

TUI Framework – Configuration Tool

*Documentation of the graphical interface of the open source TUI framework
developed by Fraunhofer IPK Berlin.*

Version 6/18/2013

Authors: Oliver Belaifa, Alexander Kaufmann, Johann Habakuk Israel

Contents

Introduction	3
Starting the Application	5
Projects and configurations.....	5
Menu bar	6
Options of the menu "File"	6
New Project	6
Open Project.....	6
Save Project/Save Project as	6
New Configuration.....	6
Import Configuration	7
Export Configuration.....	7
Exit	7
Options of the menu "Edit"	7
Change EntityInstance Name	7
Rearrange EntityInstance Positions.....	7
Delete Selection.....	7
The panels of the application.....	8
Entity-types	8
The Playground.....	9

Create New TUI Object Type	10
Parameter.....	11
Further Information.....	12
Web Ressources	12
References	12
List of Figures	12

Introduction

The TUI Framework (TUI: Tangible User Interface) is a toolkit to integrate interaction devices in interactive applications. It implements a device abstraction layer to achieve code-independence from various specific device drivers and APIs. More information about the TUI framework can be found in (Israel, Belaifa, Gispén, & Stark, 2011) and online at <https://github.com/fraunhoferipk/tuiframework>. In order to understand the terminology of this document, please refer to this paper. The following is only a brief description of the main entities:

- **TUI objects** usually consist of a set of sensors which generate data streams (e.g. three dimensional tracking data or analog button signals) and a set of physical properties which accept data streams (e.g. force feedback devices or LED arrays).
- **Devices**, e.g. tracking systems or force feedback devices are controlled by different specialized drivers.
- **TUI Application** holds software representations of each TUI object to receive and send data from and to the physical objects.

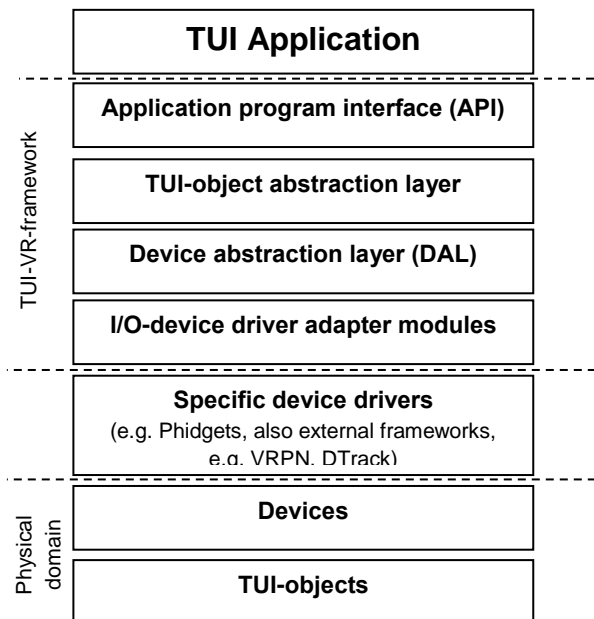


Figure 1: Principle architecture of the TUI-VR-framework

This document is a short manual for of the TUI Server configuration tool. It is a graphical tool which allows you to build and edit configuration files for the TUI Server. The tool generates configuration file in

XML format for particular setups of interaction devices (TUI devices) and software objects (TUI objects). The XML files can be used to configure the TUI server.

The advantage of this tool is that you do not have to edit the XML files manually. It allows you to make use of the broad functionality of the framework, it is for example possible to handle two devices separately on two different IP-addresses in the "Parameter Window". It also allows to create different configurations with multiple entities (Multi Stream Processors, device types or object types), which could be configured separately.

This tool is still in a pre-Beta status and has some known issues, for example:

- it doesn't remember the last opened folder or saving point
- the graphical position in the playground is not saved to the configuration
- arrows from one entity to another cannot be deleted by the context menu
- TUI server configurations, which are in the XML format, have to be created and edited by the programmer, thus failures may occur.

Starting the Application

To start the tool on a Windows system you just have to double-click on the file "TUIServerSettings". To start the tool on a Linux system you have to execute TUIServerSettings.jar on the Shell. From the start a new project is generated. The windows will be empty and you have to load a configuration or create a new one. Figure 2 shows the initial interface of the tool.

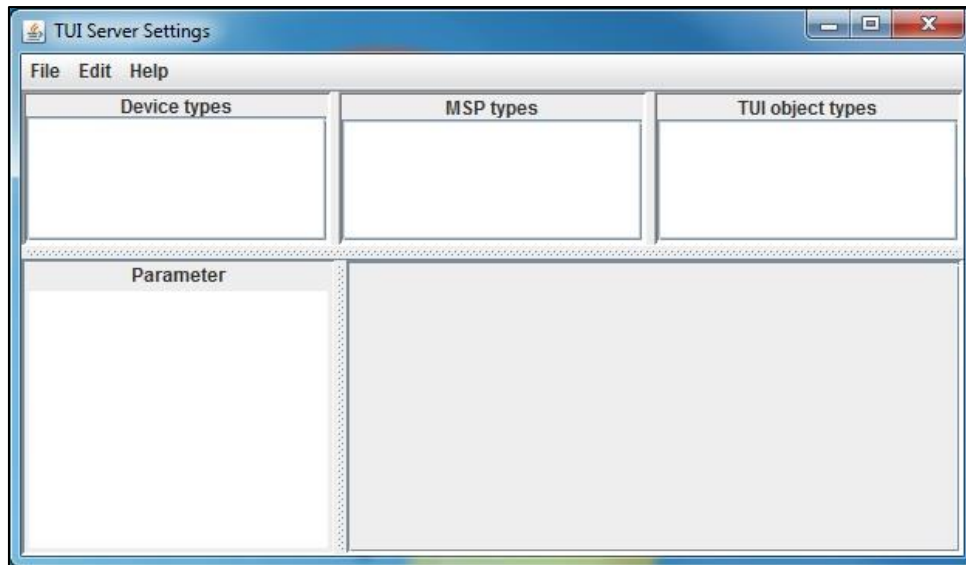


Figure 2: TUI framework

Projects and configurations

The TUI Server settings tool uses projects and configurations for the workflow. The difference between a project and a configuration is that the entity-types saved in a project are managed in a database. The configuration consists of instantiated entity-types and their connections. Project-files are saved with the ending ".tuiproj" and configurations with the ending ".xml". Only configuration can be transferred to the TUI server. Configurations can be imported in a project or exported out of a project. When a configuration is imported into a project the actual configuration will be overwritten and the imported entity-types will be integrated to the entity-type-database. So you can quickly generate various configurations.

Menu bar

If you click on the button "File" (Figure 3) in the menu bar it is possible to create, open and save a project and to create, import or export a configuration of a device. Under the button "Edit" (Figure 4) you have the possibility to rearrange the position of an entity instance or change the name of it. There you will also find the function to create a new TUI Object Type and delete a selection of the playground.

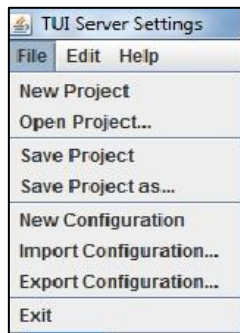


Figure 3: Options of the menu "File"

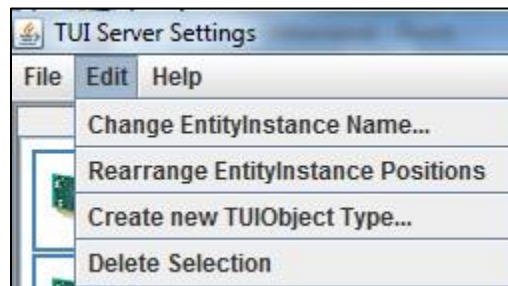


Figure 4: Options of the menu "Edit"

Options of the menu "File"

New Project

If you click on the button "New Project" in the menu bar a new project will be applied. By creating a new project the current entity-type-database and the current configuration will be deleted. The windows will be empty and you have to import or create a new configuration for your project. The project could be saved with the tasks "Save Project" or "Save Project as".

Open Project

With this task you can open an existing project and work on it. You have the possibility to change an existing configuration in the project or create a new one in it.

Save Project/Save Project as

With these options it is possible to save an existing project to an existing file or create a new name for a new project.

New Configuration

To create a new configuration in a new or an existing project the objects of the last loaded configuration will be loaded into their particular boxes, but the playground will be emptied. If there was no opened configuration before and you want to create a new one, the windows will be empty.

Import Configuration

With the option to import a configuration you can load an existing configuration for special devices or work on an existing configuration. Hereby the current configuration will be erased but the entity-type-database of the current project remains. Configurations always will be saved / exported as xml-files.

Export Configuration

This option is to export a new or an existing configuration to a xml-file for later processing.

Exit

To exit the TUI-Framework click on "Exit" in the menu "file" and the program will be closed.

Options of the menu "Edit"

Change EntityInstance Name

With this option you can change the name of an entity in the playground or one of the other windows. This option will also be shown if you right-click on an entity in one of the windows.

Rearrange EntityInstance Positions

If you have changed the positions of the entities in the playground you can rearrange the entity-instances to the last arrangement they have been saved to.

Delete Selection

With this option you can select objects and devices in the playground and delete them.

The panels of the application

Entity-types

This tool allows you to instantiate 3 different entity-types (Figure 5) in the playground. Device types, MSP types (Multi Stream Processor types) and TUI object types. Device types and MSP types are connectors which can only be defined by the programmer. TUI object types can be defined by the user. Device types are mostly hardware connections and MSP types are multi stream processors.

On the left side of an instantiated entity is the input and on the right side is the output for data. The output of a device type has always to have the same port type like the input it's connected to (example "analog" to "analog").

The different Multi Stream Processor can transfer the outcoming data of the entity-types from one to another port (for example from "MouseChannel" to "DigitalChannel"; Figure 6).

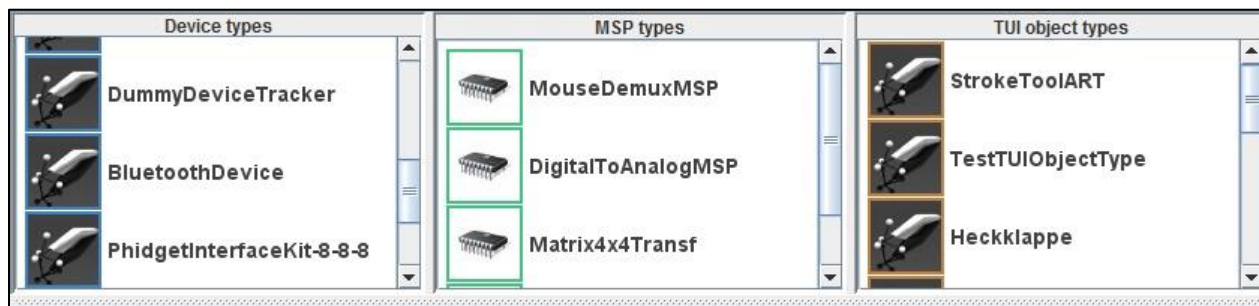


Figure 5: entity-types (Device types, MSP types, TUI object types)

The Playground

The playground shows the actual configuration. There you can connect the different entity-types to each other. Entities can be instantiated by drag-&-drop from the entity-type-panels. These entities can be connected by their ports. Only ports which got the same type can be connected. To connect more entities with others entity-types you have to instantiate a multi stream processor type. With a right-click on an entity in the playground or in the panels you have the options to change its name or delete it.

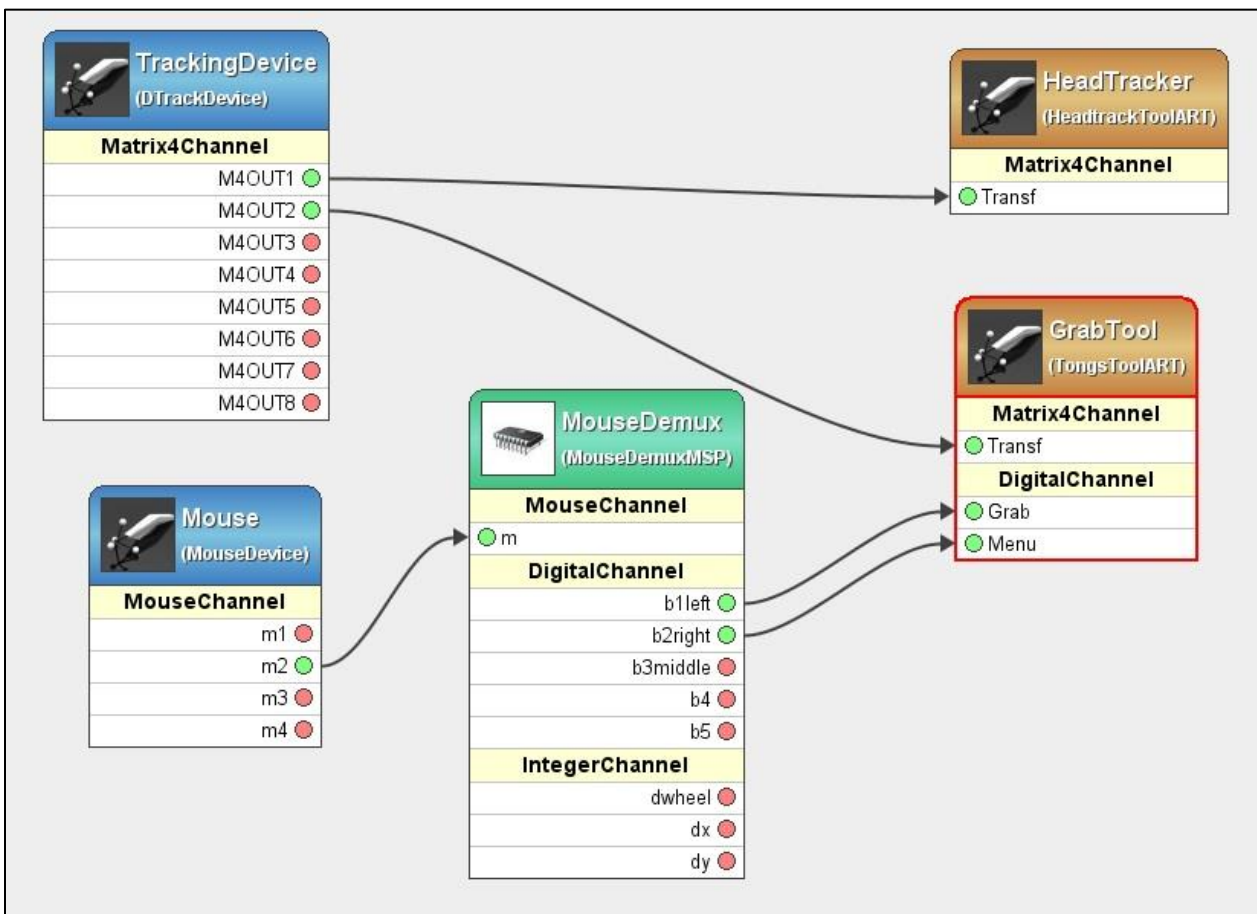


Figure 6: example for a configuration

In figure 6 is shown in an example for a configuration. The head of the instantiated device type is blue, of the MSP types green and of the TUI object types brown. On the right side of an instantiated entity are the ports for the outbound of data and on the left side the inbounds. Those ports will change their color from red to green if they are connected to an equal port of another entity. In this example configuration the *TrackingDevice* is connected to the *HeadTracker* and the *GrabTool* by the port *Matrix4Channel*. The *TrackingDevice* commits the 3D-transformation matrix of the marker-positions for the *HeadTracker* and

the *GrabTool*, so that they will be tracked in the 3D-visualization area. The *Mouse* is connected over its second port *m2* to the MSP type *MouseDemux*. This MSP type commits the data over its port *DigitalChannel* for the left button *b1left* to the *GrabTool* which has the same port type. So the button *b1left* is assigned to the function *Grab* of the *GrabTool*. The second button *b2right* is assigned to the function *Menu* of the *GrabTool*.

Create New TUI Object Type

If you click on the button "Create New TUI Object Type" a new window (Figure 7) will be opened and you can give your new TUI object type a Name and a Description. You also can add ports to connect multiple devices in the playground. You too can delete unnecessary ports. You can choose a name for your port and which type it should be and choose if it should be a source or a sink. The new object type will be shown in the box "TUI object types".

Create New TUIObjectType

TUIObjectType Name: TUI Object example

TUIObjectType Description:

Port Configuration:

Port Name	Port Type	Port Description	Source	Sink
port name	port type	port description	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Add Port Remove Selection Cancel Save

Figure 7: Create a new TUI object type

Parameter

As shown in Figure 8 in this window is listed with which IP-address a device type is connected to the server or which configuration channel it uses. You have the possibility to change the parameter, but only the programmers authorized to choose which kind of parameters the device should have (channel or IP-address). With different IP-addresses different devices could be controlled separately. The type of parameters for a device is defined in the xml-file. Also the parameters for a matrix could be changed here.

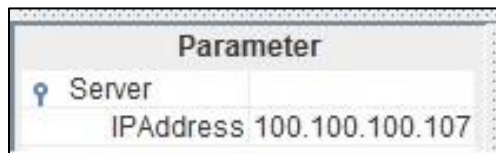


Figure 8: Parameter panel

Further Information

Web Ressources

<https://github.com/fraunhoferipk/tuiframework>

<http://www.ipk.fraunhofer.de>

References

Israel, J. H., Belaifa, O., Gispén, A., & Stark, R. (2011). An Object-centric Interaction Framework for Tangible Interfaces in Virtual Environments. *Fifth international conference on Tangible, embedded, and embodied interaction ACM TEI'11* (pp. 325–332). Funchal, Portugal: ACM Press.
doi:10.1145/1935701.1935777

List of Figures

Figure 1: Principle architecture of the TUI-VR-framework	3
Figure 2: TUI framework	5
Figure 3: options of the menu "File"	6
Figure 4: options of the menu "Edit"	6
Figure 5: entity-types (Device types, MSP types, TUI object types)	8
Figure 6: example for a configuration	9
Figure 7: Create a new TUI object type	10
Figure 8: Parameter panel	11