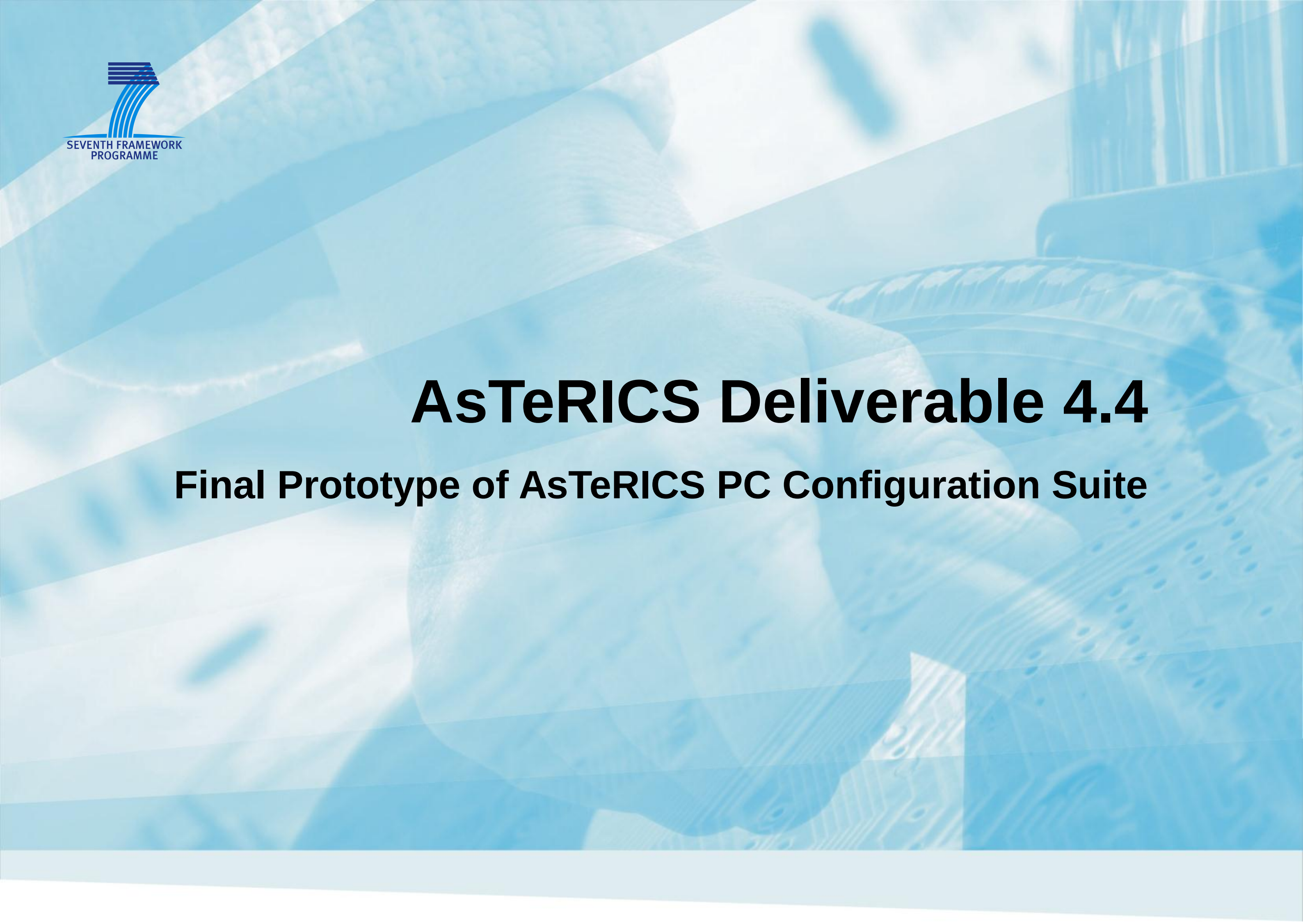
SEVENTH FRAMEWORK
PROGRAMME



AsTeRICS Deliverable 4.4

Final Prototype of AsTeRICS PC Configuration Suite

FHTW, KI-I

AsTeRICS

European Commission
Information Society and Media

Document Information

Issue Date	**30 June 2012**
Deliverable Number	**D4.4**
WP Number	**WP4 Software Development**
Status	**Final**
Dissemination Level	**CO** **PU Public** **PP Restricted to other programme participants (including the Commission Services)** **RE Restricted to a group specified by the consortium (including the Commission Services)** **CO Confidential, only for members of the consortium (including the Commission Services)**

AsTeRICS – Assistive Technology Rapid Integration & Construction Set
Grant Agreement No.247730
ICT-2009.7.2 - Accessible and Assistive ICT
Small or medium-scale focused research project

Disclaimer

The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

The document reflects only the author's views and the Community is not liable for any use that may be made of the information contained therein.

Version History

Version	**Date**	**Changed**	**Author(s)**
0.1	05.06.2012	Initial draft	C. Veigl
0.2	12.06.2012	Adding Content	R. Ossmann
0.3	13.06.2012	Adding Content	R. Ossmann
0.4.1	21.06.2012	Peer Review	T. Drajsajtl
0.4.2	25.06.2012	Peer Review	K. Kakousis
0.5	26.06.2012	Rework after peer review	R. Ossmann
Final	28.06.2012	Final changes	R. Ossmann

Glossary and Declaration of Terms

1 Terms Specific to AsTeRICS

ACS .... AsTeRICS Configuration Suite

A graphical software application running on the host PC for AsTeRICS model configuration and monitoring of the runtime system

ARE .... AsTeRICS Runtime Environment

The configured system model (which consists of pluggable components and their interconnections) which has been deployed to the execution environment (usually the AsTeRICS embedded computing platform)

EP .... Embedded Platform (aka AsTeRICS Personal Platform)

A customized hardware / PCB with high performance low power CPU and dedicated Communication Interface Modules to support connection of sensors and actuators.

Active Model .... The active model on the ARE

The model, which is active on the ARE. This means, a model was successfully transferred and deployed by the ARE. It does not say anything about, if the ARE is running or not.

ARE Storage ... A special storage for models

The ARE has a special area, where models can be stored. The models, being stored in this storage can be selected by the touch panel of the EP. So, different models can be switched easily.

Bundle File ...... Definition of a plugin

The bundle file is used to store all interface information of a plugin, so it can be instanced as a component within the ACS. These interface information are name and type, input and output ports, event listeners and triggers, properties and the optional GUI information.

Table of Content

1 Introduction

This document describes the accomplished work for the AsTeRICS Configuration Suite software application, which allows creation, modification and file management of AsTeRICS models for Assistive Technology use cases by means of a graphical user interface, as outlined in the Description of Work [1] and in the specification of the AsTeRICS system architecture (deliverable D2.1 [2] and deliverable D2.2 [3]).

The features of the ACS include not only the graphical setup of models – where sensor-, processor- and actuator-plugins are chosen, linked together and configured to form meaningful assistive scenarios – it also serves as the primary gateway to the AsTeRICS Runtime Environment (see deliverable D4.2 [4] and deliverable D4.5 [5]) which provides the necessary middleware and infrastructure to deploy AsTeRICS models for Assistive Technology applications.

By utilization of the AsTeRICS Application Programming Interface (ASAPI, see [4]), the ACS manages remote connections to the ARE, synchronizes the model state between ARE and ACS, enables download and upload of models in .xml-format to/from the runtime environment and handles logging- and error messages from the ARE. All necessary settings (like connection properties and IP addresses) can be adjusted and saved via the ACS' options dialogs.

For configuring individual parameters of plugins, the ACS features a dedicated editor which integrates into the graphical design window. All available functional components (plugins) and their properties are read from an .xml description at ACS start-up and the configuration editor adapts dynamically to these resources.

A strong focus has been laid on accessible design of the ACS application:

- Colour schemes can be adjusted and saved – a scheme with strong colour contrast is supplied per default
- The accessibility support of Microsoft .NET framework (UI Automation [7]) has been used
- All important menu options can be reached via hotkeys
- Graphical design of configurations (like drawing channel connections between plugin ports) is also possible via keyboard-only operation and hotkeys
- The design window can be magnified
- Tooltips are available for functions and property fields
- Various notification dialogs are available but can be omitted if desired

In the following sections, the functionalities of the ACS will be described in more detail, and examples for the ACS' capabilities to build AsTeRICS models will be given.

1.1 Relationship to other AsTeRICS Deliverables

This deliverable is related to the following AsTeRICS deliverables:

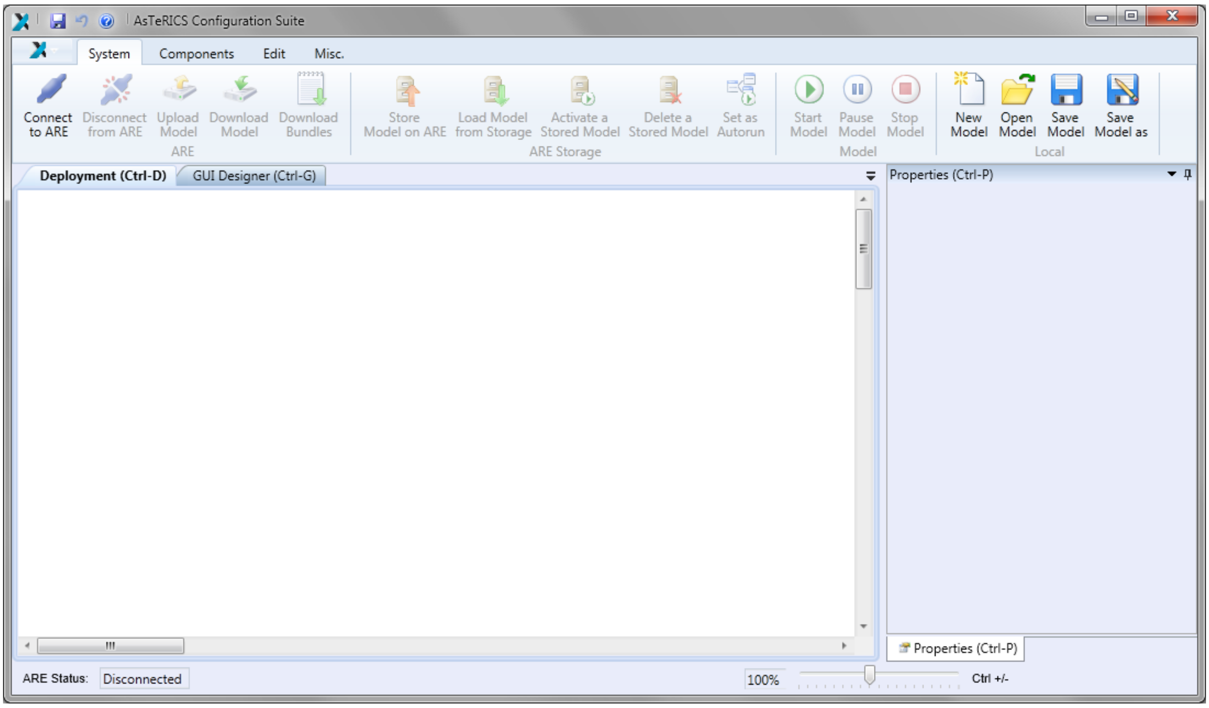
- D2.1 (System Specification and Architecture): This document describes in detail the hardware and software requirements for the AsTeRICS system and, where possible, specifies features and characteristics of the hardware and software architecture [2].
- D2.2 (Updated System Specification and Architecture): This document updates the System Specification and Architecture [2] with requirements emerged after the completion of the user-based evaluation of prototype 1. Also existing requirements have been to reflect new needs and priorities.
- D4.2 (Prototype 1 of the AsTeRICS Runtime System): This document describes the AsTeRICS Runtime Environment (ARE) [4].
- D4.5 (Final Prototype of the AsTeRICS Runtime System): This document describes the updates and improvements of the AsTeRICS Runtime Environment (ARE) [5].
- Developer's Manual: the updated version of the developer's manual for the final AsTeRICS prototype contains further information about the architecture and the AsTeRICS SW-framework, including plugin development examples and how-to, ASAPI / Thrift updates, as well as updates on naming conventions [8].

1.2 Update of Deliverable 4.1

This deliverable is an update of the AsTeRICS Deliverable 4.1 "Prototype 1 of AsTeRICS PC Configuration Suite" [6]. To increase the readability and reduce the review effort, only changes, updates and enhancements of the ACS are described.

2 Functions

The ACS is based on Microsoft .NET technology using Windows Presentation Foundation (WPF) for the graphical representation. For the menu system, Microsoft Ribbons are used. Figure 1 shows the ACS after starting it.



ARE Status: Disconnected

Figure 1: ACS after Startup

Compared to the ACS PT1, the development area is extended with the GUI designer. Within this feature, the GUI on the ARE can be designed. This means, if a component has a graphical representation (like an oscilloscope or a slider), this element will appear in the GUI designer, where the size and the position can be changed (see Figure 2).

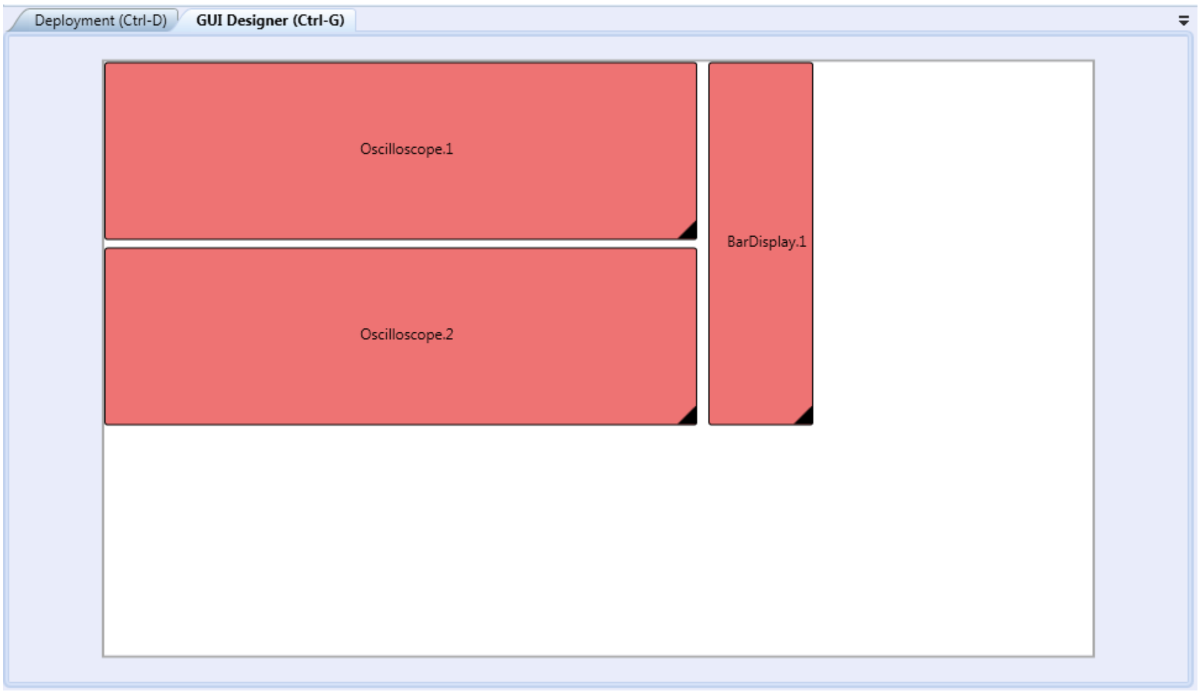


Figure 2: The GUI Designer

2.1 Create and Edit a Model

To support the users in finding their way through the increasing amount of components, the three main component categories (*Sensors, Processors, Actuators*) now have subcategories. The subcategories are freely definable within the bundle description file. Figure 3 shows the new *Components* tab, also containing the *Saved Groups* category (see section 2.1.1).

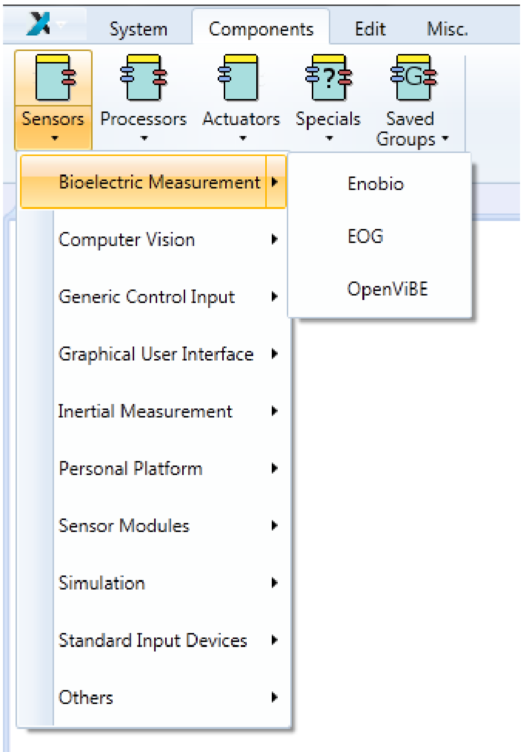


Figure 3: The Components Tab

2.1.1 The Edit Tab

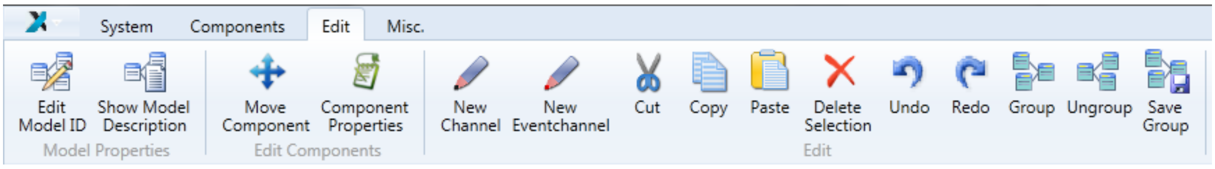


Figure 4: The Edit Tab

The *Edit* Tab has been completely reworked to increase the usability, making it easier for the users to deal with the increased functionality.

Description of the new *Edit*-Functions:

- *Edit Model ID* allows setting or editing a unique model ID. The ID is auto generated when a new model is created. This ID is used by the ARE to save model-related information.
- *Show Model Description* opens a window containing the model description. The description is divided in three parts: a *Short Description* giving an overview of the functionality of the model; the *Model Requirements* listing the requirements (e.g. special Sensors) of the model and *Detailed Description,* giving detailed information about the functions and parts of the model
- *Cut* is for cutting the selected components (incl. channels and event channels) out of the model and storing them in the clipboard.
- *Copy* is for copying the selected components (incl. channels and event channels) and storing them in the clipboard.
- *Paste* is for copying components form the clipboard to the model.
- *Delete Selection* deletes the selected components, channels and event channels. It replaces the functions *Delete Component*, *Delete Channel* and *Delete Eventchannel*.
- *Group* the selected components (incl. channels and event channels) to one group. The selected components must not contain a group. All connected ports and event ports will be available as external connection points of the group. Figure 5 shows a very simple example, the grouping of two components to one group.

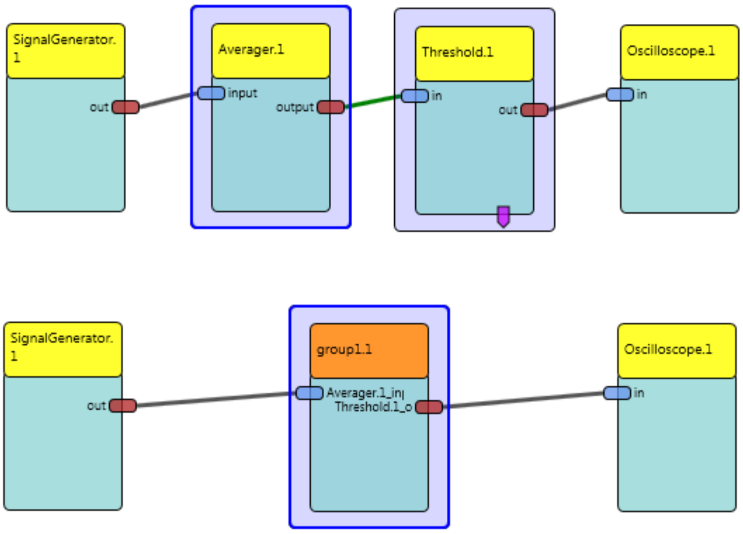


Figure 5: Grouping of two components

- *Ungroup* the selected group, showing the single components of the group.
- *Save Group* saves a selected group, so the group can be reused in this or in other models. The saved groups are available in *Components / Saved Groups* (see section 2.1).

2.2 Selection

It is now possible to select several components, channels and event channels at the same time. The selection can be made with the mouse (by dragging a selection rectangle or by the usage of the *Ctrl*-key, like in many standard programs) or with the keyboard. The selected components are marked with a blue frame. Figure 6 shows a model with several selected components.

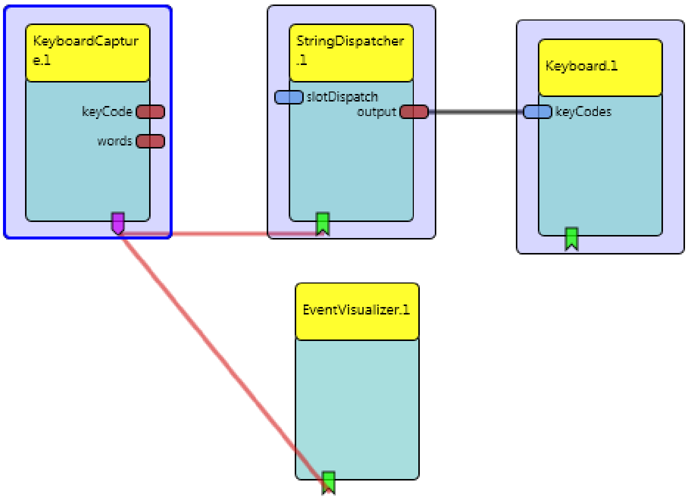


Figure 6: Model with three selected components

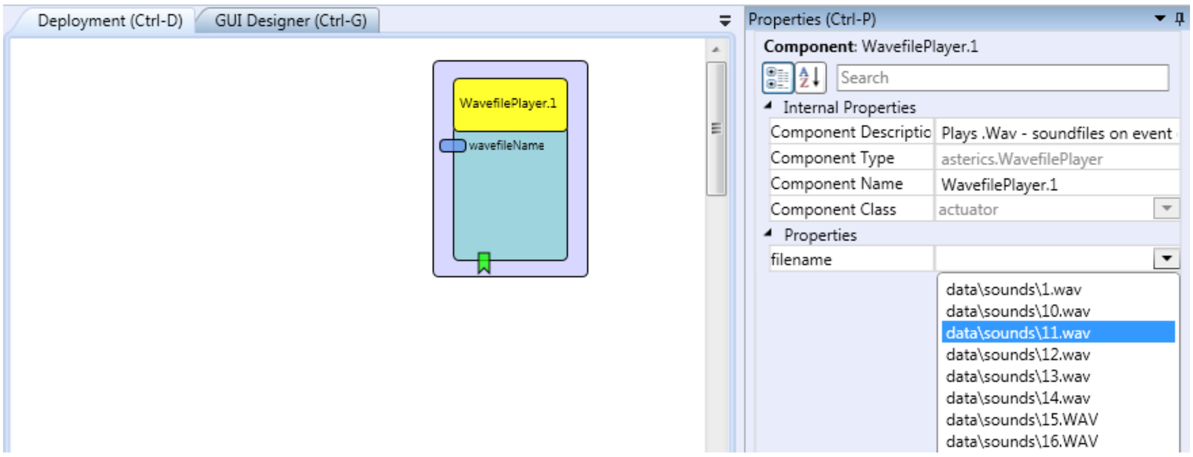
When the ACS is in the running mode (a model is running), it is now possible to select a component directly, setting the properties. In the past, this was only possible, using a combo box.

The available event triggers and event listeners will now be displayed as separate tabs when a component will be selected and displayed in the property editor.

2.3 Dynamic Properties

A new concept within the final prototype is *dynamic properties*. This means, properties in the property editor can become values from the component on the ARE. This feature might be useful in several cases, as example if a plugin is hardware dependent (selecting a soundcard or a midi player), or depending on files on the target system, like several sound files.

If a plugin is implementing a dynamic property, the values will be requested from the ARE, as soon as the ACS is synchronised with the ARE. Then, the received values can be selected in a combo box within the property editor (as shown in Figure 7).



Component: WavefilePlayer.1

Component Descriptic	Plays .Wav - soundfiles on event
Component Type	asterics.WavefilePlayer
Component Name	WavefilePlayer.1
Component Class	actuator
filename	

data\sounds\1.wav
data\sounds\10.wav
data\sounds\11.wav
data\sounds\12.wav
data\sounds\13.wav
data\sounds\14.wav
data\sounds\15.WAV
data\sounds\16.WAV

Figure 7: Plugin with Dynamic Property

2.4 Control the ARE

In the system-tab, the group *ARE* handles the functionalities for connecting to and communicating with the ARE

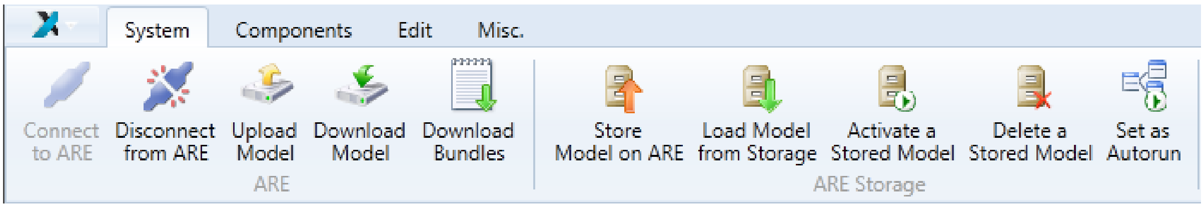
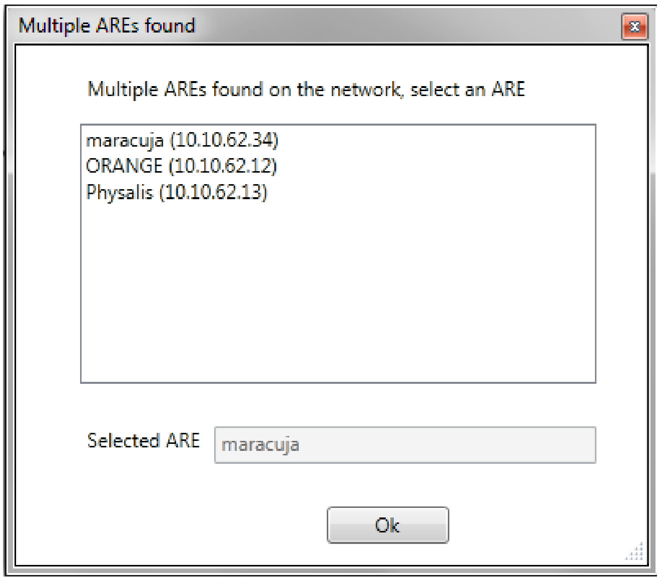


Figure 8: System Tab, ARE and ARE Storage Group

- *Connect to ARE* connects the ACS with the ARE. Besides the possibilities of using a connection dialog and setting the IP of the ARE in the options dialog, new auto detection has been implemented. This functionality scans the local network for available ARE's. If only one ARE is detected, the connection will be established, if several ARE's are detected, a selection dialog will be displayed (as shown in Figure 9).



Multiple AREs found

Multiple AREs found on the network, select an ARE

maracuja (10.10.62.34)
ORANGE (10.10.62.12)
Physalis (10.10.62.13)

Selected ARE: maracuja

Figure 9: Multiple ARE's have been detected

- *Download Bundles* transmits the bundle description from the ARE to the ACS. All components available on the ARE and their properties are stored in this bundle description. After the download, the new bundle is set active in the ACS. To store a new bundle, the *Configure Bundles* dialog can be used (see section 2.5).
- *Set as Autorun* stores the active model on the ARE storage and sets it as "autorun". Each time the ARE is started, this model will also be started.

2.5 The Miscellaneous Tab

Within the tab Miscellaneous (*Misc.*), status information and error logging can be requested from the ARE (if an active connection to the ARE is available). Additionally, the Bundle Manager can be opened and external tools can be called. The external tools are developer tools, supporting the creation of new plugins for the AsTeRICS.

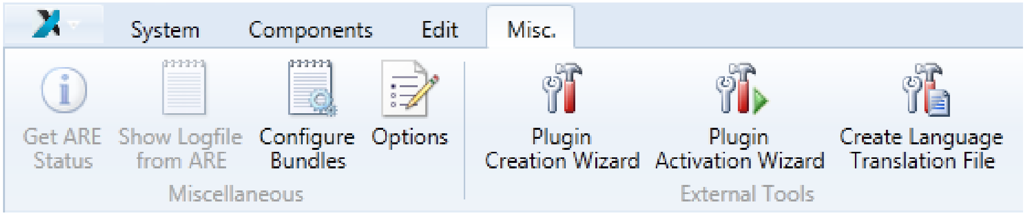
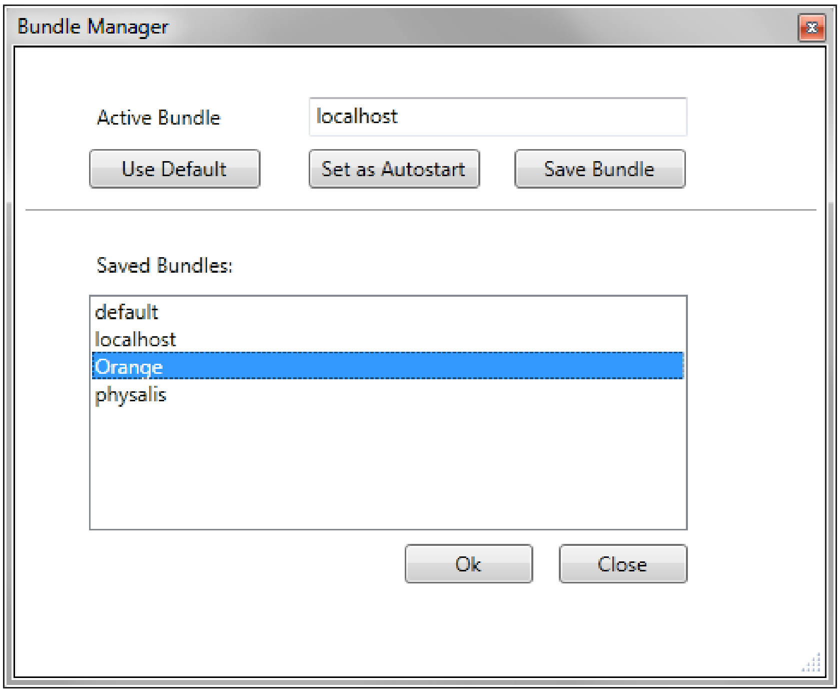


Figure 10: The Miscellaneous Tab

- *Configure Bundles* opens the Bundle Manager (see Figure 11) in which downloaded bundles can be saved and administered. Within the bundle manager, the following functionalities are provided:
  - *Use Default* sets the default ACS bundle (the one, being delivered with the ACS) as active bundle

- *Set as Autostart* sets the active bundle as autostart bundle, which will be loaded at ACS startup.
- *Save Bundle* saves the active bundle (e.g. a downloaded bundle from the ARE) into the ACS folder.
- The saved bundles list shows all saved bundles. A bundle can be selected and set active.



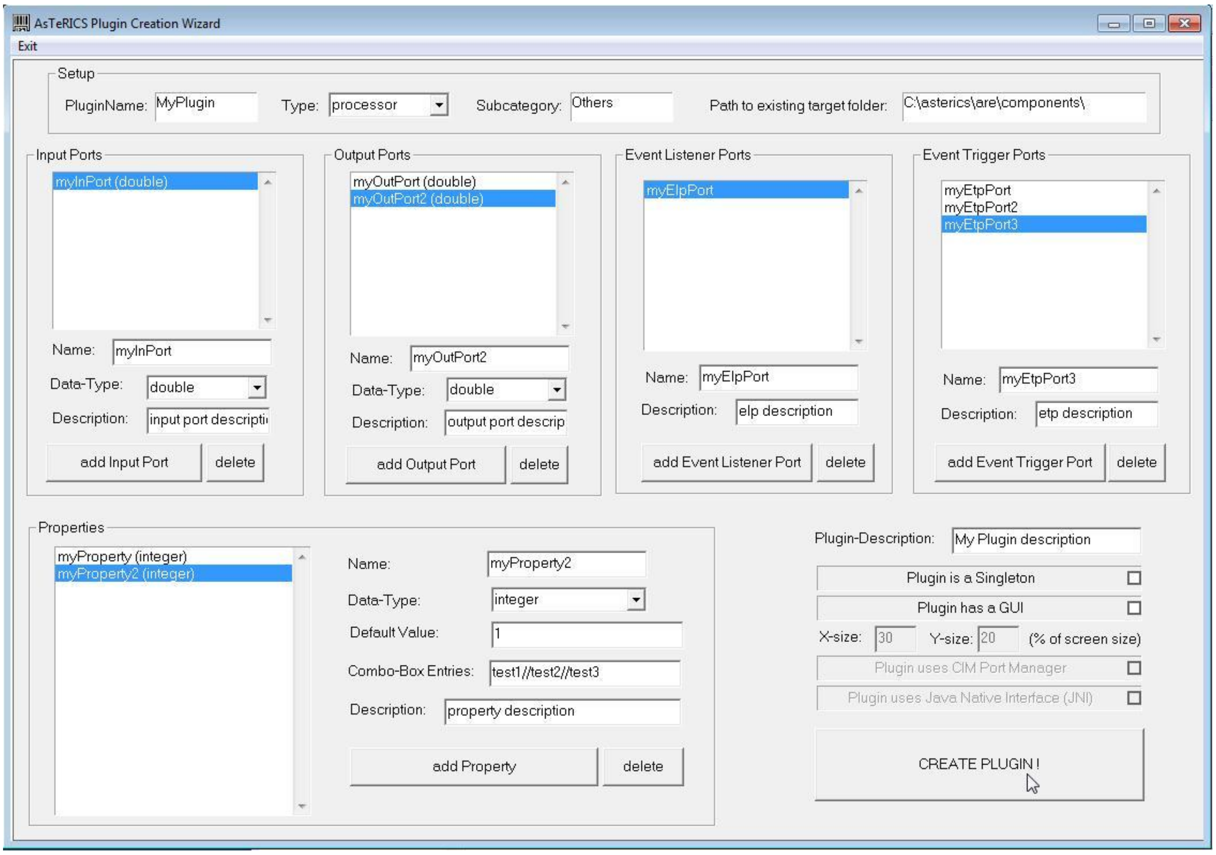
Active Bundle: localhost

Saved Bundles:

default
localhost
Orange
physalis

Figure 11: The Bundle Manager

- The *Plugin Creation Wizard* is an external tool to create the stub for a new plugin. All characteristics, like input and output ports, events and properties can be defined. As output, a java-class and all necessary files for an ARE plugin will be created. Figure 12 shows a screenshot of the tool, a more detailed function description can be found in the AsTeRICS Developer Manual [8].



AsTeRICS Plugin Creation Wizard

Setup

PluginName: MyPlugin

Type: processor

Subcategory: Others

Path to existing target folder: C:\asterics\are\components\

Input Ports: myInPort (double)

Name: myInPort

Data-Type: double

Description: input port descripti

Output Ports: myOutPort (double), myOutPort2 (double)

Name: myOutPort2

Data-Type: double

Description: output port descrip

Event Listener Ports: myElpPort

Name: myElpPort

Description: elp description

Event Trigger Ports: myEtpPort, myEtpPort2, myEtpPort3

Name: myEtpPort3

Description: etp description

Properties: myProperty (integer), myProperty2 (integer)

Name: myProperty2

Data-Type: integer

Default Value: 1

Combo-Box Entries: test1//test2//test3

Description: property description

Plugin-Description: My Plugin description

Plugin is a Singleton □

Plugin has a GUI □

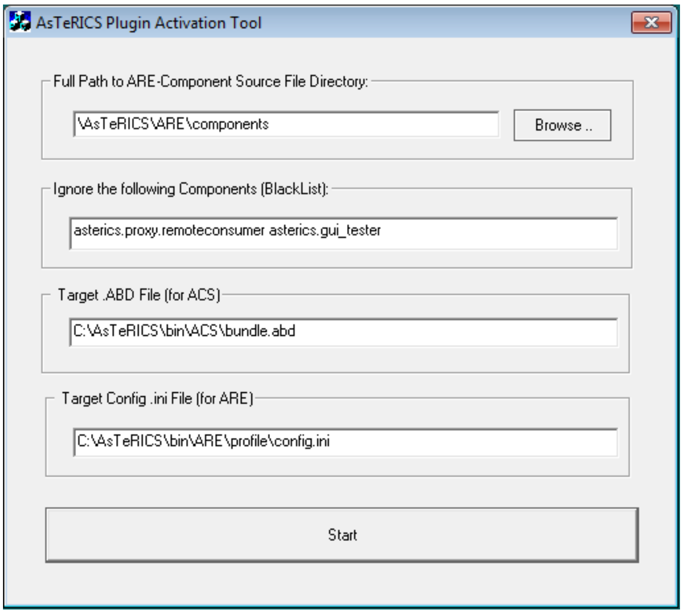
X-size: 30 Y-size: 20 (% of screen size)

Plugin uses CIM Port Manager □

Plugin uses Java Native Interface (JNI) □

Figure 12: Screenshot of the AsTeRICS Plugin Creation Wizard

- The *Plugin Activation Wizard* is an external tool to create bundle files (describing the interface of a plugin) and configuration files for the ARE. Therefore, the source code folder of the ARE will be scanned. The AsTeRICS Developer Manual [8] gives a more detailed description about this development tool, see Figure 13 for a screenshot.



AsTeRICS Plugin Activation Tool

Full Path to ARE-Component Source File Directory: \AsTeRICS\ARE\components

Ignore the following Components (BlackList): asterics.proxy.remoteconsumer asterics.gui_tester

Target .ABD File (for ACS): C:\AsTeRICS\bin\ACS\bundle.abd

Target Config .ini File (for ARE): C:\AsTeRICS\bin\ARE\profile\config.ini

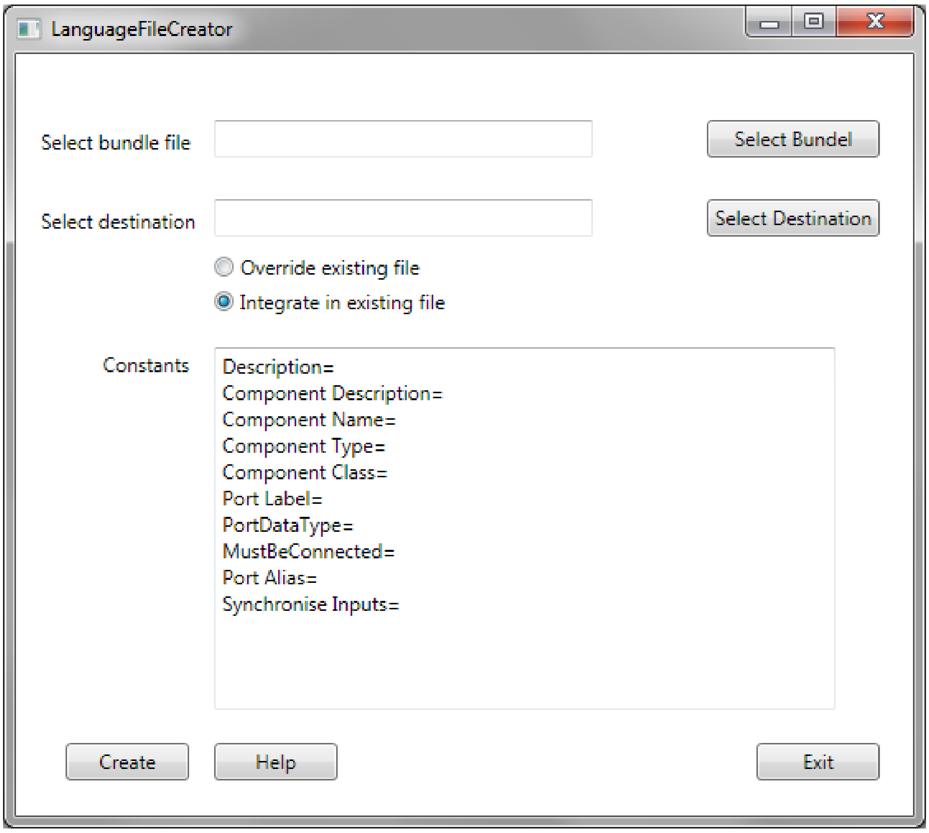
Figure 13: Screenshot of the AsTeRICS Plugin Activation Wizard

- *Create Language Translation File* opens another external tool for creating language translation files for the properties (as shown in Figure 14). This tool scans the bundle file and lists all properties in a text file, which can be extended with the translation for any language. This text file will be loaded at startup of the ACS, so also the components now have multi language support.

After selecting a bundle file, the destination file has to be set. If the destination file (a language translation file) already exists, it can be chosen, if the file should be overwritten or integrated into an existing language file. The *Constants* are strings, occurring in each plugin. By using the constants, they just must be translated once. All other strings must be translated in the created textfile, where all strings, ending with *equal to (=)* must be translated. Example:

```
Frequency=
Amplitude=
```

When the translation is finished, the language file must be copied to the *lang* subfolder within the ACS folder (if the file isn't there already). The file must be named as the supported languages of the ACS. At the moment, *en-GB*, *de-AT*, *es-ES* and *pl-PL* are supported. Furthermore, the file must have the ending .txt. For the German language file, it would look like *de-AT.txt*.



Select bundle file

Select destination

○ Override existing file
● Integrate in existing file

Constants

Description=
Component Description=
Component Name=
Component Type=
Component Class=
Port Label=
PortDataType=
MustBeConnected=
Port Alias=
Synchronise Inputs=

Figure 14: Screenshot of the Language File Creator

3 General Updates

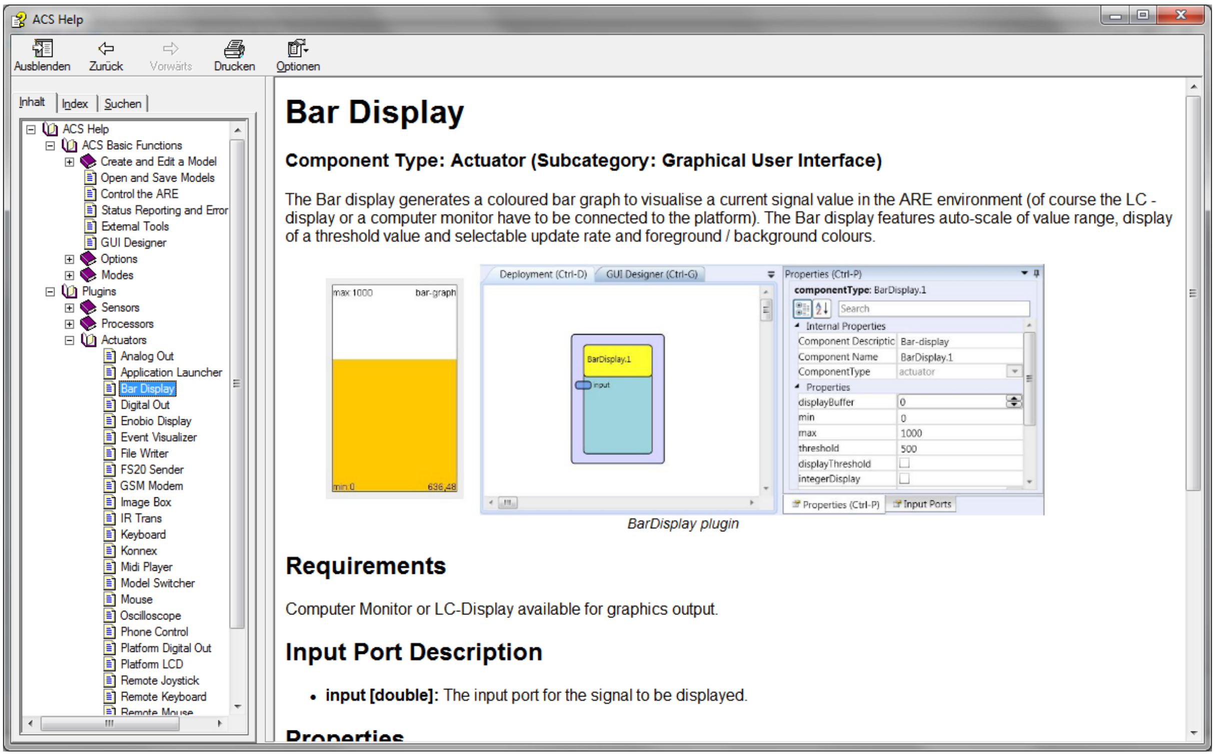
Besides the already listed new functionalities, many things had been changed in the background logic of the ACS. As a major update, the ACS now uses the .NET 4.0 framework instead of .NET 3.5. Version 4.0 of the .NET framework is faster, more stable and future-proof. Besides the framework itself also other external libraries have been updated to their latest version.

The structure of the models has been redesigned and changed. Therefore, some model processing algorithms required an update. During these updates, a version ID has been included into the model. Using this versioning, automatic updates from older models to the newest version can be processed.

The main menu has been enriched with a „Recently opened files“ list, containing the last opened models.

When a model will be updated or changed and saved, a backup file of the model will be created automatically. This was one request during the user tests.

As a final step, the help (as shown in Figure 15) has been updated and extended, containing a general help about the ACS and the function description for each component (plugin).



Bar Display

Component Type: Actuator (Subcategory: Graphical User Interface)

The Bar display generates a coloured bar graph to visualise a current signal value in the ARE environment (of course the LC - display or a computer monitor have to be connected to the platform). The Bar display features auto-scale of value range, display of a threshold value and selectable update rate and foreground / background colours.

componentType: BarDisplay.1

Internal Properties

Component Descriptic	Bar-display
Component Name	BarDisplay.1
ComponentType	actuator

Properties

displayBuffer	0
min	0
max	1000
threshold	500
displayThreshold	□
integerDisplay	□

BarDisplay plugin

Requirements

Computer Monitor or LC-Display available for graphics output.

Input Port Description

- **input [double]:** The input port for the signal to be displayed.

Properties

Figure 15: Screenshot of the ACS Help

4 Errors and Known Problems

When the ACS is connected to the ARE and the ARE is closed or crashes, the ACS has problems detecting this broken connection. This problem is the result of an inaccurate implementation within the thrift[1]-Framework, being used for the ASAPI communication. This problem will be further investigated during the integration phase and as a possible solution a self-implemented ping-command could be used.

At the moment, the ACS does not detect, if the Microsoft .NET framework is installed or not and if the right version of the framework is installed. When the framework is not installed, an error message from the Windows system will be shown, but with no link to the installation files. This will be fixed, using an installation wizard for the installation of the ACS. Within this wizard, the availability of the .NET framework can be checked and if it is missing, it can be installed. The wizard will be created at the end of work package 5 – Integration.

[1] http://thrift.apache.org/

5 Conclusion

The new functions for the final prototype fulfil two targets: on the one hand, adding new functionality, on the other hand, reducing the complexity, so that the ACS is easier to handle by non-experienced users.

New features like the GUI designer make the ARE GUI design easy, fast and understandable, so that the user interaction with the ARE is now more intuitive. The redesign of the *Edit*-Tab and functionality makes is easier to edit a model and to find a way through the many plugins and their functions. A big advantage is the new grouping functionality, where complex parts of models can be grouped to one component, saved and reused.

In summary, the ACS is a powerful tool for the design of models and the control of the ARE. The inline help system covers the functionalities of the ACS and all additional plugins, thus supporting new as well as advanced users when building highly tailored and adapted models.

References

1 AsTeRICS Description of Work, technical annex 1 of the AsTeRICS Grant agreement

2 AsTeRICS Deliverable D2.1 – “System Specification and Architecture”

3 AsTeRICS Deliverable D2.2 – “Updated System Specification and Architecture”

4 AsTeRICS Deliverable D4.2 – “Prototype 1 of the AsTeRICS Runtime System”

5 AsTeRICS Deliverable D4.5 – “Final Prototype of the AsTeRICS Runtime System”

6 AsTeRICS Deliverable D4.1 – “Prototype 1 of the AsTeRICS Configuration Suite”

7 Microsoft Windows Automation API: UI Automation. URL: http://msdn.microsoft.com/de-de/library/ms726294%28vs.85%29.aspx, accessed: 12.06.2012

8 AsTeRICS Developer Manual, V1.2 beta. URL: http://www.asterics.eu/fileadmin/user_upload/Documentation/DeveloperManual.pdf