

# OpenMI goes Linux

OATC Meeting,  
April 2008 in Delft,  
Peter Schade ( [peter.schade@baw.de](mailto:peter.schade@baw.de) )



# Numerical Models at BAW-DH

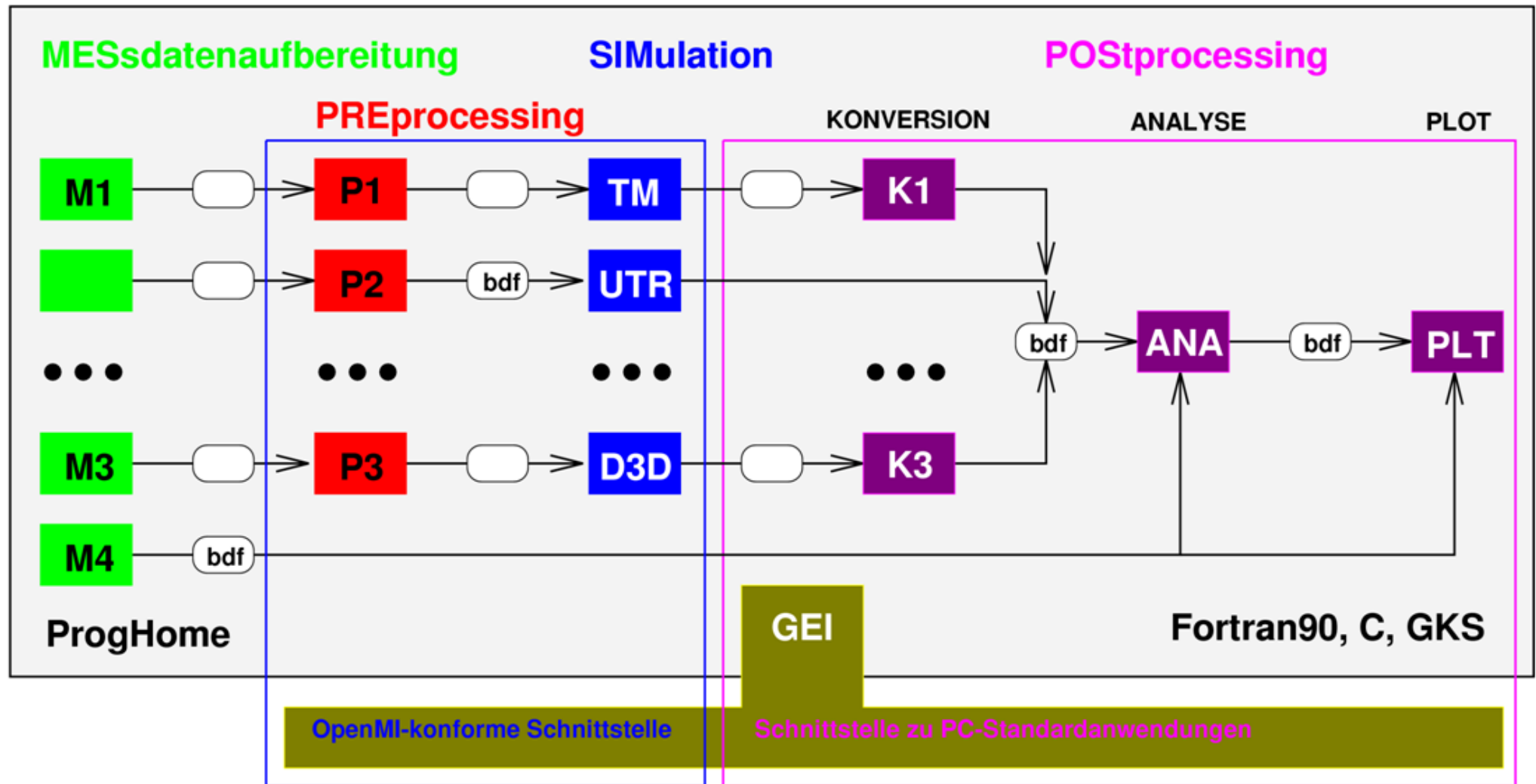
- UnTRIM : 2D and 3D finite difference method for unstructured orthogonal grids
- SediMorph : morphodynamic model which simulates the physical processes in the soil

- Delft3D : integrated modelling system Delft3D on Windows PCs
- MIKE-11 : 1D (cross-sectionally averaged) finite difference model
- TELEMAC-2D : 2D (depth averaged) finite element model
- TELEMAC-3D : 3D (depth averaged) finite element model
- SWAN : third-generation spectral wave model (Delft University of Technology)
- K-Model: a spectral wave model for unstructured orthogonal grids
  
- TRIM-2D : two-dimensional (depth averaged) finite difference model
- TRIM-3D : 3D finite difference model using the non-hydrostatic pressure

the majority of models runs on Unix / Linux systems



# Workflow at BAW-DH



# Model Hardware

Computer	CPU	OS	Comment
5 PCs	Dual Core	Windows	Reserved for Delft3D
SGI Altix 3700	Intel Itanium	64bit Linux (SLES 9 SP3)	Shared memory
SGI Altix XE 1300	66 * 2 Intel Xeon Quad Core	64 bit Suse Linux (SLES 10)	Cluster
Linux Workstation	Xeon Dual Core	32 / 64 Suse Linux (SLED10)	Test system for OpenMI

# Motivation for Linux / Mono

- Linux is a precondition for access to more numerical models.
- Especially at BAW: models on Linux systems run with a higher performance.
- Mono enables the user to run .NET on Linux, MacOS and even Microsoft Windows systems.
- Mono seems to be a mature project with an ongoing development.
- Linux and Mono are open source.



# Installed Tools on Linux WS\*

\* Workstation with SLED 10 and 64 bit support

- Mono 1.0
- Mono 2.0
- NUnit Console 1.0
- NUnit Console 2.0
- (NUnit GUI is missing)
- (the IDE MonoDevelop is missing)



# OpenMI goes Linux – Todo List 1

- selection of a Mono Compiler and a NUnit version
- which OpenMI version is suitable for Mono?
- compilation of GUI-less OpenMI-DLLs
- compilation of GEIWrapper (WLDelft Wrapper?)
- compilation of OpenMI-DLLs with GUIs
- test of the C#-part with NUnit
- generation of Fortran shared libraries, e.g. gei.xe.sl
- test of the gei.xe.sl
- test of a WLDelft Wrapper – GEIWrapper composition



# OpenMI goes Linux – Todo List 2

If the tests are successful, the Mono specific code will have to be integrated into the OpenMI main line.

“There are several approaches to porting code, depending on your goals:

- The unsupported code can simply be removed or commented out if it is not needed.
- Compiler conditional directives (#if) can be used to create separate executables for .Net and Mono.
- The runtime (.Net or Mono) can be detected and use different code.
- The code can be rewritten to use supported methods in Mono.”

from [http://www.mono-project.com/Guide:\\_Porting\\_Winforms\\_Applications](http://www.mono-project.com/Guide:_Porting_Winforms_Applications)



# Resumee

- “OpenMI goes Linux” is feasible
- project has started yesterday
- BAW needs another LC in order to run complete compositions
- hopefully small step for the developer ...

