Logical Model editor
In Cytoscape 3

Aurélien Naldi   CIG, UNI-Lausanne

OSGi (Open Services Gateway initiative)

- Service-centered specification
- OSGi framework connects the pieces (multiple implementations)
- Bundles contain code modules (JAR files with some metadata)
- Services provided by bundles
  - Published by the framework
  - Other bundles can grab them

OSGi in Cytoscape 3

- Many bundles (>100)
  - Used libraries
  - API: no code, no service (java interfaces)
  - Impl: provide services
  - GUI

- Some bundles:
  - Events, IO, Service, Task
  - Data model
  - View model, Presentation, Vizmap, Layout
  - Session, Application, Swing Application

Cy3: data model

- Based on flexible Tables
  - 3 tables per network:
    - Network itself
    - Nodes
    - Arcs
  - Lines ↔ unique identifiers (node, arc)
  - Columns:
    - Title
    - Data type: [list of] String or Boolean or Integer or Double

Cy3: caveats (Work in progress)

- API changes
- Missing parts
- Launcher changes
  - From Raw Felix to custom launcher: invisible
  - Removal of spring OSGi
  - From Felix to Karaf
- Devs are listening, fixing bugs, updating API

Adding annotations

- OSGi bundle for annotations

- New columns:
  - Comment: string
  - Links: list of strings.
    - Using MIRIAM URNs
  - Type and main ID: strings
    - Still fuzzy
    - SBO + selected MIRIAM DB
    - Node-specific

- Special treatment for references
    - Grab metadata from pubmed

Annotation GUI

- Separate bundle for the GUI
- Provides a panel (OSGi service)
- Provides a service to follow selection
  (raw cy3: listen to the selection columns)
- Creates undoable edits objects
  (cy3 has the architecture but few are created)

Logical modeling

- Add sign on edges

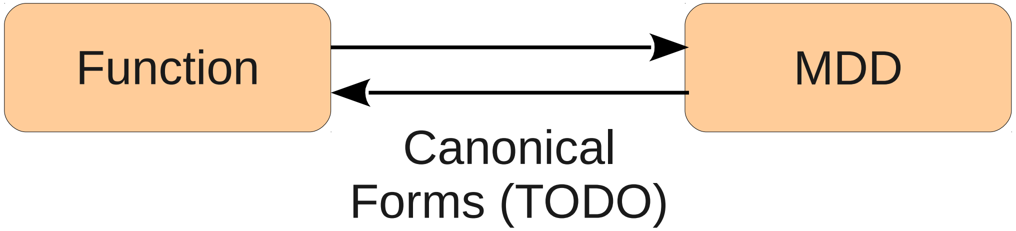
- Map custom objects on nodes
  - Can not be stored in the node table
    (session hooks to save them, needs cleanup)
  - Specialized fields and methods
    - Get logical rules (function or MDD)
    - Function generator
    - Hand-written function

- Custom object associated to the network
  - No need to save it, provides only methods

Definition of the dynamics

Logical functions (Boolean, one for each value)

- Leaves: constraints on components
- Internal nodes: Boolean operators



MDDs (single MDD for the whole function)

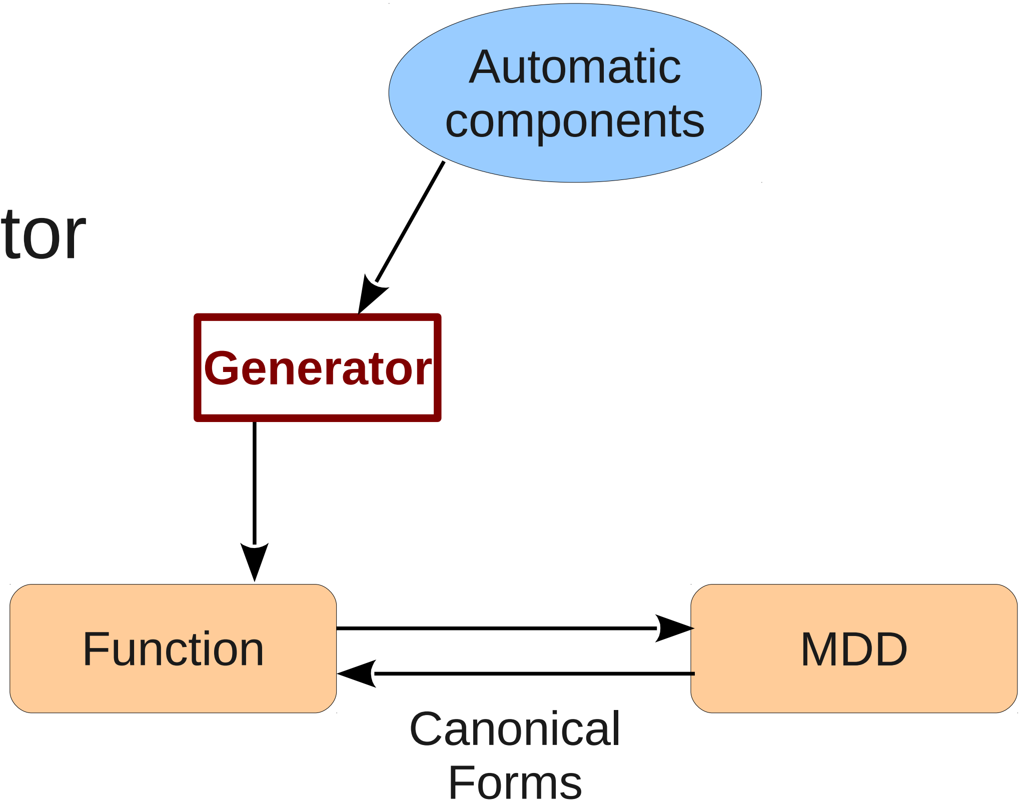
- Leaves: target value
- Internal nodes: constraints on components (ordered)

Definition of the dynamics

- SQUAD-like generator

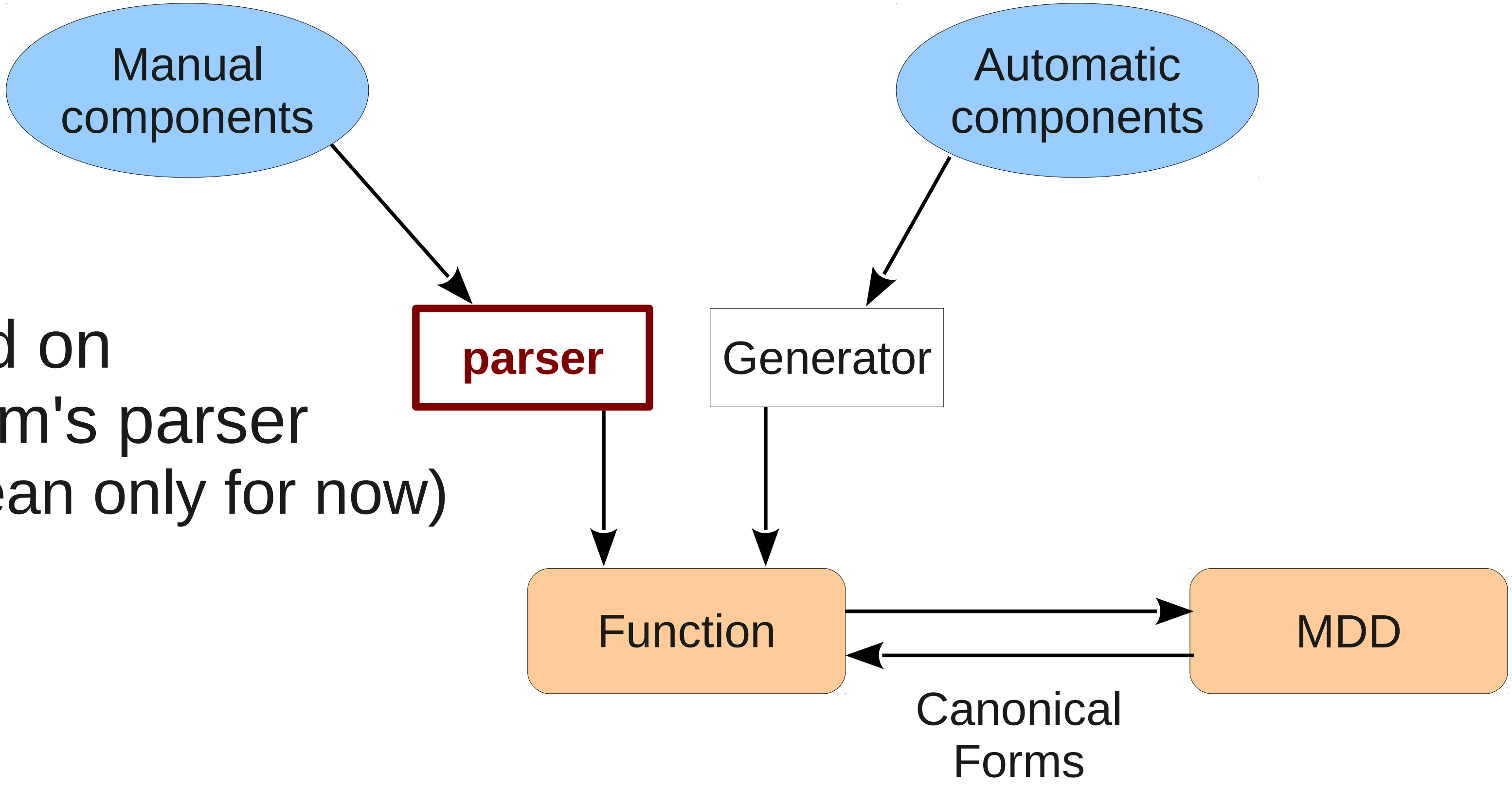
  alternatives?

- Input components



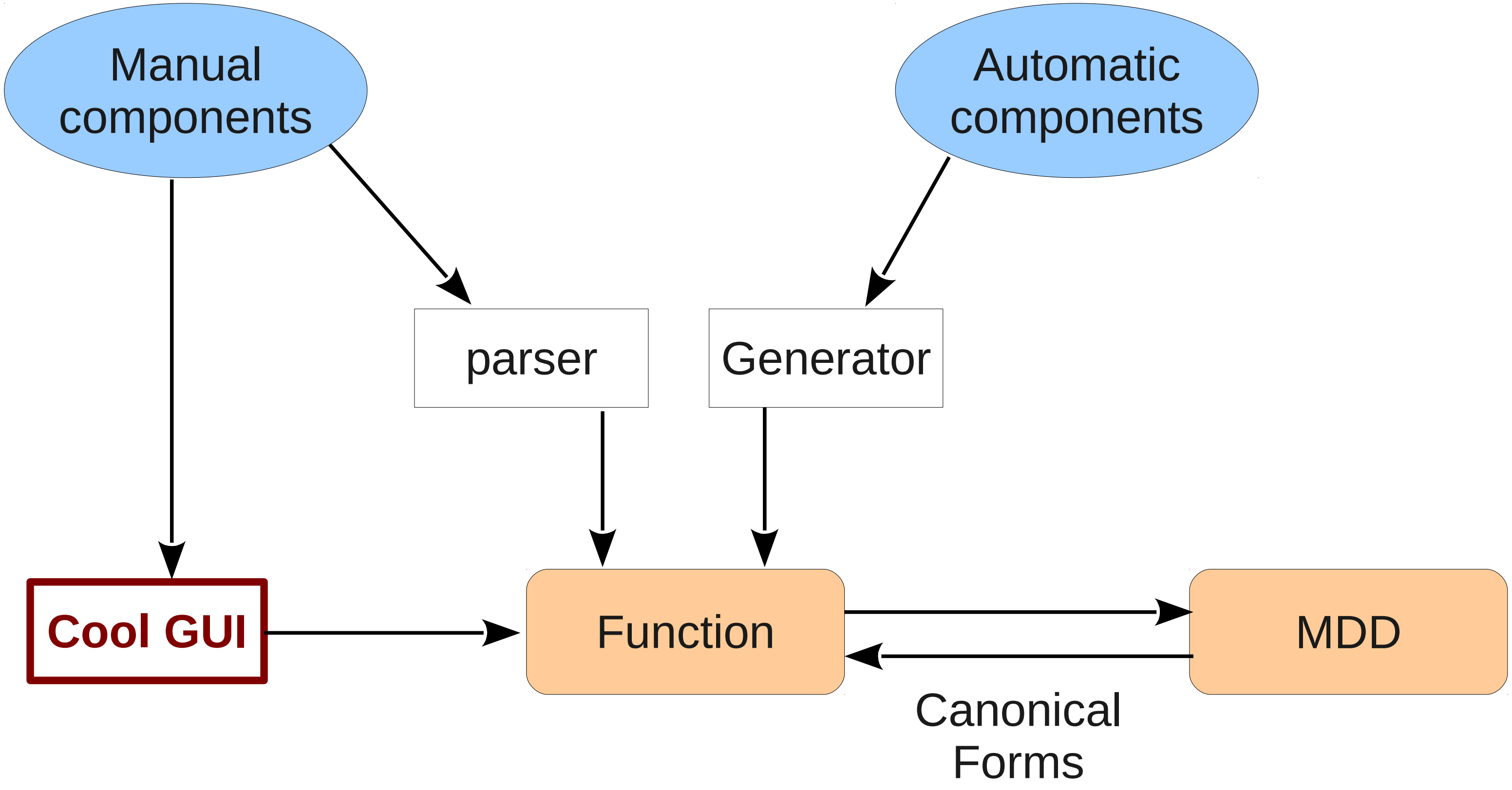
Definition of the dynamics

Based on GINsim's parser (Boolean only for now)



Update edges to reflect inferred signs

Definition of the dynamics



Someday?

How to use it?

- Get cytoscape 3 (from SVN)
  - Beware, new versions may introduce changes
  - I can provide a working snapshot
- Put a few JAR files in “framework/deploy/”
  - Annotation data model and GUI
  - MDD toolkit
  - Logical model and GUI
- Debug
- Enjoy :)

Runing tasks in cytoscape 3

- Task: ready-to-use runnable
- Creation, configuration: TaskFactory
  - OSGi service
  - Parameters defined using java annotation
- Auto-generated GUI to set-up the factory
  - Works well for SIMPLE cases
  - Custom GUI in other cases

Integrating existing tools

- Export to another tool
- Integrate existing code
  - GINsim's stable state search
  - More to come?
- Ideas/code welcome
  → **What are your needs?**

Missing parts

- Editor
  - Full multivalued support (thresholds, multi-arcs,...)
  - Initial states, input configurations, perturbations
  - Lookup interactions and components online
  - Models of models
- Simulation
- Analysis tools
- Command line interface

Format

- Exchange format: SBML
  - Cytoscape already depends on JSBML
  - TODO: update JSBML, import export glue
- Edition format:
  - Saving in the cytoscape session
    Extra file for logical functions, pure cytoscape?
  - Do we need an other format?
- Import/export to other formats?
  - Import GINsim's format
  - Adapt existing GINsim's export plugins

What comes next?

- Sharing these bundles
  - Choose a license (LGPL?)
  - Publish core code (git repository?)
  - Start adapting existing tools
- Share files and code with GINsim
  - MIRIAM annotations
  - MDD toolkit (no function generators in GINsim)
- Scripting mode
  GINsim uses jython, can we build on this?