MICOBS Documentation

Release 2.2.0

University of Alcala

November 22, 2016

Contents

1 Installation instructions 3

2 Platform model 5

3 Indices and tables 9

MICOBS is a framework for developing component-based embedded software systems under a multi-platform approach and integrating different development and analysis technologies.

It has been developed by the Space Research Group of the University of Alcala.

Contents:

CHAPTER 1

Installation instructions

You need to follow the next steps to install MICOBS on your system:

1. Install Eclipse. MICOBS works with Eclipse version Kepler (4.3) or later. You need to install the **Modelling Tools** version following the instructions found in the official Eclipse web.

2. Install Subclipse 1.8 inside your newly installed Eclipse. You can find the installation instructions here.

3. Install Xtext, Xpand and QVT operational tools inside Eclipse. Inside Eclipse select *Help* → *Install New Software....* In the new window, select to work with *Kepler - http://download.eclipse.org/releases/kepler* or the one corresponding to the version you have installed. Under the *Modeling* group, select for installation the Xtext, Xpand and QVT operational SDKs. Once selected, click the *Next* > button and follow the instructions on screen.

4. Install MICOBS from the official update site. You have to select again *Help* → *Install New Software...* and include a new location through the *Add* button. Use MICOBS as the name of the new location. The link to the update site is:

http://www.micobs.com/eclipse/update/2.2.x/

5. Select to install all the packages from the repository and click the *Next* > button. Finally, follow the instructions on screen.

CHAPTER 2

Platform model

MICOBS defines a *platform model*. It is used to describe the different working platforms on top of which the software applications can be deployed. A platform is defined as a tuple. Each element of the platform shall be defined separately using a specific textual representation defined for that purpose. Each model element has a name and a version identifier. When referring to a given model element, the naming convention used within MICOBS is using the name followed by its version between brackets. Once the constituent elements are defined, the platform can be properly defined. Not every platform shall include all the elements, i.e. some of them are optional.

2.1 Model elements

The elements used to define a platform are the following:

Operating System Application Programming Interface (OSAPI) Defines the interface provided by the underlying operating system. E.g. POSIX(v13), RTEMSAPI(4.8). This field is **mandatory**. An example of an OSAPI model is the following:

```
osapi POSIX {

        version := v13;

        language := C(C99);

};
```

Operating System (OS) Defines the operating system on top of which the software applications will run. E.g. RTEMS(4.8). This field is **mandatory**. The model of the OS includes the list of APIs it provides, together with the supported hardware platforms on which it can be run. It might also define one or more configuration parameters whose values can be set when the application is deployed:

```
os RTEMS
{
        version := 4.8;

        languages := C(C99);

        supported osapis {

                supports RTEMSAPI(4.8) {};
                supports POSIX(v13) {};

        };
```

```
        supported platforms
        {
                supported platform SPARC_v8_GCC_4_x_LEON_2_any
                {
                        architecture := SPARC(v8);
                        compiler := GCC(4.x);
                        microprocessor := LEON(2);
                        board := any;
                        languages := asm_SPARC(v8);
                };

                supported platform SPARC_v8_GCC_4_x_LEON_3_any
                {
                        architecture := SPARC(v8);
                        compiler := GCC(4.x);
                        microprocessor := LEON(3);
                        board := any;
                        languages := asm_SPARC(v8);
                };
        };

        configuration parameters
        {
                integer MICROSECONDS_PER_TICK := 10000;
        };

};
```

Architecture The hardware architecture. E.g. SPARC(v8), ia32(i686). This field is **mandatory**. An example of a textual model of an architecture is the following:

```
architecture SPARC {

        version := v8;

};
```

Compiler The specific compiler that will be used to build the software application. E.g. GCC(4.2.2). If the compiler is not relevant to the development process, the corresponding field of the platform tuple can be left *blank*. Each compile must define its target architectures or *frontends*:

```
compiler GCC {

        version := 4.x;

        frontends {

                frontend C_C99 {

                        language := C(C99);
                        architectures := ia32(i686), SPARC(v7), SPARC(v8);

                };

                frontend CPP_98 {

                        language := CPP(98);
                        architectures := ia32(i686), SPARC(v7), SPARC(v8);
```

```
                    };

                    frontend ASM_IA32_I686 {

                            language := asm_ia32(i686);
                            architectures := ia32(i686);

                    };

                    frontend ASM_SPARC_V7 {

                            language := asm_SPARC(v7);
                            architectures := SPARC(v7);

                    };

                    frontend ASM_SPARC_V8 {

                            language := asm_SPARC(v8);
                            architectures := SPARC(v8);

                    };
            };
    };
```

Microprocessor The microprocessor that is at the core of the hardware platform. E.g. LEON(2) or LEON(3). If the specific microprocessor is not relevant to the development process, the corresponding field of the platform tuple can be left *blank*. The architecture implemented by the microprocessor shall be stated when defining its model:

```
    microprocessor LEON
    {
            version := 2;

            architecture := SPARC(v8);

    };
```

Board The development board that will execute the software application. If the board itself is not relevant to the development process, e.g. the applications do not depend on the characterstics of the board, the corresponding field of the platform tuple can be left *blank*. However, if the board is to be defined, so must be the microprocessor. An example of a board model is the following:

```
    board TSIM_LEON2
    {
            version := 2.x;

            microprocessor := LEON(2);

    };
```

Once the different models of the constituent elements are defined, the full platform can be defined using a specific textual language. If a given field is not relevant, it can be omitted using the *any* key.

2.2 Platform library

Every platform and their model elements are defined as textual files that shall be stored in one or more subversion (SVN) repositories. The references to the files and their locations are stored locally in Eclipse and the list of all the installed elements can be accessed from the MICOBS menu.

Links:

- Code: https://github.com/parraman/micobs

CHAPTER 3

Indices and tables

- genindex
- modindex
- search