

SAYA SPEECH SYNTHESIS
MINI PROJECT

Elad Aviv
Gabriel Satanovsky

General Information on Saya

"(Saya is an) ... interactive communication system that communicates with human beings emotionally. Since the face and its expressions are the most important role for natural communication, we have been developing a face robot that can express facial expressions similar to human beings."

**-(Development of the Face Robot SAYA for Rich Facial Expressions,
Hashimoto, T.; Hitramatsu, S.; Tsuji, T.; Kobayashi, H.)**

Saya is the departmental robot receptionist located in the lobby of the new Alon building(37).

The mechanical control of Saya is realized via 22 face/neck McKibben-type

pneumatic actuators (each receiving a value translated to 0–10V level, via USB port),

and additionally via neck and eyes DC motors (simultaneously controlled by a

microprocessor receiving simple commands via RS-232 port).

We provide a Java wrapper API to these controls, also restricting the parameters to safe values (of course, Saya has physical protection from excessive air pressure as well).

Additionally, Saya has a video camera with regular USB interface in her left eye,

and similarly transparent connections to microphone and speakers.

The software which arrived with Saya consists of device drivers, controller firmware, and two components operating in closed loop. The first component interfaces with Microsoft Speech SDK in order to synthesize speech and produce facial expressions in response to a precompiled dictionary.

The second component uses DirectShow to extract camera input and apply

simple color filtering to locate the desired objects.
It then controls DC motors to focus on these objects.

The component that interfaced with the Microsoft speech SDK, was replaced by a java closed loop, that uses the java interface to the usb port written by Michael orlov, and a recognition module that interfaces with the Microsoft speech SDK using the cloud garden. Recognition module also was written by Michael Orlov. The new java closed loop uses also our synthesis module, through the *speak* method, that uses the Cloud garden to interface with the Microsoft speech SDK also. The java loop also uses the ability of our module to interact with the pneumatic actuators, using the face expression tags.

(The above text relies mainly on a text by [Michael Orlov](#))

Introduction

Our part in the Saya project is lips synchronization, and facial expressions integration.

The existing state before we started the work, was that Saya performed random lips movement with no relation to the speech generated.

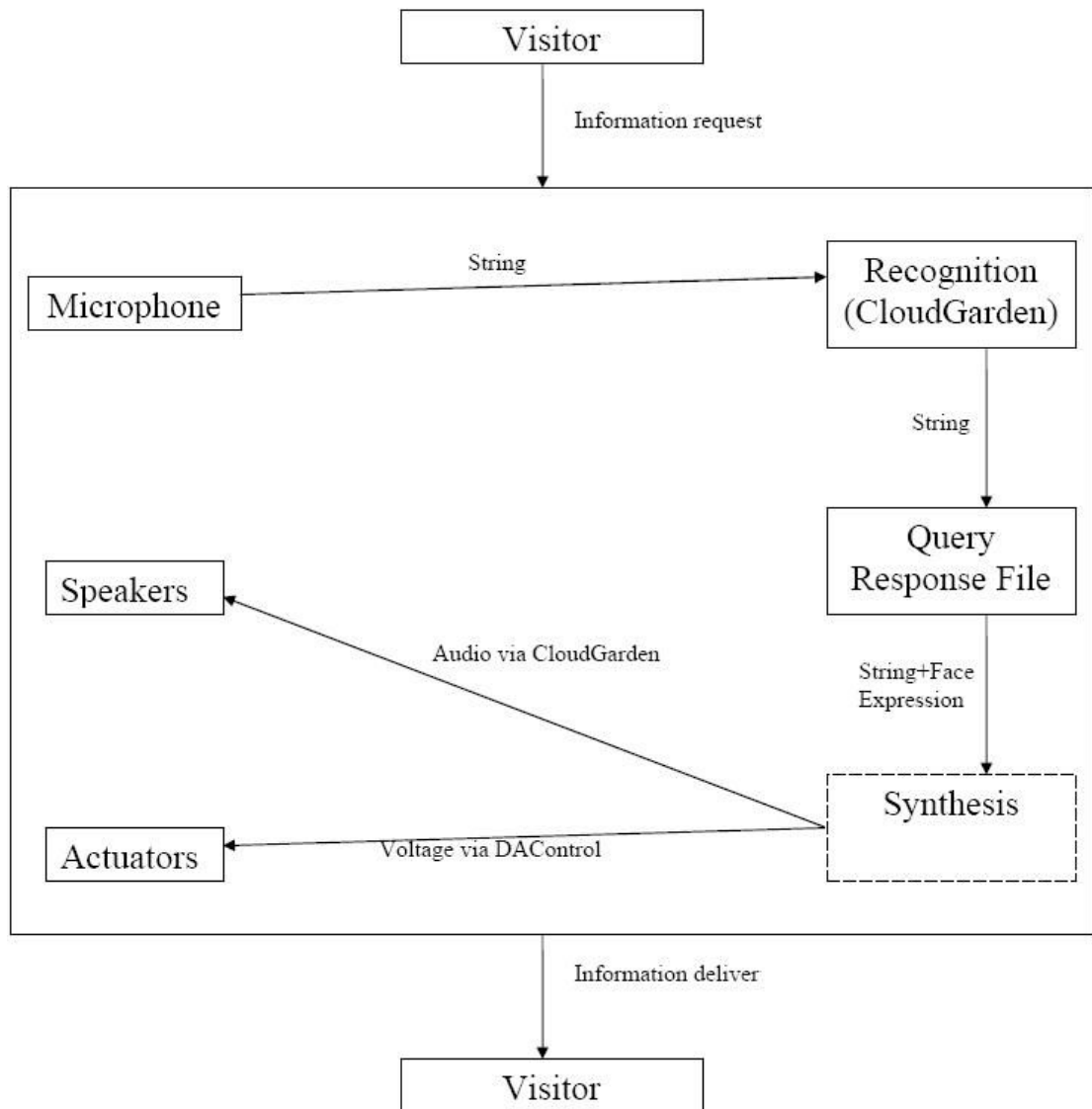
Main Components

The mini-project is composed of two main components:

- *Synthesis implementation* - The actual invocation of sound, and face expressions tag parsing.
- *Lips synchronization* - correlating between speech and lips movement.

Each of these parts will be described further on.

Project Overview



General description:

The objective in this part was to provide a simple interface to Saya's recognize-respond loop, to create an abstraction over the muscle control.



```
public interface Synthesis {  
    public void speak(String say);  
    public void done();  
}  
  
public static SynthesisImpl getInstance(){  
    return new SynthesisImpl("configexpressions.saya");  
}
```



The main loop acquires an instance of the Synthesis class, Then upon deciding the response calls speak.

The input of speak is a tagged text, when the tags specify in which facial expressions should the text between the tags be said.

For example:

- "Hello, how can I help you?<\happy>"
- "good morning<\happy>. have a nice day<\wink-right>"

Main Actors

CloudGarden

"CloudGarden is a full implementation of Sun's Java Speech API for Windows platforms, allowing a large range of SAPI4 and SAPI5 compliant Text-To-Speech and Speech-Recognition engines (in many different languages) to be programmed using the standard Java Speech API."

(Taken from official website-<http://www.cloudgarden.com/JSAPI/>)

DAControl

A JAVA API to the actuator's driver, written by Michael Orlov. Primary method being `pull(int act,int volt)` that uses the driver to pull actuator *act* with *volt* voltage.

Face Expression

A simple struct to hold the data relevant for a face expression. Mainly, the voltage for each of the 22 actuators in order to reach this expression.

Synthesis Implementation

Implements the Synthesis Interface.

Upon creation, opens the file given in the constructor, and constructs a map of , representing all known face expressions.

Then, when the **speak** function is called, the input text is analyzed for tags. If such were found, a **setFaceExpression** (a function that receives a FaceExpression object, and gradually, using DAControl, sets the voltage on the relevant actuators) command is issued.

After that a listener is assigned to the speech SDK. This listener

receives viseme events, which represent a facial shift. At this point mouth height information is extracted from the event, and is being set accordingly on Saya, again via DAControl.

Programming Issues

Inter Speech Expression

The problematic issue at this section was to display a face expression while speaking (which should initiate its own facial movement).

To solve this, the following method was introduced. First set the face expression, then speak as usual. This display fairly natural on Saya.

Mouth Size

While testing the face expressions (taken from Dr. Kobayashi's files), we noticed that the jaw opens less than regular, while speaking, also noticed is that it differs between face expressions. So we integrated a new parameter for each face expression: a mouth size addition it needs in order move lips properly.

Future Development

In the future this project should be developed on the following fields:

1) Speech recognition:

Currently the Microsoft Speech SDK is used both for speech synthesis and speech recognition.

The speech recognition engine is currently not very satisfying, Spoken text is not

recognized correctly causing the robot to output wrong answers. In the future new ,and more

suitable to the noisy lobby of the Alon building, speech recognition engines should be used.

another way to cope with the noisy environment is noise filtering, the input sound is analyzed and

cleaned from background noise. another option is using a dirctional microphone, forcing the speaking person

to stand on a certein spot or allowing visitors to move the microphone as they walk around the robot.

2) Developments in the fields of NLP and NLG:

At the moment Saya recieves an input from the speech recognition engine, look for it in a dictionary

and chooses a pre-defined answer. Saya can't react to free speech and doesn't react that way either,

causing her to sound very robotic. In the future, Saya's text analysis should use smart NLP systems,

allowing her to respond to the same question, asked in different ways, the same. This will give visitors

the feeling they can look at Saya as more as a human than a robot.

Another addition that may help guests think of Saya as more then a robot is using an NLG response System,

allowing her to react in different strings, with the same meaning. When Saya has to respond to a certein

question the program can respond in a few differnet ways, causing Saya to seem like an appropriate conversation

partner.

3) Emotion detection and Emotion mimic:

Saya has the ability to make face expressions, These are used to express feelings.

Currently, when Saya has to express a feeling such as happy or angry, she is told to do so

using the face expression tags added in our part of the project. The emotion are chosen

by the programmer that sets the dictionary and the response file, and not by Saya.

In the future a system that uses language processing to detect emotions in a given text

and react with the appropriate face expression and maybe tone, could be integrated. An additional speech analyzing

system may be developed in order to detect feelings, by analyzing the tone of the speaker

and other parameters.

4) **Life-like behavior and gestures:**

Saya has the ability of neck and facial movement, besides jaw movement. At the moment, when speaking, Saya

only moves her jaw, synchronized with the spoken text. To give a more life-like look, in future

versions of the program, programmers should use the ability of neck and head movement, corresponding to the

synthesized text, to create a more life - like imitation .

Documentation:

Available at <http://www.cs.bgu.ac.il/~eladav/report/doc/index-all.html>

About

This project was made for the course no. [202-1-4901](#) in the Ben Gurion University of the Negev.

Under the supervision of Prof. Shlomi Dolev and Michael Orlov.

presented by:

Elad Aviv - eladav@cs.bgu.ac.il

Gabriel Satanovsky - gabriels@cs.bgu.ac.il.

Reference

- **Development of Face Robot for Emotional Communication between Human and Robot** - a paper by Hashimoto, Hiramatsu and Kobayashi .

(http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4026050)

- **CloudGarden** - implementation of Sun's Java Speech API for Windows platforms.

(<http://www.cloudgarden.com>)

- **Michael Orlov's site** - information on Saya's software and hardware.

(<http://www.cs.bgu.ac.il/~orlovm/teaching/saya>)

- **SAPI** - Speech Application Programming Interface.

(<http://www.microsoft.com/msj/archive/s233.aspx>)