

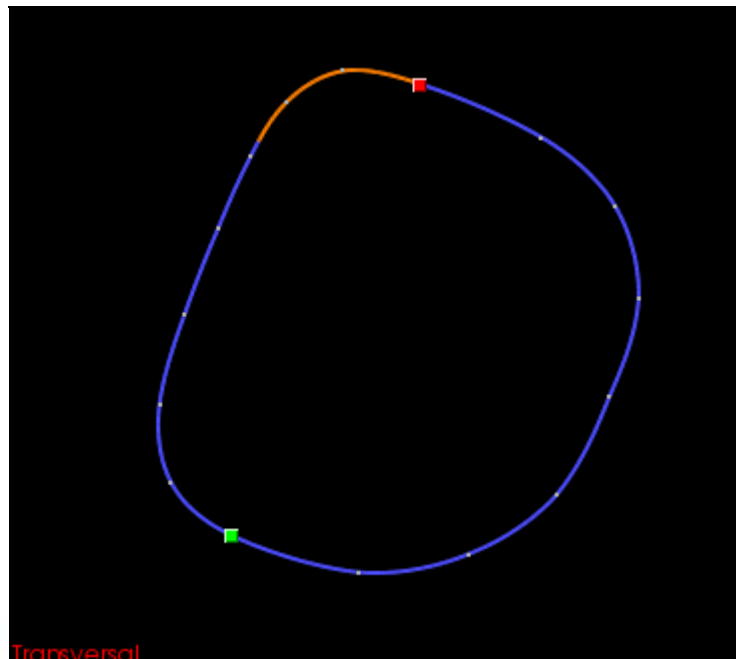
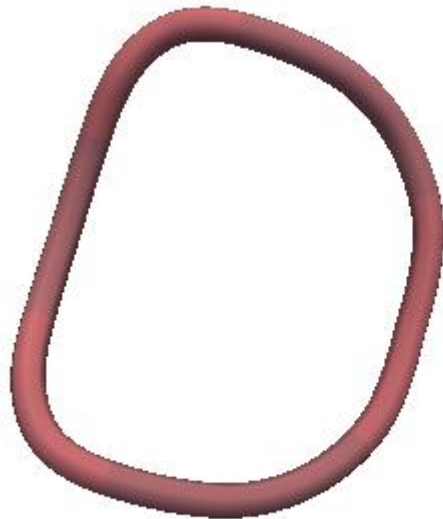
BugSquashing:

Create your own MITK Datatype (Coarse Overview)

Bastian Graser
7th March 2012

Why do I want my own MITK Datatype?

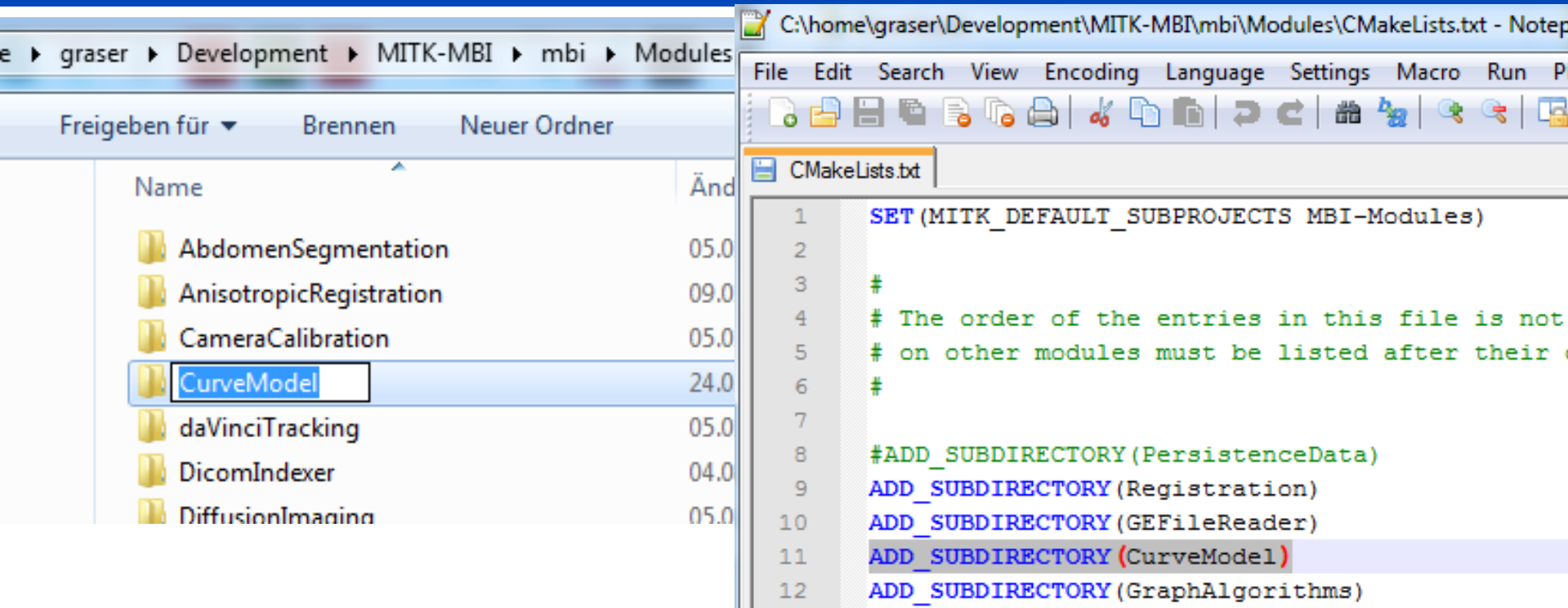
- E.g. For modelling a mitral ring!
 - Visual representation (2D and 3D)
 - Suitable fileformat
 - Intuitive interactions
- → **All that is arranged with your ObjectFactory**



First: Where do i put it?

- MITK / Core
 - Only super essential datatypes (e.g. PointSets)
- MITK / Modules
 - When you want to make it open source
- MITK-MBI / Modules
 - When you want to keep it closed source for the time being

Create a module for your Datatype



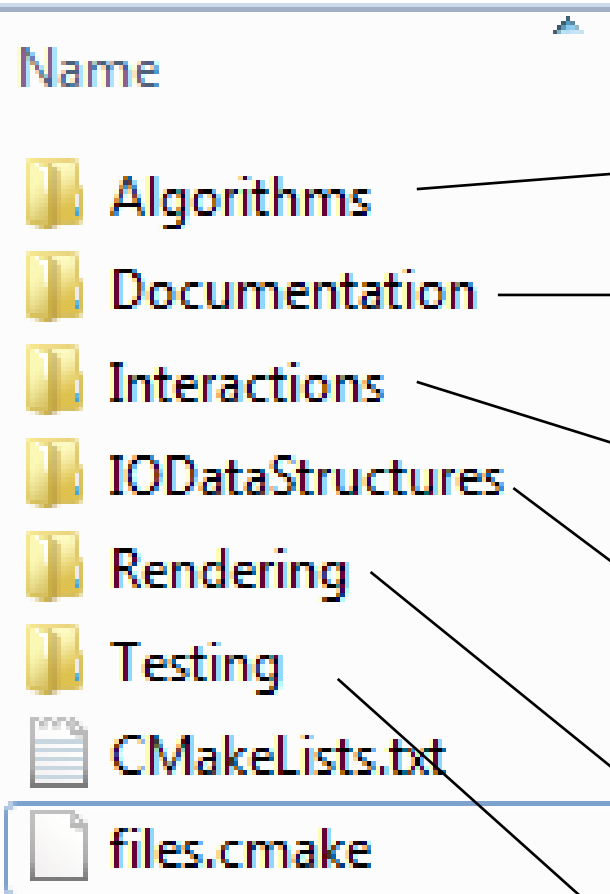
The screenshot shows a Windows Explorer window on the left and a Notepad window on the right. The Explorer window displays the directory structure: `graser > Development > MITK-MBI > mbi > Modules`. The 'CurveModel' folder is selected. The Notepad window shows the contents of `CMakeLists.txt` with the following code:

```
1 SET(MITK_DEFAULT_SUBPROJECTS MBI-Modules)
2
3 #
4 # The order of the entries in this file is not
5 # on other modules must be listed after their
6 #
7
8 #ADD_SUBDIRECTORY(PersistenceData)
9 ADD_SUBDIRECTORY(Registration)
10 ADD_SUBDIRECTORY(GEFileReader)
11 ADD_SUBDIRECTORY(CurveModel)
12 ADD_SUBDIRECTORY(GraphAlgorithms)
```

- Create your folder in `mbi\Modules`
- Add your folder to `mbi\Modules\CMakeLists.txt`

Create Folders and Cmake Files

- Create the following folders and your **CMakeLists.txt** and **files.cmake**



- Algorithms specific for this datatype

- Doxygen documentation

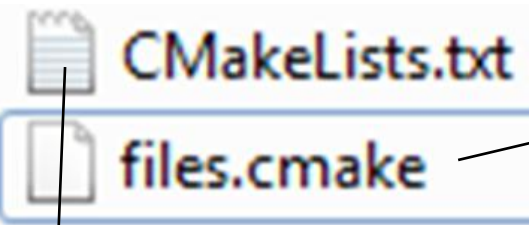
- Interactors

- DataStructure, Filereader/-Writer and ObjectFactory

- 2D and 3D Mapper

- Tests

Create Folders and Cmake Files

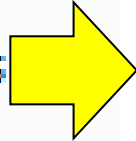


```
SET (CPP_FILES  
    IODataStructures/mitkCurve.cpp  
    ..  
)  
SET (H_FILES  
    IODataStructures/mitkCurve.h  
    ..  
)
```

```
# CREATE MODULE HERE  
MITK_CREATE_MODULE (MitkCurveModel  
    INCLUDE_DIRS IODataStructures Rendering Algorithms Interactions  
    DEPENDS MitkExt  
)  
ADD_SUBDIRECTORY (Testing)
```

Name

- Algorithms
- Documentation
- Interactions
- IODataStructures
- Rendering
- Testing
- CMakeLists.txt
- files.cmake

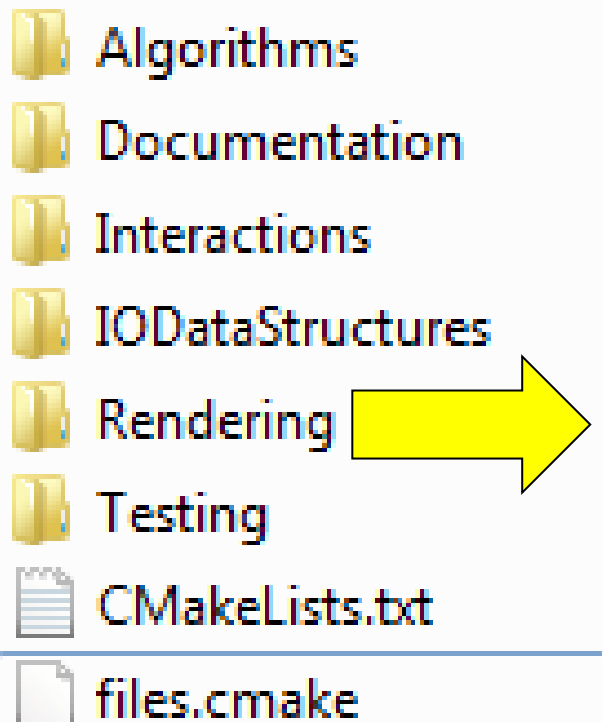


- **Datatype Class**
- Datatype Source (Superclass)
- **Datatype Writer**
- **Datatype Reader**
- DataTypeIOFactory
- DataTypeWriterFactory

Most important: The datastructure

- 2D Mapper
- 3D Mapper

Name



Algorithms

Documentation

Interactions

IODataStructures

Rendering

Testing

CMakeLists.txt

files.cmake

- To use your datatype:
 - Add DataType-Module Dependency to CMakeLists.txt
 - Call RegisterFunction.

E.g. in Constructor of Bundle View

```
#include "mitkCurveModelObjectFactory.h"
```

```
RegisterCurveModelObjectFactory();
```

- You can create your DataType-Module
- You need a class for:
 - Datatype, Reader, Writer, Renderer, ObjectFactory
 - Copy & Paste stuff from here:
 - **MITK / modules / DiffusionImaging**
 - **MITK-MBI / modules / CurveModel**
- To use it in a bundle/module, you have to call the RegisterFunction first