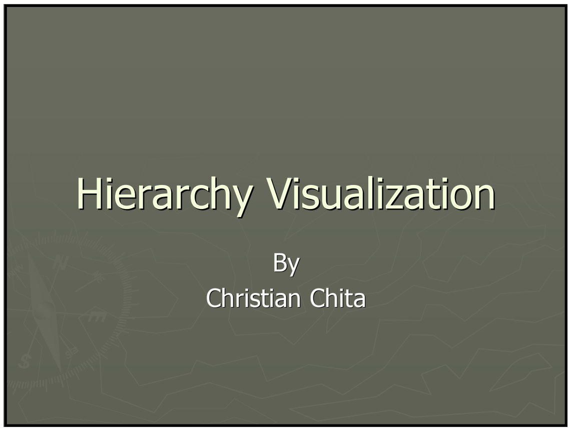
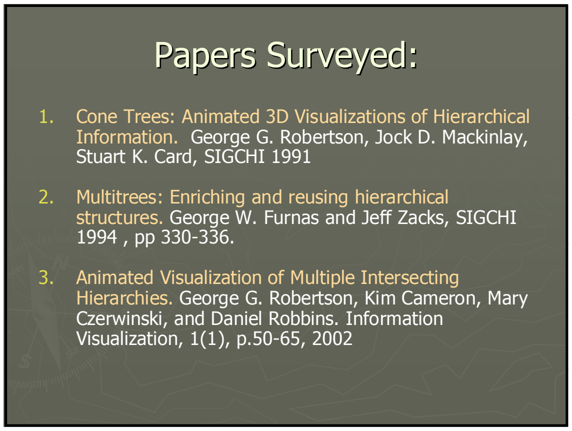


Hierarchy Visualization

By
Christian Chita



Papers Surveyed:

1. Cone Trees: Animated 3D Visualizations of Hierarchical Information. George G. Robertson, Jock D. Mackinlay, Stuart K. Card, SIGCHI 1991
2. Multitrees: Enriching and reusing hierarchical structures. George W. Furnas and Jeff Zacks, SIGCHI 1994 , pp 330-336.
3. Animated Visualization of Multiple Intersecting Hierarchies. George G. Robertson, Kim Cameron, Mary Czerwinski, and Daniel Robbins. Information Visualization, 1(1), p.50-65, 2002

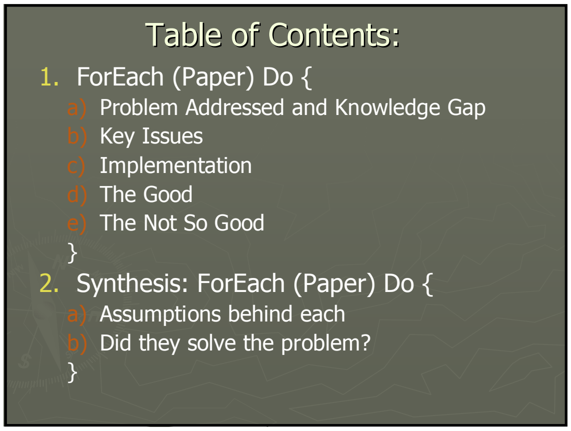
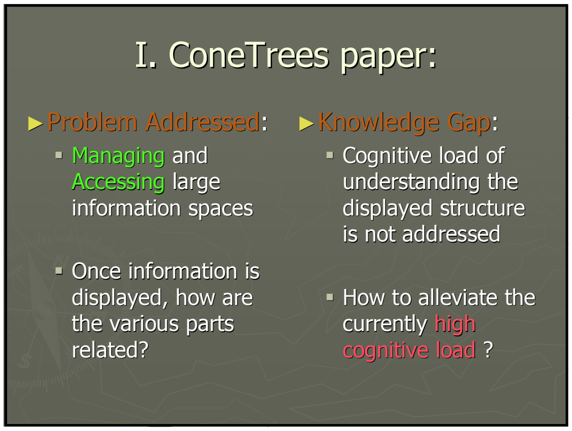


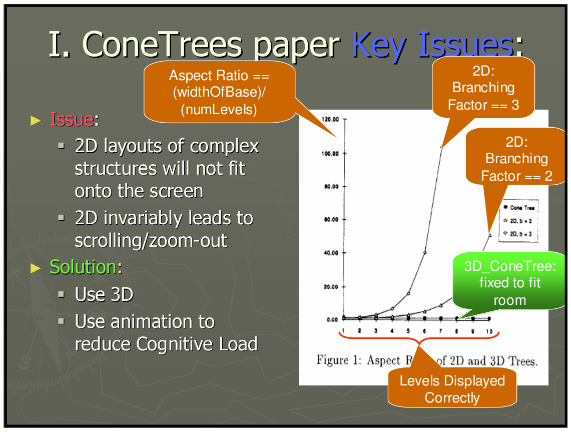
Table of Contents:

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?
   }



I. ConeTrees paper:

- Problem Addressed:
  - Managing and Accessing large information spaces
  - Once information is displayed, how are the various parts related?
- Knowledge Gap:
  - Cognitive load of understanding the displayed structure is not addressed
  - 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

Figure 1: Aspect R of 2D and 3D Trees.

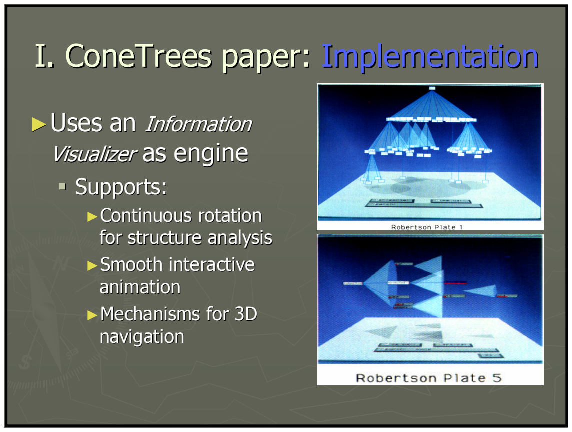
Aspect Ratio == (widthOfBase)/ (numLevels)

2D: Branching Factor == 3

2D: Branching Factor == 2

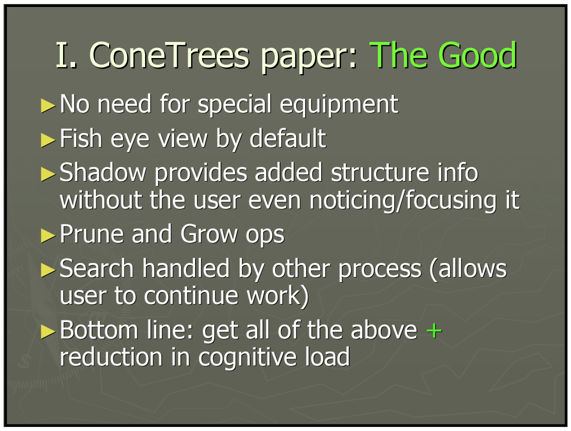
3D_ConeTree: fixed to fit room

Levels Displayed Correctly



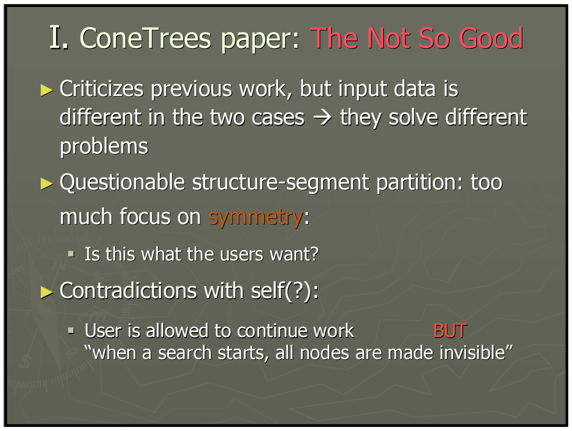
I. ConeTrees paper: Implementation

- Uses an *Information Visualizer* as engine
  - Supports:
    - Continuous rotation for structure analysis
    - Smooth interactive animation
    - Mechanisms for 3D navigation



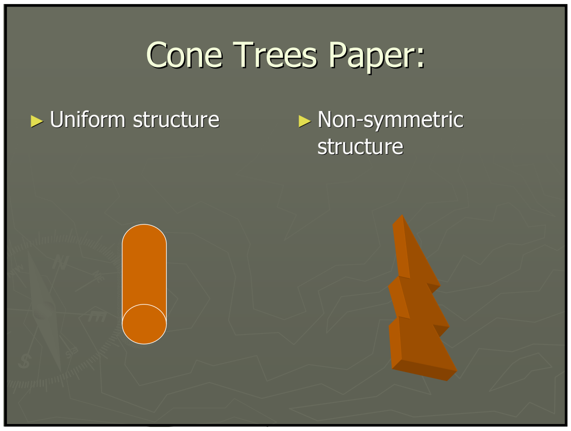
I. ConeTrees paper: The Good

- 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



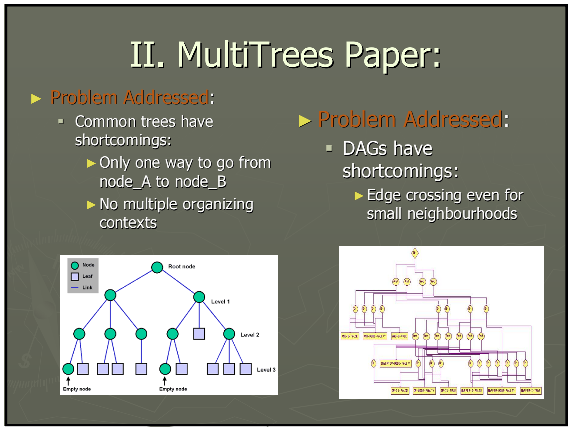
I. ConeTrees paper: The Not So Good

- 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"



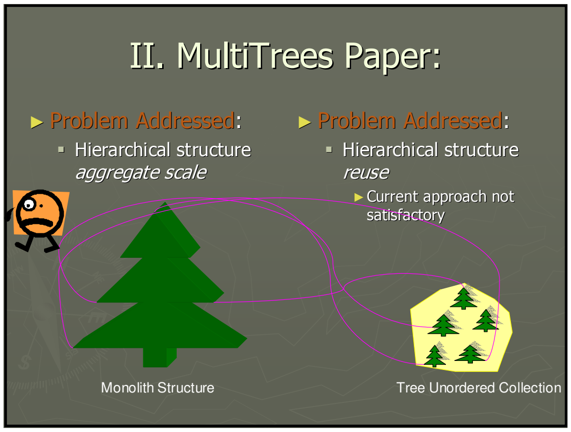
Cone Trees Paper:

- Uniform structure
- Non-symmetric structure



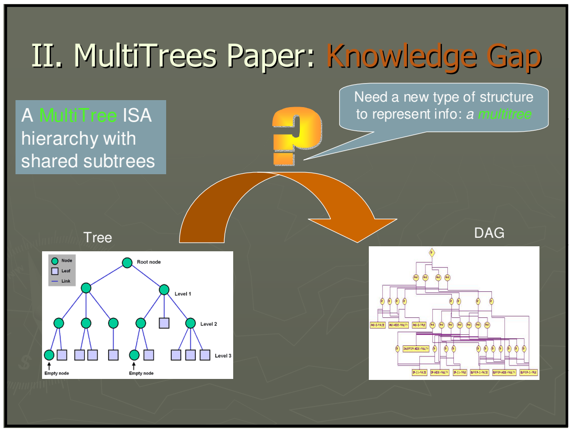
II. MultiTrees Paper:

- 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:

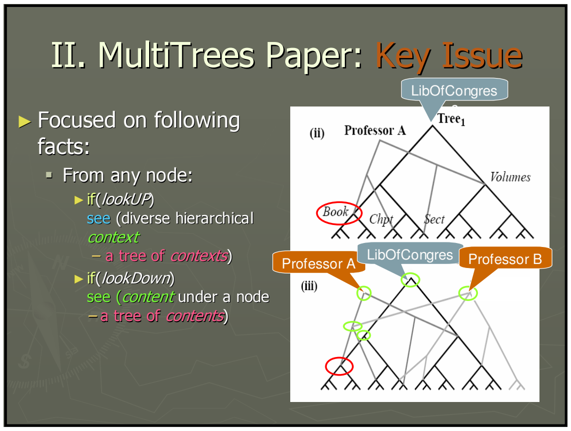
- Problem Addressed:
  - Hierarchical structure *aggregate scale*
- Problem Addressed:
  - Hierarchical structure *reuse*
    - Current approach not satisfactory



II. MultiTrees Paper: Knowledge Gap

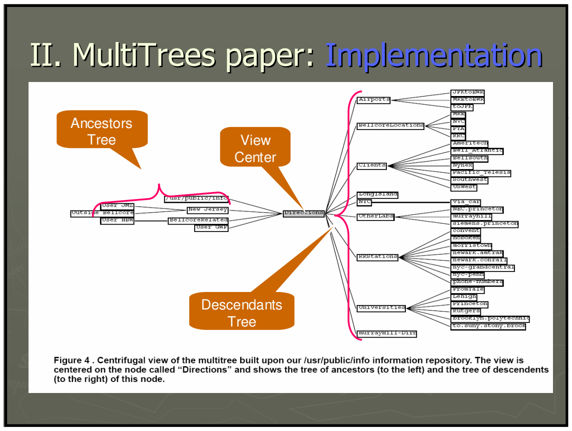
A MultiTree ISA hierarchy with shared subtrees

Need a new type of structure to represent info: *a multitree*



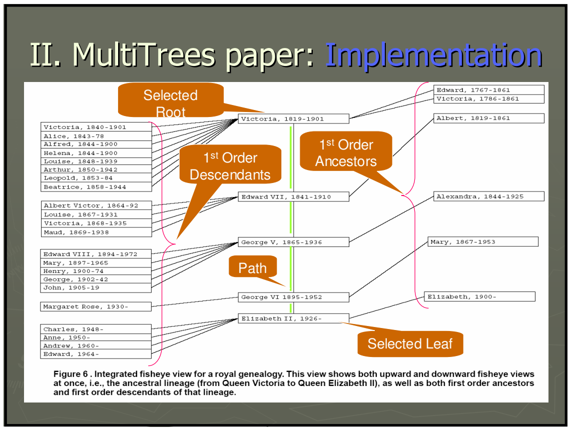
II. MultiTrees Paper: Key Issue

- Focused on following facts:
  - From any node:
    - if(*lookUP*) see (diverse hierarchical *context* – a tree of *contexts*)
    - if(*lookDown*) see (*content* under a node – a tree of *contents*)



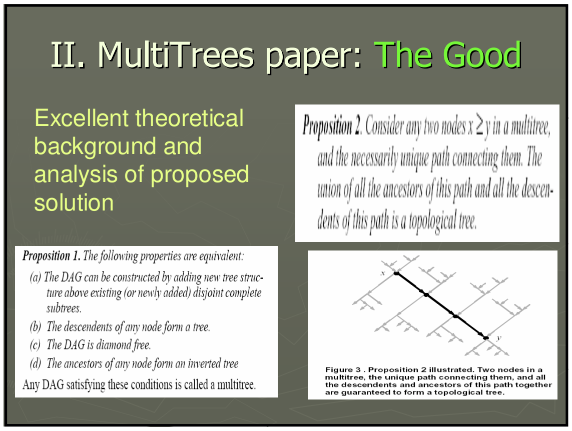
II. MultiTrees paper: Implementation

Figure 4 . Centrifugal view of the multitree built upon our /usr/public/info information repository. The view is centered on the node called “Directions” and shows the tree of ancestors (to the left) and the tree of descendents (to the right) of this node.



II. MultiTrees paper: Implementation

Figure 6 . Integrated fisheye view for a royal genealogy. This view shows both upward and downward fisheye views at once, i.e., the ancestral lineage (from Queen Victoria to Queen Elizabeth II), as well as both first order ancestors and first order descendants of that lineage.



II. MultiTrees paper: The Good

Excellent theoretical background and analysis of proposed solution

Proposition 2. *Consider any two nodes $x \geq y$ in a multitree, and the necessarily unique path connecting them. The union of all the ancestors of this path and all the descendents of this path is a topological tree.*

Proposition 1. *The following properties are equivalent:*

(a) The DAG can be constructed by adding new tree structure above existing (or newly added) disjoint complete subtrees.

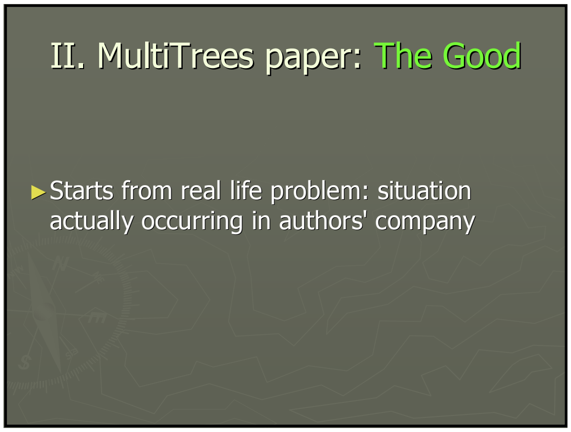
(b) The descendents of any node form a tree.

(c) The DAG is diamond free.

(d) The ancestors of any node form an inverted tree

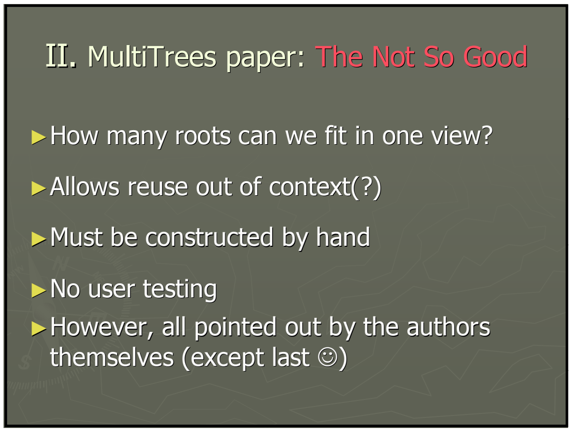
Any DAG satisfying these conditions is called a multitree.

Figure 3 . Proposition 2 illustrated. Two nodes in a multitree, the unique path connecting them, and all the descendents and ancestors of this path together are guaranteed to form a topological tree.



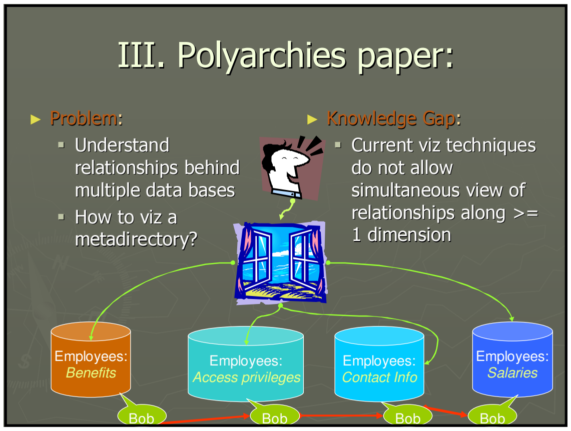
II. MultiTrees paper: The Good

- Starts from real life problem: situation actually occurring in authors' company



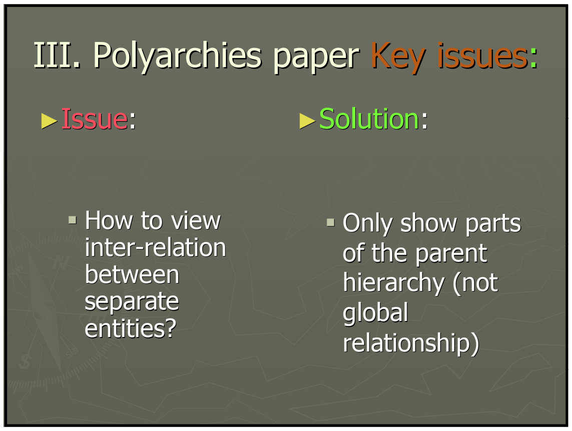
II. MultiTrees paper: The Not So Good

- 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 ☺)



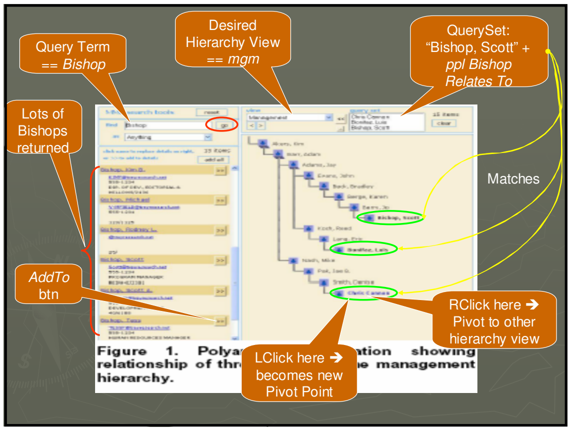
III. Polyarchies paper:

- 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 >= 1 dimension



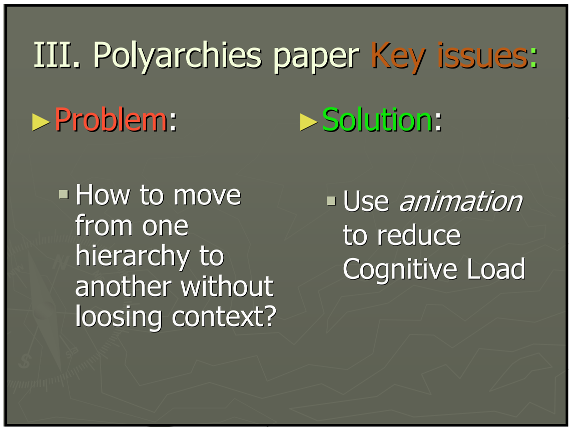
III. Polyarchies paper Key issues:

- Issue:
  - How to view inter-relation between separate entities?
- Solution:
  - Only show parts of the parent hierarchy (not global relationship)



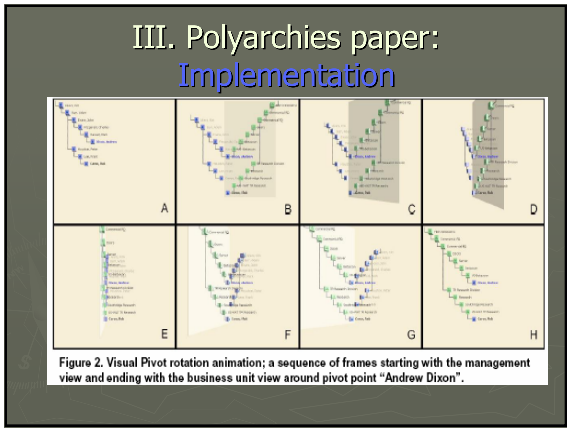
Query Term == *Bishop*
Desired Hierarchy View == *mgm*
QuerySet: “Bishop, Scott” + *ppl Bishop Relates To*
Lots of Bishops returned
Matches
AddTo btn
RClick here → Pivot to other hierarchy view
LClick here → becomes new Pivot Point

Figure 1. Polya… relationship of thr… …ation showing …e management hierarchy.



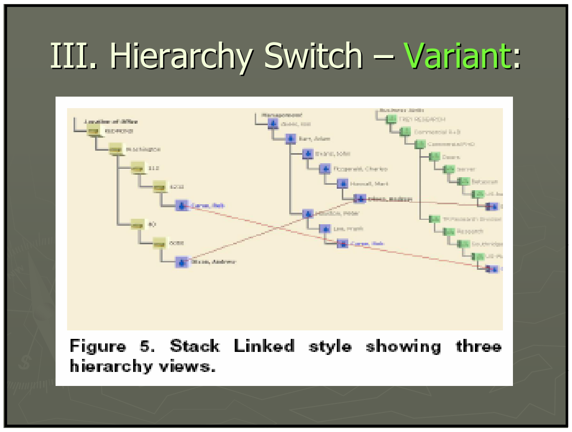
III. Polyarchies paper Key issues:

- Problem:
  - How to move from one hierarchy to another without loosing context?
- Solution:
  - Use *animation* to reduce Cognitive Load



III. Polyarchies paper: Implementation

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”.



III. Hierarchy Switch – Variant:

Figure 5. Stack Linked style showing three hierarchy views.

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

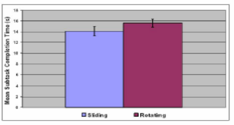


Figure 8. Study 4: Mean subtask completion time for sliding versus horizontal rotation.

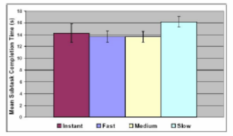


Figure 10. Study 4: Mean subtask completion time versus animation speeds.

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

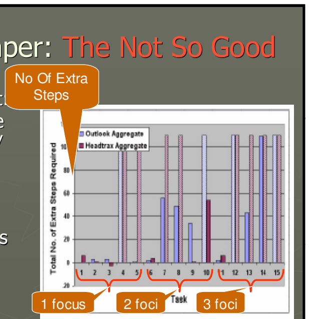


Figure 11. Extra behavioral steps required for each of 15 tasks. The crosshatched bars are cases where the task could not be done (4 tasks for Outlook and 8 for Headtrax).

III. Polyarchies paper: The Not So Good

- Quotes "":
  1. MultiTrees are multiple hierarchies with shared subtrees.
  2. But 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 this paper.

Part 2: Synthesis

- Foreach (Paper) Do {

  a) Assumptions behind each implementation

  b) Did they solve the problem?

  }

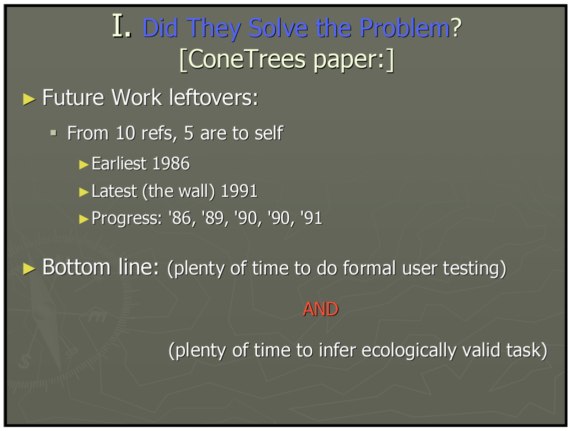
I. Assumptions Behind ConeTrees paper:

- Only visualizes hierarchical information structures; not arbitrary graphs
  - i.e. No structure, No Viz
- Future work will solve:
  - Gains of 3D layout
  - If 3D maximizes use of screen space
  - What other organizations can be usefully displayed by Cone Trees
  - What graphs can or cannot be displayed by Cone Trees

I. Did They Solve the Problem?

[ConeTrees paper:]

- What problem are we talking about?
  - Problem(s) solved:
    1. "Show the entire _structure_ of a complex organization in one viz"
    2. Shift part of cognitive load to perceptual system
  - Paper quote: "[…] this is the first time the organization chart could be seen in one visualization."
(Xerox Corp 650 executives – requires 80 pages)

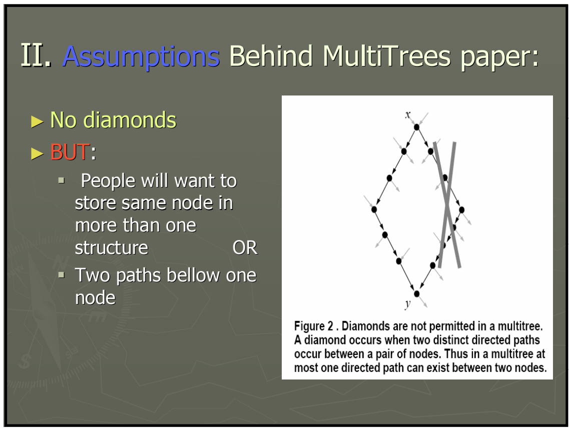


I. Did They Solve the Problem? [ConeTrees paper:]

- Future Work leftovers:
  - From 10 refs, 5 are to self
    - Earliest 1986
    - Latest (the wall) 1991
    - Progress: '86, '89, '90, '90, '91
- Bottom line: (plenty of time to do formal user testing)

AND

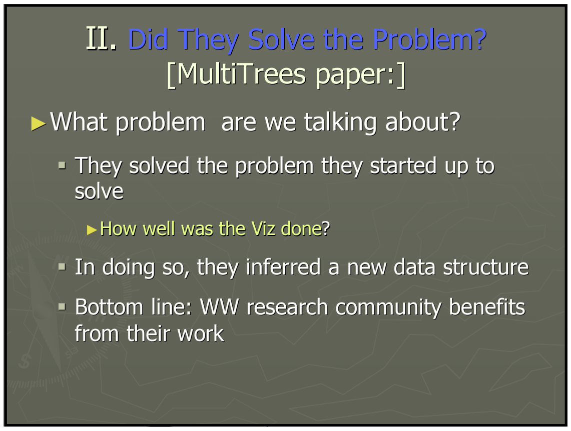
(plenty of time to infer ecologically valid task)



II. Assumptions Behind MultiTrees paper:

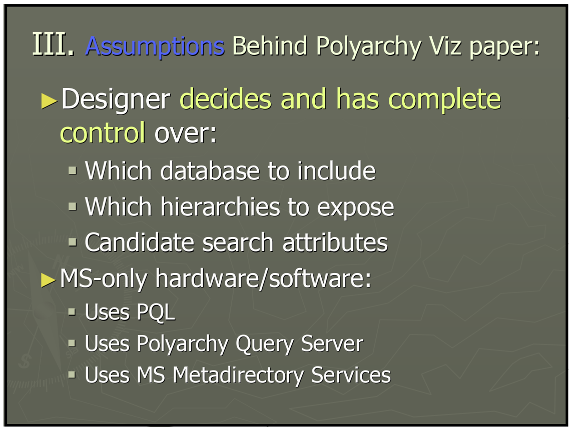
- No diamonds
- BUT:
  - People will want to store same node in more than one structure OR
  - Two paths bellow one node

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



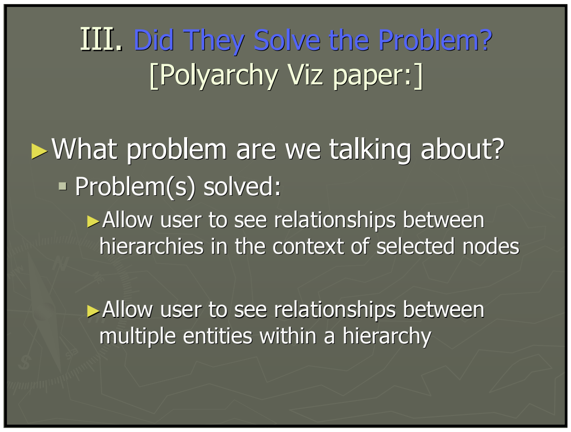
II. Did They Solve the Problem? [MultiTrees paper:]

- 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



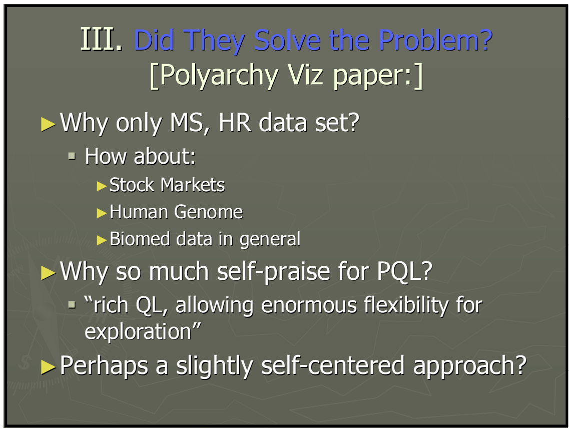
III. Assumptions Behind Polyarchy Viz paper:

- 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



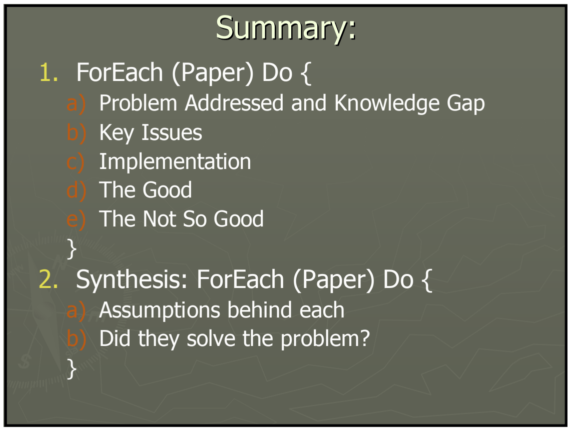
III. Did They Solve the Problem? [Polyarchy Viz paper:]

- 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



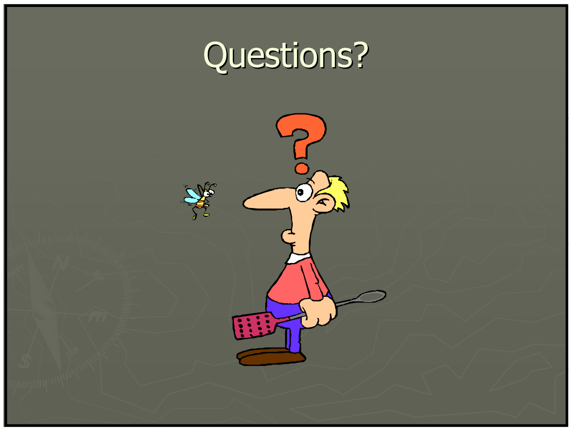
III. Did They Solve the Problem? [Polyarchy Viz paper:]

- 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?
   }



Questions?