# Framework for the Development of Spoken Dialogue System based on Collaboratively Constructed Semantic Resources

**Masahiro Araki** **Daisuke Takegoshi**
Department of Information Science
Kyoto Institute of Technology
Matsugasaki Sakyo-ku Kyoto 6068585 Japan
araki@kit.ac.jp

## Abstract

We herein introduce our project of realizing a framework for the development of a spoken dialogue system based on collaboratively constructed semantic resources. We demonstrate that a semantic Web-oriented approach based on collaboratively constructed semantic resources significantly reduces troublesome rule descriptions and complex configurations, which are caused by the previous relational database-based approach, in the development process of spoken dialogue systems. In addition, we show that the proposed framework enables multilingual spoken dialogue system development due to clear separation of model, view and controller components.

## 1 Introduction

In recent years, some large scale repositories of collaboratively constructed semantic resources (CSRs), such as Freebase[1], are available online. Those semantically structured data enable more precise search than simple text matching (e.g. "Find a dental clinic near Kyoto station opens at Saturday night.") and more complex search than simple query to relational database (RDB) (e.g. a query "Find machine learning books written by a researcher of NLP." needs cross search on a book DB and a researcher DB). Since search conditions of such queries to the structured data become complex, natural language, especially speech, for smart phone and tablet PC, is a promising method of query input.

There are some previous researches on converting natural language input to the query of structured data (Lopez et al., 2006) (Tablan et al., 2008). These researches basically concentrated on the input sentence analysis and the query construction. If the developer want to apply existing natural language understanding methods to spoken dialogue system (SDS) for structured data search, there remains fair amount of components that need to be implemented, such as speech input component, dialogue flow management, backend interface, etc.

In order to realize a development environment of SDS for structured data search, we designed a data model driven framework for rapid prototyping of SDS based on CSRs. The proposed framework can be regarded as an extension of existing Rails framework of Web application to (1) enabling speech interaction and (2) utilizing a benefit of CSRs. By using CSRs and the extended Rails framework, the troublesome definitions of rules and templates for SDS prototyping can be reduced significantly compared with the ordinary RDB-based approach.

As this data model driven approach is independent of language for interaction, the proposed framework has a capability of easily implementing multilingual SDS.

[1] http://www.freebase.com/

The remainder of the present paper is organized as follows. Section 2 describes the proposed approach to a data modeling driven development process for SDS based on CSRs and explains the automatic construction of the spoken query understanding component. Section 3 demonstrates the multilingual capability of the proposed framework. In Section 4, the present paper is concluded, and a discussion of future research is presented.

## 2 Data modeling driven approach based on CSRs

### 2.1 Object-oriented SDS development framework

We previously proposed a data modeling driven framework for rapid prototyping of SDS (Araki 2011). We designed a class library that is based on class hierarchy and attribute definitions of an existing semantic Web ontology, i.e., Schema.org[2]. This class library is used as a base class of an application-specific class definition. An example of class definition is shown in Figure 1.

```
@DBSearch
@SystemInitiative
class MyBook extends Book {
  Integer ranking
  static constraints = {
    name(onsearch:"like")
    author(onsearch:"like")
    publisher()
    ranking(number:true)
  }
}
```

Figure 1: Example of class definition.

In this example, the "MyBook" class inherits all of the attributes of the "Book" class of Schema.org in the same manner as object-oriented programming languages. The developer can limit the attributes that are used in the target application by listing them in the constraints section of the class definition. On the other hand, the developer can add additional attributes (*ranking* attributes as the type of *Integer*, which is not defined in original "Book" class) in the definition of the class.

The task type and dialogue initiative type are indicated as annotations at the beginning of the class definition. In this example, the task type is *DB search* and the initiative type is *user initiative*. This information is used in generating the controller code (state transition code, which is equivalent to Figure 2) and view codes of the target SDS.

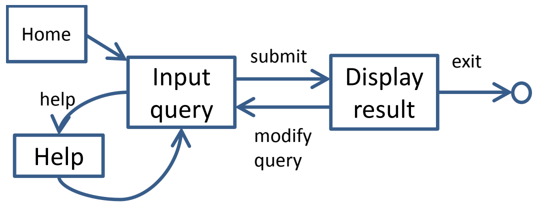


Figure 2: Control flow of the DB search task.

Using Grails[3], which is a Rails Web application framework, the proposed framework generates the dialogue controller code of the indicated task type and the view codes, which have speech interaction capability on the HTML5 code from this class definition. The overall concept of the data modeling driven framework is shown in Figure 3.

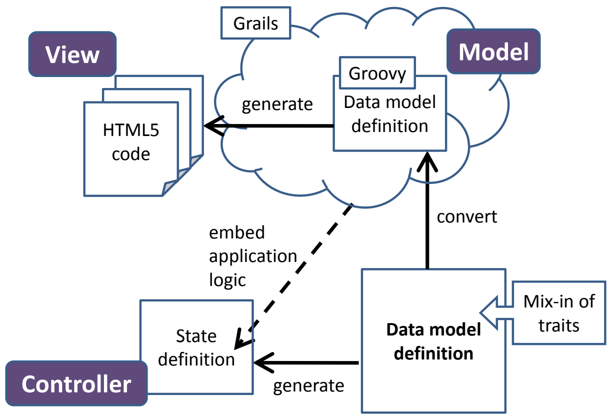


Figure 3: Overview of the data modeling driven SDS development framework.

### 2.2 Using CSRs

The disadvantage of our previous framework, described in the previous subsection, is the high dependence on the dictation performance of the speech recognition component. The automatically generated HTML5 code invokes dictation API, irrespective of the state of the dialogue and initiative type. In order to improve speech recognition accuracy, grammar rules (in system initiative dialogue) and/or the use of a task/domain-dependent language model (in mixed/user initiative dialogue)

[2] http://schama.org/

[3] http://grails.org/

are key factors. In our previous framework, the developer had to prepare these ASR-related components using language resources, which are beyond the proposed data-driven framework.

In order to overcome this defect, we add the Freebase class library, which is based on large-scale CSRs, because Freebase already includes the contents of the data. These contents and a large-scale Web corpus facilitate the construction of grammar rules and a language model that is specific to the target task/domain.

For example, the Film class of Freebase has more than 191 thousand entries (as of May 2012), most of which have information about directors, cast members, genres, etc. These real data can be used as resources to improve ASR accuracy.

In system initiative type dialogue, the contents of each attribute of the target class can construct word entries of the grammar rule for each attribute slot. For example, the grammar rule for the user's response to "Which genre of movie are you searching for?" can be constructed from the contents of the genres attribute of the Film class. We implemented a generator of the set of content words specified in the data model definition from the data of Freebase. The generator is embedded as one component of the proposed framework.

In the mixed/user initiative type tasks, since content words and functional words make up the user's utterance, we need a language model for speech recognition and a semantic frame extractor for the construction of query to semantic data. We designed and implemented a language model generator and a semantic frame extractor using a functional expression dictionary that corresponds to the attributes of Freebase (Araki submitted). The flow of the language model generation is shown in Figure 4.

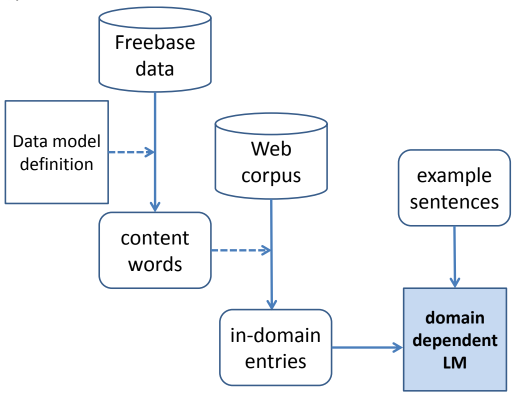
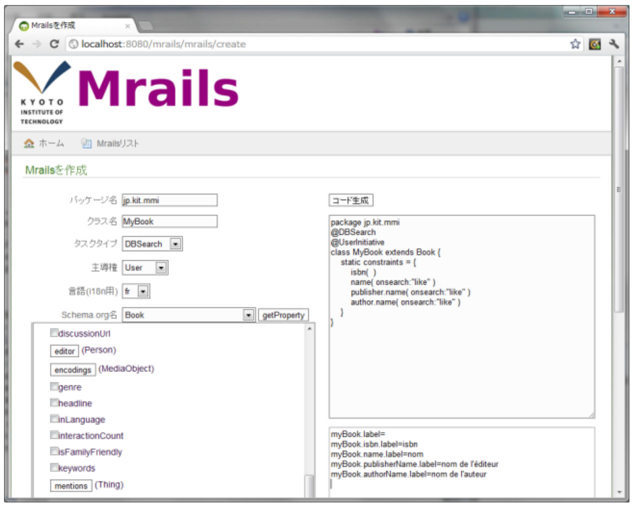


Figure 4: Construction process of LM.

## 2.3 Helper application for data definition

In order to facilitate the data-model definition process, we implemented a helper application called MrailsBuilder. A screenshot of one phase of the definition process is shown in Figure 5, which shows the necessary slots for data definition in the GUI and a list of properties once the developer selects the parent class of the target class.



| | |
|---|---|
| パッケージ名 | jp.kit.mmi |
| クラス名 | MyBook |
| タスクタイプ | DBSearch |
| 主導権 | User |
| 言語(i18n用) | fr |
| Schema.org名 | Book |

discussionUrl
editor (Person)
encodings (MediaObject)
genre
headline
inLanguage
interactionCount
isFamilyFriendly
keywords
mentions (Thing)

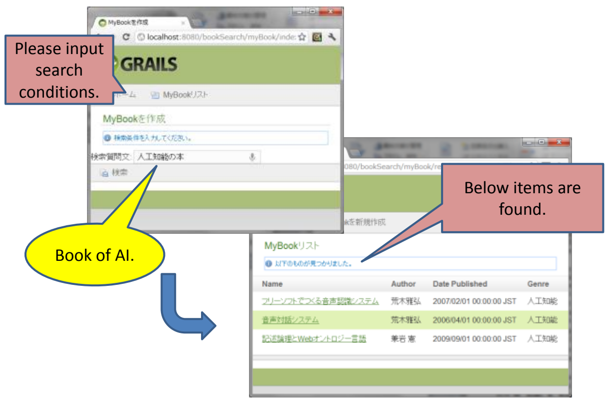
```
package jp.kit.mmi
@DBSearch
@UserInitiative
class MyBook extends Book {
    static constraints = {
        isbn( )
        name( onsearch:"like" )
        publisher.name( onsearch:"like" )
        author.name( onsearch:"like" )
    }
}
```

```
myBook.label=
myBook.isbn.label=isbn
myBook.name.label=nom
myBook.publisherName.label=nom de l'éditeur
myBook.authorName.label=nom de l'auteur
```

Figure 5: Screenshot of MrailsBuilder.

## 3 Multilingual extension of the framework

With the internationalization capability of the Grails base framework and multilingual data resources provided as CSRs, we can generate a multilingual SDS from the data model definition. All of the language-dependent information is stored in separated property files and is called at the time of the dynamic view code generation process in the interaction, as shown in Figure 6.



MyBookを作成

検索条件を入力してください。

検索質問文: 人工知能の本

以下のものが見つかりました。

| Name | Author | Date Published | Genre |
|---|---|---|---|
| フリーソフトでつくる音声認識システム | 荒木雅弘 | 2007/02/01 00:00:00 JST | 人工知能 |
| 音声対話システム | 荒木雅弘 | 2006/04/01 00:00:00 JST | 人工知能 |
| 記述論理とWebオントロジー言語 | 兼岩 憲 | 2009/09/01 00:00:00 JST | 人工知能 |

Figure 6: Example of realized interaction.

We also implemented a *contents extractor* from Freebase data. In Freebase, each class (called "type") belongs to one domain. For example, the "Dish" type belongs to the "Food & Drink" domain (see Figure 7). Although it assigned to a two-level hierarchy, each type has no inherited properties. Therefore, it is easy for Freebase data to represent a set of property values as a string instead of a uniform resource identifier (URI). Each instance has the name property and its value is written in English. For some instances, it also has the name description in another language with the language code. Therefore, we can extract the name of the instance in various languages.

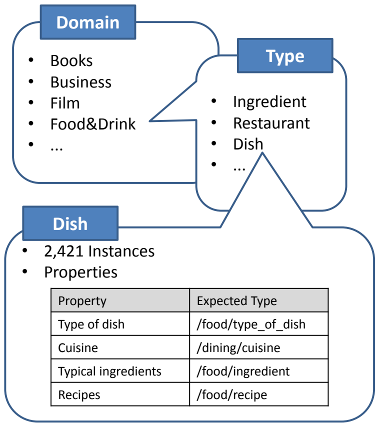


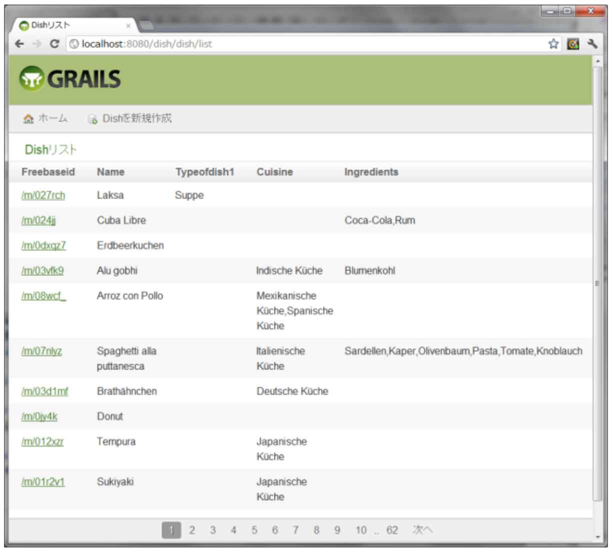
Figure 7: Domain and type of Freebase.

The input of the *contents extractor* is the model definition code as in Figure 1 and the language code (e.g., "ja" for Japanese). As an example, the "MyDish" class is defined as shown in Figure 8.

```
@DBSearch
@UserInitiative
class MyDish extends Dish {
  static constraints = {
    name()
    type_of_dish1(nullable:true)
    cuisine(nullable:true)
    ingredients(nullable:true)
    recipes(nullable:true)
  }
}
```

Figure 8: Model definition of the "MyDish" class.

The *contents extractor* outputs the instance records of the given language code and this instance can be used for LM generator explained in section 2.2. For example, the extracted words in the case of "de" (German) is shown in Figures 9.



| Freebaseid | Name | Typeofdish1 | Cuisine | Ingredients |
|---|---|---|---|---|
| /m/027rch | Laksa | Suppe | | |
| /m/024jj | Cuba Libre | | | Coca-Cola,Rum |
| /m/0dxqz7 | Erdbeerkuchen | | | |
| /m/03vfk9 | Alu gobhi | | Indische Küche | Blumenkohl |
| /m/08wcf_ | Arroz con Pollo | | Mexikanische Küche,Spanische Küche | |
| /m/07nlyz | Spaghetti alla puttanesca | | Italienische Küche | Sardellen,Kaper,Olivenbaum,Pasta,Tomate,Knoblauch |
| /m/03d1mf | Brathähnchen | | Deutsche Küche | |
| /m/0jy4k | Donut | | | |
| /m/012xzr | Tempura | | Japanische Küche | |
| /m/01r2v1 | Sukiyaki | | Japanische Küche | |

Figure 9: German contents of the "MyDish" class.

## 4 Conclusions and future research

We have proposed a framework for development of a SDS on CSRs and have explained rapid construction method of spoken query understanding component and showed its multilingual capability.

In future research, we plan to evaluate the quantitative productivity of the proposed framework.

## Acknowledgments

The present research was supported in part by the Ministry of Education, Science, Sports, and Culture through a Grant-in-Aid for Scientific Research (C), 22500153, 2010.

## References

Masahiro Araki and Yuko Mizukami. 2011. Development of a Data-driven Framework for Multimodal Interactive Systems. In Proc. of IWSDS 2011, 91-101.

Masahiro Araki and Daisuke Takegoshi. accepted. A Rapid Development Framework for Multilingual Spoken Dialogue Systems. In Proc. of IEEE COMPSAC 2012.

Masahiro Araki. submitted. An Automatic Construction Method of Spoken Query Understanding Component from Data Model Definition.

Vanessa Lopez, Enrico Motta, and Victoria S. Uren. 2006, AquaLog: An ontology-driven Question Answering System to interface the Semantic Web. In Proc. of HLT-NAACL 2006, 269-272.

Valentin Tablan, Danica Damljanovic, and Kalina Bontcheva. 2008, A natural language query interface to structured information. In Proc. of the 5h European Semantic Web Conference (ESWC 2008).