Hierarchy Visualization
By
Christian Chita

Table of Contents:
1. ForEach (Paper) Do {
   a. Problem Addressed and Knowledge Gap
   b. Key Issues
   c. Implementation
   d. The Good
   e. The Not So Good
}  
2. Synthesis: ForEach (Paper) Do {
   a. Assumptions behind each
   b. Did they solve the problem?
}

Papers Surveyed:

I. ConeTrees paper:

Problem Addressed:
Managing and Accessing large information spaces

Knowledge Gap:
Once information is displayed, how are the various parts related?

How to alleviate the currently high cognitive load?

I. ConeTrees paper Key Issues:

Issue:
• 2D layouts of complex structures will not fit onto the screen
• 2D invariably leads to scrolling/zoom-out

Solution:
• Use 3D
• Use animation to reduce Cognitive Load

I. ConeTrees paper Implementation:

Uses an Information Visualizer as engine

Supports:
• Continuous rotation for structure analysis
• Smooth interactive animation
• Mechanisms for 3D navigation

Figure 1: Aspect of 3D and 2D Trees. Levels Displayed Correctly

Robertson Plate 5
I. ConeTrees paper: The Good

- No need for special equipment
- Fish eye view by default
- Shadow provides added structure info without the user even noticing/focusing it
- Prune and Grow ops
- Search handled by other process (allows user to continue work)
- Bottom line: get all of the above // reduction in cognitive load

I. ConeTrees paper: The Not So Good

- Criticizes previous work, but input data is different in the two cases // they solve different problems
- Questionable structure-segment partition: too much focus on symmetry:
  - Is this what the users want?
- Contradictions with self(?):
  - User is allowed to continue work // BUT “when a search starts, all nodes are made invisible”

Cone Trees Paper:

- Uniform structure
- Non-symmetric structure

II. MultiTrees Paper:

- Problem Addressed:
  - Common trees have shortcomings:
    - Only one way to go from node A to node B
    - No multiple organizing contexts
- Problem Addressed:
  - DAGs have shortcomings:
    - Edge crossing even for small neighbourhoods

II. MultiTrees Paper:

- Problem Addressed:
  - Hierarchical structure aggregate scale
- Problem Addressed:
  - Hierarchical structure reuse

Need a new type of structure to represent info: a

II. MultiTrees Paper: Knowledge Gap

A Collection // ISA hierarchy with shared subtrees

Tree

DAG

Tree Unordered Collection

Monolith Structure
II. MultiTrees Paper: Key Issue

- Focused on following facts:
  - From any node:
    - ![Look Up](see diverse hierarchical quadrant)
    - ![Look Down](see content under a node
  - ![Tree of Context](a tree of content)

II. MultiTrees Paper: Implementation

Figure A: Central view of the multirees both upon upcase link information repository. The view is centered on the node named "Professor" and shows the tree of ancestors (by the left) and the tree of descendants (on the right) of this node.

Figure B: Integrated format view for a real genealogy. This view shows both branching and downward family trees of a person, as the essential format of how genealogies or family trees are constructed.

II. MultiTrees Paper: The Good

- Excellent theoretical background and analysis of proposed solution

Proposition 1: The following properties are equivalent:
- The data can be represented by an ordered hypergraph in which nodes represent individuals and edges represent relationships.
- The data is represented as a tree.
- The data is represented as a directed acyclic graph (DAG).

Proposition 2: Consider any tree with 24 as a node, and let the necessary unique path connecting them. The vector of all the ancestors of this path and all the descendants of this path is a topological vector.

II. MultiTrees Paper: The Not So Good

- How many roots can we fit in one view?
- Allows reuse out of context?
- Must be constructed by hand
- No user testing
- However, all pointed out by the authors themselves (except last 😊)
III. Polyarchies paper: 

**Problem:**
- Understand relationships behind multiple data bases
- How to viz a metadirectory?

**Knowledge Gap:**
- Current viz techniques do not allow simultaneous view of relationships along $\geq 1$ dimension

---

III. Polyarchies paper **Key issues:**

**Issue:**
- How to view inter-relation between separate entities?

**Solution:**
- Only show parts of the parent hierarchy (not global relationship)

---

III. Polyarchies paper **Key issues:**

**Problem:**
- How to move from one hierarchy to another without loosing context?

**Solution:**
- Use animation to reduce Cognitive Load

---

III. Polyarchies paper: **Implementation**

---

III. Hierarchy Switch – **Variant:**

---

Figure 1. Polyarchies paper showing relationship of the management hierarchy.

Figure 2. Visual Pivot rotation animation; a sequence of frames starting with the management view and ending with the business unit view around pivot point “Andrew Dixon”. 

Figure 5. Stack Linked style showing three hierarchy views.
### I. Assumptions Behind ConeTrees paper:

- Only visualizes hierarchical information structures, not arbitrary graphs.
  - i.e. No structure, No Viz

### Part 2: Synthesis

**Foreach (Paper) Do {**

- Assumptions behind each implementation
- Did they solve the problem?

**}**

### III. Polyarchies paper: The Not So Good

- **Quotes***:
  1. MultiTrees are multiple hierarchies with shared subtrees.
  2. Box Polyarchies are multiple intersecting hierarchies, sharing at least one node rather than sharing subtrees.
  3. Hence, MultiTrees are a subset of polyarchies.
  4. The added complexity requires a new approach as described in the paper.

### III. Polyarchies paper: The Not So Good

- How did they figured out:
  - Counting item storage in short-term memory (STM)
  - Number of comparisons with items in STM
  - Designer decides and has complete control over:
    - Which database to include
    - Which hierarchies to expose

### III. Polyarchies paper: The Good

- Used a Flash prototype first.
- Excellent formal user study (5 of them)
- Allows users to choose animation speeds.
- Good survey of previous work.
  - But it confirms their own findings
I. Did They Solve the Problem?
[ConTree paper:]
- Future Work leftovers:
  - From 10 refs, 5 are to self
    - Earliest: 1986
    - Latest (the wall): 1991
    - Progress: ‘86, ‘89, ‘90, ‘91
- Bottom line: (plenty of time to do formal user testing)
  - (plenty of time to infer ecologically valid task)

II. Assumptions Behind MultiTrees paper:
- No diamonds
- Why:
  - People will want to store same node in more than one structure OR
  - Two paths below one node

Figure 2. Diamonds are not permitted in a multitree. A diamond occurs when two distinct directed paths occur between a pair of nodes. Thus is a multitree at most one directed path can exist between two nodes

II. Did They Solve the Problem?
[MultiTrees paper:]
- What problem are we talking about?
  - They solved the problem they started up to solve
  - How well was the Viz done?
  - In doing so, they inferred a new data structure
  - Bottom line: WW research community benefits from their work

III. Assumptions Behind Polyarchy Viz paper:
- Designer decides and has complete control over:
  - Which database to include
  - Which hierarchies to expose
  - Candidate search attributes
- MS-only hardware/software:
  - Uses PQL
  - Uses Polyarchy Query Server
  - Uses MS Metadirectory Services

III. Did They Solve the Problem?
[Polyarchy Viz paper:]
- What problem are we talking about?
  - Problem(s) solved:
    - Allow user to see relationships between hierarchies in the context of selected nodes
    - Allow user to see relationships between multiple entities within a hierarchy

III. Did They Solve the Problem?
[Polyarchy Viz paper:]
- Why only MS, HR data set?
  - How about:
    - Stock Markets
    - Human Genome
    - Biomed data in general
- Why so much self-praise for PQL?
  - “rich QL, allowing enormous flexibility for exploration”
- Perhaps a slightly self-centered approach?
Summary:

1.ForEach (Paper) Do {
   a) Problem Addressed and Knowledge Gap
   b) Key Issues
   c) Implementation
   d) The Good
   e) The Not So Good
}

2. Synthesis: ForEach (Paper) Do {
   a) Assumptions behind each
   b) Did they solve the problem?
}