Enterprise Integrated Agent Framework

AI Agent辅助下的ESB整合方案

概述

1. 设最原子化Agent为AtomAgent，一招ReAct模式不停循环调用工具，并最终得到反馈。该AtomAgent可自己编写，也可来自任意Agent Framework，比如google。
2. 通过设定提示词，上下文等实例化一个特定功能BizAgent。作为企业Agent目标，每个BizAgent都是一个特定应用的代理。如人事应用Agent为HrAgent，该agent输入足够的上下文描述该Hr系统，并描述所有的API及URL，以确保HrAgent理解了该Hr Application。同理，一个企业会有很多BizAgent，PurchaseAgent, ContractAgent, FinanceAgent,….这些BizAgent组成了一个企业的Agent群，通过AgentCard录入到A2A Server供PlanAgent发现
3. PlanAgent:采用reflex模式，通过类似蒙特卡洛选择方式先走一个plan，若干该plan无法成功走通，则启用另一个plan。PlanAgent有两种驱动模式：
   1. PrePlan Mode：人为设计Plan（由若干Task组成）
   2. SelfPlan Mode：通过提示词自动设计Plan
   3. 对于PrePlan与SelfPlan Mode， PlanAgent都有ResilencePlan，一旦出发了Fail Event，PlanAgent将根据上下文和任务要求重新做一次SelfPlan
4. Plan:
   1. 每个Plan关联一系列BizAgent：对每个Agent配置相应的提示词，关心的状态，关心的上下文等
   2. 每个Plan有一系列Tasks
   3. 每个Task有一系列Status：NotStarted, Running, Done, Error, Pending
   4. 每个状态附加一个Context，该chontext是Task状态改变后的状态值集合
   5. 每个BizAgent观察对应Task状态，满足其运行条件，该BizAgent将根据观察状态的上下文及该BizAgent在Plan中的指令执行。该BizAgent的执行后，驱动对应任务的状态变更，并输入对应Context
5. plan detail

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Prompt/related plan | Status | Context |
| 任务名称 | 任务的详细描述，以上下文方式输入给Target Task对应的BizAgent / 同时该Task可以对应一个Plan B，那么这个plan B就变成了当前Plan的一个子plan。该任务的状态对应Plan B中的Main Task状态 | NotStarted | Running |  Done | Error | Pending  其中error可以驱动一个补偿agent，也可以驱动planner重新plan | {status:done, values:{…}}描述任务当前状态及一系列状态值 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Agent | Prompt | Observing Tasks | Trigger Condition | Target Task&Status |
| plan中介入的agent | 描述该Agent在该Plan中的详细任务描述 | 关心的任务列表：一旦这些任务状态改变，将会检查Trigger Condition是否满足条件 | Agent启动条件，一旦该条件满足，Agent将启动执行，启动上下文包括：Prompt，Task Context | Agent执行结束后将要更新的Task Status |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Plan | Prmpt | Main Task | Tasks | Agents |
| plan名称，id | plan描述 | 该Task的启动，表示了该Plan的启动，该Task.Done表示该次plan执行结束 | 对应Task列表 | 对应Agent列表 |

1. Example

新员工入职：(所有Tools是该系统公布的API MCP化)

HrAgent（Tools：create\_employee\_profile …）

FinanceAgent（Tools：enroll\_payroll …）

ITAgent（Tools：apply\_computer …）

InventoryAgent(Tools: outbound,check\_outbound\_status …)

AccessAgent（Tools：grant\_access …）

|  |  |  |
| --- | --- | --- |
| Plan | Prmpt | Main Task |
| ID:101  新员工入职计划 | 新员工入职将会在hr系统中注册新员工，发送欢迎消息；在财务系统中创建工资单；并且在IT系统中领取电脑；最后在门禁系统里申请门禁 | 001 |

|  |  |
| --- | --- |
| Task | Prompt/related plan |
| 001  新员工完成入职 | 新员工入职 |
| 002  新员工注册 | 新员工在Hr系统中完成新员工注册 |
| 003  申请电脑 | 新员工在IT系统中完成电脑申请 |
| 004  电脑出库 | 电脑出库 |
| 004  门禁申请 | 新员工在门禁系统里完成门禁权限申请 |

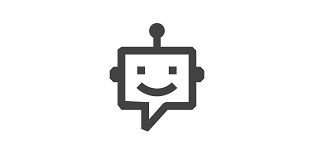
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Agent | Prompt | Observing Tasks | Trigger Condition | Target Task&Status |
| HrAgent | 根据员工的身份证号，姓名信息在HR 系统中创建员工记录 | 001: context{id:…, name:…} | 001.status == Running | 002.status = Done  002.context = {emp\_id:…,level=…} |
| ITAgent | 根据根据员工ID，员工Level申请相应的笔记本 | 002:context{emp\_id, level} | 002.status==Done | 003.status = NotStarted |
| InventoryAgent | 根据IT系统的电脑申请，新建出库单，等待员工领取 | 003:context{pc\_mode:…} | 003.status==NotStarted | 003.status = Pending |
| InventoryAgent | 轮询员工电脑领取状态，如果已经领取，结束出库任务 | 003: context{pc\_no:.. status:…} | 003.status==Pending | 003.status = Done |
| AccessAgent | 根据员工ID为新员工申请门禁 | 001: context{id:…, name:…} | 001.status == Running | 004.status = Done |
| Planner | 检查新员工入职手续是否都已完成 | 003, 004 | 003.status==Done && 004.status==Done | 001.status=Done |

\*轮询为BizAgent的标准设计能力，用于扑捉到外部系统操作产生的关键状态变化

**001**

**Running**

**Main:新员工入职**



**Hr**

**HR系统注册**

**002**

**003**

**004**

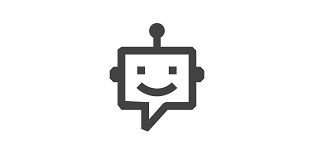
**005**

**IT 电脑申请**

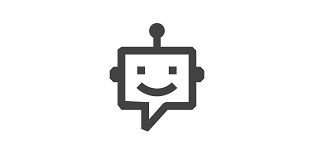
**Inventory出库**

**Access门禁**

**Done**



**IT**



**Inv**

**Running**

**create emp**

**apply pc**

**done**

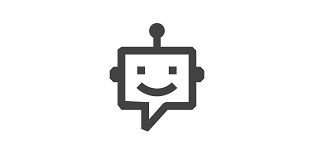
**Pending**

**create order**

**轮询**

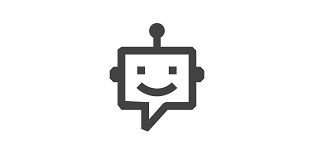
**领取电脑**

**done**



**Access**

**done**



**Planner**

**Done**

**1**

**2**

**3**

**4**

**4.1**

**3.1**

**3.2**

**3.3**

**5**

**？**

1. 设计要点

1. 任务驱动：BizAgent根据Task状态与上下文驱动执行。  
2. 补偿机制：错误触发补偿Agent或Planner重规划。  
3. 轮询机制：作为标准能力，支持捕获外部系统的异步状态变化。  
4. 组合模式：任务可嵌套Plan，使复杂流程可层层分解。