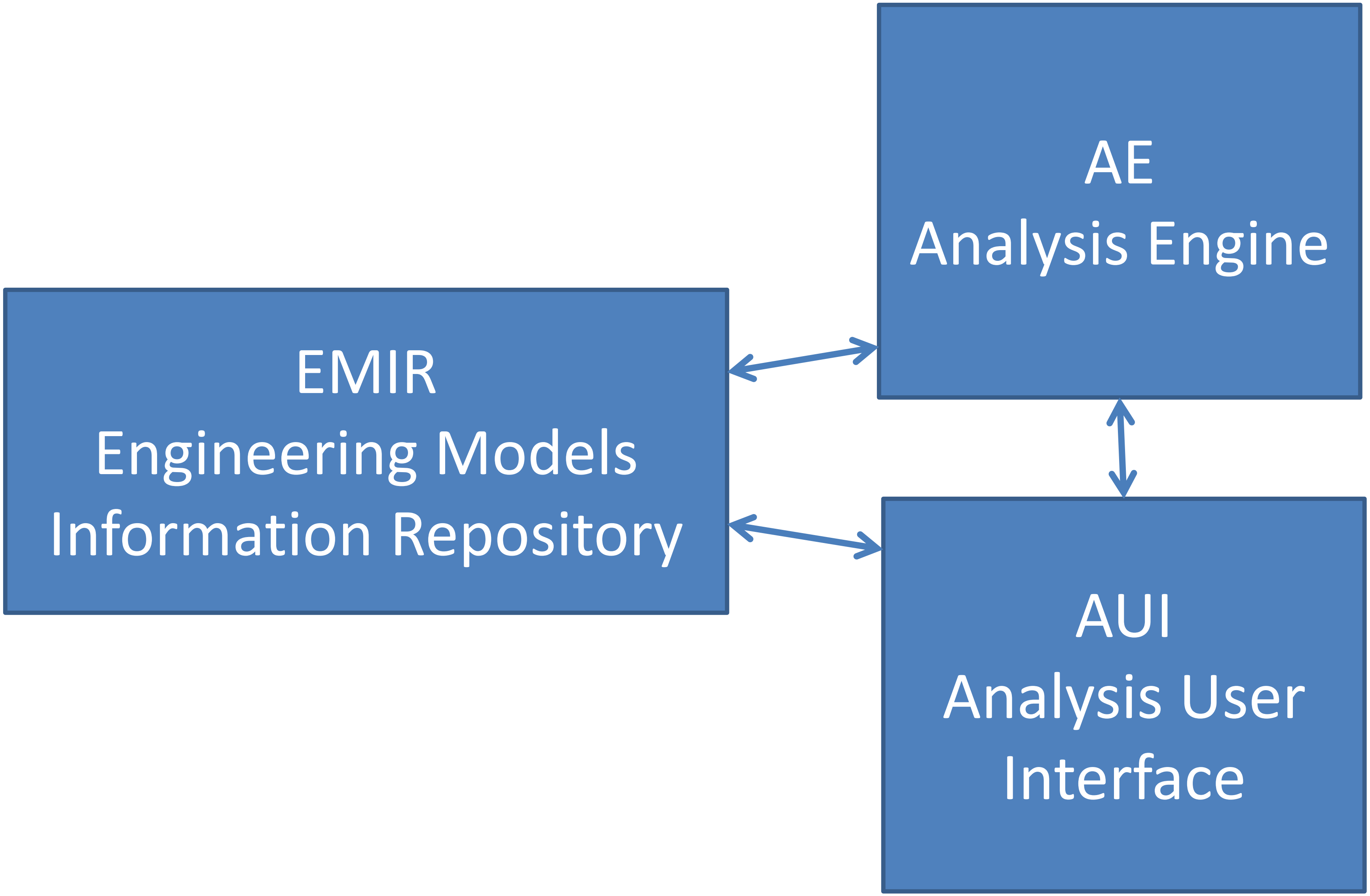
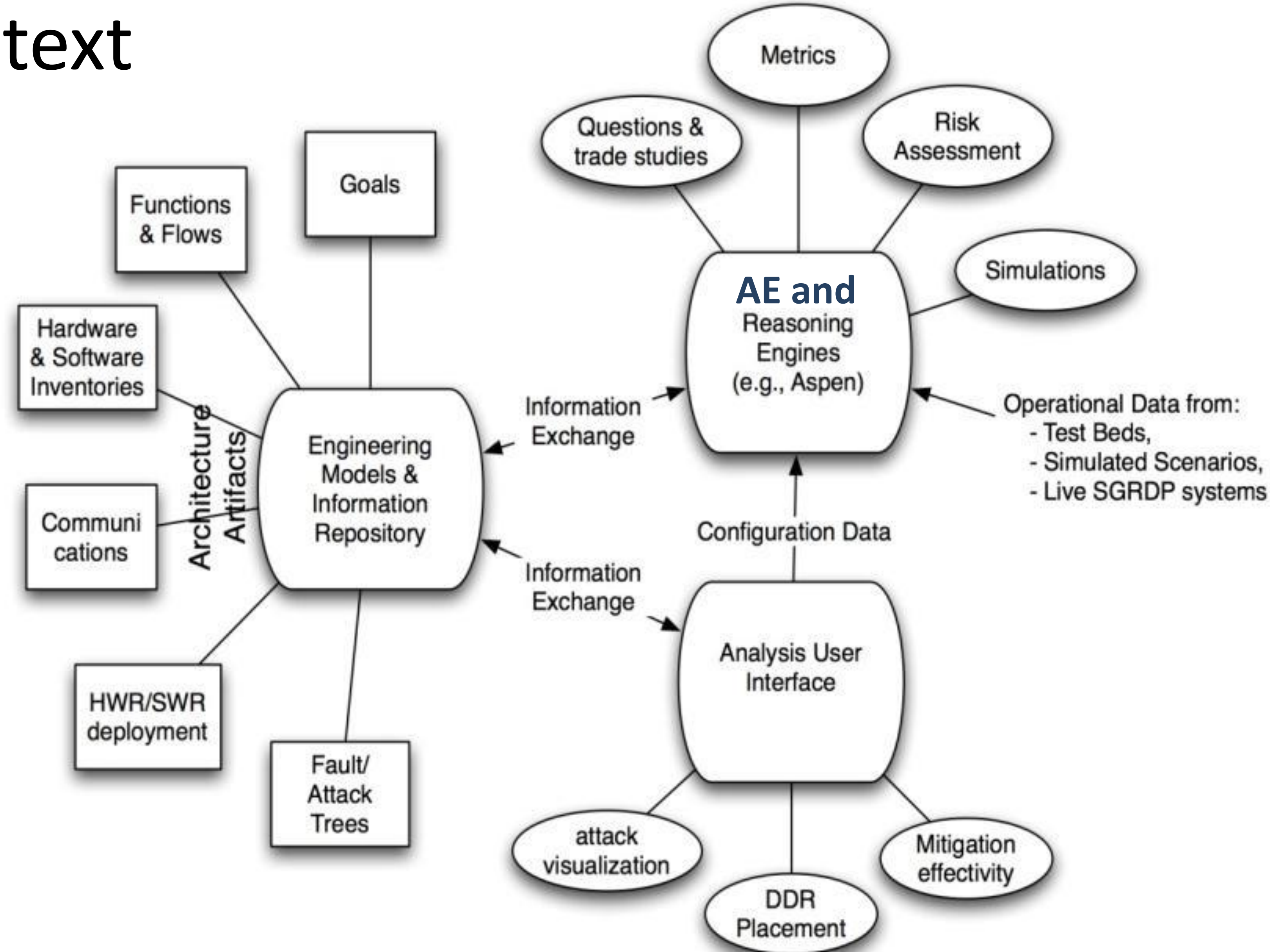
The Behavior Analysis Engine

Context

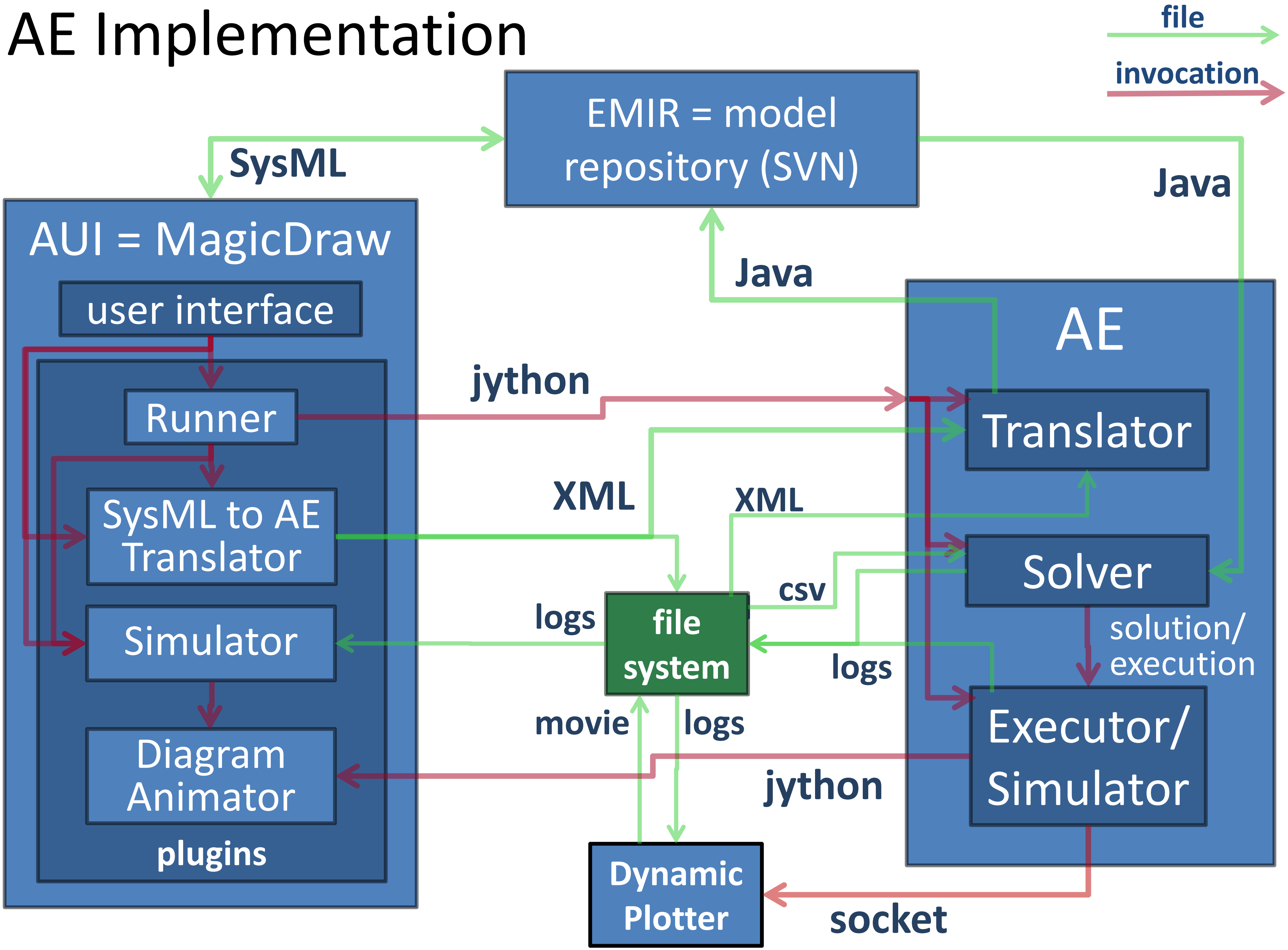

AE
Analysis Engine
EMIR
Engineering Models
Information Repository
AUI
Analysis User
Interface

Context



Metrics
Questions & trade studies
Risk Assessment
Simulations
AE and Reasoning Engines (e.g., Aspen)
Goals
Functions & Flows
Hardware & Software Inventories
Architecture Artifacts
Engineering Models & Information Repository
Information Exchange
Operational Data from:
- Test Beds,
- Simulated Scenarios,
- Live SGRDP systems
Configuration Data
Communi cations
Information Exchange
Analysis User Interface
HWR/SWR deployment
Fault/ Attack Trees
attack visualization
DDR Placement
Mitigation effectivity

AE Implementation



Comparing what we have and what we planned with related work

	Features/Criteria	language expressivity								modeling					
Approaches		time	continuous variables	continuous change	specifies problems	non-linear problems	object-oriented	high-fidelity	uncertainty	visual	graphic representations	multi-view	multi-view of same model	modeling and analysis integrated	application specific support
CS Modeling	**MagicDraw and plugins**	ordering and simple DE	**yes**	no	no	no	**yes**	no	no	**yes**	UML/SysML	**yes**	**yes**	**yes**	no
	Current AE + MagicDraw	**supports integrated DE and CT with temporal constraints**	**yes**	user must code	**yes**	**yes**	**yes**	**yes**	statistical sampling	**yes**	activity diagrams	**yes**	**yes**	**yes**	no
	CS modeling as designed: EMIR/SysML + AE + AUI/MagicDraw	**supports integrated DE and CT with temporal constraints**	**yes**	**yes**	**yes**	**yes**	**yes**	**yes**	**uncertain variables (including time) for probabilistic analysis**	**yes**	UML/SysML	**yes**	**yes**	**yes**	smart grid
System modeling & analysis tools	**summary**	DE and CT support	**yes**	**yes**	**yes**	**yes**	some	**yes**	statistical sampling	**yes**	**open**	user must code	no	**yes**	some
	Phoenix Technologies ModelCenter + MagicDraw + solvers	?	**yes**	?	**yes**	**yes**	?	?	?	**yes**	UML/SysML	user must code	**yes, via SysML**	**yes**	no
	Wolfram SystemModeler (Mathematica + Modelica + visual modeling)	DE and CT support	**yes**	**yes**	**yes**	**yes**	**yes**	**yes**	statistical sampling	**yes**	**open**	user must code	no	**yes**	partial support for many
	Ptolemy II + PTIDES	DE and CT support	**yes**	**yes**	in some cases?	?	**yes**	**yes**	statistical sampling?	**yes**	**open**	user must code	no	**yes**	different applications?
	AADL	?	?	?	?	?	?	?	?	?	graph	?	?	?	?
Simulators		DE and CT support	**yes**	**yes**	no	no	some	yes	statistical sampling	**yes**	**open**	user must code	no	no	**yes, many applications for some**
Govt/industry analysis tools	**e.g., ASPEN, EUROPA, SIPE, SHINE, BEAM, SCL**	**supports DE with temporal constraints and limited CT**	some	some	**yes**	rare	rare	no	limited	some	varied but limited	no	N/A	**yes**	somewhat
COTS security tools	**anti-virus, snort, host-based, penetration testing**	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Research	**modeling, analysis**	**some support DE with temporal constraints and some also support CT**	some	some	**yes**	some	few	few	some	few	none or varied	no	no	**yes**	**yes, usually for one application**
Problem solvers	**MILP (Mixed Integer Linear Problem) solvers like CPLEX**	no	**yes**	no	**yes**	**yes**	no?	no	user must encode	no	constraint equations	N/A	N/A	N/A	no?
	Learning tools (e.g., Weka)	no	**yes**	no	**yes**	**yes**	no	N/A	**implicit**	?	?	?	?	**yes**	?

	Features/Criteria	analysis					visualization			execution	adaptability/learning	V&V	distributed	scalable					
Approaches		simulation	detection	diagnosis	remediation	other	plotting	analysis	animation					fidelity	model size	time horizon	memory	responsiveness	for analysis
CS Modeling	**MagicDraw and plugins**	**yes**	no	no	no	?	ok	no	ok	**yes**	no	no	no	no	**yes**	**yes**	**yes**	not bad	no
	Current AE + MagicDraw	**yes**	user must code	user must code	yes, but simple	user must code	ok	no	ok	yes, but without feedback	no	verification	no	**yes**	**yes**	not bad	not bad	not bad	no
	CS modeling as designed: EMIR/SysML + AE + AUI/MagicDraw	**yes**	**yes**	**yes**	**yes**	**yes**	no	GUI query & answer	ok	**yes**	**yes**	**verification & validation**	**yes**	**yes**	**yes**	**yes**	depends on problem and solver	**yes**	**integrated solvers are not bad**
System modeling & analysis tools	**summary**	**yes**	user must code	user must code	user must code	general math and specialized control	some	**ModelCenter is good**	**good?**	**yes**	no	no	few	**yes**	?	**yes**	?	?	**integrated solvers are not bad**
	Phoenix Technologies ModelCenter + MagicDraw + solvers	**yes?**	user must code	user must code	user must code	trade space analysis	**good**	**good**	ok?	no	no	verification	no	?	?	no	?	?	**integrated solvers are not bad**
	Wolfram SystemModeler (Mathematica + Modelica + visual modeling)	**yes**	user must code	user must code	user must code	mathematical problem solving	user must code?	no?	**good?**	**yes**	user must code	verification	no	**yes**	?	**yes**	?	not bad	not well?
	Ptolemy II + PTIDES	**yes**	user must code	user must code	user must code	several "models of computation"	user must code	no?	**good?**	**yes**	in some cases; otherwise, user code		**yes**	**yes**	?	**yes**	?	?	?
	AADL	?	?	?	?	?	?	?	?	?	?	?	**yes?**	?	?	?	?	?	?
Simulators		**yes**	no	no	no	user may add code for math/control problem solving with other tools (e.g. matlab)	user must code	no	**good**	**yes**	no	validation	some	**yes**	**yes**	**yes**	**yes**	**yes**	no (Monte Carlo)
Govt/industry analysis tools	**e.g., ASPEN, EUROPA, SIPE, SHINE, BEAM, SCL**	**yes**	no?	some	some	user must code	some	?	some but limited	some	some	verification	some	no	few	not bad for some	some	some	**not bad**
COTS security tools	**anti-virus, snort, host-based, penetration testing**	no	**yes**	limited	static	risk assessment	N/A	N/A	N/A	**yes**	no?	verification	no?	**yes**	N/A	in some ways	**yes**	**yes**	**in some ways**
Research	**modeling, analysis**	**yes**	some	some	some	some	rare	rare	rare	some	some	some verification	some	no	no	no	some	some	no
Problem solvers	**MILP (Mixed Integer Linear Problem) solvers like CPLEX**	user must encode	user must encode	user must encode	user must encode	user must encode	N/A?	N/A	N/A	N/A	no	verification	no	no	not bad	N/A	no?	not bad	**not bad**
	Learning tools (e.g., Weka)	no	**yes**	no	no	**yes**	?	?	no?	N/A	**yes**	validation?	some	**yes**	**yes**	**yes**	**yes**	not bad	**not bad**

How is the AE approach different?

1. More expressive modeling language
   - simulators (e.g. Simulink) don’t solve problems
   - problem solver models often lack fidelity from lack of the language features
     - object-oriented class structures
     - continuous variables, time, state change
     - a variety of operators/functions (some just support logic)
     - quantification (“all computers connected to the public wireless network”)
   - The current AE supports these (and uncertainty to a limited degree)
   - Language enables dynamic creation of constraints
     (others that do this: MDS, some automated planners, APGEN?)
2. Poses a wider variety of questions on the same behavior model
   - For DR scenario, simulation and scheduling:
     - **What-if:** *<u>What</u>* events will occur and *<u>what</u>* happens to load and generation *<u>if</u>* responses are intercepted?
     - **When:** *<u>When</u>* do each of the events of the DR process take place?
   - Also capable of planning and model checking:
     - **What to do:** *<u>What</u>* events *<u>must</u>* execute to satisfy the constraints/achieve goals?
     - **Is possible:** Is it possible for an execution to cause a failure?
     - **Is impossible:** Is it impossible for an execution to cause a failure?
3. Scales well for problem size, ignoring problem complexity:
   - 10K+ events
   - 10K+ state variables (timelines)
   - 300K+ constraints
   - similar to ASPEN
   - CPLEX > million constraints—we plan to integrate solvers like CPLEX
4. Integrates with multi-view modeling (SysML) with access to ontologies.

Why is an expressive language important?

We need an *expressive* behavior modeling language for modeling information flow, timing, uncertainty, etc.,

Specifying meters for a scenario alternatives:

BAD, but it’s the current state-of-the-art

```
operator ami_meter_1_sends_last_gasp_message (
        pre: meter_1_last_gasp_message = false
        post: meter_1_last_gasp_message = true
)
operator ami_meter_2_sends_last_gasp_message (
        pre: meter_2_last_gasp_message = false
        post: meter_2_last_gasp_message = true
)
. . .
operator ami_meter_50000_sends_last_gasp_message (
        pre: meter_50000_last_gasp_message = false
        post: meter_50000_last_gasp_message = true
)
```

GOOD

```
class AmiMeter inherits from MeshNetworkTransceiver {
        Messages messages = { LAST_GASP, READING, . . .}
        . . .
}
AmiMeter meters[50000];
```

AUI: Posing Analysis Questions

What Load Reduction would cause grid.instability = true ? 117MW

What Load Reduction could cause grid.instability = true ? [9.2MW, infinity]

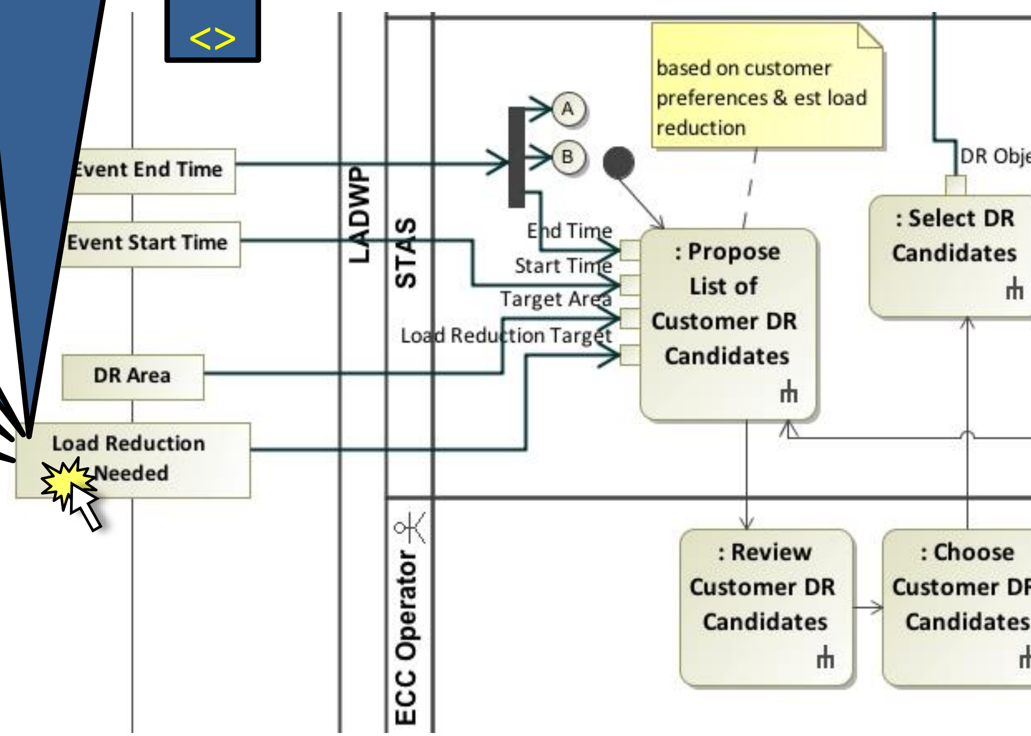
If Load Reduction = 15MW , what DR Area could cause grid.instability = true ? {7,13,51}

Load execution data for past 12 hours . . . done

Query: What unexpected events caused grid.instability = true ?
expected numDrParticipants = 4200
actualNumDrParticipants = [0,12]

Query:
If Load Reduction = 15MW would grid.instability = ? maybe

=
<
<=
>
>=
<>



How would you pose alternative questions if using other systems?

- ASPEN – figure out how to change activity and/or state/resource models to ask question
- CPLEX – figure out how to change model as a system of equations and an optimization function
- Simulink – change model and write MATLAB code
- Mathematica – figure out Mathematica code
- Wolfram SystemModeler – edit model (either graphically or in Modelica) and pose question in Mathematica.
- In our approach,
  - For the AUI, a new query statement template is added to others in a text file or GUI form:

```
query HowMany:
    parameters = // format: [<type> variable|expression <parameter name>]*
        Number variable numVarParam1
        Boolean expression boolExprParam2
    statement = “How many ” + numVarParam1 + “ could cause ” + boolExprParam2 + “?”
    statement = “For how many ” + numVarParam1 + “ is ” + boolExprParam2 + “ possible?”
```

  - In AE, add code (2 lines for this example) to an auto-generated Java class to change the model (in memory, not the original).
    - code on next slide
  - Now, this “HowMany” question can be asked of any model, for any variable in the model, and for any expression involving those variables.

AE event/behavior/constraint language

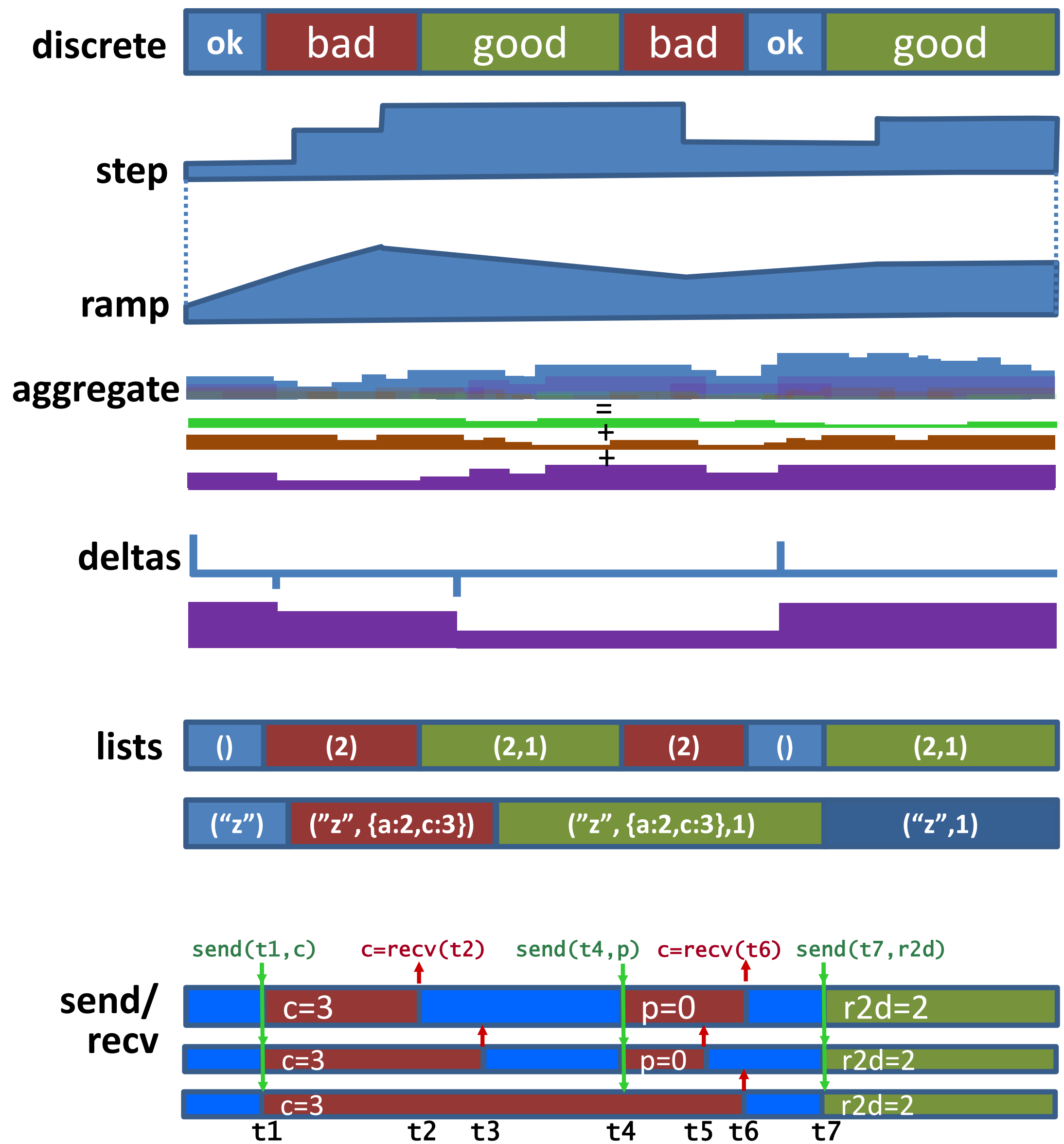
- Adds declarative behavior language elements to procedural Java for problem solving.

- **Classes** (OO inheritance, nested classes, leveraging Java)
- **Parameters**, a.k.a. variables with value domains
- **TimeVarying** – a.k.a. timelines, variables whose values are functions of time
- **Dependencies** (e.g., energy <- power * duration)
- **Constraints** (e.g., event1.end + 5 min < event2.start)
- **Events** – classes with start/end time variables
  - **Effects** – dependencies on TimeVarying
  - **Elaborations** – a.k.a. conditional decompositions, AND/OR event trees, subactivities, subgoals, methods, hierarchical task networks…

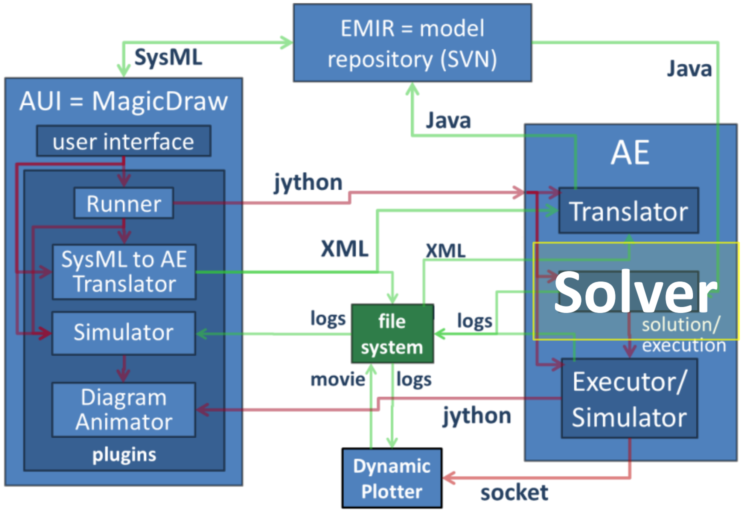
```
# This example is not an actual model.
# The syntax is modified to fit the screen.
class Customer:
  Parameter int id
  Parameter CustomerType type = Residence
  Parameter bool participate = false
  Parameter Meter meter
  TimeVarying float load = new TimeVarying("kV"+id)
  Dependency id <- meter.id

  event usePower:
    Parameter float power, actualPower
    Parameter DRObject drObj
    Parameter time lastReport, nextReport
    Parameter bool willReport
    Parameter string fileName
    TimeVarying float projectedLoad
    Dependencies
      energy <- actualPower * duration
      actualPower <- power - if(participate,0,
                                drObj.shed(power,id))
      fileName <- dataFolder + os.sep + "meter" +
                  id + "_" + startTime.day() + ".csv"
      nextReport <- lastReport + drObj.reportPeriod
      willReport <- endTime < new TimeVarying(fileName)
      projectedLoad <- new TimeVarying(fileName)
    Effect
      load.add(power, startTime, endTime)
    Elaboration
      if participate meter.report(load=load)
    Constraints
      !participate || !willReport ||
      ( report.startTime >= nextReport - 2min &&
        report.startTime <= nextReport + 2min )
```

TimeVarying (Timelines)

- **TimeVaryingMap<T>**
  - setValue(time, value)
  - unsetValue(time, value)
  - plus(number, start, end)
  - plus(TimeVarying)
  - minus, times, dividedBy
  - init(csvFileName)
- **LinearTimeline<Number>**
  - initFrom(deltaMap)
  - getDeltaMap()
- **TimeVaryingMaps<T>**
  - init(folderName)
  - init(map<csvFileName, weight>)
- **Consumable<Number>**
  - plus(number, time)
  - getDelta(t1, t2)
  - getDeltaMap()
  - init (deltaMap)
- **TimeVaryingList<T>**
  - add(time, value)
  - add(time, List)
  - addIfNotContained(time, value)
  - remove(time, value/List)
  - contains(time, value)
  - nthElement(time, n)
- **ObjectFlow<T>**
  - send(time, value)
  - sendIf(time, value, condition)
  - receive(time, value)
  - gotSomething(time)
  - addListener(ObjectFlow)

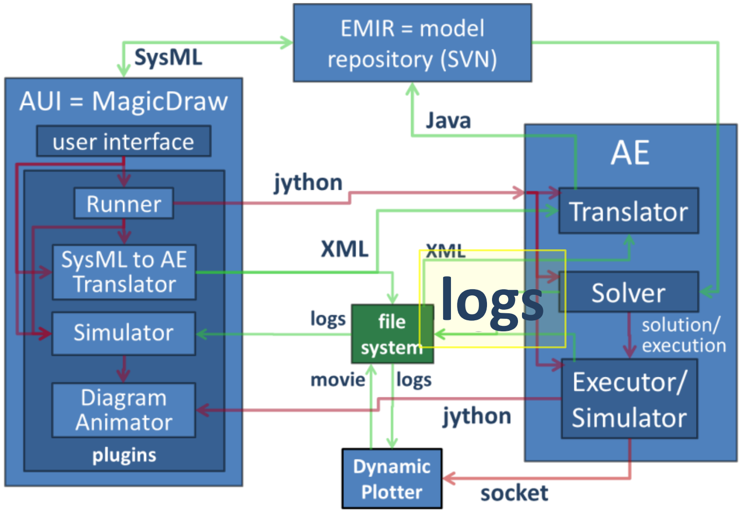


Constraint Solver



1. gather constraints
2. assign new values
3. elaborate or deconstruct events
4. apply (on unapply) effects to timelines
5. repeat

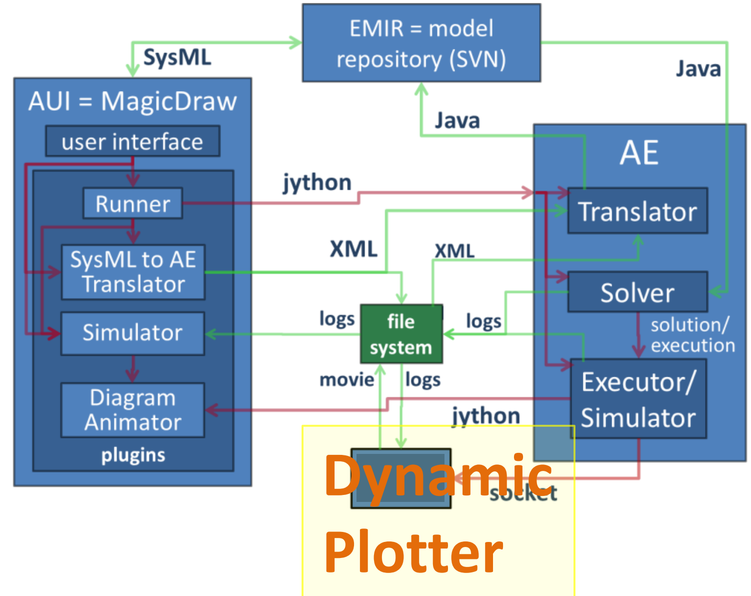
Logged output



- stats after each loop through constraints
- all constraints
- violated constraints
- execution/solution (events, parameter values, timelines)
- simulation – print event start/end and state transitions in scaled time
- snapshot simulations saved periodically during solving

Dynamic Plotter

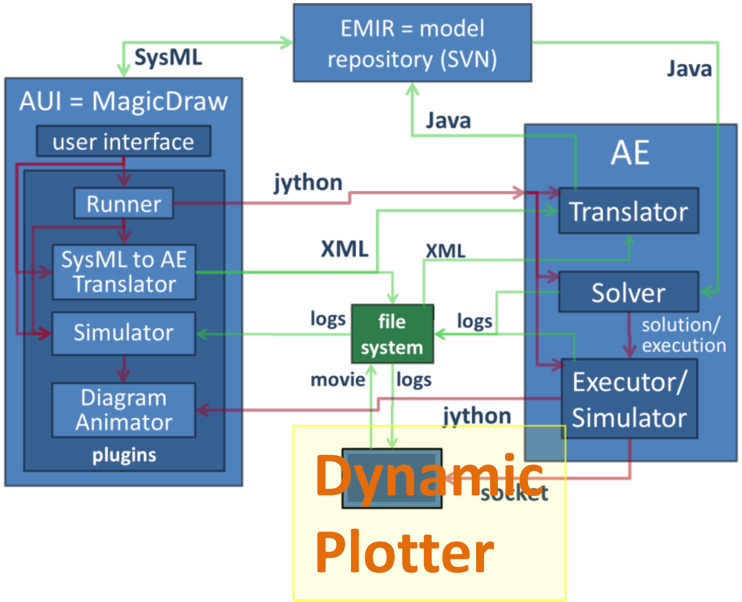
- Enthought Python
  - doesn’t integrate well with Jython (and, thus, MD)
  - invoked as standalone from file or over a socket from Java (and probably Jython).
- options for scrolling, dynamic resizing, frames per second, skipping frames to catch up with simulation, saving movie (mp4)
- does not (yet) simulate by itself, so loads from log files are not animated
- supports projected and “live” data
  - can update projections
- currently some discrepancies in rendering from Java vs file because of sampling and handling of null values



<show plot animation>

Activity Diagram Animator

- time-scaled simulators in Java and Python can drive
- corrects for time error by monitoring system time
- data from log file or Solver
- max delay between event steps



show MD animation