Proposal of a Hierarchical Architecture for Multimodal Interactive Systems

Masahiro Araki*[1] Tsuneo Nitta*[2] Kouichi Katsurada*[2]
Takuya Nishimoto*[3] Tetsuo Amakasu*[4]
Shinnichi Kawamoto*[5]

*[1]Kyoto Institute of Technology

*[2]Toyohashi University of Technology

*[3]The University of Tokyo *[4]NTT Cyber Space Labs. *[5]ATR

Outline

- Background
  - Introduction of speech IF committee under ITSCJ
  - Introduction to Galatea toolkit
- Problems of W3C MMI Architecture
  - Modality Component is too large
  - Fragile Modality fusion and fission functionality
  - How to deal with user model?
- Our Proposal
  - Hierarchical MMI architecture
  - "*Convention over Configuration*" in various layers

Background(1)

- What is ITSCJ?
  - Information Technology Standards Commission of Japan
    - under IPSJ (Information Processing Society of Japan)
- Speech Interface Committee under ITSCJ
  - Mission
    - Publish TS (Trial Standard) document concerning multimodal dialogue systems

Background(2)

- Theme of the committee
  - Architecture of MMI system
  - Requirements of each component
- Future directions
  - Guideline for implementing practical MMI system
  - specify markup language

Our Aim

1. Propose an MMI architecture which can be used for advanced MMI research

   ⇔ W3C: From the practical point of view (mobile, accessibility)

2. Examine the validity of the architecture through system implementation

   ⇒ Galatea Toolkit

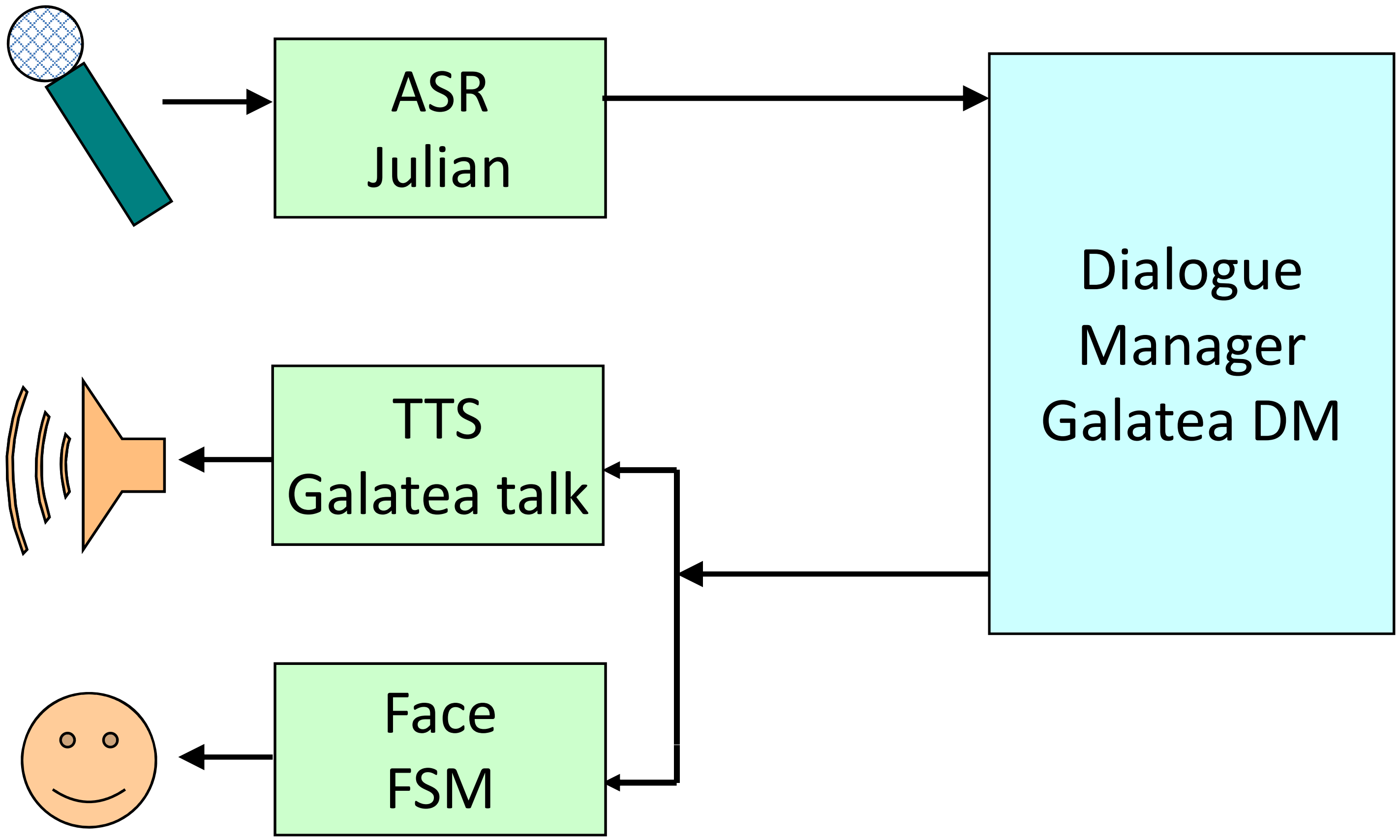
3. Develop a framework and release it as a open source

   ⇒ towards de facto standard

Galatea Toolkit(1)

- Platform for developing MMI systems
  - Speech recognition
  - Speech Synthesis
  - Face Image Synthesis

Galatea Toolkit(2)



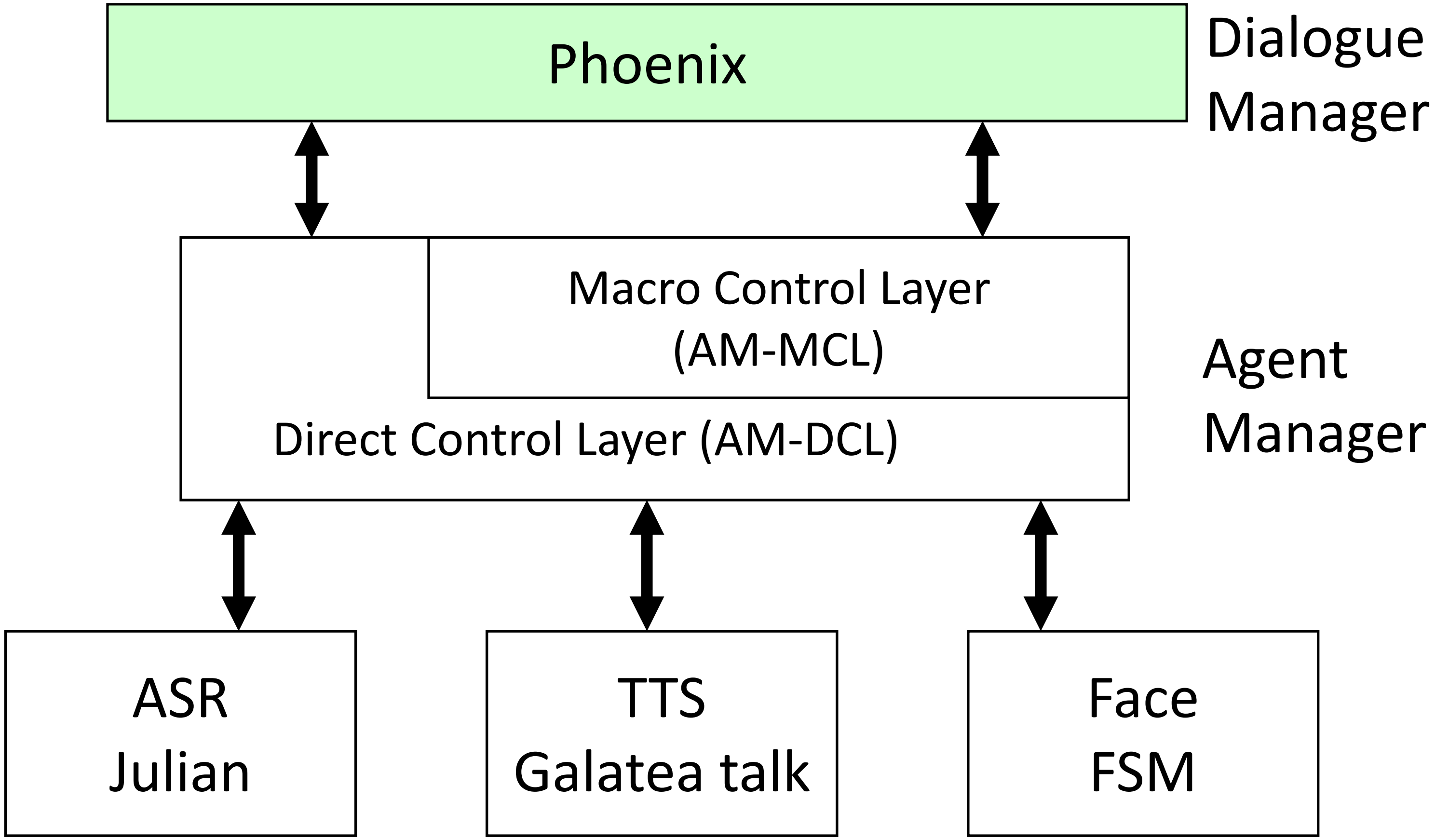
ASR
Julian

TTS
Galatea talk

Face
FSM

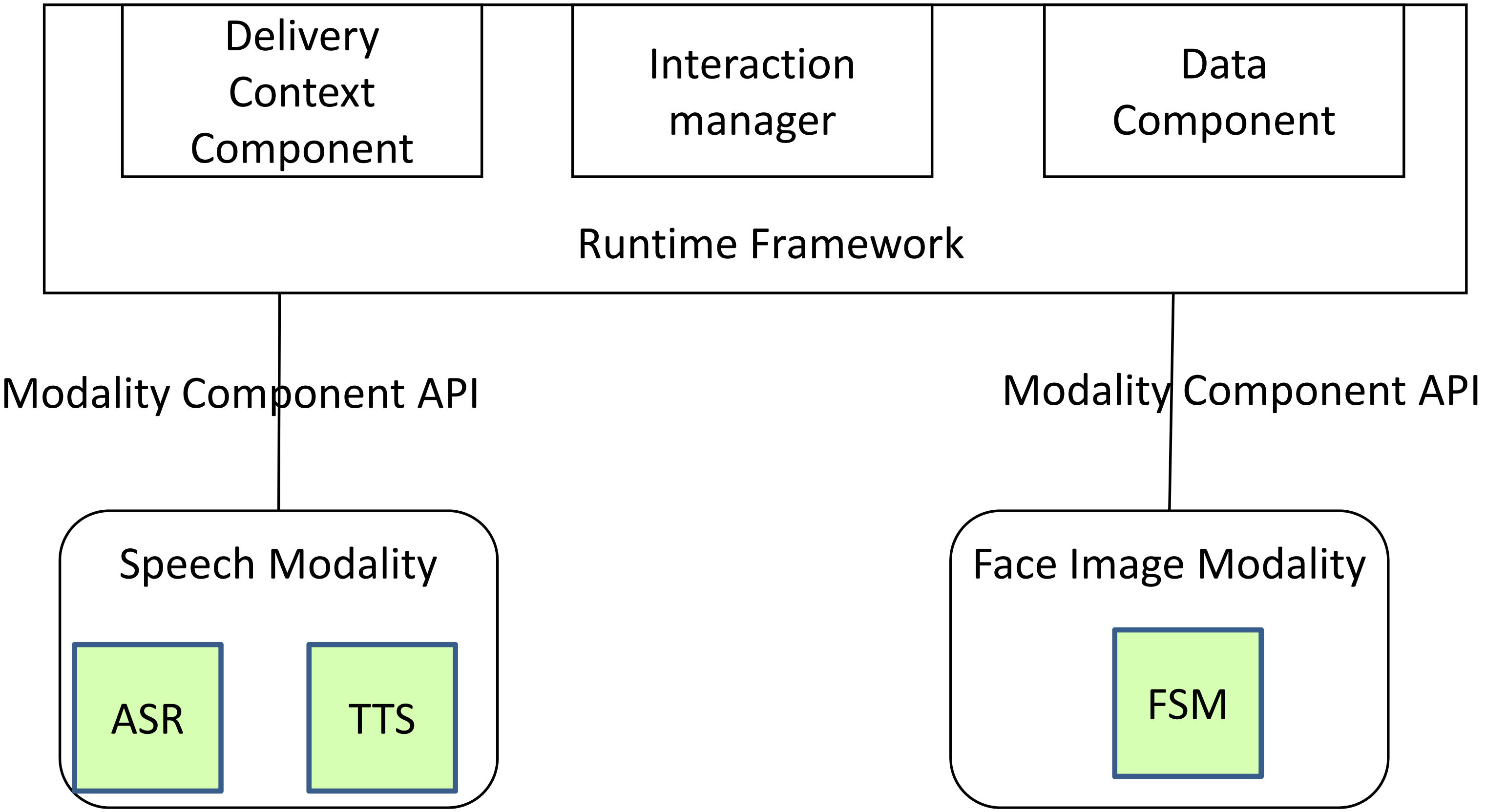
Dialogue
Manager
Galatea DM

Galatea Toolkit(3)



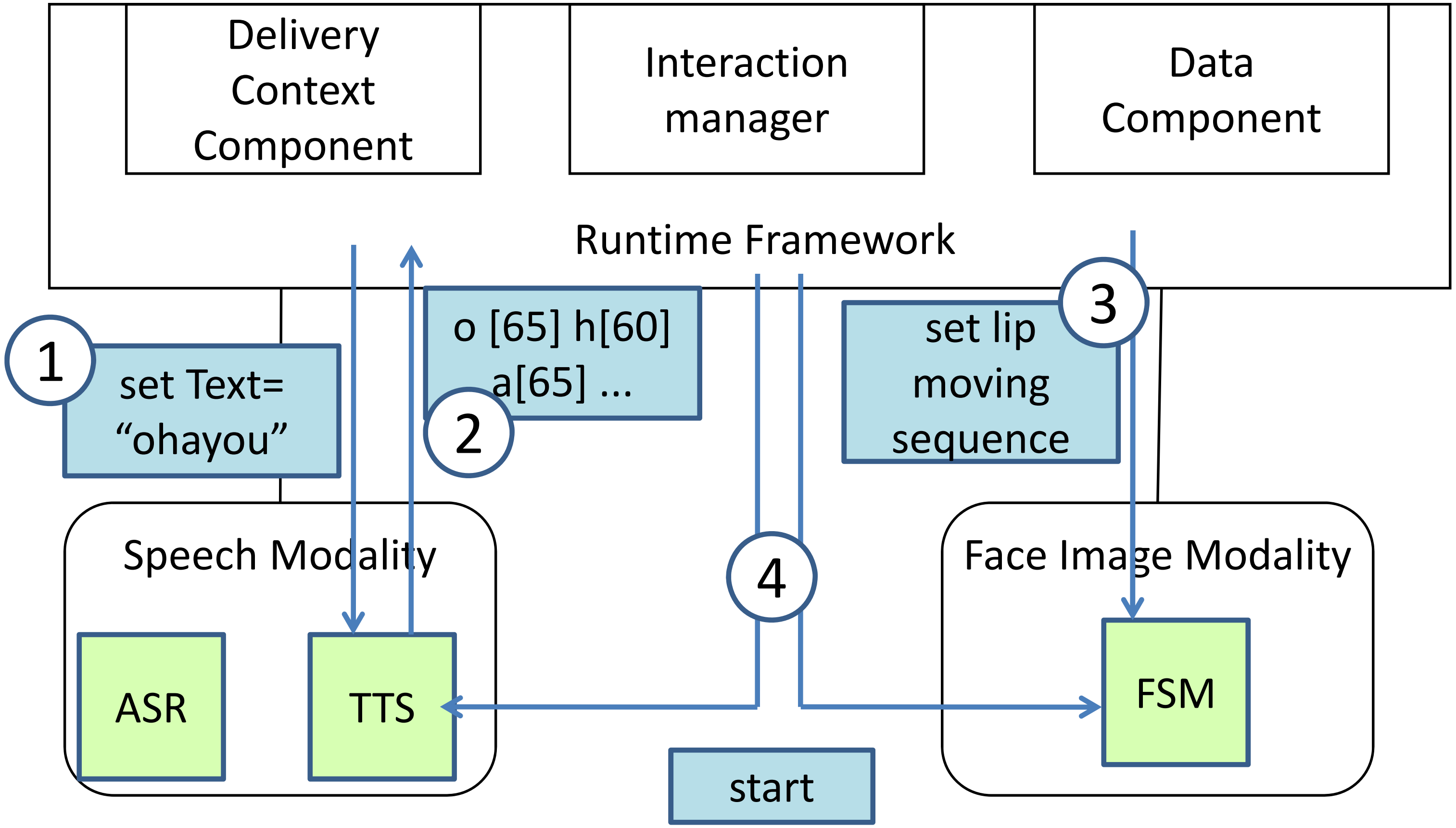
Problems of W3C MMI(1)

- The “size” of Modality Component does not suit for life-like agent control



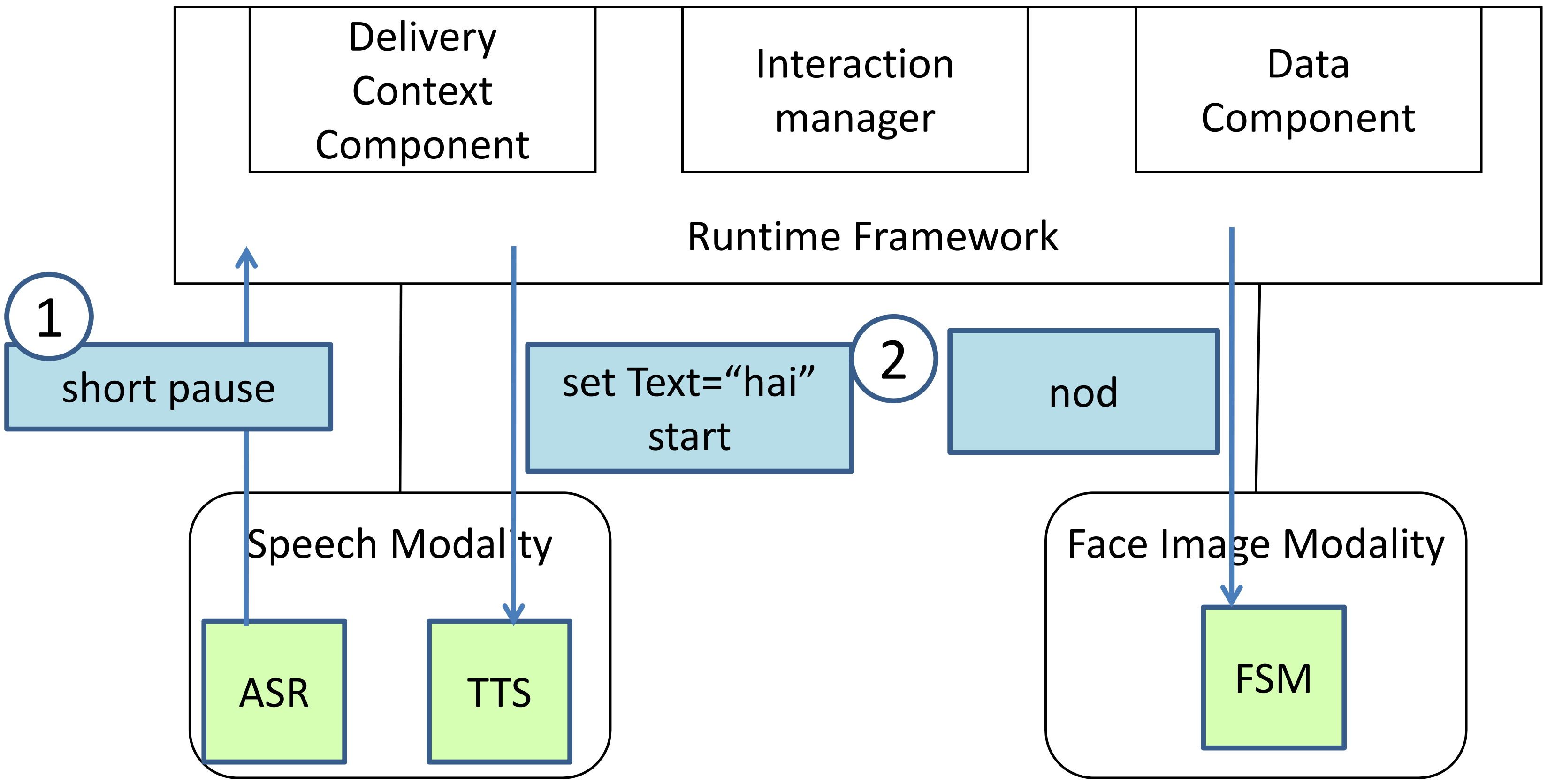
Problems of W3C MMI(1)

- Lip synchronization with speech output



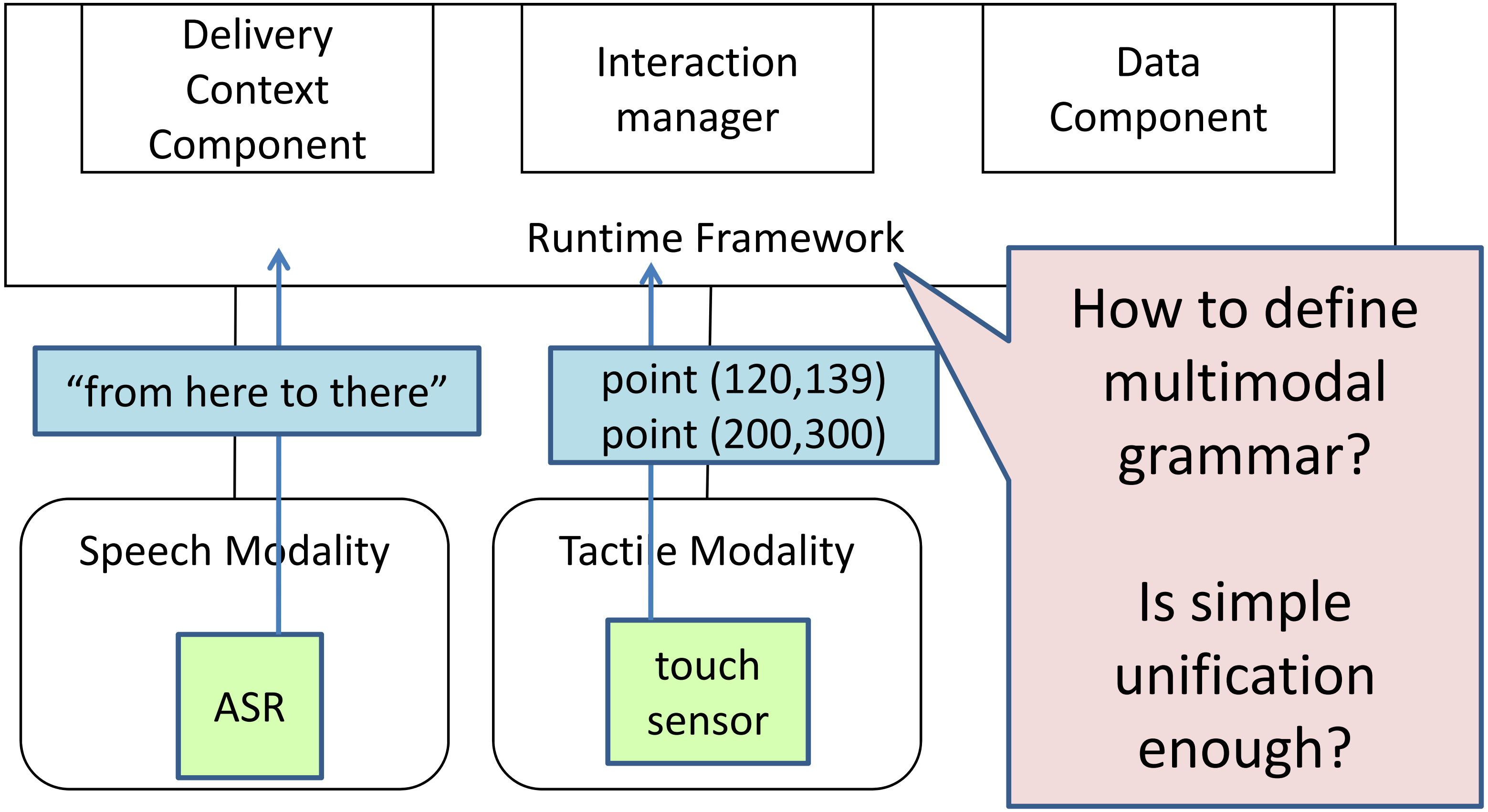
Problems of W3C MMI(1)

- Back channeling mechanism



Problems of W3C MMI(2)

- Fragile Modality fusion and fission functionality

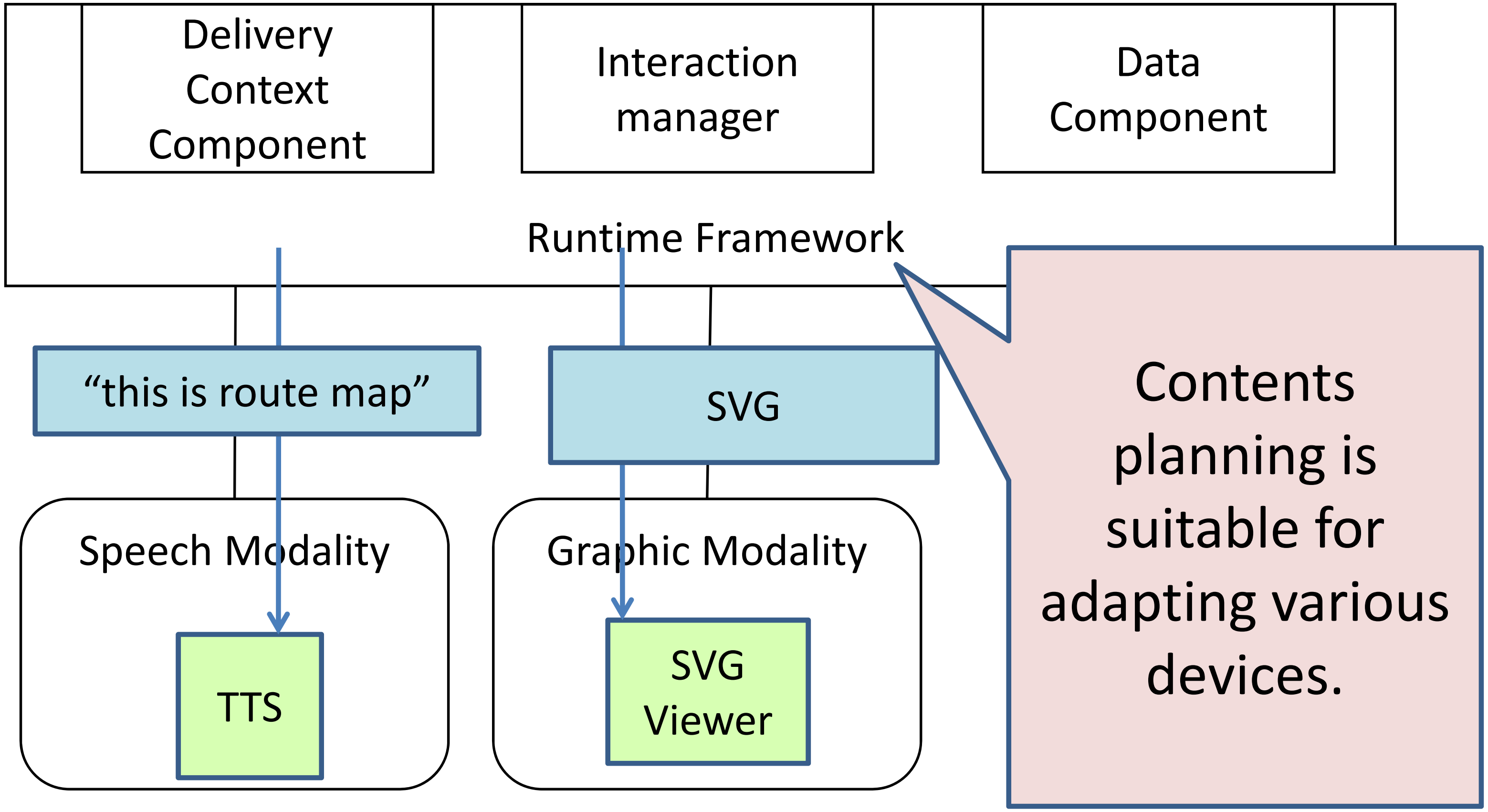


How to define multimodal grammar?

Is simple unification enough?

Problems of W3C MMI(2)

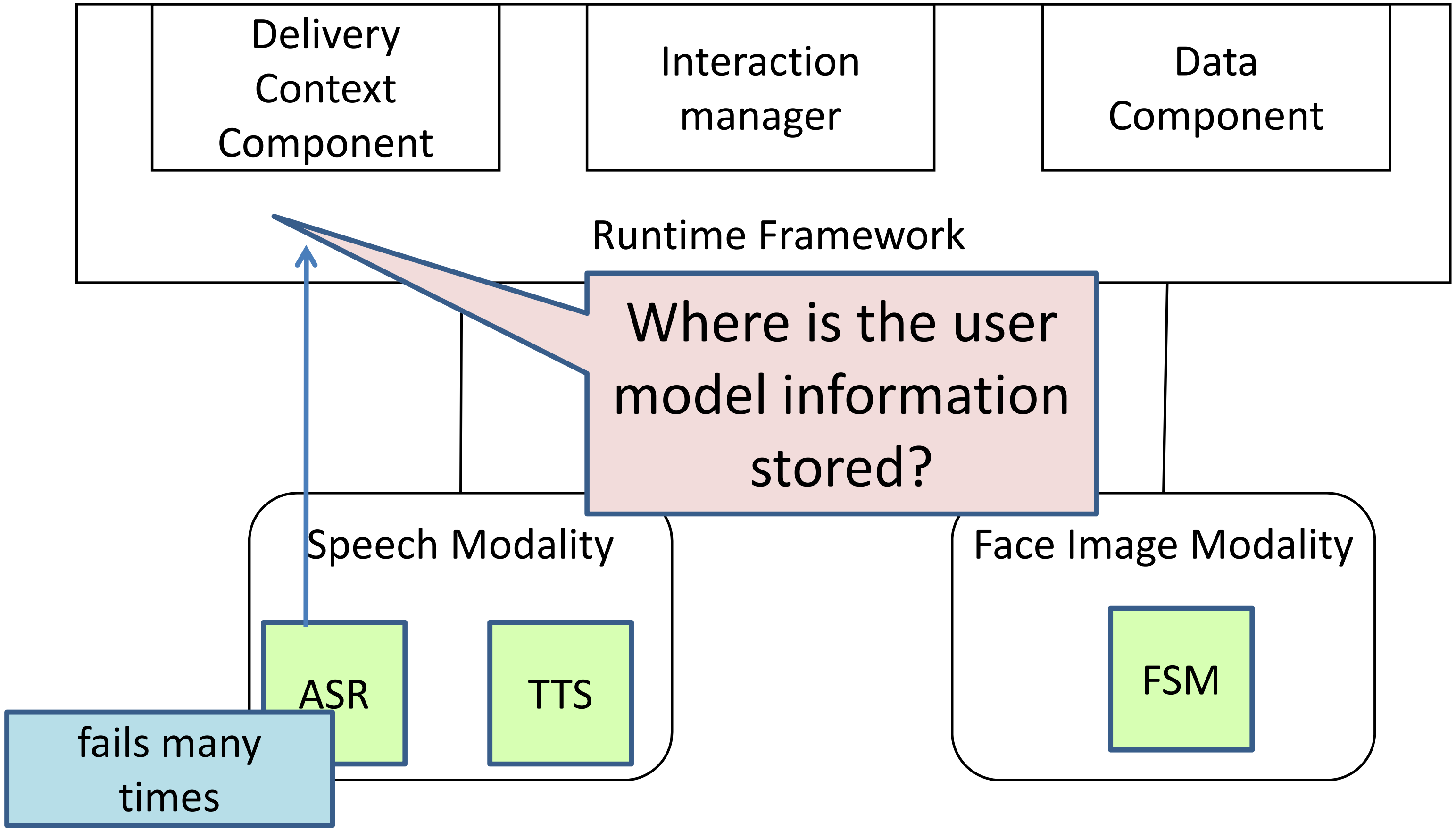
- Fragile Modality fusion and fission functionality



Contents planning is suitable for adapting various devices.

Problems of W3C MMI(3)

- How to deal with user model?



Where is the user model information stored?

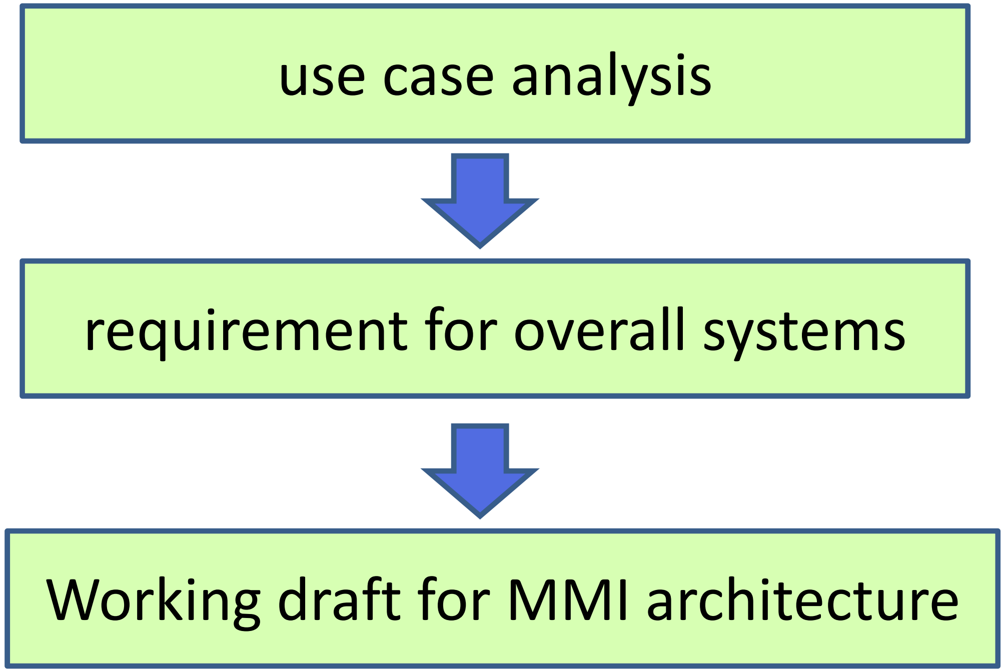
fails many times

Solution

- Back to multimodal framework
  - more smaller modality component
- Separate state transition description
  - task flow
  - interaction flow
  - modality fusion/fission

⇨ hierarchical architecture

Investigation procedure Phase 1



use case analysis

requirement for overall systems

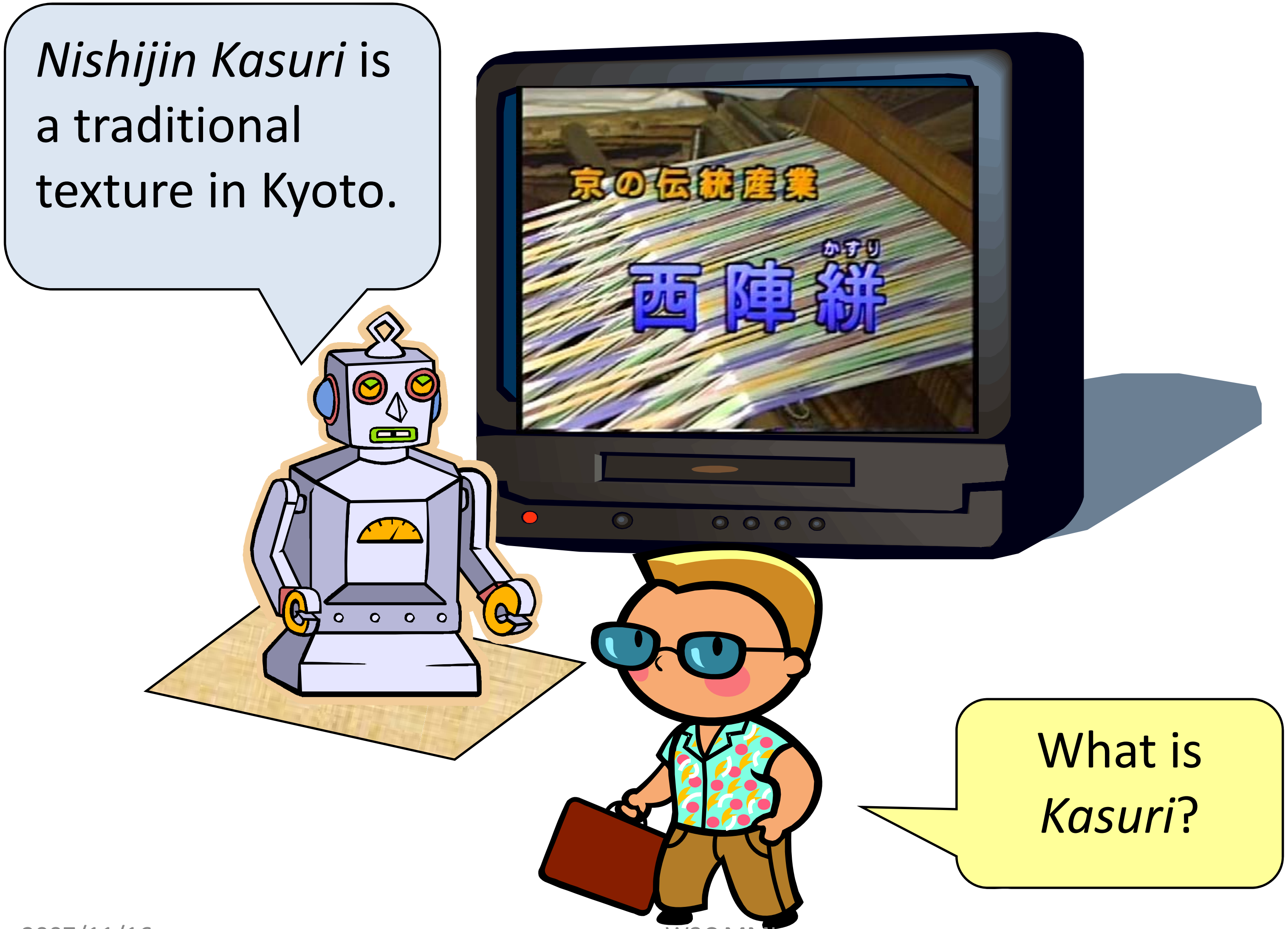
Working draft for MMI architecture

Use case analysis

	Name	input modality	output modality
a	on-line shopping	mouse, speech	display, speech animated agent
b	voice search	mouse, speech	display, speech
c	site search	mouse, speech, key	display, speech
d	interaction with robot	speech, image, sensor	speech, display
e	negotiation with interactive agent	speech	speech, face image
f	kiosk terminal	touch, speech	speech, display

Example of use case

Interaction with robot



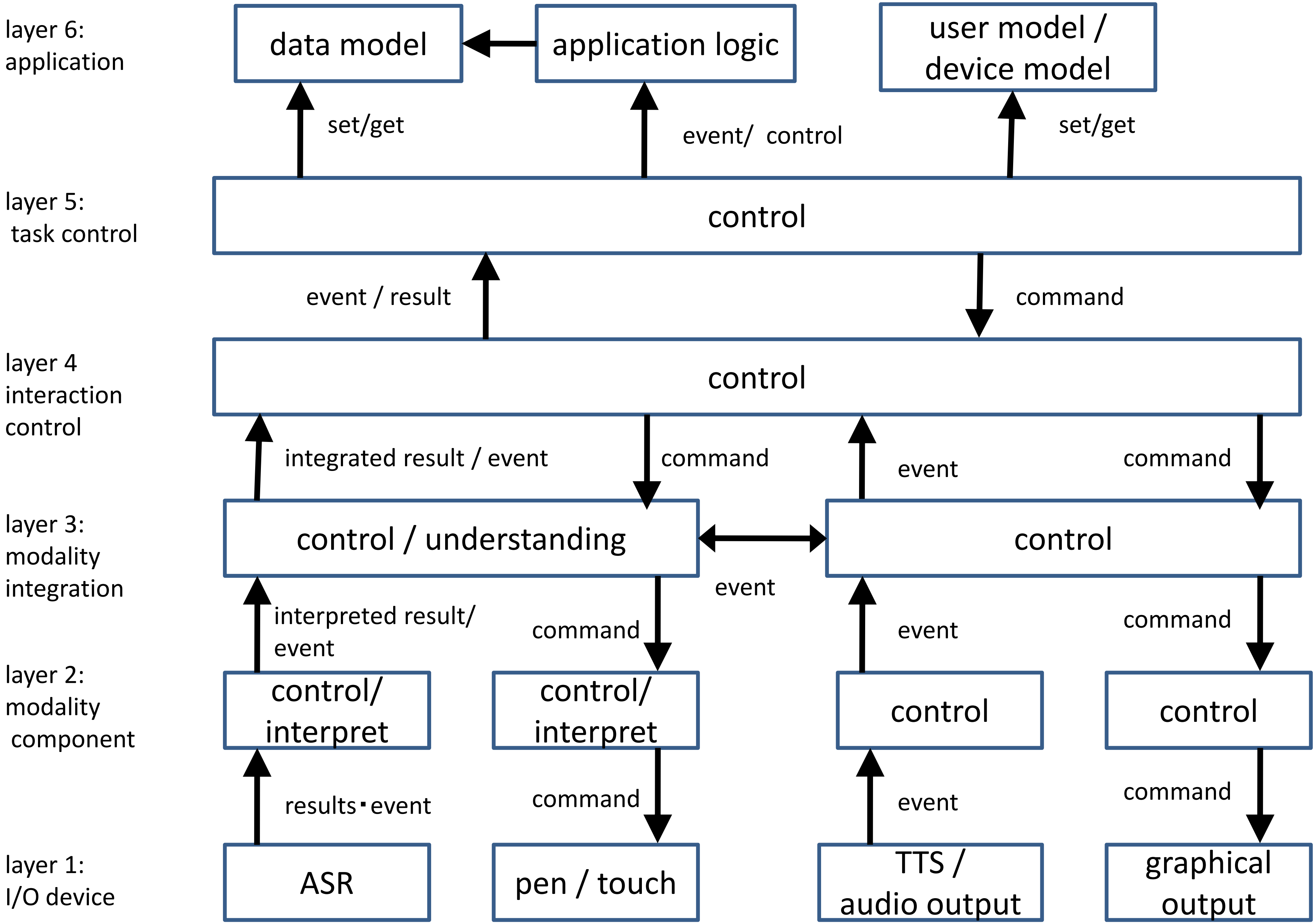
Requirements

1. general
2. input modality
3. output modality
4. architecture, integration and synchronization point
5. runtimes and deployments

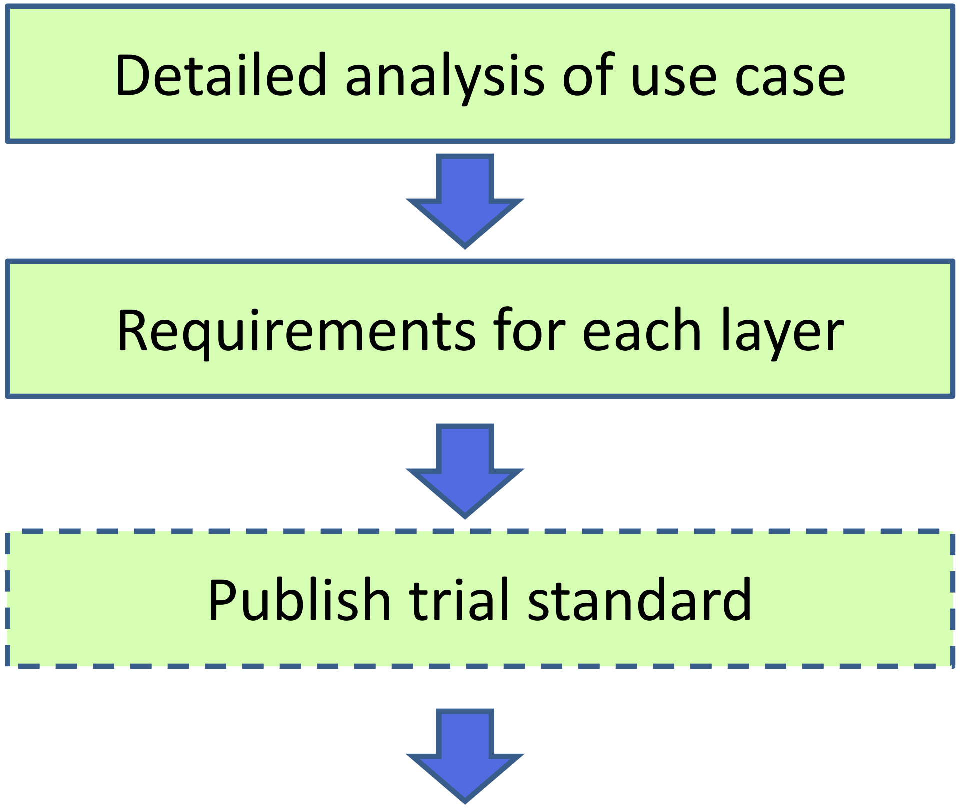
in common with W3C

6. dialogue management
7. handling of forms and fields
8. connection with outside application
9. user model and environment information
10. from the viewpoint of developer

extension


layer 6:
application
data model
application logic
user model /
device model
set/get
event/ control
set/get
layer 5:
task control
control
event / result
command
layer 4
interaction
control
control
integrated result / event
command
event
command
layer 3:
modality
integration
control / understanding
event
control
interpreted result/
event
command
event
command
layer 2:
modality
component
control/
interpret
control/
interpret
control
control
results▪event
command
event
command
layer 1:
I/O device
ASR
pen / touch
TTS /
audio output
graphical
output

Investigation procedure Phase 2



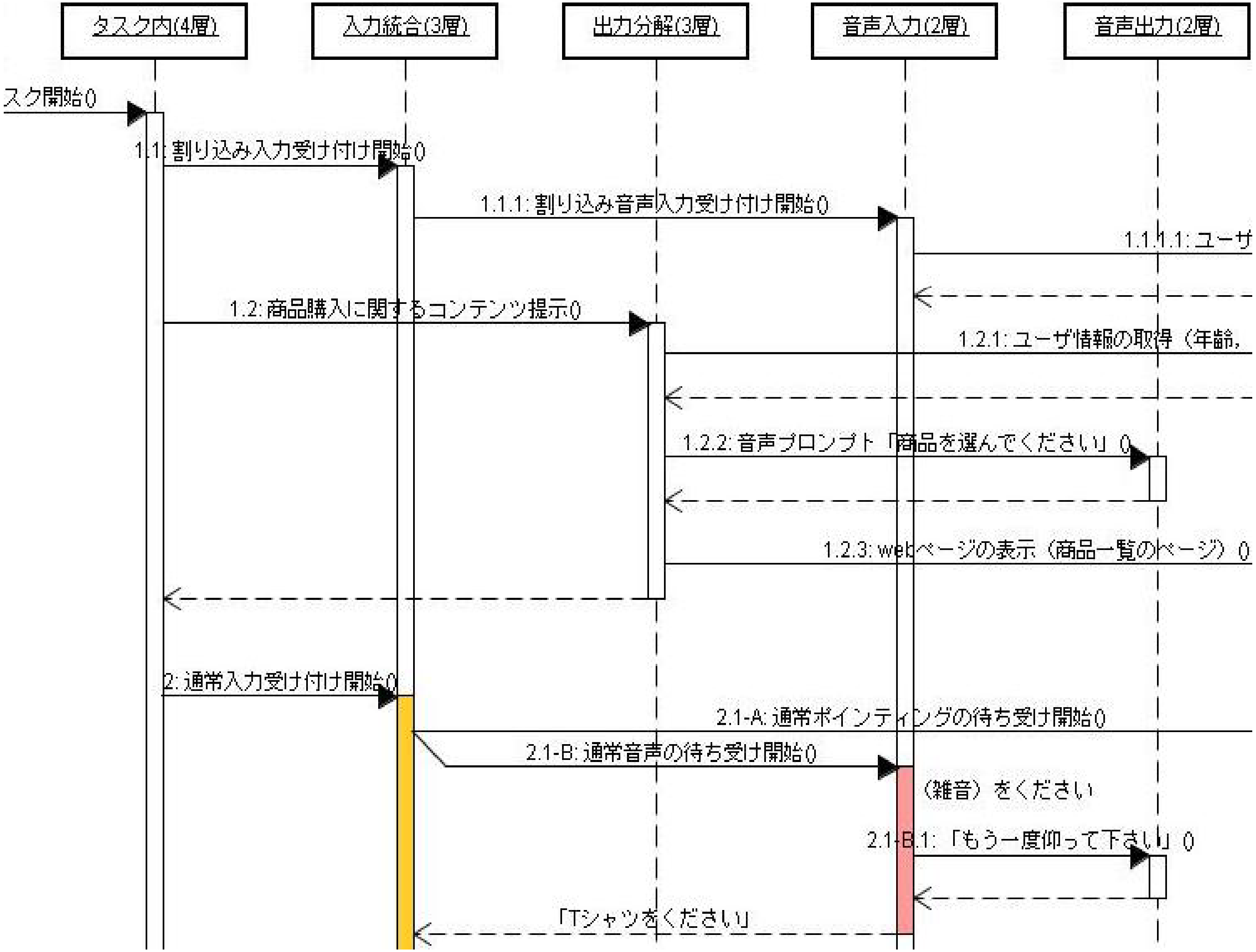
Detailed analysis of use case

Requirements for each layer

Publish trial standard

release reference implementation

Detailed use case analysis


タスク内(4層)
入力統合(3層)
出力分解(3層)
音声入力(2層)
音声出力(2層)
スク開始()
1.1: 割り込み入力受け付け開始()
1.1.1: 割り込み音声入力受け付け開始()
1.1.1.1: ユーザ
1.2: 商品購入に関するコンテンツ提示()
1.2.1: ユーザ情報の取得（年齢,
1.2.2: 音声プロンプト「商品を選んでください」()
1.2.3: webページの表示（商品一覧のページ）()
2: 通常入力受け付け開始()
2.1-A: 通常ポインティングの待ち受け開始()
2.1-B: 通常音声の待ち受け開始()
（雑音）をください
2.1-B.1: 「もう一度仰って下さい」()
「Tシャツをください」

Requirements of each layer

- Clarify Input/Output with adjacent layers
- Define events
- Clarify inner layer processing
- Investigate markup language

1st layer : Input/Output module

- Function
  - Uni-modal recognition/synthesis module
- Input module
  - Input : (from outside) signal
    (from 2nd layer)   information used for recognition
  - Output : (to 2nd) recognition result
  - Example : ASR, touch input, face detection, ...
- Output module
  - Input : (from 2nd) output contents
  - Output : (to outside) signal
  - Example : TTS, Face image synthesizer, Web browser, ...

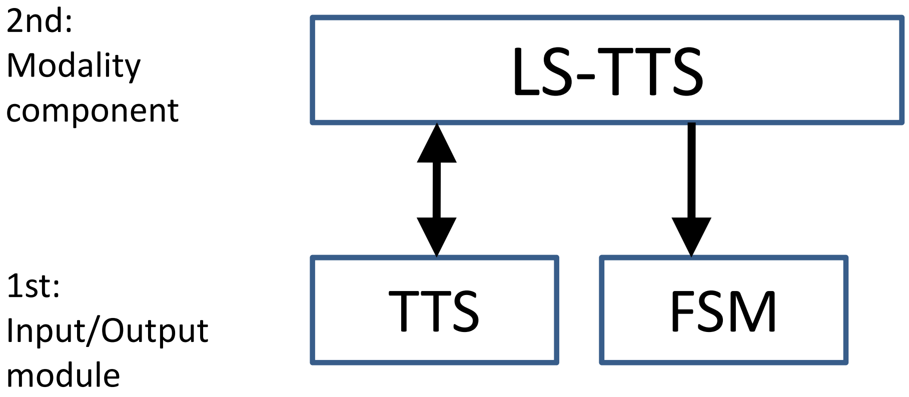
2nd : Modality component

- Function
  - lapper that absorbs the difference of 1st layer

    ex) Speech Recognition component

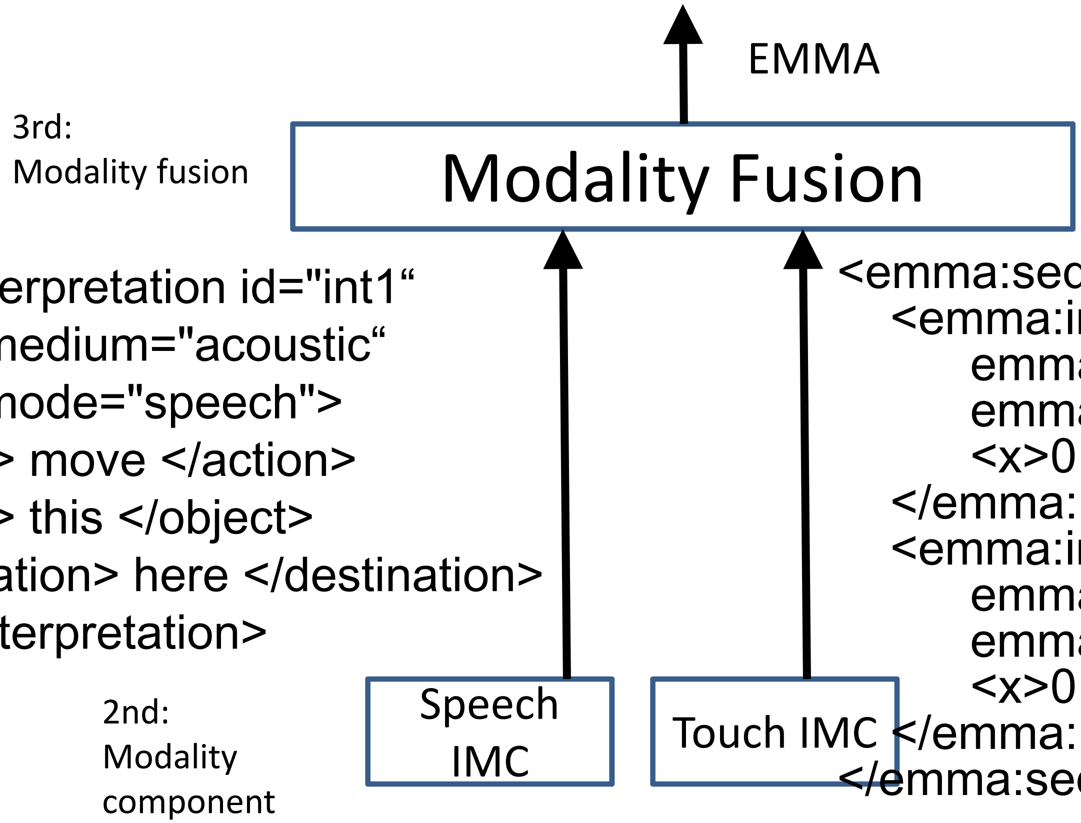
    grammar : SRGS semantic analysis : SISR
    result: EMMA
  - provide multimodal synchronization

    ex) TTS with lip synchronization



3rd : Modality Fusion

- Integration of input information
  - Interpretation of sequential / simultaneous input
  - Output the integrated result as EMMA format

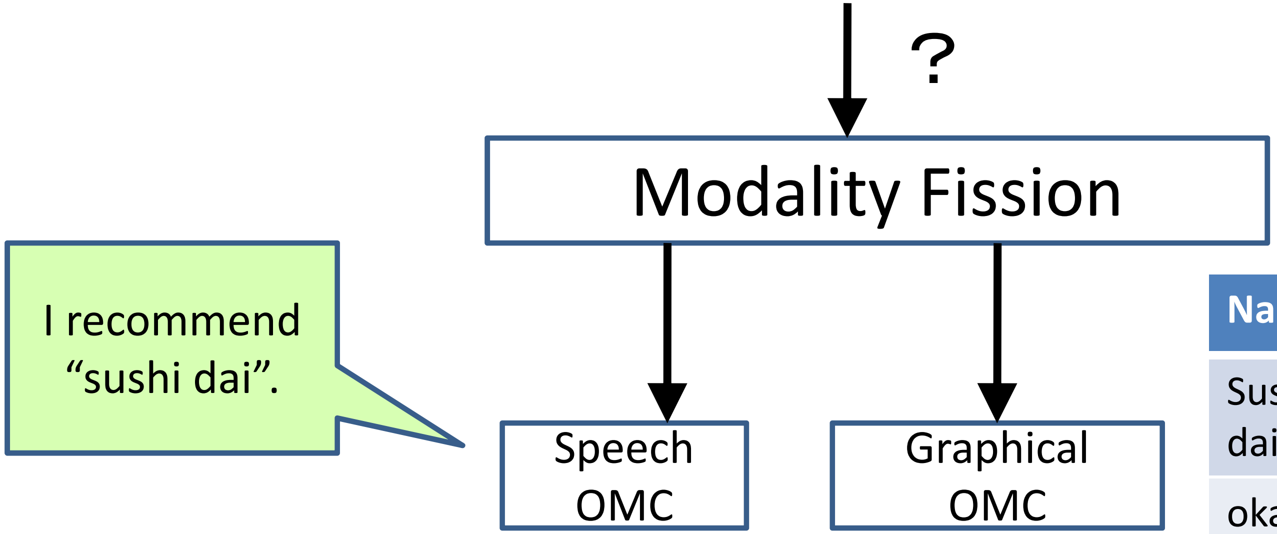


```
<emma:interpretation id="int1“
    emma:medium="acoustic“
    emma:mode="speech">
    <action> move </action>
    <object> this </object>
    <destination> here </destination>
</emma:interpretation>
```

```
<emma:sequence id="seq1">
   <emma:interpretation id="int2“
        emma:medium="tactile“
        emma:mode="ink">
        <x>0.253</x> <y>0.124</y>
   </emma:interpretation>
   <emma:interpretation id="int3“
        emma:medium="tactile“
        emma:mode="ink">
        <x>0.866</x> <y>0.724</y>
   </emma:interpretation>
</emma:sequence>
```

3rd : Modality Fission

- Rendering output information
  - Synchronization of sequential/simultaneous output
  - Coordination of output modality based on the access device



?

Modality Fission

I recommend “sushi dai”.

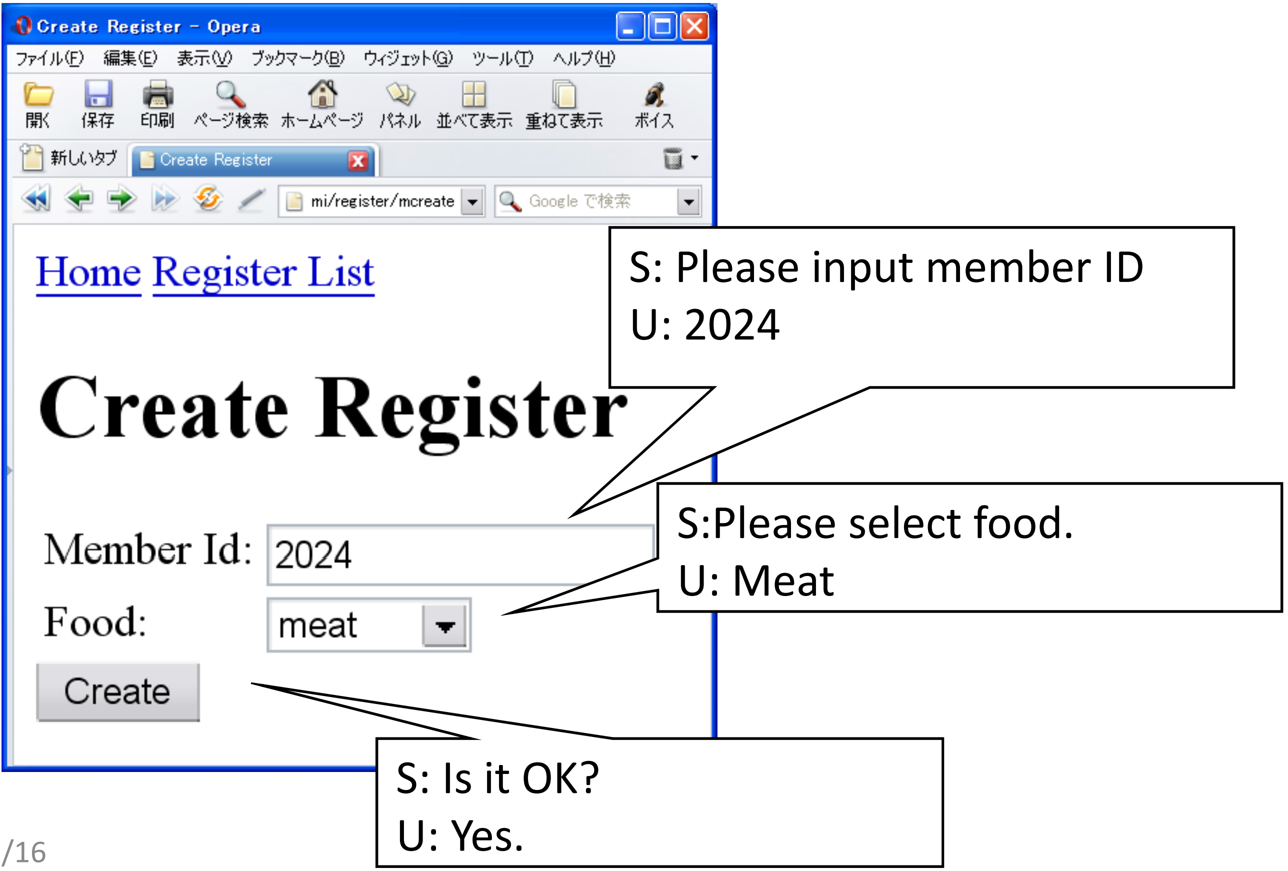
Speech OMC

Graphical OMC

Name	Price	Feature
Sushi dai	3800	good taste
okame	3650	good service
iwasa	3500	shelfish

4th : Inner task control

- Image
  - a piece of dialogue at client side


Create Register - Opera
ファイル(F) 編集(E) 表示(V) ブックマーク(B) ウィジェット(G) ツール(T) ヘルプ(H)
開く 保存 印刷 ページ検索 ホームページ パネル 並べて表示 重ねて表示 ボイス
新しいタブ
Create Register
mi/register/mcreate
Google で検索

Home Register List

Create Register

Member Id: 2024

Food: meat

Create

S: Please input member ID
U: 2024

S:Please select food.
U: Meat

S: Is it OK?
U: Yes.

4th : Inner task control

- Required functions
  - Error handling

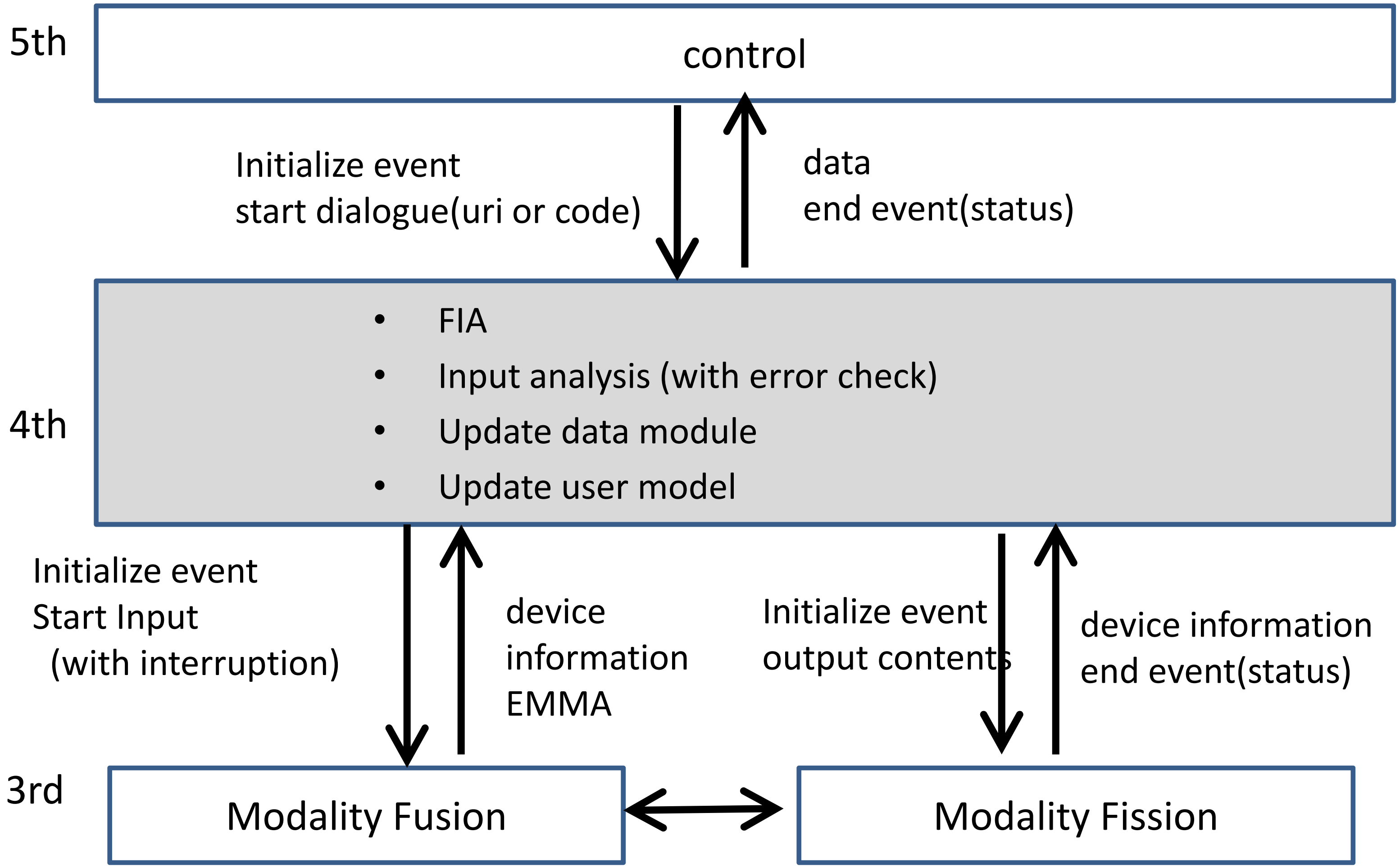
    ex) check departure time < arrival time
  - Default subdialogue

    ex) confirmation, retry, ...
  - Form filling algorithm

    ex) Form Interpretation Algorithm
  - Slot update information

    ex) process of negative response to confirmation request ("NO, from Kyoto.")

4th : Inner task control



5th

control

Initialize event
start dialogue(uri or code)

data
end event(status)

4th

- FIA
- Input analysis (with error check)
- Update data module
- Update user model

Initialize event
Start Input
(with interruption)

device
information
EMMA

Initialize event
output contents

device information
end event(status)

3rd

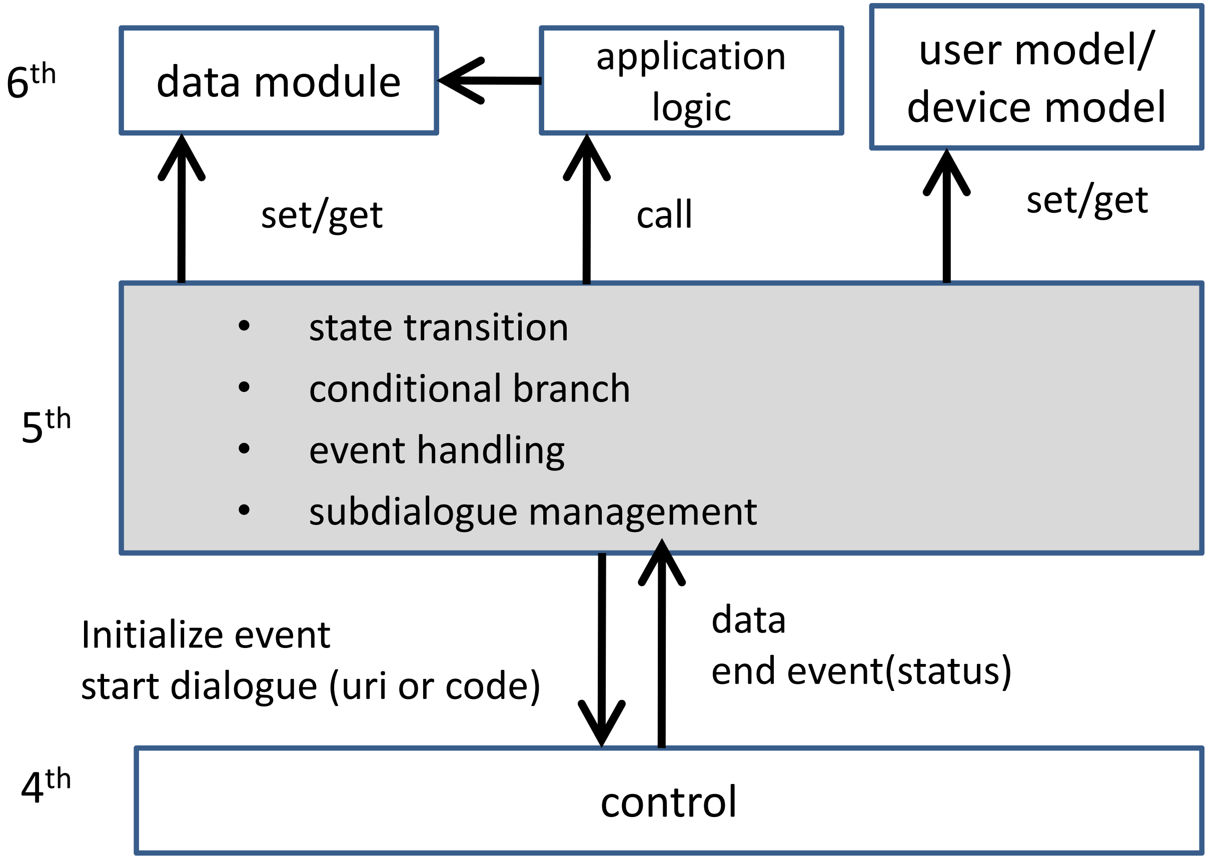
Modality Fusion

Modality Fission

5th : Task control

- Image
  - describe overall task flow
  - server side controller
- Possible markup languae
  - SCXML
  - Controller definition in MVC model
    - entry points and their processing
  - Script language on Rails application framework
    - contains application logic (6th layer)
    - easy to prototype and customize

5th : Task control



- state transition
- conditional branch
- event handling
- subdialogue management

6th : Application

- Image
  - Processing module outside of dialogue system
    - accessed from various layers
- modules
  - application logic

    ex）DB access, Web API access

    - Persist, update, delete, search of data
  - user model / device model
    - persist user’s information through sessions
    - manage device information defined in ontology

Too many markup language?

- Does each level require different markup language?
  - No.
  - simple functionality of 5th and 4th layer can provide data model approach (ex) Ruby on Rails)
  - default function of 3rd layer can be realized simple principle (ex) unification in modality fusion)
  - 2nd layer functions are task/domain independent

"Convention over Configuration"

Summary

- Problems of W3C MMI Architecture
  - Modality Component
  - Modality fusion and fission functionality
  - User model
- Our Proposal
  - Hierarchical MMI architecture
  - "*Convention over Configuration*" in various layers