OTA1-全场景语音对话管理模块设计文档

- 一、引言
  - 1.1 编写目的
  - 1.2 术语说明
  - 1.3 设计目标
- 二、可行性分析
  - 2.1 需求分析
  - 2.2 可行性分析
- 三、解决方案
  - 3.1 系统架构
    - 3.1.1 整体架构设计
    - 3.1.2 业务逻辑
  - 3.2 模块设计
    - 3.2.1 模块调用图
    - 3.2.2 调用时序图
    - 3.2.3 状态机流转示意图
    - 3.2.4 可否执行操作判断
    - 3.2.5 NLU结果融合
    - 3.2.6 中控最终融合结果
    - 3.2.7 例子
  - 3.3 需求设计
    - 3.3.1 语义打断
    - 3.3.2 TTS优化
    - 3.3.3 剧本、场景、全局的优先判断
- 四、实现方案
  - 4.1 接口设计
  - 4.2 数据结构设计
- 五、测试方案
  - 5.1 功能测试
  - 5.2 性能测试
  - 5.3 稳定性测试
- 六、开发计划

一、引言

1.1 编写目的

记录对话管理的架构设计、业务逻辑，及相关协议

1.2 术语说明

1. DST（dialogStateTracking）：对话状态追踪。职责：更新每一个任务的context信息，任务选择，是否结束。目前除场景外的对话状态在剧本中维护，但状态ID要在DST中更新。
2. 场景DP（sceneDialogPolicy）：场景与全局的PK逻辑

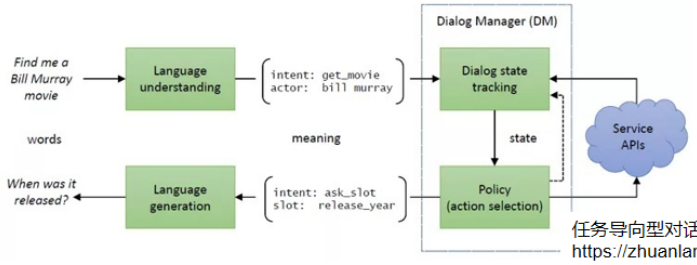
1.3 设计目标

现状与行业做法

- 现状：
  - 场景对话与全局对话完全隔离，边界性不强且互相影响
  - 单向结合query对话管理只能在NLU部分决策，方案不完备
  - 针对DUI和小爱同学的对话状态维护，没有清洗的解决方案

- 难点：
  - 对话管理如何获取执行结果
  - 如何进行可运维的任务选择
  - 如何挂起、取消、返回任务
  - 如何增强任务的可扩展性

- 行业做法：
  - 对话管理分为对话状态追踪、对话策略。区分场景的过程业内没有先例。bot的执行过程可以看做是api的调用，如下图：



任务导向型对话系统——对话管理模型研究最新进展 2020.1.3
https://zhuanlan.zhihu.com/p/100843827

1. 解决场景多轮与剧本的选择问题
2. 可配置的选择，例如GUI实现某功能前要走全局语义，实现其功能后要走场景语义

二、可行性分析

2.1 需求分析

1. 对话状态生命周期：
   a. 任务型：用户主动发起→小P引导澄清→信息完整度逻辑判断→结果执行→执行结果
   b. 问答型
   c. DUI任务型
   d. DUI闲聊型
   e. 任务型对话：
      i. 用户主动发起 → 小P引导询问（optional） → 信息完整 → 执行 → 修改（optional） → 过期（optional）→ 结束
      ii. 小P主动引导询问→ 信息完整 → 执行 → 修改（optional） → 过期（optional）→ 结束
   f. 问答/闲聊型对话：
      i. 用户主动发起 → 小P回答/闲聊 → 话题已无关 → 结束
2. 对话状态管理：
   a. 读取、选择栈内对话信息
   b. 转移、修改对话状态
   c. 存储对话信息
3. 对话策略决策：
   a. 选择NLU结果（NLU归一化）
   b. 选择Bot分发

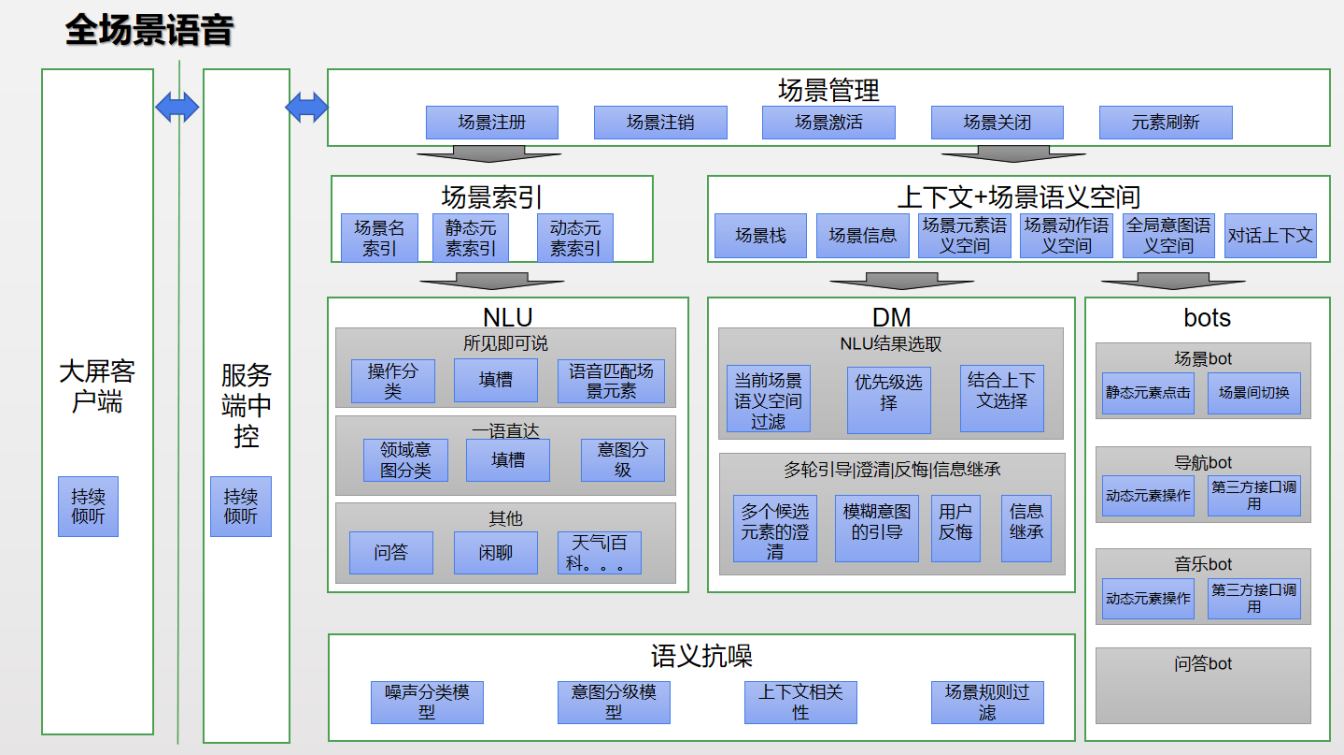
2.2 可行性分析

1. 剧本与对话状态均以状态机记录，可在类堆栈内统一
2. 小爱同学、DUI均可拆分为NLU+Bot

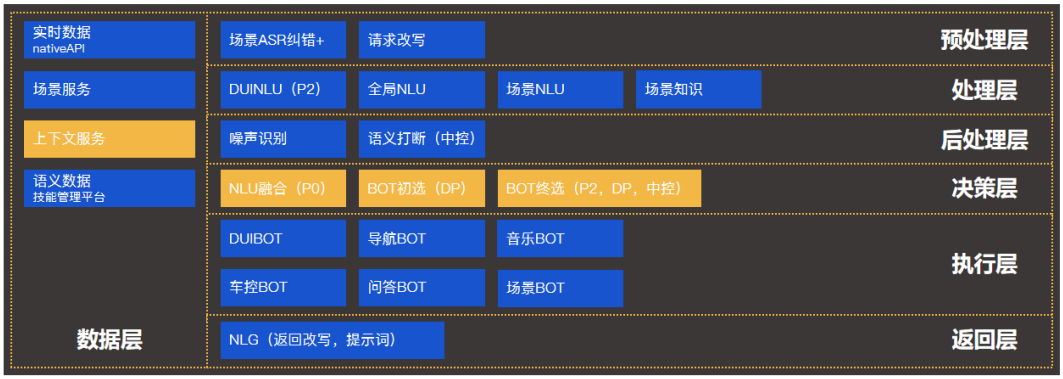
三、解决方案

3.1 系统架构

3.1.1 整体架构设计



云端服务架构图

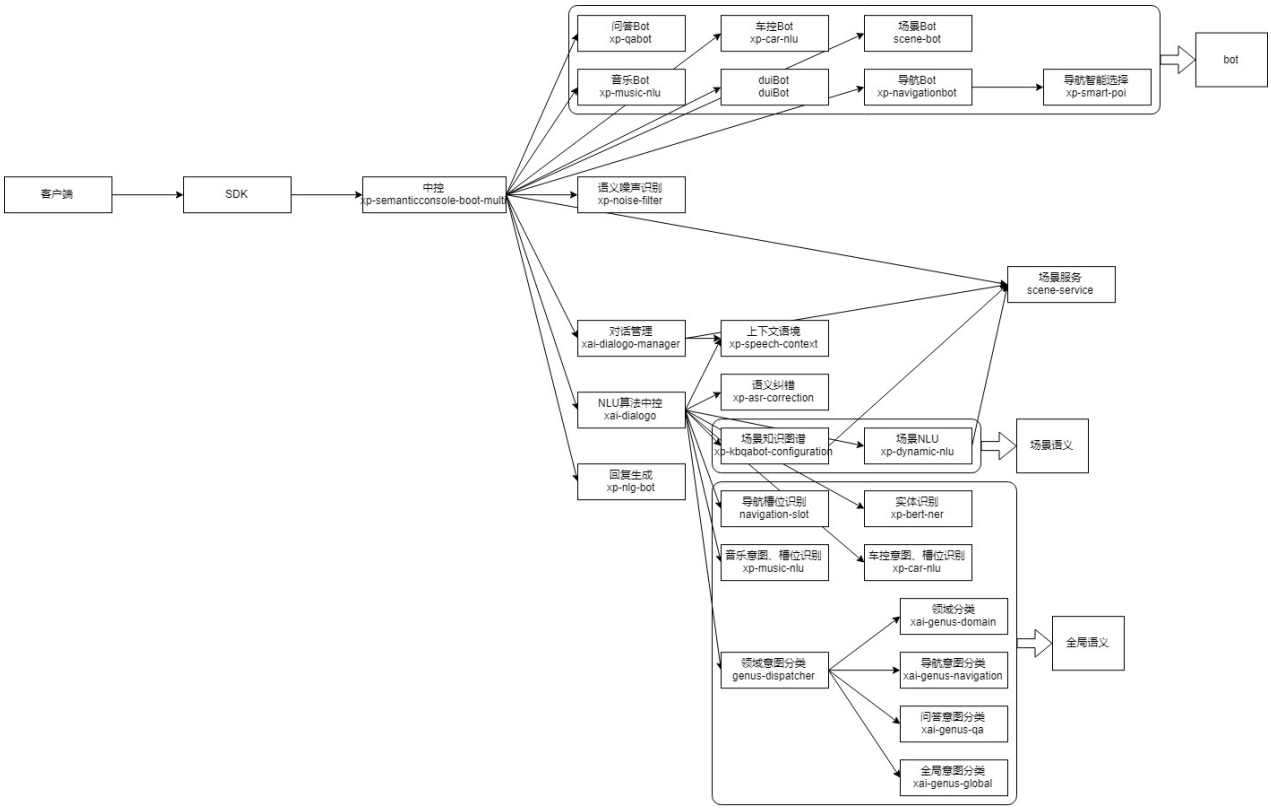


3.1.2 业务逻辑

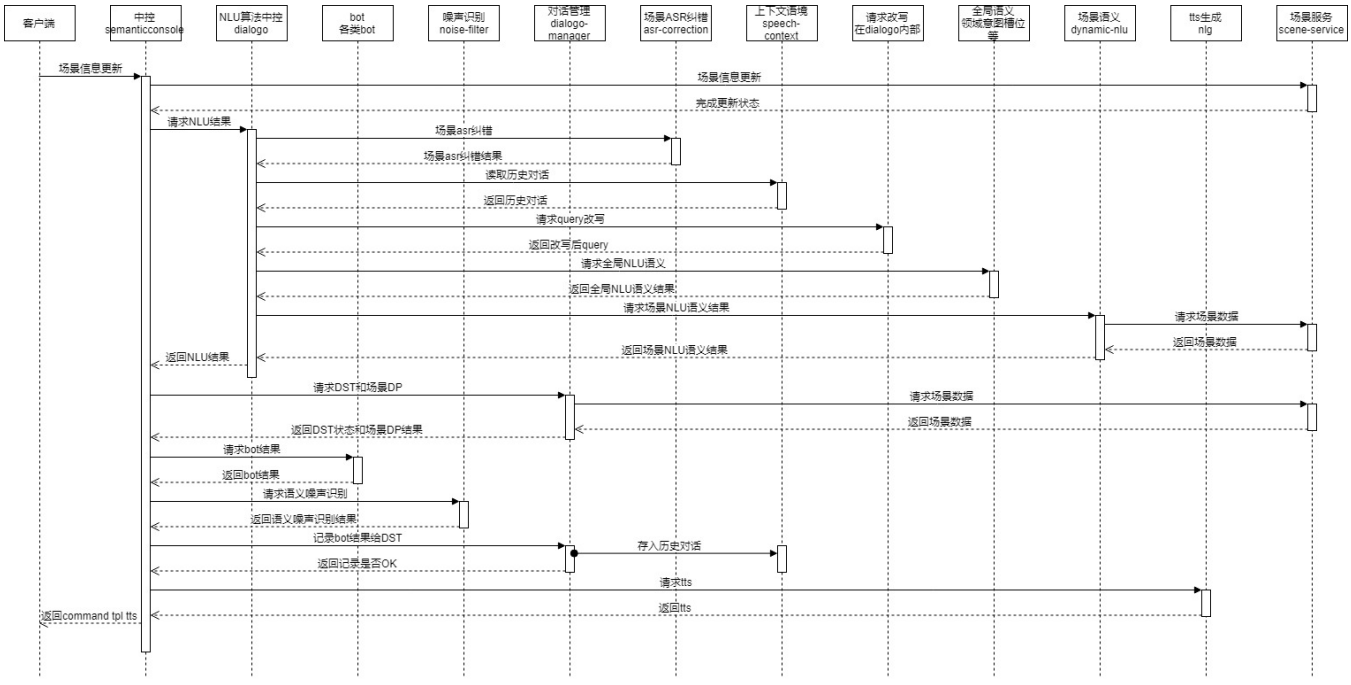
1. 操作合法性的判断 （直接忽略？默认修正？ 引导？）
    a. GUI-context
        i. 当前场景的语义范围判断
        ii. 当前场景元素的类别与action合法性判断
        iii. 当前场景元素的值的合理性判断
2. 模糊意图的澄清：
    a. 切换思必驰音色 ----》 切换到哪个音色（旧版林志玲 | 温柔女生 | 新版林志玲）
3. 有多个候选项的引导
    a. 切换XPNLU到测试环境 ---》 测试环境 ？ 测试环境多轮
4. 技术设计
    a. 对话历史
    b. 对话当前状态（模糊意图 | 多候选项 | 操作不合法 ）
    c. 对话栈
    d. 引导话术生成

3.2 模块设计

3.2.1 模块调用图

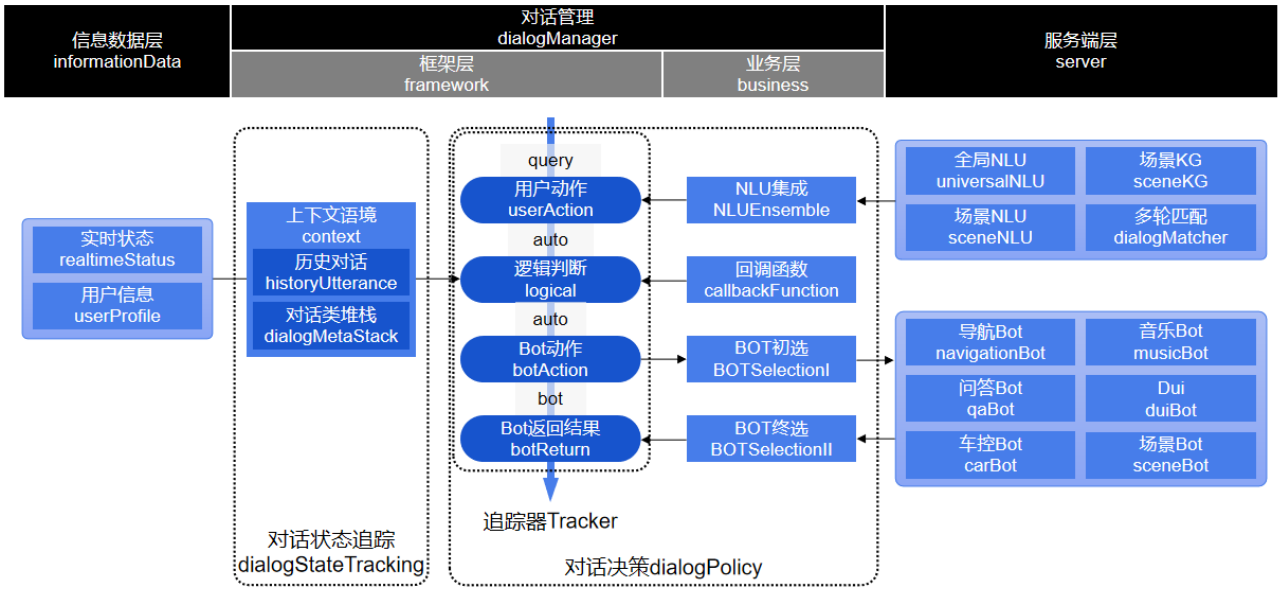


3.2.2 调用时序图



3.2.3 状态机流转示意图

对话管理内部结构



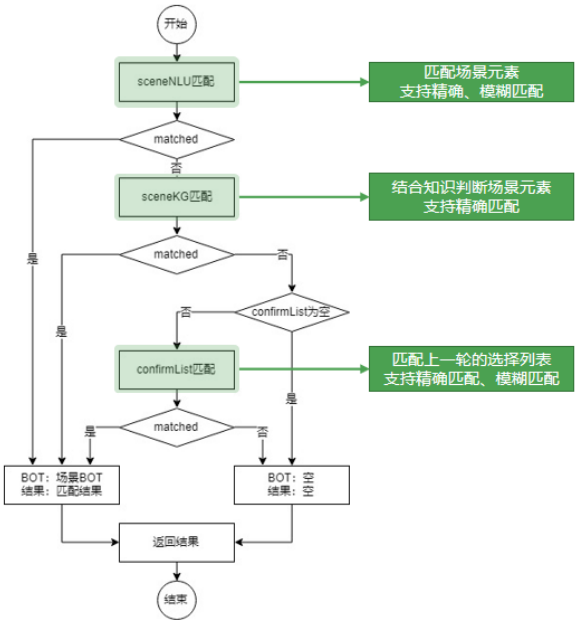
3.2.4 可否执行操作判断

1. 按组件类型判断组件的执行方式（actions），若没有actions，使用协议中组件动作的默认值，个别定制化配置是由场景服务与语义平台对接
2. 结合上文和场景NLU结果，判断信息状态（完整、不完整或冗余）

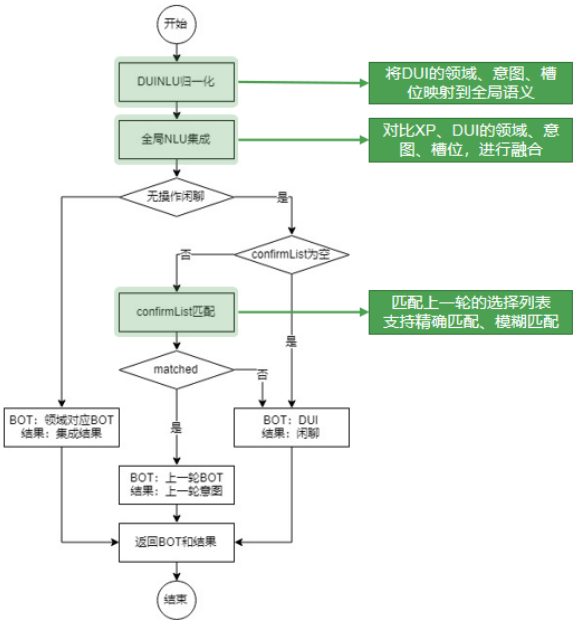
3.2.5 NLU结果融合

NLU融合：全局NLU和场景NLU分别融合，结果为两个：全局NLU和场景NLU

NLU集成逻辑



场景类NLU集成



全局类NLU集成

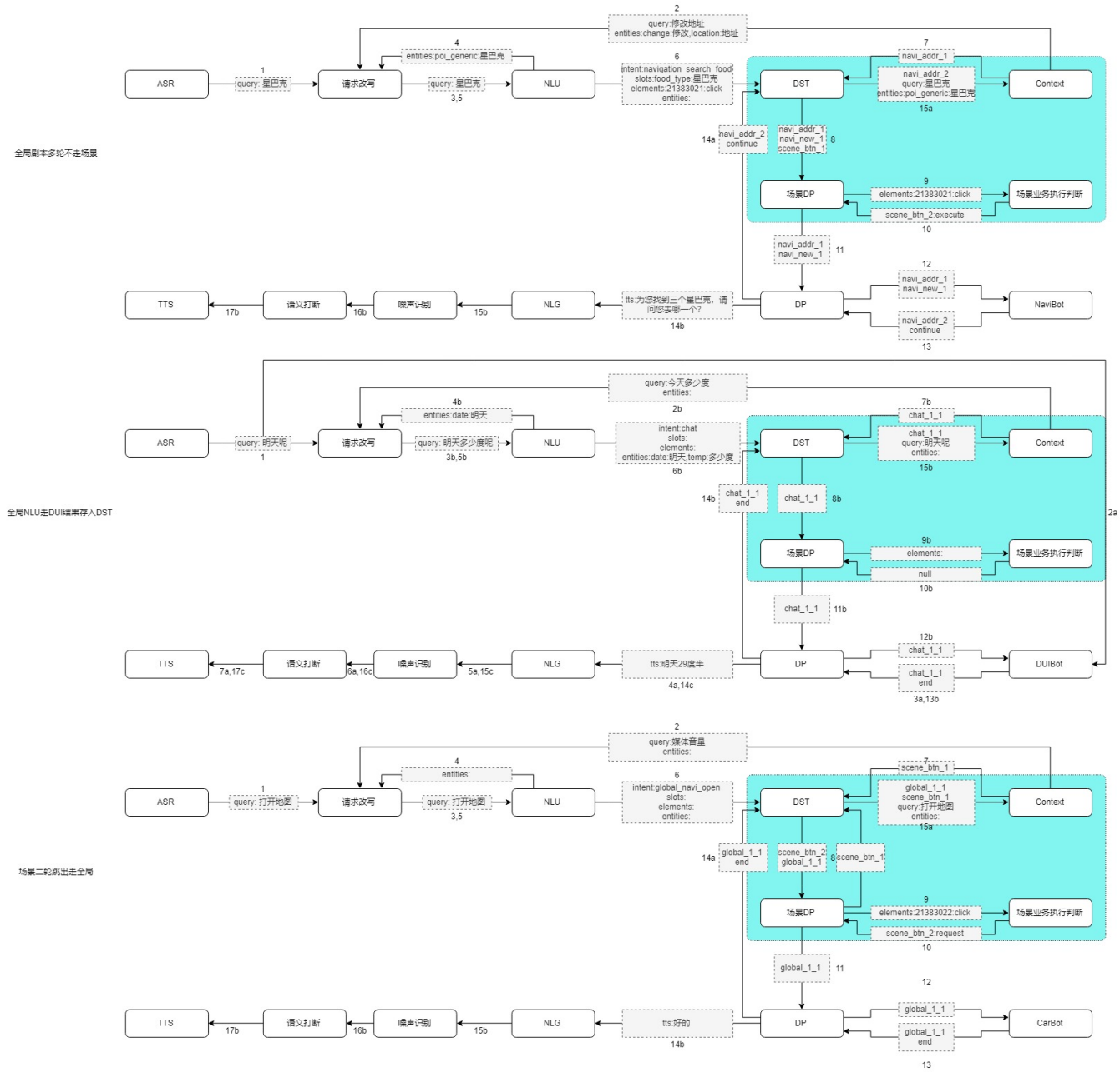
Bot初选：选择发给哪些bot

Bot终选：选择最终的分发结果

3.2.6 中控最终融合结果

selected字段	解释	stacked tracker动作	状态
noise	被判断为噪声	回退	开发中
none	没有bot结果被选中	回退	开发中
uninterrupted	不可被打断	回退	未开发
dui	融合后DUI结果被选中	选中DUI，其他回退	开发中
scene	场景bot结果被选中	选中场景，其他回退且轮数+1	已提测
一个指定的领域	这个指定领域的tts被选中	选中指定领域，其他回退且轮数+1	已提测

3.2.7 例子

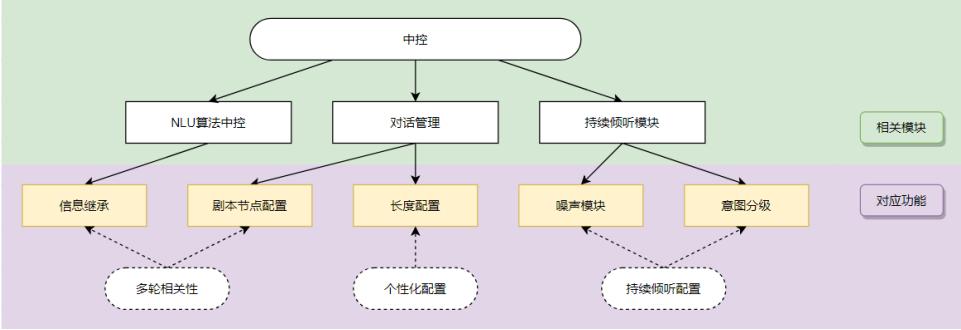


注：谈蓝色部分为xai-dialogo-manager部分

3.3 需求设计

3.3.1 语义打断

1. 需求文档：02. 语义打断
2. 功能点：
    a. 根据多轮信息继承和当前领域/意图，判断是否打断
        i. NLU算法中控结果+限定领域/意图的配置文件
    b. 根据剧本状态/节点信息判断是否打断
        i. contextId和sessionId的配置文件
    c. 根据tts长度规则配置：
        i. tts字符长度达到30字，可用global_exit和global_select及cancel槽位打断
    d. 中控将最终打断的结果回传 @如栋
        i. 清空新建trackers，恢复之前状态
3. 设计：

a. 

4. 示例：

id	utterance	analysis	interruptable	explanation
1.1	user：导航去北大			
1.2	bot：为您找到三个结果，您.....	bot没有说完，用户凭经验补全了tts，为列表选择的澄清动作		
1.3	user：第一个		true	剧本可打断：true；相关性：true；信息继承：false；tts长度打断：false
2.1	user：什么是自适应巡航			
2.2	bot：自适应巡航.....			
2.3	user：你可以休息了	global_exit是可跳出意图	true	剧本可打断：false；相关性：false；信息继承：false；tts长度打断：true
3.1	bot：为您找到三条路线，请问...			
3.2	user：距离家近的	bot没有说完，“距离家近的”不是路线选择剧本可执行的操作	false	剧本可打断：false；相关性：false；信息继承：false；tts长度打断：false
3.3	user：第一条			

3.3.2 TTS优化

1. 需求文档：系统设置12月底demo产品需求V0.3
2. 功能点：
    a. 缺少动词：只命中了控件，没有action或value，例如：蓝牙开关、媒体提示音；
    b. 包含动词，但名词不全（一）：若命中了两个有明确action的控件，需要进行澄清，例如：打开降低音量；
    c. 包含动词，但名词不全（二）：若命中多于两个有明确的action的控件，则需要进行引导，并结束任务，例如：调大音量；
    d. 只命中动词：只命中了控件且控件名称是一个动词，无论命中了几个，都进行引导，并结束任务，例如：连接；
    e. 置信度低，要询问：目前场景NLU没输出置信度，待日后新需求。
3. 设计：对话状态（dialogState）：

状态名称	中文名称	tracker状态	解释	开发状态
fail	失败	已结束	失败	已提测
execute	执行	已结束	执行某些操作	已提测
request	澄清	未结束	对某些信息（槽位、参数）进行澄清，对应功能点【b】	已提测
guide	引导	已结束	引导用户重新说，完整说，对应功能点【a, c, d】	待联调

3.3.3 剧本、场景、全局的优先判断

1. 需求文档：

四、实现方案

4.1 接口设计

【中控】与【对话管理】通信协议（赵耀）

4.2 数据结构设计

系统数据模型、数据库的设计说明。

五、测试方案

5.1 功能测试

功能测试的功能点、测试方法、测试方案建议。

5.2 性能测试

性能测试的功能点、测试方法、测试方案建议。

5.3 稳定性测试

稳定性测试的功能点、测试方法、测试方案建议。

六、开发计划

开发的时间计划、效果预计、里程碑、优先级等。

Todo's

1. 产品沟通时同步一些信息 @易晖
2. 与剧本兼容 @赵耀
3. DM插件 @赵耀
4. 跨域跳出逻辑 @赵耀