Lively Simulation

Leonardo Hübscher, Juliane Kleinknecht

End User Development Seminar SS 20
Software Architectures

14.07.2020

Outline

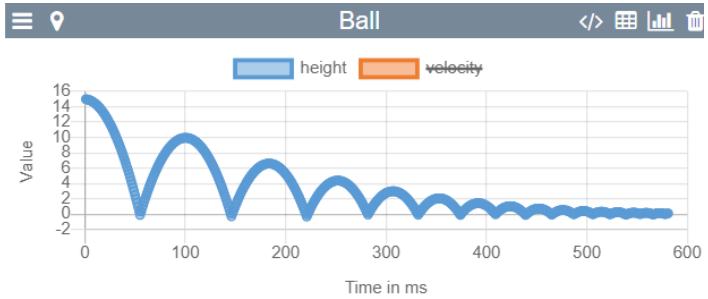
1. Simulations
2. Motivation
3. Project Goal
4. Energy Simulation
5. Demo
6. Improve End User Experience
7. What's Next?

Simulations

Real system 🌍 → Too complex ❔ 😨 → Model 📝 → Simulation ⚙

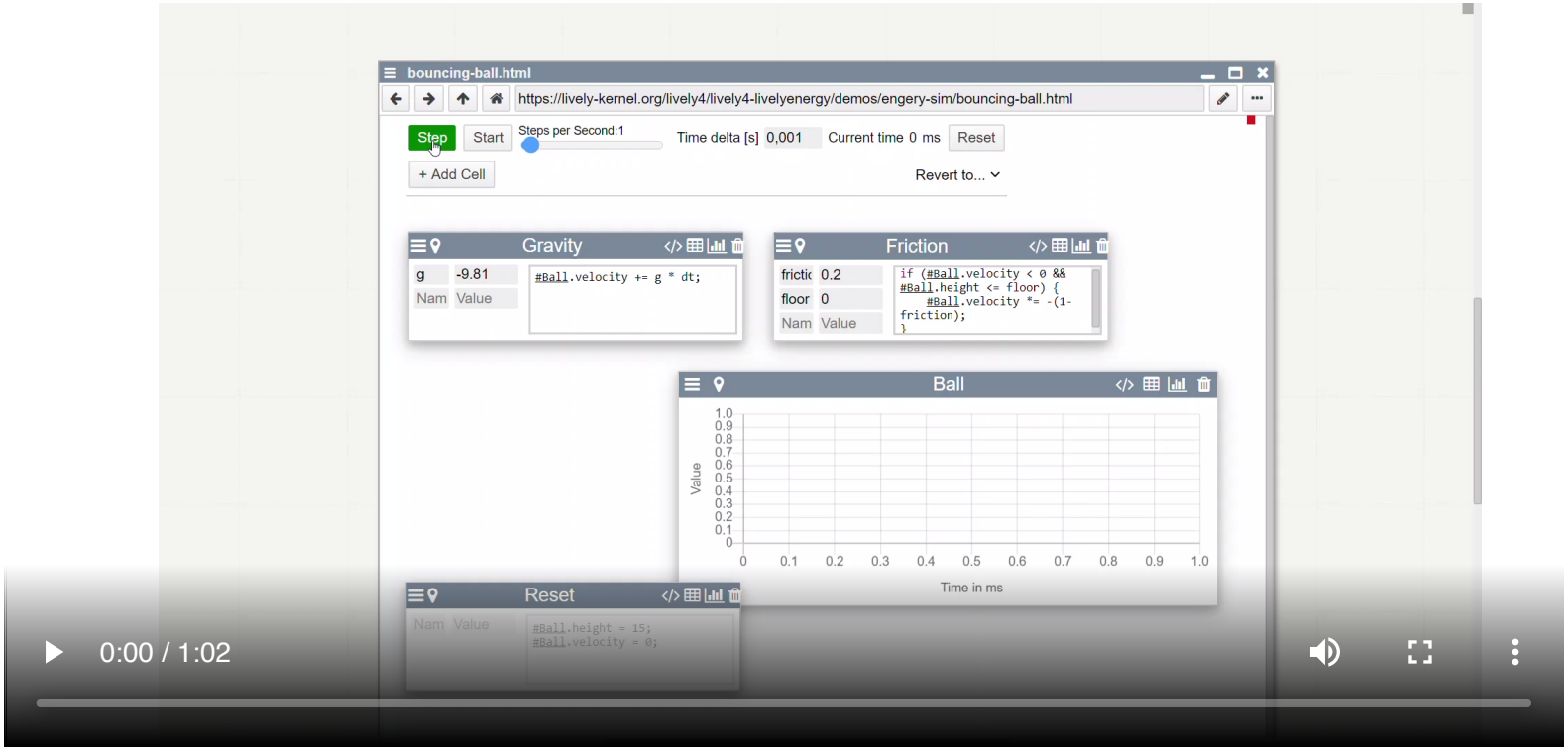
- Experimenting in simulation → finding hypothesis
- Understanding the behavior of systems
- **Dynamic simulations:** time dependent 🕑

  → observe processes and flows

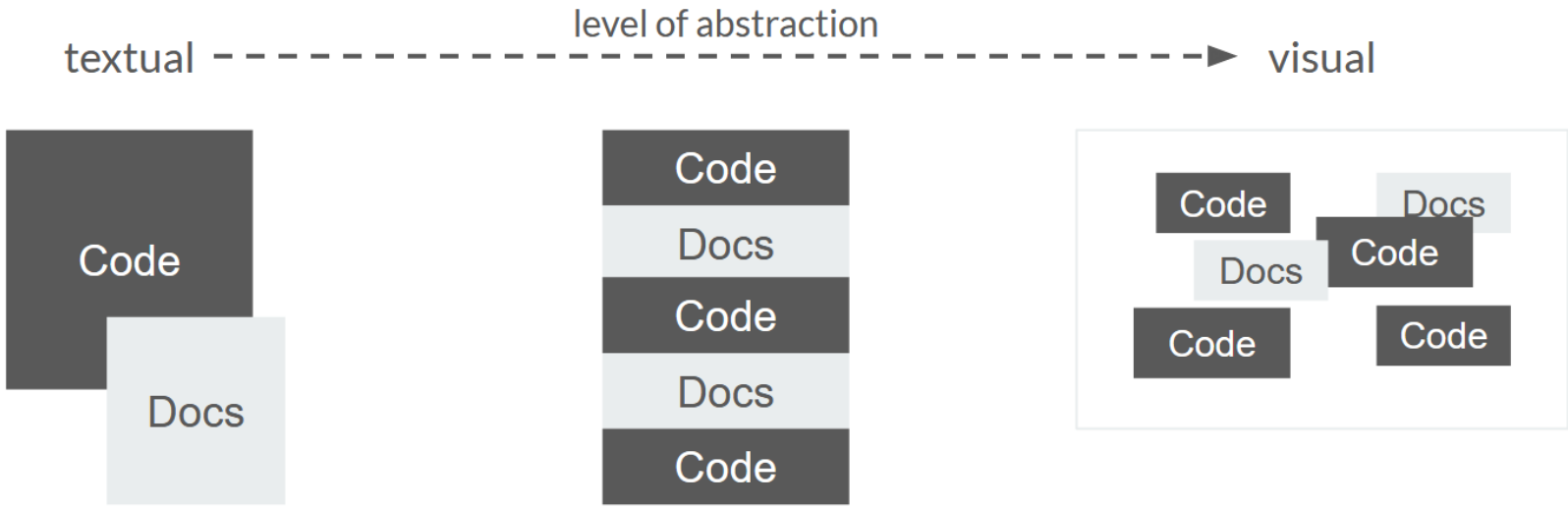


Bouncing Ball Example

Demo Bouncing Ball



Motivation



We want to enable end-users to create simulations in a visually structured way

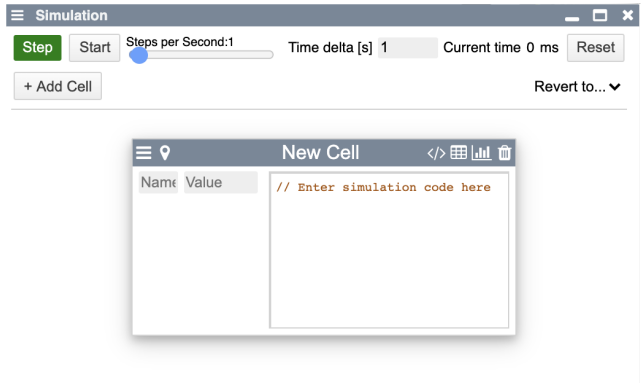
5

Project Goal

We want to enable end-users to create simulations in a visually structured way

Idea Build a semi-graphical simulation framework

- *Starting point:* PoC* in Lively Webwerkstatt (legacy)
- Allow documentation to be part of simulation
- Provide a way that allows simulation result investigation
- Proper Lively integration
- Port existing *energy simulation* from legacy

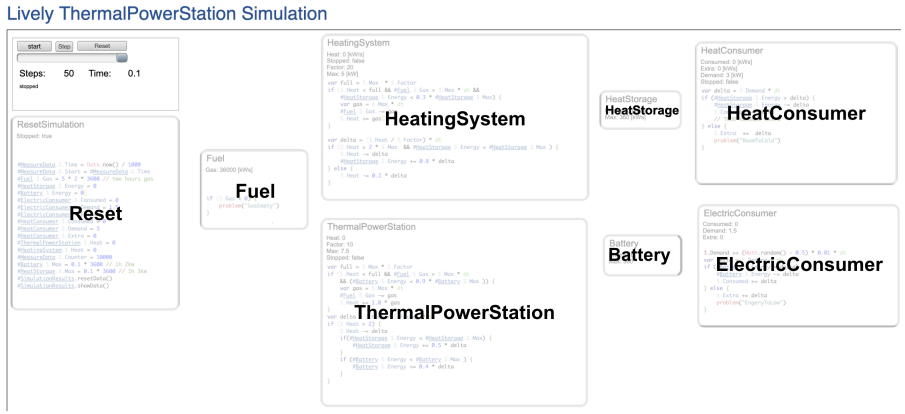


Simulation Framework

*PoC...Proof of concept

Energy Simulation

- Simulate energy flow
- Fuel → transformer → consumer
- Possible hypothesis:
  - Is heat storage big enough?
  - Is a backup system required?



Energy simulation in Lively Webwerkstatt [1]

[1] https://lively-kernel.org/repository/webwerkstatt/demos/EnergySimulationScripted.xhtml

Demo

Energy Simulation Port

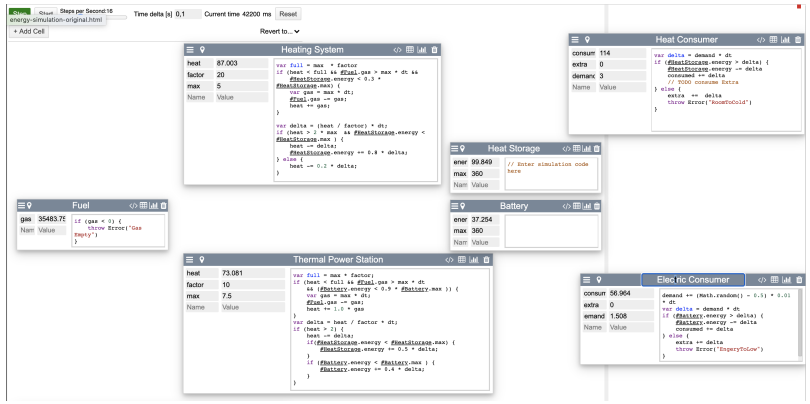
8

Demo

Let's use our framework for the energy simulation!

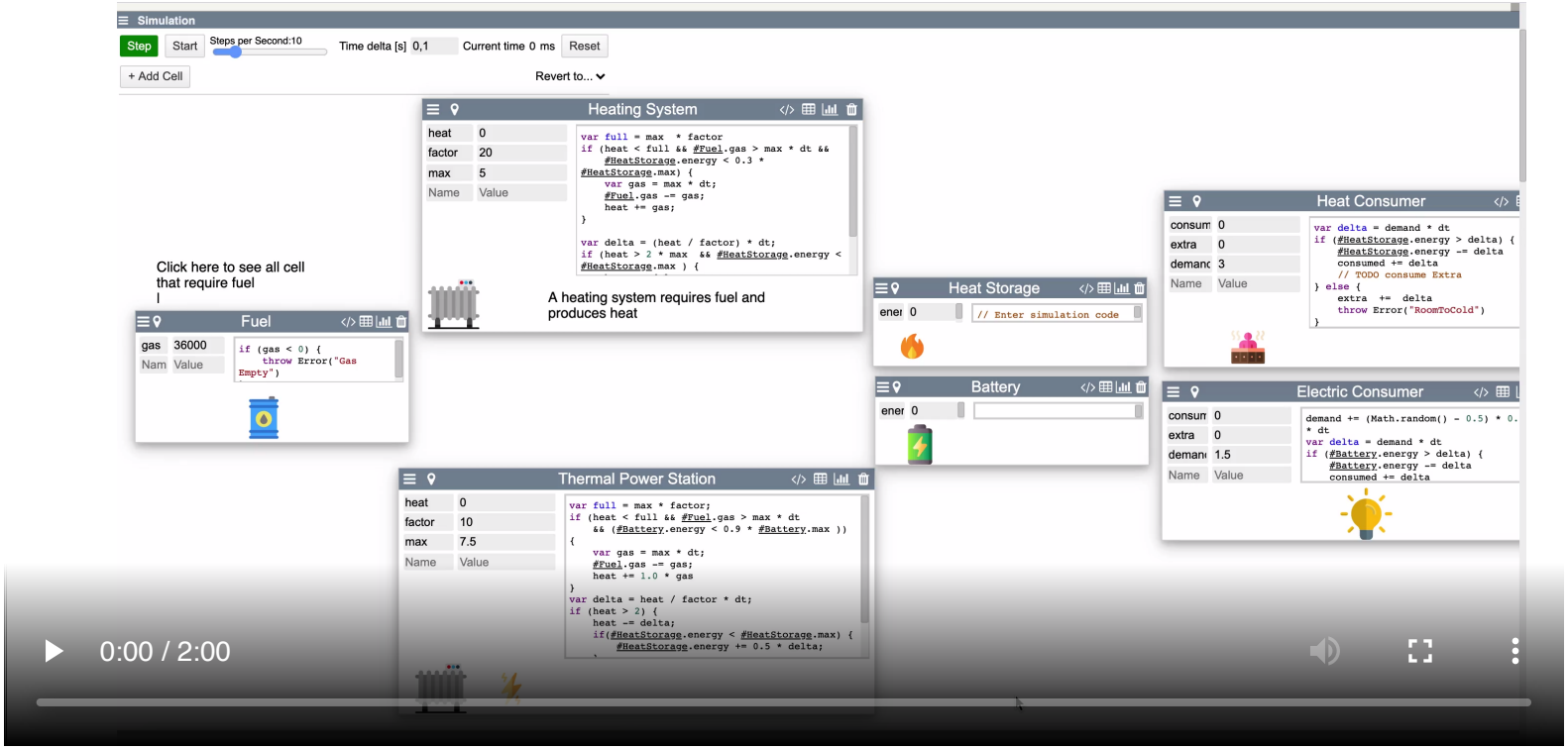
What we want to show:

1. Ported simulation is working
2. How the simulation is being edited
3. Cell locality
4. Cell state visualization



Energy simulation in Lively4

Demo

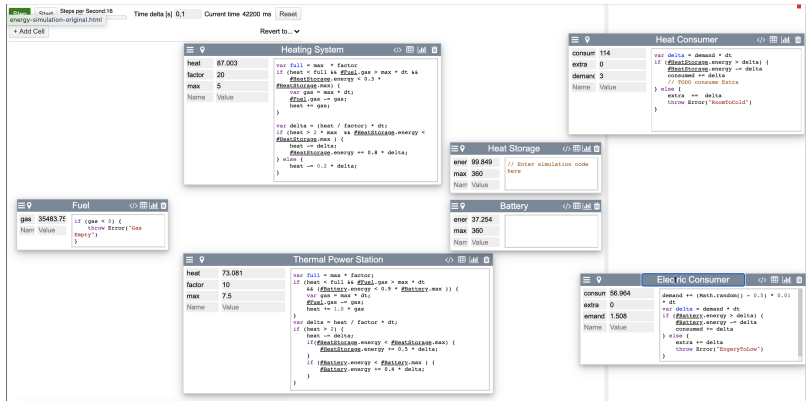

Simulation
Step
Start
Steps per Second:10
Time delta [s] 0,1
Current time 0 ms
Reset
+ Add Cell
Revert to...
Heating System
heat 0
factor 20
max 5
Name Value
var full = max * factor
if (heat < full && #Fuel.gas > max * dt &&
#HeatStorage.energy < 0.3 *
#HeatStorage.max) {
var gas = max * dt;
#Fuel.gas -= gas;
heat += gas;
}
var delta = (heat / factor) * dt;
if (heat > 2 * max && #HeatStorage.energy <
#HeatStorage.max) {
Click here to see all cell
that require fuel
A heating system requires fuel and
produces heat
Fuel
gas 36000
Nam Value
if (gas < 0) {
throw Error("Gas
Empty")
Heat Storage
ener 0
// Enter simulation code
Heat Consumer
consum 0
extra 0
demanc 3
Name Value
var delta = demand * dt
if (#HeatStorage.energy > delta) {
#HeatStorage.energy -= delta
consumed += delta
// TODO consume Extra
} else {
extra += delta
throw Error("RoomToCold")
}
Battery
ener 0
Electric Consumer
consun 0
extra 0
demani 1.5
Name Value
demand += (Math.random() - 0.5) * 0.
* dt
var delta = demand * dt
if (#Battery.energy > delta) {
#Battery.energy -= delta
consumed += delta
Thermal Power Station
heat 0
factor 10
max 7.5
Name Value
var full = max * factor;
if (heat < full && #Fuel.gas > max * dt
&& (#Battery.energy < 0.9 * #Battery.max))
{
var gas = max * dt;
#Fuel.gas -= gas;
heat += 1.0 * gas
}
var delta = heat / factor * dt;
if (heat > 2) {
heat -= delta;
if(#HeatStorage.energy < #HeatStorage.max) {
#HeatStorage.energy += 0.5 * delta;
0:00 / 2:00

Demo

Let's use our framework for the energy simulation!

What we have seen:

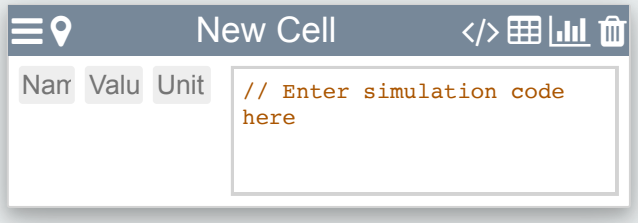
1. Ported simulation works ✅
2. Loading a simulation and adding a cell ✅
3. Independent cells/ global state access ✅
4. Log tables and charts ✅



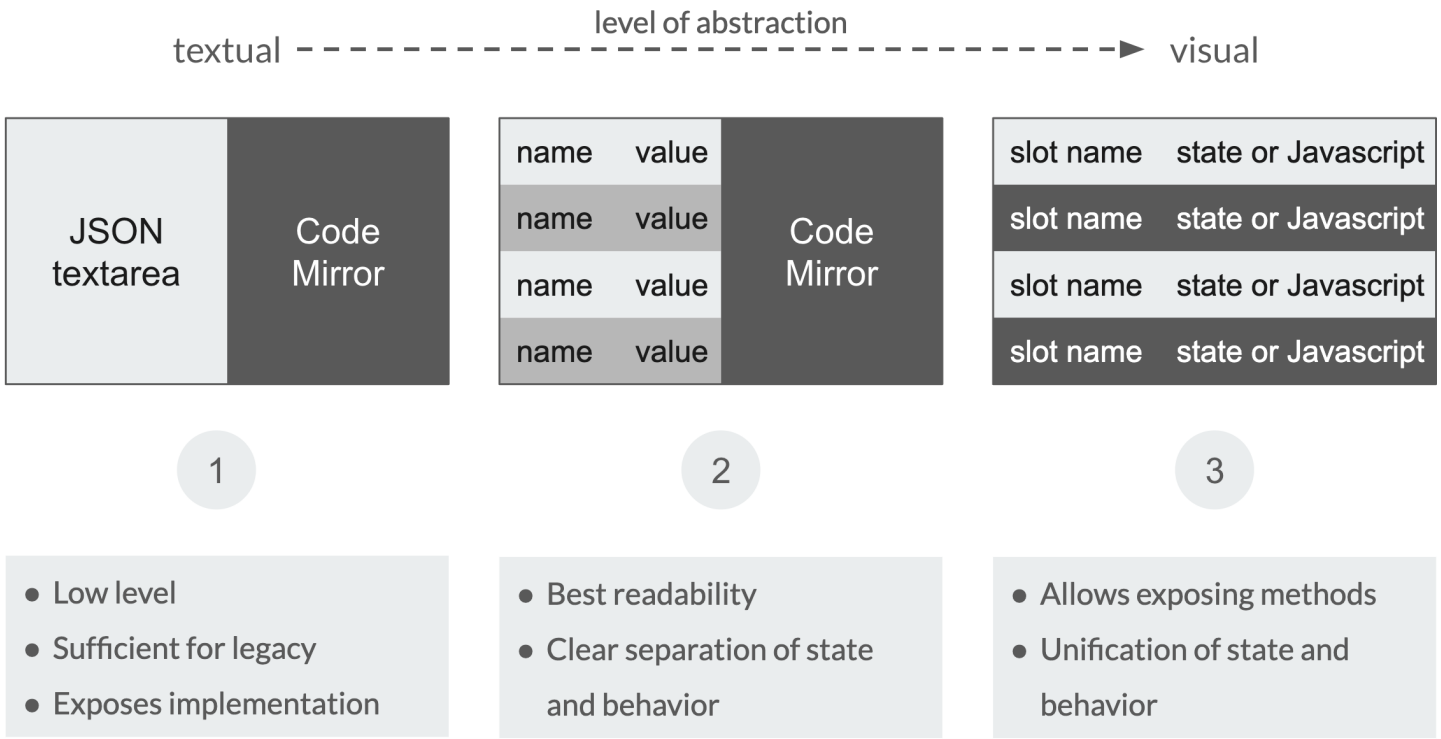
Energy simulation in Lively4

Design Improvements

Improve end user experience



Managing State and Behavior



- Low level
- Sufficient for legacy
- Exposes implementation

- Best readability
- Clear separation of state and behavior

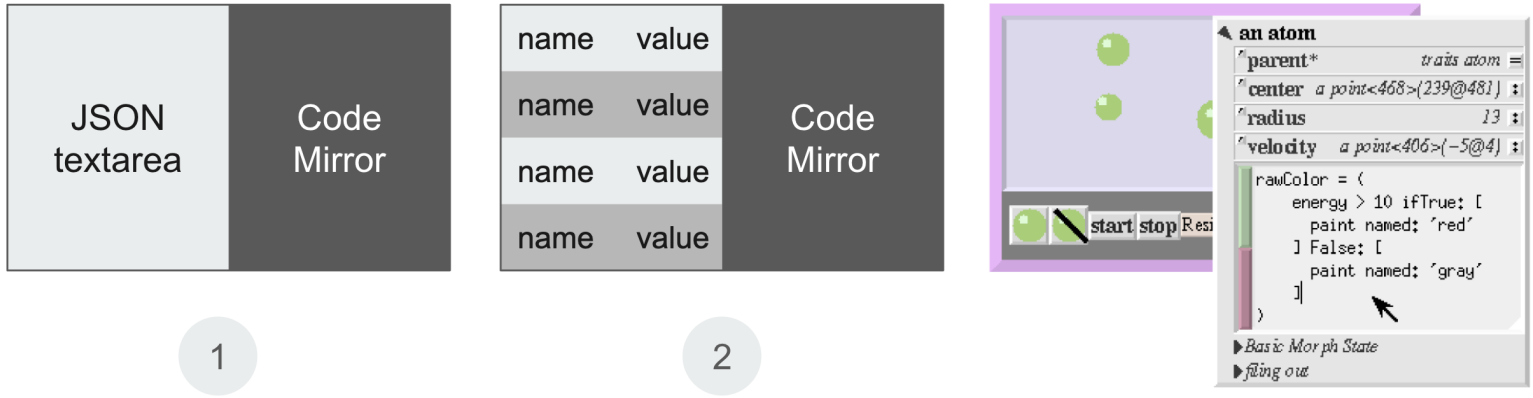
- Allows exposing methods
- Unification of state and behavior

13

Managing State and Behavior

level of abstraction

textual ----------► visual



1

- Low level
- Sufficient for legacy
- Exposes implementation

2

- Best readability
- Clear separation of state and behavior

- Allows exposing methods
- Unification of state and behavior

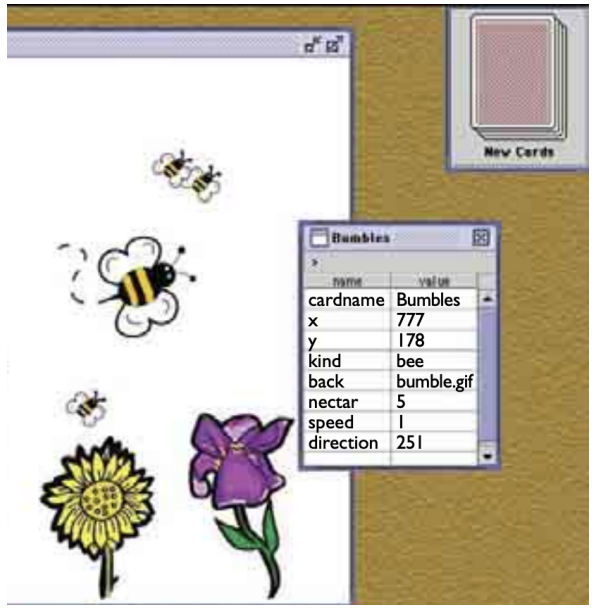
Image based on: Smith, R. B., Maloney, J., & Ungar, D. (1995, October). The Self-4.0 user interface: manifesting a system-wide vision of concreteness, uniformity, and flexibility. In Proceedings of the tenth annual conference on Object-oriented programming systems (pp. 47-60).

Cell Views

- Legacy had one centralized log/ chart
- Each cell represents one simulation unit

> *"An environment that works the way nonprogrammers expect is more inviting and helps users become more confident and productive. [2]*

→ each unit should have it's own state/ behavior, log, chart



each entity has different views [2]

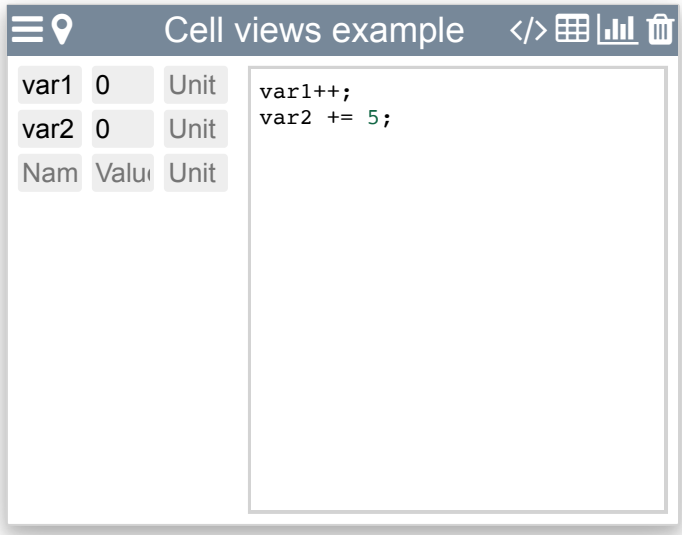
[2] Myers, B. A., Pane, J. F., & Ko, A. (2004). Natural programming languages and environments. Communications of the ACM, 47(9), 47-52.

Cell Views

- Legacy had one centralized log/ chart
- Each cell represents one simulation unit

> *An environment that works the way nonprogrammers expect is more inviting and helps users become more confident and productive. [2]*

→ each unit should have it's own state/ behavior, log, chart, **...custom view?!**



[2] Myers, B. A., Pane, J. F., & Ko, A. (2004). Natural programming languages and environments. Communications of the ACM, 47(9), 47-52.

Unit Support for DSL

Goal Allow usage of units in DSL and support automatic conversion

Legacy Add units as comment to state only

Solution Use mathjs [3] to parse the code and work with units

> *[Mathjs] features a flexible expression parser [...] and offers an integrated solution to work with [...] units, and matrices. Powerful and easy to use. [3]*

```
math.evaluate('5.08 cm + 2 inch')     // 10.16 cm
```

[3] https://mathjs.org/

Outlook: Unit Support for DSL

Goal Allow usage of units in DSL and support automatic conversion

Legacy Add units as comment to state only

~~**Solution** Use mathjs [3] to parse the code and work with units~~

> *[Mathjs] features a flexible expression parser [...] and offers an integrated solution to work with [...] units, and matrices. Powerful and easy to use. [3]*

```
math.evaluate('5.08 cm + 2 inch')     // 10.16 cm
```

Mathjs does not fully support Javascript (f.e. loops, if/else)
User would have to learn mathjs to use it

[3] https://mathjs.org/

What's Next?

Learnings 📚

- Lively integration is easier than expected
- Plenty tools & features → keeping overview is hard for newbies
- Stable internet connection required
- Dataflow for nested components is not specified

Next Steps 🏃

- Mirroring of cells

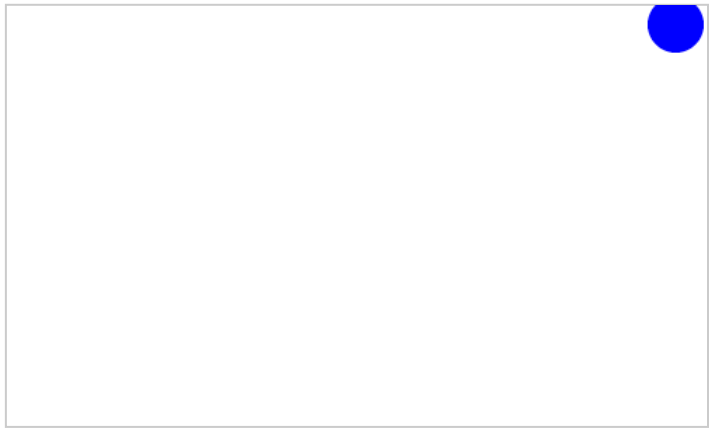
What's Next?

Future Work ➡️

→ *Gas tank simulation* shows functionality for further development

- **Variable number** of entities of a cell
- Creating a **visualization** component

Hits: 5



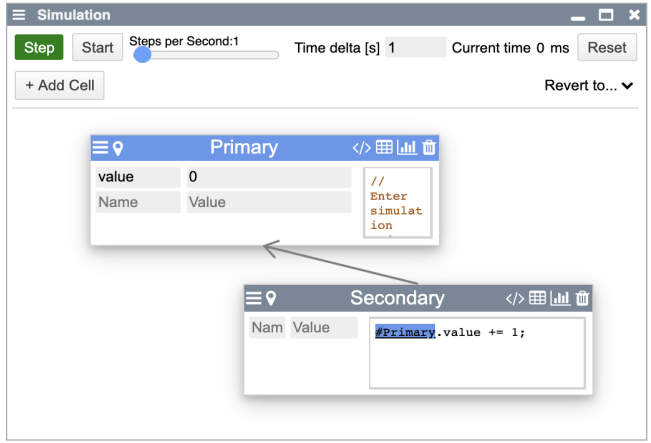
+ -

Project Goal

We want to enable end-users to create simulations in a visually structured way

Idea Build a semi-graphical simulation framework

- *Starting point:* PoC* in Lively Webwerkstatt (legacy)
- Allow documentation to be part of simulation
- Provide a way that allows simulation result investigation
- Proper Lively integration
- Port existing *energy simulation* from legacy



Simulation Framework