



基于 Agentic 机制的电信客服系统设计方法

吴珉杰, 包晔诚

(上海理想信息产业(集团)有限公司, 上海 200135)

摘要: 提出了一套具有自主性、灵活性的智能化客服系统设计方法。对 AI 领域中的 Agentic 具体机制进行了深入研究, Agentic 是一种组织和管理工作流程的方法, 这种方法特别注重自动化、智能化和自主性。这个概念的核心在于使系统能够自主决策和行动, 从而提高效率、减少人为干预, 并适应不断变化的环境。在研究 Agentic 机制的基础上, 提出了应用于电信客服系统的 Agentic 组织形式, 设计了包括 workflow 管理、基于异常的意图识别、工具接口智能调用等组件, 能很好地适用于智能化电信客服场景, 并具有一定的跨行业推广前景。

关键词: Agentic; 意图识别; 智能客服

中图分类号: TP393

文献标志码: A

doi: 10.11959/j.issn.1000-0801.2025062

0 引言

随着信息技术的飞速发展, 通信行业正面临着前所未有的挑战和机遇。技术的进步和用户需求的多样化, 使得传统的电信智能客服系统已经无法满足现代商业的需求。因此, 开发一种新型、智能化的电信客服系统变得迫在眉睫。本研究提出了一种基于 Agentic 机制的电信客服系统架构, 旨在通过自动化、智能化和自主性的方法来提升通信服务的效率与品质, 减少人工干预, 并适应不断变化的环境。

Agentic 机制作为一种新型的组织和 workflow 管理方法, 强调系统的自主决策和行动能力。通过引入 Agentic 机制, 实现根据用户的真实意图来选择相应的处置流程或信息反馈, 从而更好地满足用户需求, 提升服务质量和效率。

本研究首先深入探讨了 Agentic 机制, 分析

了其核心思想和优势。然后, 结合电信客服领域的特点, 提出了一种基于 Agentic 机制的电信客服系统架构, 并详细阐述了其设计理念和实现方法。最后, 通过实际案例分析, 验证了该电信客服系统架构在智能化通信服务场景中的适用性和有效性, 并探讨了其跨行业推广的前景。

本研究成果对于推动电信客服系统向智能化、自动化方向发展具有重要的理论和实践意义。通过引入 Agentic 机制, 通信企业可以构建更加高效、智能的服务系统, 提升用户体验, 降低运营成本, 从而在激烈的市场竞争中脱颖而出。同时, 本研究成果也为其他行业在服务领域的智能化转型提供了有益的借鉴和启示。

本研究旨在探讨一种基于 Agentic 机制的客服服务系统架构, 以应对当前电信客服领域面临的挑战, 提升通信服务的智能化水平。希望通过本研究, 能为通信服务系统的未来发展提供新的



思路和方法。

1 Agentic 机制

在阐述 Agentic 机制之前，首先让我们来回顾一下什么是 AI Agent。OpenAI 在 2023 年 6 月发表的 *LLM Powered Autonomous Agents* 一文中定义了目前 AI Agent 的主流技术框架，即 AI Agent 应由 LLM 作为整个 Agent 的大脑。

整个 Agent 应具备以下几种能力。

(1) 针对任务的规划能力：能够将复杂的问题拆解为更小的、可管理的子任务，同时具备从过去的问题中进行学习以改良未来动作的能力。

(2) 长期与短期的记忆能力：能支撑多轮交互，以及基于历史记录调整生成结果的相关能力。

(3) 工具使用能力：能够根据交互自主选择调用外部 API，比如获取实时的信息或者处理代码的能力等。

Agentic 相较于上述的 AI Agent 则更进了一步，它不再局限于被动执行指令，而是能够主动感知环境、理解目标，并自主选择行动方案，表现出不同程度的自主性和智能性，能够使整个系统或者组织具备 Agent 的特性。

Agentic 机制的核心在于将 AI Agent 的自主性和智能性提升到一个新的高度，使系统能够在更广泛的应用场景中展现出类似于人类的决策和行动能力。在 Agentic 机制中，系统不仅能够被动地响应外部指令，还能够主动地感知环境变化，理解复杂的目标需求，并自主地选择和执行最优的行动方案。

Agentic 机制的关键特点包括如下几点。

(1) 环境感知能力：Agentic 系统能够通过各种传感器和输入设备感知外部环境的变化，包括用户的行为、系统的状态、外部事件等。

(2) 目标理解能力：系统能够理解复杂的目标和需求，包括用户的意图、业务规则、系统目标等，并能够将这些目标转化为可执行的行动

计划。

(3) 自主决策能力：基于对环境的感知和对目标的理解，Agentic 系统能够自主地做出决策，选择最佳的行动方案，以实现既定目标。

(4) 行动执行能力：系统能够执行决策结果，包括调用外部 API、执行内部操作、与其他系统或 Agent 交互等。

(5) 学习和适应能力：Agentic 系统能够从过去的经验和交互中学习，不断优化决策模型，提高决策质量和效率。

(6) 多 Agent 协作能力：在复杂的业务场景中，Agentic 系统通常需要与其他 Agent 或系统协作，共同完成任务。因此，多 Agent 协作能力是 Agentic 机制的一个重要组成部分。

通过引入 Agentic 机制，电信客服系统可以实现更高的自动化和智能化水平，提供更高效、更个性化的服务。但是，由于电信客服是面向公众的服务端口，输出内容对于安全性的要求非常高，在交互环节如果完全交给 AI 实现，有可能出现舆情事故，因此本文对 Agentic 整体进行了一定的改造，在安全性与自主性间取得了平衡，在保证安全性的前提下极大地提升了客服系统的智能性。

2 系统设计方法

2.1 系统设计

在电信客服的实践中发现，真实用户的需求往往五花八门，涵盖业务的办理、咨询、投诉等多种场景，传统的客服系统主要通过机器学习或深度学习结合规则进行流程的流转，因此对交互场景往往有较大的限制，并不支持灵活的流程跳转与打断反馈，但是这些情况在真实的交互中又是非常常见的，这些限制造成了转人工率的上升。

完全基于 AI Agent 的方法虽然灵活性好，但是 Agent 的机制造成了每次交互都需要多次生成，

导致交互时延高到不可接受，同时完全基于 AI Agent 来进行生成无法保证输出内容的安全性与可靠性。因此，我们基于 Agentic 的机制与思想，结合传统客服系统，设计了一种智能化的电信客服系统，使该系统能在较低交互时延水平的情况下，能够输出可靠交互信息，同时支持灵活的流程切换与任意的插话、反悔场景。一种智能化的电信客服系统设计如图 1 所示。

该系统主要由基于 Agentic 的工作流以及执行/反馈两个大的部分组成。其中工作流由传统的工作流管理框架作为底座，通过可视化的方式对业务流程进行配置，例如，在用户订购套餐场景中，需要设置初始问候环节、信息采集环节、产品推荐环节以及订单生成环节，每个环节除了传统的配置信息，需要额外增加一个关于该环节的文本描述，该描述后续将为基于异常的意图识别 Agent 作为环节转换判断的参考信息。同时，不同的场景也是基于意图识别在交互中实现转化的，具体实现方式将在后续章节中阐述。

执行/反馈阶段将依托于大模型的结构化输出能力，根据意图来实现特定工具/接口的调用与传参，根据工具/接口的输出进一步进行与用户的交互，例如订单生成环节中，需要将用户信息以及

对应订购套餐信息发送到生产系统中的订单系统。

2.2 基于异常的意图识别

基于异常的意图识别主要是应用于交互进行时未按照计划进行的情况，此时启用该机制，进行主动的意图识别，判断用户的真实意图来决定后续是否要阻断/暂停当前的交互流程，并根据用户意图给予相应的反馈，包括但不限于转换场景、跳转环节、阻断服务、回答咨询问题等。

基于异常的意图识别组件设计逻辑如图 2 所示。

为了方便对流程进行说明，我们首先来定义场景和环节这两个概念。

(1) 场景：是指包含一个或若干个业务交互步骤的集合 A，该集合通常指一个独立的业务，例如新户办理、改性能、问题咨询等。

(2) 环节：是指在场景中的一个具体的业务步骤，即集合 A 中的一个元素，该元素通常是业务中的各环节，例如，在新户办理场景下的用户信息采集、该性能场景下的当前套餐信息确认等。

接下来逐个阐述该组件中的核心设计思想。整体流程如下。

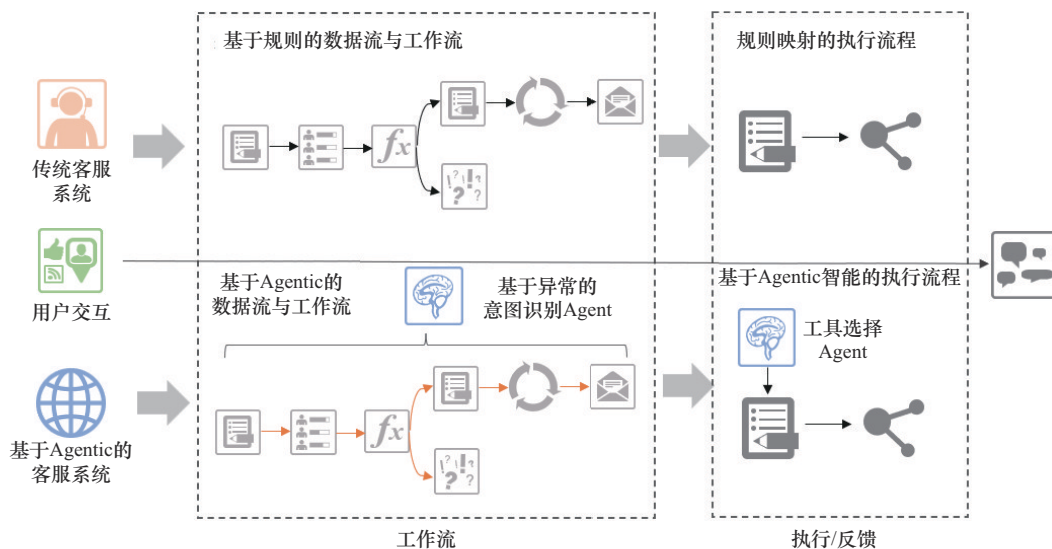


图 1 一种智能化的电信客服系统设计

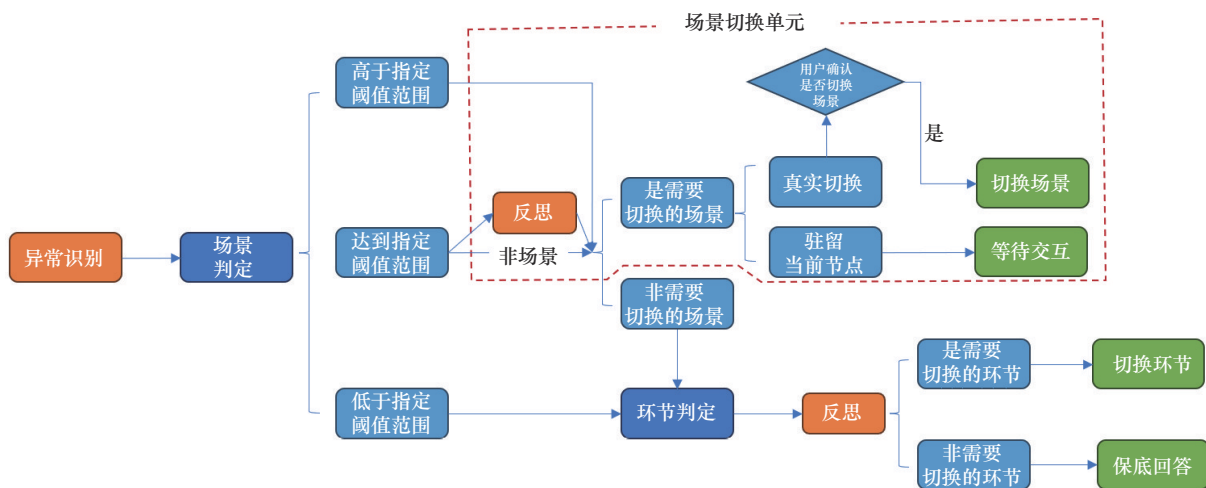


图2 基于异常的意图识别组件设计逻辑

(1) 当判定为异常时，首先进入场景判定。

(2) 当高于场景判定的阈值范围时，进一步判断是否真的要阻断当前流程，如果只是暂停，则进行暂停操作并等待后续交互，如果是切换场景，则需要与用户确认，如果不切换场景则进行环节判定。

(3) 当达到指定阈值范围时，需要增加一步反思（self-refine），以反思结果来确定是否要进行后续场景切换，如果不切换则进行环节判定。

(4) 低于指定阈值范围时，进行环节判定，首先使用 LLM 进行反思，判断是否需要切换，如果需要则进行环节跳转，如果不需要则进行保底回复，避免大模型自主生成引发安全问题。

以下为各组件设计方法。

(1) 异常识别。在客服流程中，异常主要有两种表现形式，在问题询问过程（信息收集）中未获取相关信息，或者是在正常的处置流程中未进行相关确认/否认，这两种主要异常形式显然可以通过交互过程中使用规则进行判定。

(2) 场景判定。在判断为异常后，即进入场景判定环节。在实践中，该组件可以进行渐进式的实现，在项目初期使用提示工程的形式让 LLM 来进行判断；在项目中期，已累积了一定的交互数据时，可以采用 RAG (retrieval-

augmented generation) 辅助，以求得到更加精准的结果，同时该方式也可以支撑较为复杂的业务场景，例如，在电信客服实践中，实际涉及的各类子场景数量是非常多的，单纯依赖大模型无法做到精确地选择识别，但是 RAG 可以做到，同时也能够实现快速的扩展，这也是本文推荐的方式；项目后期，场景固化后，可以利用 LLM 的微调训练机制，将 LLM 转变为一个专用的分类器，该方式是以牺牲灵活性来换取更高的场景识别准确性，在我们的实践中该方式可以达到 98% 以上的准确率。

(3) 环节判定。该部分实践过程与场景判定是类似，只是初始输入的业务知识是环节相关的描述信息。

(4) 阈值判定。该组件仅当场景判定是使用 RAG 机制时才使用，根据实际情况划定阈值，该参数会较大地影响后续的处置过程，需要进行大量的实验来确定三档阈值的划定。

(5) 反思。反思是在 NeurIPS 2023 上首次被提出的一种基于 LLM 的新机制，其主要目的是使用 LLM 自主对问题进行类似于人类的反思，通过进行更细致的思考来提升回答问题的准确率，在本系统中我们将这一思想应用于场景/环节的判断阶段，对初步确定的场景或环节使用 LLM

进行进一步的反思确认，以提升场景/环节识别的准确率。反思机制如图3所示。

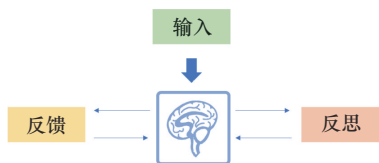


图3 反思机制

2.3 workflows 管理

workflows 管理的主要作用是将业务流程结构化，可以根据指示来调起相应的每个业务节点，如图4所示，同时集成了部分新的AI组件，例如基于LLM的智能信息采集器、基于知识图谱的推荐组件等。

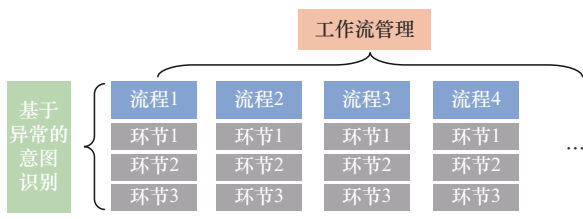


图4 workflows 管理

整个 workflows 管理框架提供了一个集中式自动化管理平台，可以使用预定义的规则和逻辑来设计、执行和监控 workflows 。该框架可以将业务流程分解为一系列定义好的任务和角色，然后按照预定的规则和条件来推动这些任务和角色的执行。以下为 workflows 管理框架的主要构成。

(1) 流程设计器：允许用户通过图形界面设计流程模型，通常使用BPMN（业务流程模型和符号）或其他流程建模语言。

(2) workflows 引擎：核心组件，负责解析流程模型，控制流程的流转，任务分配，以及状态管理。

(3) 任务列表：为不同的用户或角色提供待办任务列表，用户可以在这里看到分配给他们的任务。

(4) 表单设计器：用于设计在流程执行过程中需要收集或显示的数据表单。

(5) 业务规则引擎：用于处理流程中的决策逻辑，可以根据预定的业务规则进行判断和分析。

(6) 数据库：存储流程定义、流程实例、用户任务、业务数据等。

(7) 用户界面：供最终用户与 workflows 软件交互的界面。

本方法中涉及的 workflows 管理框架除了包含上述这些传统的核心组件，为了配合 Agentic 机制还设计了一些专用的智能化组件。

(1) 智能信息采集。基于大模型的结构化输出能力，根据采集内容不同而变更配置，进而影响内置的提示模板，实现对于多轮交互信息的灵活信息采集。

(2) 分级信息管理。针对多轮交互场景中涉及的结构化信息进行管理。一类信息，如个人信息，采集后采取被动更新策略，变更场景后继续保留。二类信息在变更场景后即清空。

(3) 知识图谱推荐。将运营商相关产品组织成知识图谱的形式，根据采集的需求信息进行关联推荐。

(4) 智能流转。传统的工作流管理工具是线性执行已编排的工作流，本方法中增加了智能流转，用基于异常的意图识别组件来识别下一步需要执行的工作流环节，之后再由智能流转组件发起对 workflows 的干预，实现灵活的流转，以适配实际的灵活场景，如用户反悔操作、更改录入信息、中途变更需求内容等。

(5) 智能工具/接口调用。将所需的工具及接口，例如，客服系统中常见的订单接口、地理信息接口等加入 workflows 处理环节，并通过自然语言对该部分进行相关描述。通过LLM的结构化输出能力构建需要传输的参数，以此来实现工具/接口的调用，来完成交互的执行环节，形成交互闭环。



3 结束语

本文提出的新型的智能电信客服系统设计方法，基于 Agentic 机制与 workflow 管理相结合，使整个客服系统具有 Agentic 所需要的自主性、灵活性与智能性。在实践中，我们对比了基于该方法设计的系统与传统的客服系统，该系统在交互流程中的意外场景明显体现出了更好的处置能力，而传统的系统则基本不具备此能力。该系统的使用大大降低了转人工率，经过模拟场景的验证，整体转人工率预计较传统系统能够降低 34.6%。

本系统模型证明了该方法的可实施性和可部署性，能够满足灵活的电信客服场景的需求，同时该系统还具有一定的自我演进特性。

参考文献：

[1] MADAAN A, TANDON N, GUPTA P, et al. Self-refine: itera-

tive refinement with self-feedback[J]. arXiv preprint, arXiv: 2303.17651, 2023.

[2] YE Y N, CONG X, TIAN S Z, et al. ProAgent: from robotic process automation to agentic process automation[J]. arXiv preprint, arXiv: 2311.10751, 2023.

[3] CHEN W Z, SU Y S, ZUO J W, et al. AgentVerse: facilitating multi-agent collaboration and exploring emergent behaviors[J]. arXiv preprint, arXiv: 2308.10848, 2023.

[4] HAN X, HU L X, DANG Y B, et al. Automatic business process structure discovery using ordered neurons LSTM: a preliminary study[J]. arXiv preprint, arXiv: 2001.01243, 2020.

[5] BROWN T, MANN B, RYDER N, et al. Language models are few-shot learners[J]. arXiv preprint, arXiv: 2005.14165, 2020.

[作者简介]

吴珉杰（1987-），男，上海理想信息产业（集团）有限公司工程师，主要从事大数据及 AI 产品研发及应用工作。

包晔诚（1985-），男，现就职于上海理想信息产业（集团）有限公司，主要从事大数据及人工智能领域产品研发及行业应用推广工作。