



AsTeRICS Introduction

Demos, Model creation, Packaging (APE)

Webinar: Integration of Alternative Input Modalities into Educational Apps

[Martin Deinhofer](#), UAS Technikum Wien, Austria

12th Dec. 2016



This project has received funding from the EU's Seventh Framework Programme for research, technological development and demonstration under grant agreement no. 610510

- Currently 12 Bachelor and 17 Master Programs
- About 4.300 graduates and 3.000 students

- R & D main areas
 - Embedded Systems
 - Digital System Design
 - Embedded software design
 - Test and verification of hardware / software
 - **Assistive Technologies** , Smart Home
 - eHealth
 - Tissue Engineering
 - Renewable Energy



[UAS Technikum Wien, Vienna](#)

AsTeRICS

AsTeRICS ...

... is a flexible **Assistive Technology (AT)** prototyping tool, able to deal with

- **unique sets of capabilities** of single users
- **changing capabilities** over time
 - degenerative diseases
 - rehabilitation progress
 - fatigue (on a single day)
- **changing needs and wishes**



The AsTeRICS Project

Assistive Technology Rapid Integration and Construction Set

FP-7 funded EU-STREP-Project 2010-2013

AsTeRICS Project was implemented 2010-2013 by 9 European partner organisations and was partially funded by European Commission under the 7th framework programme

<http://www.asterics.eu/>

Extended by follow-up projects

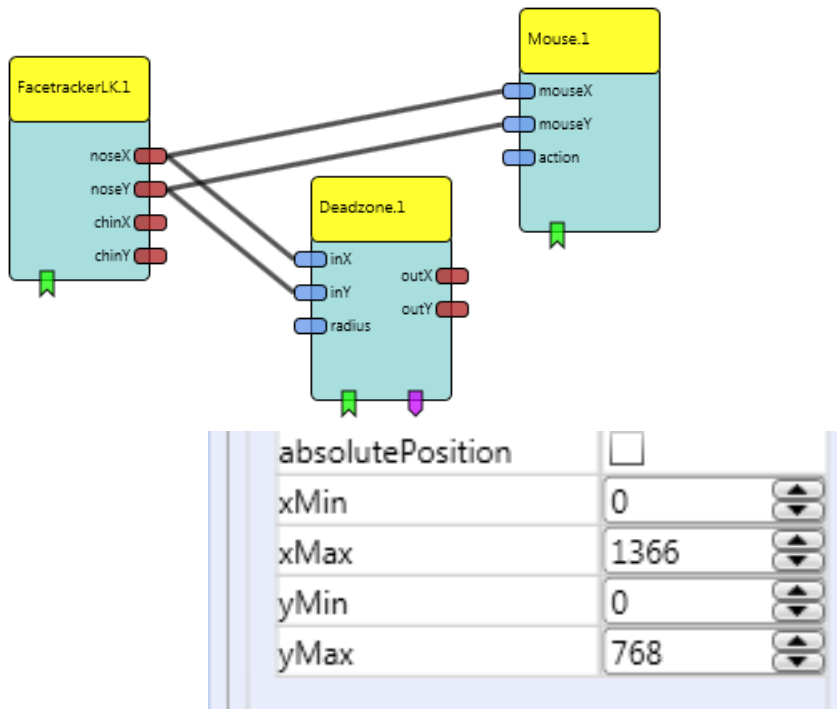
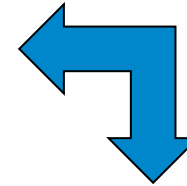
- [AsTeRICS Academy \(2013-2016\)](#)
- [Prosperity4All \(2014-2018\)](#)



AsTeRICS: Graphical design of AT-solutions

AsTeRICS Configuration Suite (ACS) (Windows)

- Connect sensor, processor and actuator modules with signal paths
- Define properties and event conditions



AsTeRICS Runtime Environment (ARE) (Win, Linux, Raspb. Pi, Mac)

- Run model configuration
- Interface sensors and actuators
- Process data

AsTeRICS plugin overview

AsTeRICS offers currently about 160 different plugins for input, processing and output. This makes it a versatile construction set for **rapid prototyping of AT solutions** with:

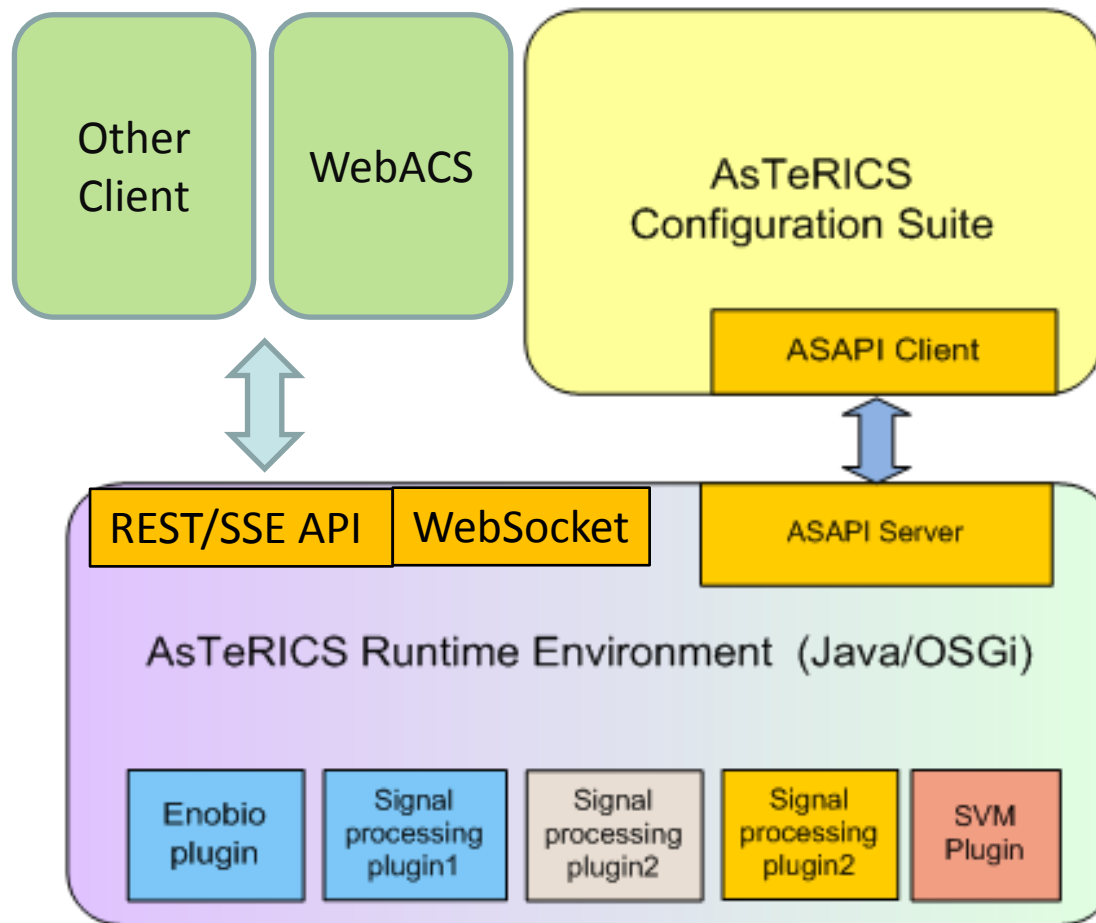
- Webcam-based face tracking
- Eyetracking
- Switches
- Scanning-based GUI widgets
- Keystroke and mouse emulation
- Capturing / Remapping of keyboard and mouse input
- Smart Home / Environmental control
- Arduino microcontroller support
- Accessible gaming
- Bioelectric interfaces: OpenEEG, OpenBCI, EMG shields



AsTeRICS technical facts

- **ACS**
 - C# (only Windows)
 - Uses ASAPI interface
- **WebACS (alpha, Developed within Prosperity4All)**
 - HTML/Javascript (all platforms)
 - Uses REST API (Developed within Prosperity4All)
- **AsTeRICS Runtime Environment (ARE)**
 - Java/OSGi middleware (all platforms)
 - Plugins
 - Java (all platforms)
 - Some use JNI / C++ native code for OpenCV, HW access etc. (Windows or all platforms)

AsTeRICS system architecture



AsTeRICS Demos

AsTeRICS hands-on demos



Camera Input
(Face tracking,
Eye tracking)



Biosignal
acquisition
and processing
(EEG, EMG, ...)



Smart Home
(KNX, enOcean,
FS20)

The demo release
contains demos about



Microcontrollers
(Arduino)



HID emulation
(Mouse, keyboard,
Joystick)



Voice control



Web integration
(Web socket)

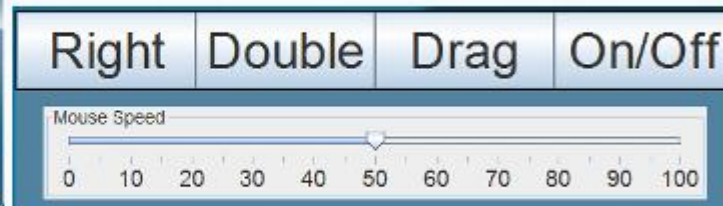
AsTeRICS hands-on demos

The [demo web page](#)

adds detailed descriptions
and shows REST API integration



Deploy and Start demo





AsTeRICS hands-on demos

Requirements

- [Demo installer](#)
- [Java Runtime Environment >= 7 \(32-bit\)](#)
- Optionally additional HW / SW

Requirements installed?

Try some demos hands-on on your computer!

[Go to the demo web page](#)

AsTeRICS Model Creation

Camera Mouse

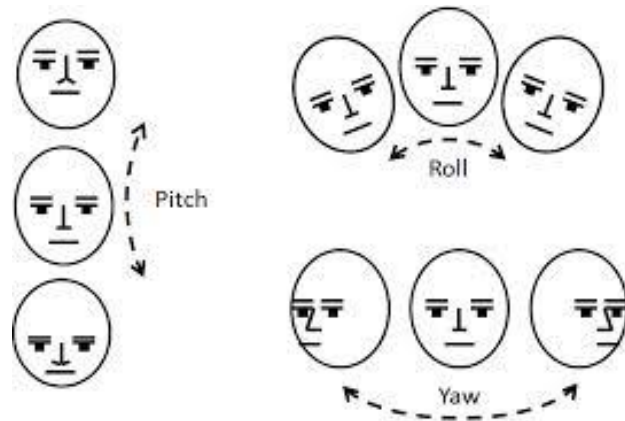
Mouse-Control by Head Movements

- In this example we will show how you can control the mouse cursor via head movements
- We use a webcam and the XFacetrackerLK sensor plugin (which tracks face movements)

Requirements

- Full [AsTeRICS v2.8 installer](#) (Windows)
- Webcam

Note: You can also try the [complete tutorial](#)

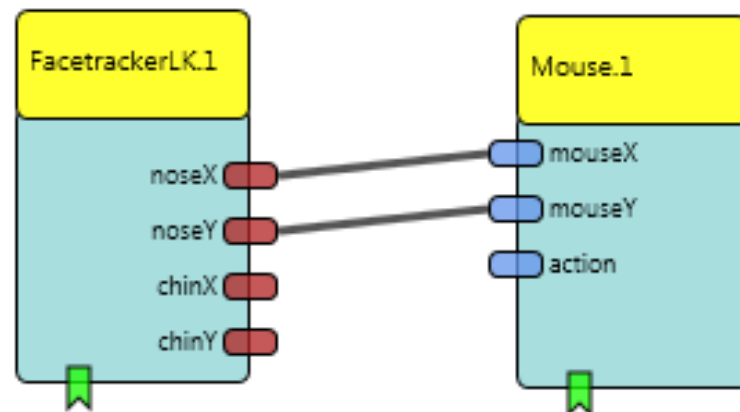


Mouse-Control by Head Movements

- Create a new model in the ACS (System → **New Model**)
- Insert the „**FacetrackerLK**“ or **XFacetrackerLK (all platforms)** sensor component
(Components → Sensors → ComputerVision → XFacetrackerLK)
- Insert the „**Mouse**“ actuator
(Components → Actuators → Input Device Emulation → Mouse)
- Connect ports „noseX“ to „mouseX“ and „noseY“ to „mouseY“

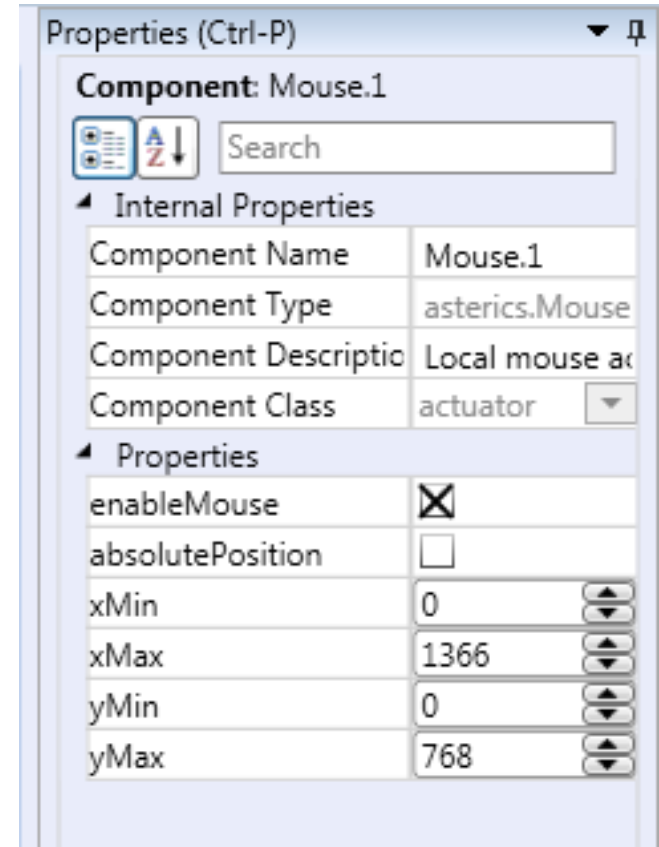
Note:

You can also search for plugins in the Components tab



Mouse-Control by Head Movements

- Mouse plugin properties:
 - **deselect** „absolutePosition“
this defines that the X and Y input values are relative changes
(The XFacetrackerLK plugin only tracks relative movements)



AsTeRICS Packaging Environment (APE)








What is AsTeRICS Packaging Environment (APE)?

Manage AsTeRICS AT solution as dedicated SW project

- Project structure template
- Set of tools and **build infrastructure** (ant)
- Creates a downstripped version of the ARE
 - (including plugins, configuration files and data files to execute the models.)
- Enables customization of ARE (Replace images, config files,...)

```
ant APE-copy
```

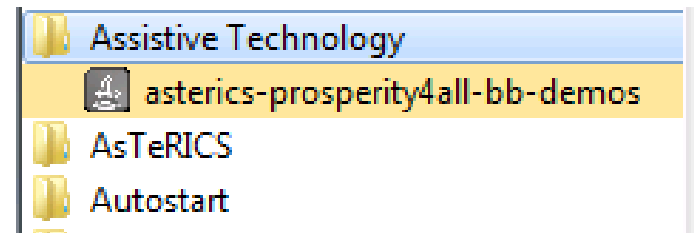
Name

-  custom
-  package
-  Tests
-  APE.properties
-  build.xml
-  imported.xml
-  README.md

```
APE-copy -DAPE.models=../ARE/models/ImageDemo.acs
```

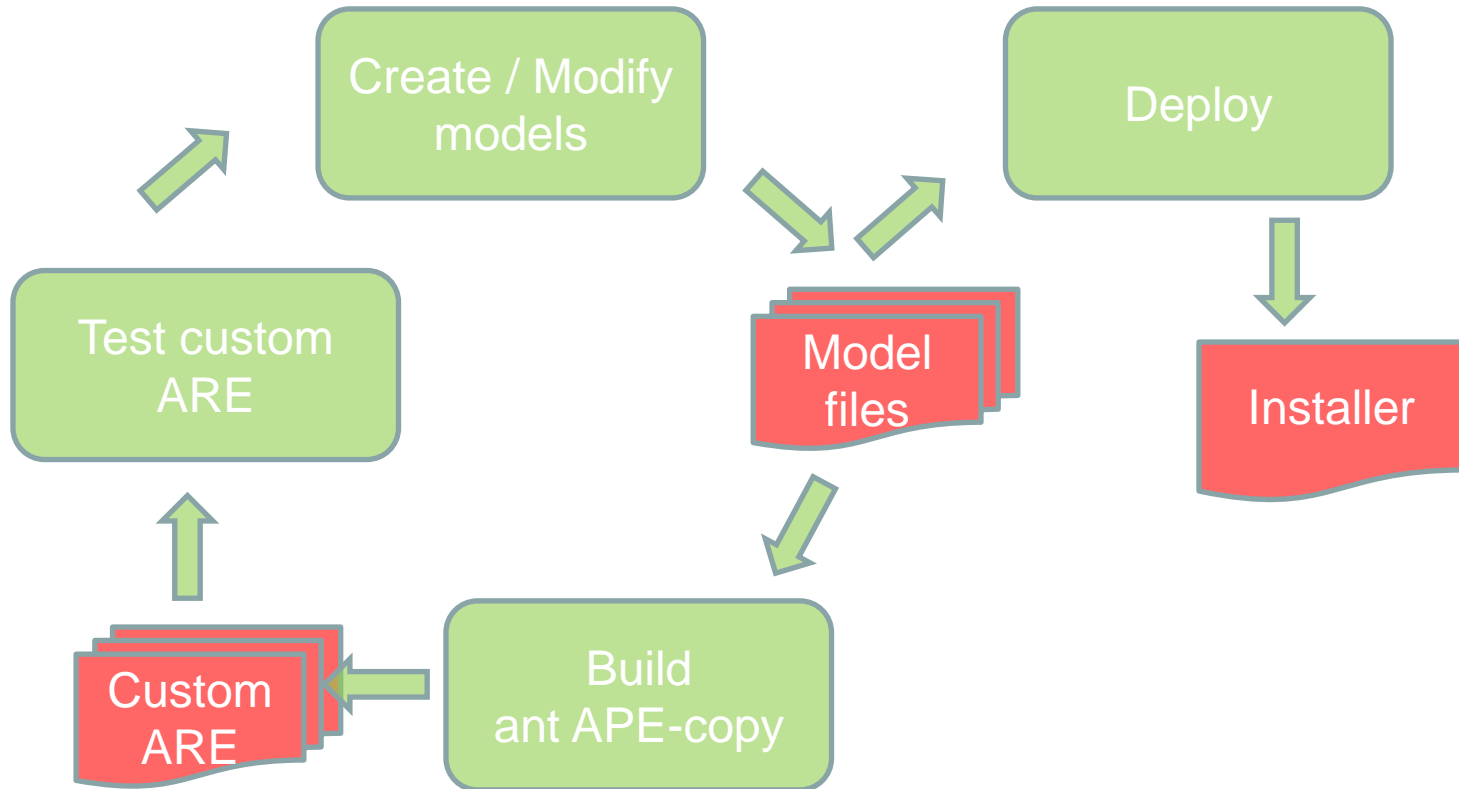
What is AsTeRICS Packaging Environment?

- Enables creation of **native installer** and **native launcher**
 - **Windows (.msi, .exe)**
 - **Linux incl. Raspberry Pi (.deb)**
 - **Mac OS X (.dmg)**
 - Desktop integration
 - With / without Java Runtime embedded
- Helps in license management
 - **GPLv3 / GPLv3 with CLASSPATH exception** or **MIT?**
 - Provides set of involved license files



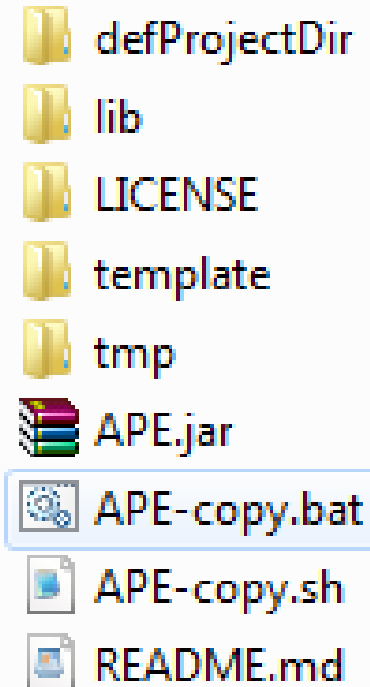
Implementers can **create, host, customize, release/deploy** AT SW based on the AsTeRICS framework **themselves**.

Intended workflow using APE



APE folder location

- Full AsTeRICS installer
 - C:\Program Files (x86)\AsTeRICS\APE
- Platform independent .zip
 - APE





APE-copy command line tool

A command line tool to create a down-stripped version of the ARE based on provided model files.

One model file

Create a downstripped ARE package of the model file `ImageDemo.acs` located in the `ARE/models` folder. By default, the result is written to the folder `<APE.buildDir>/merged`.

```
cd <APE.baseURI>  
APE-copy -DAPE.models=../ARE/models/ImageDemo.acs
```

Template project folder

ant build files

Project properties (APE.properties)

One model file, ant APE-copy

Copy the model file `<ARE.baseURI>/models/ImageDemo.acs` to the location `<APE.projectDir>/custom/bin/ARE/models` or edit the `APE.models` property in the file `<APE.projectDir>/APE.properties`. Then execute the following commands:

```
cd <APE.projectDir>  
ant APE-copy
```


Project configuration: APE.properties

```
#Set the base URI of the ARE to use
#Default: Relative to the location of APE.jar
#ARE.baseURI=../../ARE/

#Set the model src file(s) / src folder(s), either relative to APE.projectDir
or as absolute directory or file URIs
#Seperate several URIs by semicolon (;)
#Default: bin/ARE/models (this value will always be appended to the property
to include custom models in the custom/bin/ARE/models subfolder)
#APE.models=../../ARE/models/CameraMouse.acs;../../ARE/models/ImageDemo.acs

#The path to the build folder, can be relative or absolute
#Default: build
#APE.buildDir=build/

#currently: exe|msi|deb|image|all|none
#image: Only the final folder structure is created which can be used to create
a .zip file.
#all: all supported installer types for the platform are created
fx.deploy.nativeBundles=all

#Name and version of the application
fx.application.name=CamMouse-Webinar
fx.application.version=0.1

#Provide more detailed information here
fx.info.title=Assistive Technology Application
fx.info.vendor=Asterics consortium
fx.info.description=Assistive Technology Application based on AsTeRICS
fx.info.license=MIT | GPLv3 with Exception
fx.info.category=Assistive Technology
```

Demo: Create custom Camera Mouse

Requirements

[Java Development Kit 8 \(32-bit\)](#)

[Apache ant >= 1.9.1](#)

1. Create simple camera mouse model of tutorial before

2. Save model to folder

```
C:\Program Files (x86)\AsTeRICS\APE\defProjectDir\custom\bin\ARE\models
```

3. Change to

```
cd C:\Program Files (x86)\AsTeRICS\APE\defProjectDir\
```

4. Edit `APE.properties` (e.g. change `fx.application.name`)

5. Open command window (Shift+Right click, Open command window here)

6. Execute

```
ant APE-copy
```

7. Start `ARE.exe` in

```
C:\Program Files (x86)\AsTeRICS\APE\defProjectDir\build\merged\bin\ARE
```

Shrink of complete ARE of AsTeRICS framework to custom ARE of camera mouse

269 MB → 62.5 MB



Repositories of AsTeRICS demos (building blocks)

- The demos are hosted as **APE-based projects** at the [P4AllBuildingBlocks repository](#)
- The repositories are **organized in sub-folders**
- Can be built and deployed **stand-alone** (as single building block) or **in a bundle**
- Can be reused as template projects (forked)
- Check screencast about camera mouse packaging

Create a standalone
Camera Mouse
SW package in minutes

Link Summary

- AsTeRICS homepage: <http://www.asterics.eu/>
- AsTeRICS github: <https://github.com/asterics/AsTeRICS>
- Demo page: <http://asterics.github.io/AsTeRICS/demos.html>
- P4AllBuildingBlocks: <https://github.com/asterics/P4AllBuildingBlocks>
- APE: [https://github.com/asterics/AsTeRICS/wiki/AsTeRICS-Packaging-Environment-\(APE\)](https://github.com/asterics/AsTeRICS/wiki/AsTeRICS-Packaging-Environment-(APE))