EvAAL 2012. Activity Recognition track (Valencia, Spain)

Objective

This document describes the protocol used both for competitors and evaluation committee (E.C.) during the competition.

Protocol

1. Welcome and preparation:

When the competitor arrives the EC welcome warmly and explain the protocol

One person will be the responsible of checking this document twice so that nothing is missed

2. Installation and configuration/calibration:

The team have **60 minutes (it corresponds to X value in the Technical Annex)** to install and calibrate the system**. Calibration include any training that actor must do with the system.**

When installation starts, no one a part from the competitor's team must be in the place.

One measures the time for the installation time.

Installation ends when the competitor tells it. With the calculus explained in Technical Annex we can obtain the result for Installation complexity evaluation criteria.

Two people measure (metadata) the devices and 1 person makes some pictures (if competitor agrees).

3. Evaluation system set up:

Competitor is invited to launch his system and to check if the server receives correctly the information sent:

- **The IP address used will be 192.168.233.12 and port 7777 (please prepare your software to easily change the IP, port or/and NTP server in case there is any problem)**.

One checks that competitor's clock and the EvAAL system clocks are really synchronized (**competitor should indicate which ntp server is using, we are using pool.ntp.org**)

One checks the integration of the competitor's system with EvAAL system

If some problem occurs a maximum of **20 minutes (it corresponds to Y value in the Technical Annex)** is given to fix integration, otherwise the log backup solution will be used. **Log backup correspond to a txt file created by the competitor system with the same information required through sockets** (milliseconds from 1970, integer of activity) i.e.:

1340107198026,6
1340107208500,2
1340107214386,0
1340107233972,3

…

Any problem related to the integration is logged on a note (for feedback to universAAL)

4. Actor performance:

Only the actor and one EC member (responsible to annotate ground truth activities) will be inside the living lab during the competition, the competitor as well as the evaluation committee will be outside.

One is responsible for recording all the video (if the competitor agrees)

The evaluator starts the mp3 file that will guide the actor.

The actor moves and performs activities using the mp3 file. The evaluator uses a client to mark activities and transitions.

Once the mp3 finish, the actor will continue during 1 minute performing the last activity in order to capture the delayed events recognized by the competitor system. If any competitor has a delay greater than 1 minute he must indicate us before the performance begin.

After that time the evaluation software will run in order to obtain the performance criteria and the delay criteria. The software will select the recognition delay that maximizes the performance criteria.

The performance will be repeated again to obtain a second mark in performance and delay criteria and we will select the best performance.

5. Interview:

Competitor will be interviewed about Interoperability and User acceptance criteria. To avoid subjective values in user acceptance and interoperability we will use the yes/no questions in annex I and II. However E.C. can ask any other questions.

Interview will be annotated in a scoring sheet.

If any vote must be taken by the EC it is taken AFTER the competitor has left the room

6. Finalization:

The competitor is invited to answer a feedback questionnaire

The competitor is asked for the intermediate data produced during the competition, one person from the EC is responsible for collecting the data

Competitor is asked to provide the logs of the EvAAL system

All the documents are stapled together, ALL files including photos, logs, etc.

The competitor is asked to remove installation

The competitor is thanked warmly and given a small present.

Annex I: User acceptance

Usability

Does your system need the user to bring some on-body sensors/tag with him all the time?

   Can it be moved without inconvenience?
   Is it easy to wear/put and take them off
   Is it easy to keep during normal domestic activities?
   Is it easy to keep during night sleeping?
   Is it washable?

Can you use your system during a shower?
Does your system use video cameras?
Is this device/tag completely hidden?

Environment

Is your installation well hidden in the house?

Does your installation need considerably more cabling than a typical Home PC installation (including for example printer, scanner, phone connector etc.)?

Is your system useful if there are more than one user in the house?

If you change or include furniture, must your system be recalibrated?

Maintenance

Does the user need to replace/charge batteries?
Does the user need to re-calibrate the system?
   Does the user need to re-calibrate the system in less than one month?
Is it easy to notice if it is broken or malfunctioning?
Does one have to configure it?
Which is the recharge rate of your ARS (hours without recharge of any of your components)
Recharge time (in hours)

Other features

Allows your system to notify the activities to a remote server in real time?
Is your system scalable for multiuser?

Annex II: Interoperability

Interoperability

Do you provide any API for integrating your system into others?

Do you provide any written documentation for this API?

  Any code documentation (like javadoc) ?

  Any tutorial?

  Any sample application ?

Do you publish your code as open source?

Do you use any well known application-level protocol which would allow an external system to plug into the system without requiring any further specification about the protocol? (e.g. SOAP, XML-RPC, DPWS, BUT NOT TCP/IP, Ethernet)

Testing and configuration

Do you provide any tool for testing/monitoring the system?

  Is the tool graphical, easy to be used?

Do you provide any tool for configuring/calibrating your system?

  Is the tool graphical, easy to be used?

Portability

On which operating systems does your system run on?

  Windows?

  Unix/linux?

  MacOS?

  Android?

  iOS?

  Others?

Do you know of any incompatibility of your SW with others?

Can part of your code be substituted by another existing one (that you know of)? [e.g. toher existing libraries, modules that can be easily plugged to substitute yours]

HW

Can any part of your system be substituted by another commercially available one WITHOUT ANY modification on the HW and/or the firmware?

Is your firmware publicly available?

Do you publish your firmware as opensource?