

基于多智能体的敏捷需求工程过程模型^①



刘芳, 涂键, 曹舒

(东北石油大学 计算机与信息技术学院, 大庆 163318)

通信作者: 刘芳, E-mail: liuf26454@163.com

摘要: 敏捷需求过程模型适用于频繁的需求迭代场景。该方法强调以用户为中心的设计理念, 采用简洁的文本表述方式, 不依赖复杂流程和工具。将需求模型引入敏捷开发流程, 能有效解决对敏捷方法认知不足等问题, 但在频繁的需求迭代场景中, 引入的需求模型往往面临维护困难、版本过时等问题。在频繁进行需求迭代的敏捷开发过程中, 模型的复杂性导致需求模型的手动维护消耗大量资源。为解决这一问题, 本文提出基于多智能体的敏捷需求过程模型 MA-ARP。该模型通过构建多智能体自动处理系统, 利用其推理与识别能力, 根据需求变化动态更新需求模型。该方法能有效降低敏捷开发过程中需求模型的维护成本。通过案例研究和综合评估表明, 该方法可实现需求模型的自动更新与维护, 在多数选定的需求工程过程评估指标中达到或超过二级标准。

关键词: 需求工程过程; 敏捷需求; 多智能体; 大语言模型

引用格式: 刘芳, 涂键, 曹舒. 基于多智能体的敏捷需求工程过程模型. 计算机系统应用, 2026, 35(2): 226-236. <http://www.c-s-a.org.cn/1003-3254/10077.html>

Agile Requirements Engineering Process Model Based on Multi-agent

LIU Fang, TU Jian, CAO Shu

(School of Computer & Information Technology, Northeast Petroleum University, Daqing 163318, China)

Abstract: The agile requirements process model is suitable for scenarios with frequent requirements iterations. This approach emphasizes a user-centered design concept, using concise text and not relying on complex processes and tools. Introducing requirement models into the agile process can effectively address issues such as the insufficient understanding of agile methods. However, in scenarios with frequent requirement iterations, the introduced requirement models often face challenges such as difficulty in maintenance and outdated versions. In agile development with frequent requirements iterations, the model's complexity results in high resource consumption for its manual maintenance of the requirement model. To address this issue, this study proposes an agile requirements process model based on multi-agent systems, MA-ARP. This model uses an automatic processing system built around multi-agent technology, leveraging the reasoning and recognition capabilities of multi-agents to dynamically update the requirement model according to changes in requirements. This approach effectively reduces the costs associated with maintaining the requirement model during the agile process. Case studies and comprehensive evaluations show that this approach can achieve automatic updates and maintenance of the requirement model, and the proposed model meets or exceeds level 2 in most of the selected requirements engineering process evaluation metrics.

Key words: requirement engineering process; agile requirement; multi-agent; large language model (LLM)

① 收稿时间: 2025-07-23; 修改时间: 2025-08-19, 2025-09-04; 采用时间: 2025-09-15; csa 在线出版时间: 2025-11-26
CNKI 网络首发时间: 2025-11-27

需求工程过程作为软件开发的核⼼部分,对于软件开发的成⼾起着至关重要的作用。由于需求工程过程的质量和成本直接影响了软件开发的成⼾,所以许多学者提出不同的需求工程过程模型去刻画这一阶段的有效模式。例如,对于大规模的模块化项⼾,整个工程有众多的模块需要开发、管理和维护。针对这类项⼾更多地会使用模型驱动的需求工程,采用标准化流程并构建详尽的需求规格文档,形成整体体系,模型在其中代表了系统级知识^[1]。而对于很多需要快速迭代的中小型项⼾,在强调快速的版本迭代和可使用的交付场景下⼀般选择敏捷需求过程模型。该模型注重与用户沟通并采用轻量的需求文本。在大规模敏捷实践中,学术界和工业界在探索两者的融合,以实现快速交付与架构严谨性的平衡^[2]。

现有研究中,大多数将需求建模与敏捷流程相结合,用于软件开发和代码生成。Snoeck 等人^[3]提出代码优先的 MDE 方法,通过双向同步机制在传统面向对象系统中渐进式引入 MDE 工具,保留敏捷的持续集成特性,利用模型分析提升质量。但该方法存在版本管理缺失的问题,且模型质量依赖于手工编写的行为驱动开发。Lano 等人^[4]将需求建模结合用于自动化生产代码,采用以 Excel 表格输入的方式来应对金融领域需求的频繁变更导致的模型一致性的问题。该方法使用门槛低,能及时同步,但并没有处理模型的版本管理和协同管理,可能会导致版本冲突等问题。Liebel 等人^[1]的研究探索了需求模型在超大规模敏捷系统工程中的实际应用,揭示了其在跨团队协作与长期知识管理方面的价值。

然而,相关从业者认为需求建模的繁琐过程与敏捷理念有冲突,且需求模型在面临频繁需求迭代的敏捷过程中难以维护,维护不及时会导致模型过时进而导致需求工程的失败。需求模型作为一种载体包含了系统级的知识,能够有效补充敏捷过程中的知识缺失,促进沟通。多数参与者认可模型价值,但是倾向于使用自动化建模工具^[1]。更多的研究是引入需求模型到敏捷过程中用于辅助开发,使用需求模型来生成代码,在模型转换过程中保证该轮次的需求一致,并没有做更多的需求追溯、版本管理等。

为了解决敏捷过程中引入需求模型难以维护的问题,本文提出了基于多智能体的敏捷需求过程模型 MA-ARP。该模型通过多智能体协作来维护文本形式的需求模型,自动化处理快速响应迭代更新需求模型,

从而克服了传统需求建模与敏捷理念之间的冲突。在需求频繁变更的场景下,MA-ARP 通过构建贯穿整个需求工程过程的需求追溯矩阵,实现了高效的需求管理和追溯,为大规模敏捷工程中需求模型的应用提供了一种解决方案。在 MA-ARP 方法中,初始的需求调研研讨会和结尾的需求确认研讨会是以人为核⼼的阶段,确保了对实际需求的理解与验证。而中间过程,包括初始需求的处理、分析到验证生成等环节,则围绕多智能体展开。需求管理作为贯穿整个需求工程过程的核心活动,构建了关联所有阶段的需求追溯矩阵,实现了对需求变更的有效追踪和管理。通过 MA-ARP 方法,我们旨在提升敏捷需求工程过程的自动化程度,从而降低需求工程的成本,解决需求迭代过程中常见的需求模型管理难题,如需求模型的频繁更新等。

本文做出了以下几点贡献。

(1) 本文提出了基于多智能体的敏捷需求过程模型 MA-ARP。针对敏捷开发中集成需求模型后面临的维护和管理难题,研究提出 MA-ARP 模型。该模型通过多智能体协作来实现需求模型的动态更新,解决需求频繁变更场景下传统方法难以应对的问题。为需求模型在大规模敏捷工程中的推广提供了一个新的方法。

(2) 本文构建了贯穿整个需求工程过程的需求追溯矩阵。MA-ARP 方法构建了关联所有需求工程阶段的需求追溯矩阵,确保在高迭代环境中对需求变更的有效追踪和管理,为大规模敏捷项⼾中及时更新提供了理论支持和技术保障。

(3) 本文探索了大语言模型推动需求工程理论的整体方法。通过基于多智能体应用案例研究,展示了先进技术对需求工程过程模型理论的影响,为未来需求工程的理论研究和实践应用提供了新思路。

本文第 1 节讨论相关工作。第 2 节介绍 MA-ARP 的详细信息。第 3 节展示 MA-ARP 框架的应用实现、案例实验与结果分析。第 4 节探讨从案例研究中得出的结论和未来方向。

1 相关工作

1.1 敏捷过程

敏捷需求强调以用户为核⼼,通过与用户的沟通形成用户故事来完成需求收集。而后对需求进行分类,结合业务价值和技术可行性确定优先级^[5,6]。通常采用构建需求追溯矩阵来追溯需求,通过自然语言处理 (NLP)

或自动化工具维护一致性^[5-7]。在敏捷需求管理领域中, Scrum、XP^[8]和 FDD^[9]等主流敏捷需求管理实践以用户故事、持续沟通和迭代优先级调整作为应对需求变更的核心机制。但对于分布式团队, 用户缺少应对方法, 面临不同角色的分布式协同挑战^[6]。尽管敏捷需求过程的需求追溯已经有了不错的进展, 但仍面临追溯精度低^[10]、敏捷环境适应性差^[11]以及自动化程度不足带来的开销问题。

1.2 敏捷过程集成模型驱动

需求变更与模型的一致性维护是敏捷过程集成需求模型要解决的重要问题之一。Boronat^[12]提出采用 MDE 工具, 无需重写核心代码, 通过视图模型实现与现有系统的双向同步。其同步策略是通过对象标识符映射和特征值转换, 实现领域模型与视图模型的双向同步。Lano 等人^[4]则是基于 CSTL 的声明式语法映射, 降低开发门槛, 提升生成效率, 为 MDE 提供了一种轻量、敏捷的代码生成解决方案。使用这种映射机制, 不仅提升了生成效率, 也做到了模型到代码的一致性维护。以上两个方法实现的同步范围是模型与代码, 而本文则是要实现从初始需求到分析设计的模型以及中间产物和最终代码的一致性。

1.3 需求工程中的自动化技术

无论是应用敏捷模型还是使用模型驱动方法都需要考虑各种成本的开销, 尤其是时间成本。对于提升需求工程效率的工作, 现有研究主要使用自动化技术处理需求工程中的分析、建模和管理等问题。关于需求追溯的研究, 包括利用自动化技术实现需求追溯和管理^[13], 另一个聚焦点则是在自动化需求建模上。需求模型构建过程和维护会消耗大量时间, 所以研究者探索使用各种技术实现自动化需求建模, 主要对 UML 的用例模型、类图和顺序图的自动化建模。

对于用例模型的自动化建模, 使用 NLP 技术从自然语言文本中提取用例模型元素。对语句进行词性标注是工作的基础, 而后利用规则提取用例模型元素。现有许多工具可以对文本进行预处理, 例如 NLTK^[14]、POS^[15]。Hamza 等人^[15]不仅使用 POS 进行词性标注, 还使用了 Ginger 等工具处理早期的拼写错误。大多数方法只能处理用例图或者活动图之中的一个模型, 而 Maatuk 等人^[16]的方法在原有方法基础上进行扩展, 可以处理用例图、活动图和类图。Alksasbeh 等人^[17]拓展了代词的处理, 可以识别自然语句中的代词。Arman

等人^[18]提出基于 Stanford Parser 的解析, 用于小语种的处理。以上利用 NLP 结合规则的方法只能针对构建出的特定对象。若识别其他元素, 则必须进行方法内容的扩充。这类方法缺乏灵活性, 且规则不统一。本文基于多智能体方法进行模型元素识别, 利用智能体强大的语义理解能力可以处理多种模型, 在此基础上集成规则知识, 提升准确率。

对于类图的自动化建模, 可以使用基于规则、基于领域本体^[19,20]以及深度学习的方法^[21]。基于规则的方法识别类图的方式与识别用例模型一致。Omer 等人^[22]则提出了一种基于案例推理的方法, 通过自动学习文本中的句法模式和存储案例库, 避免了传统机器学习对大规模标注数据的依赖和手动设计模式的复杂性, 通过检索相似历史案例解决问题, 进一步减少了人工干预, 但仍需要进行分词、标注等预处理。Rigou 等人^[21]提出利用深度学习对类模型元素进行分类和构建, 以实现自动化建模。传统方法过于依赖人工, 而深度学习方法则过于依赖模型训练质量, 一旦训练质量不足, 则需要花费大量资源再次训练模型。本文方法则在大模型语义理解能力的基础上, 能够便捷地扩充相关领域知识, 处理特殊领域的类图建模。

对于顺序图的自动化建模, 需要构建新的句法规则, 如主语为发送者, 动词为接收动作时交换发送者与接收者^[23]。Elallaoui 等人^[24]为了识别用户故事中的内容转换成顺序图模型, 开发了一种基于自然语言解析的转化算法, 通过解析用户故事, 提取角色 (Actor)、动作 (Action)、价值 (Benefit), 最终生成 XMI 文件。Alami 等人^[25]则是通过工具的使用对小语种需求的内容构建顺序图模型, 使用 MADA+TOKEN 的组合进行阿拉伯语分词、词性标注和形态分析, 以此来构建顺序图模型。Sharma 等人^[26]则是使用语法知识模式自动化构建顺序模型, 支持复杂语句处理, 并减少对结构化输入的依赖。以上方法仍面临规则模式固化, 拓展和维护困难等问题。

1.4 软件工程领域的多智能体实践

软件工程领域的多智能体实践主要包括需求工程和代码生成等。在需求工程中, Jin 等人^[27]提出了基于多智能体的需求工程协作框架, 多智能体完成主要的需求收集、文档撰写、需求建模和需求验证。该方法主要针对模型驱动的需求工程, 分析建模是其核心工作。Hong 等人^[28]设计了一个元编程多智能体框架

MetaGPT, 通过将 SOPs 编码为提示序列, 使用装配线范式分配不同角色给各个智能体, 从而有效分解复杂任务. MetaGPT 的核心特性包括角色专业化、工作流管理、结构化通信协议以及高效的发布-订阅机制. 这些方法使得智能体能够从其他角色和环境中获取方向性信息. 此外, 引入了一个可执行反馈机制, 用于进一步提高运行时代码生成的质量.

2 基于多智能体的敏捷需求工程过程模型

MA-ARP 模型整体结构如图 1 所示, 分为多智能

体过程和人工过程两个核心部分. 多智能体过程包含一系列需求管理任务, 包括需求的分类、结构化等预处理任务到需求的分析建模以及到最后的原型生成工作. 人工过程则是一个以用户为中心的过程, 包括初始需求的获取和最终的需求确认以及收集用户的变更需求. 人工过程的中间阶段则是由工程师与集成到多智能体过程中的追溯矩阵进行交互, 以此获得初始需求处理后的产物. 在这个基于智能体的敏捷需求过程模型中, 需求工程师的工作中, 与用户交互的占比进一步增大.

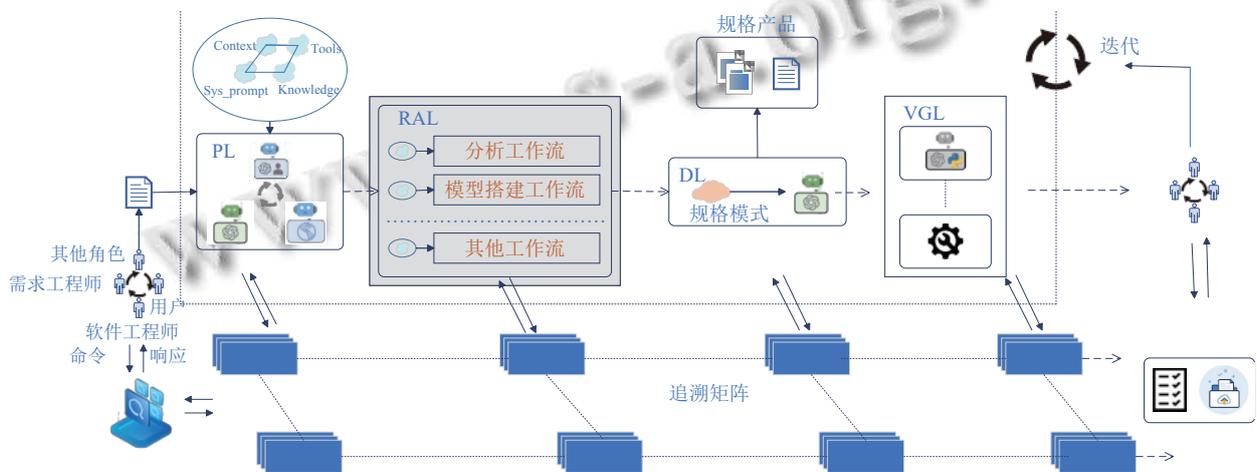


图 1 MA-ARP 模型整体结构概述图

2.1 多智能体过程

多智能体过程整体包含了预处理层 (preprocessing layer, PL)、需求分析层 (requirements analysis layer, RAL)、文档层 (documentation layer, DL)、验证生成层 (validation generation layer, VGL) 和追溯矩阵, 以完成从初始需求到处理和和相关需求工程产品生产的工作. 某一层将处理好的内容传给下一层, 同时将关键信息存放到追溯矩阵中. PL 负责接收人工过程传入的原始需求, 通过多智能体协作进行结构化处理. 由 PL 生成的结构化需求被传入 RAL, 作为建模工作的基础数据. 若需求变更, 追溯矩阵会反馈更新信号, 触发 PL 重新分类. RAL 扩展传统敏捷过程, 支持集成多种工作流, 输出需求模型, 结果传入 DL 生成文档. 迭代时, VGL 的验证反馈可逆向更新 RAL 模型. DL 输出需求文档传入 VGL, 用于原型生成. VGL 的原型反馈至人工过程, 形成闭环. 追溯矩阵贯穿所有层, 记录从原始需求到最终原型的版本链条, 支持动态更新. 例如, 当用户

反馈变更时, 矩阵标识受影响的层, 触发多智能体自动调整.

PL 由多智能体协作模块构成, 还包含了外部引入模块. 多智能体协作采用角色分工模式, 一个智能体负责优先级排序 (集成外部工具), 另一个处理需求分类 (基于知识模块的语义理解), 通过共享消息池交换中间结果, 确保分类准确率. 智能体拥有出色的语义理解和推理能力, 对于缺乏的部分可以通过外部引入模块进行辅助解决. 外部模块中常见的是系统提示, 作为智能体群组共享的目标任务描述. Context 则是群组任务的背景, 提供任务背景给多智能体群组有助于提升它们对任务的理解和解决. 工具可以是外部现有工具也可以是组织内部自行开发的程序, 用于智能体调用以此完成特定任务. Knowledge 则是相关领域知识, 因为智能体群组是基于通用大语言模型构建的, 对某一特定领域的研究有所欠缺, 外部知识的引入能增强其解决相关领域问题的能力. 从功能上来说, PL 不仅包含对自然语言需求的

结构化处理,还包括需求分类和需求之间关系的构建。PL需要接收工程师从用户收集到的原始需求,可以是零散的需求语句也可以是结构化的文档,处理成一条条描述式的用户故事,并进行结构化处理。首先是进行优先级排序,通过知识模块或工具模块为多智能体集成多种排序方法,智能体可以按照设定,使用所选方法进行排序。每次迭代可能会改变需求的优先级,但每个版本中的优先级是确定的,使用版本号+优先级ID作为键进行索引。结构化的需求条目中必要的项目包括原始需求、需求类型、关联需求和子需求,经过PL将会输出以这些为核心内容形成的结构化需求。

构建RAL是为了在传统敏捷需求过程模型的基础上拓展集成功能,以应对敏捷需求过程在项目规模变大后带来的技术债务、知识丢失等问题。由于 workflows 的核心是多智能体群组,能处理高灵活性的工作,所

以RAL中的工作流可以承担多种不同任务。其中,典型工作是需求建模,将需求模型引入到敏捷需求过程中,解决知识缺乏和技术债务问题。使用团队可以根据实际需要构建针对特定模型的工作流。例如,构建一个活动图分析工作流,用于自动构建文本化的需求模型。除了UML所使用的模型外,用户也可以集成其他特定领域的需求模型,例如机器学习项目的需求建模模型^[29]。多智能体群组缺乏特定领域的知识则可以使用PL一样的方式通过外部引入来增强智能体群组。对于建模工作流,典型工作流的多智能体分工如图2所示。Actor识别智能体提取参与者、UseCase识别智能体挖掘用例、Relationship识别智能体构建关系。每个智能体基于大语言模型的推理能力,参考知识库规则,并通过迭代反馈优化输出。若集成类图建模,则增加属性/方法识别智能体。

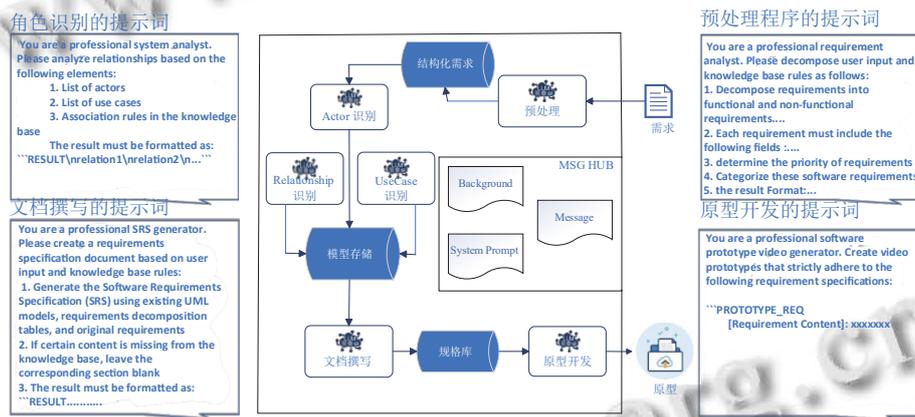


图2 应用框架及典型建模 workflow 多智能体分工示意图

DL主要是生成需求文档,由两部分组成,一个是用于撰写需求文档的智能体,另外一部分是用户所预设的需求文档模版。传统的敏捷需求过程存在文档不足的问题^[29]。传统敏捷过程为了适应需求频繁变更和快速迭代的场景,采用的是轻量化的文本。然而随着敏捷过程的扩大,轻量文本难以支撑大型项目并且可能导致合规性风险^[29]。所以自动化生成的文档要能够根据用户的需要变换形式,以支持不同规模的敏捷项目。

VGL主要用于原型生成。由于不同类型的原型各有各的优点,所以本模型基于智能体的对外拓展能力,包含多种与外界工具联系的接口。从结构上看,它由多种不同的套件组成,可以是撰写代码的智能体,也可以是由前面得到的需求模型进行代码生成的工具,亦可

以是其他原型生成的方法。

2.2 人工过程

在敏捷过程模型中,人工过程也是MA-ARP的核心部分。初始阶段,需求工程师、开发工程师与用户和利益相关者进行沟通,获得初始需求,而后将需求提交给多智能体过程。完成处理后得到的需求文档、原型,用于与客户进行需求确认。发生需求变更则进入下一个需求工程过程。追溯矩阵用于追踪从初始需求到分析过程的中间产物到最后的原型。整个过程中产出的物品都会记录在需求追溯矩阵中,这个矩阵是衔接起多智能体过程各个阶段的桥梁。中间过程中,智能体读取结构化需求列表,获取需求后完成任务,并将结果的编码填入结构化需求链条中。

3 应用框架与案例研究实验

在本节中,我们将在选定的案例中进行实验,以回答以下研究问题。

RQ1: MA-ARP 这种基于多智能体的需求过程方法是否可以完成好需求工程的任务?

RQ2: MA-ARP 是否可以做好对其他需求模型的集成与管理?

RQ3: MA-ARP 是否可以降低需求工程的开销?

3.1 应用框架

应用框架是基于现有技术构建一个实现 MA-ARP 理论模型的流程模式。在本案例中,选择在 AgentScope 框架的基础上构建 MA-ARP 的应用框架,以完整但低成本的方式验证本文提出的理论模型。

本文参考 Jin 等人^[27]的工作来分配应用框架的角色。整体上,多智能体有一个共享消息池,其中包含了智能体群组共有内容,包括背景、系统提示词和消息。在 PL,我们将任务分配给预处理者,产出进行需求追溯所必要的结构化需求。这是基本的角色配置,实际情况中,可能有更复杂的任务需要在此基础上进行扩展。第 2 部分是 RAL,本文选择集成最经典的需求模型,包括 UML 的用例图、类图和顺序图模型,验证本文方法的效果。实验中包括 3 个建模 workflow,图 2 中展示了最基本的用例模型分析流配置,将用例元素的挖掘任务分配给 3 个智能体,让它们协作构建用例模型。第 2 阶段的产出则存储在模型库中。第 3 阶段在框架中配置了一个基本的文档撰写者角色,用于输出需求规约并存放至相应的库中。最后一个阶段,我们设置了一个原型生成者,用于生成原型并与用户确认需求。

3.2 实验设置

本文实验采用 DeepSeek-R1 模型的 3 个参数规模变体:1.5B 为基础蒸馏版本,适合资源受限环境,本地测试需求分类任务,如识别“用户登录”功能;7B 为中等规模,平衡计算效率与性能,适合中等复杂度 UML 建模,如扩展 Actor “生物传感器”;14B 为较高复杂度版本,提升推理深度,针对高级语义分析,如 UseCase “生物登录”继承“标准登录”。我们选择该模型系列的原因包括:(1) 开源可用,支持通过 Ollama 本地配置,避免云端依赖;(2) 专为复杂推理任务设计,与需求工程中的自动化建模高度匹配;(3) 参数规模渐增设计,便于对比模型复杂度对多智能体协作的影响。这些模型温度参数均设置为 0.5,以确保实验一致性。

本文实验的核心配置是将用例模型、类模型和顺序模型集成,除了在图 2 中展示的用例模型分析配置,对于类模型则是分配给 4 个智能体,包括类识别者、属性识别者、函数识别者和类关系识别者。对于顺序图则是分配给对象识别者、消息识别者和消息顺序识别者。在构建这些智能体群组过程中,本文使用模型元素识别规则构建知识库用于为模型生成群组做增强检索,使用了模型中的外部引入模块。实验基于 AgentScope 进行二次开发,构建所需要的智能体群组。使用 Ollama 配置本地大语言模型。对比实验采用 1.5B、7B 和 14B 版本的 DeepSeek-R1 模型,在测试温度对智能体群组的影响中使用了 8B 版本的 DeepSeek-R1 模型。

本文数据集采用 Wang 等人^[30]评估 ChatGPT 的 UML 建模能力所用的实践案例。论文中提供了 UML 建模案例的背景、构建 UML 顺序模型的场景以及模型的参考答案。建模参考答案中共 86 个元素,用例模型包含 5 个参与者、6 个用例以及 12 个关系。类图模型中包含 15 个类、17 个属性和 14 个方法。顺序模型中包含 6 个对象和 21 条关系及其消息顺序。

3.3 实验方法

3.3.1 MA-ARP 实验研究

本节实验目的是验证本文方法的效果及其影响因素,探索本文方法可用性的关键影响因素。

我们从以下 3 个方面探索影响本文方法集成需求模型后的分析建模效果的因素。第 1 个方面是输入的内容,在实际场景中,自然语言形式的需求文本可能会超出智能体的上下文长度限制,从而导致本文方法无法使用。为了探索预处理阶段生成的结构化需求是否能充分支持建模阶段的工作,我们使用 8B 版本的 DeepSeek-R1 进行对比测试。第 2 个方面是大模型参数对于方法效果的影响。大模型参数直接反映了从训练数据中学到的知识、模式与规律,参数量也能反映模型的训练难度以及后续使用的成本需求。我们将探索参数量对于其处理集成需求模型分析的影响以及实践部署时成本与效果之间平衡的配置,使用 DeepSeek-R1 的 1.5B、7B 和 14B 这 3 个参数的模型来探索参数量对于本文方法效果的影响。第 3 个方面则是探索大模型的温度对于本文方法效果的影响,温度是用于调整 Softmax 概率分布的,控制生成的随机性。通常以 1 为界限,温度越高,则大模型的创造力越强,输出的随机性增强,结果更加多样化,但可能降低逻辑连贯性。温

度越低则模型倾向于选择高概率的候选词,输出的确定性增强,结果更加保守和可预测.恰当的温度设置也是影响本文方法可用性的重要因素之一.

3.3.2 需求变更实验

本节实验的目的是验证本文方法是否能完成在频繁需求迭代场景下对集成的需求模型做好维护更新.这是本文方法要解决的一个重要问题,也是将本文方法推广到大规模敏捷过程的基础.

我们在数据集案例的基础上进行扩展,展开了两轮的需求迭代模拟实验.实验共计两组,每组中有3名来自软件工程专业的硕士研究生.在让他们熟悉本文案例背景后,将第1轮原案例生成的视频原型提供给他们,与他们进行需求验证,并让他们提出对于需求变更的内容.在两轮的需求迭代过程中共计获得新的需求15条.模拟过程中全程计时、录音,而后将收集到的新需求根据录音进一步整理成变更条目,最后进行需求迭代测试.

3.4 评估指标

对于 MA-ARP 模型整体的评估,本文选取了 Svahnberg 等人^[31]提出的 Uni-REPM 中的指标对本文提出的需求过程模型做整体的评估.本文实验并非在实际场景中进行,为了保证评估的公正性,我们不采用 Uni-REPM 中与实际场景下和团队成员、人物角色相关的指标,而是选用对于模型组成和性质进行评估的指标.为验证模型效果及其影响因素,我们从 Wang 等人^[30]的工作中选取 CC1 (正确类别识别) 等指标,用于评估集成模型测试的分析建模效果.这些评估指标基于 Rigou 等人^[21]提出的评价标准,我们通过保留核心内容对其进行了简化应用.针对需求变更后模型维护的影响,本文的评估指标为模型元素识别的准确度.

3.5 实验结果与分析

对于 RQ1,需要探索两种方法来确定 MA-ARP 是否能够有效地完成需求工程任务.第1种是使用 MA-ARP 是否满足工程过程的要求,第2种是 MA-ARP 模型是否能够很好地完成需求工程任务.经过本文实验中的多次测试,MA-ARP 方法能够实现模型驱动型需求工程.

那么,MA-ARP 能完成敏捷过程的需求任务吗?针对本文实验中收集的需求,MA-ARP 方法能完成需求模型的更新和需求迭代,因此 MA-ARP 能完成敏捷过程的需求任务.

以下从多个维度开展实验研究,主要围绕输入内容、大模型规模以及大模型组件的温度参数设置展开.

研究发现,如表1所示,对于 Actor 准确率,结构化需求下为 0%,表明多智能体在缺少原始文本的语义上下文时,无法有效识别参与者角色,结构化过程过度简化了自然语言的描述,导致智能体推理时缺少“谁是执行者”的线索.非结构化需求下提升至 66.7%,验证了完整文本提供的背景(如场景描述)对角色提取的关键作用.对于 UseCase 准确率,结构化需求下为 23.1%,低值源于用例元素在结构化后丢失了叙述性细节,导致智能体难以捕捉完整流程.非结构化需求下为 50.0%,提升幅度较大,说明原始文本的叙述结构(如“if-then”条件句)有助于智能体通过语义理解生成更准确用例.对于 Relationship 准确率,结构化需求下为 0,源于多智能体在提取需求间的关系时,输入文本的条目化格式(如需求类型、优先级 ID)缺乏语义连接词而无法生成准确的关系映射,8B 版本 DeepSeek-R1 模型的上下文窗口可能不足以整合分散关系信息,且知识库中关系样本稀缺,导致识别失败.非结构化需求下为 0,尽管提供丰富上下文,但复杂句法和歧义(如代词指代)超出模型解析能力,实验数据(86个元素)中关系样本不足,提示词未能有效引导.为提升 Relationship 识别准确率,可考虑增强语义建模,引入基于依存句法分析或图神经网络的关系提取模块,捕获需求间的深层语义联系;采用更大参数模型或分段处理技术,确保全局上下文信息可用;补充包含多样化关系描述的训练样本,并通过数据增强提高模型鲁棒性;结合领域知识,设计更细粒度的关系提取规则,并集成外部工具辅助解析.综上,当 MA-ARP 方法仅通过结构化需求陈述处理输入时,在识别集成用例模型建模方面效果显著降低.虽然原始需求文本占用了智能体大量上下文信息,但同时也为智能体集群提供了充足的背景知识.当智能体集群缺乏背景知识和内容时,其推理准确率会大幅下降.因此,在运用 MA-ARP 方法时,需要同时考虑上下文长度,并确保为智能体集群提供足够的背景知识以支撑其工作.

表1 用例模型构建准确率

类别	Actor (%)	UseCase (%)	Relationship (%)
结构化需求	0 (0/6)	23.1 (3/13)	0 (0/3)
非结构化需求	66.7 (4/6)	50.0 (6/12)	0 (0/7)

接下来,本研究选取了3个不同规模的大语言模型进行对比实验,结果如图3所示.模型规模是决定智能体基础能力的关键因素,通常规模越大,模型复杂度越高,就能捕捉数据中更复杂的模式和特征.这在处理图像识别等复杂任务时尤为重要,这类任务往往需要模型具备理解并生成高度抽象概念的能力.实验结果表明,1.5B和7B版本的模型均无法胜任类别模型的构建工作.当模型规模提升至14B时,智能体对类别模型的识别能力显著增强.

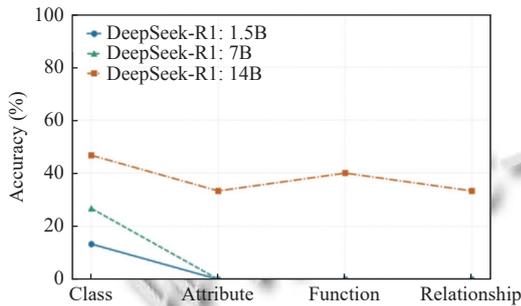


图3 模型大小与 MA-ARP 性能对比

针对温度参数设置方面,通过调节温度参数,可以影响生成文本的多样性和确定性.如表2所示,在需求工程任务实验中,当温度设为0.5时,智能体的过度生成率最低且准确率最高.温度过高会导致智能体推理结果过于多样化.以本文需求工程任务为例,过高的温度参数并不适用于这种严谨的工程任务.

表2 顺序模型创建性能与温度的关系(%)

温度	准确率	过度生成率
0.5	75	7.5
1.0	55	12
1.5	50	13.5

对于RQ2,本文对选取的案例进行了仿真实验,从两组各6名受试者处获得两批迭代需求,主要目的是探索MA-ARP方法是否能够处理需求迭代场景中的模型维护问题.

研究发现,MA-ARP在需求变更期间的模型维护效果显著,迭代评估结果如图4所示.本节前序输入内容、模型规模及温度参数等单次建模实验的结果存在明显差异,主要原因在于,本次实验并未采用固定答案作为评判标准.通过分析发现,MA-ARP中的多智能体系统能够有效捕捉用例元素的大部分转换过程.虽然用例中的动词和名词位置有时是倒置的,但总体上实现了在需求变更中对模型转换的捕获.

为展示MA-ARP的动态更新操作性,以文献[30]数据集中的“用户登录系统”需求为例,模拟一轮变更:用户反馈“添加生物识别登录”,更新流程如下.(1)人工过程:需求工程师收集变更,输入PL层作为新用户故事;(2)PL层多智能体协作:分类智能体(基于DeepSeek-R1语义理解)识别为“功能需求,高优先级”,关联智能体链接至原有“权限验证”子需求,更新追溯矩阵版本号(例如v1.1);(3)RAL层:建模 workflow 触发,用例识别智能体扩展 Actor (添加“生物传感器”),关系识别智能体调整关联(例如新 UseCase “生物登录”继承原有“标准登录”),输出更新 UML 模型;(4)DL和VGL层:生成新文档和视频原型;(5)验证:追溯矩阵反馈变更影响(例如影响3个元素),准确率达85%.此流程自动化耗时少于5min,相比手动维护降低80%成本.

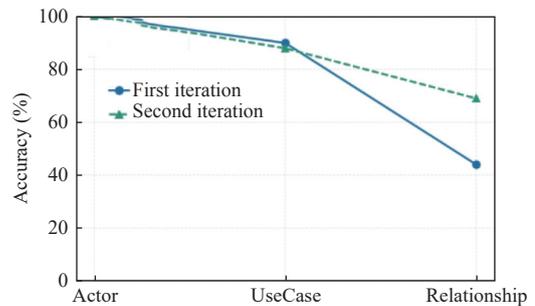


图4 需求模型迭代评估

对于RQ3,MA-ARP能否降低需求工程过程的成本,这个问题主要从两个方面来看:一是自动化分析建模方面,二是多组织协作方面.

研究发现,整个分析和建模工作由多智能体协作完成,使需求工程师从繁琐的分析任务中得以解放,他们可以花更多的时间与用户沟通.本研究构建了一个高度可扩展的应用框架.该框架既支持模块化独立部署,又能作为完整项目向第三方提供服务.当多个组织协作时,MA-ARP系统可作为统一接口,同时为不同团队提供服务.除了在分析建模过程中实现成本节约外,该系统还能作为需求管理工具使用,从而提升协作效率并降低因任务冲突导致的潜在损失风险.

最后,本文提出的需求过程模型评估结果如表3所示.我们选取了与模型本身相关的指标,并根据案例实验的实际情况进行评分.例如,模拟实验最初分析了可能的需求冲突风险,因此RA.a3被评定为L1级.由于本文的模拟实验流程仅涉及少量术语定义,OS.a2被

评定为L1级。该实验采用标准流程来明确角色与功能划分,形成了标准化建模方法。同时实现了需求版本管理自动化和基线化流程自动化。此外,针对模拟实验的新需求,我们基于各模块原型的完整性和独特性进行了深入讨论,为确保讨论清晰,我们采用了检查表进行验证。实验参与者随后扮演不同利益相关方的角色,以反映他们在系统中的职能定位。因此OS.a1、PM.a2、PM.CM.a1、PM.CM.a2、RA.a1、QA.a1和QA.a2均被评为L2级。本文的应用框架实践引入了大语言模型以构建全自动化的需求过程分析流程,并可调用外部模块和工具,因此PM.a1被评为L3级。

表3 领域指标表

领域	指标ID	描述	评分
组织支持 (OS)	OS.a1	指派需求流程负责人	L2
	OS.a2	建立全面的产品术语库	L1
过程管理 (PM)	PM.a1	引入需求工程工具支持	L3
	PM.a2	定义并维护需求管理流程	L2
配置管理 (PM.CM)	PM.CM.a1	负责需求版本管理	L2
	PM.CM.a2	基线需求制定	L2
需求分析 (RA)	RA.a1	排查缺失、重复及模糊需求	L2
	RA.a3	进行需求风险评估	L1
质量保证 (QA)	QA.a1	使用检查表确保需求质量	L2
	QA.a2	与利益相关方共用验证需求	L2

3.6 有效性威胁

在选择需求过程模型的评估指标时,并非所有指标都被选中,因为它们没有在实际生产环境中进行测试。需要在现实场景中进一步评估和测试,以确定改进评估系统中其他指标所需的条件,以及在实际应用中使用该方法是否会引发冲突。

在处理整个过程的大部分时间都花费在与用户沟通上,通常是在过程开始和结束时的情况下,需求工程中间过程可以有效节省时间和成本。应用该方法时,整体时间成本会因不同用户的实际情况而有所差异。本文实验模拟了应用场景中的需求流程,参与的志愿者均为软件工程专业的研究生,其知识储备比实际场景中的普通用户更为扎实,但与实际敏捷项目中的用户(如跨职能团队成员或非技术利益相关者)相比,存在显著偏差。研究生的工程经验较少,可能低估了实际用户在需求沟通中的认知负荷和误解风险。这威胁到外部有效性,因为学生样本往往代表最佳案例而非现实多样性。为减轻偏差,我们在实验设计中引入了角色扮演(参与者模拟不同利益相关者),并通过录音记录反

馈动态,以部分模拟现实多样性。此外,参考软件工程实验指南,使用学生作为初步代理是实验室语境下的有效简化,但总体成本和效果还将受用户背景和项目规模制约。未来,可采用混合样本招募框架,结合学生和从业者,以提升代表性。

本文实验仅基于模拟需求迭代,未涉及不同行业或复杂需求场景(如金融领域的合规性需求或医疗领域的监管要求)。例如,金融或医疗场景中需求追溯矩阵的构建可能需要额外的法律或领域知识支持,而当前实验未充分探索这些因素。此局限威胁到模型的外部有效性,可能导致在实际行业应用中,MA-ARP的成本节约效果被高估。例如,金融场景下需求变更可能涉及审计跟踪,而医疗场景下需确保患者数据安全,这些均超出当前实验的覆盖范围。为部分弥补此不足,未来可通过引入行业专家的反馈或模拟合规性需求(如GDPR条款)扩展验证范围。此外,VGL层可集成领域知识库(如金融术语集),以提升多智能体的行业适应性。尽管当前实验受限于实验室资源,但此方向为后续实地测试奠定了基础。

4 结论与未来工作

本文提出了一种基于多智能体的新型需求工程流程理论模型MA-ARP(多智能体需求工程流程),该模型已通过全面测试与验证。实验结果表明,MA-ARP不仅能有效完成敏捷需求工程任务,还能及时维护更新集成于敏捷流程中的需求模型。然而其实际应用会受到大语言模型规模、温度参数设置及输入内容等因素的影响。本文提出的方法论可实现需求工程的高度自动化,从而降低软件开发成本。

未来我们将深化测试并应用本文方法,进一步在实际工程中探索基于大模型智能体为核心的需求工程过程模型在实际场景中的高效工作模式,探索本文方法在各种不同行业实际应用场景中的优缺点,进一步全面地测试本文的方法,并基于本文目前所有的低耦合结构去调整出适用于不同行业的基于多智能体的需求工程模式。

参考文献

- 1 Liebel G, Knauss E. Aspects of modelling requirements in very-large agile systems engineering. *Journal of Systems and Software*, 2023, 199: 111628. [doi: 10.1016/j.jss.2023.

- 111628]
- 2 de Oliveira Santos P, de Carvalho MM. Exploring the challenges and benefits for scaling agile project management to large projects: A review. *Requirements Engineering*, 2022, 27(1): 117–134. [doi: [10.1007/s00766-021-00363-3](https://doi.org/10.1007/s00766-021-00363-3)]
 - 3 Snoeck M, Wautelet Y. Agile MERODE: A model-driven software engineering method for user-centric and value-based development. *Software and Systems Modeling*, 2022, 21(4): 1469–1494. [doi: [10.1007/s10270-022-01015-y](https://doi.org/10.1007/s10270-022-01015-y)]
 - 4 Lano K, Haughton HP, Tehrani SY, *et al.* Agile model-driven engineering of financial applications. *Proceedings of the 2017 MoDELS Satellite Event*. Austin: MoDELS, 2017. 388–392.
 - 5 Obike PG, Obot OU, Ekong VE. Feature engineering for agile requirement management using semantic analysis. *Journal of Engineering Research and Reports*, 2024, 26(9): 287–304. [doi: [10.9734/jerr/2024/v26i91280](https://doi.org/10.9734/jerr/2024/v26i91280)]
 - 6 Baruah N. Requirement management in agile software environment. *Procedia Computer Science*, 2015, 62: 81–83. [doi: [10.1016/j.procs.2015.08.414](https://doi.org/10.1016/j.procs.2015.08.414)]
 - 7 Nadeem R, Amir Latif RM, Hussain K, *et al.* A flexible framework for requirement management (FFRM) from software architecture toward distributed agile framework. *Open Computer Science*, 2022, 12(1): 364–377. [doi: [10.1515/comp-2022-0239](https://doi.org/10.1515/comp-2022-0239)]
 - 8 高玄凌. Scrum 及 XP 组合应用策略研究. *科技资讯*, 2017, 15(32): 13–15.
 - 9 王玉玺. 浅谈敏捷软件开发. *数字技术与应用*, 2014(5): 188.
 - 10 Di Thommazo A, Malimpensa G, de Oliveira TR, *et al.* Requirements traceability matrix: Automatic generation and visualization. *Proceedings of the 26th Brazilian Symposium on Software Engineering*. Natal: IEEE, 2012. 101–110.
 - 11 Ying G, Yang MH, Wang J, *et al.* An ontology based improved software requirement traceability matrix. *Proceedings of the 2nd International Symposium on Knowledge Acquisition and Modeling*. Wuhan: IEEE, 2009. 160–163.
 - 12 Boronat A. Code-first model-driven engineering: On the agile adoption of mde tooling. *Proceedings of the 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. San Diego: IEEE, 2019. 874–886.
 - 13 Jeong S, Cho H, Lee S. Agile requirement traceability matrix. *Proceedings of the 40th IEEE/ACM International Conference on Software Engineering: Companion*. Gothenburg: IEEE, 2018. 187–188.
 - 14 Babaalla Z, Bouziane EM, Jakimi A, *et al.* From text-based system specifications to UML diagrams: A bridge between words and models. *Proceedings of the 2024 International Conference on Circuit, Systems and Communication (ICCSC)*. Fes: IEEE, 2024. 1–6.
 - 15 Hamza ZA, Hammad M. Generating UML use case models from software requirements using natural language processing. *Proceedings of the 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO)*. Manama: IEEE, 2019. 1–6.
 - 16 Maatuk AM, Abdelnabi EA. Generating UML use case and activity diagrams using NLP techniques and heuristics rules. *Proceedings of the 2021 International Conference on Data Science, E-learning and Information Systems*. Ma'an: ACM, 2021. 271–277.
 - 17 Alksasbeh MZ, Alqaralleh BAY, Alramadin TA, *et al.* An automated use case diagrams generator from natural language requirements. *Journal of Theoretical Applied Information Technology*, 2017, 95(5): 1182–1190.
 - 18 Arman N, Jabbarin S. Generating use case models from Arabic user requirements in a semiautomated approach using a natural language processing tool. *Journal of Intelligent Systems*, 2015, 24(2): 277–286. [doi: [10.1515/jisys-2014-0092](https://doi.org/10.1515/jisys-2014-0092)]
 - 19 Herchi H, Abdessalem WB. From user requirements to UML class diagram. *arXiv:1211.0713*, 2012.
 - 20 More P, Phalnikar R. Generating UML diagrams from natural language specifications. *International Journal of Applied Information Systems*, 2012, 1(8): 19–23. [doi: [10.5120/ijais12-450222](https://doi.org/10.5120/ijais12-450222)]
 - 21 Rigou Y, Lamontagne D, Khriess I. A sketch of a deep learning approach for discovering UML class diagrams from system's textual specification. *Proceedings of the 1st International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*. Meknes: IEEE, 2020. 1–6.
 - 22 Omer OSD, Eltyeb S. Towards an automatic generation of UML class diagrams from textual requirements using case-based reasoning approach. *Proceedings of the 4th International Conference on Applied Automation and Industrial Diagnostics (ICAAID)*. Hail: IEEE, 2022. 1–5.
 - 23 Jahan M, Abad ZSH, Far B. Generating sequence diagram from natural language requirements. *Proceedings of the 29th IEEE International Requirements Engineering Conference Workshops (REW)*. Notre Dame: IEEE, 2021. 39–48.
 - 24 Elallaoui M, Nafil K, Touahni R. Automatic generation of UML sequence diagrams from user stories in Scrum process.

- Proceedings of the 10th International Conference on Intelligent Systems: Theories and Applications (SITA). Rabat: IEEE, 2015. 1–6.
- 25 Alami N, Arman N, Khamyseh F. A semi-automated approach for generating sequence diagrams from Arabic user requirements using a natural language processing tool. Proceedings of the 8th International Conference on Information Technology (ICIT). Amman: IEEE, 2017. 309–314.
- 26 Sharma R, Gulia S, Biswas KK. Automated generation of activity and sequence diagrams from natural language requirements. Proceedings of the 9th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE). Lisbon: IEEE, 2014. 1–9.
- 27 Jin DM, Jin Z, Chen XH, *et al.* MARE: Multi-agents collaboration framework for requirements engineering. arXiv:2405.03256, 2024.
- 28 Hong S, Zheng X, Chen J, *et al.* Metagpt: Meta programming for a multi-agent collaborative framework. Proceedings of the 12th International Conference on Learning Representations. Vienna: ICLR, 2024. 1–29.
- 29 Hoy Z, Xu M. Agile software requirements engineering challenges-solutions—A conceptual framework from systematic literature review. Information, 2023, 14(6): 322. [doi: [10.3390/info14060322](https://doi.org/10.3390/info14060322)]
- 30 Wang BA, Wang C, Liang P, *et al.* How LLMs aid in UML modeling: An exploratory study with novice analysts. Proceedings of the 2024 IEEE International Conference on Software Services Engineering (SSE). Shenzhen: IEEE, 2024. 249–257.
- 31 Svahnberg M, Gorschek T, Nguyen TTL, *et al.* Uni-REPM: A framework for requirements engineering process assessment. Requirements Engineering, 2015, 20(1): 91–118.

(校对责编: 李慧鑫)