Rapid GUI Development on Legacy Systems

-- A Runtime Model-Based Solution

Hui Song, Michael Gallagher, Siobhan Clarke

Lero: The Irish Software Engineering Research Centre
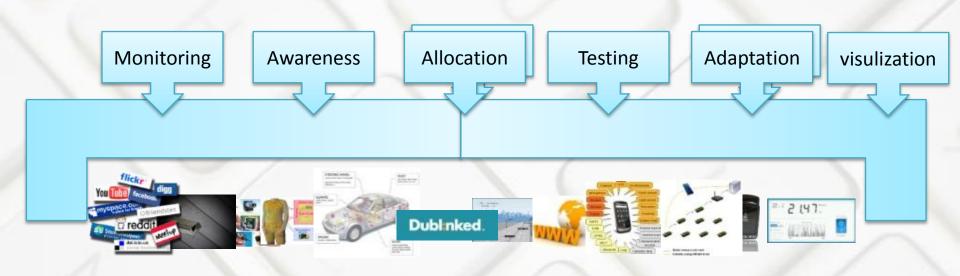Future City Group, SCSS, Trinity College Dublin
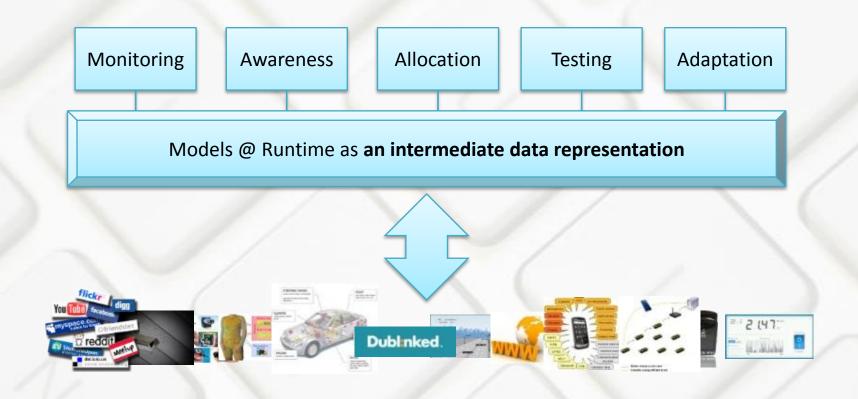
Diversity of data-usage based on software

| Monitoring | Awareness | Allocation | Testing | Adaptation | visulization |

Plenty of real-time real-world (legacy) data source
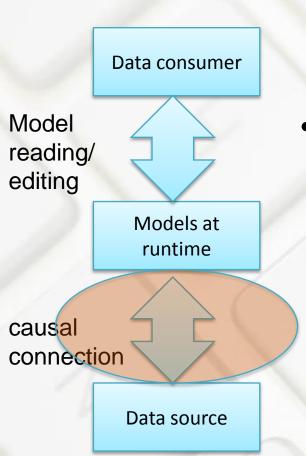
MRT for Data Processing

| Monitoring | Awareness | Allocation | Testing | Adaptation |

Models @ Runtime as **an intermediate data representation**

Data consumer

Model reading/ editing

Models at runtime

causal connection
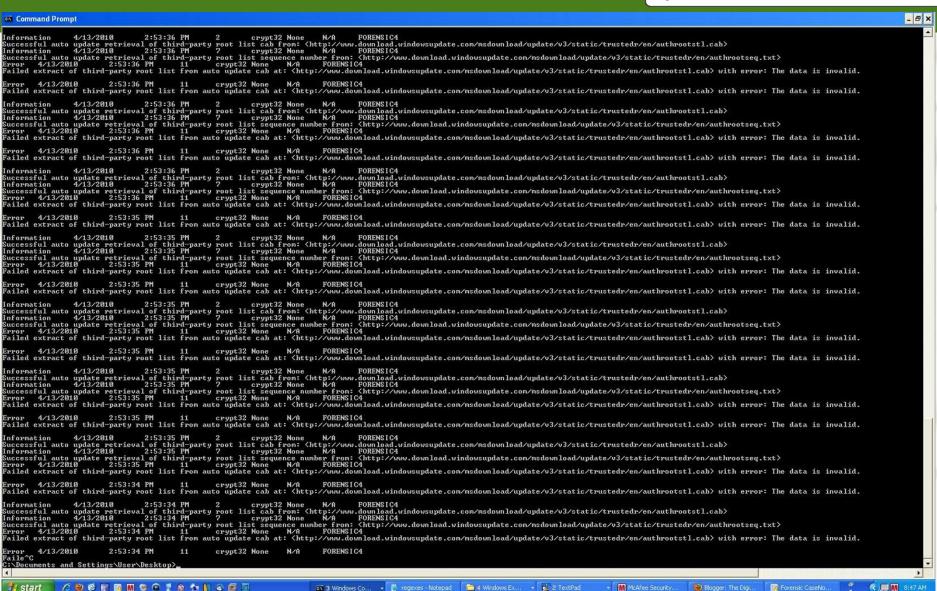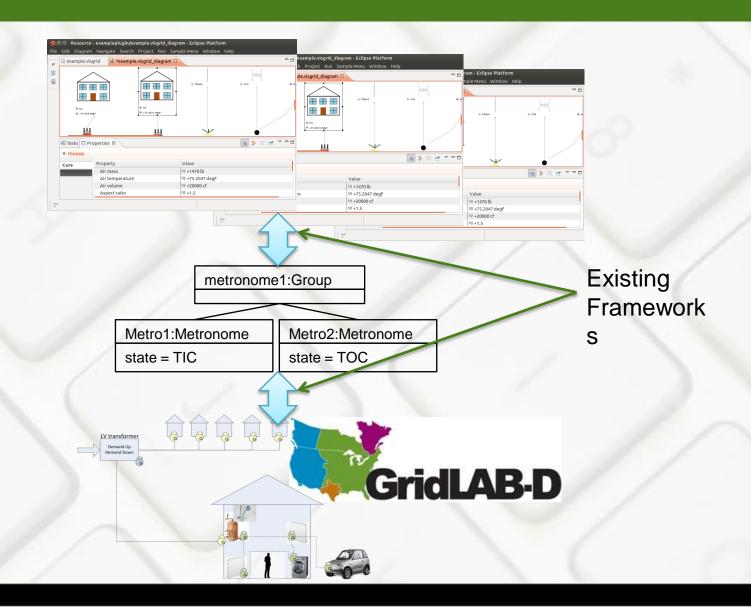
Data source

- Advantage
  - Separation of concerns
  - Self-descriptive: clear meta-models
- Difference
  - Modeling data instances, not types
    - *Directly reading and editing*
    - *Causally connected*
  - Abstraction of data, not software
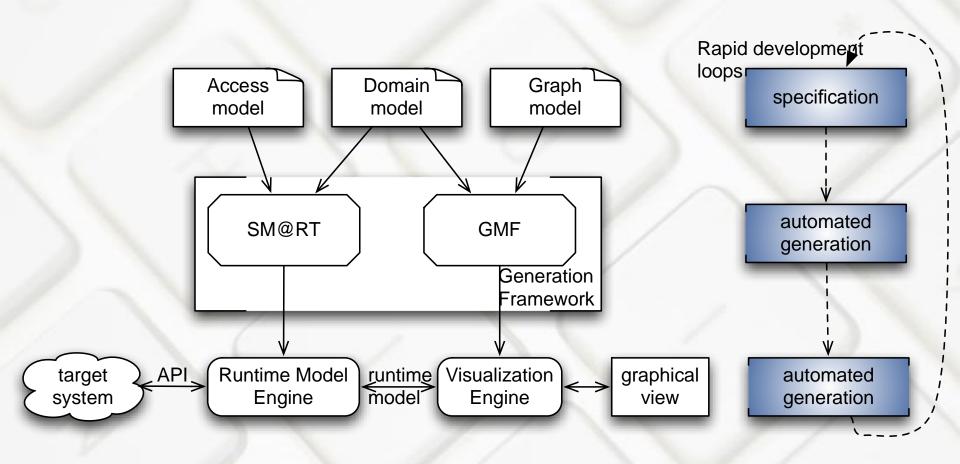   - *Data elements, not software class hierarchy*

Existing Frameworks

metronome1:Group

Metro1:Metronome
state = TIC

Metro2:Metronome
state = TOC

Meta-Modelling

automated generation

System ↔ engine ↔

- Non-Intrusive
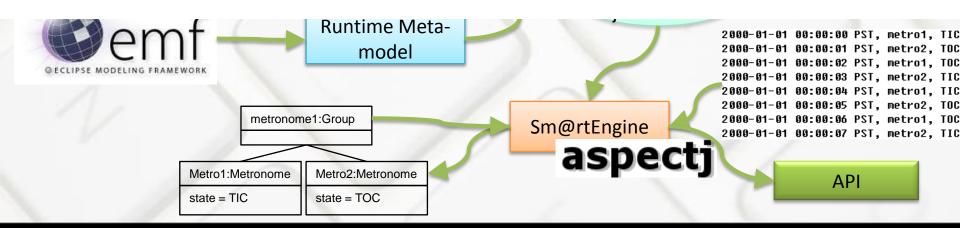- MOF/EMF
- DSL+Code generation

Xtext
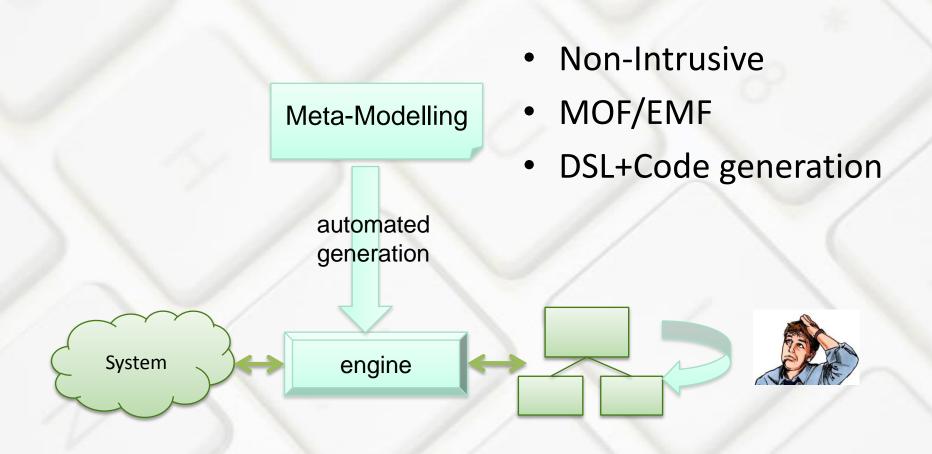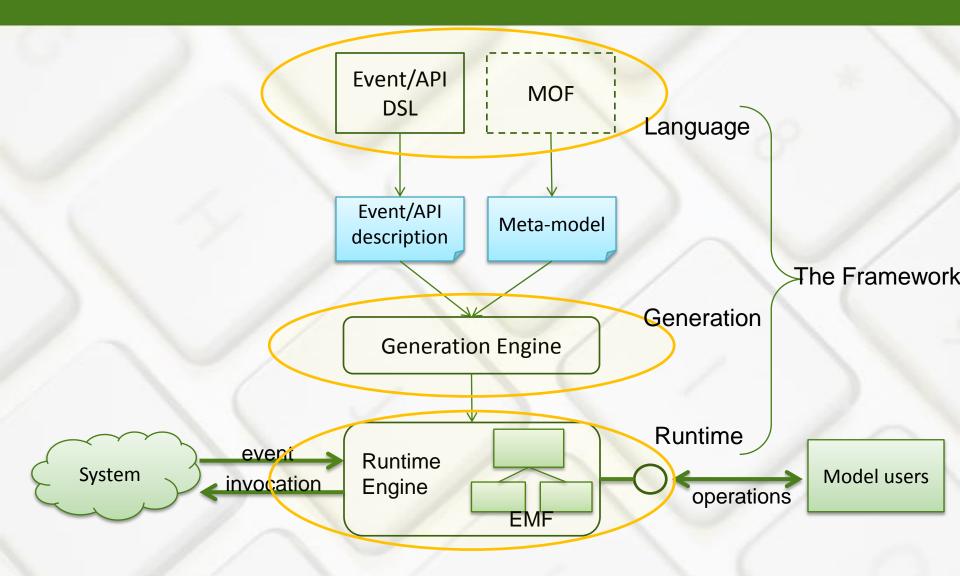
Wednesday, October 3, 13:30 - 13:40

Spotlight Presentations

Casino Foyer: Hui Song, Gang Huang, Yihan Wu: SM@RT: Supporting Models at Run-Time (Key Lab of High Confidence Software Technologies, Peking University, China)
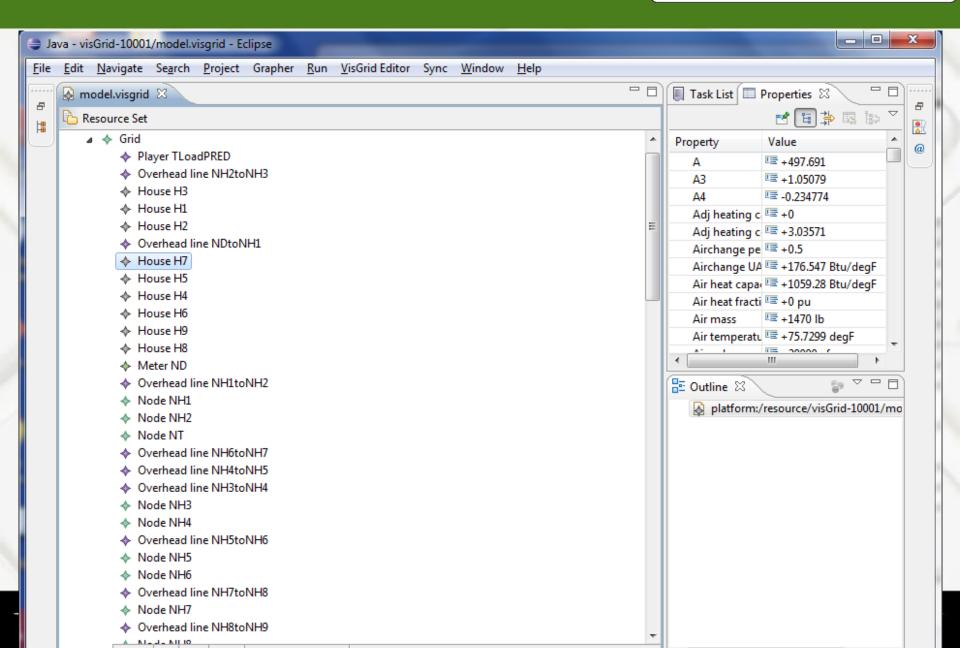
emf
ECLIPSE MODELING FRAMEWORK

Runtime Meta-model

```
2000-01-01 00:00:00 PST, metro1, TIC
2000-01-01 00:00:01 PST, metro2, TOC
2000-01-01 00:00:02 PST, metro1, TOC
2000-01-01 00:00:03 PST, metro2, TIC
2000-01-01 00:00:04 PST, metro1, TIC
2000-01-01 00:00:05 PST, metro2, TOC
2000-01-01 00:00:06 PST, metro1, TOC
2000-01-01 00:00:07 PST, metro2, TIC
```

metronome1:Group

Metro1:Metronome
state = TIC

Metro2:Metronome
state = TOC

Sm@rtEngine

aspectj

API

```
invocation GetState{
  operation get SimuElement->* {
    feature.EType.name = 'Double'
  }
  invoke returning String {
    val url = new URL("http://localhost:10001/" +
      element.name + "/" + feature.name)
    connection.setRequestMethod("GET");
    val in = ... //get a BufferedReader
    var s : String
    while ((s = in.readLine()) != null)
      if(s.contains('value'))
        return value //skip string parsing
  }
  post (Double d) {current.eSet(feature,d) }
}
```
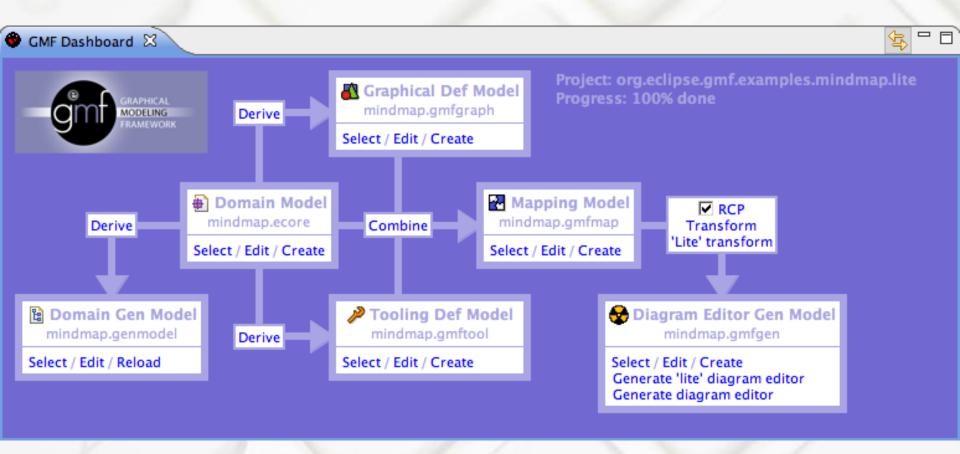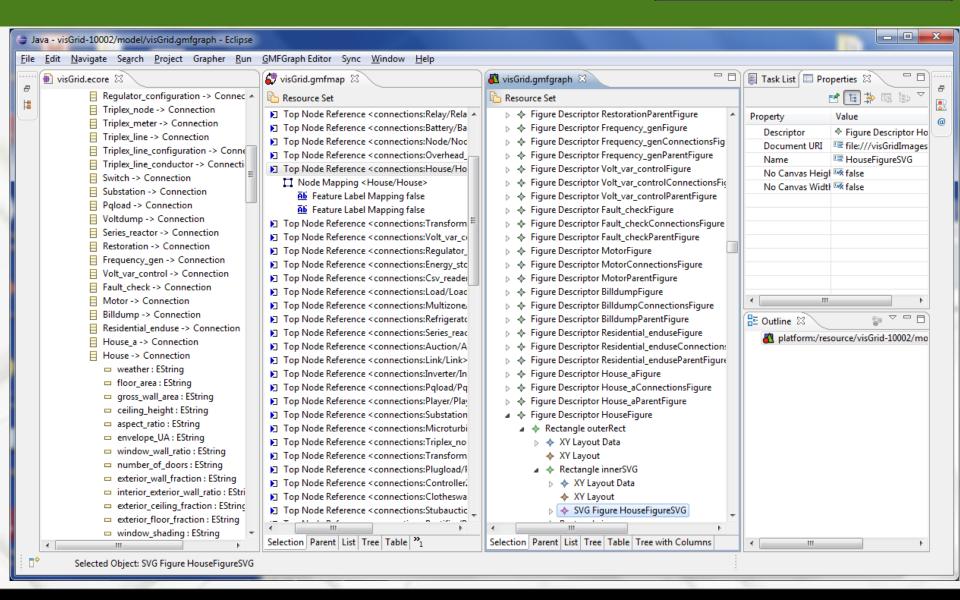
Result runtime model

Model Visualization via GMF
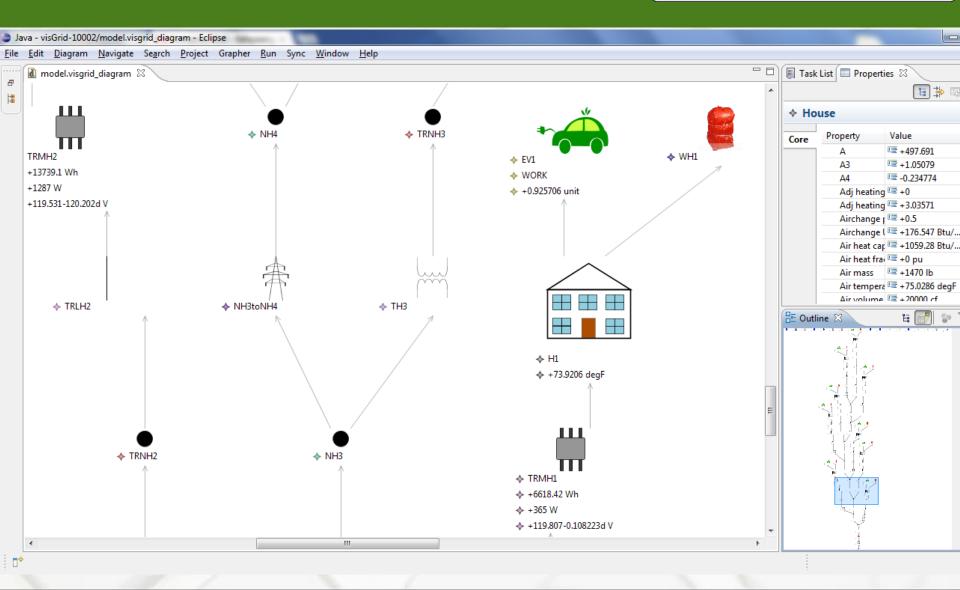
Model Visualization via GMF

Development process

Trunk-version GUI	
Scale	80 classes, 1731 attributes, 146 max for one class
Sample	78 objects, 137 on-screen attributes,
Performance	4s initialization, 2s interval in 20% CUP, 1.9s max for an object
Development process	
Coding	1 intern, 6 weeks, no experience on SM@RT or GMF
Others	1 person on SM@RT and GMF, 2 persons to clarify requirements
Vertical versions	One loop every 1 to 4 days, with an executable version
Horizontal versions	4 versions used for different users and scenarios

Conclusion

- Runtime model as an intermediate representation of data

- "World model" at runtime for data rather than software

- Runtime model for (just) visualization

- SM@RT to support such runtime models

Future work

- On GridLAB-D: New usage of runtime model
  - Self-adaptation with end-user preference
  - Runtime-model + constraint-solving + dynamic ranking
- On GUI development
  - Connect runtime model with real GUI frameworks
    - *Android, SWT… Discussing with Olivier Moises on Wazaabi*
- On data processing
  - On real sensor-based data and system logs
- On the multi-views of data
  - jQVT (compiled QVT engine) for live transformation

Thank You

Debugging

Google

AttributeNotFoundException

Search

Page 13 of about 636,000 results (0.21 seconds)

Web

Images

Maps

Videos

News

Shopping

More

Getting Instance of MBean service in JBOSS AS 7.1.1 - Stack Overflow
stackoverflow.com/.../getting-instance-of-mbean-service-in-jb...
3 answers - 12 Jun
AttributeNotFoundException: No such attribute: Instance at com.sun.jmx.
mbeanserver.PerInterface.getAttribute(PerInterface.java:63) at ...

com.sun.jmx.mbeanserver: public class: Introspector
www.docjar.com › openjdk-7
getReadMethod(clazz, element); } if (readMethod != null) return readMethod.
invoke(complex); throw new **AttributeNotFoundException**("Could not find the getter ...

Using summoning snippets for debug

- Target bugs:
    - Illegal use of generic API on specific data
    - Cause: did not know the illegal invoke might happen
    - Find such bugs: enumerate all the possible paths
- Summoning snippet for debugging
    - Comparing the summoning snippets with the data meta-model

Source code evolution

- Scenario: programming from examples/test cases
  - A common programming practice
  - No so "pragmatic" for big code on specific data through generic APIs
  - Provide summoning snippets as an intuitive assistant
- Things to do
  - IDE integration
  - Traceability
  - Refactoring primitives
    - *Copy, move, merge, split…*

IDE integration