



2025 AI+ Development
Digital Summit

AI+研发数字峰会

拥抱AI 重塑研发

05/23-24 | 上海站





2025 AI+研发数字峰会

拥抱AI 重塑研发 AI+ Development Digital Summit

下一站预告

08/08-09 | 北京站

11/14-15 | 深圳站



查看会议详情

北京站论坛设置

大模型和 AI 应用评测

智能存储与检索技术

下一代知识工程

AI+ 金融业务创新

智能需求工程

智能体与研发效率工具

AI 产品运营与出海策略

大模型安全与对齐

大模型应用开发框架与实践

智能体经济 (Agentic Economy)

智能测试工具的开发与应用

具身智能与机器人

代码生成及其改进

AI+ 新能源汽车

AI 前沿技术探索与实践

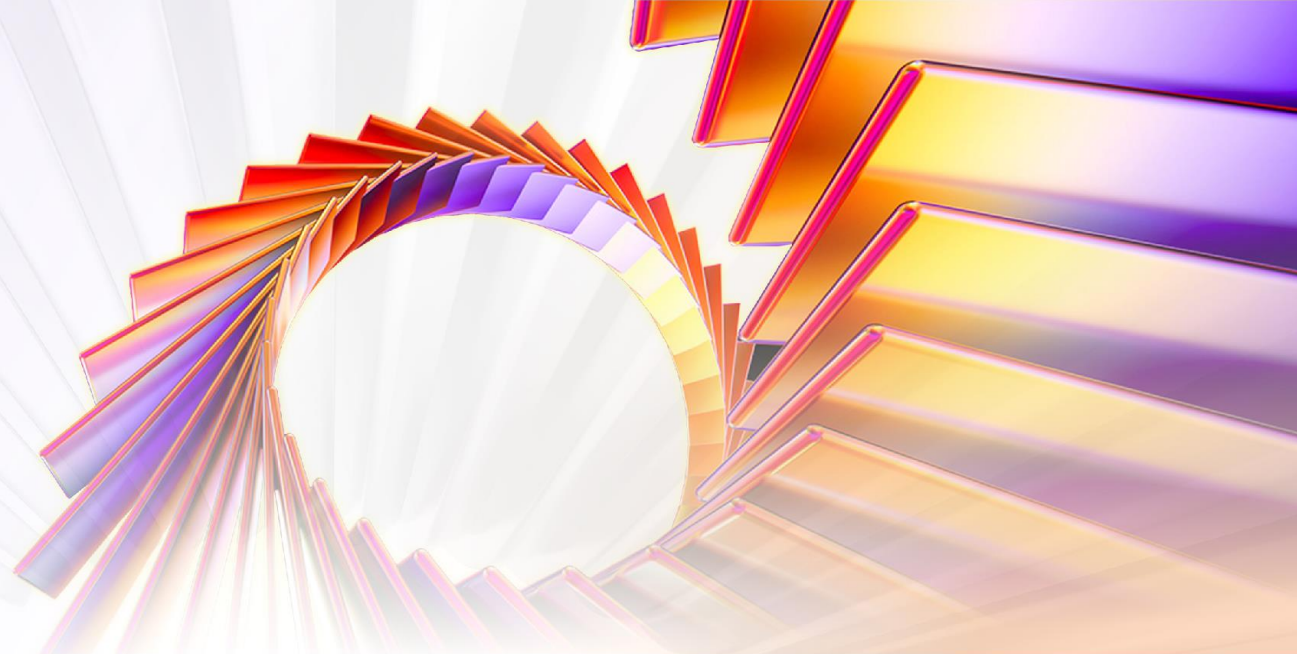


| 05/23-24 | 上海站

2025 AI+ Development
Digital Summit

AI+研发数字峰会

拥抱AI 重塑研发



脚本生成2.0

基于智能体的脚本生成与修复之道

陈超 | 中兴通讯股份有限公司



陈超

中兴通讯青年奖得主/数通产品测试经理

数通产品测试经理、资深测试专家，近十年来一直从事数通产品软件测试的工作，牵头负责有线研究院自动化脚本AI生成方案制定与落地推广。2025年，数通产品自动化脚本累计生成代码行数82W+行，个人脚本开发效率提升43%

目录

CONTENTS

- I. 问题提出
- II. 解决思路
- III. 方案介绍
- IV. 实践过程&效果
- V. 总结与展望

PART 01

问题提出

24年挑战

承接产品数**增加83%** (6->11)

- 测试设计需求数：增加25% (1574->1965) ；
- 执行用例数：增加104% (38031->77827)
- 手工执行用例数：增加20% (6500->7783) ；
- 流水线执行用例数：增加122% (31531->70044) ；
- 人均脚本开发数：增加84% (2.6个/天>4.8个/天) ；
- 流水线失败分析数：增加82% (550->1000) ；

路径分析

类别	23Q4 (现状)	24Q4 (预计)	效能提升目标	可达路径	说明
测试设计需求数	1574	1965	25%	存量用例复用提效14%	探索AI辅助测试设计，基于本地知识库实现场景推荐
				后备TSE分担11%	
手工执行用例数	6500	7783	20%	特性终结在领域提效15%	/
				领域测试环境配置提效5%	
测试开发效率 (人天)	2.6	4.8	84%	存量脚本复用提效25%	19%的提效路径未找到
				AI辅助脚本开发提效40%	
自动化执行 用例数	31531	70044	122%	静态资源夜间纳管提效50%	/
				每周产品差分运行提效39%	
				补充1000核资源提效33%	

测试开发工作流

阶段	脚本编写	环境准备	脚本调试	脚本入库
工作量分布 (单条平均)	75分钟	10分钟	95分钟	5分钟
占比	40%	5%	52%	3%

需求测试开发最耗时的活动是脚本编写、脚本调试

方案1.0 基于RAG+FewShot技术实现从用例生成脚本 提升脚本编写和脚本调试的效率

测试开发
流程

用例标准化

脚本编写

脚本调试入库

1 用例标准化

手工完成
测试用例改写

2 命令行拆分

拆分命令行配置
(与关键字1对1)

查找关键字

关键字参数编写

合并关键字

3 生成脚本片段

RAG+FewShot

查找匹配关键字
选择TOP1

查找脚本示例
动态组装提示词

向大模型提问
生成脚本片段

4 脚本组合

脚本片段
合并和排序

脚本模拟执行
校验准确率

替换关键字参数为
拓扑图变量名

5 脚本调试入库

脚本入库

脚本执行
调试修正

可执行脚本

AI
应用

应用
系统

自研星云通信大模型

自研一体化测试平台

▶ 方案1.0应用效果 项目脚本开发效率提升100%

数通测试的底层逻辑：

- **数通产品的测试动作的本质：**设备命令、仪表命令的组合
- **手工测试的本质：**通过人脑将自然语言表述的文本用例进行设备命令组合的转义
- **中台关键字的本质：**建立了每一个设备命令到唯一原子关键字的关联关系
- **自动化测试的本质：**将设备命令的组合通过原子关键字的组合实现对文本用例的脚本转义

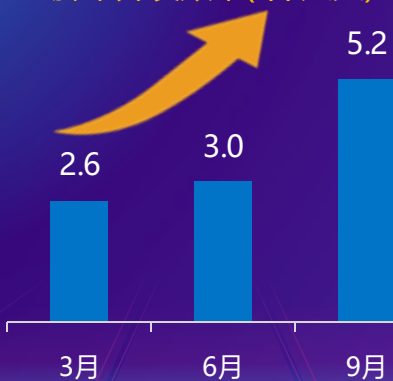
关键技术：

- **知识库：**基于积累的6K+原子关键字，2.5W+存量脚本，低成本快速构建知识库
- **确定性输入：**手工测试完成对被测对象猜想的验证，确定性输入，结果可预期
- **配置拆分：**分割配置文件为原子配置，提升模型解析精度，突破长文本限制
- **经验加速：**存正确脚本于记忆库，重复任务快速调用，提升响应速度，减轻AI负担

脚本开发效率提升

100%

脚本开发效率(个/人天)



AI脚本生成效率提升

40倍

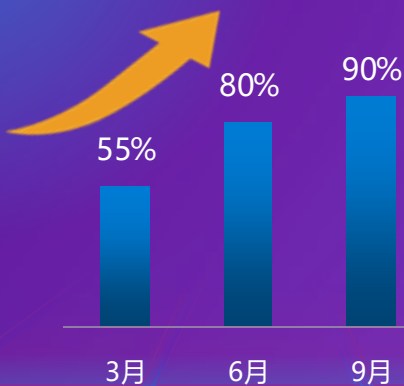
AI脚本生成时间(秒/个)



AI脚本生成准确率提升

50%

AI脚本生成准确率



► 方案1.0问题 脚本生成采纳率提升依赖于人的参与度 无法替代人完成测试执行工作

人精准定义任务，大模型才能高效工作

存在的问题

现状：

- 方案1.0需记录手工执行的命令和日志，将文本用例转换为大模型可理解格式，才能生成高采纳率的脚本

根因：

- 缺乏统一的标准，自然语言的多样性和复杂性给大模型带来了额外的认知负担和处理挑战

用例非标准化表达，大模型理解困难

现状：

- 测试开发涵盖配置、校验等多个环节，每脚本平均需40个关键字和300个参数，开发易错且效率低；方案1.0要求人工预分配任务内容，大模型完成指定任务，辅助人完成测试开发工作

根因：

- 大模型不理解领域知识情况下，任务复杂度越高，生成错误概率越大



人的参与度，决定了方案1.0生成脚本的采纳率



问题提出 如何才能让大模型“替代”人完成测试执行工作？

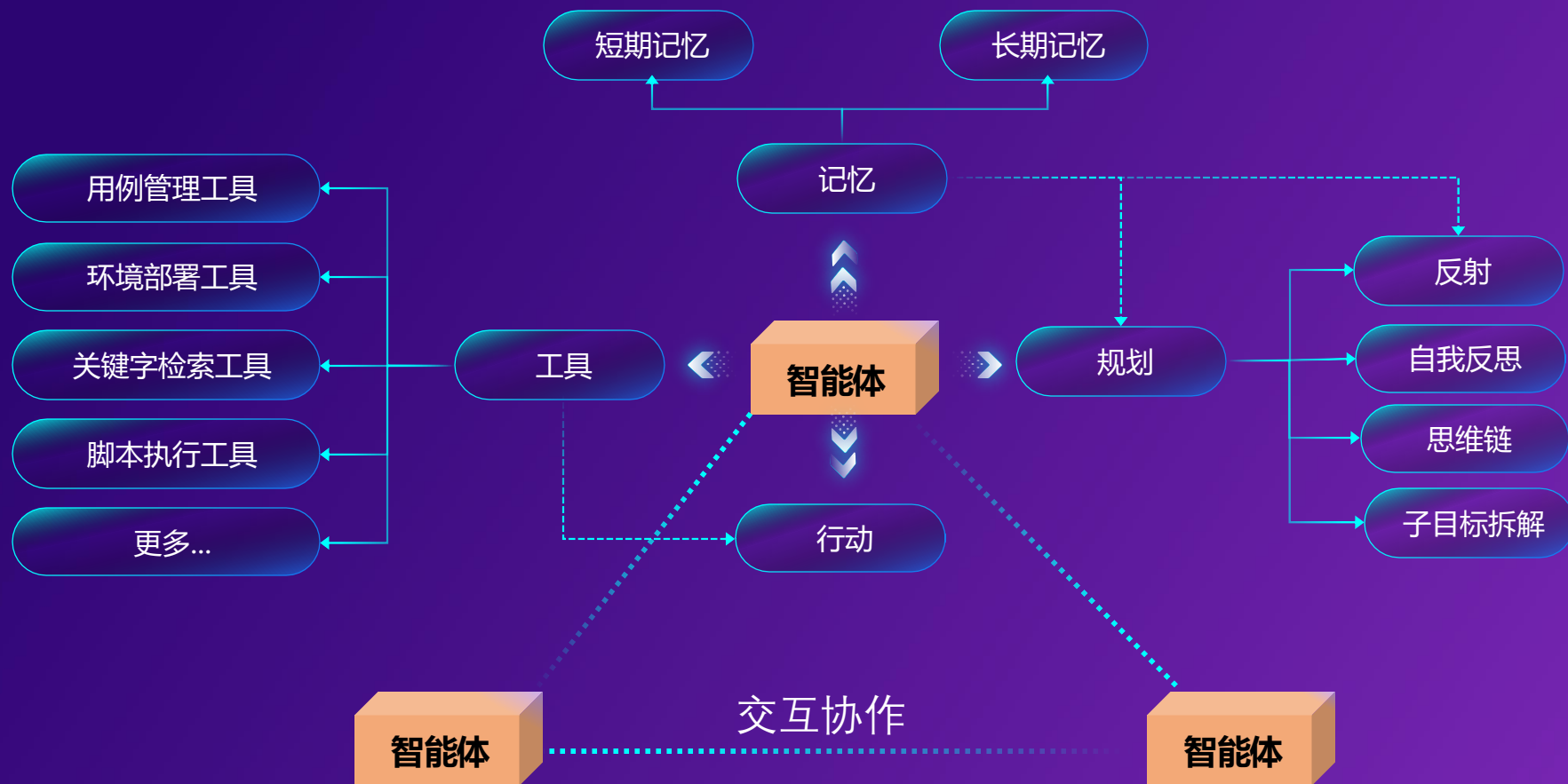


当前大模型不具备测试开发需要的全部能力

PART 02

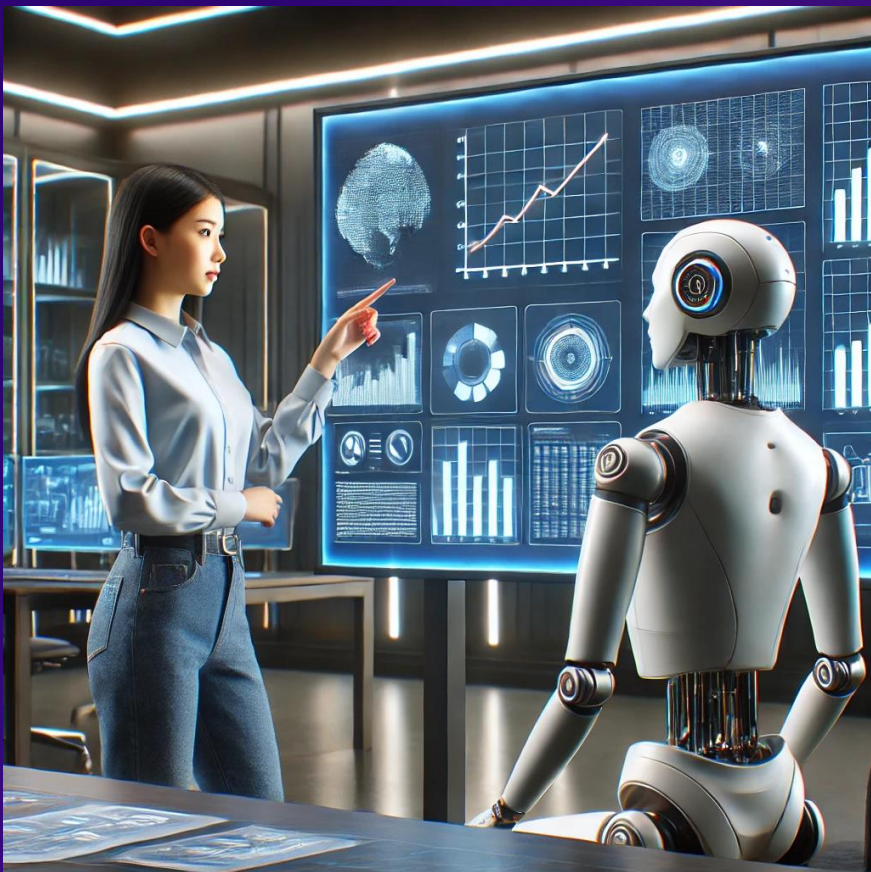
解决思路

智能体的能力 智能体让大模型具备了“替代”人完成测试开发执行的全部能力



智能体具备学习与适应、规划和决策、推理、使用工具的能力

脚本生成2.0方案的核心是将智能体视为团队成员



新手期：人对智能体工作流程进行指导，并对结果进行分析和反馈，智能体进行反思和修正。

成长期：智能体通过持续的学习，积累经验（记忆），可以独立完成部分测试开发工作

成熟期：智能体可以独立解决未知领域的问题，实现替代人完成测试开发工作

► 方案升级 从“个体”到“团队”

方案1.0

人作为团队成员
使用工具（大模型）完成工作

- 人获取任务，理解用例，拆分任务；
- 人提供配置命令、标准化用例；
- 人使用脚本生成工具完成脚本编写
- 人理解脚本内容，申请环境并执行；
- 人分析失败日志，明确脚本问题；
- 人完成脚本问题修正

更智能

更高效

方案2.0

人作为团队Leader，下发任务
智能体作为团队成员，调用工具完成工作

- 人获取任务，人下发任务，**智能体**拆分任务；
- **智能体**获取知识，完成用例标准化改写；
- **智能体**调用脚本生成工具完成脚本编写；
- **智能体**理解脚本内容调用工具申请环境并执行脚本；
- **智能体**分析失败日志，明确脚本问题；
- **智能体**完成脚本问题修正；
- 人完成交付物质量确认；

方案1.0存在的问题

- 大模型不理解领域知识情况下，**任务复杂度越高，生成错误概率越高**
- **用例缺乏统一的标准**，自然语言表达增加了大模型的认知负荷和处理难度



方案2.0关键技术

- **基于SOP固化工作流程**
 - 1) **规范AI处理过程**，复杂任务拆解为规范的子任务和处理流程，允许具有专业知识的智能体来验证中间结果
 - 2) 增强过程的可解释性，利用智能体**反思能力**实现过程错误自修复
- **基于意图识别的用例标准化**
 - 1) 基于复杂的业务逻辑，提炼抽象通用的**表达范式**
 - 2) 利用**思维链推理**，分步骤完成文本用例到标准化用例的生成

PART 03

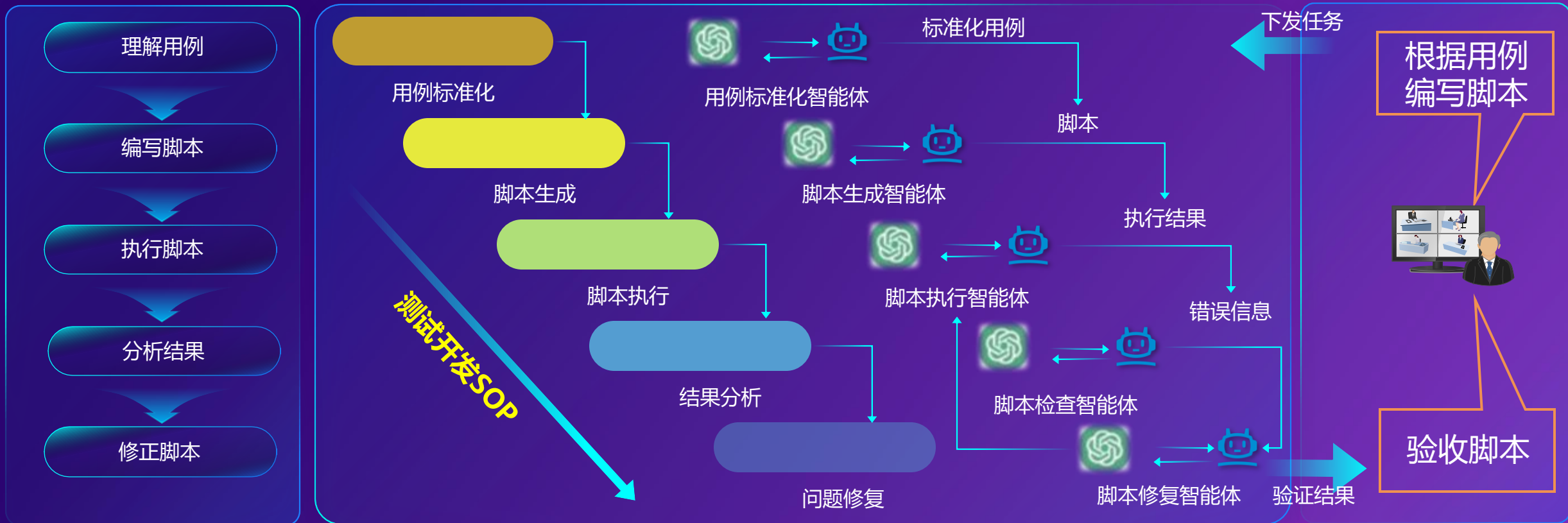
实践方案

方案2.0 基于SOP的多智能体系统 帮助智能体实现“替代”人完成测试执行工作

测试开发流程

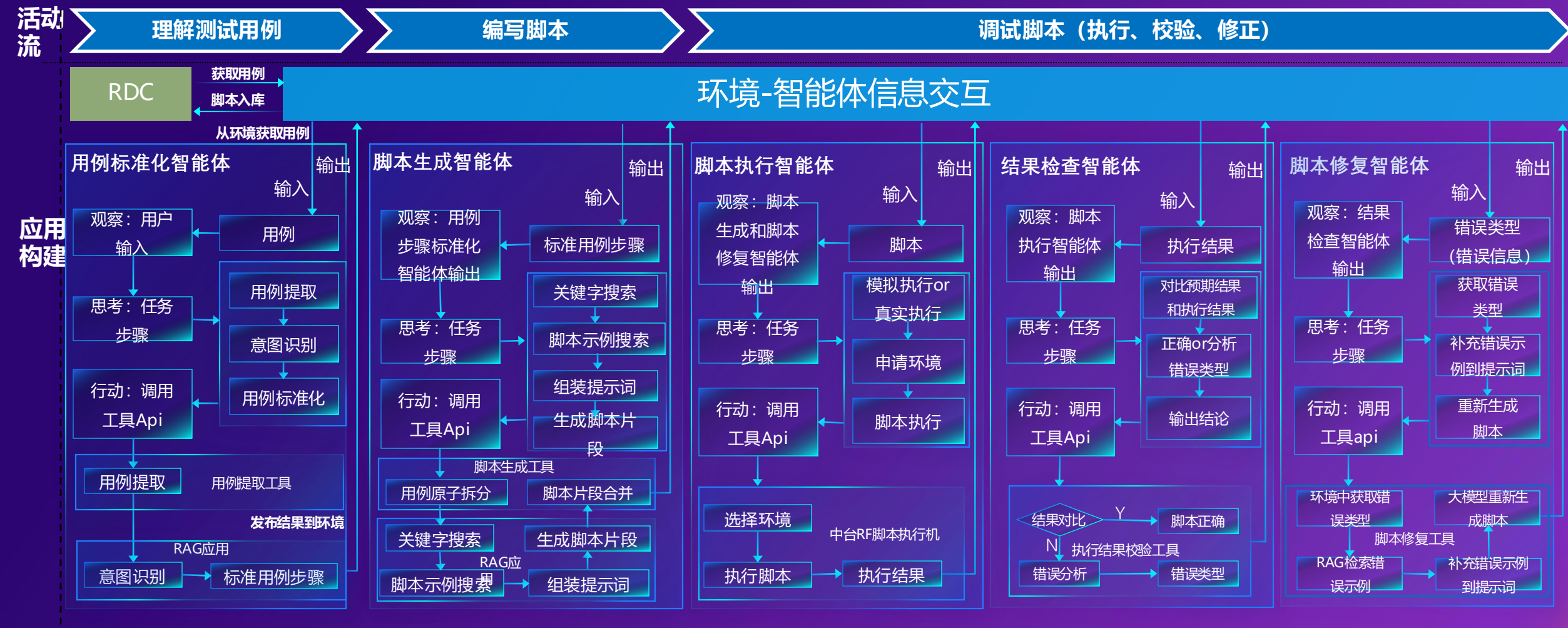
基于智能体协作的测试开发SOP

人机交互



将复杂任务流程标准化 (SOP)，分阶段和按步骤完成整个工作流，降低单智能体对大模型理解、规划能力的要求

方案2.0智能体 workflow 测试开发智能体实现从用例到脚本自动生成



用例标准化关键技术

基于意图识别 解决自然语言表达大模型理解困难的问题

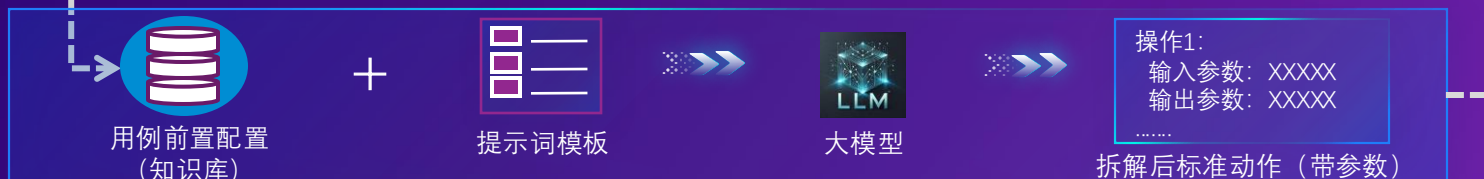
①意图识别



②匹配标准动作



③参数补全

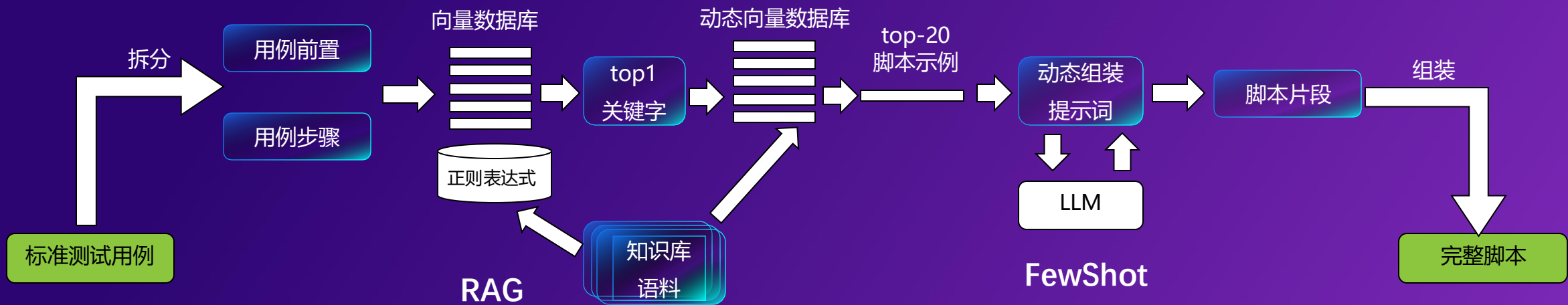


④用例组装



基于思维链推理，通过意图识别、标准动作匹配、参数补全、用例组装逐步完成生成标准化测试用例

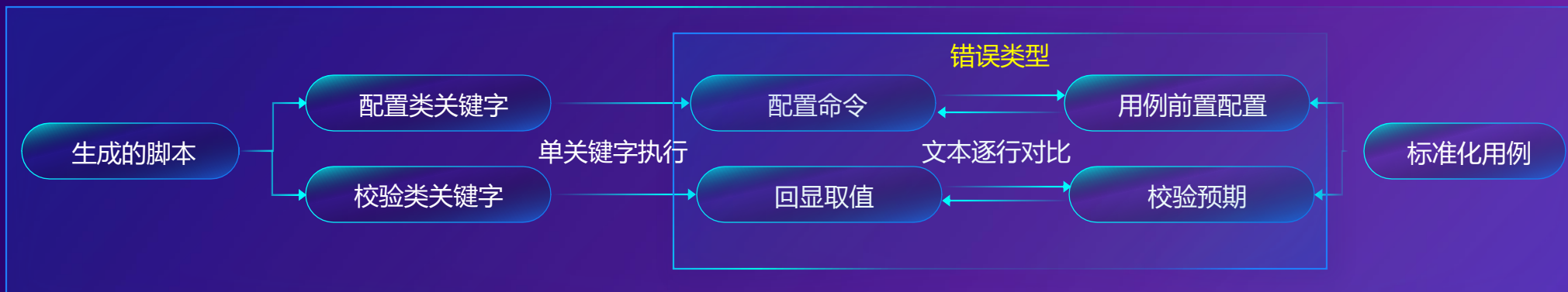
脚本生成关键技术 复用1.0方案 实现脚本自动生成



智能体将用例拆分为前置和步骤，利用RAG定位脚本示例，结合标准步骤和示例动态构建FewShot提示词，生成脚本

脚本执行\检查关键技术 分级执行，资源有效利用

①模拟执行 (关键字级)

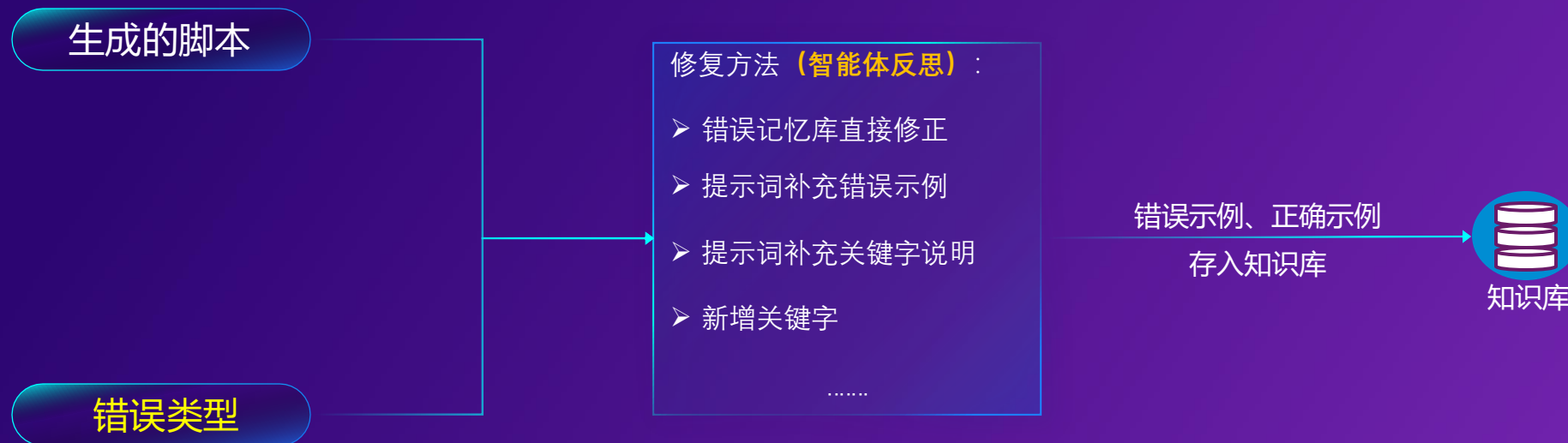


②真实执行 (脚本级)



“关键字级”模拟执行，快速识别80%生成错误，省时省资源；脚本级执行，识别健壮性问题

脚本修复关键技术 智能体根据脚本执行结果进行反思 完成错误修复



- **反思修复：**根据经验积累和动态记忆（智能体从环境中感知到的信息），对历史的推理决策进行反思
- **知识积累：**“过程知识”存入知识库，将仅靠沟通传播的知识，结构化的管理起来，降低知识流转成本

知识质量测评 自动评测 识别当前知识建设节点和模型能力匹配度



- 从产品配置手册提取语料，经工具清洗转换，TSE完成评估后抽取为业务场景库、意图-标准动作库的知识
- 基于产品配置手册，基于大模型构建评测集，利用自动评测，识别当前知识建设节点和模型能力匹配度
- TE对用例标准化Agent和脚本生成Agent输出的内容进行人工审核和调试，对存在质量问题的知识进行修正

测试开发工程化 嵌入研发流程

全流程管理



PART 04

实践过程&效果

①获取用例信息

②用例标准化

③生成脚本

④测试执行

1.自动获取文本用例内容

数据中心-测试执行

使用帮助 问题反馈

选择产品:

15K

用例

TC-EvpnVxlan4-分布式三层转发-FUN-000501 【u4o4】 untag子接口映射lrb接口接入evpn隧道三层转发测试 (untag报文)

+ 添加用例

用例标准化

流量模型

流量名称	流量模型	发送端口	接收端口	选配	操作
stream1	L3业务单播IPv4报文	TCenter--port1....	TCenter--port4....	配置选配	删除
stream2	L3业务单播IPv4报文	TCenter--port4....	TCenter--port1....	配置选配	删除
+					

测试步骤

step1:TC1和TC4分别发起arp请求。查看解析结果。

result1:TC1和TC4的arp解析成功。DUT1和DUT2上查看lrb1接口arp学习正常

step2:TC1上发送三层流量stream1;TC4上发送三层流量stream2。

result2:验证测试仪TC1能收到stream2;测试仪TC4能收到stream1;流量无丢包。

step3:在DUT1的vxlan vpls实例下删除ac侧接口然后重新配置绑定:

result3:配置正确

step4:再次发送stream1和stream2

result4:验证测试仪TC1能收到stream2;测试仪TC2能收到stream1;流量无丢包。

【u4o4 EvpnVxlan分布式对称转发】

1、接口链路层：
DUT1的port2作为自身sg1的成员口,DUT2的port2作为自身sg1的成员口，配置参考【Smartgroup接口Active模式】
DUT1的port3作为自身sg1的成员口,DUT2的port3作为自身sg1的成员口，配置参考【Smartgroup接口Active模式】
DUT1的sg1作为自身sg1.4001的成员口，DUT2的sg1作为自身sg1.4001的成员口，配置参考【Vlan子接口】
DUT1的port1作为自身port1.101的成员口，配置参考【UntagVlan子接口】
DUT2的port4作为自身port4.102的成员口，配置参考【UntagVlan子接口】

2、公网路由层：
DUT1的loopback1、sg1.4001和DUT2的loopback1、sg1.4001配置IP地址，配置参考【IP地址配置】，剩余接口不配置IP地址
DUT1的loopback1、sg1.4001和DUT2的loopback1、sg1.4001使能Ospf2，配置参考【Ospf2】，剩余接口不使能【Ospf2】

3、公网隧道层：此场景不需要配置

4、业务转发层
DUT1和DUT2的loopback接口配置BGP邻居，参考配置【u4o4 EvpnVxlan分布式对称转发】；
DUT1的port1.101和DUT2的port4.102作为access-point的接入口,剩余设备不能使能u4o4 EvpnVxlan分布式对称转发。

① 步骤标准化

② 生成脚本

③ 执行脚本

步骤标准化正在进行中...

步骤标准化 生成脚本 执行脚本

①获取用例信息

②用例标准化

③生成脚本

④测试执行

2.完成文本用例步骤标准化改写

数据中心-测试执行

选择产品:
15K

用例

TC-EvpnVxlan4-分布式三层转发-FUN-000501 【u4o4】 untag子接口映射irb接口接入evpn隧道三层转发测试（untag报文）

+ 添加用例

步骤标准化

流量关键字

step1

文本片段	类型	意图	标准动作	数据模版	标准步骤	操作
TC1发起arp请求	报文	测试仪发送arp请求	命令格式：【描述】测试仪<Host Name>.发送ARP 命令范例：【描述】测试仪Host1发送ARP	<Host Name>:	[推荐描述][描述] STEP1：测试仪host1发送ARP和ND	🗑️ 🔍
TC4发起arp请求	报文	测试仪发送arp请求	命令格式：【描述】测试仪<Host Name>.发送ARP 命令范例：【描述】测试仪Host1发送ARP	<Host Name>:	[推荐描述][描述] STEP1：测试仪host2发送ARP和ND	🗑️ 🔍
DUT1查看irb1接口arp学习情况	业务表项检查	检查arp学习	命令格式： 【业务表项检查】<被测设备>.show arp 检查ip address为<ip v4 addr>的arp条目接口信息为<接口名称>. 命令范例： 【业务表项检查】<被测设备>.show arp 检查ip address为20.1.1.2的arp条目接口信息为irb1	<被测设备>: <ip v4 addr>: <接口名称>:	[推荐描述][验证] RESULT1：DUT1 show arp 检查ip address为10.1.1.1的arp条目接口信息为irb1	🗑️ 🔍



实践过程

①获取用例信息

②用例标准化

③生成脚本

④测试执行

3. 标准化用例生成完整测试脚本

✓

步骤标准化

✓

生成脚本

步骤标准化

生成脚本

执行脚本

完整脚本

TC-FUN-000100 场景配置生成.robot

TC-FUN-000100 场景配置生成_help.robot

*** Settings ***
Documentation Description:
... 测试套自定义关键字
Resource ztesw/Master.robot
Resource zxrUFP/Master_zxrUFP.robot
*** Variables ***
\${Script_PATH} \${CURDIR}

*** Keywords ***
TC-FUN-000100 场景配置生成_setup
 log rest_setup
 # Started by AICoder, pid:beae3a74c10511efaba6287b09cb0588
 配置interface device=DUT1 interface=smartgroup1 no_shutdown=\${TRUE}
 配置lACP interface device=DUT1 interface=smartgroup1 lACP_mode=802.3ad
 配置interface device=DUT1 interface=\${[v]}[DUT1__interface__intf2__name] no_shutdown=\${TRUE}
 配置interface device=DUT1 interface=\${[v]}[DUT1__interface__intf3__name] no_shutdown=\${TRUE}
 配置lACP interface device=DUT1 interface=\${[v]}[DUT1__interface__intf2__name] smartgroup_mode=active
 配置lACP interface device=DUT1 interface=\${[v]}[DUT1__interface__intf3__name] smartgroup_mode=active
 配置interface device=DUT1 interface=smartgroup1.4001 no_shutdown=\${TRUE}
 配置vlan interface device=DUT1 interface=smartgroup1.4001 encapsulation_dot1q=4001
 配置interface device=DUT1 interface=\${[v]}[DUT1__interface__intf1__name] no_shutdown=\${TRUE}
 配置interface device=DUT1 interface=\${[v]}[DUT1__interface__intf1__name].101 no_shutdown=\${TRUE}
 配置vlan interface device=DUT1 interface=\${[v]}[DUT1__interface__intf1__name].101 encapsulation_untag=\${TRUE}

配置说明生成前置脚本

✓

步骤标准化

✓

生成脚本

步骤标准化

生成脚本

执行脚本

完整脚本

TC-FUN-000100 场景配置生成.robot

*** Settings ***
Resource TC-FUN-000100 场景配置生成_help.robot
Suite Setup Run Keywords Robot Initial AND MyScript Initial
Suite Teardown MyScript Teardown

*** Test Cases ***
TC-EvpnVxlan4-分布式三层转发-FUN-000501 【u4o4】 untag子接口映射lrb接口接入evpn隧道三层转发测试（untag报文）
 [Documentation]
 ... == 测试步骤 ==
 ... | STEP1 | 模板 |
 ... == 编写人 ==
 ... | |
 ... == 编写日期 ==
 ... | 2024.12.25 |
 [Tags] AI_生成 TYPE_功能测试 TOP_2X VERSION_通用 PRI_5 CD_1 分层_ST
 [Setup] TC-EvpnVxlan4-分布式三层转发-FUN-000501 【u4o4】 untag子接口映射lrb接口接入evpn隧道三层转发测试（untag报文）_setup
 # Started by AICoder, pid:fbdb238fc21511ef9b46287b09cb0588
 log STEP1.1: 测试仪host1发送ARP和ND
 测试仪Host发送ARP和ND TCenter HostName=host1
 log STEP1.2: 测试仪host2发送ARP和ND
 测试仪Host发送ARP和ND TCenter HostName=host2
 log RESULT1.3: DUT1 show arp 检查ip address为10.1.1.1的arp条目接口信息为lrb1
 Wait Until Keyword Succeeds 3min 5 检查ARP信息 device=DUT1 ip_address=10.1.1.1 get_info=interface expect=lrb1
 log RESULT1.4: DUT2 show arp 检查ip address为20.1.1.1的arp条目接口信息为lrb1
 Wait Until Keyword Succeeds 3min 5 检查ARP信息 device=DUT2 ip_address=20.1.1.1 get_info=interface expect=lrb1

标准化用例步骤生成脚本步骤



4.执行测试脚本并返回执行结果

脚本执行日志

SUITE UFP 0000001 15K-onTop.xml 20241225 09:30:01.748

Status: 1 test, 1 passed, 0 failed

Message: 6分52秒

SETUP BuiltIn.Run Keywords 导入提供全局变量, \${SUITE NAME}, GlobalVariables.json AND 导入提供全局变量, \${SUITE NAME}, SRGlobalVariables.json, AND 导入中台模块全局变量, -file, D:\bin\src\AutoCenter\lib\ux25k\integration\testProject\testCases\UFP\GlobalVariables.yml 20241225 09:30:02.344

SUITE Benmm 20241225 09:30:06.193

Status: 1 test, 1 passed, 0 failed

Message: 6分48秒

SUITE 场景生成 20241225 09:30:06.198

Status: 1 test, 1 passed, 0 failed

Message: 6分48秒

SUITE D7D5C7A0-94FF-11E8-B451-00163E0076C6 20241225 09:30:06.203

Status: 1 test, 1 passed, 0 failed

Message: 6分48秒

SUITE TC-FUN-000100 场景配置生成 20241225 09:30:06.209

Status: 1 test, 1 passed, 0 failed

Message: 6分37秒

SETUP BuiltIn.Run Keywords Robot initial, AND, MyScript Initial 20241225 09:30:35.859

版本检查 收集show version信息 20241225 09:30:57.876

TEST TC-FUN-000100 场景配置生成 20241225 09:30:57.868

VERSION ZXUFPV2.2040E15.15K Built on 2024/11/19 04:01:33

TEARDOWN TC-FUN-000100 场景配置生成_help.MyScript Teardown 20241225 09:36:36.374

配置项检查 收集配置项信息 20241225 09:36:42.376

脚本执行日志

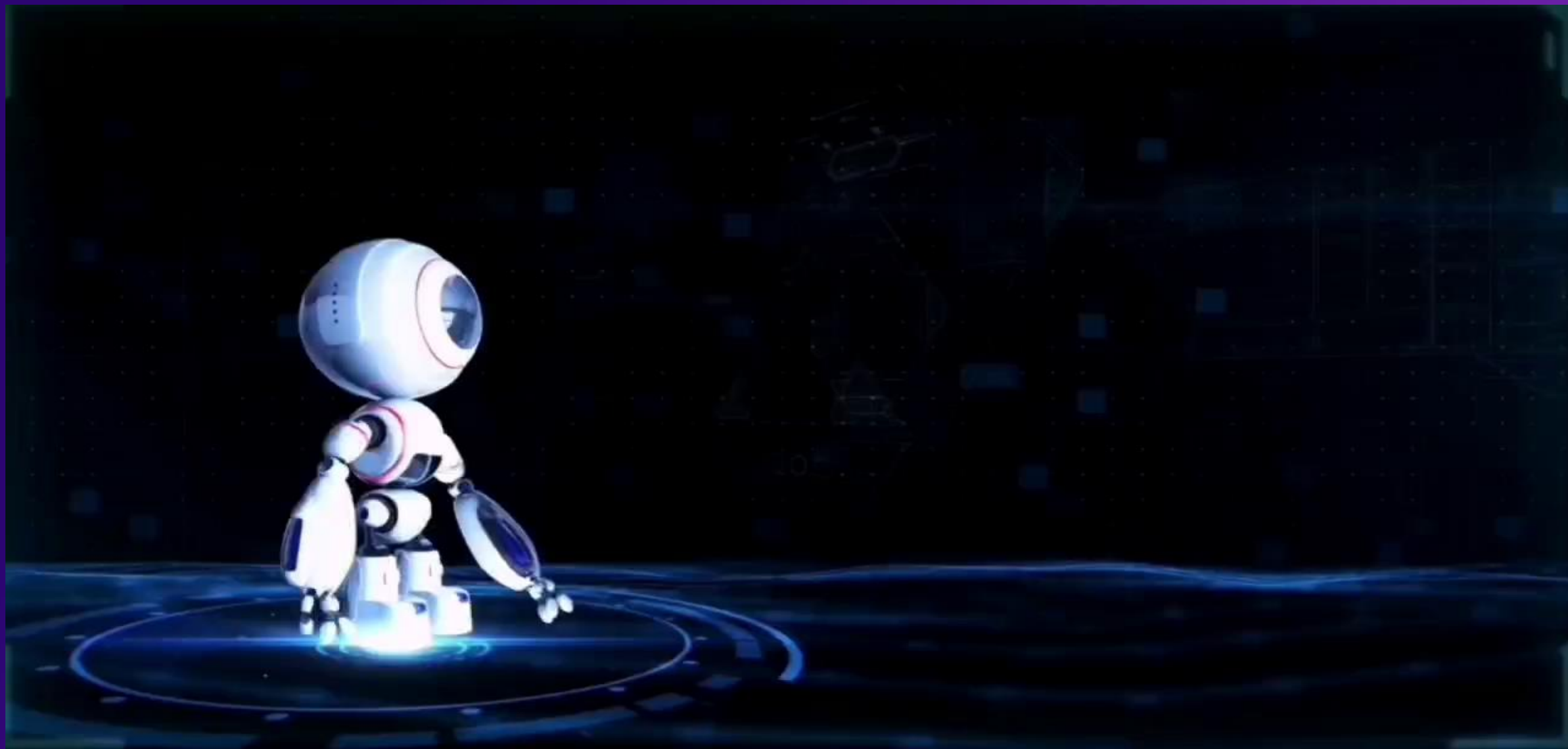
SUITE UFF 物理拓扑 15K-prjTop.xml	20241225 09:59:57.837
SETUP BuiltIn.Run Keywords 导入模块全局变量, \${SUITE NAME}, GlobalVariables.json, AND, 导入模块全局变量, \${SUITE NAME}, SRGlobalVariables.json, AND, 导入中台模块全局变量, -file, d:\bianzyt\AutoCenter\lib\zx15k\integration\TestProject\TestCases\UFF\Global_Variables.yaml	20241225 09:59:58.407
SUITE Benmm	20241225 09:10:02.198
SUITE 场景生成	20241225 09:10:02.209
SUITE D7D5C7A0-94FF-11E6-B451-00163E0076C6	20241225 09:10:02.217
SUITE TC-FUN-000100 场景配置生成	20241225 09:10:02.226
SETUP BuiltIn.Run Keywords Robot Initial, AND, MyScript Initial	20241225 09:10:32.258
版本检测 收集show version信息	20241225 09:10:55.011
TEST TC-FUN-000100 场景配置生成 VERSION ZXUFVPV2.20.40B15_15K Built on 2024/11/19 04:01:33	20241225 09:10:55.002
配置终端检测 收集设备配置信息	20241225 09:10:58.560
SETUP TC-FUN-000100 场景配置生成_help.TC-FUN-000100 场景配置生成_setup	20241225 09:11:09.718
KEYWORD 测试Host关键字, 测试配置UnTag IPv4 Host TCcenter, port_handle=intf1, HostName=host1, MacAddr=00:10:94:00:00:4d, Ipv4Addr=10.1.1.1, Ipv4Mask=24, Ipv4SubAddr=10.1.1.1	20241225 09:14:28.167
KEYWORD 测试Host关键字, 测试配置UnTag IPv4 Host TCcenter, port_handle=intf4, HostName=host2, MacAddr=00:10:94:00:00:4e, Ipv4Addr=20.1.1.1, Ipv4Mask=24, Ipv4SubAddr=20.1.1.1	20241225 09:14:28.829
KEYWORD 检查Interface关键字, 获取接口MAC地址并格式转换 device=DUT1, interface=irb1, format=colon	20241225 09:14:29.692
KEYWORD 测试流量类关键字, 测试配置流量和Traffic Profile TCcenter, port_handle=intf4, StreamName=stream1, FrameLen=150, EthSrc=00:10:94:00:00:4d, EthDst=\${DUT1mac}, VlanUserPriority=7, L2=Ethernet_Vlan, L3=ipv4, IpSrcAddr=10.1.1.1, IpDstAddr=20.1.1.1, TrafficLoad=\${tv}[random_stream_load2], TrafficLoadUnit=\${tv}[random_stream_loadunit]	20241225 09:14:30.105
KEYWORD 检查Interface关键字, 获取接口MAC地址并格式转换 device=DUT2, interface=irb1, format=colon	20241225 09:14:30.334
KEYWORD 测试流量类关键字, 测试配置流量和Traffic Profile TCcenter, port_handle=intf4, StreamName=stream2, FrameLen=128, EthSrc=00:10:94:00:00:4f, EthDst=\${DUT2mac}, VlanUserPriority=7, L2=Ethernet_Vlan, L3=ipv4, IpSrcAddr=20.1.1.1, IpDstAddr=10.1.1.1, IpSrcMask=255.255.255.0, TrafficLoad=\${tv}[random_stream_load2], TrafficLoadUnit=\${tv}[random_stream_loadunit]	20241225 09:14:30.923
KEYWORD BuiltIn.Log 步骤1: 测试Host1发送ARP和ND	20241225 09:14:31.179
KEYWORD 测试操作类关键字, 测试Host发送ARP TCcenter, HostName=host1	20241225 09:14:31.192
KEYWORD BuiltIn.Log 步骤2: 测试Host2发送ARP和ND	20241225 09:14:31.246
KEYWORD 测试操作类关键字, 测试Host发送ARP TCcenter, HostName=host2	20241225 09:14:36.627
KEYWORD BuiltIn.Log RESULT3.1: DUT1 show arp 检查ip address为10.1.1.5的arp条目接口信息为irb1	20241225 09:14:36.883
KEYWORD BuiltIn.Wait Until Keyword Succeeds 3min, 5, 检查ARP信息, device=DUT1, ip_address=10.1.1.5, get_info=interface, expect=irb1	20241225 09:14:41.416
KEYWORD BuiltIn.Log RESULT4.1: DUT2 show arp 检查ip address为20.1.1.5的arp条目接口信息为irb1	20241225 09:14:41.439



结果执行pass

实践过程 完成脚本错误自修复





实践效果 释放人力 提升质量

交付能力
提升项目

交付效能



需求交付效率
提升**76%**



自动化覆盖率
增加**65%**



测试吞吐
提升**88%**

交付质量

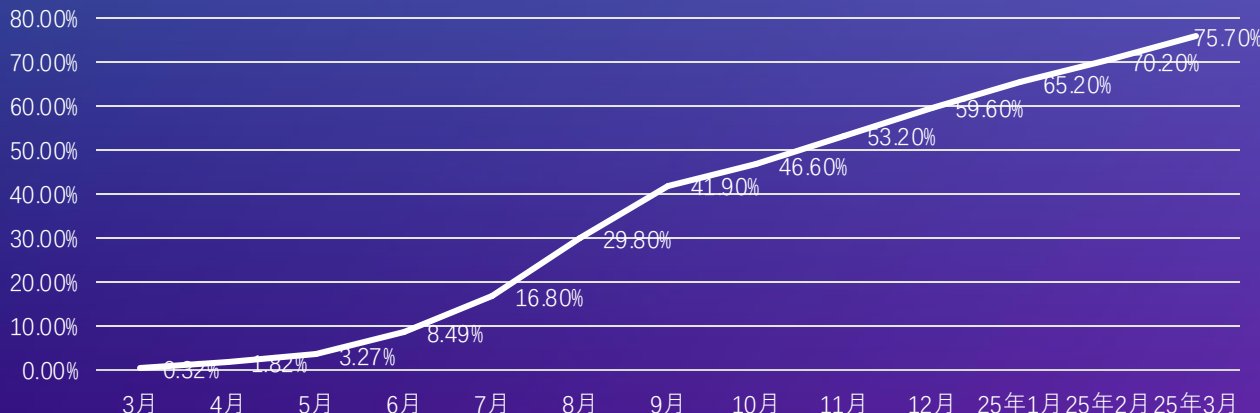


产品质量
提升**17%**



因脚本问题泄露的故障
零

脚本生成率趋势图



- AI应用覆盖到**200+**需求, **5K+**用例
- AI累计生成代码达到**82W+**行
- 脚本生成采纳率 > **90%**
- 2024全年脚本生成率呈**指数级增长**

收益越来越显著

PART 05

总结与展望

► 思考：如何才能真正提效？

指标尚可，提效的外部感知不明显

原因分析：

- 指标无法反应真实情况，生产过程依然是人机协作，人工审核、修正、验收的工作被忽视也难以准确统计
- 知识工程落地需要信仰，知识建设投入大，门槛高，反馈延时长，这类工作往往需要信仰，信仰是稀缺资源

关键变化

模型能力
肉眼可见的提升

评测识别问题，知识精准补充

改进思路：

- 大模型的生产力是模型能力和知识“化学反应”的结果，知识建设应当基于应用对象的能力变化而调整，教一个3年工作经验的骨干远比教一只有3个月经验的新人要容易得多
- 通过人机协作的过程，识别需要人工干预的问题，横向推理形成通用的范式来解决同类问题；通过将这些通用的范式纳入评测集，在生产前就进行知识全景的识别，精准补充



知识建设匹配模型能力，提效需要“量变”引起“质变”

从人类智能到人工智能

人类智能



人工智能



下一步

人工时代

- 人通过理解用例、查询关键字、复用旧脚本完成脚本编写
- 通过代码补全、断点调试等手段辅助编写和调试

1.0

- 人利用大模型的推理模仿能力，通过RAG和Fewshot技术，实现脚本生成
- 大模型替代人编写脚本，脚本编写效率大幅提升

2.0

- 人将现有的测试活动总结为标准的作业程序，并将复杂任务拆分为多个子任务
- 多个智能体基于标准程序各自完成子任务，并通过与人合作完成任务
- 利用人工评估反馈，大模型反思的方式提升任务完成的质量

- 大模型能力进一步提升，对产品有深入理解，更好的指令跟随能力以及反思能力
- 智能体随着模型能力提升，可以自主构建标准作业程序，智能完成工具的使用
- 智能体自主采集测试活动中生成的数据，伴随式完成模型训练，实现智能体自学习和自成长

智能科技，AI的新时代

参与调研您将优先获得



AiDD定制版
《AI+软件研发精选案例》



专属学习顾问
1对1需求对接

AiDD会后小调研

AiDD峰会致力于协助企业利用AI技术深化计算机对现实世界的理解,推动研发进入智能化和数字化的新时代。作为峰会的重要共建者,您的真知灼见对我们至关重要。衷心感谢您的参与支持!



扫码参与调研

2025 AI+研发数字峰会

拥抱 AI 重塑研发

科技生态圈峰会 + 深度研习

——1000+ 技术团队的选择



敦煌站

K+ 思考周®研习社

时间: 2025.08.29-30



上海站

K+ 金融专场

时间: 2025.09.26-27



香港站

K+ 思考周®研习社

时间: 2025.11.17-18



K+峰会详情



上海站

AI+研发数字峰会

时间: 2025.05.23-24



北京站

AI+研发数字峰会

时间: 2025.08.08-09



深圳站

AI+研发数字峰会

时间: 2025.11.14-15



AiDD峰会详情

AiDD 峰会



2025 AI+研发数字峰会

AI+ Development Digital Summit

感谢聆听!

扫码领取会议PPT资料

