New Kratos interface based on Customlib
Kratos GUI - Outline

- Introduction
- Initial requirements
- Tools available in GiD
- Problemtype structure
- Examples
- Future lines
Initial requirements

- TK: Simplify controls
- TCL: Script share & script isolation
- Easy tree configuration
- Easy “add new features”
- Modular: almost puzzle
- Open Source (visit our github page)
Tools available in GiD

Until GiD v13:
- Cnd, bas, hand made widgets

Since GiD v13:
- CustomLib
- Extra needs:
  - XML includes and Unique names

Since GiD v14:
- Smart Wizard package
Problemtype structure

- **Kratos common:**
  - Event handle
  - App coordination
  - Shared scripts

- **Application**
  - Define tree structure
  - Define tree content
  - Define write process
  - Definition vd inheritance
Problemtype structure

Kratos application market
Problemtype structure

Kratos application market
One application defines:

**Input**
- .spd files -> Tree structure
- .tcl file -> Load tree content & event handle
- .xml files -> Tree content

**Output**
- .tcl files -> Write events for:
  - Mdpa
  - ProjectParameters.json
Problemtype structure

Structure + Contents = Application tree
Example: Fluid dynamics
Example: Fluid dynamics
Example: Fluid dynamics
Example: Structural truss
Example: Solid water tank
Example: Stenosis wizard
Example: App interaction FSI
Future lines

- Wizard as a standalone product
- Transform in distributed files
- Follow the steps of Kratos Multiphysics team (>60 contributors in github)
- Implement stages
- PFEM multigroup conditions
- ...
THANKS