低代码开发在金融系统中的应用研究

■ 中国人民银行宿州市中心支行 裴立公

摘 要：近年来，随着信息化技术的演进以及编码技术的发展，以低投入、高效率为基本目标的低代码运用迅速崛起，成为促进软件代码编程应用、强化手工编码效率、增强业务架构延展能力的重要工具。由于金融系统的数据化演化趋势越来越明显，使得低代码开发成为金融系统进行业务重构、系统架构调整和数据应用建模开发的重要助手，在金融系统中得到大面积的推广应用。基于此，文章对低代码在金融系统的应用进行了深入研究，通过对低代码开发概念的梳理以及对低代码开发中架构元模型在金融系统的应用原则的剖析，确立了低代码在金融系统应用的具体行为准则，并在此基础上提出了金融系统中低代码开发应用的建议。

关键词：低代码；金融系统；应用研究

随着数字化的全面转型，银行等金融系统单位已经逐步转型为数据生产商。面对着大量的瞬时性数据，金融系统急需更加高效、简洁的代码开发工具和新的程序架构，使一线金融工作人员能自主依据系统架构和实际的任务要求调整建模程序，实现数据的再处理，满足业务需求。在这种情况下，低代码开发成为金融系统面对大容量数据时所进行的必然选择。

一、低代码开发的概念及应用

低代码是近年来逐步在代码程序开发和基础架构建设上延展出来的一种新概念。最早，低代码由佛罗斯特咨询公司提出，该公司认为，为了应对更加复杂的数据化世界，迎接即将到来的大数据时代，编程语言及编程代码的架构处理应当逐步形成一种简单化、便利性的操作流程，以帮助数据终端一线工作人员在最简单的层面上实现应用程序功能的再组合、再开发和模块的深度挖掘，以便于使这些建模后的数据模块能够适应一线工作的实际需求。

当前在低代码的开发应用中，基本上可以分为两个应用研究方面。

一是以低代码平台为基准，对原始的程序架构和代码平台进行简单的数据接口调整，从而实现低代码运行环境中的表单设计流程调整、框架组合，满足日常大容量数据的处理需求。例如，腾讯大力推动代码开发的应用，将微信小程序直接导入代码开发运行环境中，实现低代码运行环境中的表单设计流程调整、

作者简介：裴立公（1968-），男，安徽宿州人，工程师，研究方向：金融科技。
收稿日期：2022-06-13

框架组合，以便于小微企业和个人能够单独地在微信小程序中构建独立的低代码开发套件，并且通过这些套件进行自由组合，形成低代码开发的运行环境，最终生成功能相对多样、要素相对齐全的低代码，开发成小程序。这极大地拓展了腾讯在微信生态链上的内容要素搭建，驱动腾讯微信小程序的生态繁荣。

二是以模型为驱动，构建更深层次的模型架构，形成元模型架构，从而实现对具体数据处理流程的重新搭建。相较于前一种，这种低代码的应用更加具有深层次灵活性和开发实践，可以称之为低代码可视化数据建模开发应用。例如，当前一些苹果手机运行服务商和开发提供商也提供了低代码开发模型架构，开发人员只需要进入开发运行环境中独立使用这些分散化的功能模块自由组合搭配，就能形成较为复杂的低代码可视化数据运行程序。

目前市场上较为流行的是第二种可视化建模开发应用。相较于简单的接口驱动，可视化建模开发应用使数据应用更加具有针对性、大容量和高速度的特征。传统的编程开发和程序建模过程是对具体业务流程、数据集合过程、任务分解调配过程的资源组合与要素分析，如构建业务流程图、搭建业务分工图、实施业务结构关系框架建设，而随着低代码系统的全面推行，这些过程将逐步降低进入门槛，使得一线工作人员能够直接参与到具体的数据处理设计中来，实现了“所想即所得”的模块化、自动化处理能力。特别是在低代码发展下实施的可视化数据模块开发应用，可以有效地推动数据建模的平台化应用，一线工作人员可以依据自身的数据需求，实现对不同应用模块的自由设计、自由分拆、自由搭建，从而形成系统化、复杂性的业务架构重组，用以支持自身的工作需求，从而大大提高工作的效率。

二、低代码开发中的架构元模型在金融系统的应用原则

当前在低代码领域的开发应用中，由数据模型驱动而形成的架构元模型能够容纳一切语言程序和编译器，并且能够按照程序指令正确运行环境要素，完成对程序模块的按需组合。大致而言，一个体系结构全面的架构元模型往往是由业务框架、数据库和应用方法3个模块组成。这3个模块相互支撑、互为因果，共同形成了模块驱动型编程语言的架构。架构元模型是低代码开发的核心组成，将其应用在金融系统中，能够有效地驱动金融系统实现金融业务流程优化，促进金融业务效率提升。

本文认为一个相对合格的低代码架构元模型应用在金融系统中应当坚持以下几个原则。

一是坚持金融问题聚焦原则。应当将元的模型开发和应用聚焦到金融系统的核心任务上来，如针对金融系统进行数据报送、账目管理、信贷数据清洗等问题，应当实施单项决策，对每一项任务进行有针对性的元模型框架设计，通过低代码平台开发，可以针对金融系统中的任何一个聚焦性问题实施元模型的组合调整设计，最终完成元模型的应用。例如，建设银行安徽省分行为有效进行分支行数据管理和账目管理，利用低代码开发运行系统建立了基础的数据管理员模型架构。该行要求各地市分支行自行实施低代码应用，通过低代码模块的自由组合来进行每日资金往来报送、账目管理体系的报送管理，对信贷数据进行全面地清洗和分析。通过一系列的模块化自由化操作，各分支行可以依据自身开发金融要素的侧重点进行自主的原模型设计。有的分行侧重于以对公业务为主，实施适用于自身的模块建设；有的分支行侧重于以普惠小额贷款为主，进行数据模块的叠化操作。总之，不同的分支行针对自身的偏向业务进行了自由的模块化组合，服务于自身的金融主业发展。

二是坚持简洁原则。由于低代码开发的核心要义就是满足非专业型编程设计人员的大数据个性化内容处理需求，因此，使用低代码元模型框架进行任务设计和可视化模型搭建时，应当使整个金融数据结

构形成可视化界面。故可视化编程模块应尽可能简洁高效，以最少的步骤完成最多的任务。在金融系统中，核心的低代码可视化模块设计不应超过20层代码搭建，如此才能使整个金融数据的处理得以快速、高效地完成。以苏州市农商行为例，该行根据省联社的低代码开发运行架构，基于自身的个性化内容处理进行了低代码环境的编程设计、模块搭建，使用低代码元素进行可视化的多元模块叠加，搭建了最多具有16个层级的代码架构。通过这种简洁性、少代码层级、多元的代码模块的统一规划，该行基于自身的运行系统，设计了多样性、作用多元的可视化界面，极大地提升了非专业型编程设计人员的业务处理能力，也使其可以基于非专业型的编程设计架构进行多元化的操作，提高了一线工作人员处理金融业务、进行信贷资金追踪的便利性。

三是规则明确原则。在使用低代码中的元模型框架处理金融问题、实现金融数据的可视化设计时，应当使整个设计目标清晰高效，使整个低代码的运行命令单一、准确、直接、清晰，从而有助于对大规模金融数据的快速处理和点对点反馈。例如，广州农商行科技开发部门为了有效地实现各分支行对多元金融环境的便利化处理，使用低代码开发套件进行模块化操作，使各分支行的一线辅助性科技人员能够按照自身的业务侧重点进行搭建，构建畅通的数据管理系统和信息处理系统。因此，该行基于明确的规则设定了自身的低代码原模型运行管理原则，制定13条相互连通、互有支持、安全协作的保障性低代码运行原则，并将这些运行原则生成代码编辑系统来确保代码运行时能够在这一框架内进行。这种有效的处理方式实现了广州农商行各分支行金融数据的可视化设计、可视化输入、可视化操作和可视化运行，同时又确保了低代码运行环境的金融基础安全性，确保了整个低代码运行环境中金融数据能够安全稳定地运行。

三、金融系统中的低代码开发

在金融系统中实施低代码开发，涉及系统架构的重整、业务建模的开展和应用建模的推广。在这些领域，可通过低代码触发架构元模型的自由组合调整，最终根据业务需要设计形成新一代数据管理系统。

一般可以针对金融系统内部实施客户身份识别、信贷数据对比、支付命令环节查询和中间业务调整等。在这些业务中，可以采用架构驱动的模式，以可视化模块来进行数据建模，通过业务需求分析，以可视化建模框架进行低代码工具开发、数据管理，不断调整金融项目及模型的需求，最终达到个性化匹配、精准化建模。

例如，徽商银行针对其系统内部的客户身份识别和信贷数据匹配机制进行了低代码开发，通过对客户身份识别系统进行模块化创建，与信贷数据模块化指令进行匹配，将数据驱动、人物画像等多种模块进行自由搭配，构建了有针对性的的代码运行环境，提升了整个低代码在功能实现、数据配比、数据库维护方面的应用能力，从而形成了各个市级分行能够按照自身金融业务发展特色进行个性化的身份识别、信贷数据匹配的低代码运行架构。这种运行环境使得各个分支行具有相对个性化、精准化的建模能力，大大加深了技术人员对金融系统的理解，有利于各一级分行进行个性化的金融信息收集和金融产品匹配。

按照《银行业务建模指南》，可以对数据核算进行单独的数据建模，通过低代码的应用，使用分布式架构生成代码，并部署到系统上运行，最后完成分布式核心系统的部署与运行，确保金融数据安全、稳健、高效地得到处理。

在信用卡管理环节，同样可以使用可视化建模技术，通过低代码的可视化建模开发，对信用卡的核心参数、账户数据及客户信息进行有效的程序设计与应用。开发前端应用模块，通过配置可视化应用模块来增强用户体验，实现金融系统业务要素与程序要素的深度结合，增强数据架构驱动下的Web设计能力，使

低代码数据架构与信用卡业务建模充分融合，推动金融机构信用卡管理程序优化。

总而言之，将低代码的开发环境应用在金融系统中正当其时。金融数据高度个性化的业务需求与低代码自由化、简单化的建模调整相互搭配，有效地促进了金融系统要素的充分应用。因此，在金融系统中嵌入低代码思想，推动低代码程序的应用开发，促进金融系统工作效率的提升，将会进一步加深低代码运行、开发环境的流畅化，使银行系统的金融管理能力大幅度提高，数据匹配性、安全性和业务开发能力得到有效优化，并强化银行的个性化功能，通过在编程程序和非编程程序之间建立起沟通渠道，推动金融深化和金融的自动化变革。FTT

参考文献：

[1]聂爱华. 互联网思维下的政府治理与服务创新[J]. 中国海洋大学学报(社会科学版)，2021(3):96-103.

[2]郭卫丹，刘宗凡，邱元阳，等. 深入评析低代码开发平台[J]. 中国信息技术教育，2021(11):80-86.

[3]MENDIX. 低代码开发让企业变身“快鱼”[J]. 软件和集成电路，2021(11):60-62.

[4]佚名. 西门子收购TimeSeries，助力客户加快数字化转型步伐[J]. 自动化博览，2021(5):3.

[5]智锦. 基础设施即代码加速金融IT云化转型[J]. 中国金融电脑，2021(z1):84-87.

[6]MENDIX. 企业治理转型助力数字经济和低代码发展[J]. 软件和集成电路，2021(12):12-13.

[7]佟伟. 低代码软件开发平台Mendix落地中国市场[J]. 现代制造，2021(2):16-17.

[8]曹严明，曹蕤. 模型驱动的低代码平台及其在金融业务系统迁移中的应用[J]. 中国金融电脑，2021(8):60-64.

(上接P79)

对地方法人金融机构科技伦理的督促与指导，区别法人金融机构与非法人金融机构科技伦理治理的侧重点，以更贴近实际、更有效的治理指标体系，加强对法人金融机构科技伦理的治理，从源头上加大对各类风险的约束与治理，防范于未然。

（四）加强宣传，增强金融科技伦理认同

当前，金融科技伦理治理属于新的课题，需要加大宣传与引导。一是要常态化开展面向金融机构从业人员的全方位、分众化的金融科技伦理知识宣传，推动金融机构从业者提升金融科技伦理意识，理性对待金融科技伦理问题。二是将金融科技伦理纳入新员工入职培训，从工作之初就树立金融科技伦理道德底线，系好科技伦理第一颗纽扣。三是加强对金融科技人才的伦理意识培养，将金融科技伦理意识、道德底线融入人才培养的全进程，增强其伦理认同感，共同推进金融业高质量发展。FTT

参考文献：

[1]人民银行金融消费权益保护局. 金融消费者权益保护典型案例(2020)[M]. 北京：中国金融出版社，2021.

[2]尹振涛. 金融科技快速发展带来的风险隐患及对策建议[J]. 中国远洋海运，2021(6):24-26.

[3]边万莉. 金融科技面临四大伦理挑战，治理体系如何落实[EB/OL]. http://www.21jingji.com/article/20220418/herald/303fbad2edb6785b20511623388bca4b.html.

[4]肖钢. 加快构建金融科技伦理治理体系[EB/OL]. https://news.cnstock.com/news,yw-202203-4838641.htm.