Create conversational agents for Android

Carmelo Ferrante
Prof. Giuseppe Riccardi

LPSMT-Spring 2013

Signals & Interactive Systems LAB

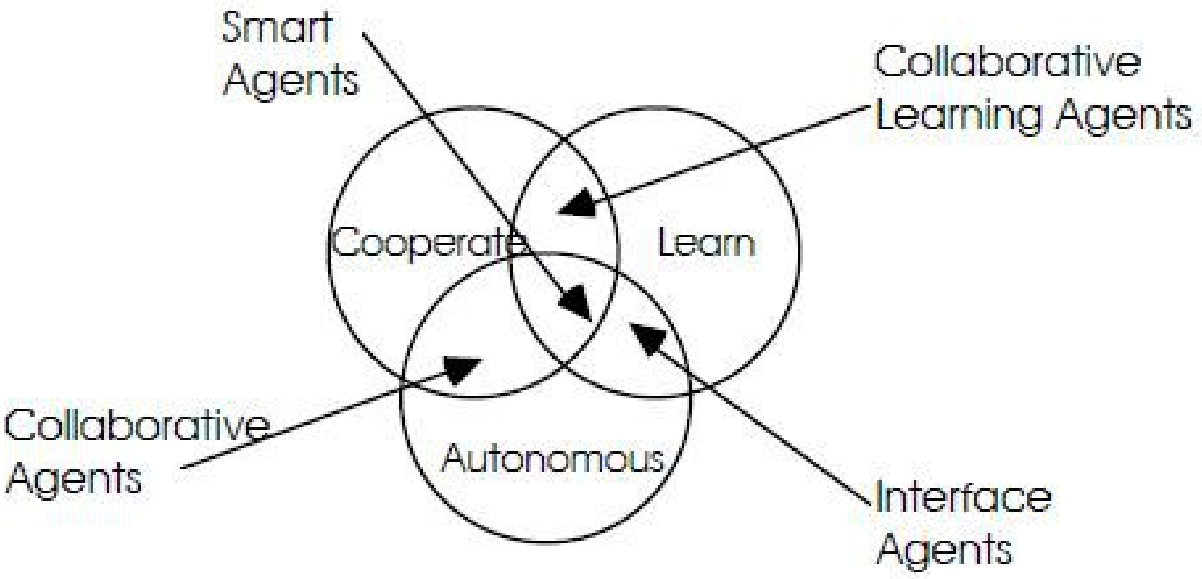
Outline

- Definition of Conversational Agent
- Examples of agents
- How to realize it: a possible architecture
- The AT&T Speech Mashup Service
- What's AT&T Speech Mashup
- AT&T Architecture
- AT&T Speech Mashup Web Portal
- Web Portal functionalities
- Into details: Grammars and SSML Markup
- API and Clients developing
- What is a dialog flow
- “Hello Lab” tutorial for Android

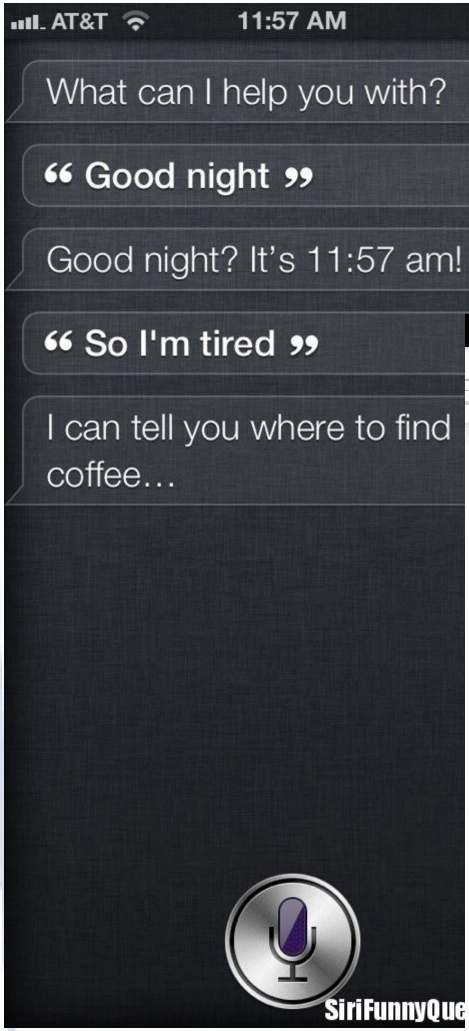
Definition of Conversational Agent

An agent is a system to which the user can delegate the execution of his tasks. It has at least 4 main properties:

1. Autonomy
2. Reactivity
3. Pro-activeness
4. Social ability



Examples of agents



AT&T 11:57 AM 78%

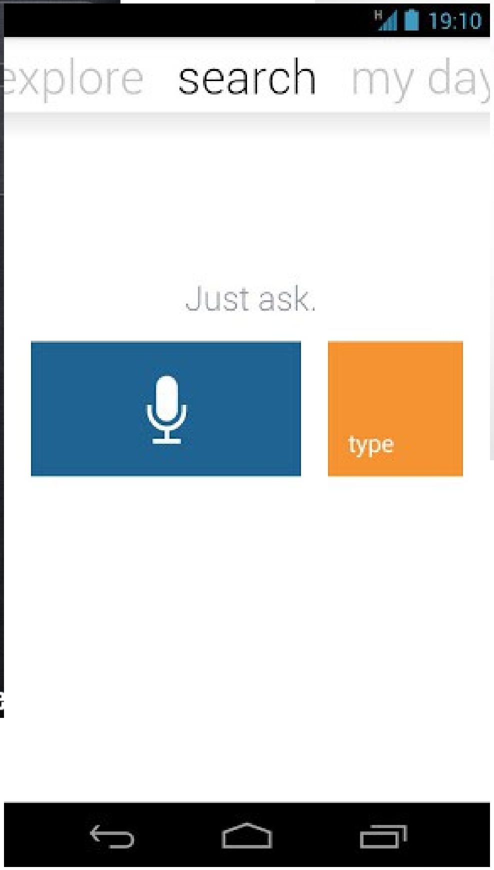
What can I help you with?

“ Good night ”

Good night? It’s 11:57 am!

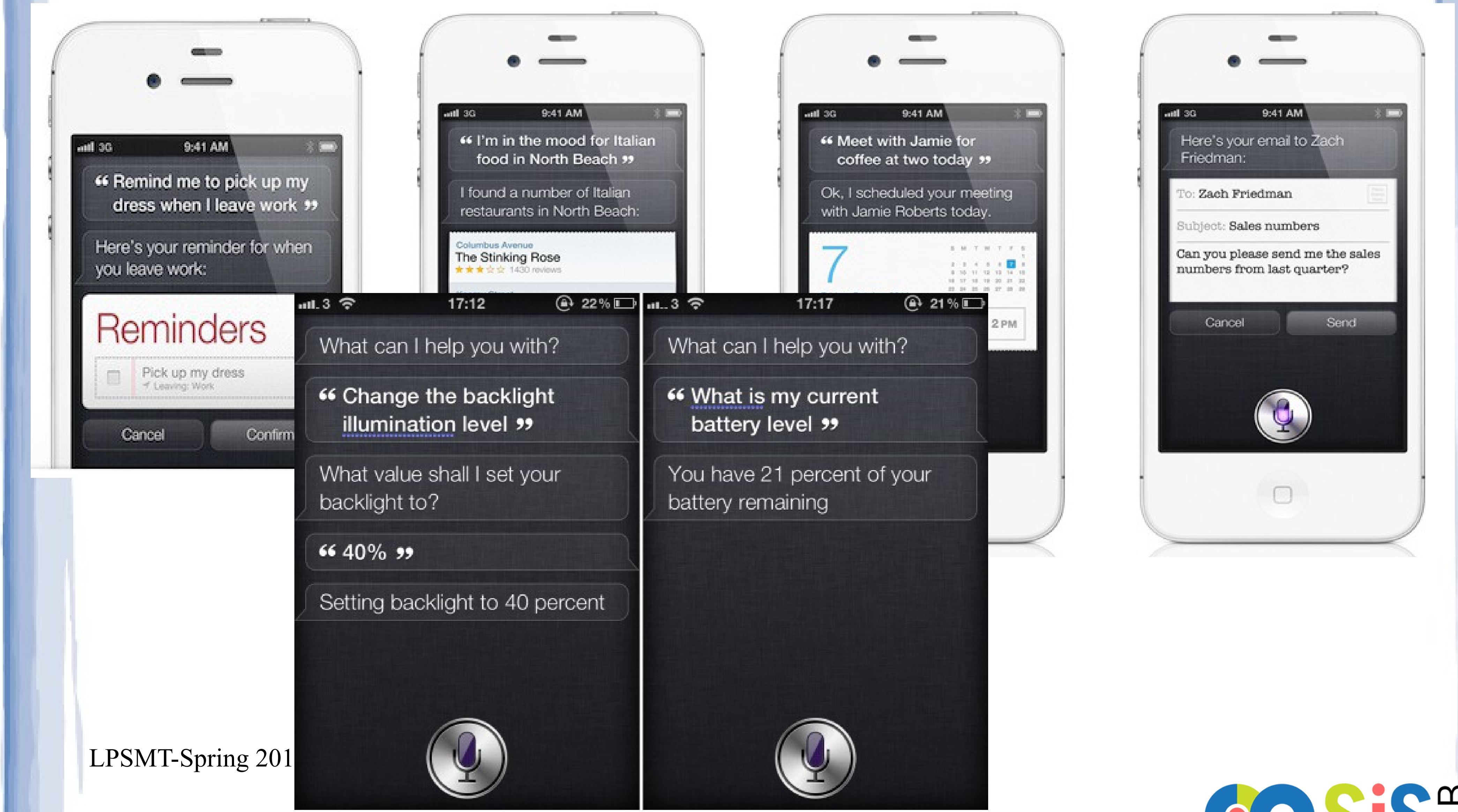
“ So I'm tired ”

I can tell you where to find coffee…



[Video Examples](#)

Examples of agents



“Remind me to pick up my dress when I leave work”

Here's your reminder for when you leave work:

Reminders

□ Pick up my dress
Leaving: Work

“I'm in the mood for Italian food in North Beach”

I found a number of Italian restaurants in North Beach:

Columbus Avenue
The Stinking Rose
★★★☆☆ 1430 reviews

“Meet with Jamie for coffee at two today”

Ok, I scheduled your meeting with Jamie Roberts today.

Here's your email to Zach Friedman:

To: Zach Friedman

Subject: Sales numbers

Can you please send me the sales numbers from last quarter?

What can I help you with?

“Change the backlight illumination level”

What value shall I set your backlight to?

“40%”

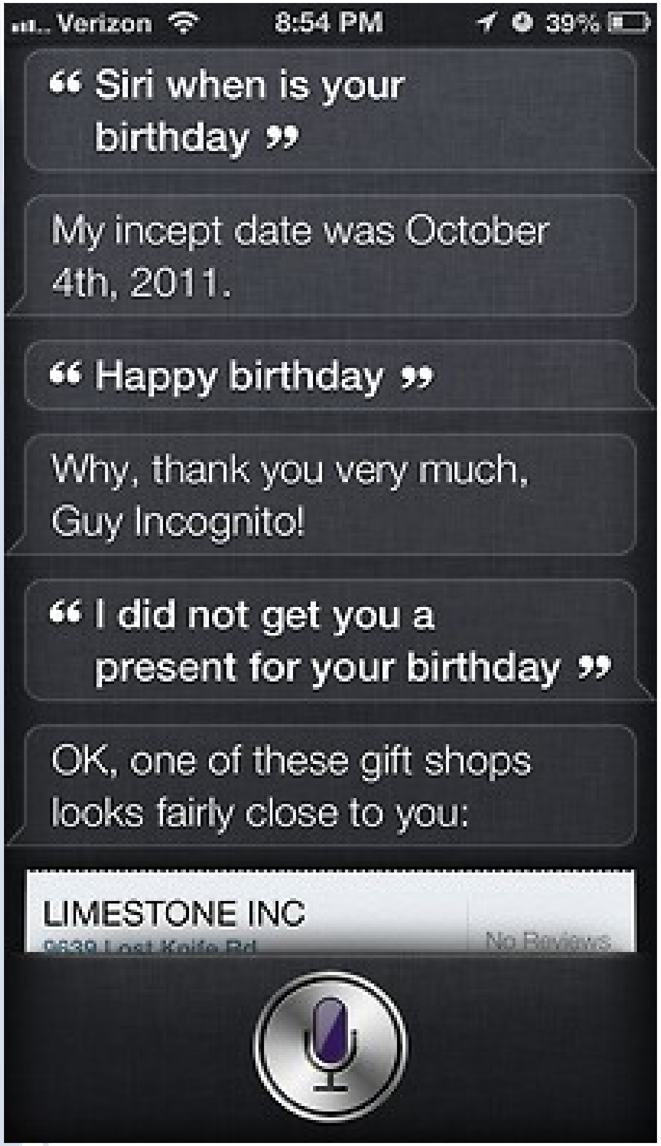
Setting backlight to 40 percent

What can I help you with?

“What is my current battery level”

You have 21 percent of your battery remaining

Funny examples of agents



“ Siri when is your birthday ”

My incept date was October 4th, 2011.

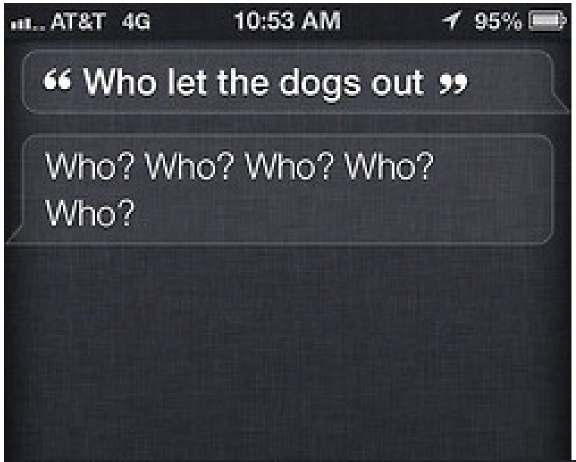
“ Happy birthday ”

Why, thank you very much, Guy Incognito!

“ I did not get you a present for your birthday ”

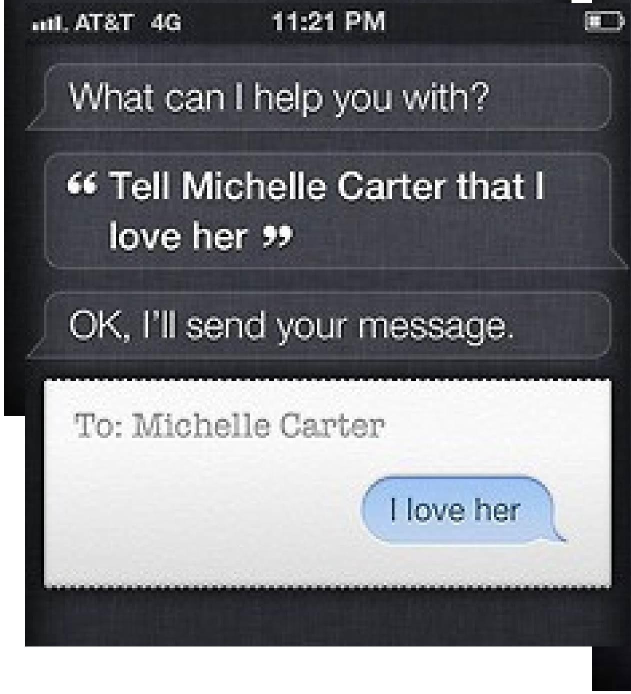
OK, one of these gift shops looks fairly close to you:

LIMESTONE INC
9639 Lost Knife Rd
No Reviews



“ Who let the dogs out ”

Who? Who? Who? Who? Who?



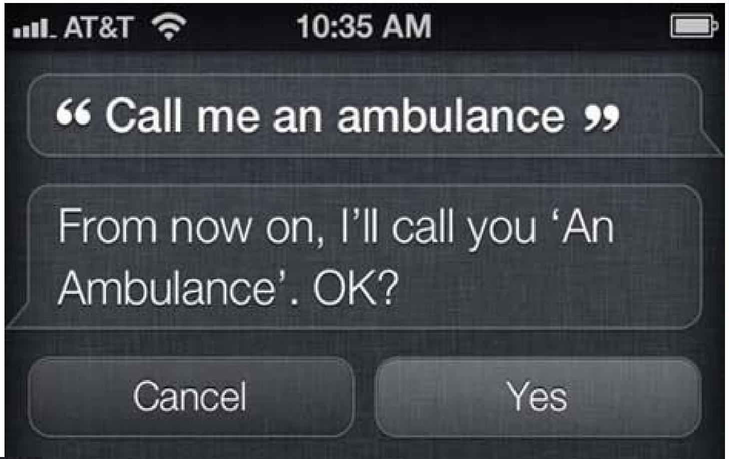
What can I help you with?

“ Tell Michelle Carter that I love her ”

OK, I'll send your message.

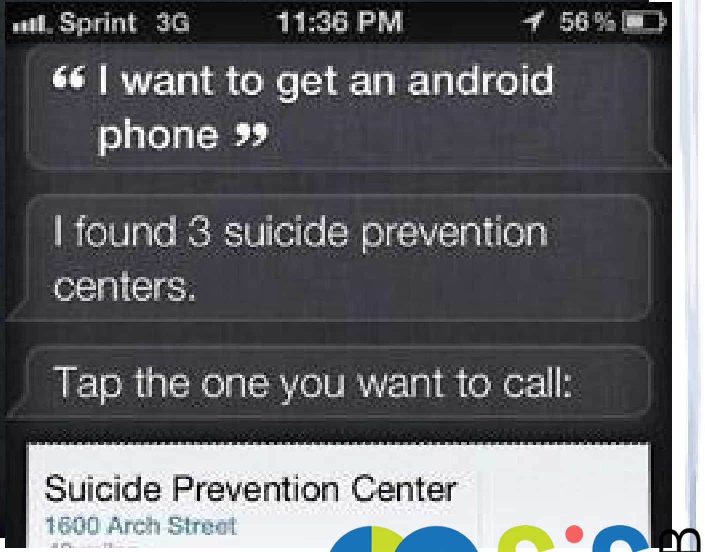
To: Michelle Carter

I love her



“ Call me an ambulance ”

From now on, I'll call you 'An Ambulance'. OK?



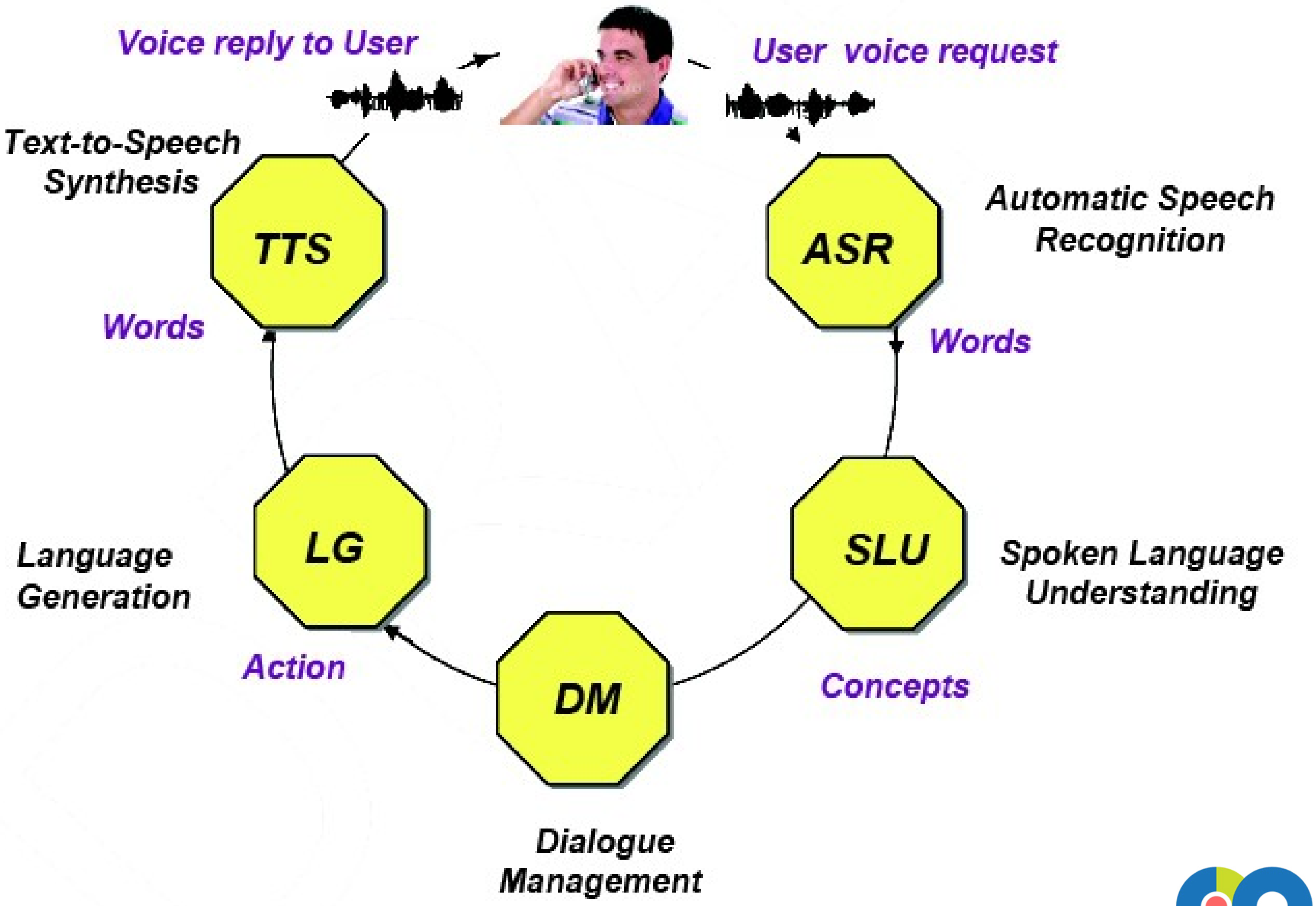
“ I want to get an android phone ”

I found 3 suicide prevention centers.

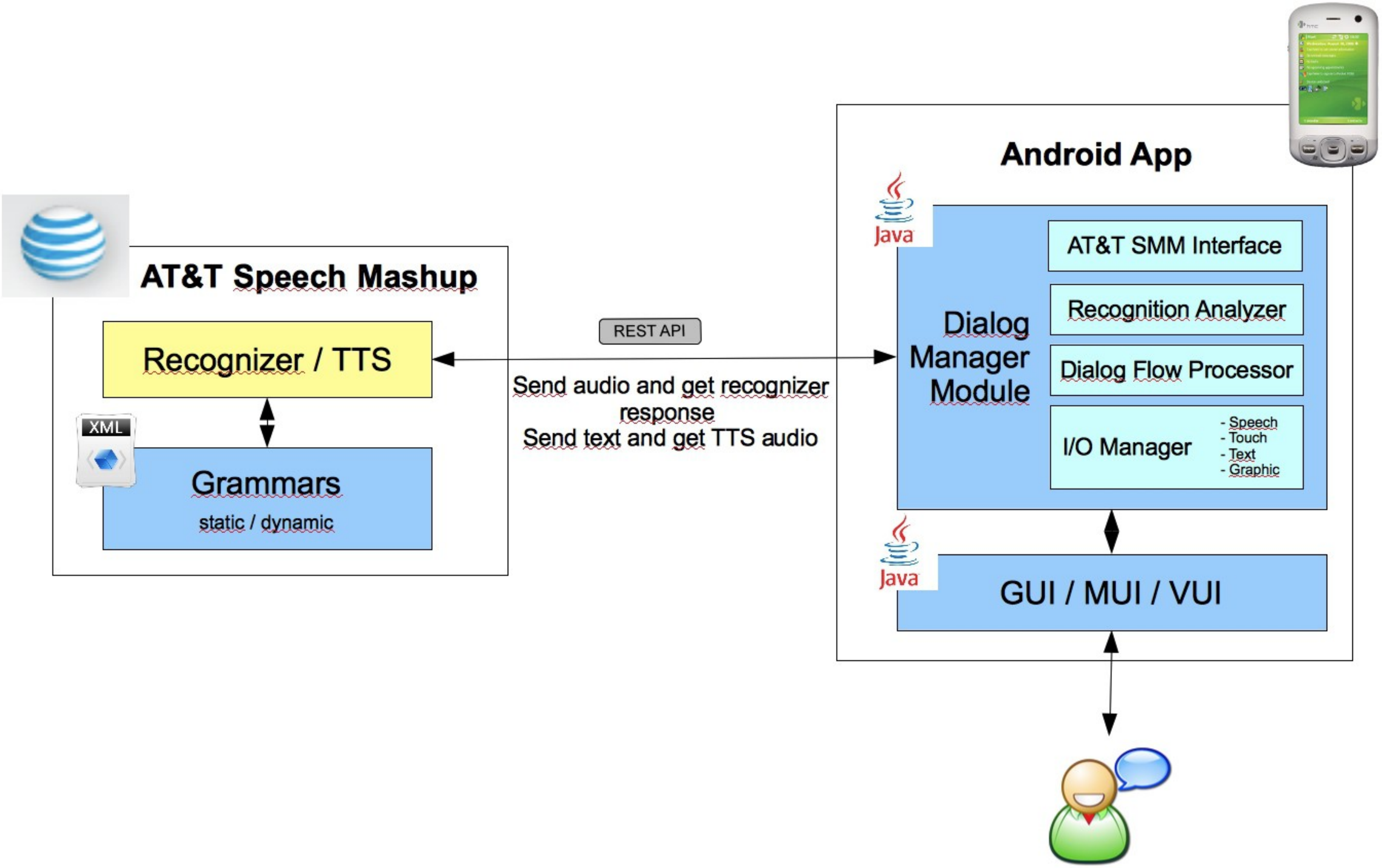
Tap the one you want to call:

Suicide Prevention Center
1600 Arch Street

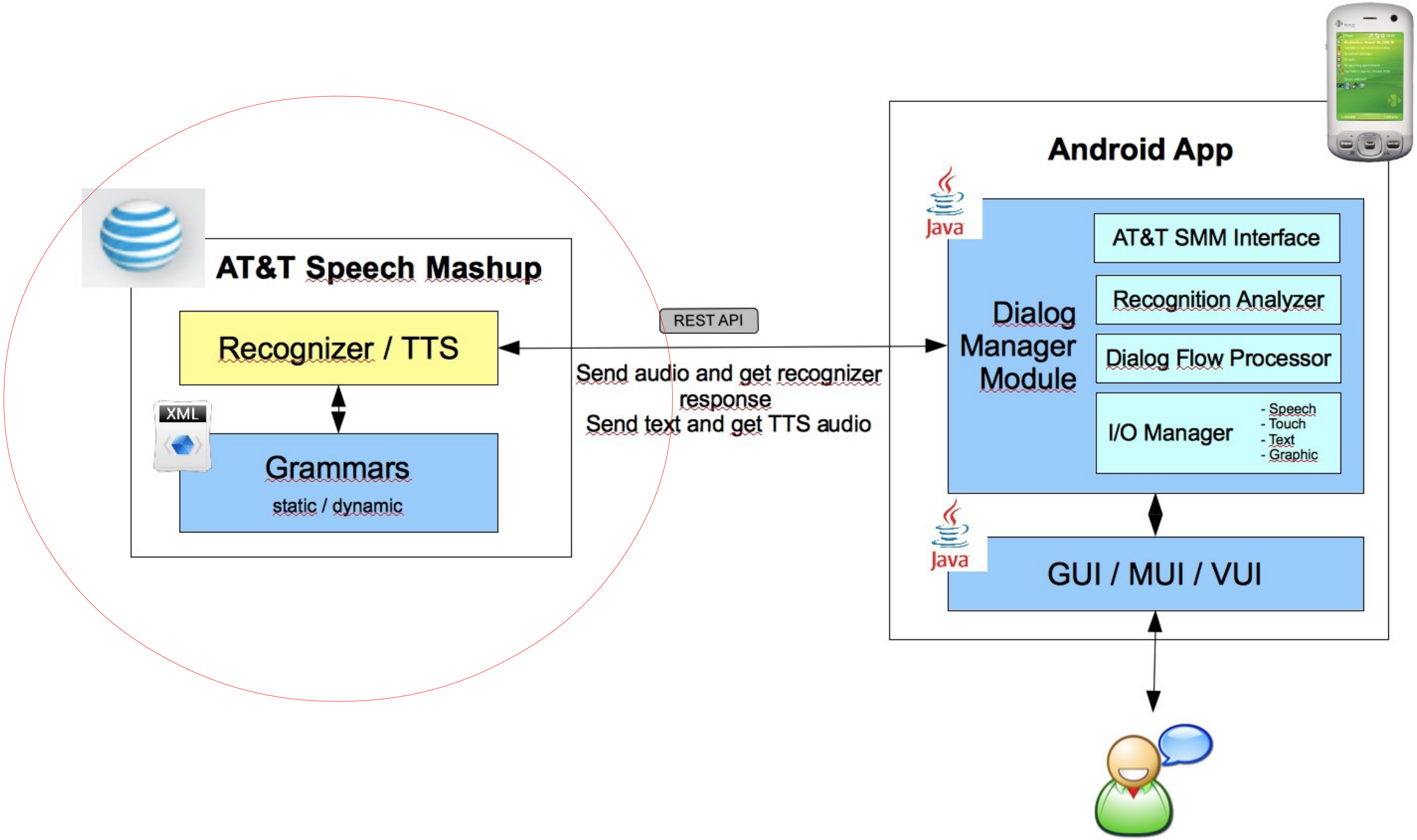
Basic architecture of a generic Spoken Dialogue System



A possible architecture



A possible architecture



SiS LAB
Signals & Interactive Systems

AT&T Speech Mashup

What's AT&T Speech Mashup

An AT&T speech mashup portal is a web service that implements speech techonologies, including both automatic speech recognition (ASR) and text to speech (TTS) for web application

Speech mashup can be created for almost any mobile device, including the iPhone, as well as web browsers running on a PC or Mac, or any othe network-enabled device with audio input

Using it, then, we can create complex speech applications using all the AT&T developing instruments.

AT&T Speech Mashup

What's AT&T Speech Mashup – Watson ASR

One of the fundamental component of the Mashup is the Watson ASR.

The Watson ASR is the automatic speech recognition component of the WATSON system responsible for converting spoken language to text.

Recognition main steps are:

- Identify the speech features
- Map features to basic language sounds contained in the acoustic model
- Match sounds to phrases and sentences in the grammar

AT&T Speech Mashup

What's AT&T Speech Mashup – Grammars

ASR refers to user defined grammars to match sounds.

Actually the admitted grammar formats are the XML standard (W3C) usually called GRXML and the deprecated proprietary Watson BNF (WBNF)

As we are going to see it's possible to upload grammars or use the shared and builtin versions provided by the portal

AT&T Speech Mashup

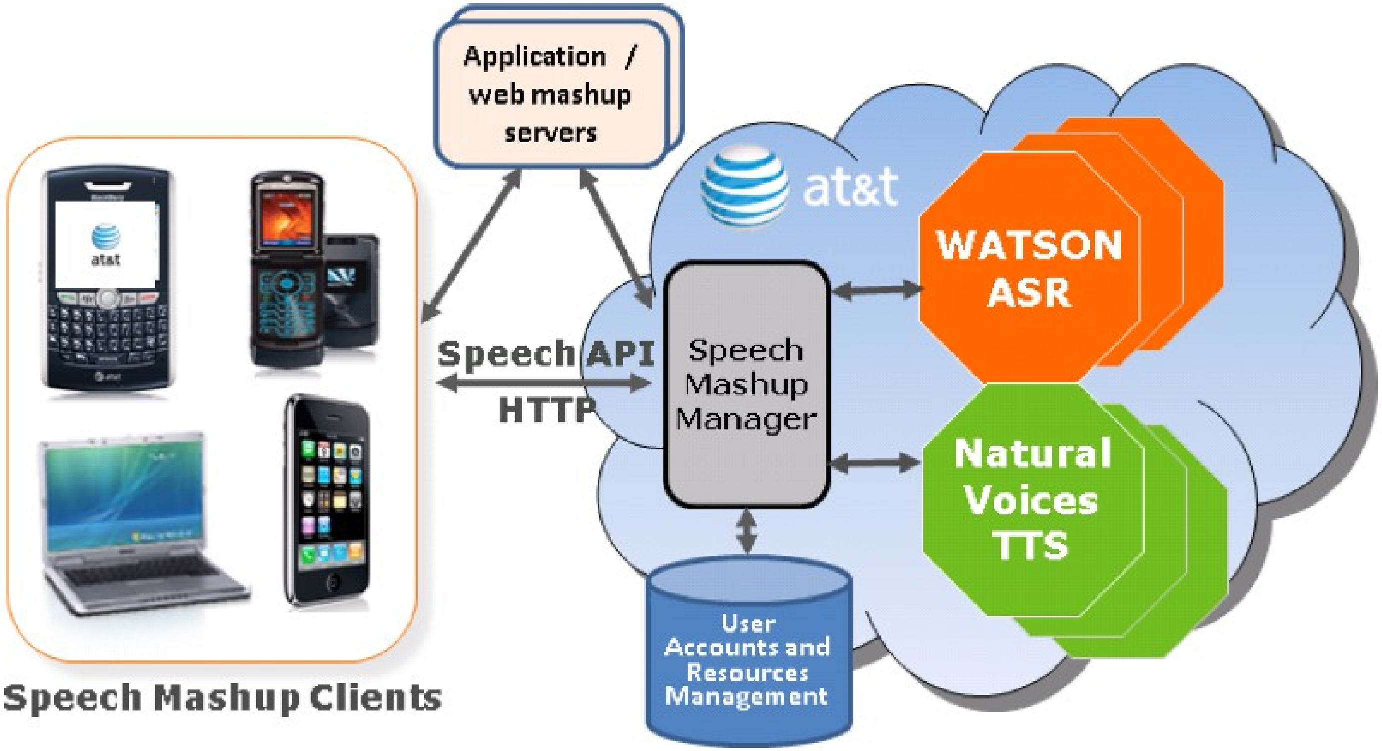
What's AT&T Speech Mashup – TTS

The TTS, called Natual Voices, has bult-in rules for normalizing text (such as converting common abbreviations to words) and assigning prosody to make the generated speech sounds as natural as possible.

In addition, Natural Voices (the TTS System) properly interpret Synthesized Speech Markup Language (SSML) tags embedded in the text to more closely control normalization, pronunciation and prosody

AT&T Speech Mashup

AT&T Speech Mashup Architecture



AT&T Speech Mashup

AT&T Speech Mashup Web Portal

AT&T Speech Mashup provide a web portal to test and manage applications you create using the API

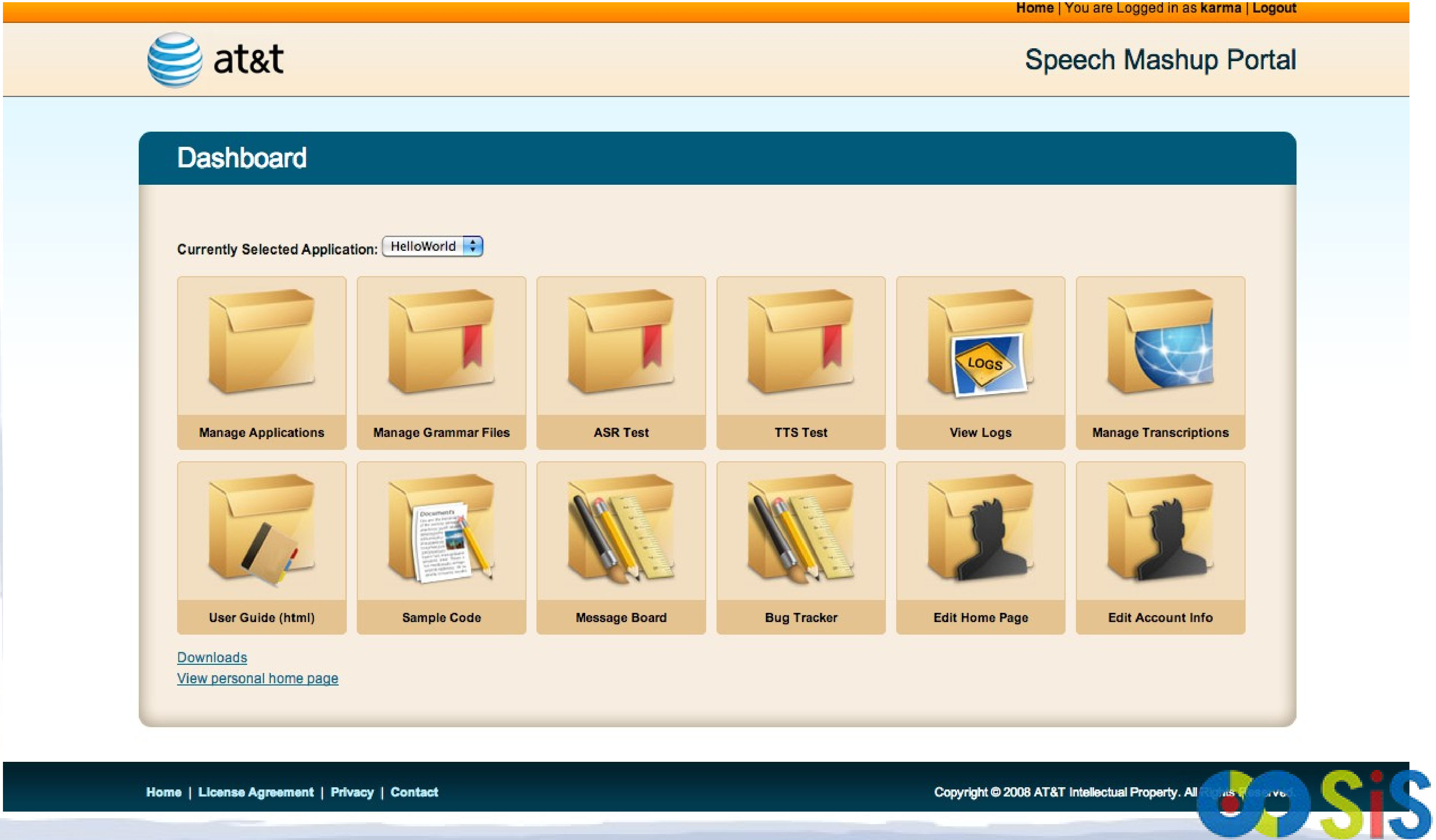
To use it and the API just register at the link:

https://service.research.att.com/smm/

You'll get the access to the platform and a unique UUID to send as a parameter when using the webservice

AT&T Speech Mashup

AT&T Speech Mashup Web Portal



Home | You are Logged in as **karma** | Logout

at&t

Speech Mashup Portal

Dashboard

Currently Selected Application: HelloWorld

Downloads

View personal home page

Home | License Agreement | Privacy | Contact

AT&T Speech Mashup

AT&T Speech Mashup Web Portal

Sections:

- **Manage Application**: in this page you can create different applications containing different grammars and dictionaries
- **Manage Grammar Files**: here you can upload, compile and view grammars
- **ASR Test**: In this section is possible to test the grammars by instantly recording an audio file
- **TTS Test**: in this page is possible to test the TTS by writing some text to be read
- **View Logs**: page containing all the logs of the applications
- **Manage Transcription**: this link open the interface for transcribing the recorded and uploaded audio files, so that it's possible to evaluate the recognition results
- **User Guide**: link to download the official guide

…

AT&T Speech Mashup

AT&T Speech Mashup Web Portal

…

- **Sample Code**: link to download the zipped file containing the clients examples
- **Message Board**: Link to google groups to ask about the AT&T Speech Mashup
- **Bug tracker**: Link to Bugzilla to report application bugs
- **Edit Home Page**: in this form you can write the HTML for your personal home page. The link to your personal homepage is below the two images rows
- **Edit Account Info**: in this page it's possible to change password, email and other fields associated to your profile

AT&T Speech Mashup

Web Portal functionalities

The portal, then, provide the following useful functionalities:

- Create and edit applications
- Upload, delete, rename, edit and view grammars
- Compile uploaded grammars even using special options, like SpeedVsAccuracy, vadSensitivity and nbest or changing the acoustic model and the associated dictionary
- Share grammars with all the other users. In future versions will be possible also to select users you want to share the grammars with
- Upload, delete, rename and edit dictionaries
- Istantly test the ASR selecting which grammar to use for the recognition process
- Get the ASR results in different formats: JSON (flat or nested slots), Watson JSON (indented or not), XML and EMMA
- Test TTS voices, even selecting the voice, using SSML Markup, getting notification on bookmarks, phonemes, viseme or word and getting the results in two possible formats: simple or ogg

AT&T Speech Mashup

Web Portal functionalities

…

- Creating your own voice by uploading audio or using their interface for registering it. This part of the portal is not in documentation yet
- Check logs of all the applications
- Create transcriptions, selecting audio files to transcript by filtering per date
- Evaluate results with external tools after downloading transcription files

In addition the portal permits to set two URLs to be invoked before the ASR and after it. Through these options it's possible to modify the input parmeters (like the audio got from the user speech) using an external webservice and send the elaborated data as input for the ASR and to elaborate the results before sending it back to the client, so that you can send different types of data, or use other statistics to decide which of the nbest it's better to use.

This method permits to upgrade the performances of the system, without modifying the client software.

Signals & Interactive Systems

AT&T Speech Mashup

Into details: XML Grammars

This grammar matches only the words ”internet”, ”call” and ”map”.

```
<grammar version="1.0" tag-format="semantics/1.0" xml:lang="en-US" root="word">
    <rule id="word">
        <item repeat="1">
            <one-of>
                <item>internet</item>
                <item>call</item>
                <item>map</item>
            </one-of>
        </item>
    </rule>
</grammar>
```

AT&T Speech Mashup

Into details: XML Grammars

<one-of> tag create a list in which one of the contained <item> is possible

Repeat attribute set how many times the item should be repeated. If there isn't this attribute with a “0-1” value, the item must be said from the user

The special rule GARBAGE (<ruleref uri="GARBAGE"/>) define everything.

The weight attrbute in the item tags define the weight to be associated to the word in the generated finite state machine. It must be between 0.0 and 1.0

If using the tag-format semantic in the definition of the grammar (<grammar tag-format="semantics/1.0" root="object">) then, it's possible to add a <tag> element to override the returned value of a grammar component using a script. Example:

```
<rule id="object">
  <one-of>
      <item>home        <tag> out="newloan"   </tag> </item>
      <item>refinancing  <tag> out="refi"      </tag> </item>
      <item>refinance    <tag> out="refi"      </tag> </item>
      <item>loan         <tag> out="newloan"   </tag> </item>
      <item>interest     <tag> out="rates"     </tag> </item>
      <item>rate         <tag> out="rates"     </tag> </item>
      <item>rates        <tag> out="rates"     </tag> </item>
  </one-of>
</rule>
```

An example of a grammar (1/2)

```
<grammar version="1.0" tag-format="semantics/1.0" xml:lang="en-US" root="main">
    <rule id="main">
        <item weight="0.1" repeat="0-1"><ruleref special="GARBAGE"/></item>
        <ruleref uri="#first"/>
        <ruleref uri="#preintent"/>
        <ruleref uri="#intent"/>
        <ruleref uri="#verb"/>
        <item weight="0.1" repeat="0-1"><ruleref special="GARBAGE"/></item>
        <ruleref uri="#article"/>
        <ruleref uri="#filters"/>
        <ruleref uri="#business"/>
        <ruleref uri="#place"/>
        <ruleref uri="#regards"/>
        <item weight="0.1" repeat="0-1"><ruleref special="GARBAGE"/></item>
        <tag>
          out.intent=rules.intent.intent;
          out.category=rules.business.category;
          out.place=rules.place.place;
          out.filterby=rules.filters.filterby;
        </tag>
    </rule>
```

Rules

Semantic returns
of the grammars

An example of a grammar (2/2)

```
<rule id="intent">
        <tag>
            out.intent="";
        </tag>
        <item repeat="1">
            <one-of>
                <item>I'd like<tag>out.intent="search";</tag></item>
                <item>see<tag>out.intent="search";</tag></item>
                <item>visit<tag>out.intent="search";</tag></item>
                <item>want to book<tag>out.intent="reserve";</tag></item>
                <item>book<tag>out.intent="reserve";</tag></item>
                <item>want to reserve<tag>out.intent="reserve";</tag></item>
                <item>reserve<tag>out.intent="reserve";</tag></item>
                <item>want to reserve<tag>out.intent="reserve";</tag></item>
                <item>want<tag>out.intent="search";</tag></item>
                <item>I'm looking for<tag>out.intent="search";</tag></item>
                <item>take me to<tag>out.intent="navigate";</tag></item>
                <item>take us to<tag>out.intent="navigate";</tag></item>
                <item>find<tag>out.intent="search";</tag></item>
                <item>where<tag>out.intent="navigate";</tag></item>
                <item>need<tag>out.intent="search";</tag></item>
                <item>go to<tag>out.intent="navigate";</tag></item>
                <item>get me to<tag>out.intent="navigate";</tag></item>
                <item>get us to<tag>out.intent="navigate";</tag></item>
                [...]
```

Declaration of a rule

Semantic meaning

Matched word

AT&T Speech Mashup

Into details: SSML Markup

SSML is a standardized XML markup language for modifying the way text is processed by TTS engines. Some of this markups are:

- Lenght of a pause: Begin <Break time="3s"/> now
- Say as: <say-as interpret-as="value"> text </say-as> (value can be acronym, address, currency, date, ignore-case, lines, literal, math, measurement, number, spell, telephone, time). Example: <say-as interpret-as="date" format="dmy"> 1/2/2008 </say-as>
- Speak tag, to define language to use: <speak xml:lang="en_us">
- Voice tag, to set the voice to use, or the language: <voice name="mike"> text </voice>
- Paragraph tag, to define a paragraph or change language: <p xml:lang="en_us">
- Sentence tag (or <s>, to set a sentence or change the language: <s xml:lang="es_us">
- Mark tag, to send back a bookmark to the client: <mark name="bm1"/>
- Break tag, to set a pause: <break strength="level"/> (level can be none, x-weak, weak, medium, strong, or x-strong)

Signals & Interactive Systems

AT&T Speech Mashup

Into details: SSML Markup

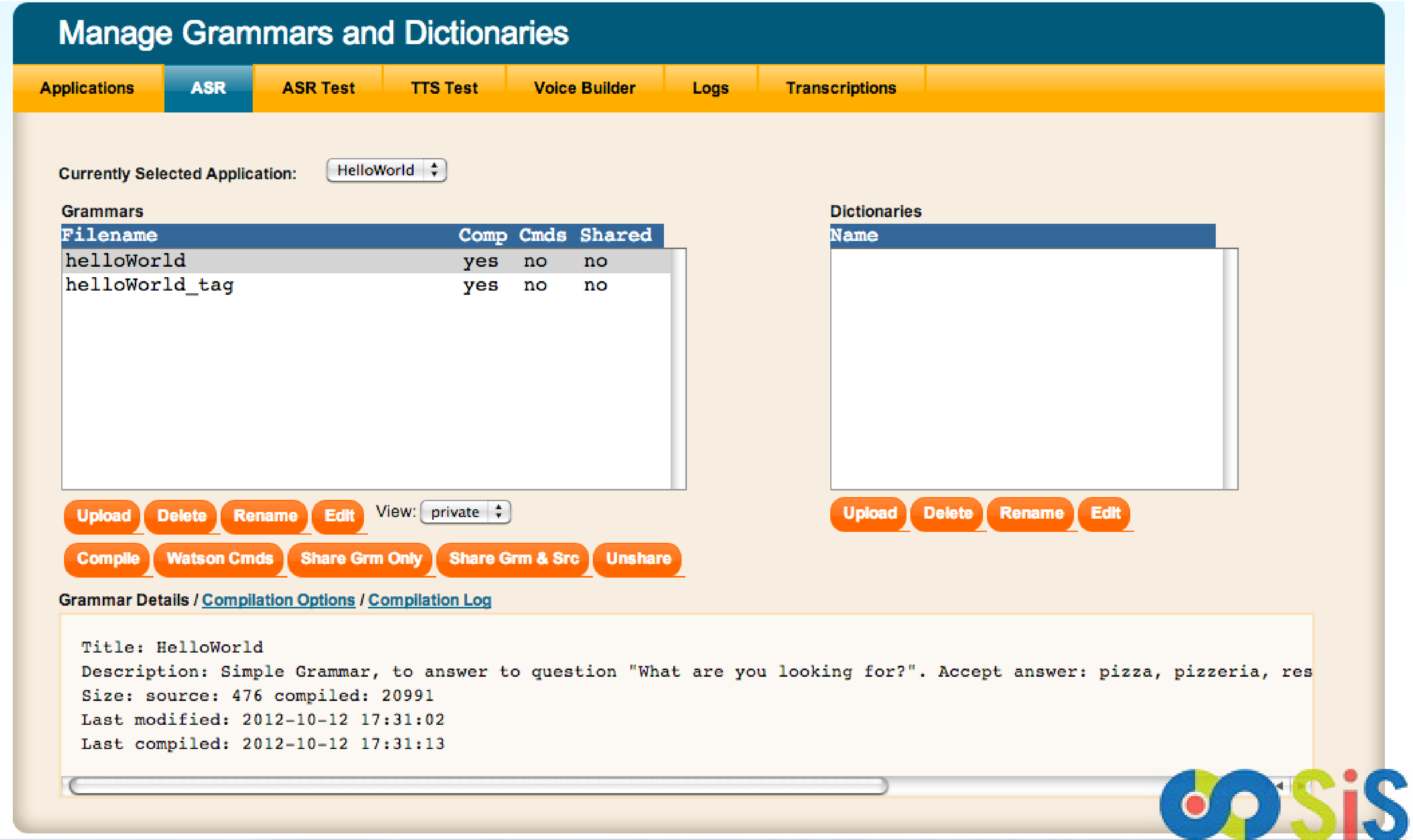
…

- Prosody volume level: <prosody volume="level | n | n%">text</prosody> (level can be can be silent, x-soft, soft, medium, loud, x-loud, default; number between 1 and 100)
- Prosody rate value: <prosody rate="value"> text</prosody> (value can be x-fast, fast, medium, slow, x-slow, or default or a percentage in this form: 2 = 200%, 0.5 = 50% of the default rate)
- Prosody Emphasis, to define the emphasis to use: <emphasis level="level">text </emphasis> (level can be strong, moderate, none, or reduced)
- Phoneme tag, to specify a particular pronunciation: <phoneme alphabet="darpa" ph="f aa 1 dh er 0"/>

More informations at https://service.research.att.com/smm/download/SpeechMashupGuide.html.zip/#_Toc295729831

AT&T Speech Mashup

Into details: Grammar Tools



Currently Selected Application: HelloWorld

View: private

Grammars

Filename	Comp	Cmds	Shared
helloWorld	yes	no	no
helloWorld_tag	yes	no	no

Dictionaries

Name

Grammar Details / Compilation Options / Compilation Log

```
Title: HelloWorld
Description: Simple Grammar, to answer to question "What are you looking for?". Accept answer: pizza, pizzeria, res
Size: source: 476 compiled: 20991
Last modified: 2012-10-12 17:31:02
Last compiled: 2012-10-12 17:31:13
```

AT&T Speech Mashup

Into details: Grammar Tools

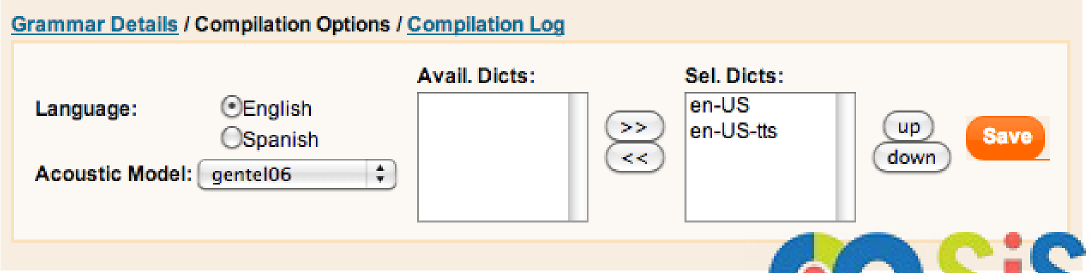
The grammaras tools are separated in 4 main sections:

- The list of the grammars
- The list of the dictionaries
- The actions available for the selected grammar
- The box with details, compilation options and logs

To create a grammar you need to create it locally on your PC and the upload it on the portal.

After you upload the grammar you need to compile it. Then you can try to change the options for compiling. In the Compilation Options section you can select language (english or spanish), acoustic model and dictionaries to use.

There is no documentation about the available acoustic models.



Grammar Details / Compilation Options / Compilation Log

Language: ● English ○ Spanish

Acoustic Model: gentel06

Avail. Dicts:

Sel. Dicts: en-US, en-US-tts

AT&T Speech Mashup

Into details: Grammar Tools

In the logs section, finally, you can check the logs when compiling the grammar.

When compiling a grammar it's possible to set even more parameters by pressing on the button “Watson Cmds”. These are the paramters that WATSON admits tha must be in the form “*set name=value*”. Possible parameters are:

- config.speedVsAccuracy

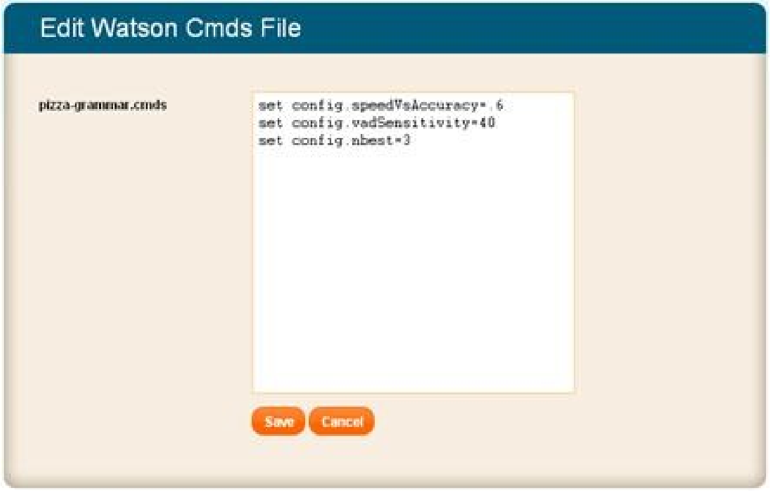
    Value range: 0.0 – 1.0 (default is .5)

- config.vadSensitivity

    Value range: 1-100 (default is 50)

- config.nbest

    Value range: n (default is 1)



Edit Watson Cmds File

pizza-grammar.cmds

```
set config.speedVsAccuracy=.6
set config.vadSensitivity=40
set config.nbest=3
```

Modifying these parameters you can set if the grammar should be more accurate or faster (speedVsAccuracy) or to be more or less sensitive when determing that audio is speech (vadSensitivity). With the nbest parameter, finally, you can set how many results you want the grammar to give you back.

AT&T Speech Mashup

Into details: Grammar Tools

You can even set the endpointing parameters, to make application decide automatically when the user start and stop speaking:

activateEvh "timeouts"

activateEvh "speechstart-hmm"

timeouts.firstTimeout = 400

timeouts.secondTimeout = 500

After the first timeout millisecond of silence WATSON will give you back a result only if it has a confidence score higher than the recognition threshold.

After the second timeout, insetad, it will give you back the result in any case.

It's possible to set the Cmds even through the API by adding parameters in a string like this:

...&control=activateEvh+%22timeouts%22%3BactivateEvh+%22speechstart-hmm%22%3Btimeouts.firstTimeout+=+400%3Btimeouts.secondTimeout+=+500

AT&T Speech Mashup

Into details: Grammar Tools

Unfortunately the documentation does not specify how to set the recognition threshold even if it say that you may want to set it in the application.

The GRXML grammar, in addition, in the W3C standard explicity says:

"*Speech recognizer configuration: The grammar format does not incorporate features for setting recognizer features such as timeouts, recognition thresholds, search sizes or N-best result counts.*"

Then, at the moment, is not known where to set the recognition threshold for the application.

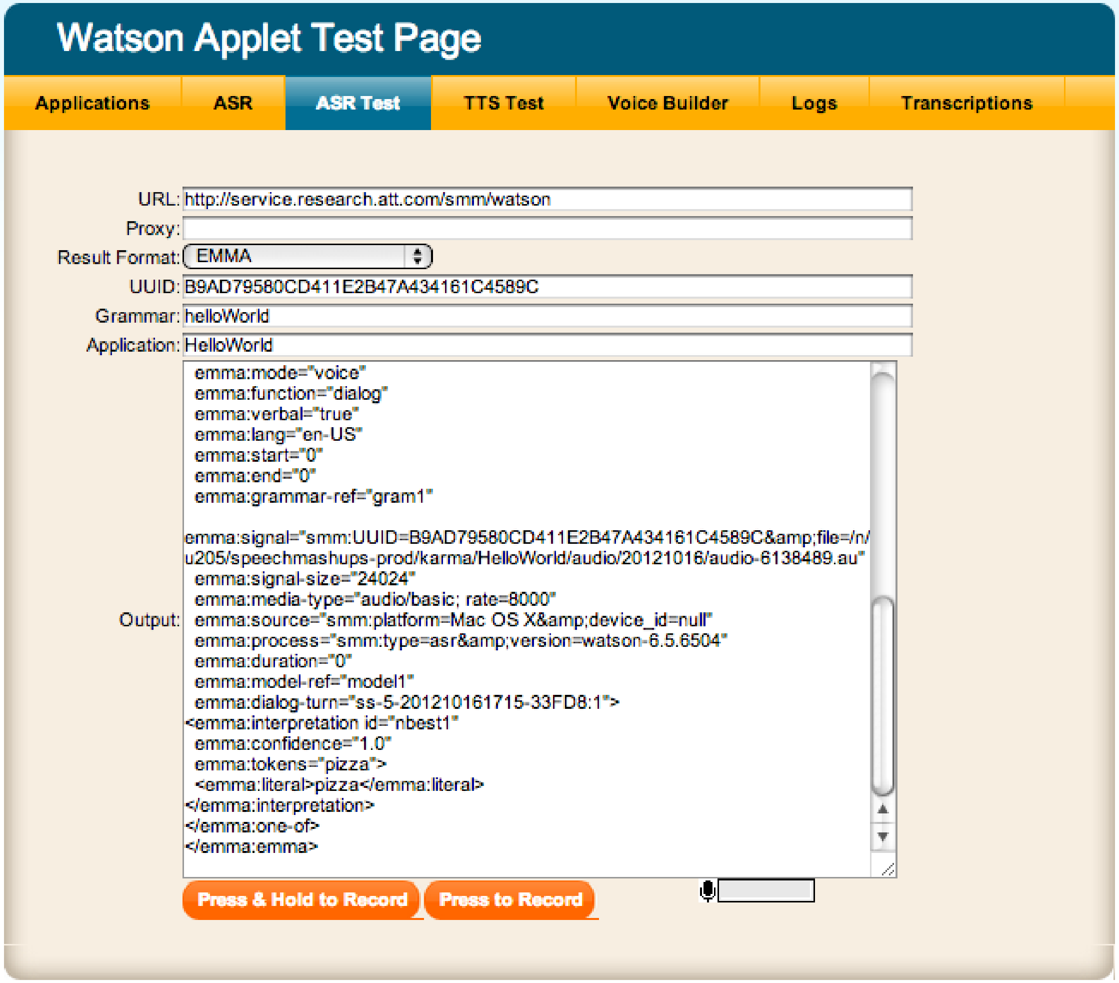
About dictionaries the documentation only says that there are 2 different dictionaries: a general large dictionary and another dictionary for TTS that generate spell for words not contained in other dictionaries. Nothing else is said, but it seems that you can add dictionaries and include them during the grammar compilation, to obtain the desired results.

AT&T Speech Mashup

Into details: ASR Tools

The Portal gives you also the opportunity to test the grammar instantly. In the ASR Test page you can choose the grammar to use, the application and the result format.

Then, thanks to a Java applet, you can press a button and start speaking. The recorded voice, will be processed using the grammar and the application, and the interface will return the result of the ASR.



URL: http://service.research.att.com/smm/watson

Proxy:

Result Format: EMMA

UUID: B9AD79580CD411E2B47A434161C4589C

Grammar: helloWorld

Application: HelloWorld

Output:

```
 emma:mode="voice"
 emma:function="dialog"
 emma:verbal="true"
 emma:lang="en-US"
 emma:start="0"
 emma:end="0"
 emma:grammar-ref="gram1"

emma:signal="smm:UUID=B9AD79580CD411E2B47A434161C4589C&amp;file=/n/
u205/speechmashups-prod/karma/HelloWorld/audio/20121016/audio-6138489.au"
 emma:signal-size="24024"
 emma:media-type="audio/basic; rate=8000"
 emma:source="smm:platform=Mac OS X&amp;device_id=null"
 emma:process="smm:type=asr&amp;version=watson-6.5.6504"
 emma:duration="0"
 emma:model-ref="model1"
 emma:dialog-turn="ss-5-201210161715-33FD8:1">
<emma:interpretation id="nbest1"
 emma:confidence="1.0"
 emma:tokens="pizza">
 <emma:literal>pizza</emma:literal>
</emma:interpretation>
</emma:one-of>
</emma:emma>
```

AT&T Speech Mashup

Into details: TTS Tools

The TTS Test Portal section permits to test the TTS to decide which voice is better to use and which special tag can be added to make the voice as much real as possible.

The input fields are:

- The application to use
- The text to speech
- The voice to use (Crystal and Rosa are female voices, Mike and Alberto are male; Crystal and Mike are English while Rosa and Alberto are Spanish. The versions whose names end in 16 are 16-bit voices; the others are 8-bit)
- Use or not SSL Markup
- Notification to send
- The container type of the generated audio file

There is also an output field:

- The returned message from bookmarks

AT&T Speech Mashup

Into details: TTS Tools

We can try the TTS just by inserting the text and pressing the “Play Prompt” button.

Let's try something:

- *Now <prosody volume="x-loud">I can speak loudly</prosody> but <prosody volume="x-soft">I can speak even softly</prosody>.*
- *I can do even break <Break time="3s"/> while speaking or <prosody rate="x-slow">change my rate while speaking</prosody> <prosody rate="0.01">from slow</prosody><prosody rate="1.5">to the highest speed ever</prosody>*
- *I want to emphasis that <emphasis level="strong">I am really happy today</emphasis>while<emphasis level="reduced">yesterday I was really really sad </emphasis>*
- *Do I have to pronunciate father in this way: <phoneme alphabet="darpa" ph="f aa 1 dh er 0"/> or in this way <phoneme alphabet="darpa" ph="p aa 1 t er 0"/>?*
- *I can say a string in different ways: as a date: <say-as interpret-as="date" format="dmy"> 10/12/2012 </say-as> or as a math calculation <say-as interpret-as="math"> 10/12/2012 </say-as>*
- *Would you like to know when I'm saiyng this <mark name="this_bookmark"/> so that you can start with another process?*

AT&T Speech Mashup

Into details: Logging Tools

In the logs it's possible to select the day to check the log of, and to read all the informations about the gotten processes.

In the log are available the following informations:

- Date and time of the request
- Passed parameters (like appname, cmd, grammar, resultFormat, uuid)
- WATSON response and logs (like debug, version, session_id, hypotesis, audio file with the link to download it and so on)
- The EMMA response (or another response type if another one has been selected)
- The post processing detail and logs
- The TTS logs in case of a TTS request

AT&T Speech Mashup

API and Clients developing

AT&T Speech Mashup provide also API to develop applications for quite all clients.

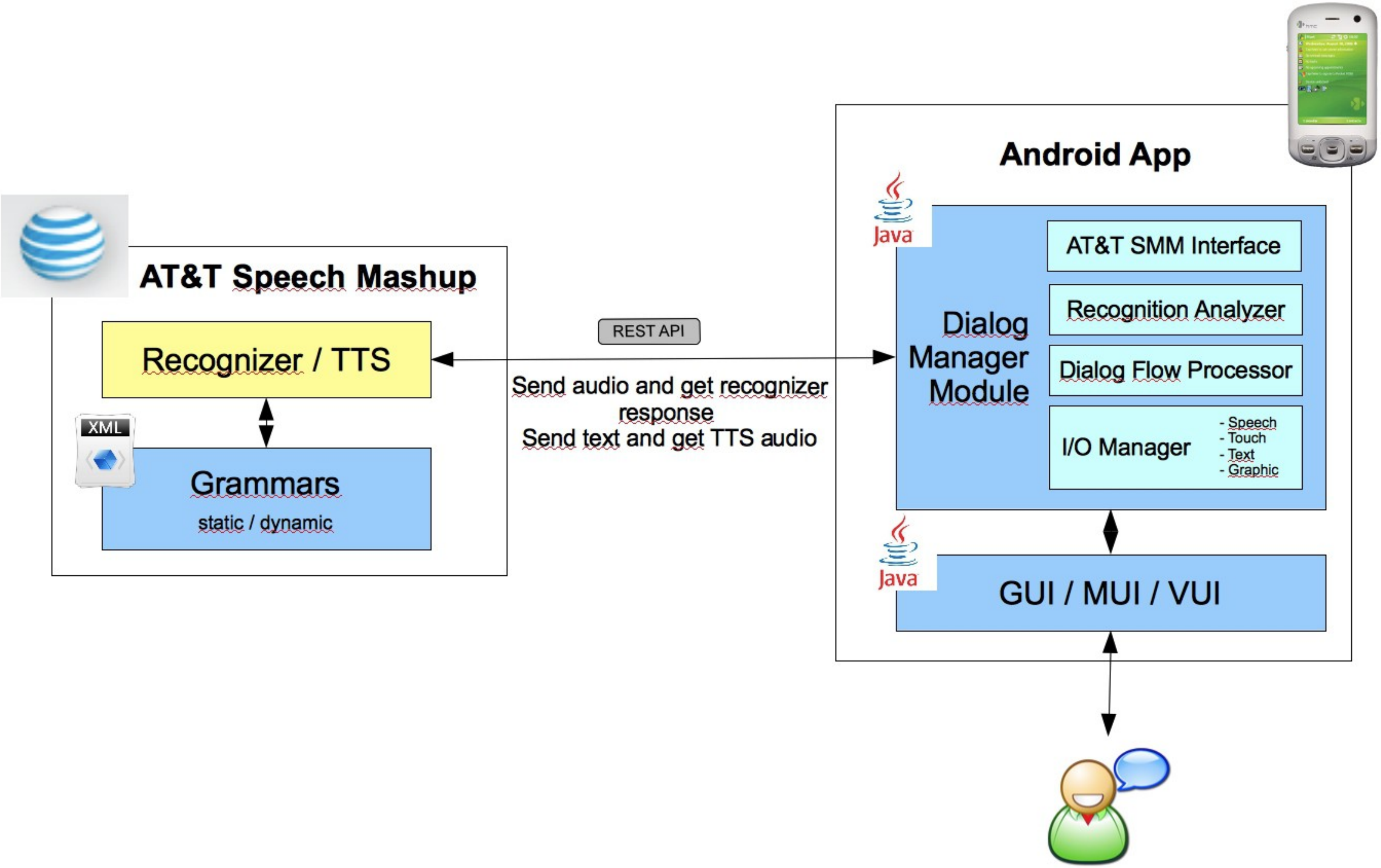
The API functionalities are the same explained before for the web portal.

Supported platforms are iPhone, Android, Blackberry and other devices supporting Java, web browsers supporting java like Safari, Firefox, Chrome and Internet Explorer.

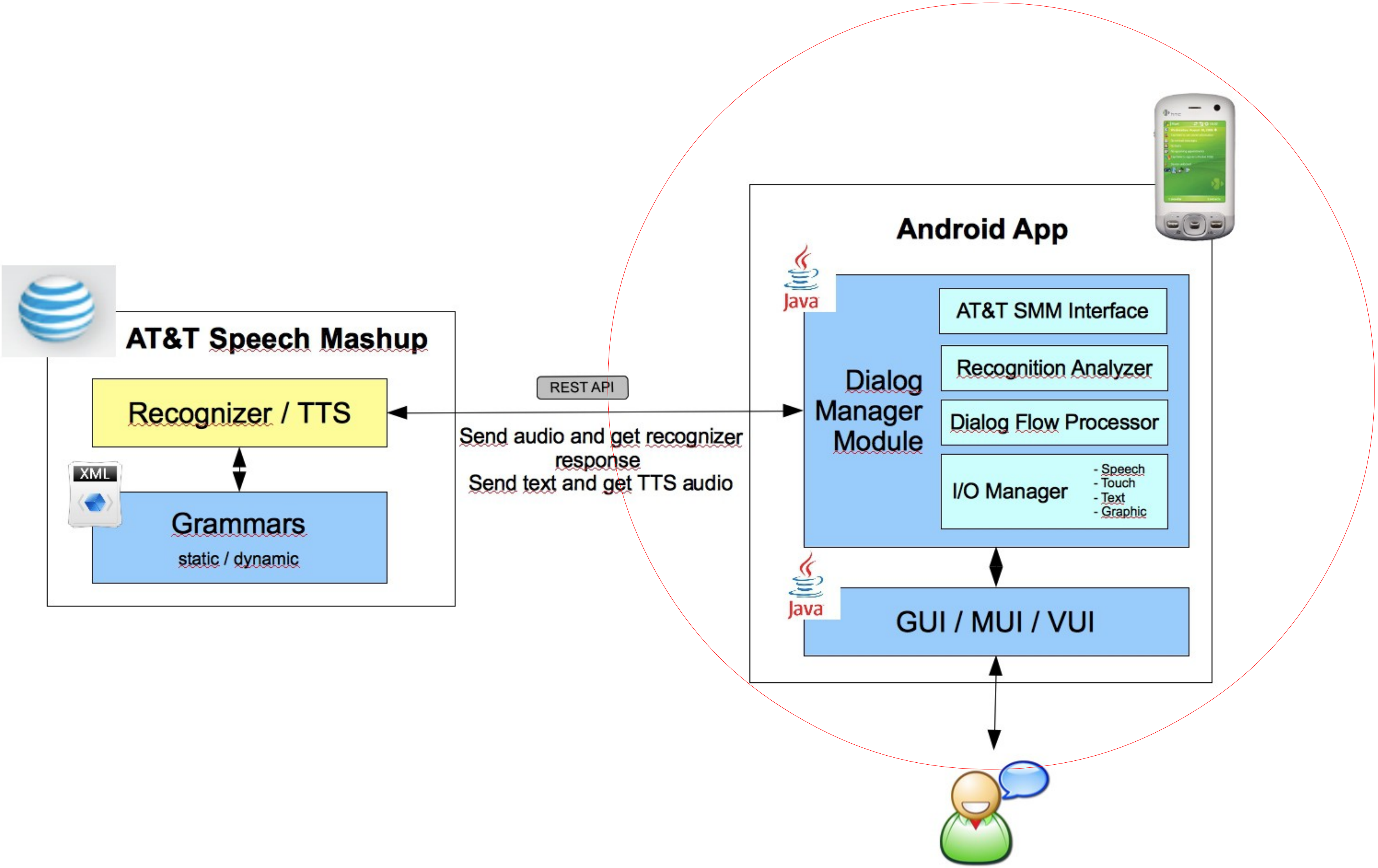
The API can be used using the REST principles, even with the simple use of the command wget. Example:

```
wget \
  --post-file=sample-date.amr \
  --header 'Content-Type: audio/amr' \
  --server-response 'http://service.research.att.com/smm/watson?cmd=rawoneshot&grammar=en-us-date&uuid=<your own
UUID>&appname=<application ID>&resultFormat=emma' \
  -O response.emma
```

A possible architecture

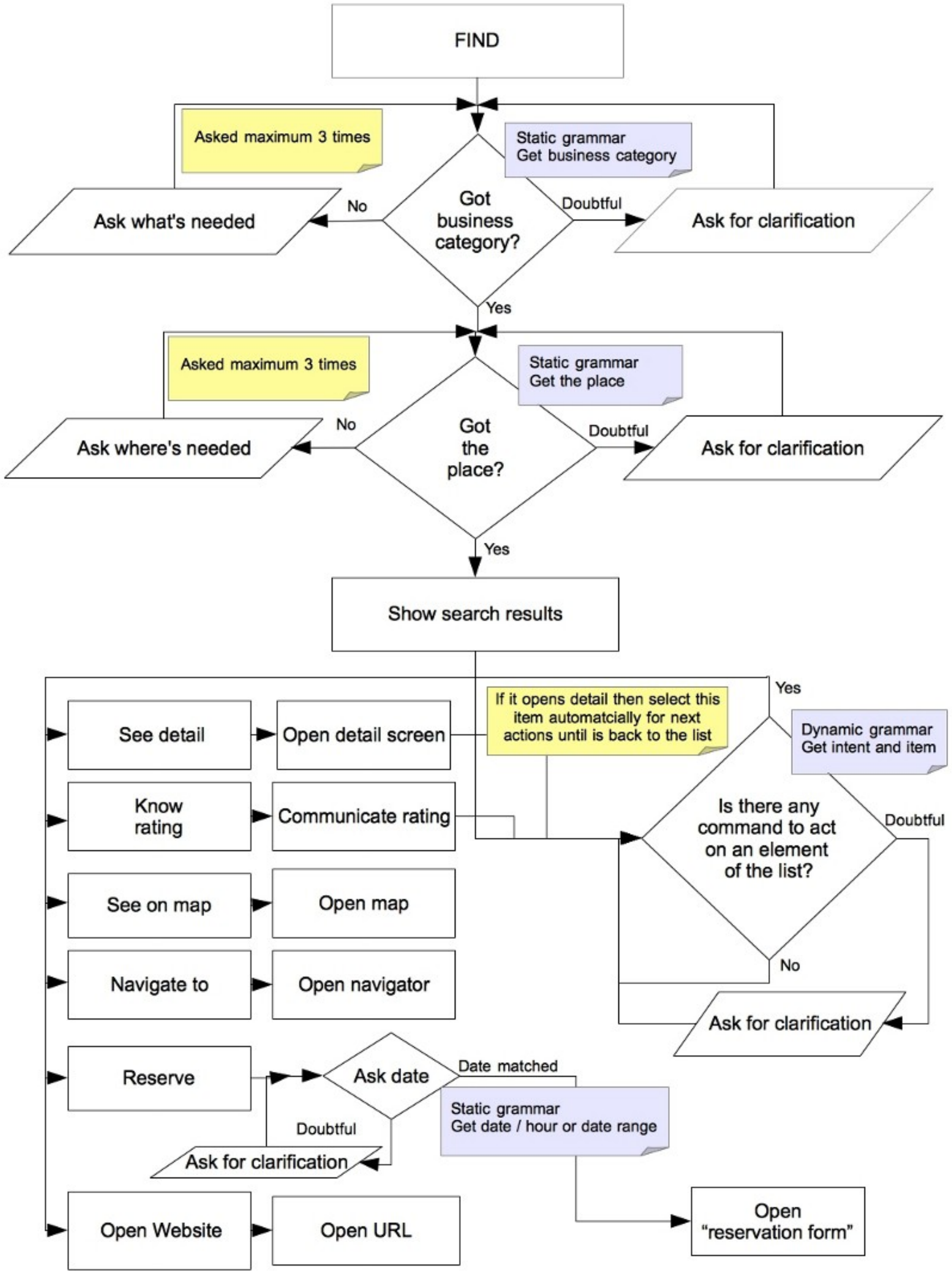


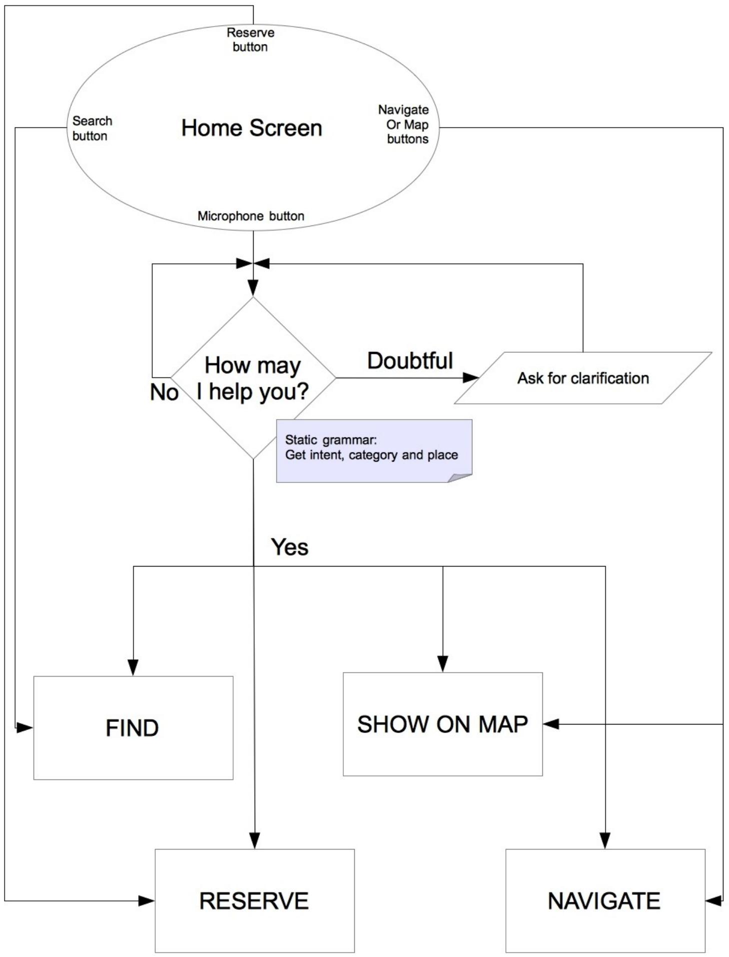
A possible architecture



Signals & Interactive Systems LAB

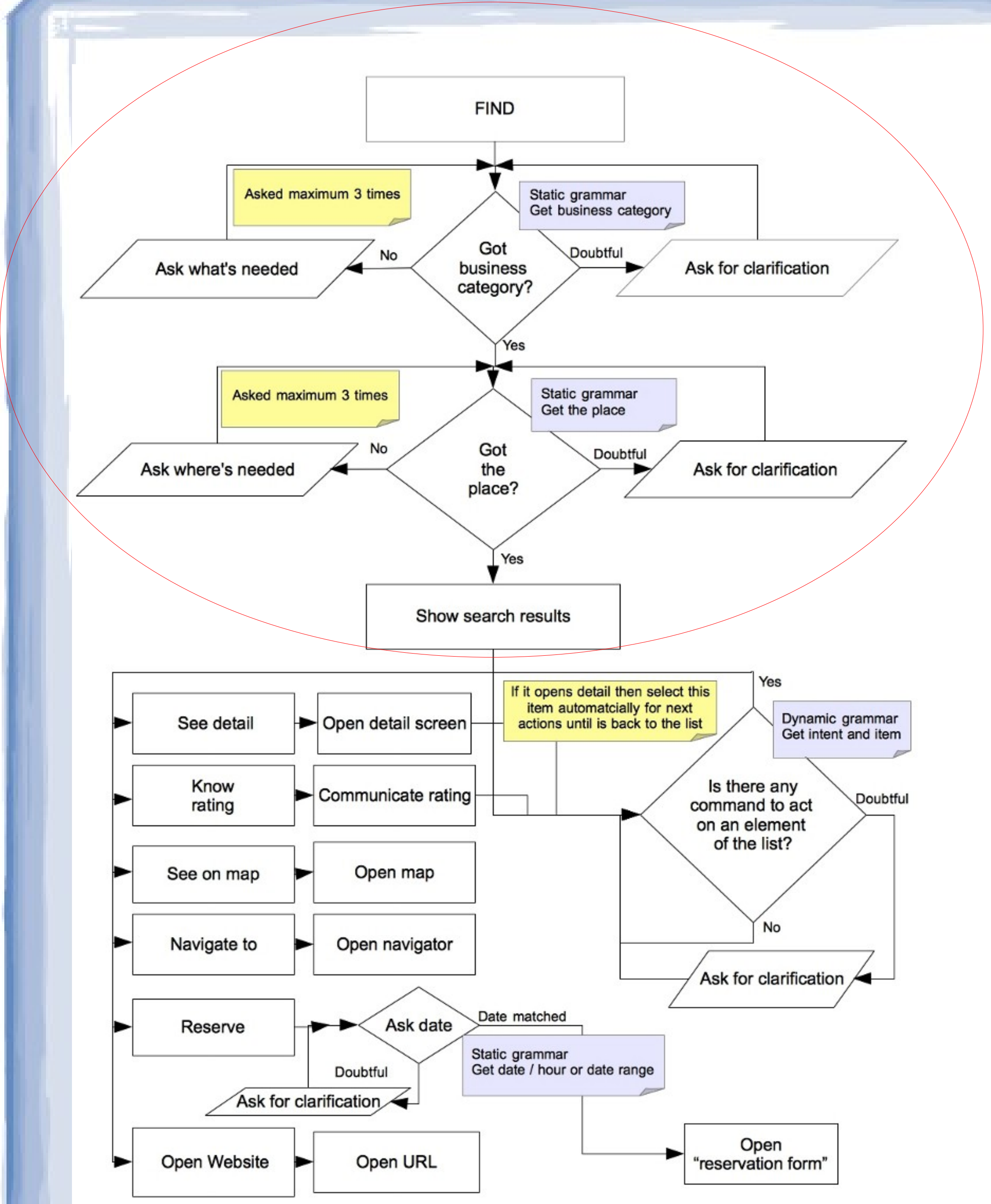
Dialogue flow example

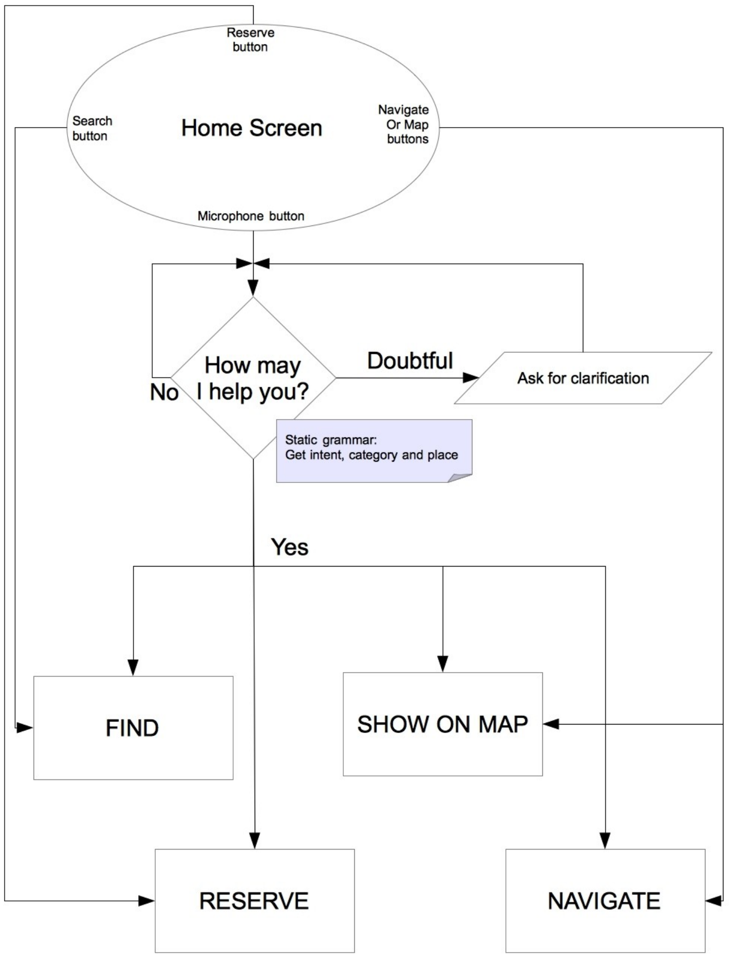




SiS LAB
Signals & Interactive Systems

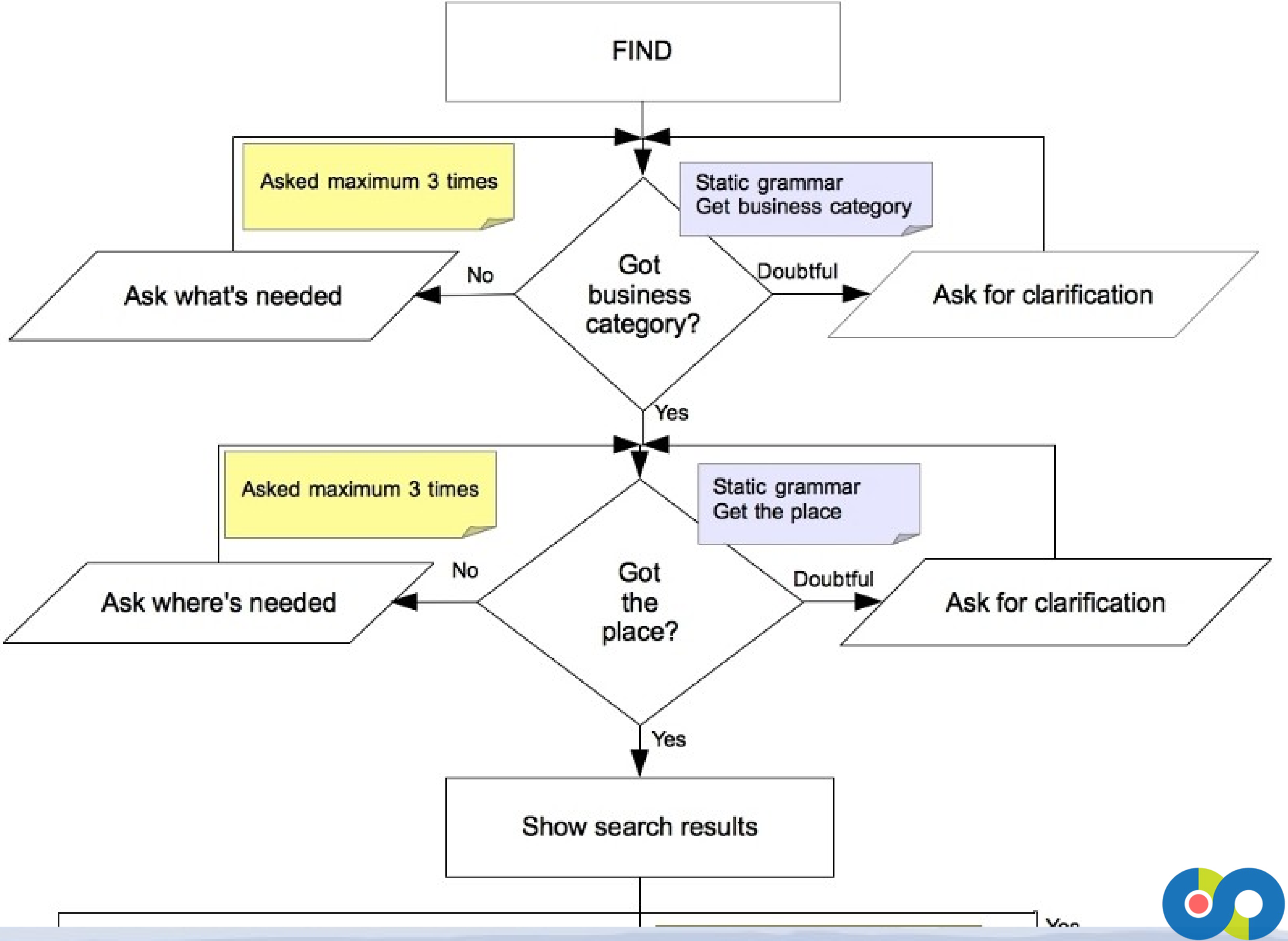
Dialogue flow example





SiS LAB
Signals & Interactive Systems

Dialogue flow example



FIND

Asked maximum 3 times

Static grammar
Get business category

Got business category?

No

Ask what's needed

Doubtful

Ask for clarification

Yes

Asked maximum 3 times

Static grammar
Get the place

Got the place?

No

Ask where's needed

Doubtful

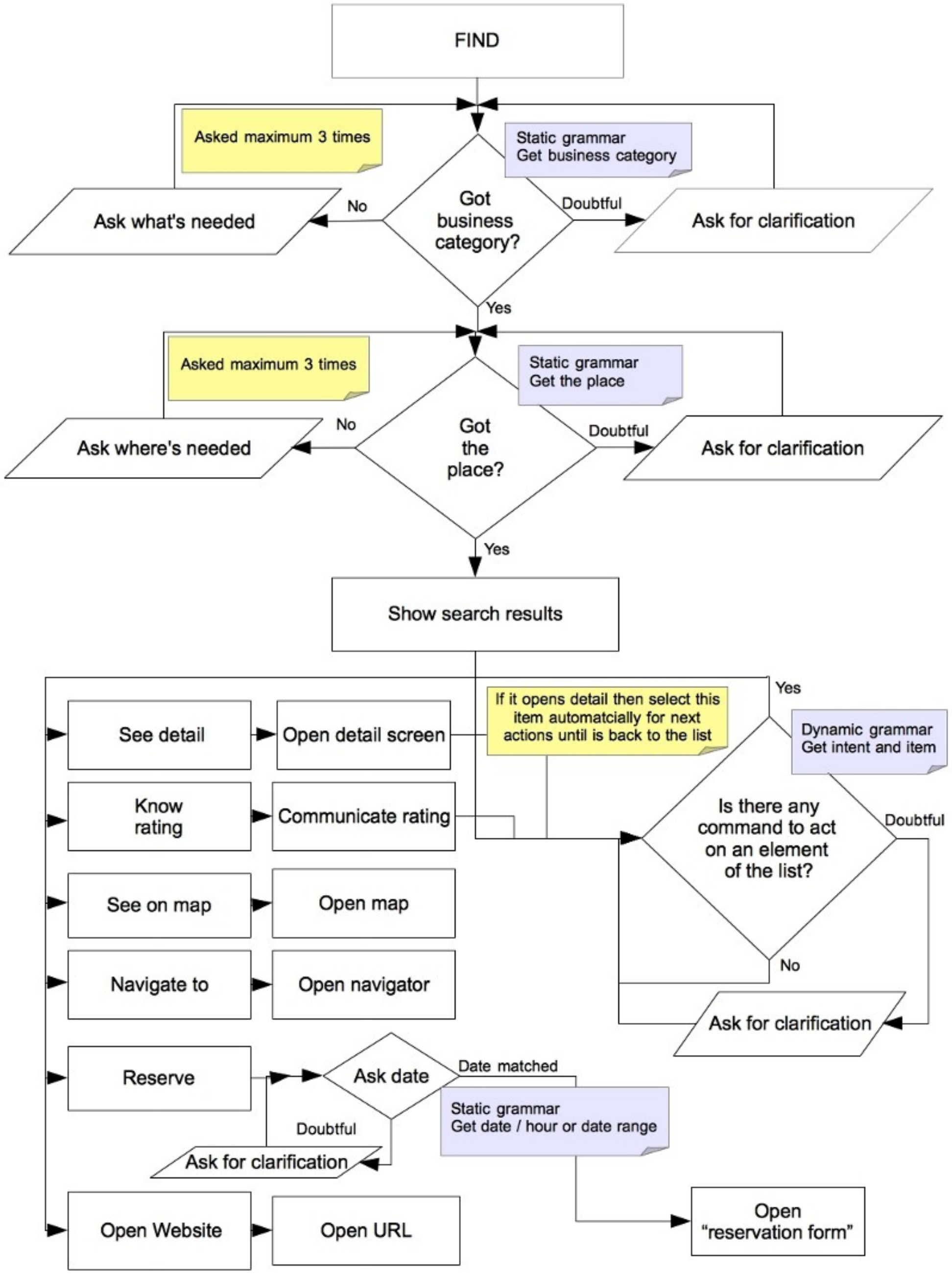
Ask for clarification

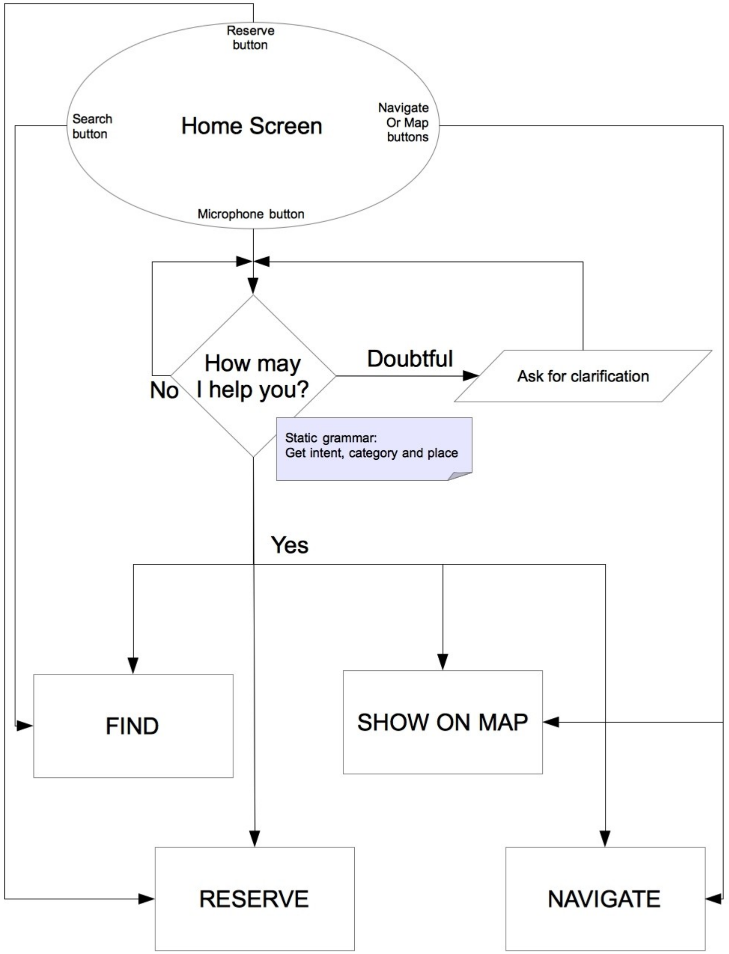
Yes

Show search results

Yes

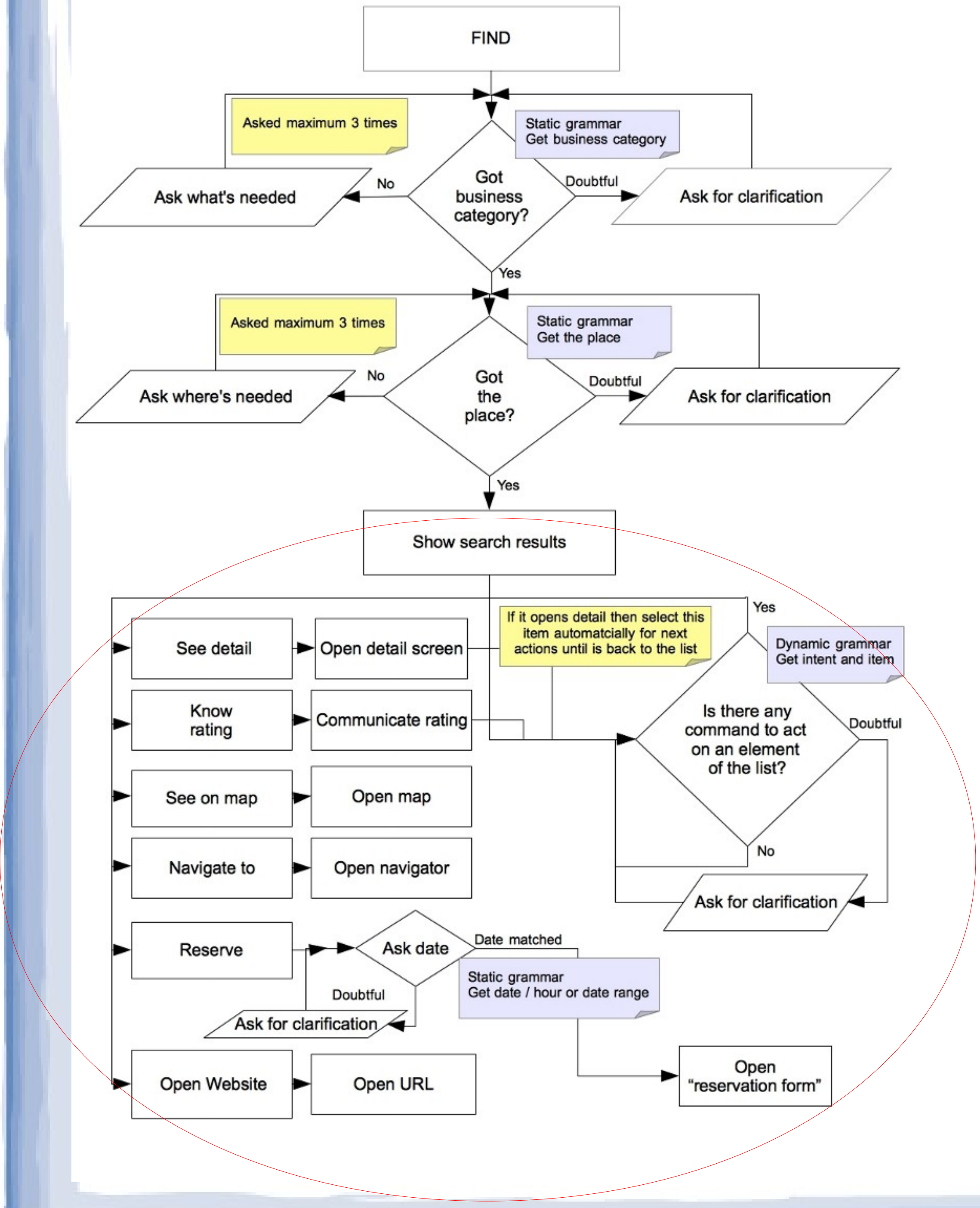
Dialogue flow example

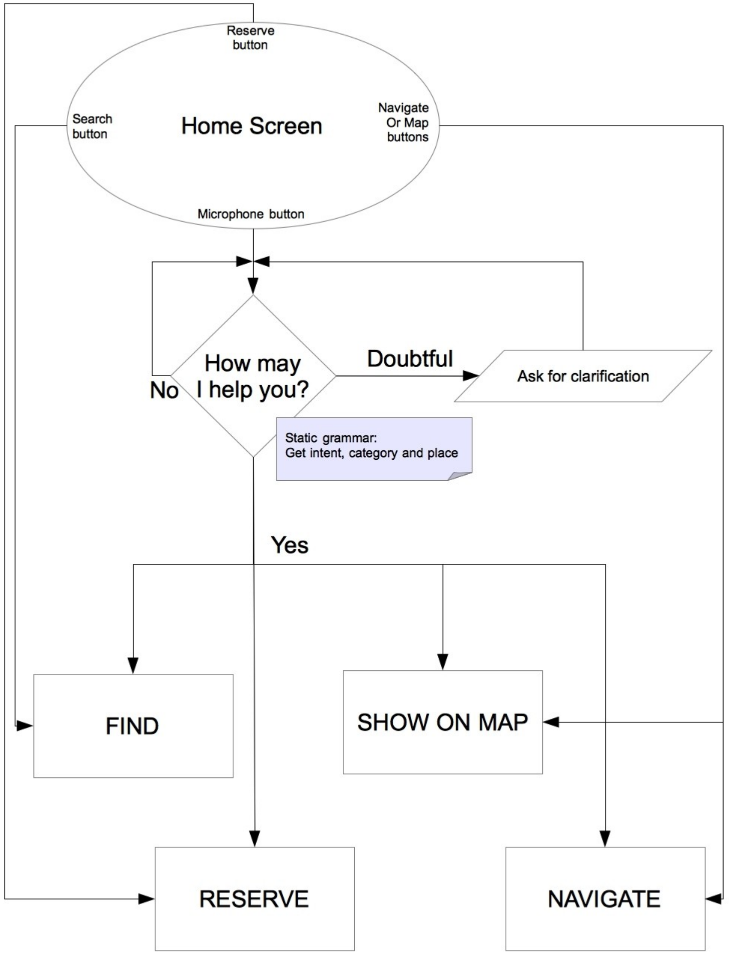




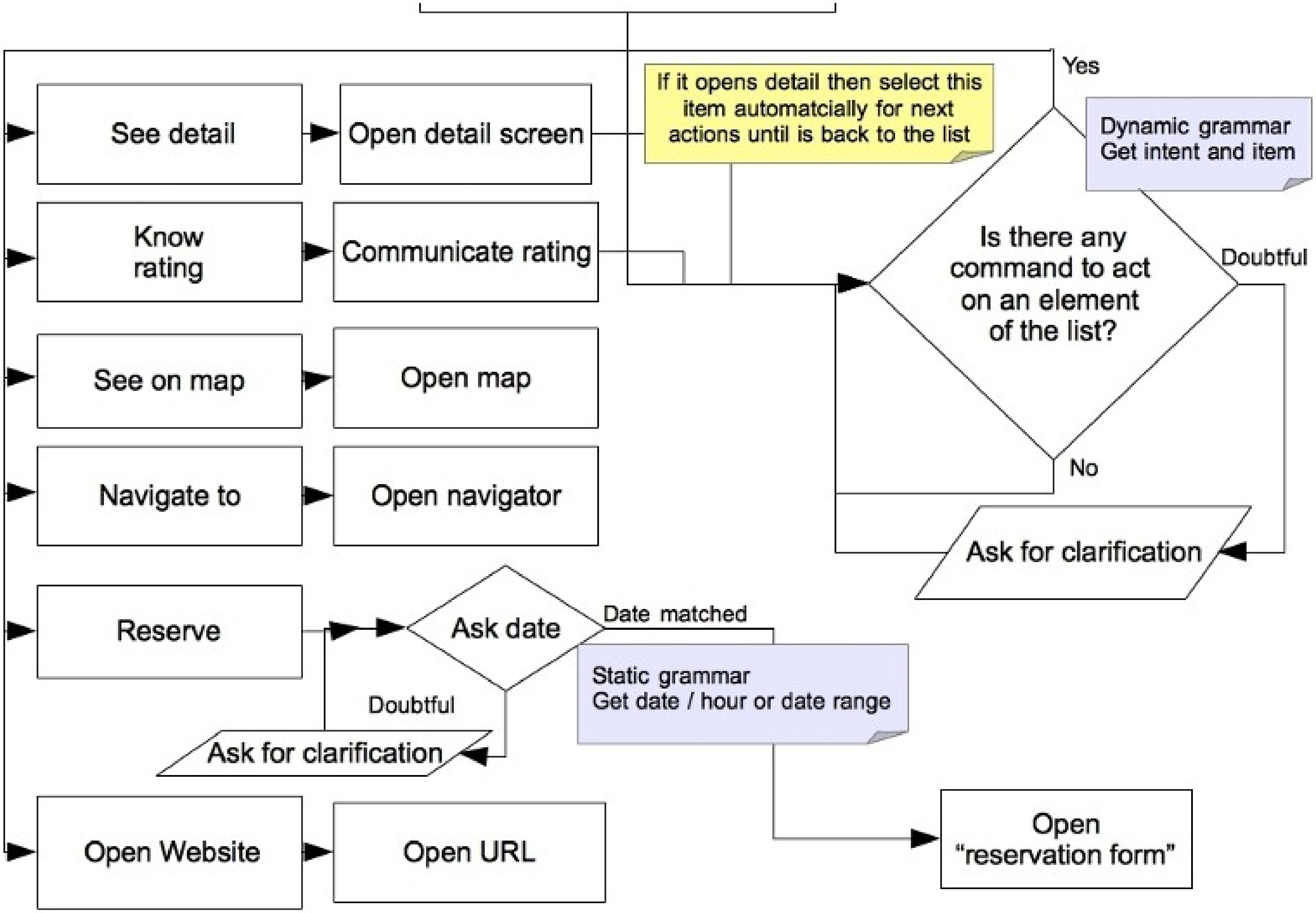
Signals & Interactive Systems

Dialogue flow example





Dialogue flow example



LPSM

Signals & Interactive Systems

“Hello Lab” tutorial – Web Portal

- Subscribe to the AT&T Web Portal
- Create the HelloLab Application
- Create grammar and upload it
- Compile the grammar
- Test the grammar through the ASR Test
- Edit the grammar until you get the results you need, recompiling it everytime and testing the ASR after each modification
- Start developing your application on the mobile device

“Hello Lab” tutorial for Android - Eclipse

- Create a new Android project
- Donwload HTTPComponents from http://hc.apache.org/downloads.cgi , import the .jar files inside the prject in a folder named libs. Configure the build path to get these files.
- Edit the manifest and add the uses permission for INTERNET, WRITE_EXTERNAL_STORAGE e RECORD_AUDIO
- Edit the main.xml with the visualization you prefer (like a textView, a button and another textView)
- Start modifiyng the main Activity: add inside the object the properties you'll need in all the project to be sure they are going to be seen in all your code.
- Set in these variables the URLs for the AT&T ASR and TTS putting in the arguments even your UUID
- Add a listener to the button. To have the hold effect use the OnTouchListener and check if the event corresponds to MotionEvent.ACTION_DOWN or MotionEvent.ACTION_UP
- On ACTION_DOWN create the HelloLab directory in the storage and start recording the voice

“Hello Lab” tutorial for Android - Eclipse

- On ACTION_UP stop recording and save the result in a temporary file
- Send the file to the AT&T WATSON ASR and get the response in a String variable
- Parse the variable to get what you need from the results and set the text of your TextView, or add an element in a ListView
- If you want to say something send the request to the AT&T TTS, save the returning file and play it
- Now you have the basis! Enrich the code and application as you prefer...