The REMES tool-chain consists of the following three tools:

(i) a REMES editor for modeling behaviors of embedded components,
(ii) a REMES simulator to test timing and resource behavior prior to formal analysis,
(iii) an automated transformation from REMES to Priced Timed Automata, needed for formal analysis.

**REMES Editor**
- GUI to create formal behavioral models of components
- Based on ECore Models + EMF/GMF
  - A composite mode consists of:
    - several submodes (atomic modes)
    - edges and conditional connectors that connect the submodes
    - init- and entry points through which the modes are entered
    - exit points through which the modes are exited
    - number of associated constants, variables, and resources displayed in separate compartments components.

**REMES Simulator**
- Model to model transformation to transform behaviors into source code that simulates the modeled system.
- Transformation performed over intermediate models created from REMES models.
- The main benefit of the simulator is its ability to track each resource as a separate variable.

**Transforming REMES to PTA**
- Transformation implemented by the model to model transformation language ATL (Atlas Transformation Language)
- The transformation rules applied to REMES diagrams result in UppaalLite models representing the same behavior.
- UppaalLite editor is a tool to visually inspect transformation results.
- Formal analysis:
  - Feasibility analysis
  - Optimal and worst-case resource consumption
  - Trade-off analysis