

Object-oriented Framework for Multimodal Interaction System Development

Masahiro Araki

Department of Information Science, Kyoto Institute of Technology
Matsugasaki Sakyo-ku Kyoto 6068585 Japan

araki@kit.jp

Abstract—We have developed a framework for prototyping a multimodal dialogue system based on the object-oriented paradigm. This demonstration shows the rapid prototyping process from a simple data model definition to working spoken/multimodal dialogue system.

I. INTRODUCTION

We have already proposed a framework for prototyping multimodal interaction (MMI) systems [1]. The innovative idea of our proposal is to take in an object-oriented paradigm to MMI system development. It enables very rapid prototyping of typical MMI systems and easy modification or extension to the resulted prototype.

The proposed framework is based on MVC (Model-View-Controller) model. We use Grails¹ (for GUI Web application framework) with additional object-oriented features enabling spoken interaction to cover the model and controller parts. As a view part, we designed an extended HTML language for multimodal interaction description and developed a multimodal browser which interprets the extended HTML. This browser uses information of class library referred in a data model definition for modality dependent processing, such as ASR control by grammar and semantic interpretation of user's input.

II. DEMO DESCRIPTION

The development process begins by defining the data model with some constraints which reflect task and domain (Fig.1). This data definition language is based on Groovy script which is used in Grails. We add some traits for speech interaction in an object-oriented manner.

```
class Message extends SlotFilling
with SystemInitiative {
    String tel with TelephoneNumber
    String whereabouts
    static constraints = {
        tel(matches: /[0-9]{10,11}/)
        whereabouts(inList:
            ['home', 'refuge', 'hospital'])
    }
}
```

Fig. 1. Example of data model definition (Emergency message board system)

The normal font part of Fig.1 is Groovy code from which Grails generates MVC code for GUI Web application. The italic font part is our extension for speech interaction. We prepare following three classes for realizing speech interaction.

- Task class: Slotfilling, DBSearch, Explanation
- Initiative class: SystemInitiative, MixedInitiative
- Attribute: concept set following <http://schema.org>

From this data definition, our framework automatically modifies Grails' MVC code to multimodal (GUI+speech) interaction code reflecting the traits (Fig.2).

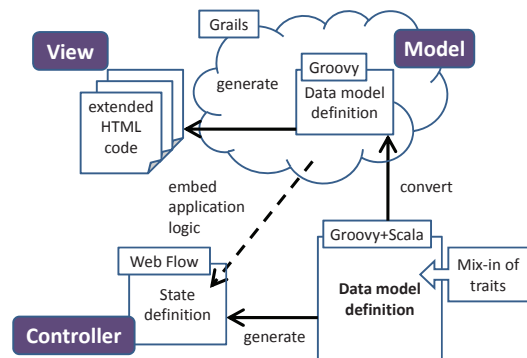


Fig. 2. Prototyping process of multimodal interactive system.

The example of interaction with the generated MMI system is shown in Fig.3.



Fig. 3. Example of realized MMI interaction (slot filling + system initiative).

REFERENCES

- [1] M. Araki and Y. Mizukami, "Development of a Data-driven Framework for Multimodal Interactive Systems", in *Proc. IWSDS*, pp.91-101, 2011.

¹ <http://grails.org>