基于大数据的软件智能化开发方法与环境

谢冰[1]，彭鑫[2,3]，尹刚[4,5]，李宣东[6]，魏峻[7,8]，孙海龙[9,10]
1. 北京大学信息科学技术学院，北京 100871；2. 复旦大学计算机科学技术学院，上海 200438；
3. 上海市数据科学重点实验室，上海 200438；4. 绿色计算产业联盟，北京 100036；
5. 湖南智擎科技有限公司，湖南 长沙 410073；
6. 计算机软件新技术国家重点实验室（南京大学），江苏 南京 210023；
7. 中国科学院大学，北京 100190；8. 中国科学院软件研究所，北京 100190；
9. 软件开发环境国家重点实验室（北京航空航天大学），北京 100191；
10. 北京航空航天大学计算机学院，北京 100191

摘要
阐述了围绕软件工程大数据的汇聚组织、知识表示提炼、软件工具智能化和智能开发服务环境等关键技术开展的一系列研究工作，建立了基于大数据的软件智能化开发技术体系，研发关键性的软件智能化开发工具，形成了“人-工具-数据”融合的新一代软件智能化开发环境，并构建了软件智能化开发云平台。面向万众创新的社会需求，构建了服务大众的公共服务平台；针对企业创新能力的提升，提供了智能化的企业软件开发环境。

关键词
软件复用；大数据；智能化软件开发；知识图谱；推荐

中图分类号：TP311 文献标识码：A doi: 10.11959/j.issn.2096-0271.2021001

Big data based intelligent software development methodology and environment

XIE Bing[1], PENG Xin[2,3], YIN Gang[4,5], LI Xuandong[6], WEI Jun[7,8], SUN Hailong[9,10]
1. School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China
2. School of Computer Science, Fudan University, Shanghai 200438, China
3. Shanghai Key Laboratory of Data Science, Shanghai 200438, China
4. Green Computing Consortium, Beijing 100036, China
5. Hunan Intelligent Engine Technology Co., Ltd., Changsha 410073, China
6. State Key Laboratory for Novel Software Technology at Nanjing University, Nanjing 210023, China
7. University of Chinese Academy of Sciences, Beijing 100190, China
8. Institute of Software, Chinese Academy of Sciences, Beijing 100190, China
9. State Key Laboratory for Software Development Environment(Beihang University), Beijing 100191, China
10. School of Computer Science and Engineering, Beihang University, Beijing 100191, China

Abstract

A series of researches were conducted on the collection and organization of software engineering big data, software development knowledge representation and extraction, intelligent software development tools and service platforms. The purpose is to establish big data based intelligent software development technique systems, develop intelligent software development supporting tools, and form the next-generation intelligent software development environment and cloud-based platforms incorporating human, tools, and data. The outcome of the project includes a public service platform for the widespread innovation of the people and a series of intelligent software development environments for enterprises.

Key words

software reuse, big data, intelligent software development, knowledge graph, recommendation

1 引言

以开源软件为代表的互联网软件开发具有边界开放、群体分散、交付频繁、知识复杂等特征。与此同时，企业软件开发也逐渐转向以开发运维一体化（DevOps）为特征的云化开发平台。这些网络化开发方式产生了包含源代码、缺陷报告、版本历史、测试用例等在内的全生命周期数据[1]。例如，开源软件社区GitHub已经集聚了超过2.7亿个软件项目，软件开发问答网站Stack Overflow已经积累了超过1 700万个与软件开发相关的问题。这些数据中蕴含的规律可以通过统计和机器学习等技术进行吸收和泛化，用于构造各种智能化的软件工程工具[2]。

智能化软件开发一直是软件工程追求的核心目标之一。学术界著名的以软件开发智能化为核心主题的自动化软件工程（ASE）会议始于20世纪80年代。近年来，在软件工程领域顶级会议ICSE、FSE等中出现了越来越多基于数据、知识驱动的开发智能化技术研究。例如，于2001年发起的软件仓库挖掘会议（MSR）已经得到了广泛关注，并开启了一个重要的软件工程研究子领域。2010年，Robillard M P等人[3]综述了软件工程中的智能推荐系统，指出这些系统在大范围的软件开发活动（如代码复用、软件维护等）中显著地提升了软件开发者的工作效率与质量。

软件开发中的知识获取与应用一直是产业界关注的焦点之一。软件开发问答网站Stack Overflow利用专家回答的群智机制，提供了大量软件开发问题的答案。Eclipse和Visual Studio等集成开发环境（integrated development environment，IDE）都提供了代码自动补全功能。此后流行的IntelliJ IDEA则将智能编码支持作为特色，提供了智能化代码规范检查、自动生成Java规范的基础方法框架、自动补充方法或类代码框架等智能推荐支持。近年来，Eclipse和Visual Studio都在云开发平台方面取得了突破，在云端可以汇集大量开发数据，为更高层次的基于大数据的开发智能化提供了基础平台。

国内主要软件工程研究团队在此方面也开展了大量的研究工作。北京大学2012年提出了知识驱动的软件复用方法；南京大学在基于数据的软件分析、测试方面进行了算法、工具和实践研究；中国科学院软件研究所基于云平台和数据分析技术，在软件运行时测试和演化方面开展研究；北

京航空航天大学在开源软件数据的基础上研究了开发人员推荐问题；国防科技大学在开源数据收集和知识获取方面进行了大量的工作，维护并运行了Trustie社区和网络群体化软件开发环境。国内的浪潮通用软件有限公司、金蝶软件（中国）有限公司等也在获取开发人员操作、数据等方面研发了相应的工具环境；CSDN和OSCHINA等在软件开发技术论坛、代码托管和软件资源汇聚方面建立了大规模的社区。

当前，软件智能化开发成为热点的关键原因在于新时代带来的技术发展的新环境：开源及企业软件开发产生了大数据源，机器学习和信息检索技术的发展提供了知识获取的核心支撑，企业领域工程的广泛实践积累了大量的领域资源。然而，作为智能化软件开发基础的软件开发数据具有规模巨大、碎片分散、快速膨胀的特点。在此基础上实现智能化软件开发支持仍然需要解决一系列基础性的数据采集分析以及知识抽取利用等方面的问题，并以智能推荐、问答等方式提升软件开发工具的智能程度，提高软件开发的质量和效率。在此基础上，智能化的软件工具可以基于数据和知识向开发人员提供推荐、检索和问答等方面的智能化支持。

围绕相关方面，学术界和工业界已经开展了大量的技术研究和实践探索。然而，从总体方法论和技术体系来看，目前的研究和实践探索仍然局限于特定的技术关注点，使用的数据都是针对特定问题本身进行采集的，缺少大数据环境支撑的跨领域智能化技术研究，也没有形成完善的技术体系和环境。为此，笔者团队在国家重点研发计划项目“基于大数据的软件智能开发方法和环境”的支持下，围绕软件工程大数据的汇聚组织、知识表示提炼、软件工具智能化和智能开发服务环境等关键技术开展研究工作，建立基于大数据的软件智能化开发技术体系，研发关键性的软件智能化开发工具，形成“人–工具–数据”融合的新一代软件智能化开发技术体系和环境，并构建软件智能化开发云平台。本项目构建的基于大数据的软件智能化开发方法和环境面向万众创新的社会需求，运行服务大众的公共服务平台；针对企业创新能力提升，提供智能化的企业软件开发环境。

本文将从系统架构、核心技术、应用效果3个方面介绍基于大数据的软件智能化开发方法与环境。

2 系统架构

基于大数据的软件智能化开发方法与环境整体技术架构如**图1**所示。整个方法体系和环境以开源及企业软件项目代码仓库、交付制品、部署和运维监控等多种类型的软件开发数据源为基础，包含软件大数据汇聚及知识提炼、软件智能化开发支持、软件智能化开发服务3个层次。其中，软件大数据汇聚及知识提炼通过自动化的方法采集和汇聚各种类型的软件开发数据，形成自生长的多源异构软件大数据环境，在此基础上，以知识图谱、经验案例、分类器、规则、模板等多种形式提炼和抽取各种软件开发知识。软件智能化开发支持从软件构造、测试验证、群体协作、运维演化4个重要的方面构建相应的工具平台和支撑环境，为相应的软件开发活动提供智能化支持。软件智能化开发服务基于以软件仓库为中心的分布式智能化开发环境集成技术，构建软件智能化开发云环境运行体系结构与集成框架，实现高可扩展的智能开发环境集成与部署，从而建立面向公众和企业的软件智能化开发

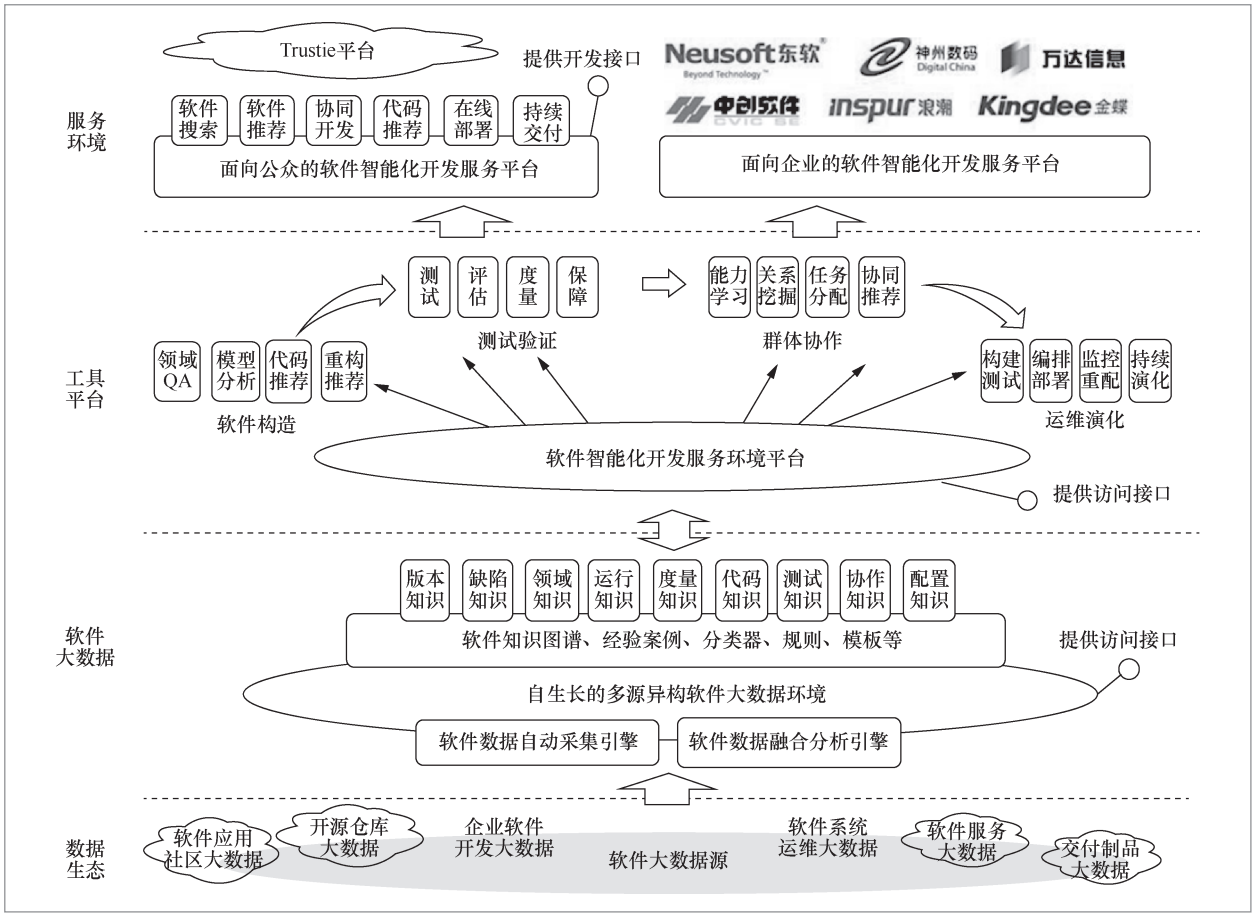

图1 基于大数据的软件智能化开发方法和环境整体架构

服务平台。

(1)软件大数据汇聚及知识提炼

基于主动感知、定向采集、多源关联、增量检测等技术，构建软件工程大数据处理体系结构与支撑系统，形成自生长的多源异构软件大数据环境。建立软件大数据的数据分类和数据汇聚、收集和整理技术体系，研发了相应的采集、存储和服务平台，原始数据、处理后数据以及元数据等不同类型的数据涵盖开发、交付、应用等不同阶段。在此基础上，利用自然语言处理、深度学习、数据挖掘、优化搜索等智能化技术，建立软件开发智能推荐技术研究体系，基于源数据提炼知识图谱、代码模式、主题模型等核心软件知识，形成一批智能推荐、问答技术与工具。

(2)软件智能化开发支持

围绕软件开发中的软件构造、测试验证、群体协作、运维演化4个重要方面，分别形成相应的智能化工具体系，提供数据驱动的智能化推荐和优化技术。

● 在软件构造方面，构建以代码库为核心的软件构造大数据环境以及相应的软件构造知识分析和提炼方法，提供软件构造智能问答、软件开发知识图谱可视化、代码生成与补全、自动化重构推荐等智能化软件构造支持。

● 在测试验证方面，采用机器学习、启发式搜索、自然语言理解等智能化途径，面向测试用例生成、代码模型检验、静态分析缺陷警报确认、程序缺陷定位和修复等软件质量保障的多个方面提供复杂软件

测试与验证智能化支持。

● 在群体协作方面，基于软件大数据的收集与分析构建了软件开发者知识库，形成基于多源软件大数据的开发者知识库体系结构，提供基于学习曲线的开发者能力动态刻画方法和跨社区的开发者画像等智能化支持，同时构建了大规模开发者智能协作支撑环境。

● 在运维演化方面，汇聚了以 Docker 镜像为代表的大规模开发运维一体化数据，形成了自生长、可追溯的领域数据集合。通过智能化持续集成与持续部署流水线系统等一系列工具系统，形成了面向开发运维一体化的运行演化智能支撑环境，提升了开发运维一体化过程的动态调节能力。

(3)软件智能化开发服务

在软件智能化开发技术与工具的基础上，通过以软件仓库为中心的分布式智能化开发环境集成技术，构建软件智能化开发云环境运行体系结构与集成框架，实现高可扩展的智能开发环境集成与部署，建立面向公众和企业的软件智能化开发服务平台。目前已经基于Trustie平台，通过提升改进形成了软件智能化开发服务环境平台IntelligentDE，同时基于Eclipse Che架构实现了智能化推荐工具的整体集成。最终建立的软件智能化开发服务平台面向公众提供网络化的智能化开发服务，同时面向企业提供私有化部署的智能化开发支持。

3 核心技术

基于大数据的软件智能化开发方法与环境包括7个方面的核心技术：软件大数据汇聚、软件知识提炼、智能化软件构造、智能化测试验证、智能化协作、智能化运维演化、智能化开发服务环境。

3.1 软件大数据汇聚

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。通过对当前软件工程领域的数据进行分析，从数据类型、数据格式、数据用途和所属的软件生命周期阶段等多个方面进行归纳，建立了综合互联的软件工程大数据分类体系（如图2所示），以支持多维度、多谱系、贯通性的软件知识提炼和智能释放。该体系包括开发数据、交付数据和应用数据三大类，分别又细分为多个子类，并将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。

围绕软件工程大数据分类体系，面向智能开发服务建立了贯穿数据源、数据存储、数据处理与数据服务的全链条软件大数据框架，实现对海量软件工程数据的采集、分析和应用等全链条管理。整个软件工程大数据管理框架与环境包括数据源、数据存储、数据处理与数据实例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。通过相应的爬虫可以实现对这些多源异构数据的实时、增量抓取和汇聚。

● 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，如软件知识图谱通过数据解析、融合等技术进行数据的二次加工和处理；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。

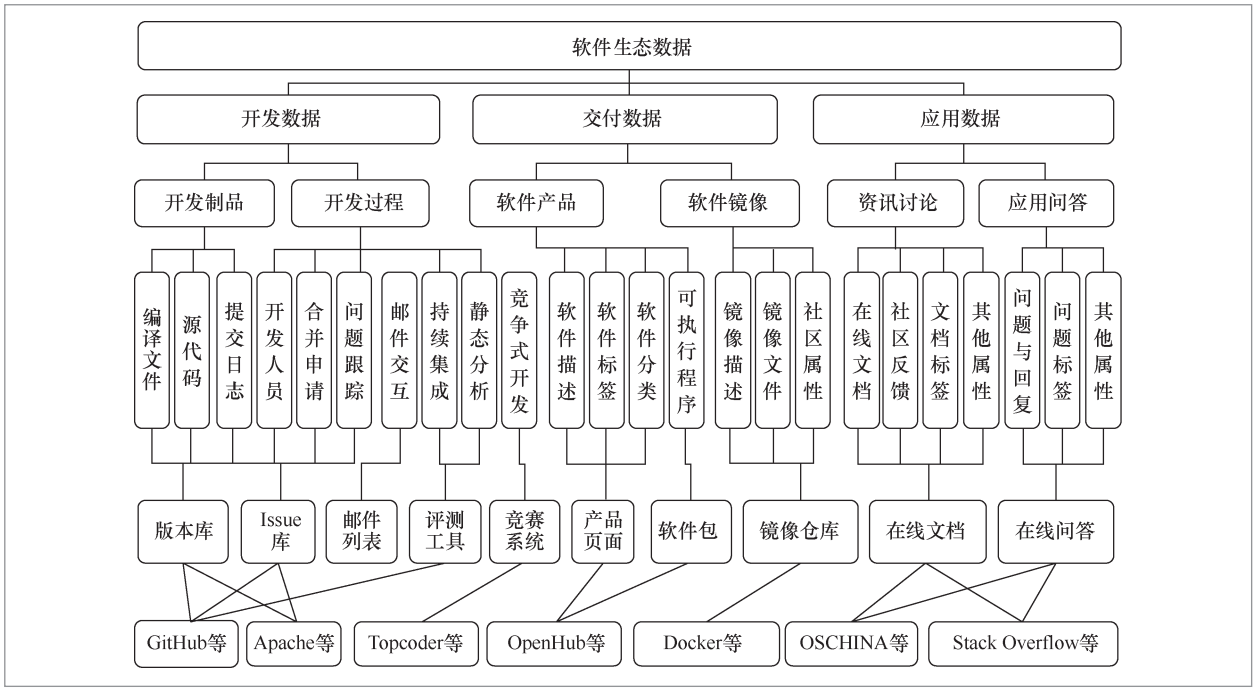

图 2 软件工程大数据分类体系

● 数据实例：通过丰富的接口和服务，针对不同的需求与应用提供相应的数据服务，针对不同类型的数据，对外提供不同的数据服务，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、以运维为目标的数据服务等。

在此基础上，针对网页数据、版本库数据、缺陷库数据等，提出了主动感知、定向采集、多源关联和增量检测等一系列关键技术，设计构建如**图3**所示的分布式数据采集处理框架，部署了分布式爬虫，实现了基于网页爬虫和应用程序接口（application programming interface，API）的数据获取、数据包下载等多种收集方式。

其主要特点如下：

● 构建了大规模多类型的分布式爬虫，实现对全球数十个开源社区的代码仓库、开发历史及软件知识等类型数据的主动感知、定向采集和增量更新；

● 对于异构的代码数据和代码历史提交等数据，基于Git版本管理工具实现了差异分析和增量拉取；

● 对于软件知识数据，通过对不同网站帖子的历史浏览等数据进行分析，形成帖子的关注度衰减模型，从而基于该模型确定对每个网站帖子的更新抓取频率。

采用实时监测、元数据本地存储、按需获取与构建的模式，在兼顾可控性和效费比的情况下，针对不同类型数据的软件数据进行汇聚、收集和整理，实现对全球PB级开源数据（包括原始数据、处理后数据以及元数据）的自主掌控。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，部署于UCloud和阿里云等公有云，并提供统一的汇聚数据说明和访问入口。在此基础上，建立了如下3个软件工程大数据获取与服务平台。

（1）全球开源软件检索与分析平台OSSEAN

针对软件问答社区、开源软件项目、软

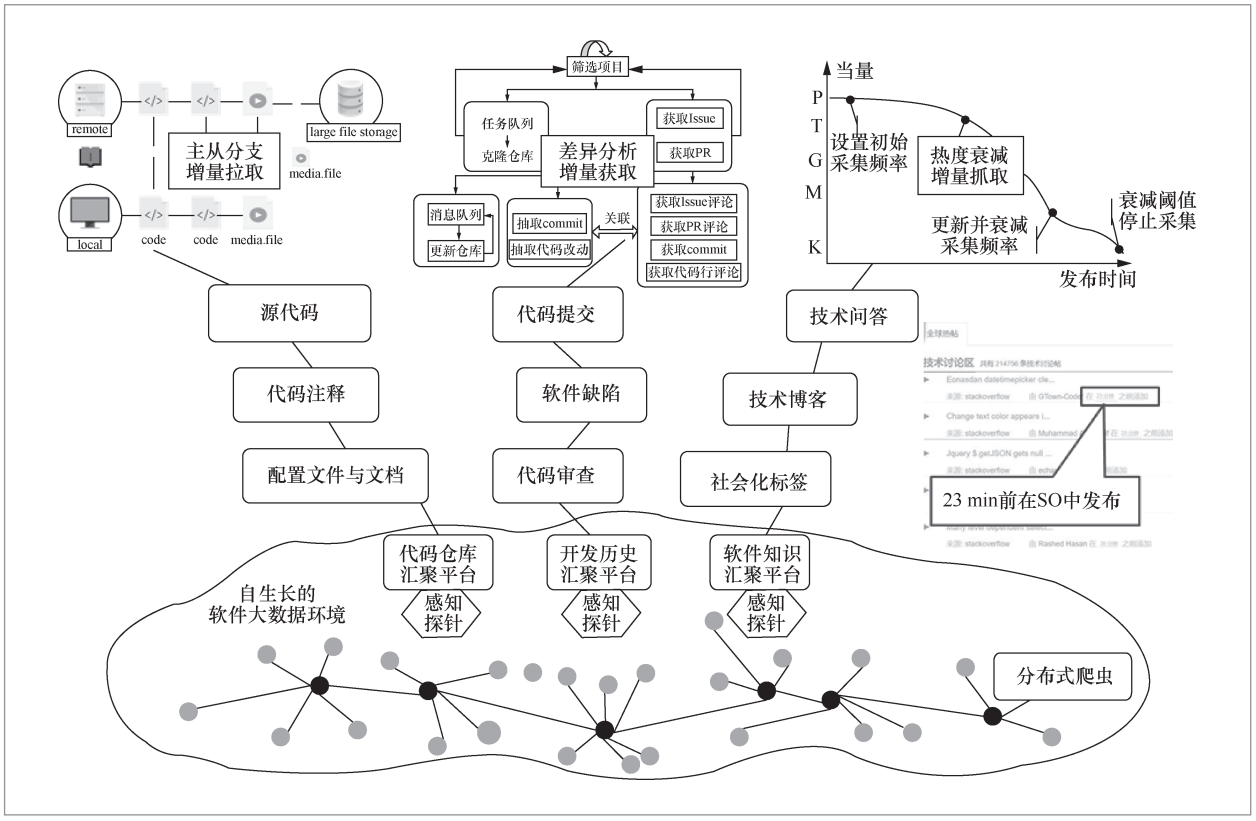

图 3 分布式数据采集处理框架

件开发工具、开发者等类型的数据进行了汇聚和收集整理，构建了面向全球开源软件的检索与分析平台OSSEAN。其包括3层：数据获取层、数据分析层、数据展示层。其中，数据获取层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，同时通过数据挖掘对抽取的数据进行分析（例如社区关联、软件评估等）；数据展示层对得到的数据处理结果按照平台展示的数据格式进行处理，并将处理结果存放于数据缓冲池中，为平台的展示提供数据支持。目前，OSSEAN的数据获取模块已覆盖全球20多个主要开源社区，通过持续监控，实时抓取了超过 1 400万个开源项目/仓库元数据以及超过2 000万条在线讨论数据，同时提供了开源软件分析、检索、排序以及热点话题分析等服务。

（2）软件工程大数据共享平台SBD

依托Trustie平台，采用开放共享、分散存储、平台汇聚、按需获取的方式，设计和制定了多源异构的统一的大数据门户，形成大规模软件工程大数据共享平台SBD。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，平台部署于UCloud和阿里云，提供汇聚数据说明及访问入口。目前，平台通过原始数据、处理后数据以及元数据等形式实现了对涵盖开发、交付、应用等不同阶段的软件工程数据的跟踪、获取，为软件工程研究、智能化开发工具等提供不同类型的数据和服务。

（3）软件交付制品获取与管理平台RAISE

针对Docker镜像设计了一种海量数据汇聚、管理、知识抽取[4]和质量评价的系统化方案和服务，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数据的自动获取与增量更新，实现了数据的可发现和可追踪。研发了面向软件交付制品（Docker镜像）的数据获取与在线管理服务平台RAISE，以提高制品质量、制品统一管理和实现高效复用为目标，提供多维度的Docker制品信息统计与可视化展示，支持Docker镜像的分析、检索、排行、评价和修复推荐等。

3.2 软件知识提炼

为了实现智能化软件开发目标，需要通过多种手段从这些数据中提炼软件知识，形成智能化软件开发支撑能力。针对这一目标，以机器学习、知识图谱、数据挖掘、信息可视化等智能化技术为基础，以检索、推荐、问答、查错等服务形式为呈现方式，形成了一系列软件知识提炼与应用方法体系。其中，基于机器学习和知识图谱这两种主要的智能化分析技术形成的软件开发知识提炼方法与技术体系分别如**图4**和**图5**所示。

基于机器学习（含深度学习）的智能化分析以源代码、软件开发库、项目开发文档、众包开发网站、众包问答、API文档等软件开发数据为基础，通过训练数据准备和模型训练形成具备智能化推荐能力的服务形态。其中训练数据准备和模型选择是关键。训练数据准备阶段的目的是从原始的软件开发数据中抽取符合智能化推荐能力需要的训练样本，例如针对API使用代码推荐的API/控制单元与代码上下文的对照关系、针对缺陷修复分派推荐的缺陷报告与修复者及所属模块的对照关系等。为此，本项目通过程序分析、信息抽取和过滤等多种分析方法，从原始软件开发数据中抽取相关信息，并形成包含所需对照关系的训练样本。在模型选择上，本项目广泛使用了先进的深度学习技术，包括面向API使用代码推荐的Tree-LSTM模型、面向缺陷

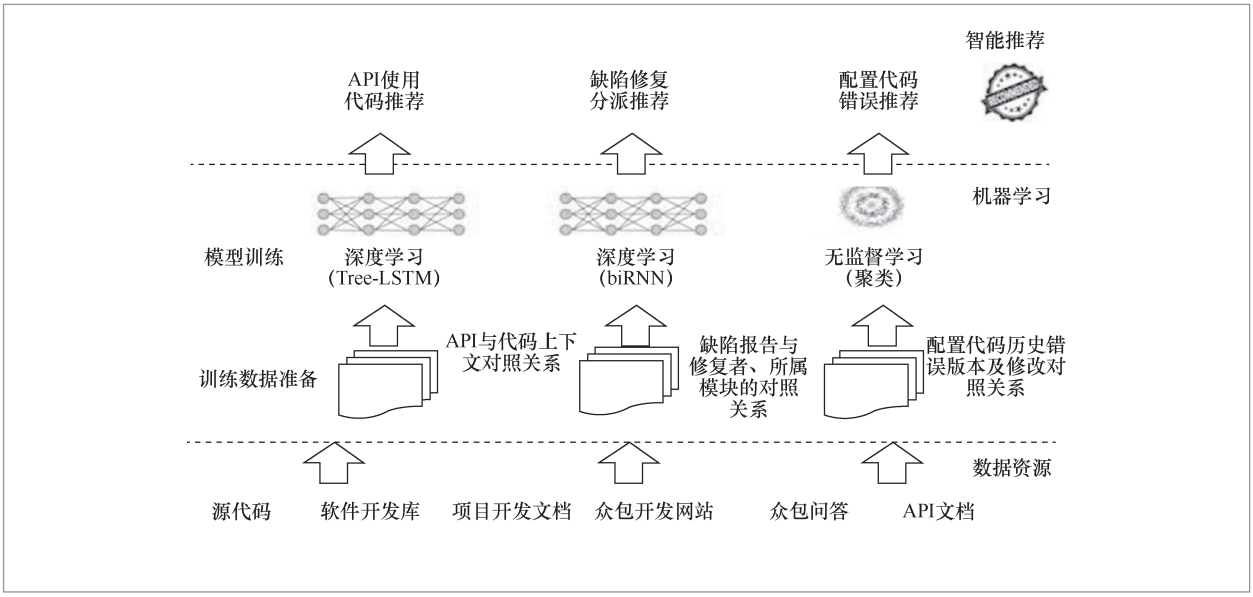

图 4 基于机器学习的软件开发知识提炼方法与技术体系

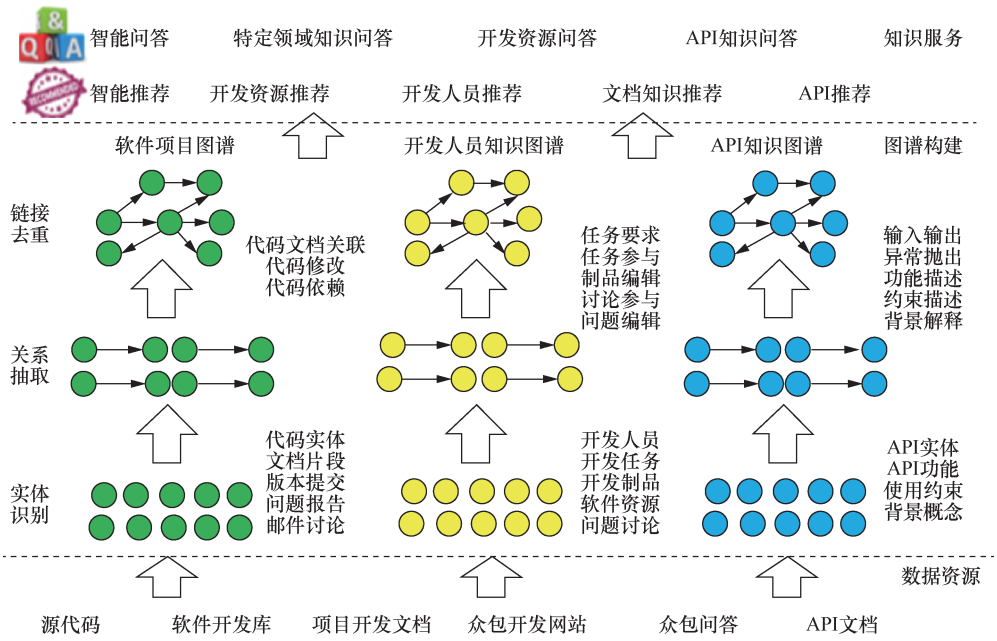

图 5 基于知识图谱的软件开发知识提炼方法与技术体系

修复分派推荐的biRNN模型等。此外，本项目也使用了基于聚类的无监督学习等传统技术，以适应不同的数据规模和应用场景。

软件是人类思维的抽象，反映着人们对客观世界的认识。软件开发是高度知识化的过程。软件代码、文档、数据等都体现着人们的认知知识。这些知识具有结构化和普遍关联的特点，同时广泛分布在不同的软件数据中。作为描述知识的一种基础性表述方式，知识图谱能够显式地定义基于概念实体间关联关系建立起来的知识体系。大部分人类知识体系采用专家编撰（如领域知识图谱）或是群智编写（如Google Knowledge）的方式。软件领域相关知识若采用群智或专家编写方式，会存在工作量大、质量难以保证的问题。本项目确定了基于多源异构软件数据自动获取、整理相应软件知识图谱的技术目标，并在利用知识图谱进行推理、匹配等方面展开研究，有效获取了软件知识，为一系列相关智能化支持功能提供了支撑。

基于知识图谱的软件开发知识提炼方法与技术体系具体包括软件项目图谱和API知识图谱等。其中，软件项目图谱自动构造与问答系统SnowGraph（software knowledge graph）[5]采取了一种以源代码为核心、多种信息抽取方法相结合的方式实现软件知识图谱的自动构造，并基于插件机制支持面向新数据源和新知识的扩展。与其他主流软件知识图谱进行对比，SnowGraph在以下几个方面具有优势：以源代码为核心，采取多种信息抽取的方法协同进行知识图谱的全自动化构造；采用插件机制，使得知识图谱对其他数据源的软件相关数据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；数据来源多样、广泛关联、语义丰富，实

体类型和关系类型丰富，在对应的软件知识表示和知识利用上具备更好的效果。

SnowGraph还实现了基于知识图谱的问答服务[5]。针对事实型问题，SnowGraph提出了一套基于推理子图的自然语言-形式化查询转化方法；对于复杂的自然语言提问，SnowGraph提出了一套基于知识图谱的语义匹配机制，经过Stack Overflow实际数据集的检验可知，其比传统信息检索技术更准确。值得一提的是，通过分析开源软件API之间的语义关联，并构造开源软件项目知识图谱，项目提出了问答文档中关键代码元素与用户查询之间的相似度匹配方法，实验表明，该方法在软件问答文档的检索准确性上超过了Stack Overflow自身的检索工具，能够帮助开发者更快速地定位到相关问答内容。

SnowGraph的系统框架如**图6**所示。该系统遵循数据-信息-知识-智慧（data-information-knowledge-wisdom，DIKW）的层次体系：对于一个待复用的软件项目，以自动化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之中，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。

3.3 智能化软件构造

智能化软件构造主要面向编码、代码评审、代码重构这3种主要的软件构造活动，分别提供智能化代码辅助完成、代码变更差异分析和智能化自动重构功能。

（1）智能化代码辅助完成

软件开发人员经常依赖于各种通用

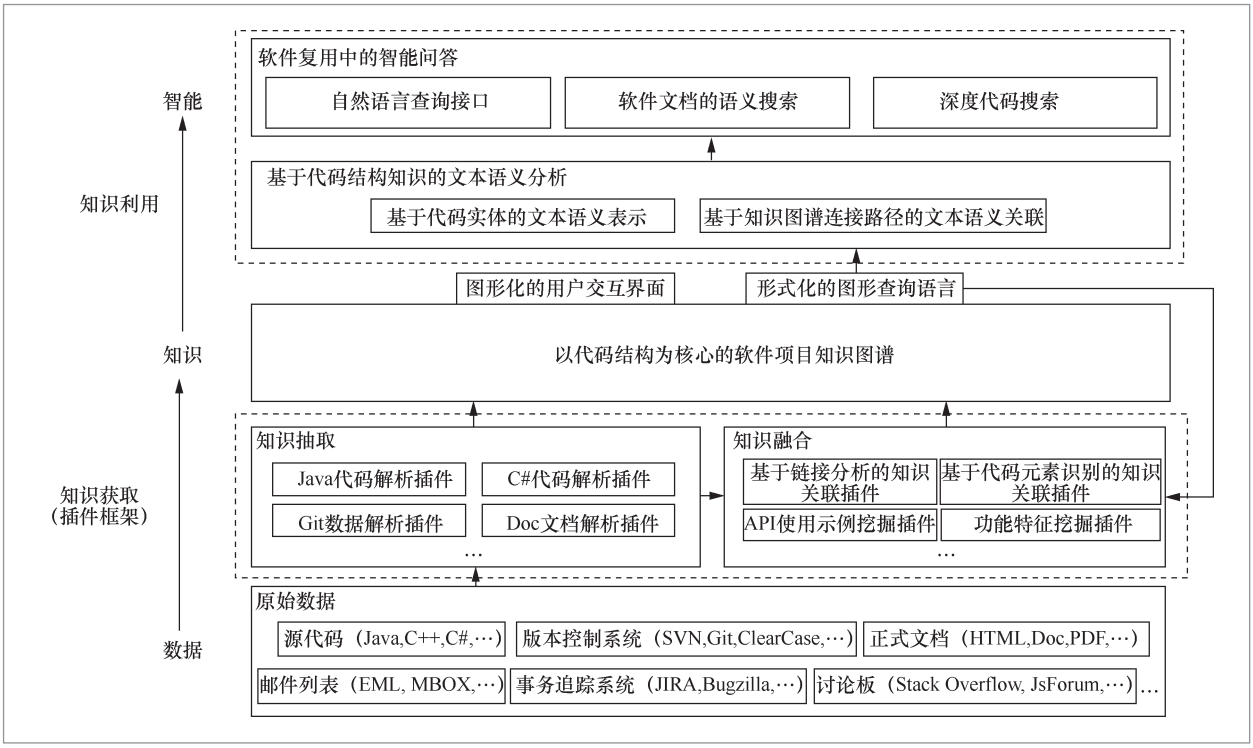

图 6 软件项目知识图谱自动构造支持平台的系统框架

API（例如JDK、Android中的API）来完成开发任务。但是这类API数量众多，开发人员难以熟知所有API的功能及其使用场合。如何在具体的代码上下文中选择合适的API，并按照正确的方式使用，是开发人员经常遇到的难题。为此，项目团队提出了一种基于上下文分析及深度学习的代码生成式补全方法[6]，如**图7**所示。该方法采用Tree-LSTM模型，并采用一种基于抽象代码树结构的代码表示。其中，结点表示抽象的API使用语句、变量声明、赋值语句或控制结构，边表示它们之间的控制流关系。

该方法的预测模型训练主要包括语句模型训练和参数模型构建两部分。语句模型训练以训练数据源代码中的方法为单位构造训练样本，主要思路是在代码的不同位置上移除一部分代码，得到由代码上下文与待预测代码构成的训练样本。在此基础上，将所有训练样本抽象树结构表示中的每一个结点所表示的抽象API映射为向量，然后将所有训练样本输入Tree-LSTM进行训练，得到语句模型。参数模型训练基于训练数据中的代码抽象树结构中的数据依赖路径进行统计分析，计算每个结点与候选API推荐结点产生数据依赖的概率。概率越高，其结点表示的变量越有可能成为候选API推荐中的参数。

该方法的代码推荐流程包括以下步骤：

- 用户输入包含待完成代码的程序；
- 根据用户的输入，运行语句模型和参数模型，给出API推荐结果；
- 用户根据API推荐结果进行选择；
- 根据用户的选择更新当前用户输入的程序。

（2）代码变更差异分析

在代码评审、代码合并以及回归测试过程中，开发人员需要花费大量的时间来理解代码变更，代码变更差异分析是一种常用的辅助理解手段。针对现有的基于文本和语法树的方法的不足，项目团队提出了一个新的基于抽象语法树的代码差异分析方法ClDIFF[7]。该方法在传统的基于抽象语法树的代码差异分析结果上进行抽象和总结，大大减少了基于抽象语法树差异的编辑序列大小，同时可以抽取高层代码差异及其之间的关联关系。

ClDIFF的设计框架中主要包含3个步骤，如**图8**所示。首先是“预处理”步骤，即将变更前后的代码源文件转换成抽象语法树，并通过哈希值比较，去除没有发生变化的声明结点，从而提高后续代码差异分析的效率；然后是“简洁代码差异的生成”步

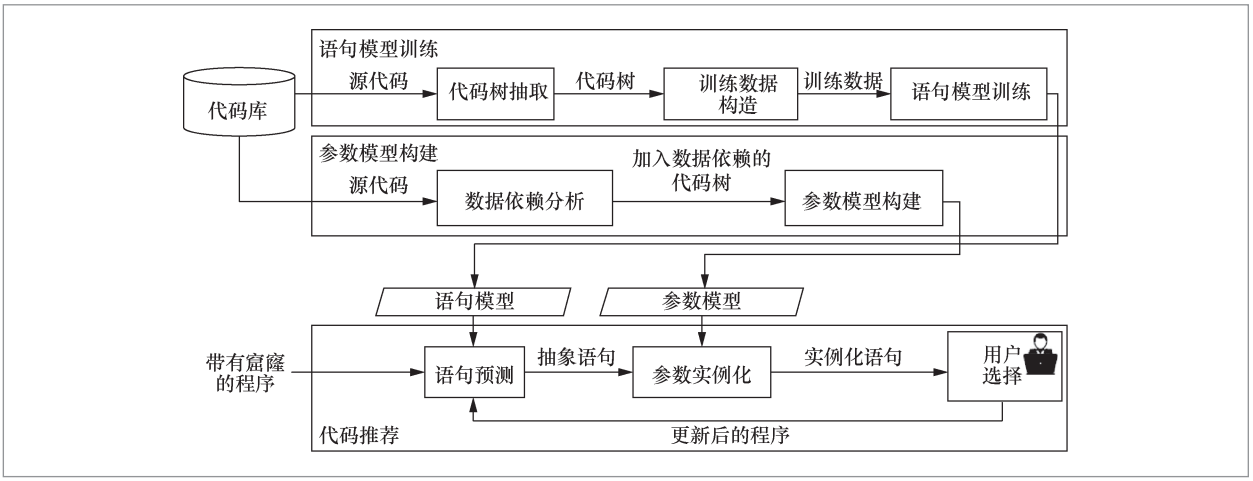

图 7　基于深度学习的代码生成式补全方法 [6]

骤，即利用传统的基于抽象语法树的代码差异分析方法生成细粒度的代码差异（即抽象语法树结点的编辑操作序列），并对序列中的编辑操作在语句或者声明级别上进行聚合和归纳，从而生成更简洁的代码差异（例如，增加一个条件判断的代码块）；最后是“简洁代码差异的关联”步骤，即利用启发式方法在生成的代码差异之间建立5种预定义的关联关系（例如，3个新增的条件判断代码块是一样的）。

（3）智能化自动重构

软件重构是提高软件质量的一种重要技术手段。它在不改变软件外部行文特性的情况下，通过优化、调整软件的内部结构来提高软件的质量，尤其是软件的可读性和可扩展性。目前IDE对软件重构的自动化支持仅限于重构的执行环节，对于重构机会的发掘与重构方案的推荐则没有强力的工具支撑。在实际软件构造的过程中，经常出现实现代码与理想高层设计不一致的问题，这就需要结合业务逻辑和设计知识给出理想的设计。因此，项目团队提出了一种基于高层设计及搜索算法的自动化重构方法[8]，并开发了高层设计驱动的软件自动化重构工具。针对在长期演化过程中发生设计质量退化的复杂软件系统，该方法能够通过高层设计与代码的自动映射提供交互式的重构建议，及时识别设计演进过程中的腐化苗头，从而提高软件系统的设计质量，保障设计的平滑演进。

3.4 智能化测试验证

智能化测试验证的目标是基于软件开发和维护过程形成的各类软件工程数据和机器学习等智能化途径，突破软件测试和验证技术面临的障碍，控制相关问题的复杂性，提高相关技术与工具的有效性和可扩展性。

基于程序运行信息、缺陷报告、编程经验交流信息、静态分析警报、技术文档、历史修复记录、用户评价等软件开发和演化过程中产生的数据，采用机器学习、启发式搜索、自然语言理解等智能化途径，围绕测试、验证、分析、调试和修复等软件质量保障环节，针对测试用例生成、代码模型检验、静态分析警报确认、程序缺陷定位和修复信息推荐中的难点问题展开研究工作[9-11]，具体包括：

- 在测试用例生成方面，针对现有技术难以处理非线性运算、浮点运算、第三方函数调用等程序复杂特征的难点，提出基于智能搜索的复杂软件测试用例生成技术；
- 在代码模型检验方面，针对现有技术难以应对源代码验证的规模和复杂性问题，提出基于不可行路径分析与学习的智能化有界模型检验技术；
- 在静态分析警报确认方面，针对静态分析警报数量非常大、目前基本上靠人工处理的难题，提出静态分析缺陷警报智能化分析与确认技术；

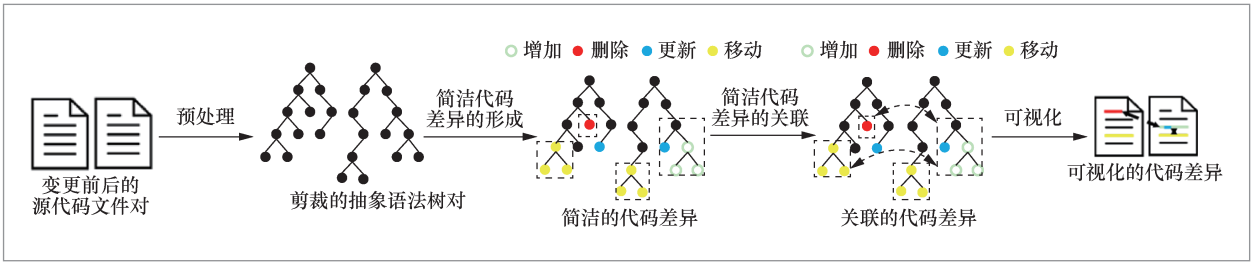

图 8 基于抽象语法树的代码差异分析设计框架

● 在程序缺陷定位方面，针对现有技术的定位精度和效率不高的问题，提出基于缺陷报告与历史修复记录的软件代码缺陷智能定位技术；

● 在程序缺陷修复信息推荐方面，针对现有技术的修复推荐准确度不高的问题，提出基于社区问答网站的软件缺陷修复信息智能推荐技术。

3.5 智能化协作

软件开发作为以开发者为中心的智力活动，开发者协作始终是影响软件开发效率和质量的重要问题之一。基于大数据的软件开发智能协作以提高开发者的协作效率为目标，建立基于多源软件大数据的开发者知识库体系结构以及开发者能力评估和关系挖掘、开发者智能推荐、软件资源智能推荐等智能协作关键技术体系，通过学习开发者的能力特征、挖掘隐含的协作关系，建立以开发者为中心的知识库。在此基础上，根据开发者能力和行为等进行开发者的推荐和智能的任务分配，基于开发者关联进行软件资源（代码、问答等）的智能推荐和重用，从而实现软件的智能协作开发。智能化协作技术体系如图9所示。

（1）数据源

针对基于多源软件大数据的开发者知识库体系结构，实现了面向GitHub、Stack Overflow、Topcoder、CSDN 4个互联网软件平台及东软集团股份有限公司、万达信息股份有限公司等企业内部代码管理平台中开发大数据的分布式增量爬取工具，实现对多源异构的软件开发大数据的获取和存储；实现面向开发者特征和关系挖掘分析的分布式处理架构，支持增量式数据的实时分析处理，构建了包含4 397万名开发者、24亿个结点、80.7亿条协作关系的知识库；研制了面向开发者和开发资源的

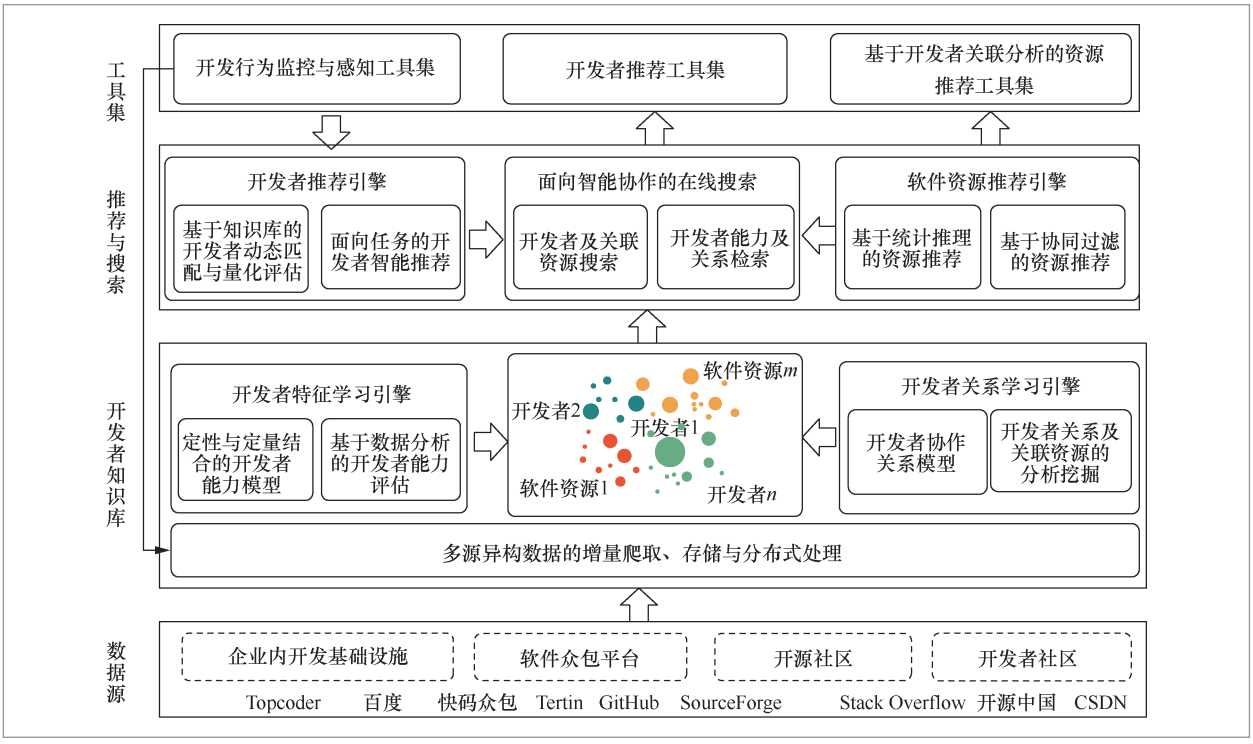

图 9 智能化协作技术体系

高效管理技术和管理工具，实现对开发者和资源关联关系的高效存储、维护、更新及检索。

（2）开发者知识库

针对开发者能力评估和关系挖掘技术，提出了基于学习曲线的开发者能力动态刻画方法[12]、跨社区的开发者画像方法[13]和开发者情感分析方法[14]；建立了定性定量相结合的多维开发者能力模型，从17个维度对开发者的能力进行全面刻画；建立开发者协作关系模型，主要包含开发者-开发者社交关系、直接协作关系和间接协作关系；实现了开发者的能力评估技术和工具，能够实现对开发者能力属性的定性判断或量化计算；实现了协作关系模型研制相应的挖掘分析技术和工具，建立了开发者-开发者与开发者-资源之间的量化关联。

（3）推荐与搜索

针对面向开发任务的开发者智能推荐，提出了基于元学习的众包开发者推荐方法[15]、团队协作能力评估指标以及开发者团队推荐方法[16]；提出了面向GitHub的代码评审者推荐[17]、面向Stack Overflow的问题回答者推荐、面向Topcoder的众包开发者推荐、面向企业GitLab的问题解决者推荐等关键技术，并开发了相应的支撑工具，形成了软件开发者智能推荐工具集，极大地提高了自组织模式下的软件开发效率。

（4）工具集

针对基于开发者关联的软件资源推荐，提出了基于深度神经网络的程序代码向量化表示方法[18]，提高了代码克隆等任务处理的准确性；实现了基于开发者行为偏好与上下文信息的开发资源智能推荐工具集，包含Java编程助手工具和代码API自动生成工具[19]，辅助开发者进行代码、问答、缺陷修复方案等软件资源的高效获取。

基于以上研究工作，项目团队建立了开发者关联分析的智能协作工具集合，初步形成了大规模软件智能协作开发支撑环境iCoOper平台，该平台已在阿里云服务器上稳定运行超过1年。

3.6 智能化运维演化

快速响应变化并持续交付是面向互联网软件（特别是云应用）开发运维追求的主要目标之一。传统软件过程存在软件生命周期“开发部署-运行演化”两阶段割裂，各阶段数据汇聚与知识提炼、关联、运用程度低下的问题，无法实现正向过程自动化与反馈调节智能化相融合的一体化开发与运维，最终影响软件开发运维的效率和质量。

针对上述问题，以软件开发运维中多源异构数据汇聚和领域知识提炼方法为基础，结合多种知识运用技术，提升软件开发与运维有机集成的智能化程度，通过开发运维一体化的软件过程实现软件生命周期各阶段的无缝集成，最终实现正向过程自动化与反馈调节智能化相融合的开发运维一体化智能支撑环境。该环境使用Docker技术和Kubernetes集群管理技术搭建基础设施环境，集成项目研发的多个工具系统，覆盖了数据汇聚管理[20]、知识抽取[4]、运维支撑技术与持续集成（continuous integration，CI）/持续交付（continuous delivery，CD）过程管理3个方面，涉及开发运维一体化中的部署配置[21]、测试评估[22-23]和运行演化等多个软件生命周期阶段，有助于提升开发运维一体化中的智能化程度和过程调节能力。

该环境实现了开发运维从“事件驱动的单向自动化处理模式”向“传统模式与知识驱动的反馈调节模式相结合”的转换。开发运维一体化框架基于“实时分析、闭环调节、在线演化、持续交付”的

体系，实现了知识驱动的自调节开发运维一体化框架，支持大规模云应用的持续交付。相比于已有的技术和研究工作，该框架具有过程灵活定制、执行适应性自动调节的能力。如**图10**所示，其实现途径和机制具体包括：

- 面向CI/CD过程的实时监测与智能分析，形成智能决策方法；
- 基于智能决策的过程适应性调节，形成闭环调节机制；
- 闭环调节驱动构件开发与应用组装的持续演化，支撑从“构件”到“应用”的高效持续交付。

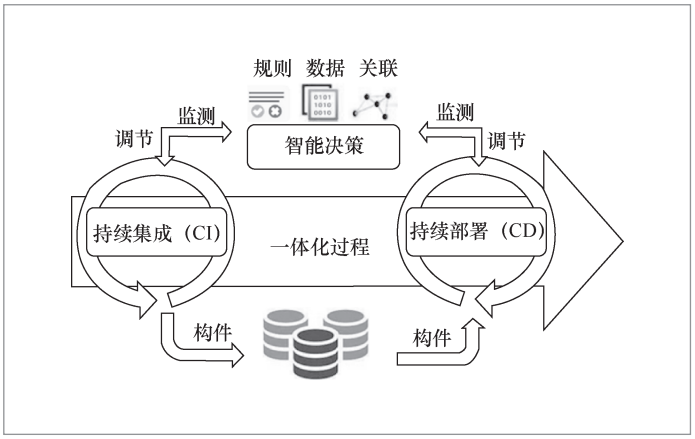

图 10 自调节的开发运维一体化

3.7 智能化开发服务环境

软件智能化开发服务环境突破以软件仓库为中心的分布式智能化开发环境集成技术，构建软件智能化开发云环境运行体系结构与集成框架，实现高可扩展的智能开发环境集成与部署。围绕软件开发制品和活动的完整数据，设计智能化开发环境的集成方法与工具，包括运行环境集成和桌面开发环境的集成机制、基于Eclipse Che的架构设计与交互技术，以及智能化开发工具管理中心，设计实现了以软件项目版本管理为中心，以Trustie平台为基础，基于版本库串联Trustie的协同开发环境、Che在线编程环境以及DevOps运行部署环境，实现按需接入、高可扩展的智能化开发体系架构。依托该架构，协同开发、在线编程和部署运维等智能化工具和服务均可以基于插件模式快速集成，并形成智能化开发环境。

在面向创新实践的软件智能化开发公共服务平台方面，面向创新实践和安全可控需求，突破智能化推荐等关键技术，构建面向开放创新和人才培养的软件智能化开发公共服务平台，支撑我国开源生态的建设。本项目针对开源生态中软件资源和知识资源呈现的多层次、多维度特性，研究了面向协同开发社区的代码贡献者推荐算法、基于多维特征的开源项目个性化推荐方法，以及跨项目的贡献审查者推荐方法等技术。在此基础上，面向国家基础及前沿技术安全可控和自主创新的重大需求，依托智能开发技术与工具，构建形成了开源模式的开放创新服务环境和大规模开放在线实践（massive open online practice，MOOP）模式的人才培养服务环境，围绕大规模开放创新实践和人才培养开展了大规模的应用和服务。

在软件智能化企业开发支撑技术与服务平台方面，研发企业大型软件资源库和领域知识图谱，为企业智能化软件开发提供支撑技术与服务平台。项目汇聚涵盖需求、设计、开发、测试、运行、运维、支持服务等软件全生命周期的数据资源，突破了装备制造和智能仓储领域的知识库构建技术，研发了面向企业管理软件的智能化云开发服务环境。在此基础上，浪潮通用软件有限公司集成项目研究成果搭建的智能化企业开发平台有效提升了现有开发方法和平台的智能化水平。同时，应用项目研发的软件工程教育平台开展企业研发人员的

培训，可有效提升研发人员的开发实践能力，降低企业的运营成本。

4 应用效果

面向公众的软件智能开发服务平台通过衔接高校、科研机构、软件园区、软件企业等利益相关方，降低软件开发门槛、释放软件开发潜力，推动软件驱动的“产学研用”创新创业生态的构建。

在多源异构软件工程大数据的汇聚与管理方面，研究并建立了覆盖“开发–交付–应用”的软件生命全周期的数据分类体系，突破主动感知、定向采集、多源关联、增量检测等关键技术，构建了自生长的软件工程大数据共享平台SBD。SBD以原始数据、处理后数据以及元数据等形式汇聚了全球主流开源社区的开源软件项目、软件问答讨论、软件开发工具、Docker镜像、开发者等不同类型的数据；通过对数据源进行感知与监测，利用元数据分析比对等技术实现对数据变化的感知和追踪，实现持续演化的软件工程大数据的增量式获取，提升对开源软件大数据的掌控能力。目前，平台上可跟踪和获得的数据总量超过1.5 PB，其中集中存储约25 TB，分散存储约217 TB，实时监控数据约1 278 TB。

在数据与工具开源方面，针对Apache开源软件基金会中的192个开源软件项目（共计482.48 GB软件项目数据），分别构建了相应的软件项目知识图谱。以Apache Lucene项目的知识图谱为例，共抽取出了378 897个实体，以及这些实体之间的1 902 683条关联关系；在此基础上，对外提供了软件项目智能问答服务。开发者可以提出自然语言问题，系统基于知识推理在知识图谱以及项目文档中进行搜索，并给出答案。同时，项目将项目团队自主研发的、自主可控的35项智能开发工具与相关软件数据进行了汇聚整理，开源到木兰开源社区。

在国内开源生态建设方面，项目支持GCC建成了面向“开放计算架构+开源软件技术”的中国绿色计算开源技术和产品开发社区，汇聚了国内的企业团队和大众贡献者；基于Trustie构建的中国绿色计算开源技术已经发展成我国最大的ARM开源开发和创新社区，有效支持了产、学、研、用各界的开源开发和评测活动；支持新一代人工智能产业技术创新联盟OpenI启智开源社区的建设，推动人工智能领域开源开放协同创新生态的构建；支持的可控开源创造行动是我国2020年启动的重大创新体系建设计划，核心是激活和汇聚开放群体智慧和贡献，构建一种可持续发展的生态。

在科学研究创新方面，项目有效地帮助了micROS机器人操作系统团队、NuBot国际顶尖机器人竞技团队、Trustie群体化软件开发研究团队等解决了智能化软件开发和持续性质量改进等难题。

在高校实践教学方面，智能化实践学习平台头歌（EduCoder）支撑产教各界开发了超过6.9万个在线计算机训练项目、391万个开源代码仓库，各类师生和开发者超过80万人，提供实践课程3 900余门。此外，项目还参与支撑了3届“全国高校绿色计算创新大赛”，参赛人数总量超过2万人次。

5 结束语

围绕基于大数据的软件智能开发方法和环境，项目团队提出了一套大数据驱动的软件智能化开发方法，涉及软件开发多个主要过程中的智能化支撑技术。项目团队研发了一批软件智能化开发工具原型

系统，在基于知识图谱的软件开发问题复杂查询、数据驱动的测试、智能化群体协作、智能化开发运行一体化决策等方面均提供了基于软件大数据的智能推荐和开发支持。基于对国际开源软件社区级技术的整体分析和研究，项目团队建立了一套互联网及开源软件数据资源的获取汇聚技术和方法，以及融合利用技术方案，目前可以跟踪和获取超过1.5 PB的软件工程数据，分析监测了392万个开源软件，为全球开源领域的4 397万名开发人员建立了画像，提升了我国对此类数据的掌控能力。

在前期国家计划形成的软件资源共享与群智开发平台基础上，项目团队进一步发展了软件开发中的数据智能支撑功能，形成了较为完善的云化开发平台，并对外提供公共服务。目前，开源开发平台注册的各类用户约41.5万人，开源项目1.5万个；开源教育平台汇聚了5.4万个开源训练项目，293万个开源代码仓库，来自982所高校与企业的1.1万名注册教师、33.2万名注册学生和开发者，提供实践课程1 600余门。在6家大型软件开发企业中取得了一批应用示范的成果，显著提高了软件企业的生产效率和质量；形成了一个自主可控的软件开发共享服务的技术框架，并基于此方案支持了一批国内开源社区的支撑环境建设，包括“云计算与大数据”重点专项的集成平台的环境建设。

参考文献:

[1] ZHANG D M, HAN S, DANG Y N, et al. Software analytics in practice[J]. IEEE Software, 2013, 30(5): 30–37.

[2] HINDLE A, BARR E T, GABEL M, et al. On the naturalness of software[J]. Communications of the ACM, 2016, 59(5): 122–131.

[3] ROBILLARD M P, WALKER R J, ZIMMERMANN T. Recommendation systems for software engineering[J]. IEEE Software, 2010, 27(4): 80–86.

[4] 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(5): 957–971.

[5] LIN Z Q, ZOU Y Z, ZHAO J F, et al. Improving software text retrieval using conceptual knowledge in source code[C]//2017 32nd ACM/IEEE International Conference on Automated Software Engineering. Piscataway: IEEE Press, 2017: 123–134.

[6] CHEN C, PENG X, SUN J, et al. Generative API usage code recommendation with parameter concretization[J]. Science China Information Sciences, 2019, 62(9): 51–72.

[7] HUANG K F, CHEN B H, PENG X, et al. ClDiff: generating concise linked code differences[C]//The 33rd ACM/IEEE International Conference on Automated Software Engineering. New York: ACM Press, 2018: 679–690.

[8] LIN Y, PENG X, CAI Y F, et al. Interactive and guided architectural refactoring with search-based recommendation[C]//The 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering. New York: ACM Press, 2016: 535–546.

[9] WANG Y, WANG L Z, YU T T, et al. Automatic detection and validation of race conditions in interrupt-driven embedded software[C]//The 26th ACM SIGSOFT International Conference on Software Testing and Analysis. New York: ACM Press, 2017: 113–124.

[10] 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.

[11] YAO Y, TONG H H, XU F, et al. Scalable algorithms for CQA post voting prediction[J]. IEEE Transactions on Knowledge and Data Engineering, 2017: 1723–1736.

[12] WANG Z Z, SUN H L, FU Y, et al. Recommending crowdsourced software developers in consideration of skill improvement[C]//2017 32nd IEEE/ACM International Conference on Automated Software Engineering. Piscataway: IEEE Press, 2017: 717–722.
[13] YAN J F, SUN H L, WANG X, et al. Profiling developer expertise across software communities with heterogeneous information network analysis[C]// The 10th Asia-Pacific Symposium on Internetware. [S.l.:s.n.], 2018: 1–9.
[14] DING J, SUN H L, WANG X, et al. Entity-level sentiment analysis of issue comments[C]//The 3rd International Workshop on Emotion Awareness in Software Engineering. New York: ACM Press, 2018: 7–13.
[15] YE L T, SUN H L, WANG X, et al. Personalized teammate recommendation for crowdsourced software developers[C]//The 33rd ACM/IEEE International Conference on Automated Software Engineering. Piscataway: IEEE Press, 2018: 808–813.
[16] ZHANG Z Y, SUN H L, ZHANG H Y. Developer recommendation for Topcoder through a meta-learning based policy model[J]. Empirical Software Engineering, 2019, 25(1): 1–31.
[17] XIA Z L, SUN H L, JIANG J, et al. A hybrid approach to code reviewer recommendation with collaborative filtering[C]//2017 6th International Workshop on Software Mining. Piscataway: IEEE Press, 2017: 24–31.
[18] DING J, SUN H L, WANG X, et al. Entity-level sentiment analysis of issue comments[C]//The 3rd International Workshop on Emotion Awareness in Software Engineering. New York: ACM Press, 2018: 7–13.
[19] TIAN Y F, WANG X, SUN H L, et al. Automatically generating API usage patterns from natural language queries[C]//2018 25th Asia-Pacific Software Engineering Conference. Piscataway: IEEE Press, 2018: 59–68.
[20] CHEN W, XU P X, WU G Q, et al. A hierarchical categorization approach for system operation services[C]//The 24th IEEE International Conference on Web Services. Piscataway: IEEE Press, 2017.
[21] CHEN W, WU G Q, WEI J. An approach to identifying error patterns for infrastructure as code[C]//2018 IEEE Symposium on Software Reliability Engineering Workshops. Piscataway: IEEE Press, 2018.
[22] WANG J, DOU W S, GAO Y, et al. A comprehensive study on real world concurrency bugs in node.js[C]//The 32nd IEEE/ACM International Conference on Automated Software Engineering. Piscataway: IEEE Press, 2017.
[23] WANG J, DOU W S, GAO C S, et al. Context-based event trace reduction in client-side JavaScript applications[C]// The 11th IEEE Conference on Software Testing, Validation and Verification. Piscataway: IEEE Press, 2018.

作者简介

谢冰（1970– ），男，博士，北京大学教授、信息科学技术学院常务副院长、软件研究所所长，国家杰出青年科学基金获得者，中国软件行业协会理事，中国计算机学会高级会员，*Chinese Journal of Electronics*编委，入选教育部新世纪优秀人才支持计划、北京市科技新星计划，获得“中创软件人才奖”。主要研究方向为软件工程、计算机理论科学和分布式系统等。

作者简介

彭鑫（1979– ），男，博士，复旦大学教授、计算机科学技术学院副院长、软件学院副院长。中国计算机学会软件工程专业委员会副主任，*Journal of Software*: *Evolution and Process*联合主编，*ACM Transactions on Software Engineering and Methodology*编委，《软件学报》编委，*Empirical Software Engineering*编委，IEEE软件维护与进化国际会议（ICSME）执行委员（2017—2020年）。2016年获得NASAC青年软件创新奖。主要研究方向为软件开发大数据分析、智能化软件开发、云原生与智能化运维、泛在计算软件系统等。

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。

李宣东（1963– ），男，博士，南京大学计算机科学与技术系教授、博士生导师，软件学院院长，主要研究方向为软件建模与分析、软件测试与验证。

魏峻（1970– ），男，博士，中国科学院软件研究所研究员、博士生导师，主要研究方向为软件工程、网络分布式计算等。

孙海龙（1979– ），男，博士，北京航空航天大学计算机学院教授、博士生导师，主要研究方向为智能软件工程、群体智能和分布式系统。

收稿日期： 2020-12-28

通信作者： 谢冰，xiebing@pku.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)