FERN
Framework for Evaluation of chemical Reaction Networks

Florian Erhard

Institut für Informatik, Ludwig-Maximilians-Universität München

April 15, 2007
Contents

Overview
  Definition
  Example
  Complex?
Contents

Overview
- Definition
- Example
- Complex?

Structure
- Main classes / interfaces
- UML of Network related classes
- Observer system

Additional features
- Gnuplot
- Analysis
- Stochastics

Documentation and Examples
Contents

Overview
  Definition
  Example
  Complex?

Structure
  Main classes / interfaces
  UML of Network related classes
  Observer system

Additional features
  Gnuplot
  Analysis
  Stochastics
Contents

Overview
  Definition
  Example
  Complex?

Structure
  Main classes / interfaces
  UML of Network related classes
  Observer system

Additional features
  Gnuplot
  Analysis
  Stochastics

Documentation and Examples
What is FERN?
A fast, extensible and comprehensive framework for simulation and analysis of chemical reaction networks.
What is FERN?
A fast, extensible and comprehensive framework for simulation and analysis of chemical reaction networks.

What is FERN not?
A second Dizzy / Copasy / Stocks. You cannot do simulations by mouseclick but by a very few lines of code.
What is FERN?
A fast, extensible and comprehensive framework for simulation and analysis of chemical reaction networks.

What is FERN not?
A second Dizzy / Copasy / Stocks. You cannot do simulations by mouseclick but by a very few lines of code.

Advantage
If you are able to write many reasonable lines of code, you can do arbitrarily complex simulations.
**Elementary Example**

```java
1 Network net = new RNMLNetwork(new File("some_net.xml"));
2 Simulator sim = new GibsonBruckSimulator(net);
3 Observer obs = sim.addObserver(new AmountIntervalObserver(sim,1,X));
4 sim.start(50);
5 System.out.println(obs);
```
Complicated?

- 100 classes / interfaces
- with 12906 lines
Complicated?

- 100 classes / interfaces
- with 12906 lines

But...

- only 3 central classes
- half of the lines are javadoc
Structure
Structure
Observer

Observers can be registered at the Simulator and record various data about the simulation runs. Some are able to handle repeated runs (and yield average data), some are not (refer to the javadoc).
Observer

Observers can be registered at the Simulator and record various data about the simulation runs. Some are able to handle repeated runs (and yield average data), some are not (refer to the javadoc).

Data

Depending on what kind of data observers record, they provide different kind of data (plain text, gnuplot, histograms,...).
Observer

- initialize()
- Observer.started()
- while (SimulationController.goOn())
  - Observer.step()
  - performStep()
  - Observer.thetaEvent()
  - Observer.fireReaction()
  - AmountManager.fireReaction()
- Observer.finished()
- Observer.print()

- Observer
  - started()
  - step()
  - thetaEvent()
  - fireReaction()
  - finished()
  - print()
A class GnuPlot is included in FERN which enables you to
A class GnuPlot is included in FERN which enables you to

- retrieve the gnuplot data as String or save it to a file
A class GnuPlot is included in FERN which enables you to

- retrieve the gnuplot data as String or save it to a file
- get the plot command
A class GnuPlot is included in FERN which enables you to

- retrieve the gnuplot data as String or save it to a file
- get the plot command
- plot the data by calling gnuplot from java
A class GnuPlot is included in FERN which enables you to

- retrieve the gnuplot data as String or save it to a file
- get the plot command
- plot the data by calling gnuplot from java
- retrieve the plot as image object or save it to a png file
A class GnuPlot is included in FERN which enables you to

- retrieve the gnuplot data as String or save it to a file
- get the plot command
- plot the data by calling gnuplot from java
- retrieve the plot as image object or save it to a png file
- show the plot in a JFrame
The AnalysisBase class provides

▶ additional index structures
▶ generic bfs / dfs
The AnalysisBase class provides

- additional index structures
- generic bfs / dfs

Used by included algorithms

- ShortestPaths
- AutocatalyticNetworkDetection
Random number generation is handled in one central Stochastics class.
Random number generation
is handled in one central Stochastics class

- easy replacement of the random number generator
Random number generation is handled in one central Stochastics class

- easy replacement of the random number generator
- make simulations deterministic by explicitly setting the seed value
Random number generation is handled in one central Stochastics class

- easy replacement of the random number generator
- make simulations deterministic by explicitly setting the seed value
- count number of random numbers generated from various distributions
javadoc

comprehensive documentation of each class and method
javadoc
comprehensive documentation of each class and method

Technical overview
A detailed technical overview about FERN’s implementation is accessible as pdf document.
Examples

Maybe most important...
There are many examples included in the package fern.example which demonstrate different aspects of FERN.