Distributed Computing and Systems
Chalmers university of technology

Growing Squares: Animated Visualization of Causal Relations

Niklas Elmqvist (*elm@cs.chalmers.se*)
Philippas Tsigas (*tsigas@cs.chalmers.se*)

ACM Conference on Software Visualization 2003
June 11th to 13th, San Diego

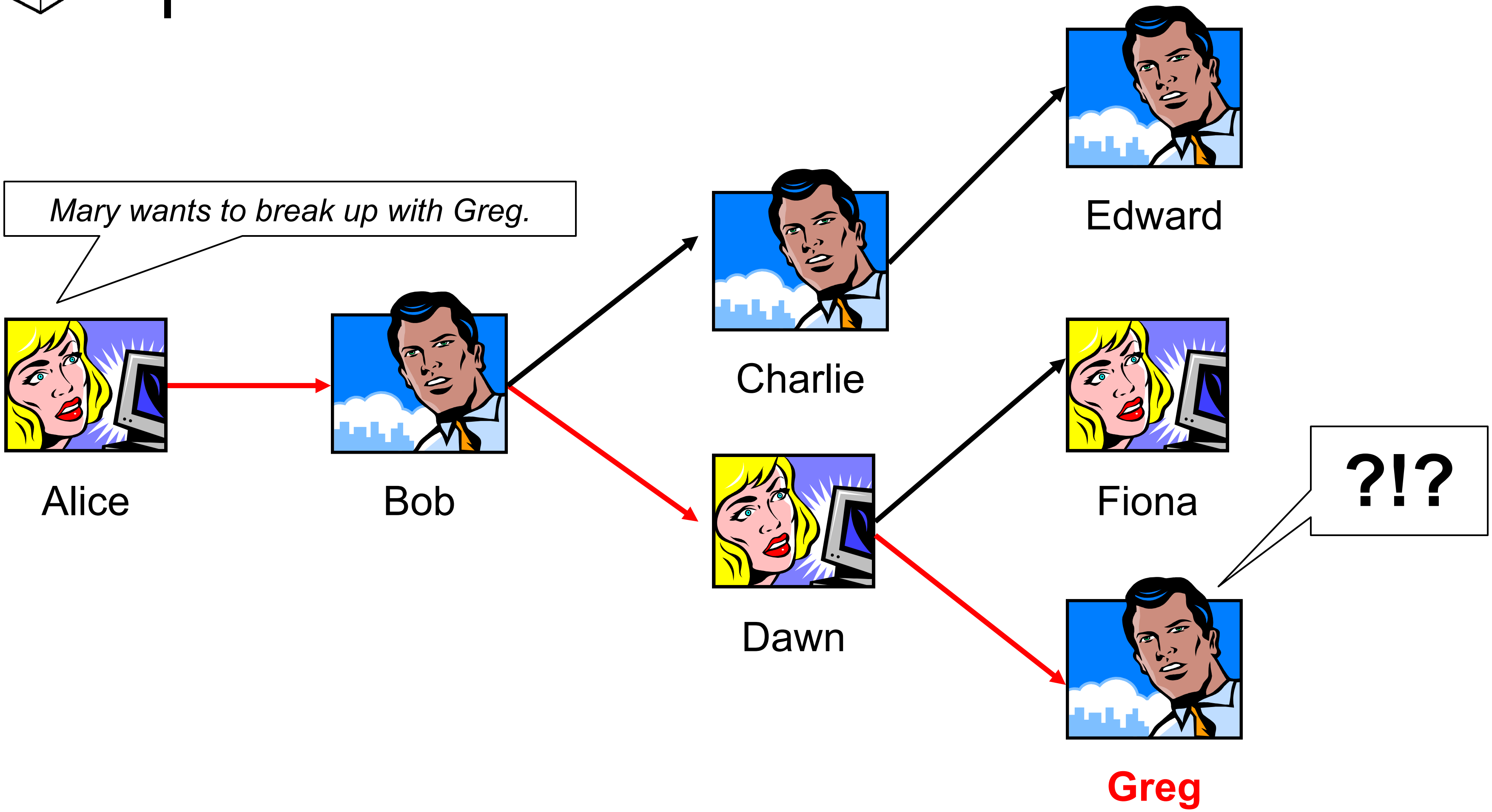
Outline

- Motivating Example (aka Awful-Clipart Slide)
- Problem Statement
- Growing Squares
- User Study
- Results & Discussion
- Conclusions & Future Work

Example: Rumor Spreading

- Suppose Alice knows a specific piece of information (a **rumor**)
  - **Example:** *Mary wants to break up with Greg*
- If we observe which people she speaks to, we can deduce which people *potentially* learns the rumor
- These observations can be used to deduce the **information flow** in the system of people

Example: Rumor Spreading (2)



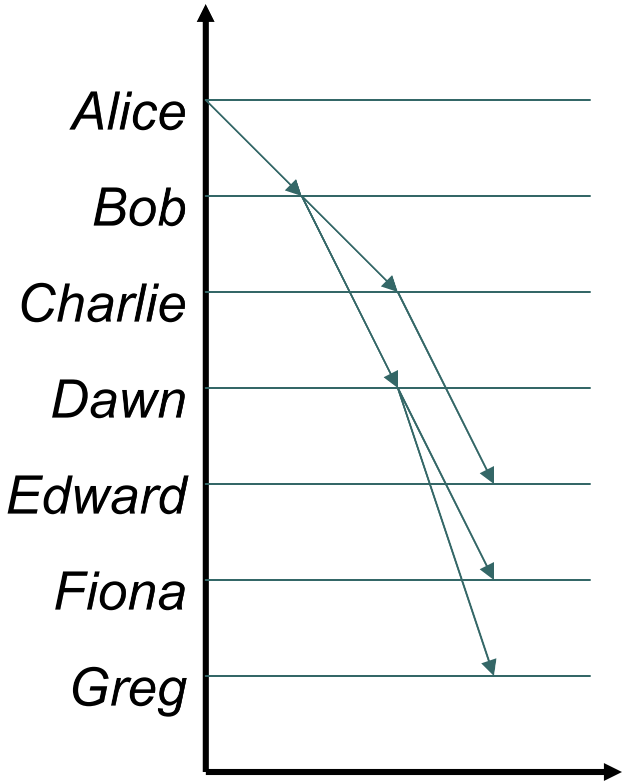
Example: Rumor Spreading (3)

- We want to be able to visualize this rumor spreading system
- In the general case, each of the participants (**processes**) have their own information that is propagated (i.e. not a single rumor)
- Visualizing this is a difficult problem

Causal Relation Visualization

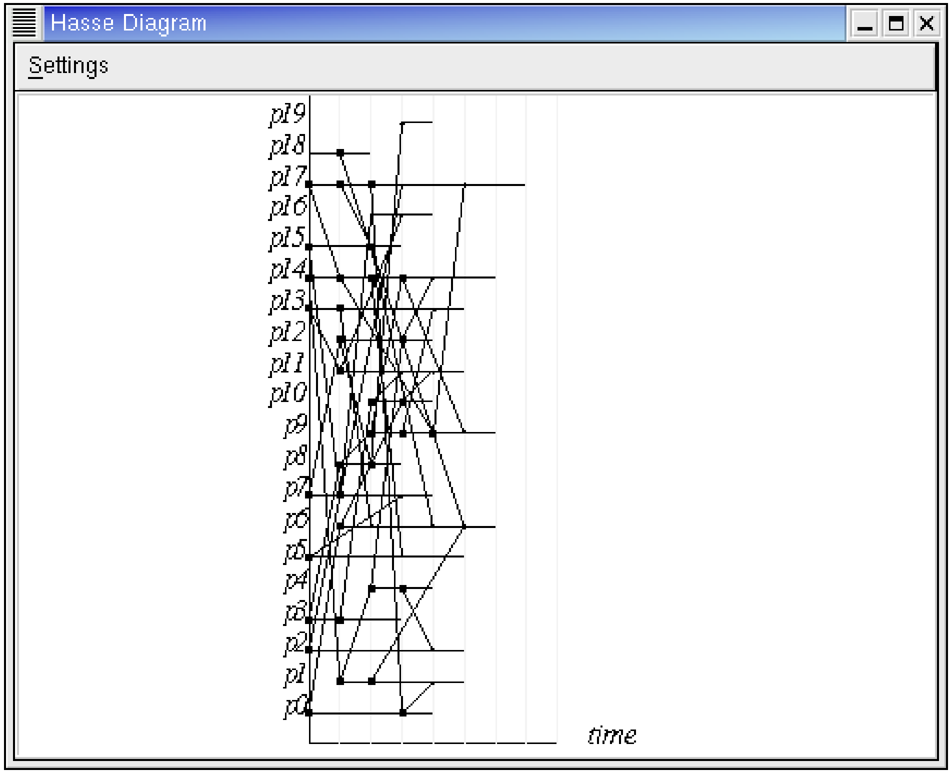
- Formally, we are looking to visualize systems of *causal relations*
  - **Def**: The *causal relation* $\rightarrow$ is a relation that connects two elements (*events*) x and y as $x \rightarrow y$ iff x is the cause of y.
  - Sets of *events* are called processes $P_1, \ldots, P_N$
    - *Internal events* are sequential and causally related
    - *External events* interconnect processes through *messages*
- We want to visualize a graph of such mutual causal dependencies between events
  - **Example**: Hasse diagrams (next slide)

Hasse Diagrams



- **Hasse** diagrams use a straightforward approach
  - Each process has a lifeline
  - Messages are represented as arrows between lifelines
- This example is fairly simple and structured
  - Well-suited to Hasse diagrams

Hasse Diagrams: Example



- Distributed system with *n=20* processes and 60 system events
  - Not unrealistic situation
- Difficult to comprehend
  - Intersecting and coinciding message arrows
  - Fine granularity

Hasse Diagrams: Problems

- Some problems with Hasse diagrams:
  - Works fine for smaller systems
  - Fine granularity make them unsuitable to larger systems of causal relations
  - The user must manually maintain ”the context” of the relations
    - Users may have to backtrace every single message to get a clear picture of the system
    - Vital information is scattered

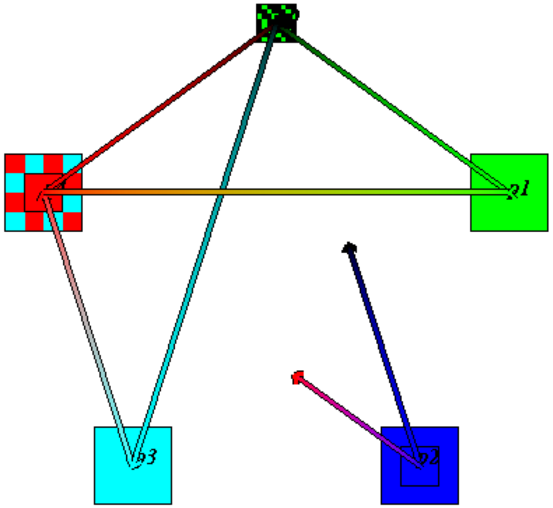
Problem Statement

- **Problem**: Visualizing the information flow and causal relations in a system of communicating processes
- **Goals** for the visualization:
  - Focus on information flow
  - Make dependencies explicit
  - Address the problems of Hasse diagrams
  - Utilize the computer medium (color, animation, etc)

Applications

- General information flow problems
  - Rumor spreading example
- Software visualization
  - Learning, designing, or debugging distributed programs and algorithms
    - **Examples**: Finding deadlocks, detecting synchronization errors, determining the critical path abstraction, longest sequential thread, chain of dependencies, etc.

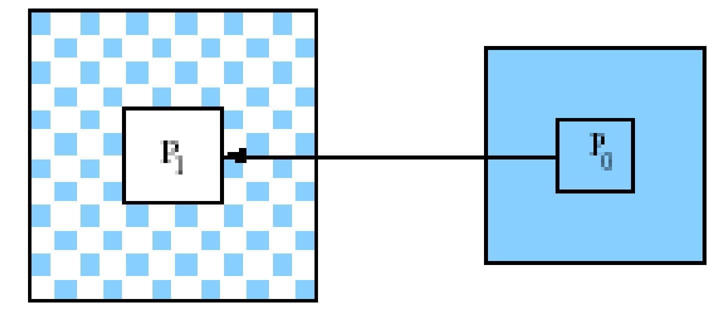
Growing Squares



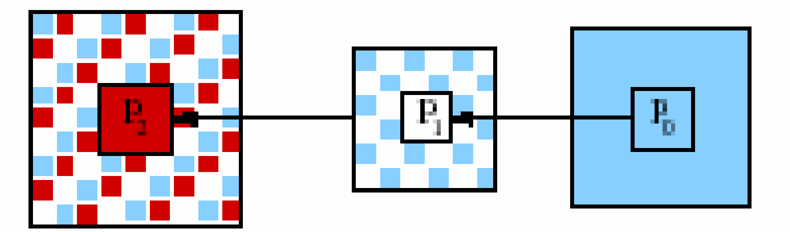
- The **Growing Squares** visualization technique was designed to solve this problem
- **Metaphor**: pools of color spreading on a piece of paper
  - Each pool (square) is a process/node and has a unique color
  - Squares grow in size over time
  - Messages from one square to another will add color to the destination square
  - Dependencies to a single square is easy to see just by studying its color makeup

Growing Squares: Visualization

- Message passing
  - Process P_0 (blue) sends a message to process P_1 (white)
  - Process P_1 will now have both blue and white in it

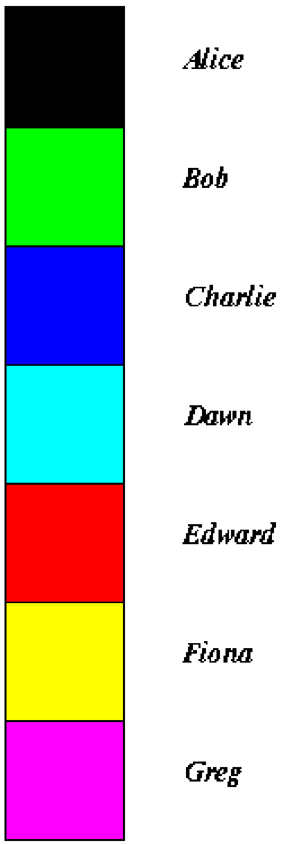


- Transitivity
  - Process P_0 (blue) sends a message to process P_1 (white)
  - Then, process P1 sends a message to P_2 (red)
  - Transitivity is clearly visible in P_2

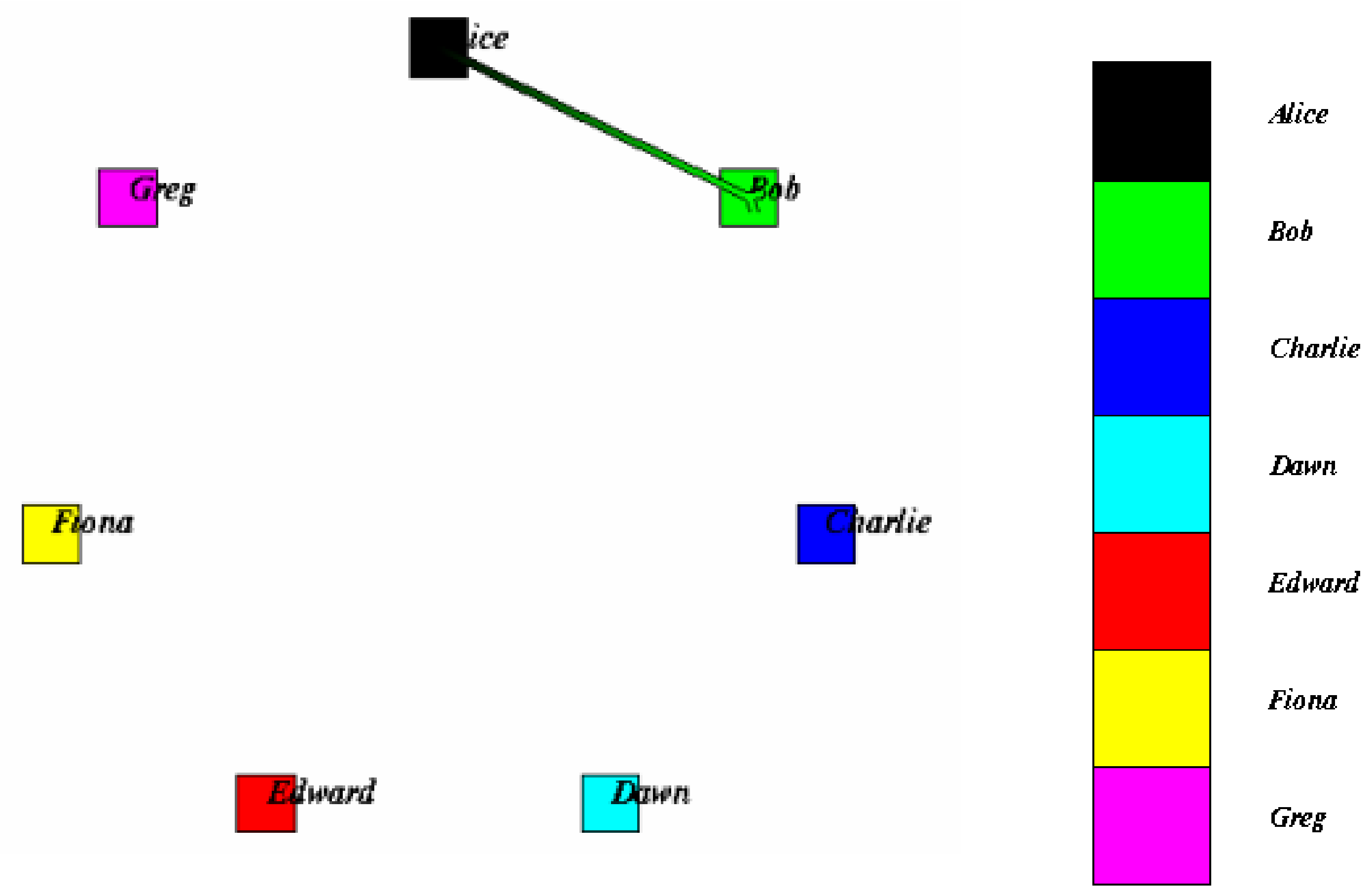


Growing Squares: Example

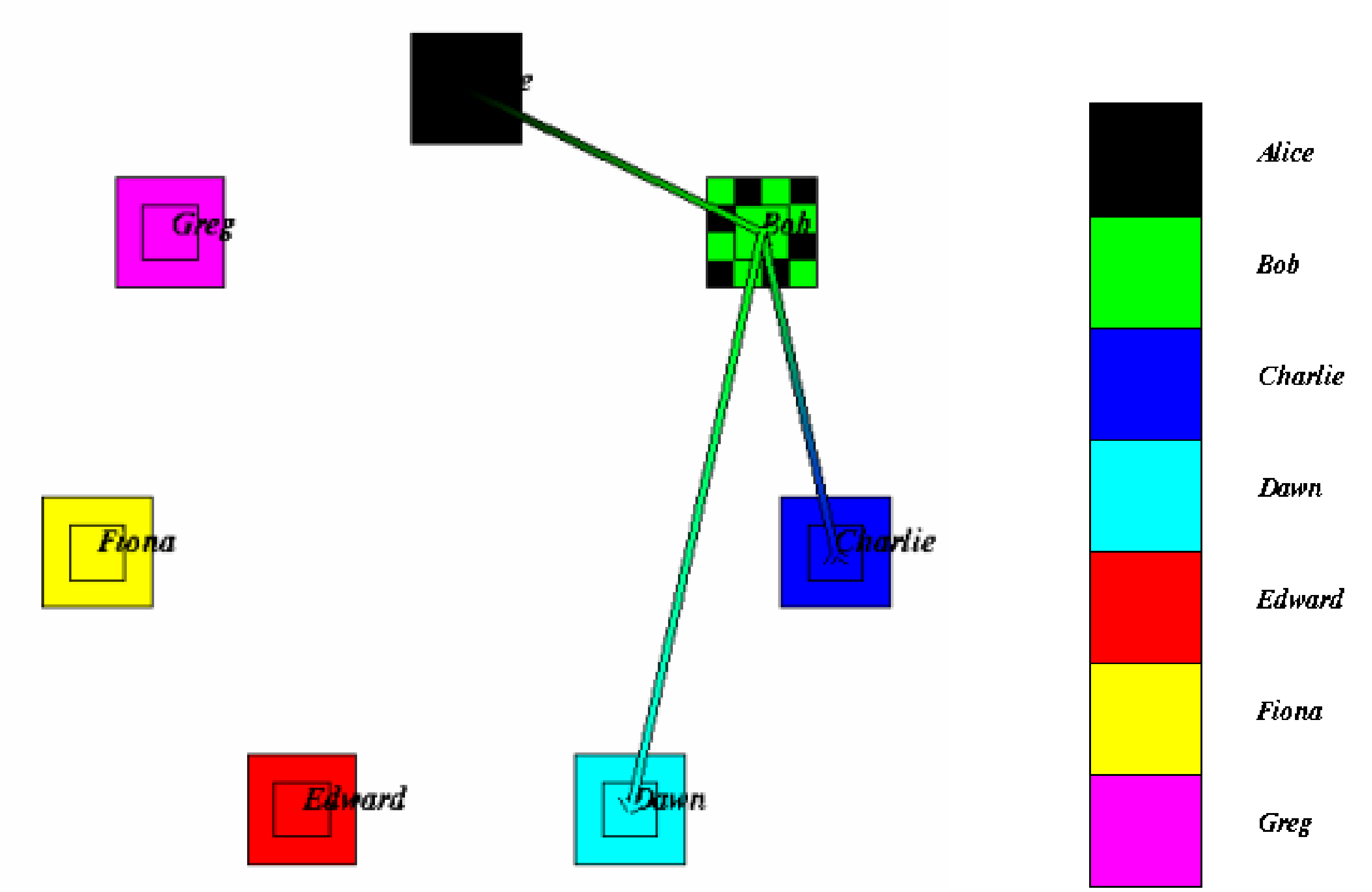
- We turn to our rumor spreading problem again
- The Growing Squares technique assigns each person (process) a **unique** color
  - Colors are uniformly distributed across the RGB spectrum
- We will now study the timesteps of the algorithm



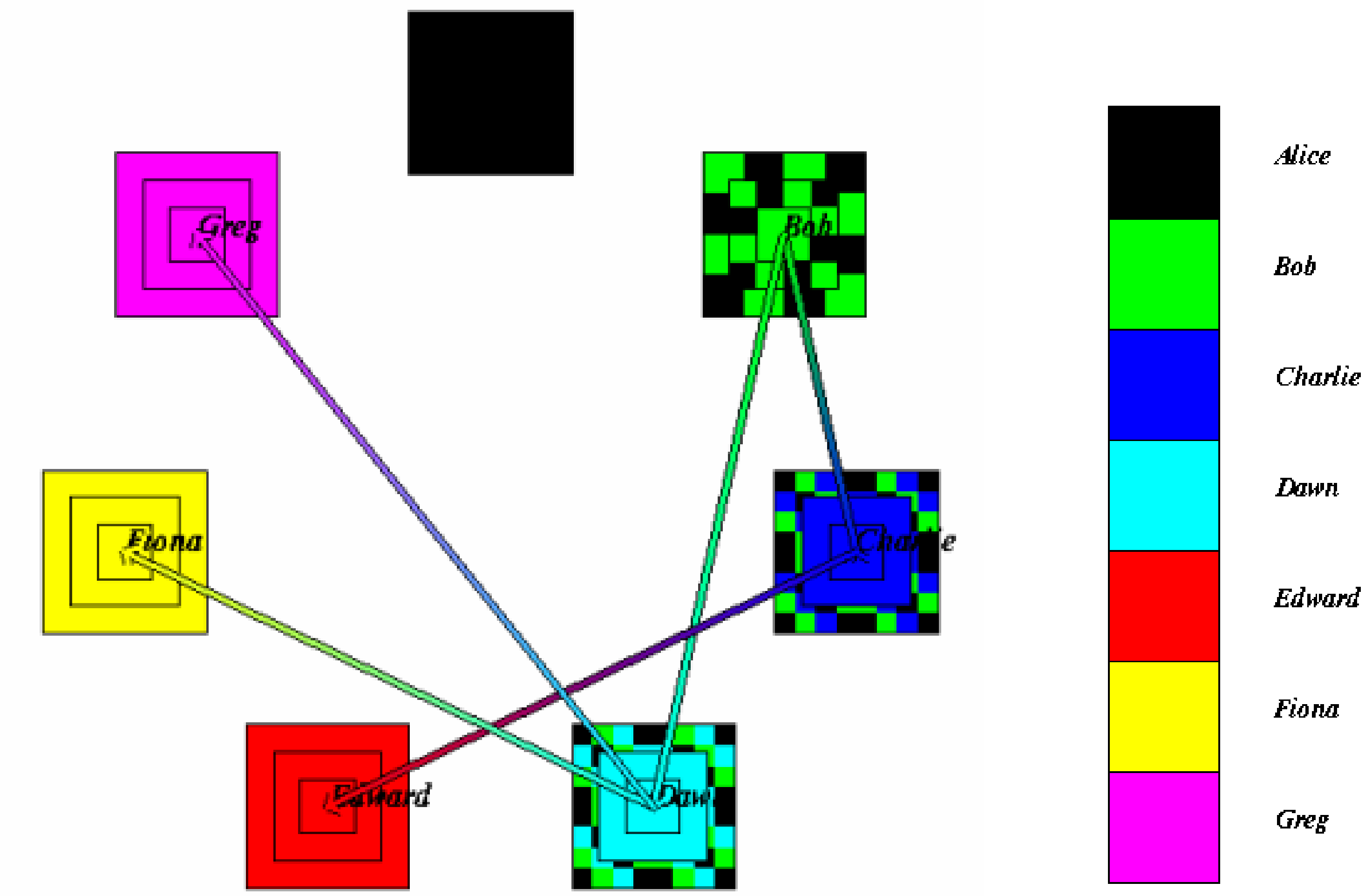
Growing Squares: Example (1)



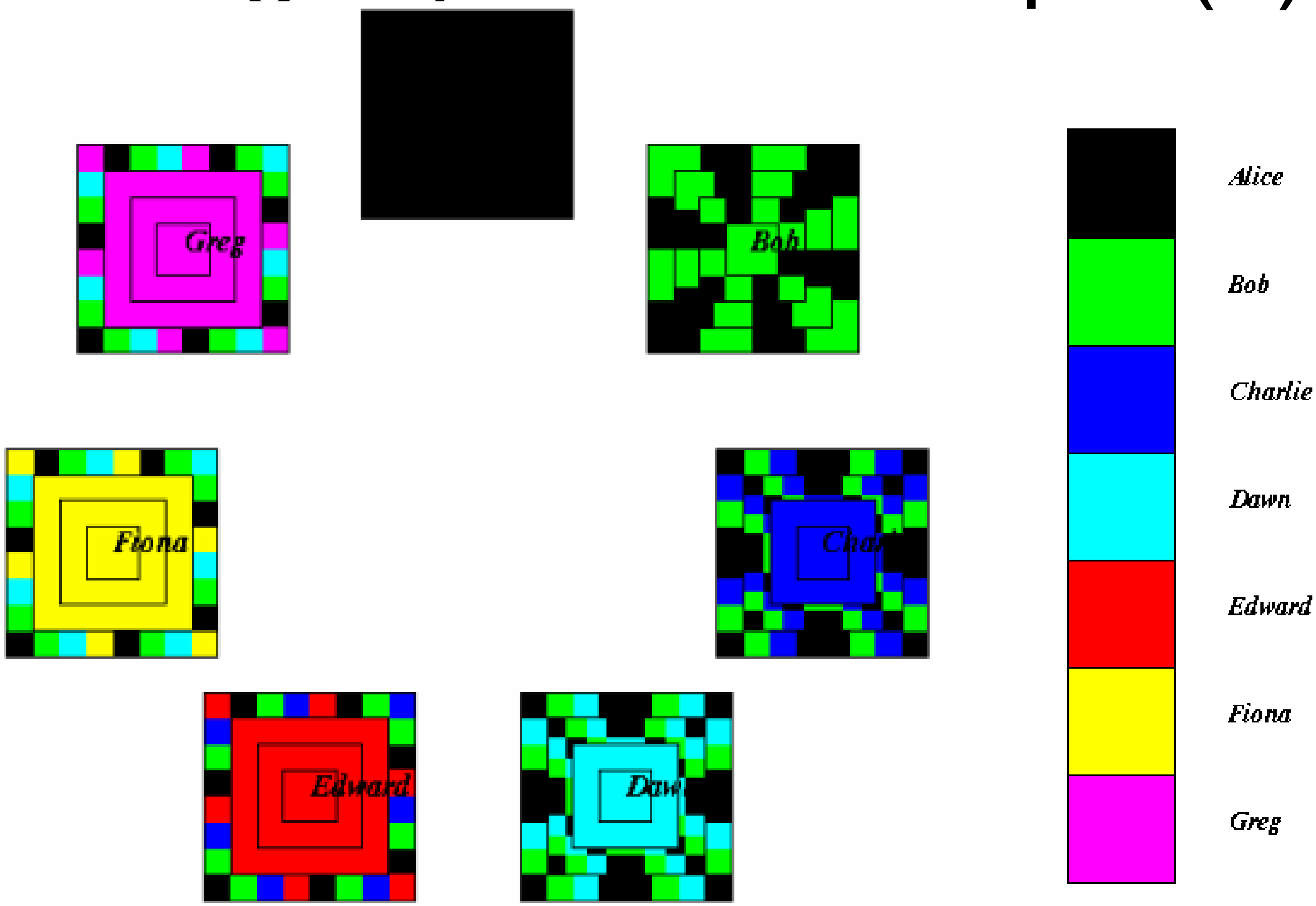
Growing Squares: Example (2)



Growing Squares: Example (3)



Growing Squares: Example (4)



Growing Squares: Benefits

- Self-contained information
  - Not scattered, like in Hasse diagrams
  - No need for backtracing of messages
- Intuitive influence color mapping
  - Processes ”contaminate” other processes
  - Sometimes difficult to distinguish colors
- Use of animation
  - Can help to give an intuitive understanding
  - Can make it more difficult to get a quick overview
  - More suited to a computer than print

Growing Squares: Design and Implementation

- Color scales
  - Investigated perceptually uniform color scales (LOCS)
  - Uniform RGB scale was most practical
- Zoomable interface
  - To allow for seeing details despite limited screen estate
- Continuous animation
  - Present the dynamic execution of the system
- Implementation
  - C++ and OpenGL in Linux using GTK--

User Study

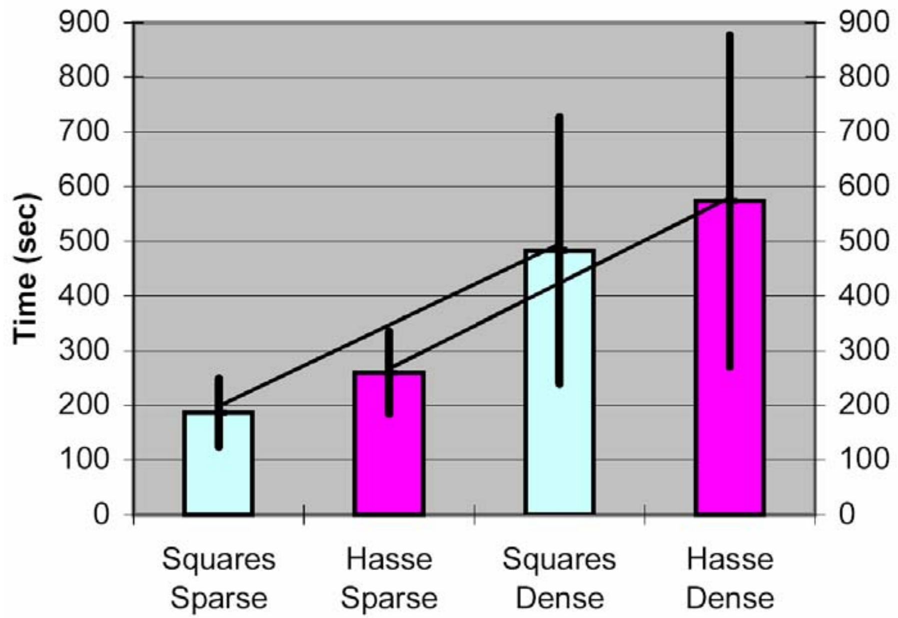
- A formal user study comparing Hasse diagrams to Growing Squares was performed
  - Two-way repeated-measures ANOVA
  - Independent variables (both within-subjects):
    - Visualization type: Hasse or GS
    - Data density: sparse and dense
- 4 different data sets: 1 of each data density for each visualization type
- 12 subjects participated in the test
  - All subjects knowledgeable in distributed systems

User Study: Tasks

- Each data set required the user to solve 4 common questions related to causal relations:
  1. Find the process with longest duration
  2. Find the process that has had the most influence on the system
  3. Find the process that has been influenced the most
  4. Is process *x* causally related to process *y*?
- Times were measured for these tasks
- Users were also asked for their subjective opinion of the visualization

Results

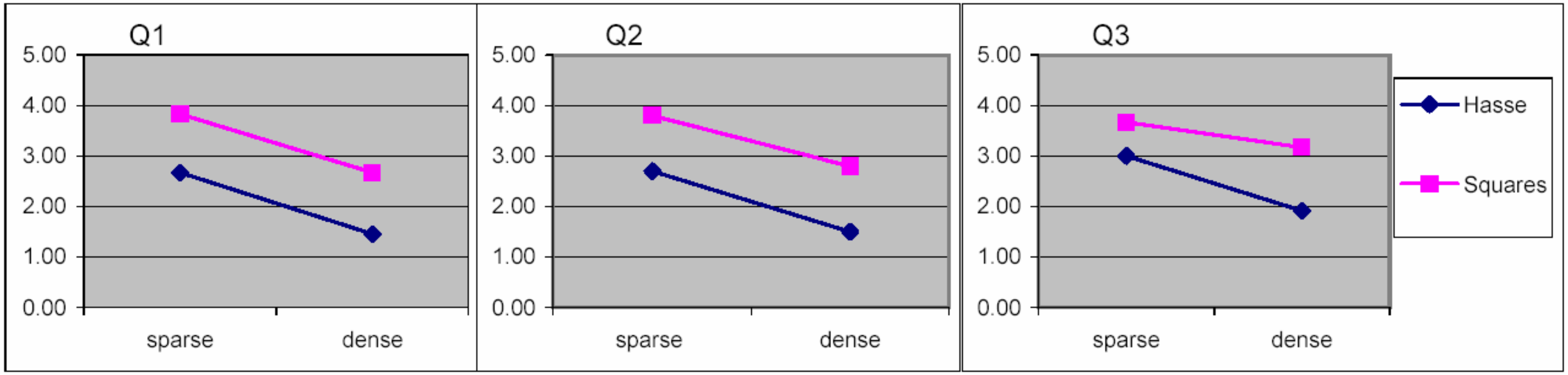
- Performance measurement
  - Users were more efficient using our technique than Hasse diagrams
  - GS seems to scale with size similarly to Hasse diagrams
  - GS is significantly faster for the sparse density
  - **No** statistically significant difference for GS vs Hasse in the general case



Mean task completion times (standard deviations)

Results (2)

- Subjective ratings
  - Very positive user feedback
  - Users consistently rated GS over Hasse diagrams in all respects (ease-of-use, enjoyability, efficiency)
  - These readings were **all** statistically significant



Ease-of-use *Efficiency* *Enjoyability*

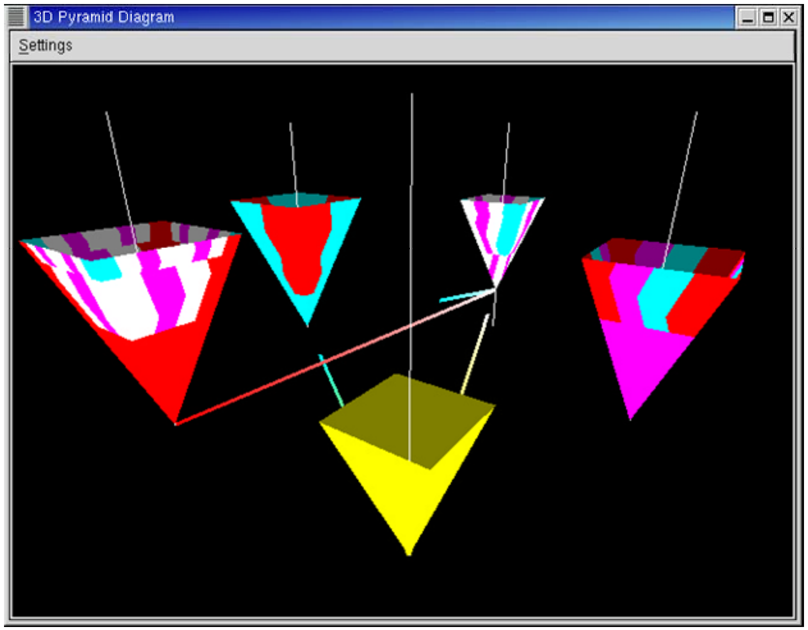
Discussion

- GS is faster, but not significantly so
  - *Few test subjects* – wildly varying completion times
  - *Colors hard to distinguish* – explore alternate ways
  - *No prior training using GS* – extended testing
- Our method appeals more to users
- GS and Hasse diagrams provide different views of system
  - A combination of the two visualizations is ideal

Conclusions

- Visualization of causal relations is crucial for understanding complex distributed and parallel systems
- Traditional visualization techniques (**Hasse diagrams**) fall short
- **Growing Squares** is a novel idea of visualizing distributed systems focused on the information flow
- Our visualization technique:
  - more efficient to use than Hasse diagrams, though not significantly so
  - significantly more appealing to users than Hasse diagrams

Future Work



- Extensions to the original technique (i.e. Growing Pyramids in 3D)
- Related visualization techniques (Growing Polygons to be presented at InfoVis 2003 in Seattle)
- More complex adaptions that make use of the 3rd dimension

Questions?

- Contact information:
  - Address:

    Niklas Elmqvist and **Philippas Tsigas**
    Department of Computing Science
    Chalmers University of Technology
    SE-412 96 Göteborg, Sweden
  - Email:
    - {elm|tsigas}@cs.chalmers.se
  - Project website:
    - http://www.cs.chalmers.se/~elm/projects/causalviz
  - Demo this afternoon