面向智能化软件开发的开源生态大数据

张洋[1]，王涛[1]，尹刚[2,3]，余跃[1]，黄井泉[3]
1. 国防科技大学计算机学院，湖南 长沙 410073；2. 绿色计算产业联盟，北京 100036；
3. 湖南智擎科技有限公司，湖南 长沙 410073

摘要
开源软件开发过程中包含大量有价值的数据，针对其数据规模巨大、碎片分散、快速膨胀的特点，研究了软件工程开源生态大数据体系，提出了一种自生长的采集处理框架与汇聚共享环境，阐述了基于软件工程开源生态大数据的智能化软件开发，以及基于软件工程开源生态大数据分析挖掘的典型应用，为面向智能化软件开发的开源生态大数据研究与应用提供相关指导。

关键词
智能化软件开发；开源软件；开源生态；大数据

中图分类号：TP311.5　　文献标识码：A　　doi: 10.11959/j.issn.2096-0271.2021007

Big data of open source ecosystem for intelligent software development

ZHANG Yang[1], WANG Tao[1], YIN Gang[2,3], YU Yue[1], HUANG Jingquan[3]
1. Department of Computer Science, National University of Defense Technology, Changsha 410073, China
2. Green Computing Consortium, Beijing 100036, China
3. Hunan Intelligent Engine Technology Co., Ltd., Changsha 410073, China

Abstract
The open source software development process contains a lot of valuable data, which is huge in scale, fragmented, and rapidly expanding. Aimming to the characteristics, the big data structure of open source ecosystem of software engineering was studied, and a self-growing collection and processing framework and a convergence and sharing environment was proposed. The related research on the development of intelligent software based on open source big data of software engineering, and typical applications based on analysis and mining of open source big data of software engineering were expounded, and relevant guidance for the research and application of big data of open source ecosystem for intelligent software development was provided.

Key words
intelligent software development, open source software, open source ecosystem, big data

1 引言

自20世纪末以来，开源软件在现代社会的各个领域得到了广泛的应用，取得了令人瞩目的成就。Black Duck公司2017年的调查报告[1]显示，全球86%的企业在搭建业务时全部或部分使用了开源软件，其中60%的公司还在增加开源软件的使用比重。开源软件的开发活动以互联网软件社区为平台，其开发过程和制品数据对外开放，允许不同类型的开发者参与其中，形成一种大众参与的开源模式[2]，给开源世界带来了强大的生产力。开源模式中大众贡献者可以自由地实践分布式协同，催生了许多群体化软件开发方法和一系列优质的开源社区。特别是，近年来云计算、大数据、人工智能、物联网等对国民经济发展产生重大影响的信息化基础设施绝大多数是以开源软件为核心构建而成的，开源软件已经在全球软件产业占据主导地位。

与传统工业化软件生产相比，大众化开源软件生产的开发数据和应用数据高度开放且规模巨大。目前，支持大众化软件生产和应用活动的开源社区包含了大量有价值的数据，如软件代码、软件版本、容器镜像等软件制品和过程数据，以及软件问答、软件评价等软件交流和反馈数据，这些数据涵盖开发数据、交付数据及应用数据等全维度数据类型，涉及开发制品、开发过程、软件产品、软件镜像、咨询讨论与应用问答等各个方面，具有规模巨大、碎片分散、快速膨胀的特点。如何构造高扩展、高性能的软件工程开源生态大数据处理体系结构，建立多源异质、广泛关联、语义丰富、覆盖全面的软件工程开源生态大数据环境，分析提炼软件知识并设计实现辅助开发工具，以提升软件开发的智能化程度，已成为重要的科学问题。

本文研究了软件工程开源生态大数据体系，并提出了一种自生长的采集处理框架与汇聚共享环境；然后，介绍了基于软件工程开源生态大数据的智能化软件开发，以及基于软件工程开源生态大数据分析挖掘的典型应用，以期为面向智能化软件开发的开源生态大数据研究与应用提供相关指导。

2 软件工程开源生态大数据

传统软件数据挖掘主要关注同质和局部软件工程数据，难以适应软件工程开源生态大数据呈现出的异构多源、类型复杂、持续增长、广泛互联等新特性，全局视角下考察软件大数据价值仍面临巨大挑战。本文以“主动感知、定向采集、多源关联、增量检测”机制构建自生长的软件大数据环境，建立综合互联的大样本软件数据集合，以支持多维度、多谱系、贯通性的软件知识提炼和智能释放。

2.1 软件工程开源生态大数据体系

通过对软件生态进行全面梳理和分析，本文将软件社区分为开发社区和应用社区，涵盖软件开发、发布和应用等不同阶段，对软件工程大数据进行调研分析。软件工程大数据以代码、文档、开发记录等文本为主体，语义丰富。为此，本文构建了系统的软件工程开源生态大数据体系（如**图1**所示），涉及开发制品、开发过程、软件产品、软件镜像、咨询讨论和应用问答等各个方面，覆盖GitHub、Apache、Topcoder、Docker Hub、OSCHINA以及Stack Overflow等各类型主流开源社区，为软件工程研究和实验提供了一个较完整

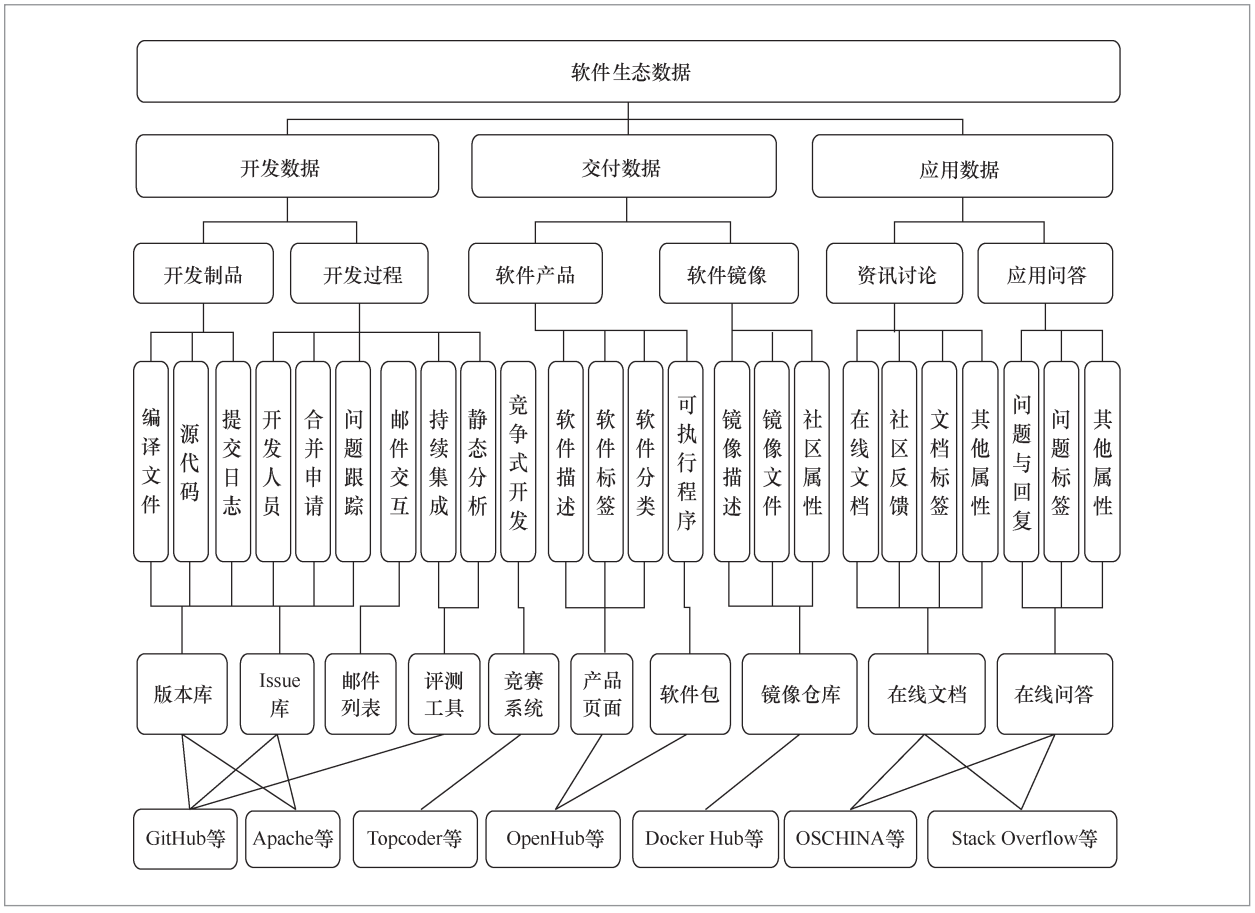


图1 软件工程开源生态大数据体系

的全局视图。

该体系主要包括开发数据、交付数据和应用数据三大类，每一类又被细分为多个子类。最终，该分类体系与当前的多种软件仓库、社区和论坛的具体数据格式建立了映射。具体如下。

（1）开发数据

开发数据以软件工程中软件开发设计的制品和过程为核心，其中开发制品包括源代码、编译文件和提交日志，开发过程则涉及开发人员、合并申请、问题跟踪、邮件交互、持续集成和竞争式开发等多个方面。具体地，开发数据对应的数据源包括版本库、Issue库、邮件列表、评测工具和竞赛系统，涉及的数据源实例有GitHub、Apache和Topcoder等。

（2）交付数据

交付数据主要是交付生产环境的软件产品，同时随着虚拟化技术和容器技术的发展，软件镜像作为一种特殊的软件产品交付方式逐步兴起并大量存在，因此其也单独作为交付数据的一个子类。软件产品数据涉及软件的方方面面信息，包括软件描述、软件标签、软件分类和可执行程序本身；软件镜像与之类似，包括镜像描述和镜像文件等。交付数据主要来源于软件产品页面、软件包和镜像仓库，具体实例包括OpenHub和Docker Hub等。

（3）应用数据

应用数据主要包括资讯讨论和应用

问答，其中资讯讨论包括在线文档、社区反馈、文档标签以及其他属性等；应用问答则主要针对以提问和解答为主要形式的在线社区活动，涉及的数据包括问题与回复、问题标签和其他属性等。应用数据的主要数据来源为在线文档和在线问答，具体实例有OSCHINA和Stack Overflow等。

在此基础上，本文提出了分布式异构存储的总体策略，设计和制定了多源异构的软件工程大数据管理框架与环境。其总体主要包括以下4个层次。

- 数据源：软件工程大数据涵盖开发、发布、应用、运维等不同过程、不同类型和不同源的数据，包括版本库、代码仓库、配置制品、软件镜像等。
- 数据存储：主要实现对大规模异构软件工程大数据的高效存储和访问。
- 数据处理：围绕特定的任务和目标，将存储的数据按需展开，并进行相应的处理，形成软件知识库。通过数据解析、融合等技术进行数据的二次加工和处理，包括通过分析不同数据类型之间的关联和依赖、基于图数据库等存储技术构建软件领域的知识图谱，进一步通过数据按需展开机制有效降低存储资源的占用情况。
- 数据实例：通过丰富的接口和服务，针对不同的需求和应用提供相应的数据服务。针对围绕的不同数据类型，对外提供的数据服务被分为4类，包括以项目为中心的数据服务、以测试为目标的数据服务、以人为中心的数据服务和以运维为目标的数据服务等。

2.2 软件工程开源生态大数据采集处理框架

本文提出一种“增量式、多模式”的自生长数据采集处理框架（如**图2**所示），该框架能够针对不同类型的软件数据进行汇聚、收集和整理。具体地，针对网页数据、版本库数据、缺陷库数据等不同类型的数据，本文研究了主动感知、定向采集、多源关联和增量检测等关键技术，设计部署了分布式爬虫，实现了网页爬虫、基于应用程序接口（application programming interface，API）的数据获取和数据包直接下载等多种收集方式。具体如下。

- 基于网络爬虫的数据收集方法：针对特定的软件库采用定点爬取的方式，通过分析特定数据源网页中的数据特点和Schema格式，基于爬虫常用的标签匹配和正则表达式匹配等策略获取相应的数据信息。为了解决爬取效率低和重复爬取的问题，采用分布式网络爬虫技术进行多任务的并行处理，从而大幅提高大规模软件数据的爬取效率，另外，基于时间戳等信息实现周期性、增量式爬取，避免数据的重复获取。其中，多数软件数据基于爬虫技术获取，包括Apache基金项目的源码、邮件、网页和版本控制，Eclipse社区项目的缺陷报告和代码，配资代码库和Docker Hub中的元数据信息、代码制品及容器的Dockerfile，CSDN的博客、问答和论坛等。
- 基于API的数据收集方法：除了网络爬虫，部分开源软件库对外提供获取和下载数据信息的开放API，因此可以通过调用API的方式获取相应的数据信息，其中包括Topcoder的众包开发数据、Apache基金会项目的缺陷报告等。
- 数据包直接下载方法：一些社区对历史数据进行压缩存档，并直接对外提供数据下载地址，例如Stack Overflow社区的文档数据等。

在此基础上，软件工程开源生态大数据采集处理框架整体可分为3层，包括数

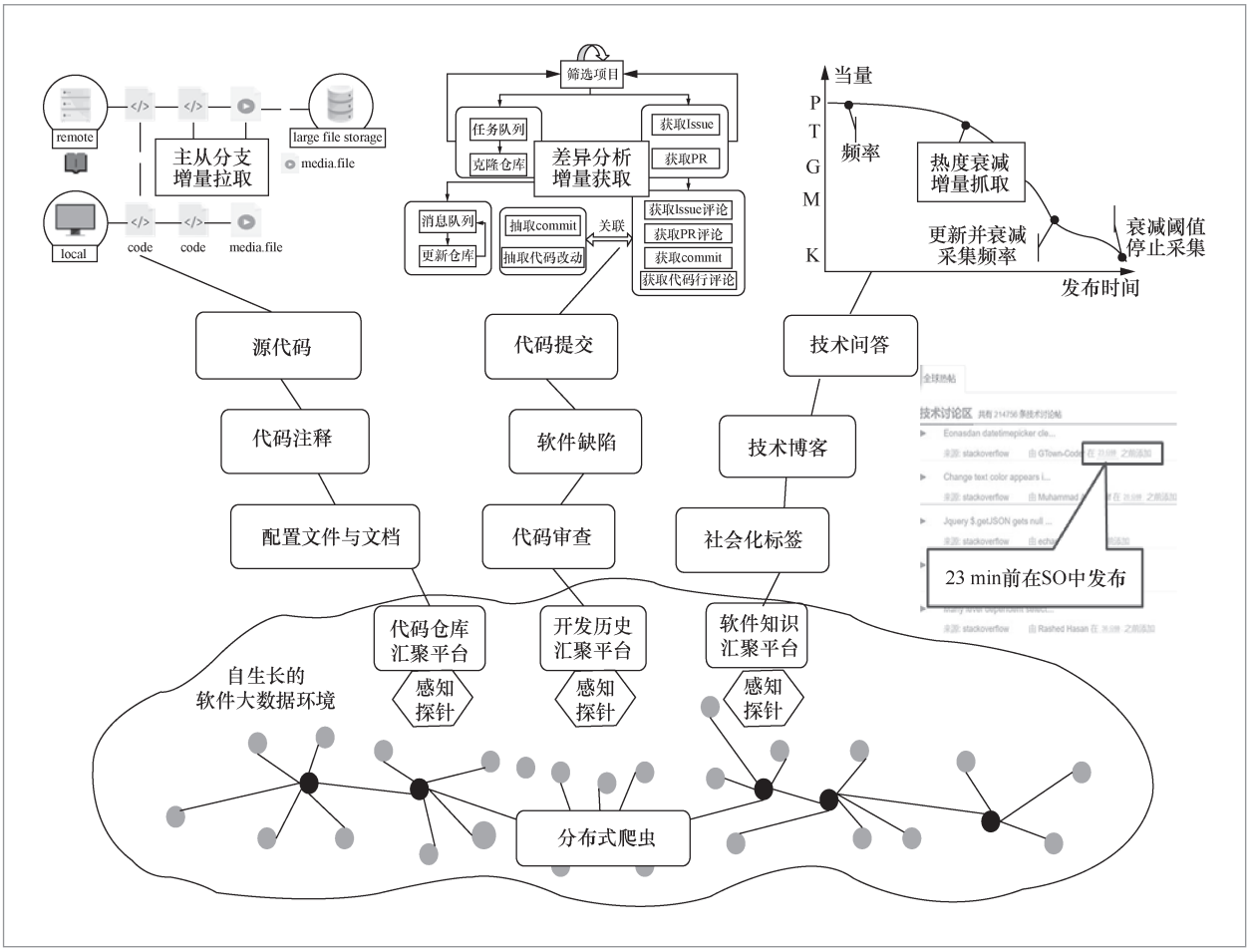


图 2 软件工程开源生态大数据采集处理框架

据获取层、数据分析层以及持久化层，通过3个层次的协同配合，最终完成从数据的最初采集、数据的分析处理，到最后数据的展示。其中，数据获取层的主要功能是完成数据的采集工作，为平台提供高效、稳定、持续、准确的数据服务。数据分析层对数据获取层获取的页面信息进行抽取，提取出每个页面中的关键信息，并对抽取结果进行验证，通过验证的页面数据会被存储到数据库中，为数据分析做准备。该层还有一些数据挖掘算法，对所抽取的数据进行数据分析操作，包括社区关联、软件评估等。持久化层按照最终需要展示的数据格式，对之前得到的数据处理结果进行处理，并将处理结果存放于数据缓冲池中，最终源源不断地向展示平台传输，为平台的展示提供数据支持。

2.3 软件工程开源生态大数据的汇聚共享

针对软件工程开源生态大数据规模大、多样性和异构性的特点，本文采用开放共享、分散存储、平台汇聚、按需获取的方式，形成一个大规模软件工程大数据共享平台，如图3所示。目前该平台已实现对全球PB级开源代码和镜像等软件工程大数据的跟踪、获取，为软件工程研究和实

图 3　软件工程大数据共享平台

践提供了坚实的数据基础。

软件工程大数据共享平台采用分散存储、平台汇聚的模式提供共享管理和在线访问统一入口，屏蔽底层数据存储的分散性和多样性，提供风格一致的门户环境。具体存储结合实际数据特征采用结构化存储与非结构化存储相结合的方式，平台核心架构设计如下。

- 以多源异构的软件工程开源生态大数据为数据源，基于网络爬虫技术实现数据的定向收集和通用收集，同时还包括基于API的数据获取和直接获取。
- 采用结构化与非结构化相结合的方式进行存储，针对元数据信息主要采用基于关系数据库和NoSQL数据库的方式进行存储，并结合联合文件系统（aufx和zfs等）等技术支持非结构化数据（尤其是文档和代码）的有效存储。为了减少数据存储的资源占用问题，针对软件版本数据提供了按需进行版本分支展开的机制。
- 为本地数据增加索引，提供知识描述表作为本地数据描述、共享和权限控制的基本单位，同时支持知识描述表的继承关系和多种映射规则定义，并提供知识图谱作为知识描述表之间逻辑关联关系的描述工具和知识检索工具。
- 平台提供统一访问门户，并可根据需求动态调整数据类别条目，通过统一资

源接口，支持用户快速获得原始或加工过的各类软件工程数据。

平台被部署于UCloud和阿里云上，提供汇聚数据说明及访问入口，小规模数据集被直接上传至共享平台，大规模数据集/应用服务由数据共享单位自行管理。该平台具有高度灵活性和可扩展性，能够基于软件工程开源生态大数据体系建立多样化的数据类型导航机制，涉及的典型软件工程数据包括软件工程科研数据、开源代码数据、Docker镜像数据以及知识图谱数据等不同类型。

3 基于软件工程开源生态大数据的智能化软件开发

立足于软件工程开源生态大数据，越来越多的研究团队基于不同类型的开源大数据进行智能化的软件开发研究。如**图4**所示，从基于代码、日志、邮件等开发数据的知识图谱构造，到融合缺陷、社区问答等开发数据和应用数据的软件代码缺陷定位与修复、问答资源推荐，再到针对配置文件、容器镜像等交付数据的运维数据管理，这些前沿探索为人们更好地利用开源大数据辅助智能化软件开发提供了方法指导与工具支持。

3.1 基于开发数据的软件知识图谱构造

相关研究针对软件缺陷发现而构造的软件缺陷知识图谱（bug knowledge graph，BKG）[3]利用主题模型LDA（latent dirichlet allocation）和文本相似度算法分别自动化地进行实体识别和关系抽取，但其受限于缺陷报告数据和源代码数据，不支持其他数据源的知识扩展。针对通用软件的弱点，相关研究推理并构造了CWE KG（common weakness enumeration knowledge graph）[4]，通过预定义的模板进行规范，采取社区平台的方式进行人工编辑，但是其并非自动化方法，受限于与软件弱点相关的文档数据，不支持其他数据源的知识扩展。此外，针对软件开发在线问答，相关研究构造了HDSKG（harvesting domain specific knowledge graph）[5]，通过依赖解析和基于规则的方法进行实体和关系的识别，采用支持向量机（support vector machine，SVM）的分类算法评估三元组，但是其是半自动化方法，主要来源于Stack Overflow的在线问答数据，可扩展性受限。因此，基于软件工程开源生态大数据中的开发数据，相关研究通过探索面向多源异质、动态增长的软件大数据的软件知识自动识别、抽取、关联与融合过程，提炼出大规模、内容全面、语义丰富的软件知识图谱，为软件构造过程中的智能化辅助服务提供了基础支撑[6-7]。给定一个可复用的软件项目及与其相关的大量软件项目数据，这些研究提出的方法能够自动从中抽取出概念、实体等结点，并建立这些结点之间多源异构、类型丰富的关系，形成这个软件项目特定的一个软件项目知识图谱。该知识图谱基于图数据库，实现了对知识图谱的存储、索引、查询等基础支持的建立，为软件项目数据的解析、关联融合与知识挖掘提炼提供了支撑环境。在数据解析方面，图谱兼容多种不同类型的软件工程数据，包括：软件源代码、各种版本控制系统（如SVN、Git）的版本记录、邮件列表日志、缺陷追踪系统日志、HTML网页文档、Word文档、PDF文档等。同时，该知识图谱构造工具内置了多种可追踪性关联分析与知识补全模块，能够充分对不同来源的软件数据间的语义关联进行智能恢复补全，且能够从数据中提炼出更适合管理者、开发者与复用者理解的知识。

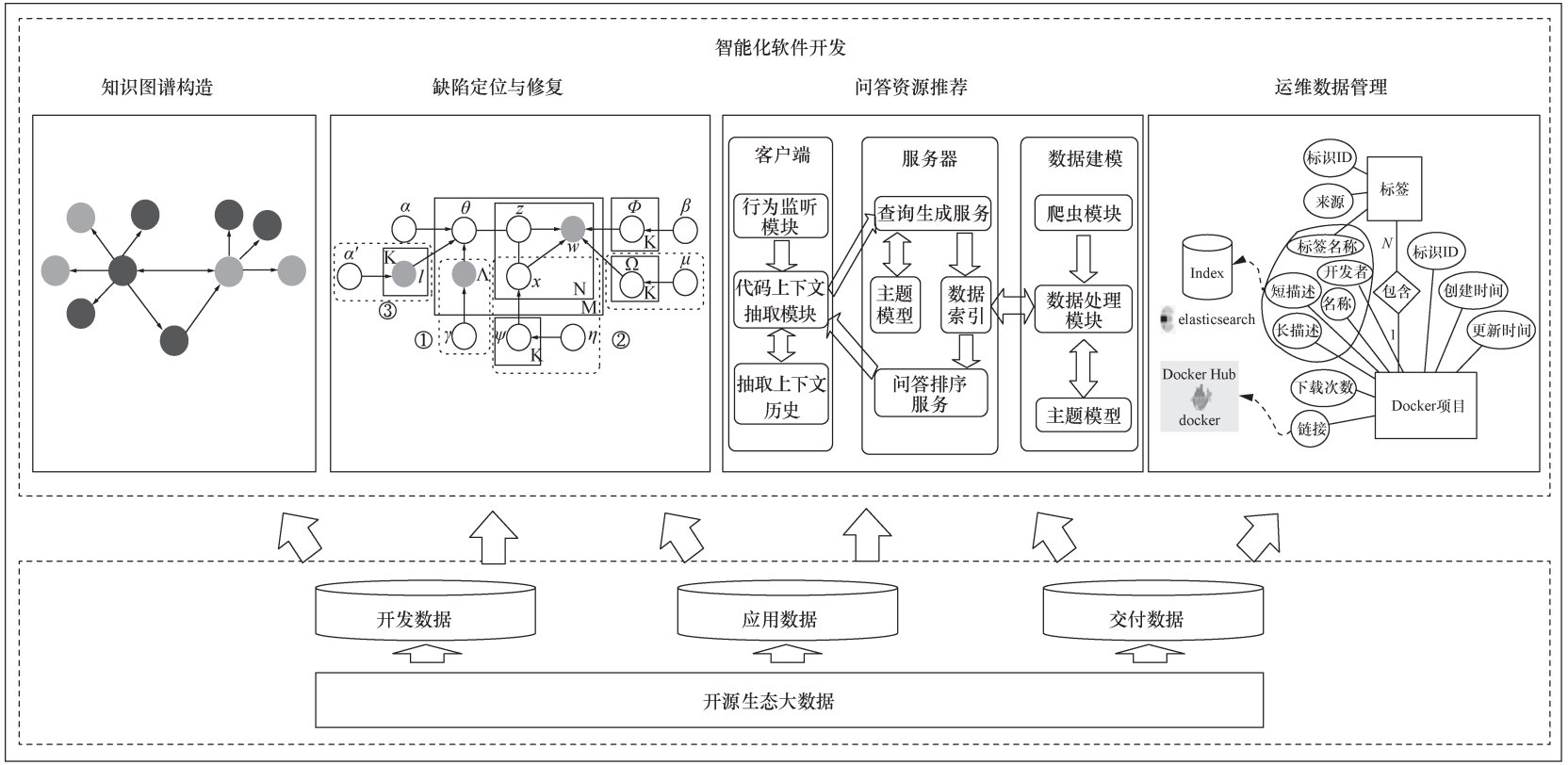


图 4　基于软件工程开源生态大数据的智能化软件开发研究

3.2 基于缺陷与社区问答数据的软件代码缺陷智能定位与修复

基于软件工程开源生态大数据中的开发数据和应用数据，相关研究团队围绕基于缺陷和社区问答数据的软件代码缺陷智能定位与修复问题开展了大量研究[8-9]。他们针对缺陷报告与关联代码间的词共现关系、缺陷报告文本附着的元数据，以及代码规模越大越容易出现软件缺陷等特点，构造了一个基本的监督式文本主题模型STMLocator，并给出了相关训练与预测算法。相较于传统的信息检索方法[10]以及基于频谱的方法[11]，该方法能够将修复历史作为监督信息，同时将文本相似度与语义相似度结合，实现更高精度的缺陷定位。在开源项目Eclipse的多个子项目（PDE、Platform、JDT）收集的真实数据上的预测准确率相较同类工作最高提升23.6%。另外，为了研究缺陷报告文本附加信息对缺陷定位的影响，Wang Y J等人[8]提出了L2SS+模型簇，在多个实际数据集上的实验结果表明，L2SS+CM模型，即产品模块信息对缺陷定位准确率影响最显著，预测准确率较同类工作最高提升18.7%。在基于社区问答网站的软件缺陷修复信息智能推荐技术方面，为了进一步精确获取和分类社区问答网站的内容，相关研究分析了网站文本内容的自动标签推荐问题，针对标签与文本词之间的共现关系、标签之间的关联关系等特点，Tang S J等人[9]提出了一个iTAG深度学习模型，并给出了相关训练与预测算法。在从Stack Overflow程序员社区问答网站等获取的真实数据集上，iTAG模型较其他同类工作（#TagSpace[12]、Maxide[13]等）在预测准确率上有显著提升。另外，iTAG模型在结果可视化解释、发现更多合理的标签等方面具有优势。

3.3 基于上下文感知的软件问答资源推荐技术

近些年，相关研究团队立足于软件工程开源生态大数据中的应用数据和交付数据，围绕上下文感知的软件问答资源推荐技术开展了富有成效的研究[14]。目前大部分的软件问答推荐系统没有考虑上下文，有的虽然考虑了上下文，但是还是以代码本身关键词为主，没有考虑其中的语义信息，也没有充分挖掘已有的海量的问答知识[15-17]。参考文献[14]针对此设计了基于主题模型的软件问答推荐系统，推荐系统通过主题模型组织整理现有的全量问答数据，并推断用户代码的行为，通过不同场景下的用户行为抽取不同的代码上下文，根据代码上下文向开发人员推荐不同的问答。推荐问答的关键操作是计算代码上下文与问答数据的相关性，其中的一个关键问题在于问答数据的范围，直观的选择是计算代码上下文与全量问答数据的相关性，然而这在实际应用中是不可能的，因此首先要做的是缩小计算范围，计算代码上下文和一部分问答数据的相关性。为了缩小计算的范围，降低计算复杂度，Shao B等人[14]抽取代码上下文中的关键词，利用关键词检索得到较小的问答数据集合，然后计算代码上下文和每一个问答数据的相关性，对所有结果进行排序后推荐给开发人员。推荐系统监测开发人员开发代码，当开发人员停止代码或者程序运行时，抛出异常等行为触发推荐机制，推荐系统通过抽象语法树和主题模型分析代码上下文，抽取关键词，检索多源数据，对返回的数据进行分析排序，并最终展示给用户。

3.4 面向软件一体化开发运维的数据汇聚和知识管理

基于软件工程开源生态大数据中的交付数据，相关研究团队针对Docker镜像设计了一种海量数据汇聚、管理、知识抽取和质量评价的系统化方案和服务[18-19]。其成果首先实现了增量式、高并发的Docker数据汇聚和管理方法，支持对Docker Hub上百万级Docker项目数据的自动获取与增量更新，实现了项目数据的可发现和可追踪。具体地，在数据汇聚方面，针对Docker项目数据量大、难以获取项目整体列表等技术问题，参考文献[18-19]提出了一种增量式的数据并发获取方式，突破了高命中率的检索关键词生成算法、高效的Docker项目元数据更新比对与融合算法，从而提高数据获取的覆盖度以及对Docker项目数据更新的敏感度。在基于返回的检索结果列表进行Docker项目详细元数据获取的过程中，考虑到检索结构数据量大（通常能够达到万级或十万级），他们采用“分而治之”的策略，对大规模的搜索返回结果实施并发的多线程获取方式，显著地提升了数据汇聚效率。最后，基于多种启发性规则对初步获取的Docker项目元数据进行分析、对比和过滤，识别其中的冗余和更新内容，保证数据内容的一致以及数据内容的更新，实现了海量Docker项目相关数据的高效汇聚和增量更新。

4 基于软件工程开源生态大数据分析挖掘的典型应用

近年来，越来越多的基于软件工程开源生态大数据分析挖掘的应用诞生，并不断发展。其中，针对软件问答社区（如Stack Overflow）、开源软件项目（如Apache、安卓项目等）、软件开发工具（如Eclipse）、开发者数据（众包开发者数据、开发过程数据等）类型的数据进行了汇聚和收集整理，OSSEAN构建了面向全球开源软件的检索与分析平台。OSSEAN平台的基本思路为：在软件消费社区中找到关于软件项目的文档；利用这些文档对软件进行评估、比较、排序。通过获取到的海量的开源社区数据，OSSEAN平台能够提供一些有趣的服务，如开源生态系统的度量、软件排序以及热点话题分析。OSSEAN平台数据获取模块已覆盖全球20多个主要的开源社区，并对这些社区进行持续监控、实时抓取，抓取和分析的数据包括超过140万个开源项目/仓库的元数据以及超过2 000万条在线讨论数据。同时，OSSEAN平台的跨社区关联与分析模块通过对多源异构数据的深度互联，建立相应的异质信息网络，并在此基础上实现对开源软件的分析、检索与排序等服务[20]。

SnowGraph（software knowledge graph）在对开源社区软件项目进行大量调研与实践的基础上，设计并实现了软件项目知识图谱自动构造支持平台。对于一个待复用的软件项目，SnowGraph能够以自动化的方法对其中的多源异构数据进行处理，将分散、非结构化的信息提炼为广泛关联、语义性强的知识，并以知识图谱的形式进行表示；在此基础上，将知识图谱融入机器对无结构文本的处理过程，进而为复用者提供准确有效的智能问答服务，从而提高软件复用过程的效率与质量。目前，SnowGraph的项目原始数据规模大约为500 GB，其中包含软件项目源代码、软件开发版本记录、软件缺陷追踪记录、软件开发邮件记录以

及Stack Overflow在线论坛记录等。最终，SnowGraph自动构造了192个软件项目知识图谱，图数据库规模大约为100 GB。此外，SnowGraph平台集成了软件项目的智能问答服务系统，即开发人员在复用一个软件项目时，提出与软件领域相关的开发问题，知识图谱返回相应的代码/文档作为答案，从而辅助开发人员进行软件复用。与BKG[3]、CWE KG[4]、HDSKG[5]等其他主流软件知识图谱相比，SnowGraph采取了多种信息抽取的方法来协同进行知识图谱的全自动化构造，自动化程度更高，并且具备很好的通用性和可扩展性，能够对未来可能出现的新的知识需求、知识来源，以及知识抽取、关联、提炼等进行支持。此外，SnowGraph实体类型和关系类型明显比其他主流软件知识图谱丰富，从而在对应的软件知识表示和知识利用的任务上具备更好的效果。

CodeWisdom开发了代码大数据与智能化软件开发研究成果展示与服务平台。该平台在GitHub等开源软件社区的软件代码及软件开发历史、Stack Overflow等软件开发问答网站的问答知识，以及API文档等互联网软件开发资源的基础上，利用程序分析、深度学习、自然语言处理、知识图谱、数据挖掘等技术，充分发掘代码大数据中蕴含的知识，通过检索、推荐、问答、可视化等多种手段提供智能化软件开发支持。

5 结束语

本文提出了软件工程开源生态大数据体系，构建了自生长的采集处理框架和汇聚共享环境，并阐述了基于软件工程开源生态大数据的智能化软件开发的相关研究以及典型应用情况。面向智能化软件开发的开源生态大数据研究具有很强的研究意义和实际价值，未来需要进行进一步的深入探索。

参考文献:

[1] Black Duck Software, Inc. 2017 open source 360 degree survey[R]. 2017.

[2] GOLDEN B. Succeeding with open source[M]. New Jersey: Addison-Wesley Professional, 2005.

[3] WANG L, SUN X B, WANG J W, et al. Construct bug knowledge graph for bug resolution[C]//The 2017 IEEE/ACM 39th International Conference on Software Engineering Companion. Piscataway: IEEE Press, 2017: 189-191.

[4] HAN Z B, LI X H, LIU H T, et al. DeepWeak: reasoning common software weaknesses via knowledge graph embedding[C]//The 2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering. Piscataway: IEEE Press, 2018: 456-466.

[5] ZHAO X J, XING Z C, KABIR M A, et al. HDSKG: harvesting domain specific knowledge graph from content of webpages[C]//The 2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering. Piscataway: IEEE Press, 2017: 56-67.

[6] WANG M, ZOU Y Z, CAO Y C, et al. Searching software knowledge graph with question[C]//International Conference on Software and Systems Reuse. Cham: Springer, 2019: 115-131.

[7] 凌春阳, 邹艳珍, 林泽琦, 等. 基于图嵌入的软件项目源代码检索方法[J]. 软件学报, 2019, 30(5): 1481-1497.
LING C Y, ZOU Y Z, LIN Z Q, et al. Approach to searching software source code with graph embedding[J]. Journal of

Software, 2019, 30(5): 1481–1497.
[8] WANG Y J, YAO Y, TONG H H, et al. Bug localization via supervised topic modeling[C]//The 18th IEEE International Conference on Data Mining. [S.l.:s.n.], 2018: 607–616.
[9] TANG S J, YAO Y, ZHANG S W, et al. An integral tag recommendation model for textual content[C]//The 33rd AAAI Conference on Artificial Intelligence. [S.l.:s.n.], 2019: 5109–5116.
[10] LUKINS S K, KRAFT N A, ETZKORN L H. Bug localization using latent dirichlet allocation[J]. Information and Software Technology, 2010, 52(9): 972–990.
[11] XUAN J F, MONPERRUS M. Learning to combine multiple ranking metrics for fault localization[C]//The 2014 IEEE International Conference on Software Maintenance and Evolution. Piscataway: IEEE Press, 2014: 191–200.
[12] WESTON J, CHOPRA S, ADAMS K. #TagSpace: semantic embeddings from hashtags[C]//The 2014 Conference on Empirical Methods in Natural Language Processing. [S.l.:s.n.], 2014: 1822–1827.
[13] XU M, JIN R, ZHOU Z H. Speedup matrix completion with side information: application to multi-label learning[C]//The Advances in Neural Information Processing Systems. [S.l.:s.n.], 2013: 2301–2309.
[14] SHAO B, YAN J F. Recommending answerers for stack overflow with LDA model[C]//The 12th Chinese Conference on Computer Supported Cooperative Work and Social Computing. New York: ACM Press, 2017: 80–86.
[15] YAO Y, TONG H H, XIE T, et al. Detecting high-quality posts in community question answering sites[J]. Information Sciences, 2015, 302(C): 70–82.
[16] AHASANUZZAMAN M, ASADUZZAMAN M, ROY C K, et al. Mining duplicate questions of stack overflow[C]//The 2016 IEEE/ACM 13th Working Conference on Mining Software Repositories. Piscataway: IEEE Press, 2016: 402–412.
[17] ARWAN A, ROCHIMAH S, AKBAR R J. Source code retrieval on Stack Overflow using LDA[C]//The 2015 3rd International Conference on Information and Communication Technology. Piscataway: IEEE Press, 2015: 295–299.
[18] YIN K, ZHOU J H, CHEN W, et al. D-Tagger: a tag recommendation approach for docker repositories[C]//The 10th Asia-Pacific Symposium on Internetware. New York: ACM Press, 2018: 1–10.
[19] CHEN W, ZHOU J H, ZHU J X, et al. Semi-supervised learning based tag recommendation for Docker repositories[J]. Journal of Computer Science and Technology, 2019, 34(5): 957–971.
[20] YIN G, WANG T, WANG H M, et al. OSSEAN: mining crowd wisdom in open source communities[C]//The 2015 IEEE Symposium on Service-Oriented System Engineering. Piscataway: IEEE Press, 2015: 367–371.

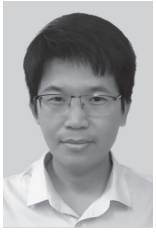
作者简介

张洋（1991– ），男，博士，国防科技大学计算机学院助理研究员，中国计算机学会会员，主要研究方向为实证软件工程、软件版本库挖掘、DevOps等。

王涛（1986– ），男，博士，国防科技大学计算机学院副研究员，中国计算机学会会员，主要研究方向为分布式计算、软件工程、数据挖掘等。

尹刚（1975– ），男，博士，绿色计算产业联盟实践教学工作委员会副主任，中国计算机学会会员，主要研究方向为在线教育、分布式计算、软件工程、数据挖掘、云计算等。

余跃（1988– ），男，博士，国防科技大学计算机学院副研究员，中国计算机学会会员，主要研究方向为数据挖掘、实证软件工程、社交化编码等。



黄井泉（1986– ），男，湖南智擎科技有限公司高级工程师，主要研究方向为在线教育、软件工程、数据挖掘等。

收稿日期： 2020-10-20
通信作者： 王涛，taowang2005@nudt.edu.cn
基金项目： 国家重点研发计划基金资助项目（No.2016YFB1000800）
Foundation Item: The National Key Research and Development Program of China (No.2016YFB1000800)