Avoiding complexity in the development of corporate grid applications using the REST api

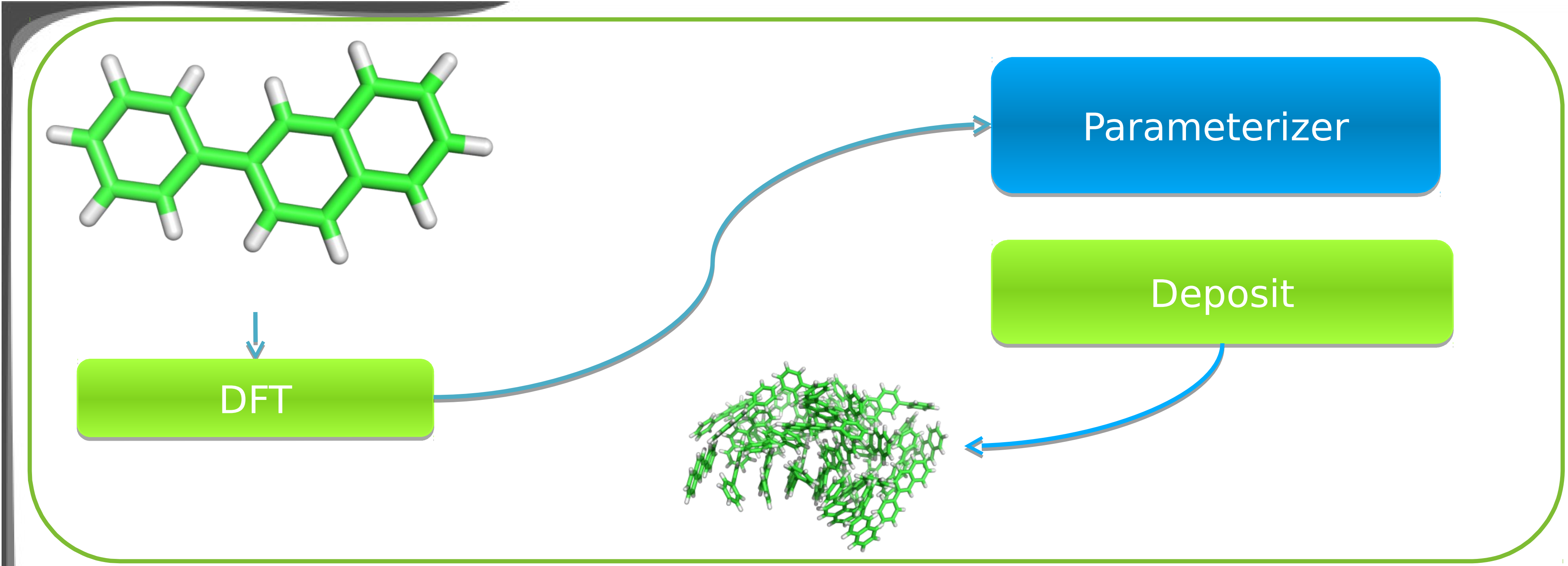
Timo Strunk
Nanomatch GmbH

Talk

- Our UNICORE Use-case
  - Why we require UNICORE
- Our Software until now
  - Difficulties
  - What went wrong, what did we learn?
- Current and Future implementations
  - REST API
  - UI

About Nanomatch

- Spin-off company based on Code of MMM@HPC and the Wenzel group in KIT
- We investigate
  - Thin-Film morphologies for OLED based devices
  - Electron/Hole Mobilities
- Or to put this in another way
  - We try to answer
    - Will my TV turn on with a specific material?
    - Will my TV work for more than five minutes?



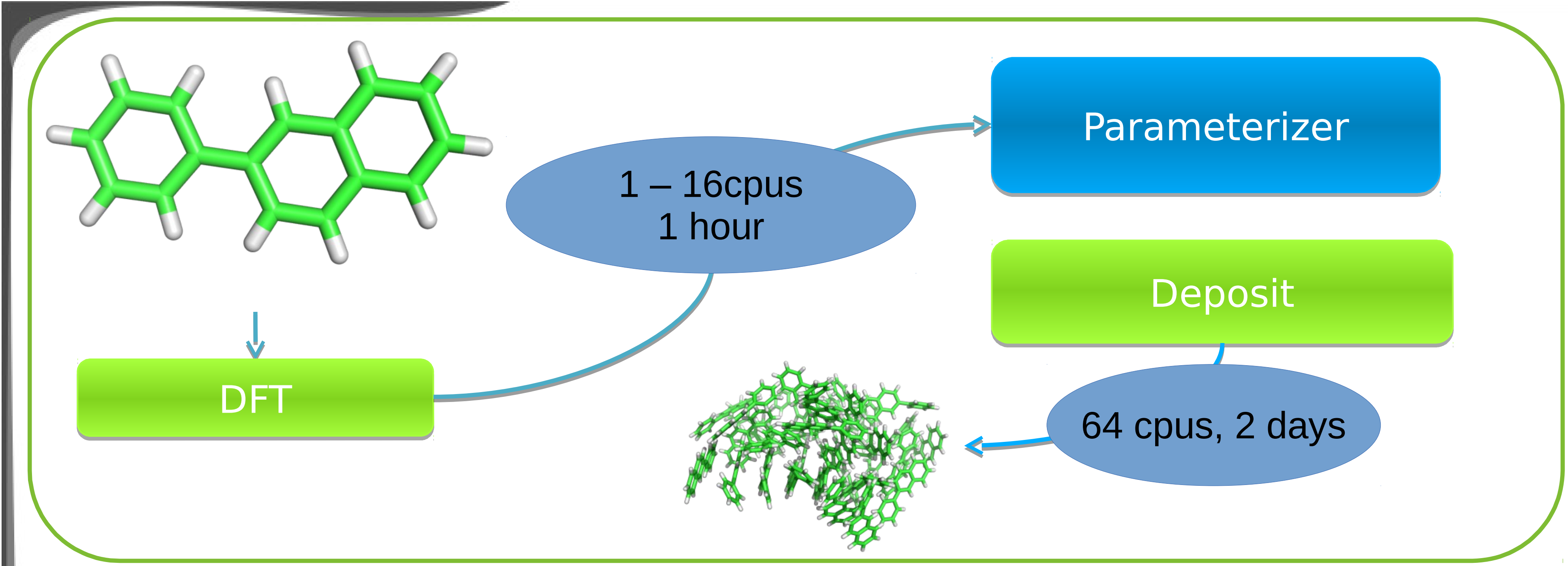
DFT

Parameterizer

Deposit

$$k_{if} = \frac{2\pi}{\hbar} \left|\boxed{J_{if}}\right|^2 \frac{1}{\sqrt{4\pi \boxed{\lambda_{ij}} k_b T}} \exp\left(- \frac{(\lambda_{ij} + \boxed{\Delta E_{ij}})^2}{4 \lambda_{ij} k_b T} \right)$$

Energetic disorder	*ΔE_{if} and $\sigma(\Delta E)$*
Electronic coupling	*J_{if}*
Reorganization energy	*λ_{if}*



DFT

1 – 16cpus
1 hour

Parameterizer

Deposit

64 cpus, 2 days

$$k_{if} = \frac{2\pi}{\hbar}\boxed{|J_{if}|}^2 \frac{1}{\sqrt{4\pi \boxed{\lambda_{ij}} k_b T}} \exp\left(-\frac{(\lambda_{ij} + \boxed{\Delta E_{ij}})^2}{4\lambda_{ij} k_b T}\right)$$

Energetic disorder

Electronic coupling

Reorganization energy

64 cpus
1 week
or 1000 cpus
1 day

Software

What we wanted

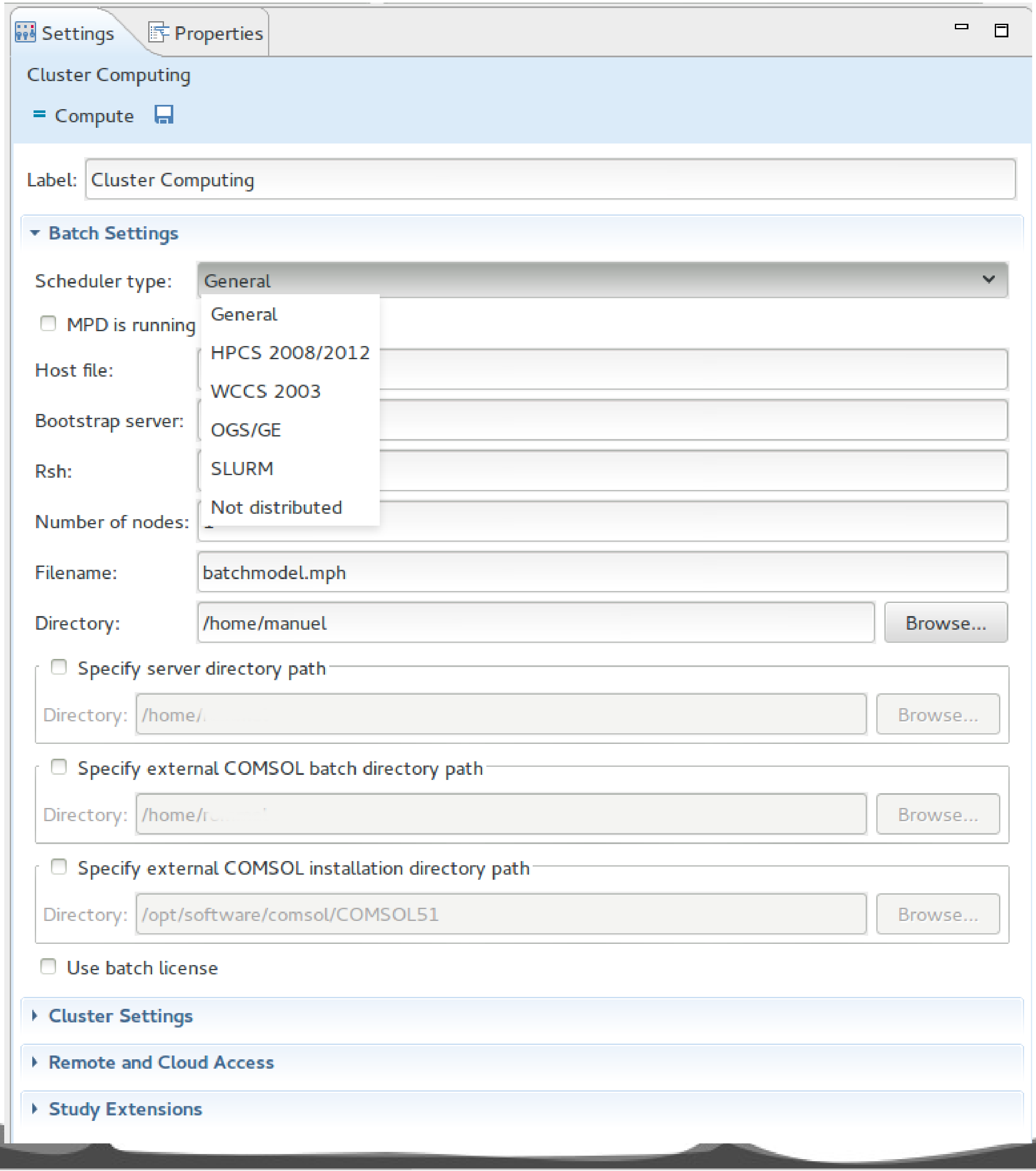
- UI - Click to work
- Input molecule
  - Get Morphology
- Input Morphology
  - Get Mobility

What we had

- Complex
- Script based
  - Parameters in scripts
  - Parameters as commandline
- Some MPI programs
- Some OpenMP

→ We required a unified UI, but needed to interface directly with a cluster / grid

Other implementations



Settings | Properties

Cluster Computing

= Compute

Label: Cluster Computing

Batch Settings

Scheduler type: General

- General
- HPCS 2008/2012
- WCCS 2003
- OGS/GE
- SLURM
- Not distributed

☐ MPD is running

Host file:

Bootstrap server:

Rsh:

Number of nodes: 1

Filename: batchmodel.mph

Directory: /home/manuel | Browse...

☐ Specify server directory path

Directory: /home/[redacted] | Browse...

☐ Specify external COMSOL batch directory path

Directory: /home/r[redacted] | Browse...

☐ Specify external COMSOL installation directory path

Directory: /opt/software/comsol/COMSOL51 | Browse...

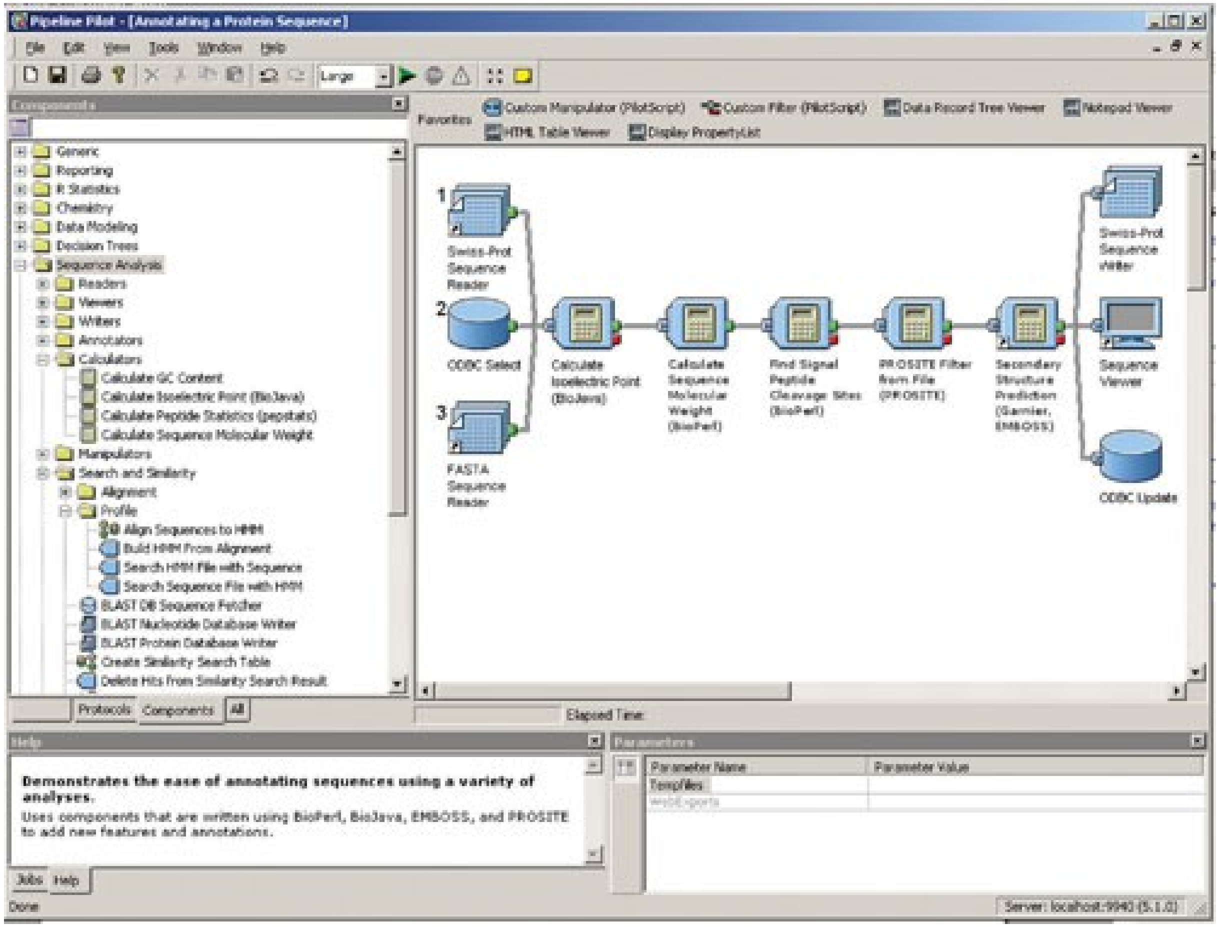
☐ Use batch license

Cluster Settings

Remote and Cloud Access

Study Extensions

Other implementations cont.


Pipeline Pilot - [Annotating a Protein Sequence]
File Edit View Tools Window Help
Large
Components
Generic
Reporting
R Statistics
Chemistry
Data Modeling
Decision Trees
Sequence Analysis
Readers
Viewers
Writers
Annotators
Calculators
Calculate GC Content
Calculate Isoelectric Point (BioJava)
Calculate Peptide Statistics (pepstats)
Calculate Sequence Molecular Weight
Manipulators
Search and Similarity
Alignment
Profile
Align Sequences to HMM
Build HMM From Alignment
Search HMM File with Sequence
Search Sequence File with HMM
BLAST DB Sequence Fetcher
BLAST Nucleotide Database Writer
BLAST Protein Database Writer
Create Similarity Search Table
Delete Hits From Similarity Search Result
Protocols
Components
All
Favorites
Custom Manipulator (PilotScript)
Custom Filter (PilotScript)
Data Record Tree Viewer
Notepad Viewer
HTML Table Viewer
Display PropertyList
1
Swiss-Prot Sequence Reader
2
ODBC Select
3
FASTA Sequence Reader
Calculate Isoelectric Point (BioJava)
Calculate Sequence Molecular Weight (BioPerl)
Find Signal Peptide Cleavage Sites (BioPerl)
PROSITE Filter from File (PROSITE)
Secondary Structure Prediction (Garnier, EMBOSS)
Swiss-Prot Sequence Writer
Sequence Viewer
ODBC Update
Elapsed Time:
Help
Parameters
Jobs
Help
Done
Server: localhost:9940 (5.1.0)

Demonstrates the ease of annotating sequences using a variety of analyses.

Uses components that are written using BioPerl, BioJava, EMBOSS, and PROSITE to add new features and annotations.

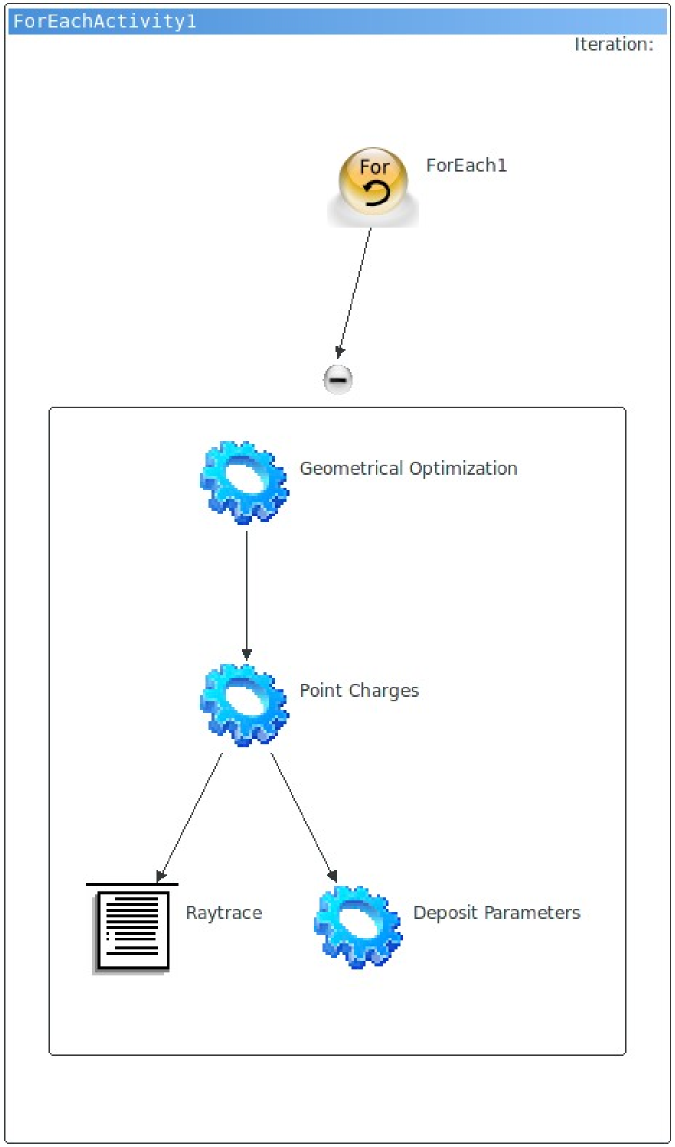
Parameter Name	Parameter Value
Tempfiles	
WebExports	

Competitors' solutions

- Specific to grid interface ☹
  - Torque, SLURM, etc.
- ssh based ☹
- Sometimes expensive ☹
- Mostly modular ☺

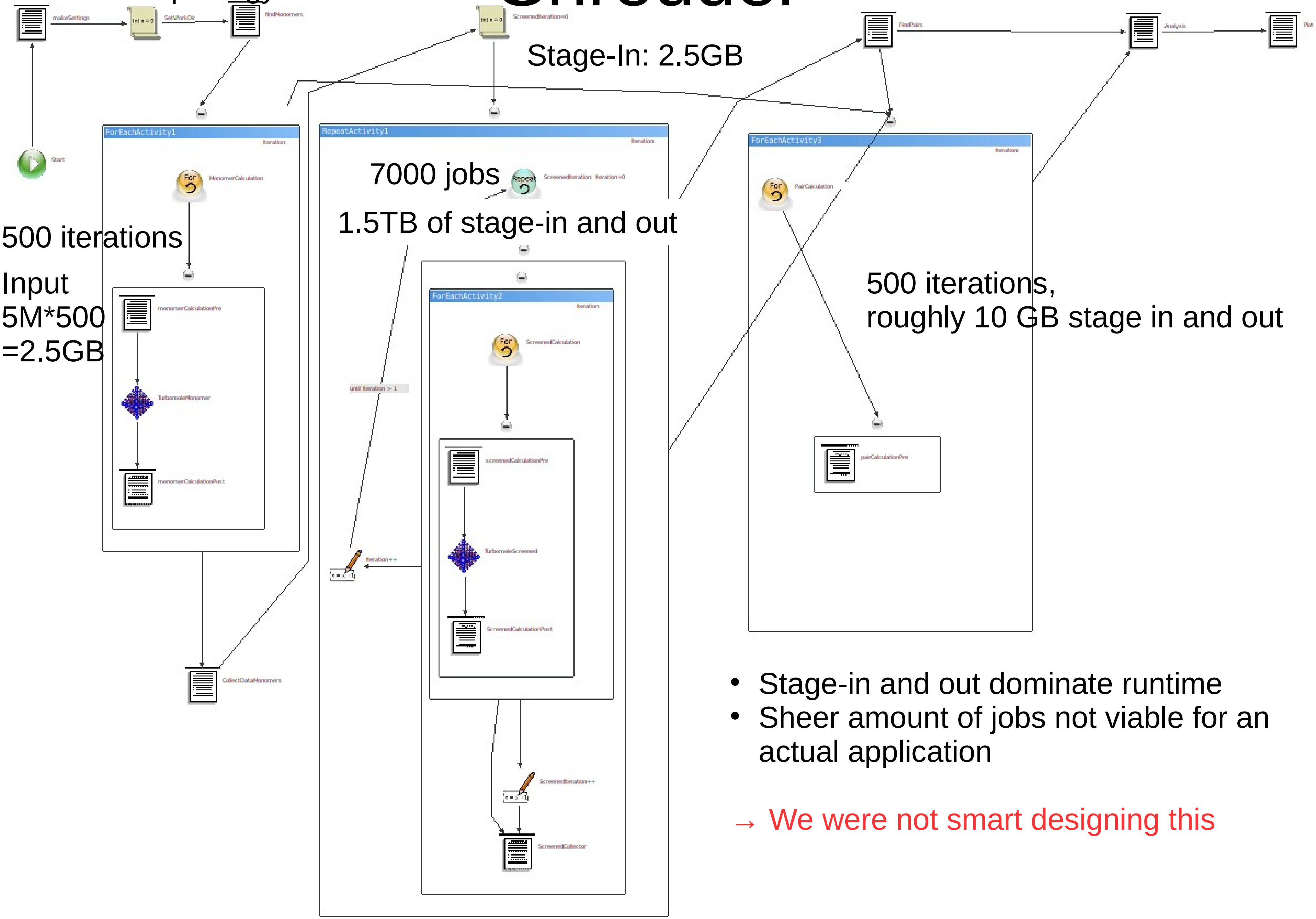
→ UNICORE

Parameterizer



- First success story
  - Unbundling allows for different allocations
    - Geo Opt
    - Point Charges
- Batch processing
- Negative Points
  - Data duplication
  - Addressed in WF Server 7.x

Shredder



5M morphology

Stage-In: 2.5GB

500 iterations

Input
5M*500
=2.5GB

7000 jobs

1.5TB of stage-in and out

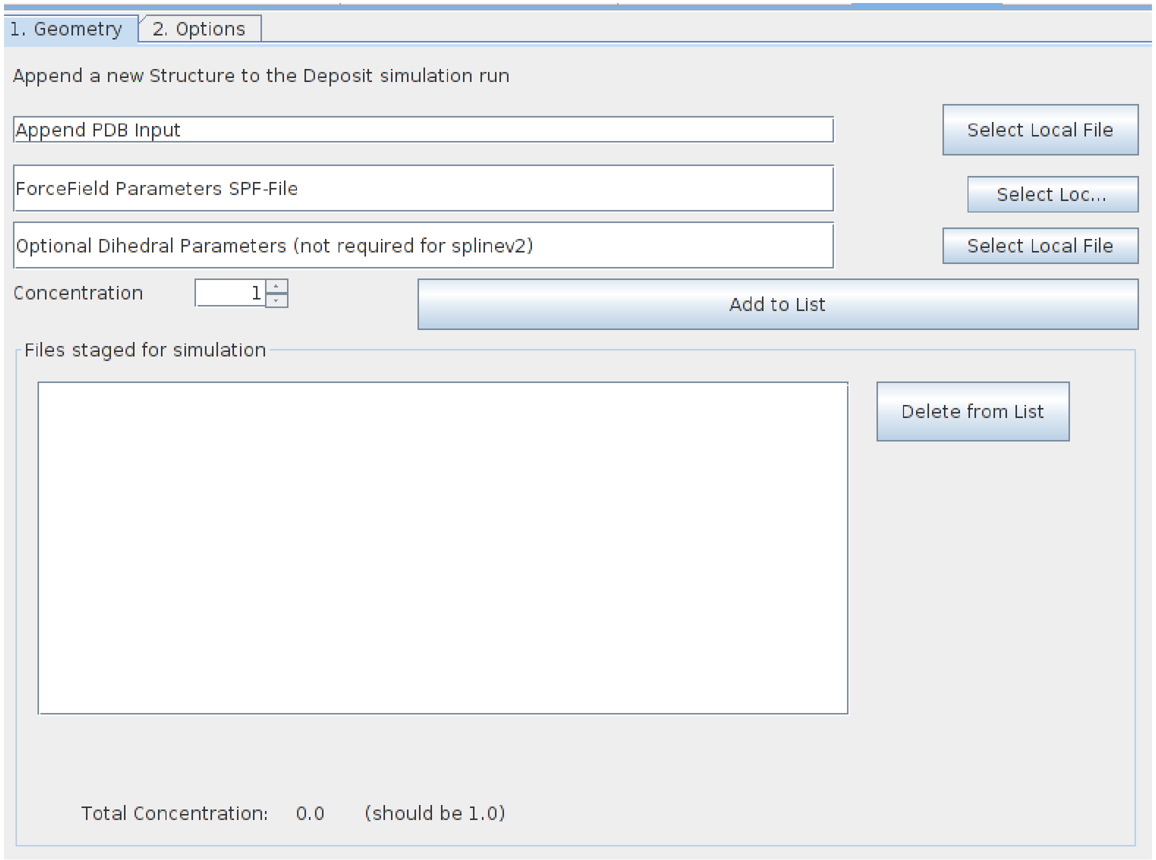
500 iterations,
roughly 10 GB stage in and out

- Stage-in and out dominate runtime
- Sheer amount of jobs not viable for an actual application

→ We were not smart designing this

Single Application UIs

- Scientific origin of our software leads to a large and diverse knowledge of our developers



Append a new Structure to the Deposit simulation run

Append PDB Input

ForceField Parameters SPF-File

Optional Dihedral Parameters (not required for splinev2)

Concentration 1

Total Concentration: 0.0 (should be 1.0)

- Most of us know Python
- Some know C++
- None know Java

- Mix of SWT and Swing
- Not everyone respects the Gridbean-model

Solution
Generic Gridbean

Summary of our mistakes

- Workflows
  - Data transfer often abused
  - Non-portable, not inheritable
- Individual application GUIs
  - Gridbean concept not sustainable in our group
  - Java code ends up non-maintainable due to
    - SWT, Swing mix
    - Gridbean / non Gridbean storage
  - Very slow development cycle for trivial UIs

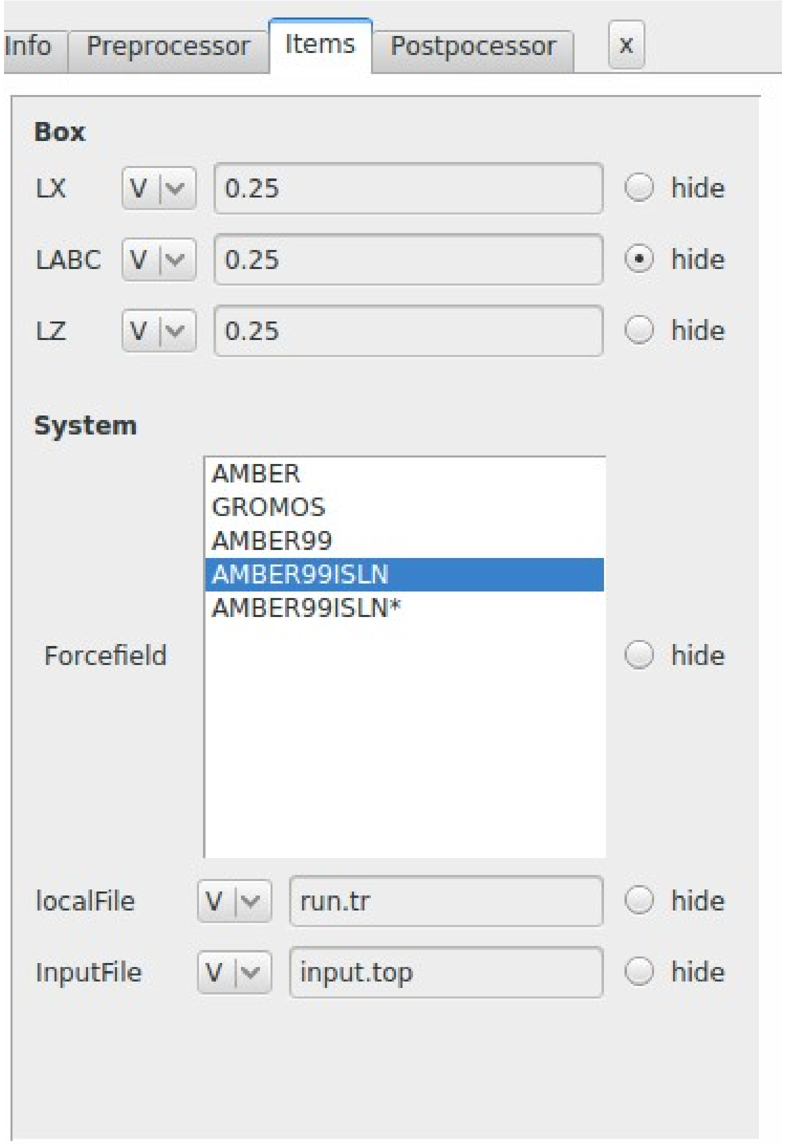
Lessons learned

- Do not convert your existing applications into Workflows
- Do not invest into GUIs, where none are required (Generic Gridbean)

Current and Future Developments

- REST API allows fast new client development
- Remove development stress from the scientific developers
  - Write Input/Output specification
  - Complex purely declarative GUI akin to Generic Gridbean
- Write multiple input formats
  - No Shell variables, but hierarchical
    - YML, XML

Declarative Client (PySide - QT)



Box

LX V 0.25 ○ hide

LABC V 0.25 ● hide

LZ V 0.25 ○ hide

System

Forcefield ○ hide

AMBER
GROMOS
AMBER99
AMBER99ISLN
AMBER99ISLN*

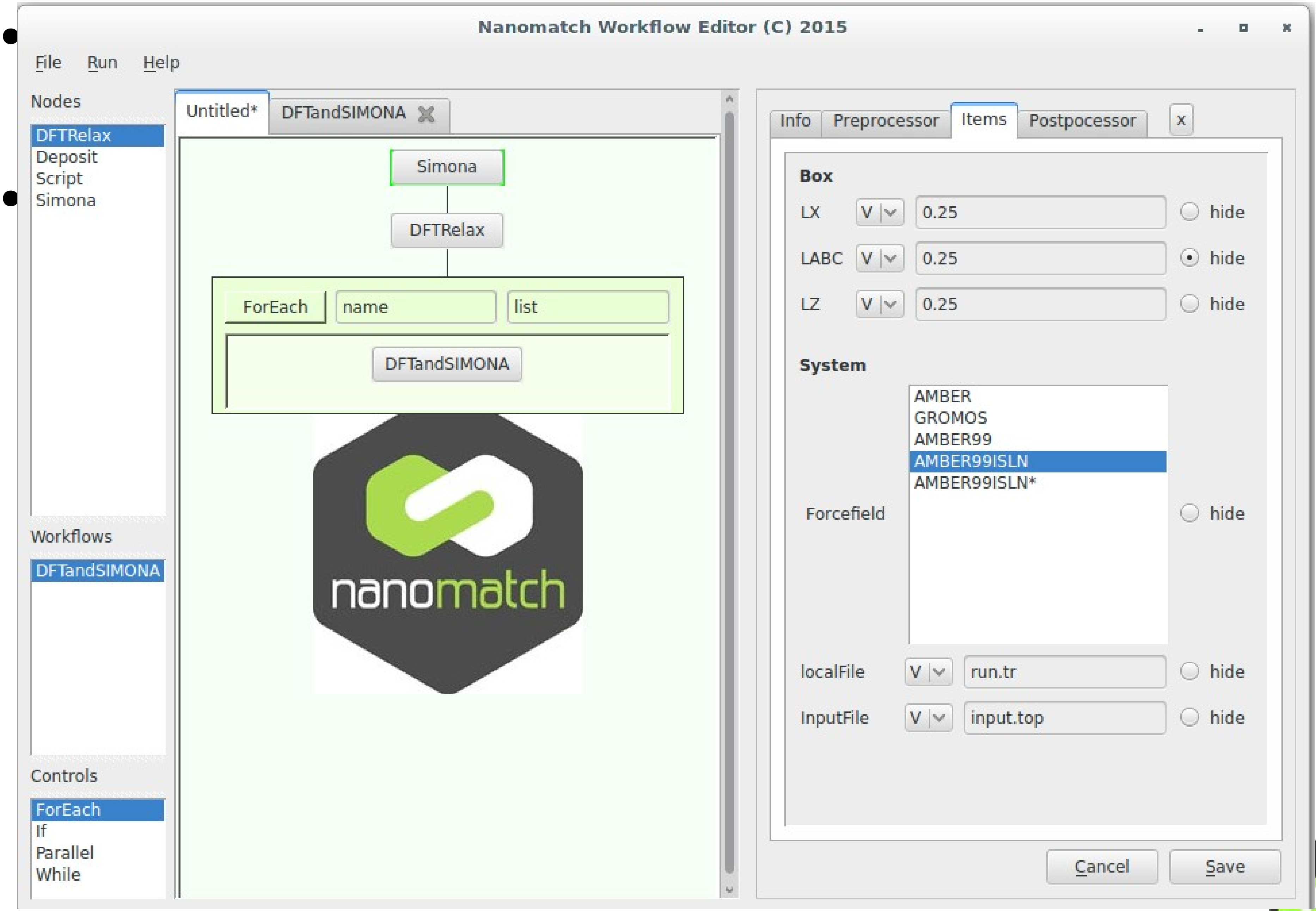
localFile V run.tr ○ hide

InputFile V input.top ○ hide

```
<Template name="Simona">
  <Section name="Box">
    <Float name="LX" hidden="False">
      0.25
    </Float>
    <Float name="LABC" hidden="True">
      0.25
    </Float>
    <Float name="LZ">
      0.25
    </Float>
  </Section>
```

Once you “render” the job, you get a YML with the exact same structure as above minus the markup information

Workflow encapsulation



Box

LX	V	0.25	○ hide
LABC	V	0.25	● hide
LZ	V	0.25	○ hide

System

Forcefield: AMBER, GROMOS, AMBER99, AMBER99ISLN (selected), AMBER99ISLN* ○ hide

localFile	V	run.tr	○ hide
InputFile	V	input.top	○ hide

Conclusions

- Past
  - Large freedom in development
  - Bad Choices, long development times
    - Huge Appreciation for Generic Gridbean
  - Custom GUIs only necessary for file preparation
    - Better handled by external thread
- Present and Future
  - Make a universial GUI to easily pass all parameters required for runtime only
  - Do not require any executable code for simple GUIs

Thank you for listening