Agenda

- Part 1
  - Introduction to MDD for RT/E systems & MARTE in a nutshell
- Part 2
  - Non-functional properties modeling
  - Outline of the Value Specification Language (VSL)
- Part 3
  - The timing model
- Part 4
  - A component model for RT/E
- Part 5
  - Platform modeling
- Part 6
  - Repetitive structure modeling
- Part 7
  - Model-based analysis for RT/E
- Part 8
  - MARTE and AADL
- Part 9
  - Conclusions
Embedded System Hardware is now Repetitive

- **Multicore**
  - Today 4 to 8 cores
  - Tomorrow: 16 to 64 cores

- **Processor meshes**
  - Ex: Tilera Tile64

- **SIMD units**
  - Data parallelism
The Future of Embedded Applications is **Parallel**

- **Multimedia**
  - Video coding/decoding
  - HDTV

- **Detection systems**
  - Radar
  - Sonar

- **Telecom**
  - Software radio
  - Wireless communications

**Computation models**
- Multidimensional signal processing
- Stream processing
- Data parallelism
Repetitive Structure Modeling

- **Motivation**
  - *Multidimensional regular parallelism*
    - Nested loops
    - Multiprocessor Systems
  - *Compact representation*
    - Application
    - Hardware platform
    - Association

- **Form**
  - New notation / stereotypes
Concepts of Repetitive Structure Modeling

- **Concepts**
  - **Shape** (extension of *multiplicity*)
    - To model multidimensional arrays
  - **Link topology** (extension of *connector* and *allocate*)
    - To model the topology of the links between multidimensional arrays
    - Pattern-based regular topologies

- **Basic idea: regular tiling of multidimensional arrays by multidimensional sub-arrays**
  - Regular spacing of points inside a tile
  - Regular spacing of tiles
  - Inherits from the Array-OL language
Shape Modeling

- **New notation**
  - Refinement of the multiplicity notation
  - Vector of UnlimitedNaturals

- **Examples**
  - $16 \rightarrow \{4,4\}$
  - $* \rightarrow \{512,128,*\}$
### Link Topology Modeling

**Profile**

- **RSM**

**Metaclasses**

- **UML::Connector**
- **UML::ConnectorEnd**

**Stereotypes**

- **DefaultLink**
- **LinkTopology**
- **InterRepetition**

**Stereotype Attributes**

- **RepetitionSpaceDependence**: `IntegerVector [1]`
- **isModulo**: `Boolean = false`

- **PatternShape**: `ShapeSpecification [1]`
- **RepetitionSpace**: `ShapeSpecification [1]`

**Stereotype Descriptions**

- **Reshape**
- **Tiler**

**Attributes**

- **origin**: `IntegerVector`
- **paving**: `IntegerMatrix`
- **fitting**: `IntegerMatrix`
- **tiler**: `TilerSpecification`
**Hardware Platform Example**

- **SIMD unit**
  - 16 processors

- **Topology**
  - Toroidal 4×4 grid
  - Bidirectional connections
    - North-South
    - East-West
Tiling an Array

- **Needed shapes**
  - Array shape
  - Pattern shape
  - Repetition space shape

- **Tiler**
  - **Fitting**: regular spacing of the points of the tiles
    - Index $i$
    - Scanning the pattern
  - **Paving**: regular spacing of the tiles
    - Index $r$
    - Scanning the repetition space

```
origin+(paving fitting).\( (r) \mod \text{array.shape} \)
```
Graphical Interpretation of a Tiler (1/2)

- **Fitting**
  - Column vectors
  - Basis of the tile
  - Pattern shape
  - Bounds of the fitting

- **Paving**
  - Column vectors
  - Basis of the placement of the tiles
  - Repetition space
  - Bounds of the paving
  - Origin
  - Coordinates of the reference point of the reference tile

\[
F = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \text{pattern } s_{\text{pattern}} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \quad \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \text{array } s_{\text{array}} = \begin{pmatrix} 6 \\ 6 \end{pmatrix}, \quad P = \begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix}, \quad \text{repetition } s_{\text{repetition}} = \begin{pmatrix} 3 \\ 2 \end{pmatrix}
\]
Graphical Interpretation of a Tiler (2/2)

- **Fitting**
  - Column vectors
    - Basis of the tile
  - Pattern shape
    - Bounds of the fitting

- **Paving**
  - Column vectors
    - Basis of the placement of the tiles
  - Repetition space
    - Bounds of the paving
  - Origin
    - Coordinates of the reference point of the reference tile

\[
F = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad o = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad P = \begin{pmatrix} 1 & 0 \\ 0 & 3 \end{pmatrix}
\]

\[s_{\text{pattern}} = \begin{pmatrix} 3 \\ 2 \end{pmatrix}, \quad s_{\text{array}} = \begin{pmatrix} 4 \\ 6 \end{pmatrix}, \quad s_{\text{repetition}} = \begin{pmatrix} 4 \\ 2 \end{pmatrix}\]
Application Example

- **Samples from 512 hydrophones around a submarine**
  - Shape of the input data = $512 \times \infty$
- **Repetition of FFTs**
  - For each hydrophone
  - Sliding window of 128 samples every 32 time steps
Distribution

- Refinement of Allocation
- Similar to the reshape stereotype of the connectors

**Principle**
- Tiling both ends
  - Two tilers
- With the same tiles
  - One pattern shape
  - One repetition space

**Power of expression**
- At least all HPF data distributions
Distribution Example

- Distribution of the FFT computations to the SIMD unit
  - No spatial distribution of the infinite dimension (time steps)
  - Bloc distribution of the 512 FFTs for each time step
    - Size of the bloc = 32
    - On the 16 elementary processors
**Distribution Example**

- **Distribution of the FFT computations to the SIMD unit**
  - No spatial distribution of the infinite dimension (time steps)
  - Bloc distribution of the 512 FFTs for each time step
    - Size of the bloc = 32
    - On the 16 elementary processors
Complex Hardware Example: Tile64

- **Challenge**
  - Model the architecture
  - In the most compact way

- **Proposal**
  - 8x8-repetition of the processing element
  - 4-repetition of the DDR2 controller
  - Factorization of the ports
Processing Element Repetition

<<DefaultLink>>
<<Tiler>>
fitting = "{{0,0},{1,0}}",
origin = "{{0,0},{0,0}}",
paving = "{{0,0},{0,0}}"

<<DefaultLink>>
<<Tiler>>
fitting = "{{0,0},{1,0}}",
origin = "{{0,0},{0,0}}",
paving = "{{0,0},{0,0}}"

<<DefaultLink>>
<<Tiler>>
fitting = "{{0,0},{1,0}}",
origin = "{{0,0},{0,0}}",
paving = "{{0,0},{0,0}}"

<<DefaultLink>>
<<Tiler>>
fitting = "{{0,0},{1,0}}",
origin = "{{0,0},{0,0}}",
paving = "{{0,0},{0,0}}"

<<DefaultLink>>
<<Tiler>>
fitting = "{{0,0},{1,0}}",
origin = "{{0,0},{0,0}}",
paving = "{{0,0},{0,0}}"

<<DefaultLink>>
<<Tiler>>
fitting = "{{0,0},{1,0}}",
origin = "{{0,0},{0,0}}",
paving = "{{0,0},{0,0}}"

<<DefaultLink>>
<<Tiler>>
fitting = "{{0,0},{1,0}}",
origin = "{{0,0},{0,0}}",
paving = "{{0,0},{0,0}}"

<<DefaultLink>>
<<Tiler>>
fitting = "{{0,0},{1,0}}",
origin = "{{0,0},{0,0}}",
paving = "{{0,0},{0,0}}"

<<DefaultLink>>
<<Tiler>>
fitting = "{{0,0},{1,0}}",
origin = "{{0,0},{0,0}}",
paving = "{{0,0},{0,0}}"

<<DefaultLink>>
<<Tiler>>
fitting = "{{0,0},{1,0}}",
origin = "{{0,0},{0,0}}",
paving = "{{0,0},{0,0}}"

<<DefaultLink>>
<<Tiler>>
fitting = "{{0,0},{1,0}}",
origin = "{{0,0},{0,0}}",
paving = "{{0,0},{0,0}}"

<<DefaultLink>>
<<Tiler>>
fitting = "{{0,0},{1,0}}",
origin = "{{0,0},{0,0}}",
paving = "{{0,0},{0,0}}"
DDR2 Controller Connection to the Grid
Conclusion on RSM

- General mechanism to handle
  - Multidimensional structures (arrays)
  - Tiling by sub-structures (non orthogonal or sparse tiles possible)
  - Links between such structures (cyclic or non cyclic connection patterns possible)

- Necessary to handle massive regular parallelism
  - Compactness of the model
  - Efficiency, maintainability, readability

- Relations with the rest of MARTE
  - Uses VSL
  - Benefits from the component model (flow ports)
  - Applies to both application and hardware components
  - Extends allocation

- Limitations
  - Handles only arrays (no fancier shapes)
  - Would benefit from a custom (visual) tiler editor
    - Under development