Experiment Wizard Tutorial

The following is an example of an imaginary experiment and it's configuration for ExpWiz. This particular experiment is designed to work on computers running windows.

Requirements:

Experiment Wizard - http://expwiz.sourceforge.net/

Perl – http://perl.org/

Audacity – http://aduacity.sourceforge.net/

AutoHotKey – http://autohotkey.com/

Experiment description

The experiment is designed to find the voiced reaction time of the subject to a visual stimulus. Subject must say a predefined word when the stimulus is presented. To analyze the data we need the time of the stimuli presentation and the timing of the response. Stimuli are presented by a computer program, which logs the time of the presentation. The subjects response is recorded using Audacity. The moment recording was started is also logged so we can find the reaction time. Two different timings for the stimuli presentation make sure that we do not bias our results. These timings must also be counterbalanced between the subjects.

Hardware: computer for a subject with a microphone, computer for the experimenter.

Software: program to show visual stimuli (Show.ahk), audio recording software (Audacity).

Procedure

A subject comes in, signs informed consent form, receives description of the experiment, goes through training, performs two parts of the experiment, goes through debriefing, receives reword, and leaves.

Configuration

The experiment description is as follows:

```
<Name>Experiment Example</Name>
<Author>Your Name Here</Author>
<Email>Your@email.here</Email>
<Description>Experiment example for ExpWiz tutorial</Description>
<AppName>perl ..\..\Shared\NetCli.pl</AppName>
```

We decided that 4 experiments should be enough for our purposes. Counterbalancing gives us two stage combinations, which we call Path1 and Path2. Path2 will be the reverse of Path1.

```
<ExperimentCodes>
        <Code Name="Tutorial1" Configuration="Path1" />
        <Code Name="Tutorial2" Configuration="Path2" />
        <Code Name="Tutorial3" Configuration="Path1" />
        <Code Name="Tutorial4" Configuration="Path2" />
</ExperimentCodes>
```

start is the build in command on windows, so we specify it as such. This allows to run the verification of the files required by the experiment.

```
<AssumedToBePresent>
	<Command>start</Command>
</AssumedToBePresent>
```

For simplicity this tutorial runs the experiment on the same computer where ExpWiz is installed. By changing IP address here it's possible to make it run on any other computer that is accessible through the network and has the proper files installed.

```
<IPResolutions>
	<Resolve Name="SomeComputer" IP="localhost" />
</IPResolutions>
```

Stages:

1. Power up – power the hardware
2. Subject paperwork – get required signatures, show required documents
3. Training – just like an experiment but without recording
4. Experiment – record the subject
5. Debriefing – explain subject the purpose of the experiment
6. Power down – shutdown the hardware

Stage 1: Power up

This stage shows prompts for the experimenter that can be used a check list. We also want to make sure that hardware is operating properly. We can look at the monitor to see if it works, but for the microphone we must run a program to see if the recording is possible. Doc/Tutorial directory contains the scripts that we need to make the test operational.

Note: It's possible to use Auto Script Writer[1] to create scripts. It's very convenient for mouse click recordings.

The following is the setup for this stage:

```
<Stage Name="Power up">
       <Execute>Switch on subject's computer</Execute>
       <Execute>Check if monitor is on</Execute>
       <Execute
              IP="SomeComputer"
              RemoteCommand="StartAudacity.ahk">
              <Description>Start audacity</Description>
       </Execute>
       <Execute
              IP="SomeComputer"
              RemoteCommand="StartRecording.ahk  c:\\no_data_saving.tmp" />
       <Execute MinimumTime="2">Check if microphone is connected</Execute>
       <Execute
              IP="SomeComputer"
              RemoteCommand="StopRecording.ahk">
              <Description>Stop audio recording</Description>
       </Execute>
       <Execute
              IP="SomeComputer"
              RemoteCommand="CloseAudacityWithoutSaving.ahk">
              <Description>Close audacity without saving</Description>
       </Execute>
</Stage>
```

[1] Supplied with AutoHotKey program http://autohotkey.com

Stage 2: Subject paperwork

For simplicity we use paper documents, but we also could run PDF reader on the subject computer to present relevant information.

```
<Stage Name="Subject paperwork">
       <Execute>Present informed consent form</Execute>
       <Execute>Get signature</Execute>
       <Execute>Present task description document.
User should say On when the star is shown in the top left corner of the screen.</Execute>
</Stage>
```

Stage 3: Training

It's time to start the application that shows visual stimuli to the subject, but no need in recording yet.

```
<Stage Name="Training">
       <Description>Traing for the subject. It will last for 10 seconds</Description>
       <Execute
              IP="SomeComputer"
              RemoteCommand="start Show.ahk c:\\no_data_saving.tmp 3" />
       <Execute MinimumTime="10" ProceedToNext="true" >Wait for 10 seconds for the user to
respond.</Execute>
       <Execute
              IP="SomeComputer"
              RemoteCommand="Show.ahk" >
              <Description>Stop previous blinking</Description>
       </Execute>
       <Execute>Check if subject has any questions</Execute>
</Stage>
```

Stage 4: Experiment

It's time to start the experiment which should have two parts with different timing for stimuli presentation. Given that the only difference between the parts is timing we can setup the stage template that we can use. This stage must start audacity, start stimuli presentation, start recording, wait for the duration of the experiment, stop recording, save data, and stop stimuli presentation. Notice that %PAUSE% denotes the variable that will have the value of the pause we would like the stage to have. %PART% contains the part of the experiment, it can be part 1 or part 2. We use this variable to differentiate the recorded wav files.

```
<StageTemplate Name="Blink">
       <Description>Start blinking with a %PAUSE% second pause.</Description>
       <Execute
              IP="SomeComputer"
              RemoteCommand="StartAudacity.ahk" >
              <Description>Make sure audacity is started</Description>
       </Execute>
       <Execute
              IP="SomeComputer"
              RemoteCommand="start Show.ahk c:\\%EXPCODE%.log %PAUSE%" >
              <Description>Start blinking with the given puase</Description>
       </Execute>
       <Execute
              IP="SomeComputer"
              RemoteCommand="StartRecording.ahk c:\\%EXPCODE%.log" >
              <Description>Start recording in audacity</Description>
       </Execute>
       <Execute MinimumTime="10" ProceedToNext="true" >Wait for 10 seconds for the user to
respond.</Execute>
       <Execute
              IP="SomeComputer"
              RemoteCommand="StopRecording.ahk" >
              <Description>Stop recording</Description>
       </Execute>
       <Execute
              IP="SomeComputer"
              RemoteCommand="SaveDataAndCloseAudacity.ahk c:\\%EXPCODE%%PART%.wav" >
              <Description>Saved data and close audacity.</Description>
       </Execute>
       <Execute
              IP="SomeComputer"
              RemoteCommand="Show.ahk" >
              <Description>Stop previous blinking</Description>
       </Execute>
</StageTemplate>
```

We declare stages with different timings and we also counterbalance them. Replace tags are used to change the value of the variables.

```
<Stage Template="Blink" Configuration="Path1">
       <Replace Text="%PAUSE%" Value="2"/>
       <Replace Text="%PART%" Value="_Part1"/>
</Stage>
<Stage Template="Blink" Configuration="Path1">
       <Replace Text="%PAUSE%" Value="5"/>
       <Replace Text="%PART%" Value="_Part2"/>
</Stage>
<Stage Template="Blink" Configuration="Path2">
       <Replace Text="%PAUSE%" Value="5"/>
       <Replace Text="%PART%" Value="_Part1"/>
</Stage>
<Stage Template="Blink" Configuration="Path2">
       <Replace Text="%PAUSE%" Value="2"/>
       <Replace Text="%PART%" Value="_Part2"/>
</Stage>
```

Stage 5: Debriefing

Once we collected enough data we can save it, close applications and debrief the subject.

```
<Stage Name="Debriefing">
       <Description>End of the experiment</Description>
       <Execute>Debrief the subject</Execute>
</Stage>
```

Stage 5: Power down

Shutdown the equipment if required. Execution of this stage also saves the experiment code as performed. Next time the application is started the next experiment code will be chosen by default.

```
<Stage Name=”Power down”>
       <Execute>Shut down subject's computer</Execute>
       <Execute>Switch off the lights</Execute>
</Stage>
```

Running the experiment

Full configuration file for the tutorial is in doc/tutorial/Tutorial.xml. Every time before the subject comes in, we use StartTutorial.cmd located under doc/tutorial/ directory, to start the server for controlling the computer (in real life the server is always running on) and start ExpWiz GUI:

```
rem Make sure that the server is executed on this computer
start perl ..\..\Shared\NetSer.pl localhost
rem Start ExpWiz
start ..\..\bin\ExpWiz.exe --ExperimentConfig Tutorial.xml --LogFile c:\ExpWizTutorial.log
--Verify
```

Confirm subject's code and execute the first stage. Follow the prompts and execute stages when appropriate to go through the whole experiment. Once all subjects have done the experiment, the data can be analyzed using log and sound files stored on a subject computer. It's easy to add another command in the Debriefing stage to copy data files to the experimenter computer or run processing on it immediately.

Note: If you change the IP from localhost to some other IP, the same IP must be used for starting NetSer.pl.

```
perl NetSer.pl IP_OF_THIS_COMPUTER
```

The end.