1. Home 2 Order Example

This example converts models of a house into purchase order of door required to build the house.

1.1.   Directory Organization

- 📁 **Home2Order**
  - ○ House2Order.mga   - The transformation file
  - ○ House2Order.xme   - transformation exported
  - ○ 📁 **Meta**
    - ▪ 📁 **Icons**   **-** Icons for the paradigms
    - ▪ HouseModel.mga   - House metamodel
    - ▪ HouseModel.xme   - House metamodel exported to XML
    - ▪ GS_HouseModel.xmp   - House paradigm file
    - ▪ Order.mga   - Order metamodel
    - ▪ Order.xme   - Order metamodel exported to XML
    - ▪ GS_Order.xmp   - Order paradigm file
  - ○ 📁 **Models**
    - ▪ myHouse1.mga   - model for HouseModel paradigm
    - ▪ myHouse1.xme   - Example model exported
  - ○ 📁 **Udm**   **-** Will contain the Udm meta files
  - ○ 📁 **Gen**   **-** Will contain the CG tool generated files

1.2.   How to run Home 2 Order example?

Open House 2 Order transformation model
  - ○ Directly open $/House2Order.mga, if it fails, open GME, choose File/Import XML, and choose $/ House2Order.xme

House2Order.mga contains the transformation rules, UDM compatible meta information paradigms and configuration information. Following is the folder structure which is shown in browser:

- ▪ 📁 House2Order
  - ○ 📁 GS_HouseModel   - Input Metamodel in UML class diagram format

  - ○ 📁 GS_Order   - Output Metamodel in UML class diagram format
  - ○ 📁 zt_House2Order   - Folder containing the transformations
  - ○ 📁 zz_Config   - Folder containing configuration information

Run the House 2 Order transformation model
  - ○ Invoke the GReAT Master Interpreter with icon 🏃 (**This is a required step for the first time running**). Use the default file paths and names provided.
  - ○ The transformations can be invoked in various ways

1. GR Engine – Performs the transformations in an interpretive manner
2. GR Debugger – Provides a user interface and debugging features such as break points, single step, step into etc.
3. Code generator – Converts the transformation into code that can be compiled and executed.

- o To run GR Engine, it could be done either :
  - In the GReAT Master Interpreter dialog, check the box "Run GR Engine"**-or-**
  - Directly invoke the GR Engine interpreter with icon .
  - The default input file is $/Models/MyHouse1.mga
  - The output files will be $/Models/MyOrder1.xml
- o To run the GR Debugger
  - Open a command prompt and go to the sample directory $/. Invoke GRD by calling GRD.exe , then load the config file $/config.mga **-or-**
  - Directly invoke the GR Debugger interpreter with icon .
- o To run Code Generator, it could be done either :
  - In the same dialog of GReAT Master Interpreter, check the box of "Run Code Generator"; **-or-**
  - Directly invoke the Code Generator interpreter with icon ; **-or-**
  - Open a command prompt and go to the sample directory $/. Invoke CG by calling CG.exe , with config file $/config.mga
  - After the files have been generated open the generated Visual Studio project using VS71 and compile the project
  - You can run the generated code with default arguments by setting the working directory to be ..\ and Program argument to be –d (default)