

Κατανεμημένα Συστήματα II

Μάθημα Ελεύθερης Επιλογής, Εαρινού Εξαμήνου

Τομέας Εφαρμογών και Θεμελιώσεων

Ορέστης Ακριβόπουλος

Τετάρτη, 23 Μαρτίου 2011
Αίθουσα EAITY/0.I.6

Outline

1 Wiselib

2 Installing Wiselib

Τι είναι η wiselib ?

- Μία βιβλιοθήκη αλγορίθμων για ετερογενή δίκτυα αισθητήρων
- Επιτρέπει την εκτέλεση ενός υλοποιημένου αλγορίθμου σε συστήματα διαφορετικής αρχιτεκτονικής όπως :
 - Sensor Nodes: iSense, Contiki, TinyOS
 - Shawn, το οποίο είναι περιβάλλον εξομοίωσης
- Είναι εξ' ολοκλήρου γραμμένη σε C++
- Χρησιμοποιεί templates
- Το κύριο πλεονέκτημα της είναι ότι επιτρέπει την υλοποίηση αλγορίθμων με **generic** κώδικα, ο οποίος είναι **platform independent**.

Παράδειγμα

Η βασική ιδέα του τρόπου λειτουργίας της wiselib

A **concept** only exists in **documentation**. It describes exactly how an implementation must look like.

```
concept Radio
{
    static void enable();
}
```

Παράδειγμα

```
class iSenseRadioModel {  
    static void enable() { ... };  
}  
  
class ShawnRadioModel {  
    static void enable() { ... };  
}
```

```
concept Radio  
{  
    static void enable();  
}
```

A **model** is an **implementation** of a concept. It offers at least each **method** and each **type** that is described in the concept.

Παράδειγμα

```
class iSenseRadioModel {  
    static void enable() { ... };  
}  
  
class ShawnRadioModel {  
    static void enable() { ... };  
}
```

```
concept Radio  
{  
    static void enable();  
}
```

```
template<typename Radio>  
class Algorithm {  
    void enable() {  
        Radio::enable();  
    }  
};
```

Another implementation awaits a **model** as **template parameter**, but knows only the **concept**.

Παράδειγμα

```
class iSenseRadioModel {  
    static void enable() { ... };  
}  
  
class ShawnRadioModel {  
    static void enable() { ... };  
}
```

```
concept Radio  
{  
    static void enable();  
}
```

```
template<typename Radio>  
class Algorithm {  
    void enable() {  
        Radio::enable();  
    }  
};
```

Such a model is then
passed as a **template
argument** to an algorithm

```
Algorithm<iSenseRadioModel> algorithm_isense;  
Algorithm<ShawnRadioModel> algorithm_shawn;
```

Παράδειγμα

```
class iSenseRadioModel {  
    static void enable() { ... };  
}
```

```
class ShawnRadioModel {  
    static void enable() { ... };  
}
```

```
template<typename Radio>  
class Algorithm {  
    void enable() {  
        Radio::enable();  
    }  
};
```

```
Algorithm<iSenseRadioModel> algorithm_isense;  
Algorithm<ShawnRadioModel> algorithm_shawn;
```

```
concept Radio  
{  
    static void enable();  
}
```


Παράδειγμα

```
class iSenseRadioModel {  
    static void enable() { ... };  
}
```

```
class ShawnRadioModel {  
    static void enable() { ... };  
}
```

```
template<typename Radio>  
class Algorithm {  
    void enable() {  
        Radio::enable();  
    }  
};
```

```
Algorithm<iSenseRadioModel> algorithm_isense;  
Algorithm<ShawnRadioModel> algorithm_shawn;
```

```
concept Radio  
{  
    static void enable();  
}
```

Outline

1 Wiselib

2 Installing Wiselib

Installing Wiselib

- Το site της wiselib: <http://www.wiselib.org/>
- Κατεβάστε την wiselib από το svn repository:
<https://svn.itm.uni-luebeck.de/wisebed/wiselib/trunk/>
- Πληροφορίες για την εγκατάσταση του Shawn Simulator θα βρείτε εδώ: <http://www.ceid.upatras.gr/courses/katanemhmena/ds1/>

Installing Wiselib

Μεταβείτε στο φάκελο που κατεβάσατε την wiselib και μετονομάστε:

- το *wiselib.stable/std_config.h* σε *wiselib.stable/config.h*
- το *wiselib.testing/std_config_testing.h* σε *wiselib.testing/config_testing.h*

Μεταβείτε στο φάκελο *trunk/applications* και μετονομάστε:

- το *Makefile.template* σε *Makefile.local*
- στη συνέχεια πρέπει να ορίσετε στο *Makefile.local* τα paths
 - *WISELIB_BASE*
 - *WISELIB_PATH*
 - *WISELIB_PATH_TESTING*
 - *SHAWN_INCLUDE_DIR*
 - *SHAWN_LIB_DIR*

Εγκαταστήστε την βιβλιοθήκη Boost C++ μέσω του package manager του λειτουργικού σας.

Installing Wiselib

Μεταβείτε στο φάκελο *trunk/pc_apps/timer_test* και εκτελέστε στην κονσόλα τις ακόλουθες εντολές:

- *make*
- *./timer*

Μετά την εκτέλεση των παραπάνω εντολών θα πρέπει να δείτε το ακόλουθο μήνυμα: ***It's time!***

Shawn, Wiselib Integration

Αντιγράψτε το φάκελο *trunk/shawn_apps/wiselib* μέσα στο φάκελο του Shawn στο directory *shawn/src/legacyapps* .

Μεταβείτε από την κονσόλα στο directory *shawn/buildfiles* και εκτελέστε :

- `ccmake ../src`
- Θα πρέπει να θέσετε στην τιμή ON τα
 - `BUILD_LIB_WITH_MAIN`
 - `MODULE_APPS_EXTERNAL_APPLICATION`
 - `MODULE_LEGACYAPPS_WISELIB`
 - `LOG_DEBUG`
 - `LOG_INFO`
- Θα πρέπει επίσης να ορίσετε τα paths των
 - `INCLUDE_PATH_WISELIB_STABLE`
 - `INCLUDE_PATH_WISELIB_TESTING`
- στη συνέχεια δώστε 'c', 'g'.

Shawn, Wiselib Integration

Εκτελέστε την εντολή *make*.

Στην συνέχεια εκτελέστε την εντολή

`./shawn -f ../src/legacyapps/wiselib/start_wiselib_shawn.conf` για να ξεκινήσει η εκτέλεση του Wiselib Application.