Need
Simple, intuitive utility to look around locally in large directed graphs.
Surprisingly, we couldn't find one. Had to write one.

Key layout requirements
Lay out only a small local subgraph.
Readable exploration, not global structure.
Change subgraph as you move into new territory.
But not force-directed layout adjustment.
Dynamic version of Sugiyama-style algorithm.
Directed edges always start & end flowing down.
Hypergraph edges merge inputs & split outputs.
Preserve sibling order if semantically meaningful.
Responsive and stable.
Animate first toward quick layout.
Change course when final layout arrives.
Don’t change layout too much: favor stability.
Smooth animation and fade to help visual tracking.

Topological keyboard navigation
Use arrow keys to let user find the paths & hyperpaths through the tangles.

Why browse large hypergraphs?
Declarative or omniscient debugging
We built a declarative programming language (Dyna).
Tracing execution would be confusing & irrelevant—the order of computations is up to the compiler.
Instead, we wanted debugging to let you explore where a value came from and how it was used.

Producer-consumer networks
Family trees
Digital circuits
Chemical reactions
Flow of manufacturing materials
Proof forests (from theorem provers)
Parse forests (from natural-language parsers)
Multiple inputs combine into reusable output(s).

Ordinary large graphs
Ever wanted to use dot on a large or dense graph?
Trees or near-trees, social and physical networks,
data structure layout, finite-state automata, call graphs…
Dyna is "browseable dot."

Asynchronous dynamic layers
Dynamic application graph (∞)
User coloring, contraction,… (∞)
Navigation (∞)
Visible subgraph
Delayed layout
Slow animation to layout

Visual Navigation through Large Directed Graphs and Hypergraphs
Johns Hopkins University – the “Dynasty” project – http://dyna.org/Dynasty

This work has been supported in part by NSF ITR grant IIS-0313193 (“Weighted Dynamic Programming for Statistical Natural Language Processing”) to the first author and by Joseph C. Pistritto Research authors’ only. We also thank Jeffrey Heer for the prefuse toolkit.

IEEE Symposium on Information Visualization, October 2006.