Beyond ‘look & feel‘ virtualisation

Wolfgang Pree

Software Research Center
cs.uni-salzburg.at/SRC
Contents

- state-of-the-art: virtual somethings
- what does it take to go beyond?
  - sample domain: automation systems
Virtual ‘somethings’
virtual buildings

Walt Disney Concert Hall in Los Angeles

virtual

reality
beyond virtual buildings

- for conceptual purposes only (virtual walk-throughs etc.)
- play no role in the detailed design and construction of the finished structure
beyond virtual buildings

- for conceptual purposes only (virtual walk-throughs etc.)
- play no role in the detailed design and construction of the finished structure
- vision: **represent a building as a full-fledged model** that includes information about the relationships between these objects, so that when one object is changed (e.g., a window is made bigger) any related objects are automatically updated (the wall surrounding it gets thicker).
other domains

- more advanced models than in the construction domain exist already in the automotive and avionics domain
- AUTOSAR might be considered as first step to describe a car’s hardware and software infrastructure
- so far no appropriate models exist for automation systems
What does it take?
representation of domain knowledge

- domain-specific component model
  - how generic or specific?
  - hierarchical composition
  - versioning
  - persistence
  - navigation
  - etc.
- visual representation(s)
Physical aspects of a component

- **Physical View** represents physical and if applicable electrical aspects such as plugs and wires.
Functional View: Represents functional aspects, such as PID controllers and limit monitors. This view is similar to dataflow modeling languages such as Simulink.

Parameter View: Represents variability aspects in terms of name/value pairs, e.g. plug shape descriptions or PID controller values.
## sample component definitions

<table>
<thead>
<tr>
<th>Category: Engine</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Component</strong></td>
</tr>
<tr>
<td>E₁</td>
</tr>
<tr>
<td>E₂</td>
</tr>
</tbody>
</table>
reuse of components by adding properties

<table>
<thead>
<tr>
<th>Category: Engine</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Component</td>
<td>Cylinders</td>
<td>Inertia</td>
<td>$N_{\text{max}}$</td>
<td>Ignition</td>
</tr>
<tr>
<td>$E_1$</td>
<td>8</td>
<td>1.06 kgm$^2$</td>
<td></td>
<td>Plug 15</td>
</tr>
<tr>
<td>$E_3 \leftarrow E_1$</td>
<td>8</td>
<td>1.05 kgm$^2$</td>
<td>12,000 rpm</td>
<td>Plug 15</td>
</tr>
</tbody>
</table>

- Engine $E_3$ is defined by copying the definition of engine $E_1$
- the properties $\text{Cylinders}$ and $\text{Ignition}$ are inherited and their values are unchanged,
- the property $\text{Inertia}$ is inherited but its value was changed, and
- a new property $N_{\text{max}}$ is added
compositions, locations and relationships

COMPONENT SampleAutomationSystem

COMPONENTS
  Dyno : APADyno127
  IO: someIO
  CompNode: autPC

END

LOCATIONS
  TestCell
  ControlRoom

END

RELATIONS
  Dyno.BendingBeam.Plug CONNECTS someIO.X26
  Dyno AT TestCell
  ...

END

END