服务器角色配置：当服务器添加完毕后，会按照容器云操作系统中需要构建的模块自动为每台服务器配置其需要安装的功能组件。此时，该租户也可以根据实际的服务器资源情况手工调整各个模块所需安装的服务器；

6) 私有云构建：点击一步部署，这时各个服务器会按照步骤 5 中配置好的功能组件进行自动安装，各个功能组件将从 repository 下载。此时，如果该租户构建了私有 repository，则执行效率会更快；

7) 私有云域名配置：在第 4 步该租户的管理界面中可以配置该私有云的域名，这时会自动生成 Nginx 域名配置文件后拷贝到 Nginx 所在的目录中，并且重新加载其名下的所有域名配置文件，完成整个私有云的搭建。

通过以上的分形过程，CSS 开发者立刻拥有了一个和 CSS 成长性构造云服务几乎一样的私有云，它基于分布式拓扑结构的服务架构模型，采用无中心和冗余备份策略，拥有良好的可扩展性和鲁棒性，因此，能够很好地满足复杂软件系统的成长性构造需求。

3 面向工业领域的复杂软件系统成长性构造

复杂软件系统成长性构造框架从以下两个方面加速了复杂软件系统的开发。1. 海云分形架构使我们能够通过创建一个分形租户和订购所需的服务来复制一个新的容器云，并且它可以共享所有的父类资源，也可以向其后代传播能力。容器云支持开发人员实现 CSS 的可伸缩性，更容易使用微服务技术；2. 公共服务资源使开发人员能够跨组织进行协作。公共镜像库中提供了一个跨组织资源共享机制，该机制服务于复杂软件系统及其成员。共享资源包括 Docker 镜像、War 包和安装文件等。资源目录服务使开发团队能够跨组织发现和重用数据，这些能力能够促进 CSS 开发团队跨组织分享资源，实现社会化协作开发 CSS。尤其是在积累了大量的工业资源后，这些资源将加快缩短 CSS 的

市场开发时间。工业领域复杂软件系统成长性构造框架如图 3 所示。

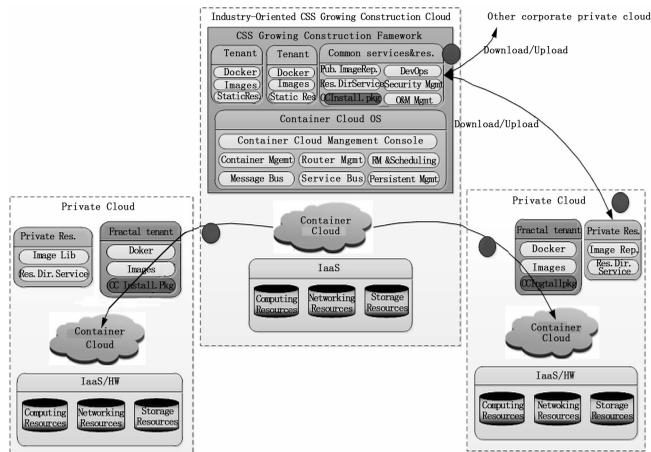


图 3 工业领域复杂软件系统成长性构造框架

4 验证案例：智慧园区共性服务云系统

基于本文所提出的面向工业领域的复杂软件系统成长性构造框架可实现面向智慧园区建设的智慧园区共性服务云平台，该平台能够面向不同园区的服务需求快速建立私有云环境，实现所需智慧园区服务的快速共享和复制，并基于这些服务快速构造面向某一特定智慧园区的复杂软件系统。该平台的目标是能够逐渐汇聚智慧园区的共性服务，通过平台实现跨园区的资源共享，支持智慧园区系统的开发者们能够基于平台重用各种资源，快速开发符合自身要求的智慧园区复杂软件系统。区别于传统复杂软件系统开发时间长、部署过程复杂、园区各自为政的特点，该平台基于轻量化容器技术和微服务架构，能够快速搭建私有云环境和重用各种服务。尤其是针对部分园区出于对安全考虑，不愿将数据上传至公有云平台的情况，这种混合云的实现方式不但能够满足其构建私有云环境，把数据以及企业应用保存在本地，从物理环境上保证数据、应用安全的需求，而且为复杂软件系统的扩展和成长性构造提供了诸多服务支持，使用户获得云端服务资源共享带来的便利。

面向智慧园区用户的定制化需求，使用该框架开发复杂软件系统的具体流程如下：

- 1) CSS 开发者在云端新建一个分形租户，该租户包括云操作系统安装包、所需各类镜像资源等；
- 2) CSS 开发者在终端园区内部，配置完成私有云所需的硬件环境，确保与云端之间网络互通；
- 3) 通过分型租户连接园区内部终端硬件集群，为终端硬件集群搭建私有云环境；
- 4) CSS 开发者在本地使用私有云环境，基于获得的云端资源，实现 CSS 的开发过程；

值得一提的是，CSS 开发者在私有云环境下产生的各种软件、数据等资源，能够在海端容器化为服务，并通过将该容器上传至公有云镜像库供公众使用并获得收益。而

且，私有云环境具备弹性伸缩、灾备等云主机特性，随着业务量和用户的增长能够持续为 CSS 提供扩展能力。

目前，快速扩展的智慧园区共性服务云平台提供以下基础服务：

- a) 产业地图：提供了基于 GIS 地图的工业用地管理、落地企业管理以及土地利用情况调查等服务。
- b) 项目管理：产业项目的全过程跟踪预警管理，提供了包含信息源的分配，意向、签约、在建、投产各阶段信息跟踪，以及项目实施进度的预警等服务。
- c) 招商管理：提供了包含土地、厂房、商业楼宇、办公楼宇的招商资源数据管理服务。
- d) 企业管理：提供了全区企业数据汇聚管理功能，包含企业落地信息、统计信息、工商信息、税收信息、认定信息、知识产权信息等查询及管理服务。
- e) 绩效评估：包含园区的综合调查统计服务，以及依托于平台大数据分析模型所构建的企业绩效评估服务。

除此之外，还包含产业发展全景视图、区域发展预测预警、招商引资智能推荐、产业生态分析、企业信用评估和管理者驾驶舱等十几项定制化增值服务，这些服务大多是需要根据具体园区数据二次定制实现的智能服务。通过基于 CSS 成长性构造框架的智慧园区共性云服务平台，为各种园区复杂软件系统的开发提供平台支持，帮助他们节省资金成本，缩短部署时间和快速推出服务。

综上所述，基于本文所提框架建立的智慧园区共享服务云平台在智慧园区复杂软件系统成长性构造方面具有以下 5 大技术优势：

- 1) 快速的 CSS 开发能力：通过内置海云分形架构、目录服务和资源共享的开发环境，支持 CSS 的成长性构造和快速交付。
- 2) 复杂软件系统平滑演进能力：通过 CSS 各成员的容器化、模块化定义，提供了 CSS 可分期建设，成长性构造能力，并赋予所有 CSS 成员独立演化的能力。
- 3) 统一硬件/存储/安全方案：通过容器云操作系统提供了高效安全的基础资源集群控制，易管理易维护，实用环保。
- 4) 第三方系统集成能力：通过标准目录服务和元素据的定义，支持多层次集成：数据集成、服务集成和应用集成。
- 5) 系统资源共享能力：按需配置资源，提高资源的利用率；统一标准的数据管理方法，实现云共享。

5 结束语

本文提出了一种面向工业领域的复杂软件系统成长性构造框架，该框架使用容器云、多租户和统一目录服务技术，实现跨组织的资源共享、协同和面向复杂软件系统成长性构造私有云环境的快速生成，使用此框架不需要从零开始构建系统，大大减少了开发部署时间。从云端向终端拓

(下转第 146 页)