Languages & Visualizations to Enable Effective End User Programming

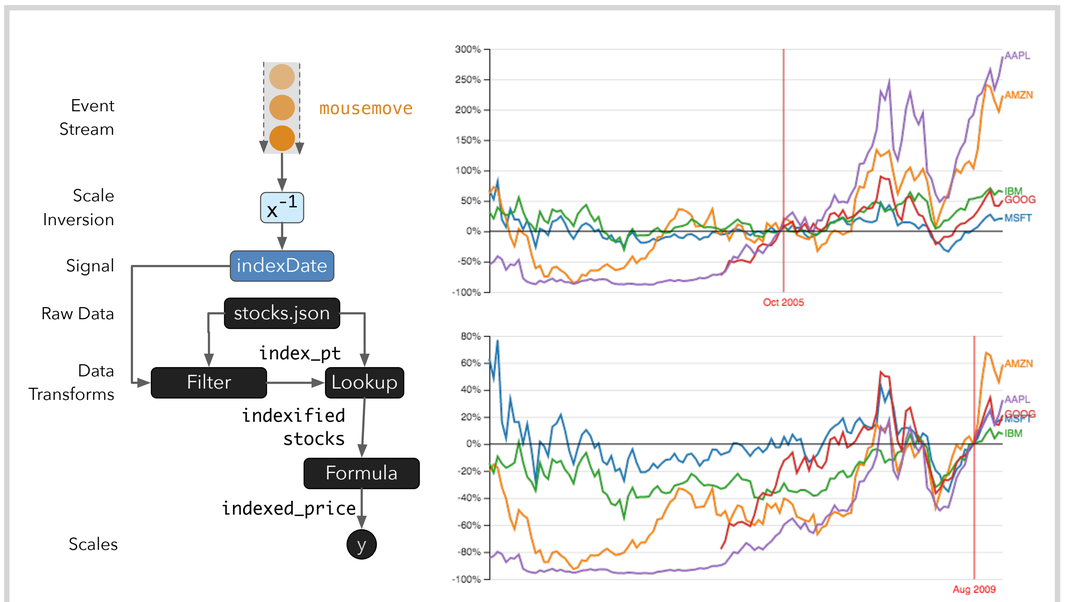
Jane Hoffswell

Declarative programming languages raise the level of abstraction allowing users to focus on relevant domain-specific details, while deferring implementation details to the underlying system. **Visualizing program behavior** promotes program understanding to reduce the separation between user specification and system implementation.

New declarative, domain-specific languages

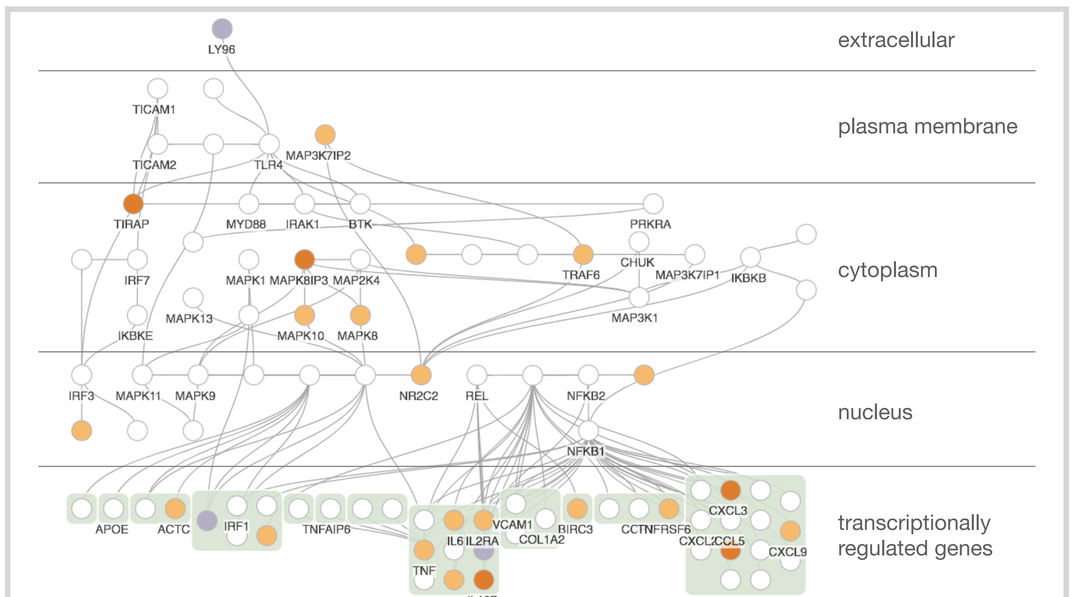
Reactive Vega

Streaming Dataflow Architecture for Declarative Interactive Visualization



SetCoLa

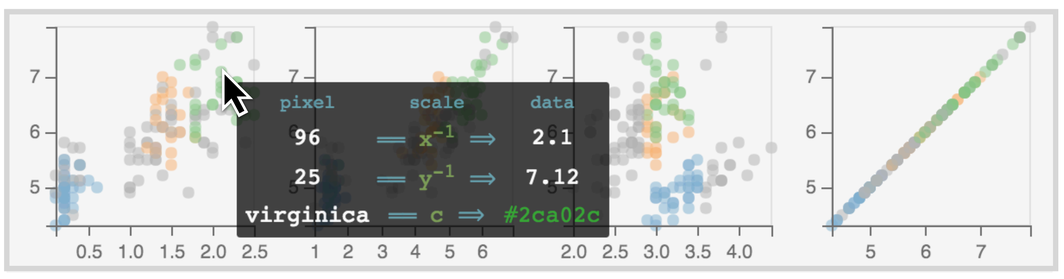
High-Level Constraints for Graph Layout



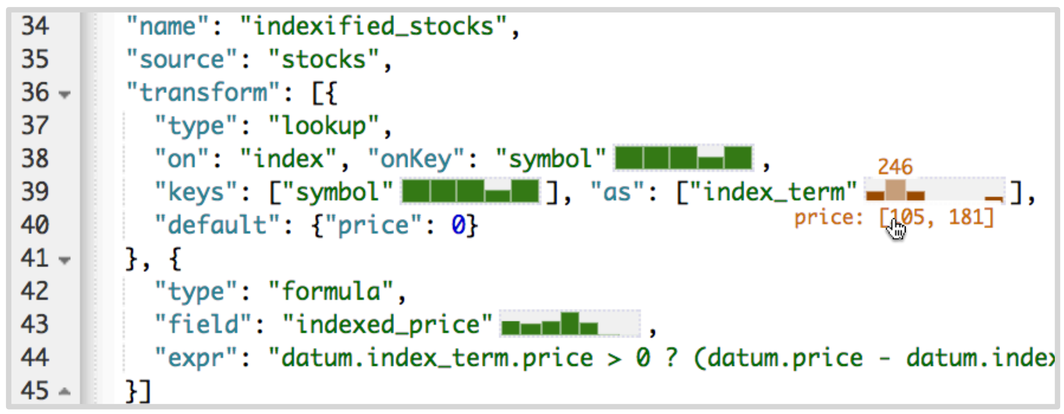
New program understanding techniques

Understanding Reactivity

Visual Debugging Techniques for Reactive Data Visualization



Augmenting Code with In Situ Visualizations to Aid Program Understanding



Understanding Constraints

Do you work with constraints? Let's chat!

Does the output satisfy all of my constraints?

Do the constraints produce the output I expected?

What is missing? What constraints do I need to add to produce the output that I want?

What impact does a given constraint have?

What happens if I remove this constraint?

What was my rationale for adding this constraint?

How do I correctly encode my intent as a constraint?

PAUL G. ALLEN SCHOOL OF COMPUTER SCIENCE & ENGINEERING