

SEMAINE API Tutorial

eNTERFACE 2010, Amsterdam, Marc Schröder

1. Installing the SEMAINE system

0. If needed, install a java SDK 6.0 or newer
1. Unpack SEMAINE-2.0-java.zip and either SEMAINE-2.0-windows.zip (on windows) or SEMAINE-2.0.1-source.zip (on Linux and Mac)
2. For linux and mac, you need to build from source.
 - (a) Make sure the necessary build tools are installed: gcc/g++, autoconf, make. For example, on Ubuntu Linux, the following should get you close:

```
sudo apt-get install automake autoconf gcc g++ libasound2-dev make libtool patch libx11-dev
```

On Mac OS X, use e.g. MacPorts to install autoconf.

- (b) Then, start the build process:

```
bin/buildall.sh
```

Compiling will take an hour or so.

- (c) Install activemq by unpacking the tar.gz package.
3. For Windows we ship binaries. These are tested with Windows Vista.
 - (a) Simply unpack the zip file.
 - (b) If you have a Camera that is supported by OpenCV, you may also want to install the SemaineVideoComponent.msi file (check Firewire driver requirements).
4. For the purpose of this tutorial, also unpack the file SEMAINE-tutorial.zip.

2. Starting the SEMAINE system

1. On Linux and Mac:
 - (a) First start activemq. Wait a few seconds until it is running.
 - (b) Now start all required parts, native and java. For example for the Speech-only version of the SEMAINE system, run each of the following commands in a separate shell:

```
SEMAINE-2.0.1/bin/run_components/start_component_tum.opensmile
```

```
SEMAINE-tutorial/java/semaine-speech2speech.sh
```

2. On Windows, the process of starting the SEMAINE system is fully automated. Just click start.bat. (click stop.bat to stop the system).
 - (a) The simplest way to start the system in different configurations may be to modify a copy of start.bat.

3. **Compiling and running the examples of minimal emotion-oriented systems**

The SEMAINE-tutorial folder contains the sources, build files and config files needed to run the examples shown in the presentation.

0. Make sure the SEMAINE-tutorial folder is located in the same parent folder as the SEMAINE-2.0 folder.
1. To compile the java sources, run the java build script:

```
cd SEMAINE-tutorial/java
ant
```

2. To run the java examples:

```
cd SEMAINE-tutorial/java
./semaine.sh config/example-XYZ.config
```

- (a) example-hello.config: try to understand how config file and java sources play together exactly.
- (b) example-mirror.config: In addition to this, you must start opensmile as described in Section 2 above.
Note: There are too many smiley outputs – why?
Modify EmoticonOutput.java such that there are not more emoticons than necessary.
- (c) example-swimmer.config: Again, start opensmile in parallel.
Can you configure a version of the swimmer's game without TTS output?
What could you do to make the game easier to win?

3. To compile the native examples:

```
cd SEMAINE-tutorial/c++
./autogen.sh
./autogen.sh (yes, twice)
./configure --with-semaineapi=../../SEMAINE-2.0.1/c++/
make
```

4. To run the native examples:

```
cd SEMAINE-tutorial/c++
./hello
```

and in a separate shell:

```
cd SEMAINE-tutorial/java
./semaine.sh config/message-logger-only.config
```

4. Write your own Java or native SEMAINE components

Use the example code as starting point to experiment.

Build a version of the hello example system that mixes native and Java components.

5. Set up a distributed system

Run ActiveMQ on one machine with a known IP address, say 123.45.67.89. On all machines running any SEMAINE components of the system, define the environment variable CMS_URL to point to the ActiveMQ server as follows:

On windows:

```
set CMS_URL=tcp://123.45.67.89:61616
```

(look inside start.bat if you use that, it contains such a setting)

On Linux/Mac (bash):

```
export CMS_URL=tcp://123.45.67.89:61616
```

then start the java or native clients.