Visualisierung verteilter Systeme

Visualisierung von kausalen Abhängigkeiten

von Susanne Schulz

Gegenstand dieses Vortags

- Growing Squares: Animated Visualization of Causal Relations
- Causality Visualization Using Animated Growing Polygons

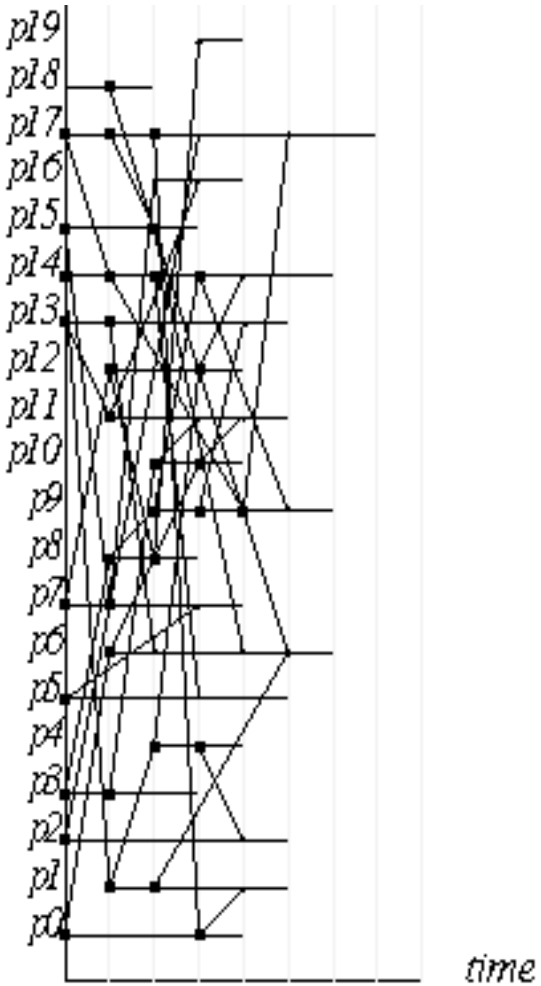
- Verfasst von

Niklas Elmqvist und Philippas Tsigas
Department of Computing Science
Chalmers University of Technology, Schweden

Gegenstand dieses Vortags

- neue Technik zur grafischen Darstellung von kausalen Abhängigkeiten
- als Computer-Animation

Weg-Zeit Diagramme



- Traditionell Darstellung durch Weg-Zeit-Diagramme
  - Je nach Anzahl der Nachrichten und der beteiligten Prozesse, kann die Visualisierung unübersichtlich und die Analyse entsprechend Aufwendig sein.

Weg-Zeit Diagramme (2)

- Probleme
  - Nachrichten müssen eventuell zurückverfolgt werden
  - Wichtige Informationen sind verstreut, und nicht schnell erfassbar

Ziele

- Zielsetzung war
  - den Informationsfluss darzustellen
  - Abhängigkeiten zwischen Prozessen sichtbar zu machen

Benutzer Studie

Aufgaben die an die Benutzer gestellt wurden:

- Finde den Prozess, der am längsten aktiv war.
- Finde den Prozess, der den meisten Einfluss auf das System hat – der die meisten Nachrichten verschickt.
- Finde den Prozess der am stärksten von anderen Prozessen beeinflusst wird.
- Ist Prozess *x* kausal abhängig von Prozess *y*?

Idee

- Die Prozesse seinen Farbtöpfe.
- Nachrichten die im System umhergeschickt werden, tragen die Farben des Absenders.
- Die empfangende Farbe beeinflusst die Farbe des Empfängers.

Kausale Abhängigkeiten

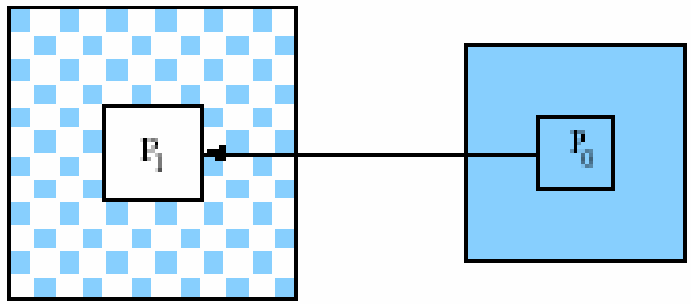
- Es gibt eine Menge von Prozessen $P_1..P_N$, diese können als disjunkte Mengen von Ereignissen des gesamten Systems betrachtet werden.
- Es gibt 2 Arten von Ereignissen:
  - Interne Ereignisse, die auf einem Prozess sequenziell ablaufen und
  - Externe Ereignisse die Prozesse durch Nachrichten verbinden.

Kausale Abhängigkeiten

- Zwei Ereignisse sind kausal abhängig:
  - Falls $x_k, y_l \in P_i$ und $k<l$ dann $x_k \rightarrow y_l$
  - Falls $x \in P_i$, $y \in P_k$, $i \neq k$, $x = send(m)$ und $y=receive(m)$ dann $x \rightarrow y$

Wachsende Quadrate – Growing Squares

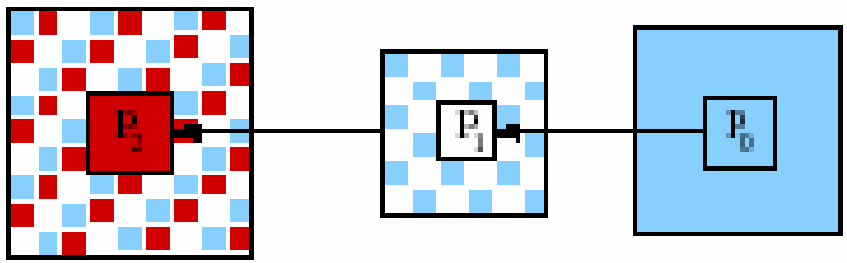
- 2 Prozesse P_0 und P_1
- P_0 sendet an P_1 eine Nachricht
- Zweiter „Ring“ stellt sich nach Empfang der Nachricht als Mix der Farben des Senders und das Empfängers dar



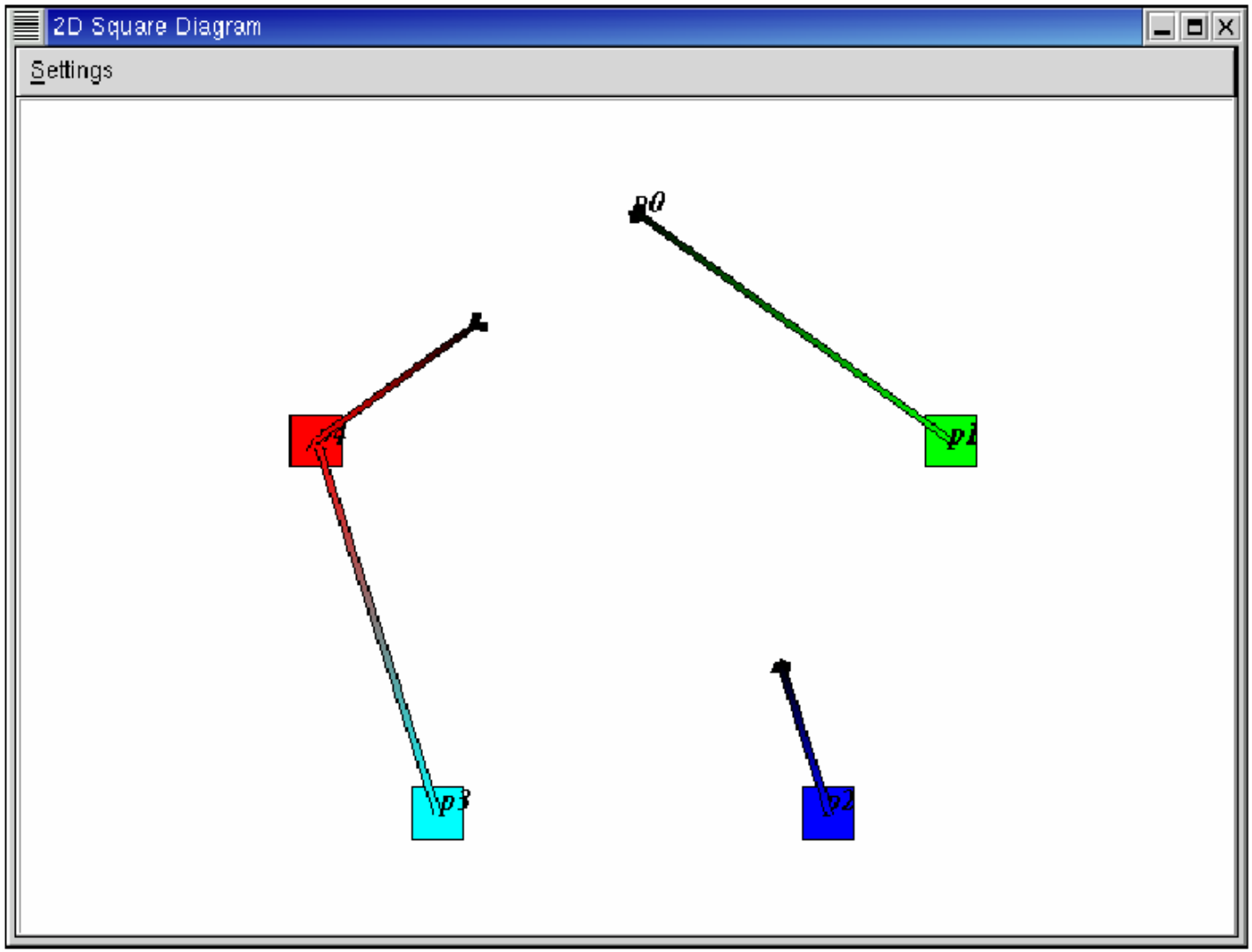
Transitivität

- Transitivität
  - a➔b und b➔c,
    dann a➔c

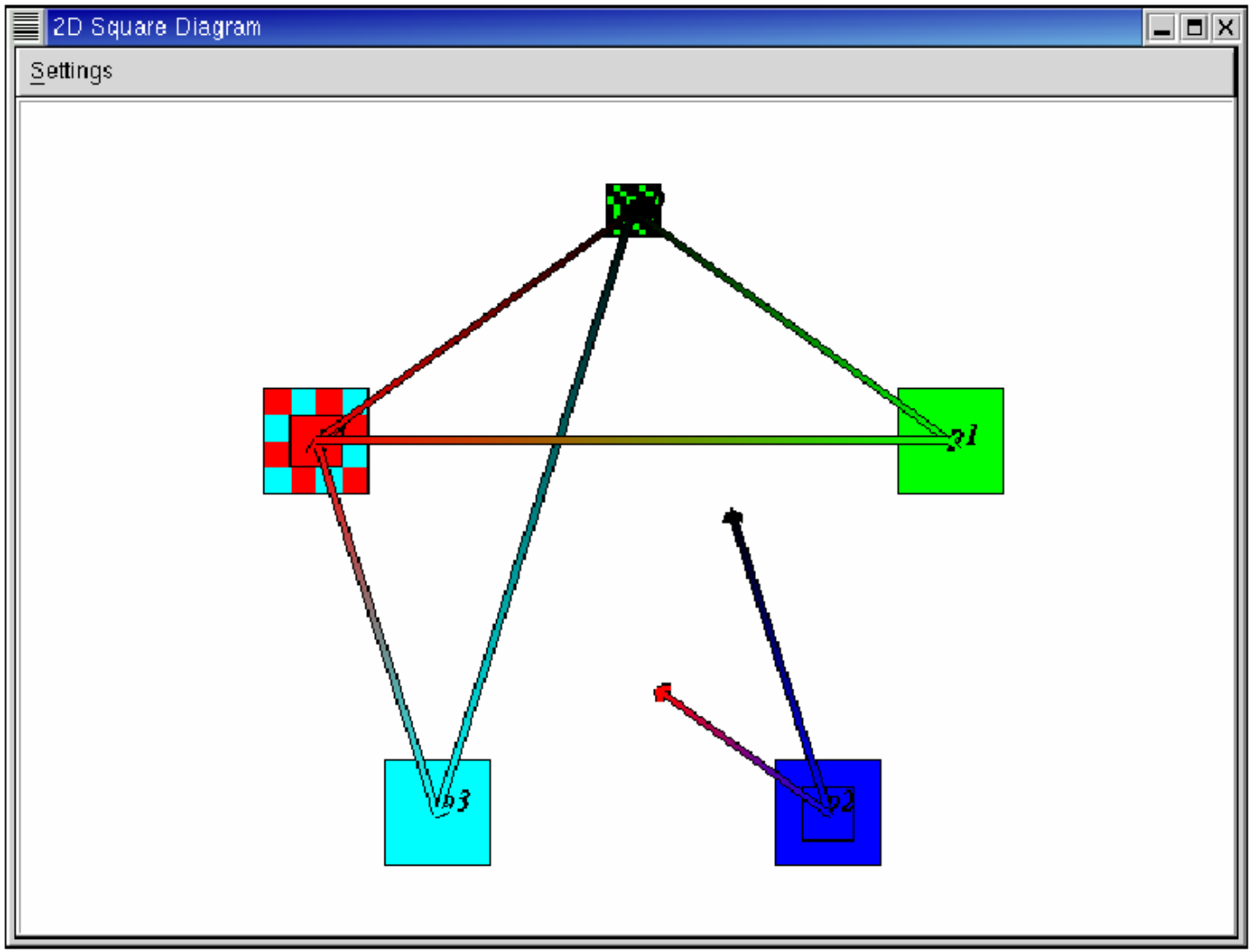
- Der Zustand eines Prozesses wird mitgeschickt
  - P_0 ➔ P_1 und P_1 ➔ P_2
    P_0 ➔ P_2



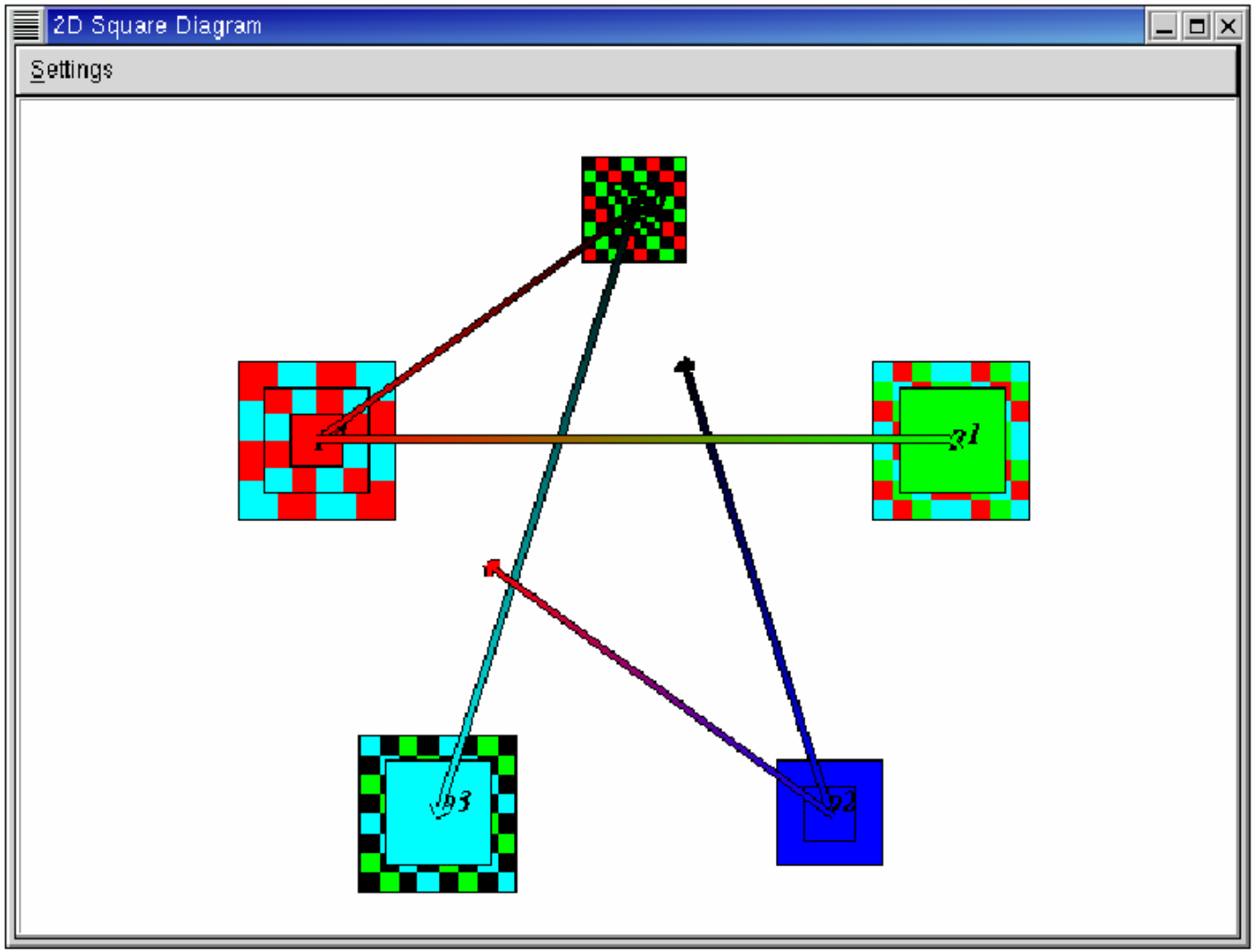
Nachrichten Austausch 1



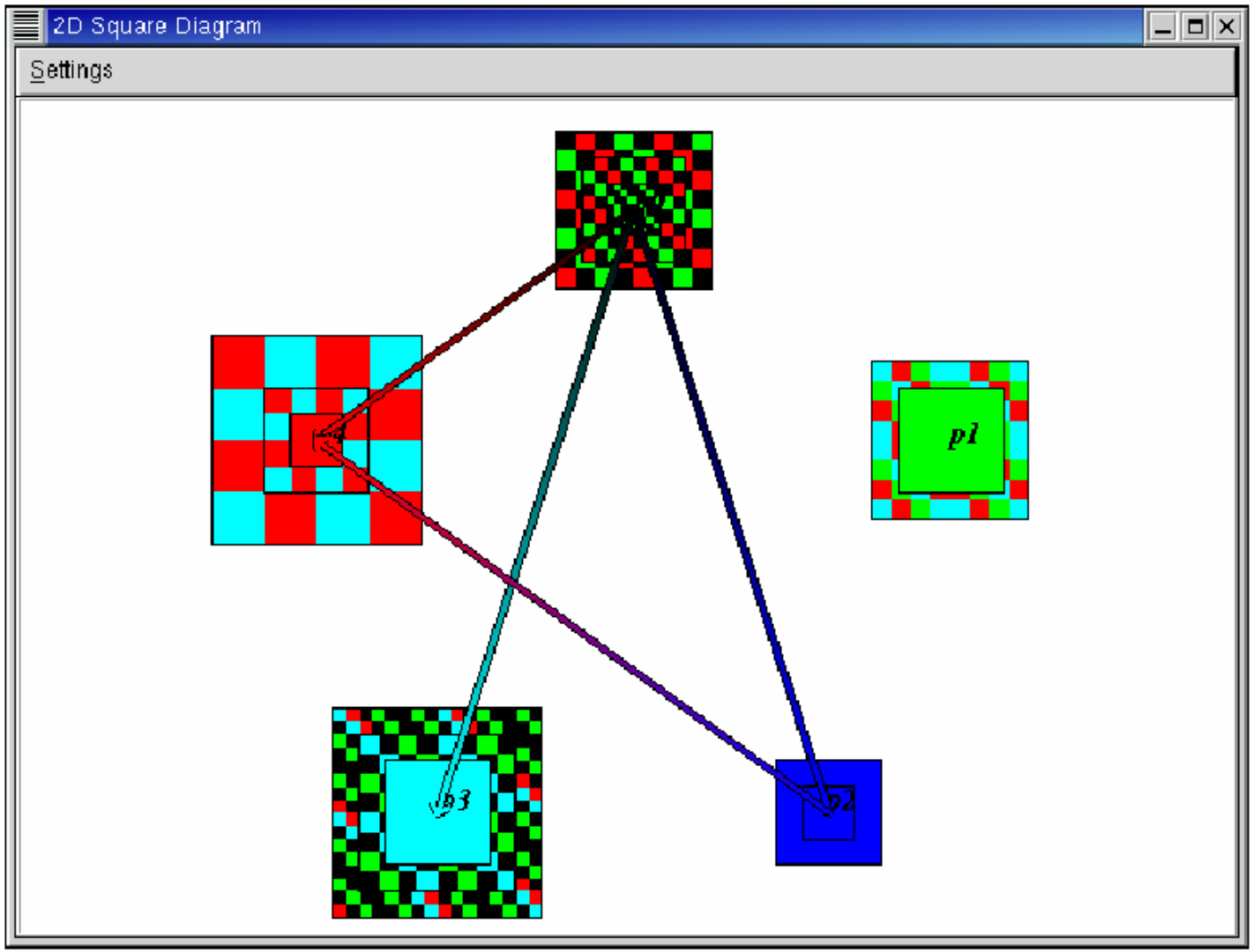
Nachrichten Austausch 2



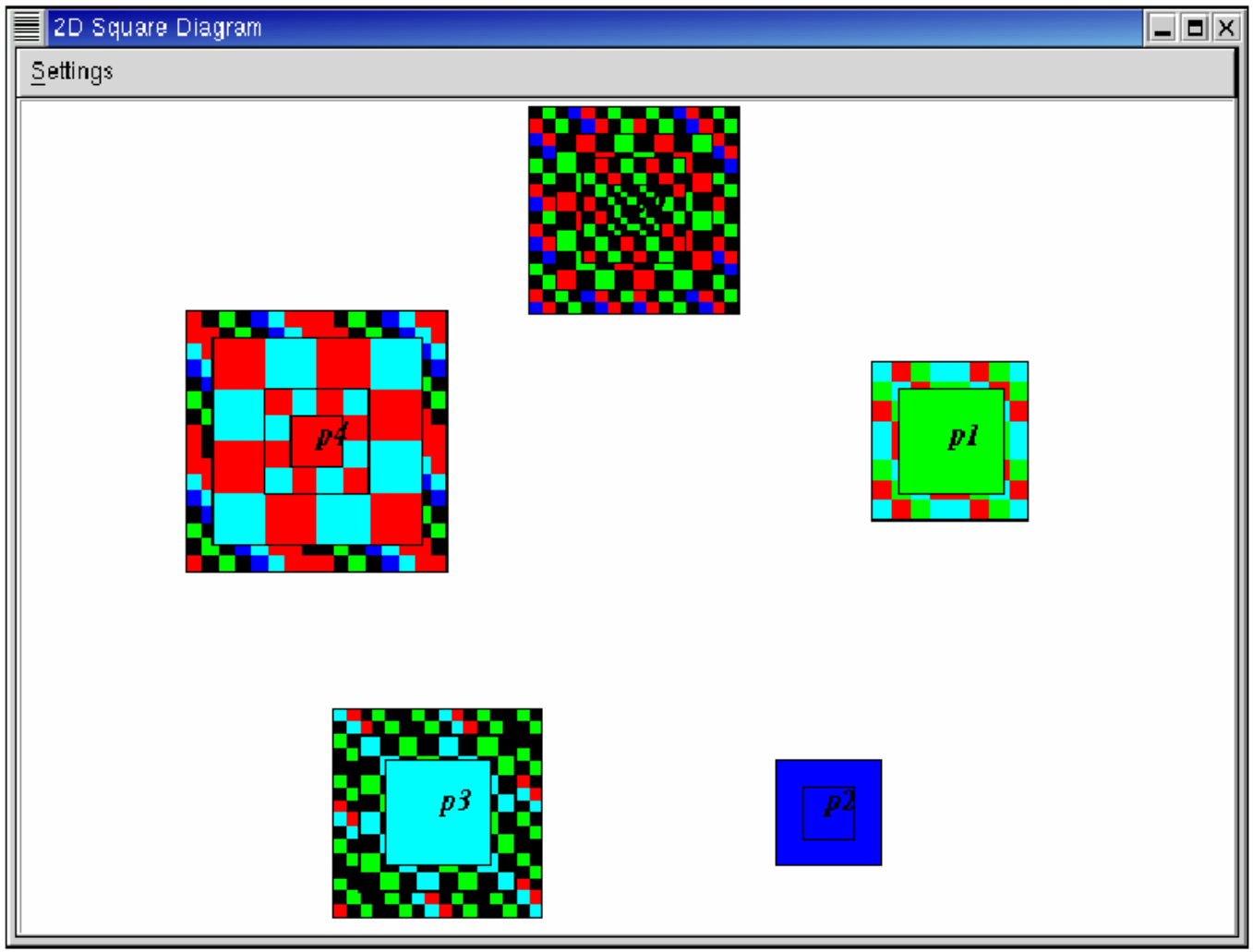
Nachrichten Austausch 3



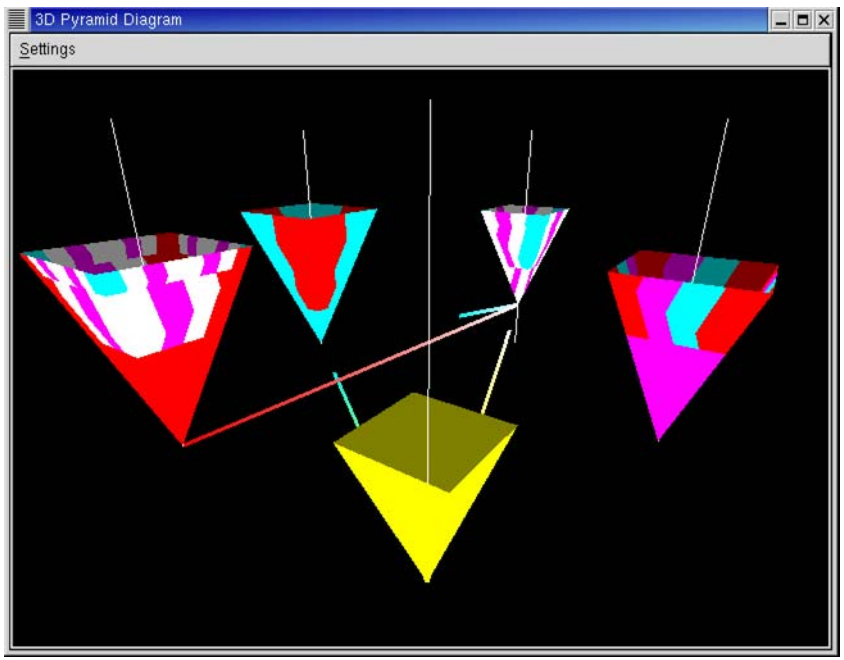
Nachrichten Austausch 4



Nachrichten Austausch 5

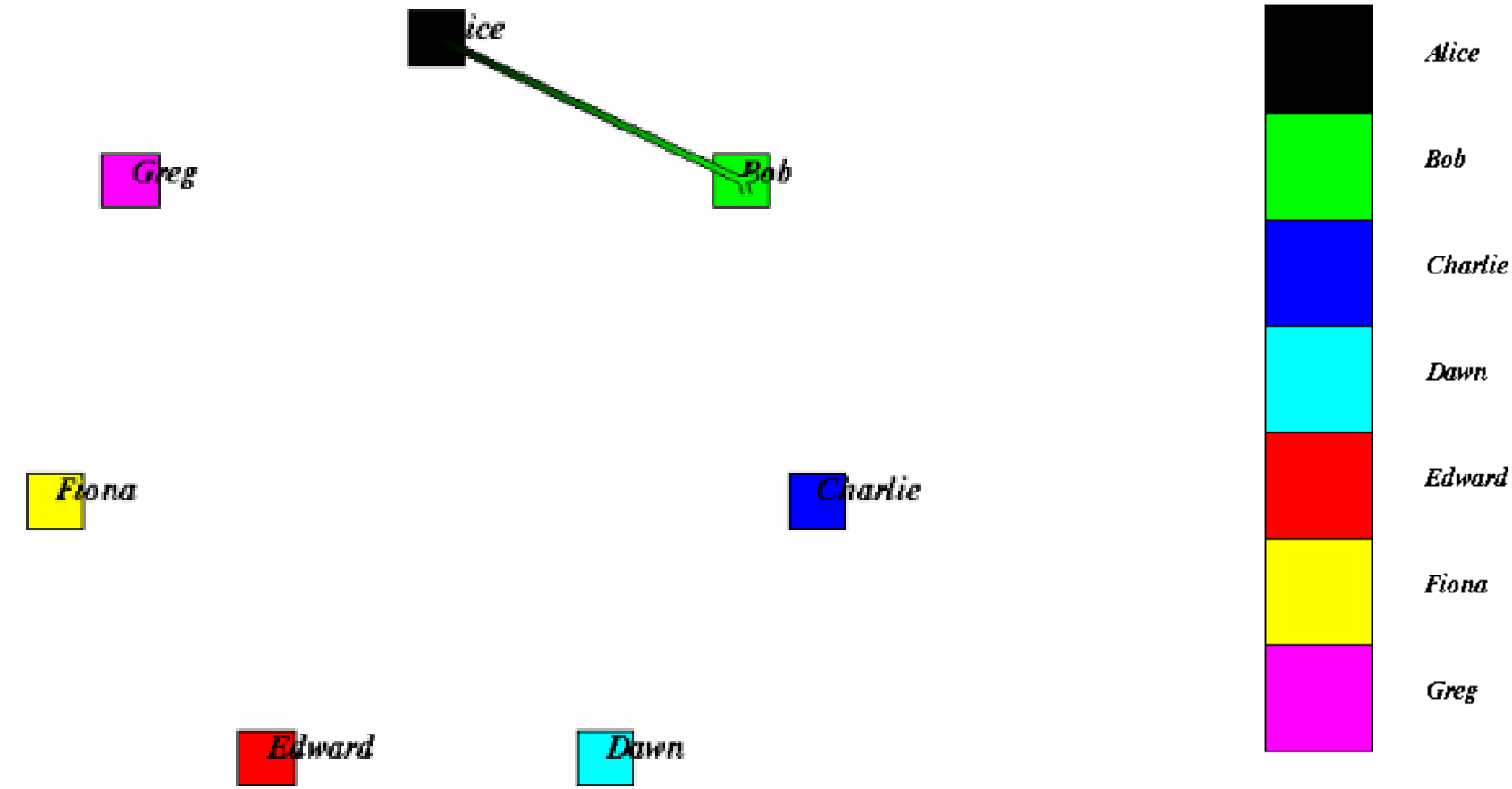


Sonstige Ideen – 3D Visualisierung

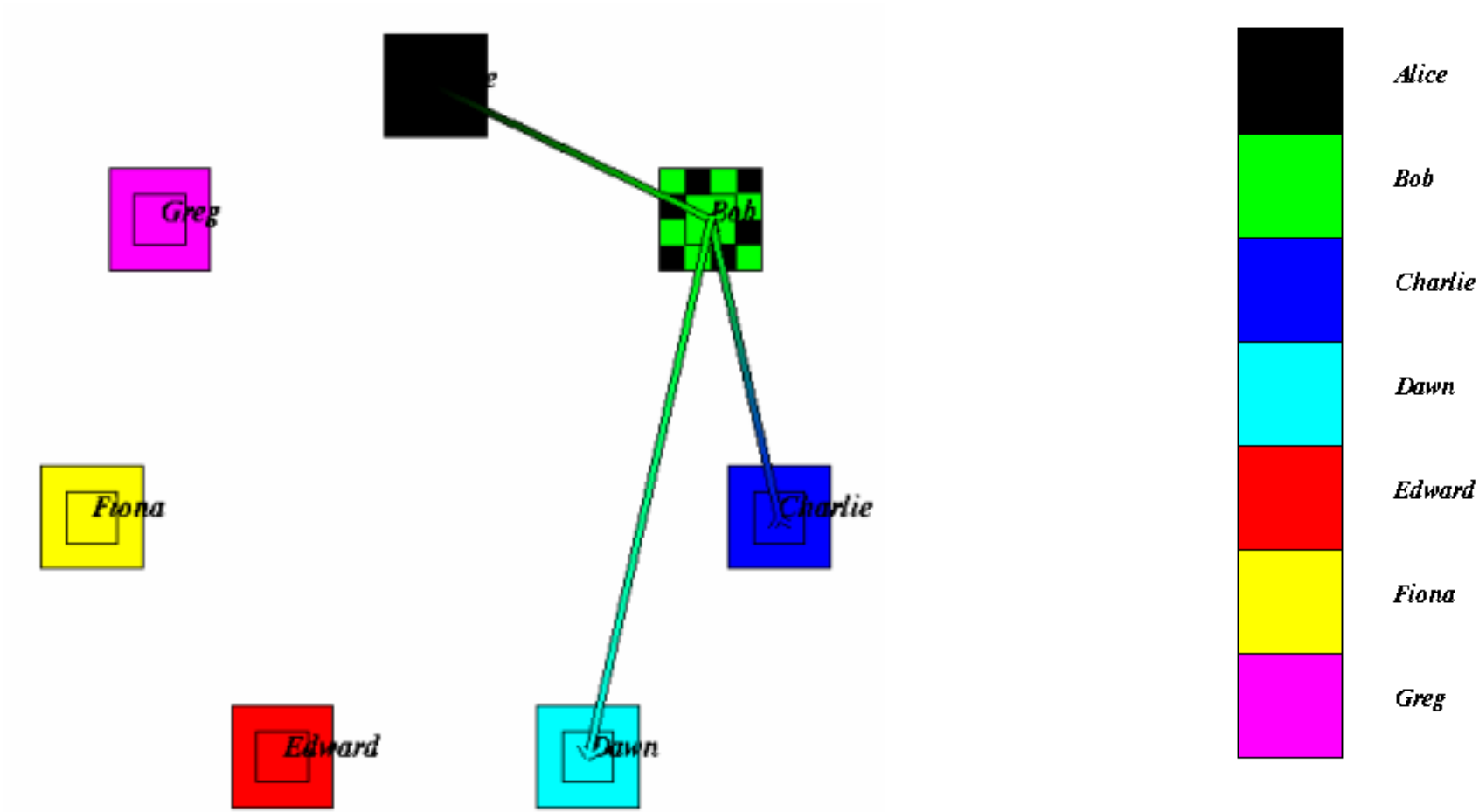


- Growing Pyramids
  - Erweiterung der „Growing Squares" um eine z –Achse, welche die Zeit wiederspiegelt

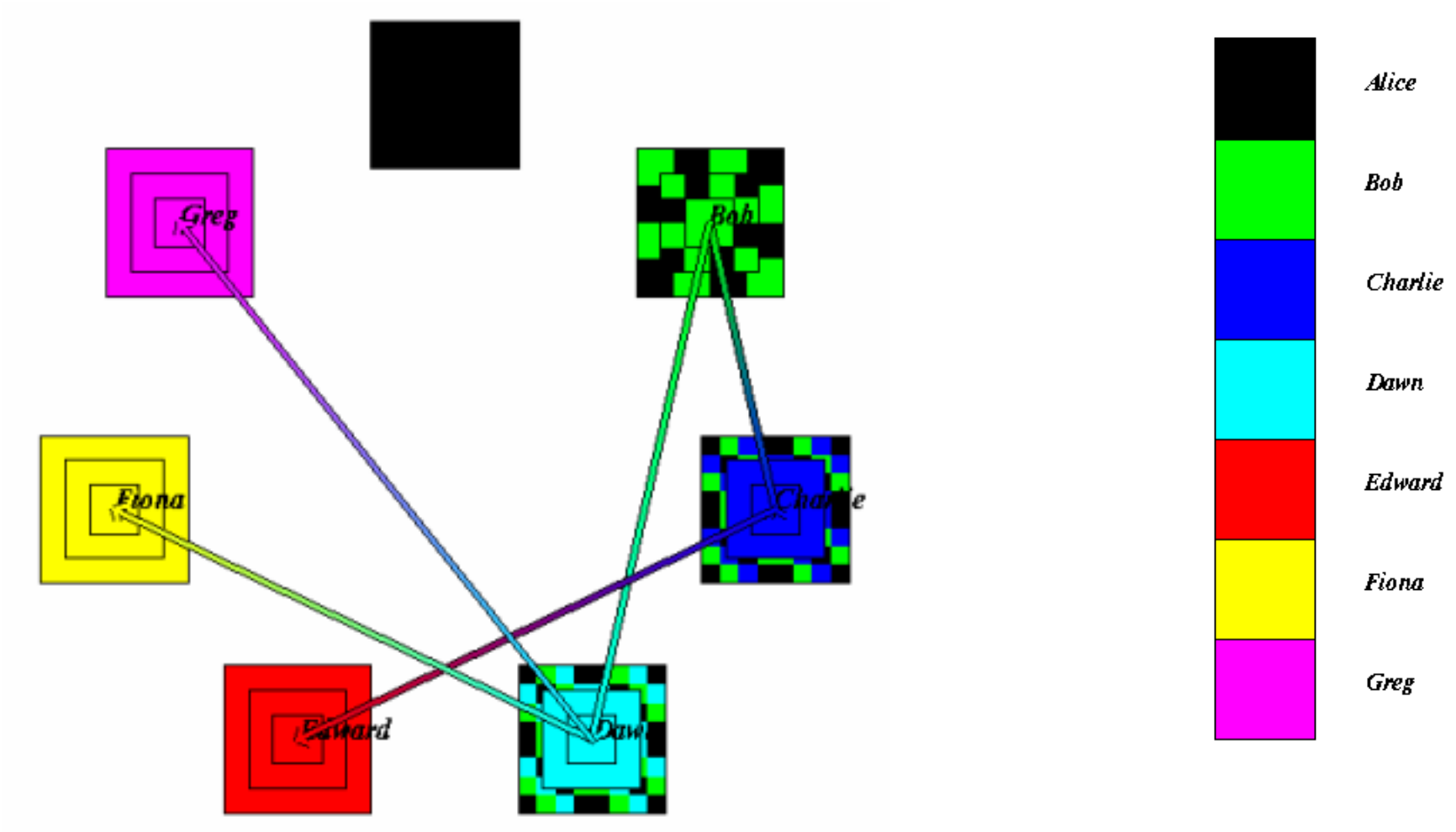
Nachrichten Austausch 1



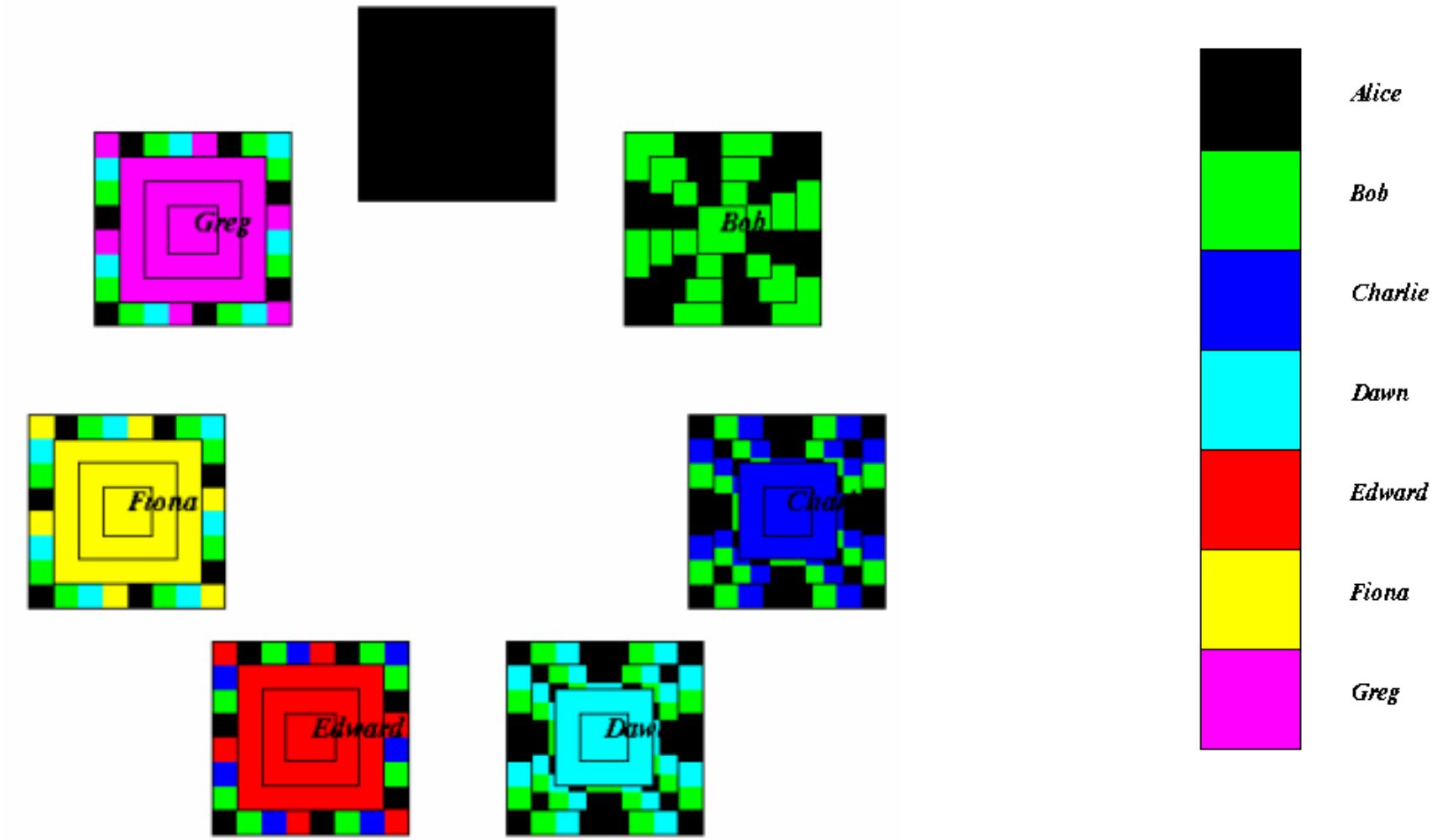
Nachrichten Austausch 2



Nachrichten Austausch 3



Nachrichten Austausch 4 (Endprodukt)

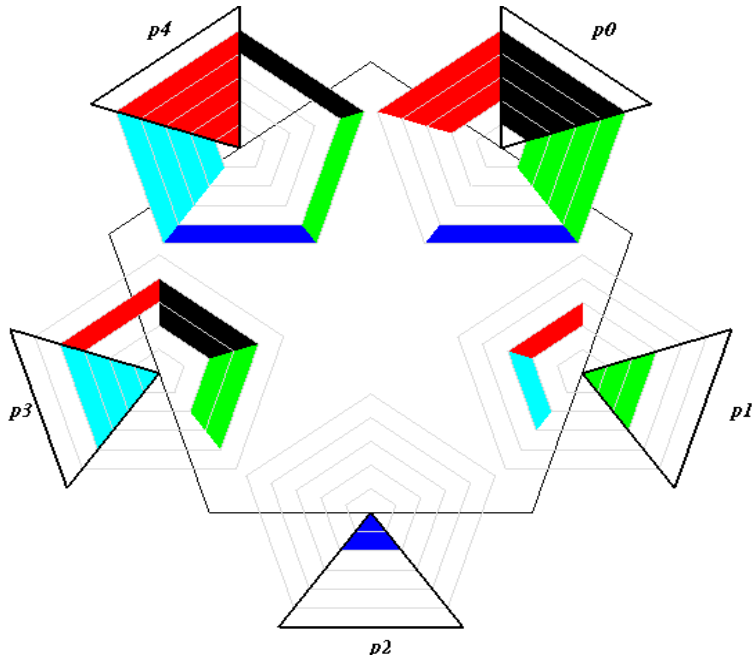


Probleme mit dieser Variante

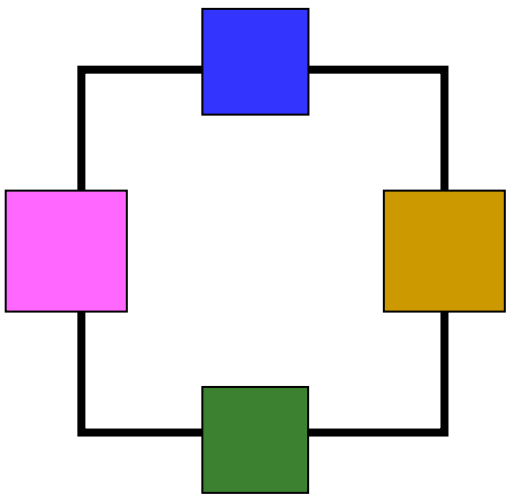
- Bei vielen Prozessen ist es schwierig Farben zu finden die sich genügend unterscheiden um sie auseinander halten zu können
  - Ähnliche Farben werden leicht verwechselt
  - Es gibt keine globale Zeit

Wachsende Polygone - Growing Polygons

- Idee:
  - Es gibt n Prozesse, die alle eine eigene Farbe besitzen
  - Jeder Prozess wird durch ein n-seitiges Polygon repräsentiert
  - Für jeden anderen Prozess gibt es einen dreieckigen Abschnitt auf jedem Polygon



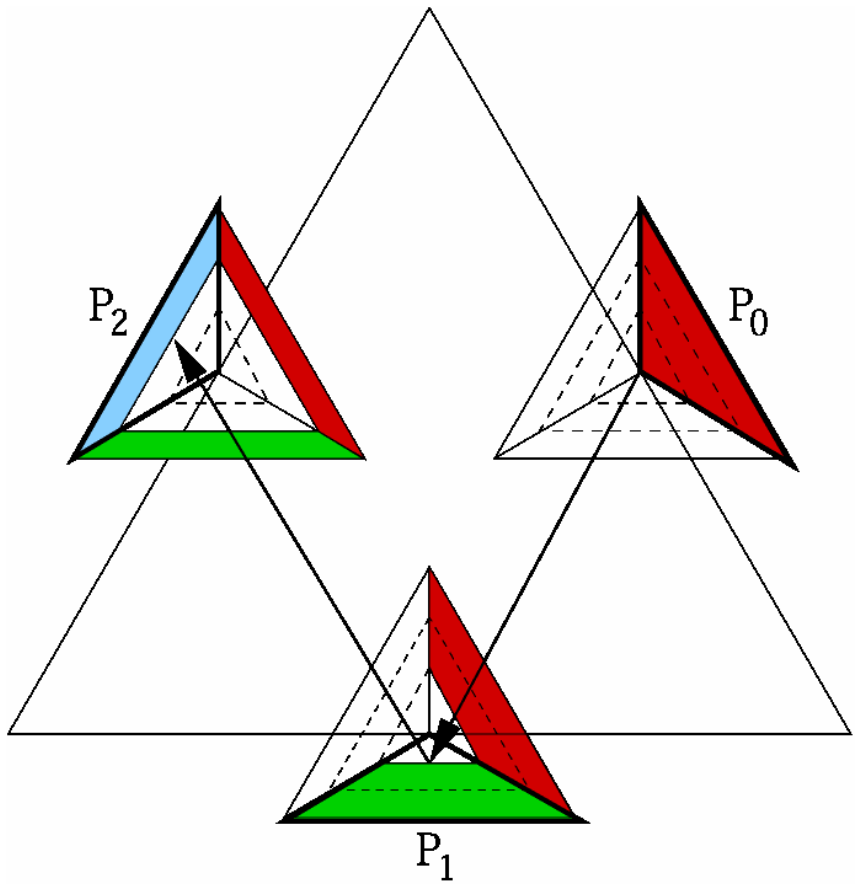
Wachsende Polygone - Growing Polygons



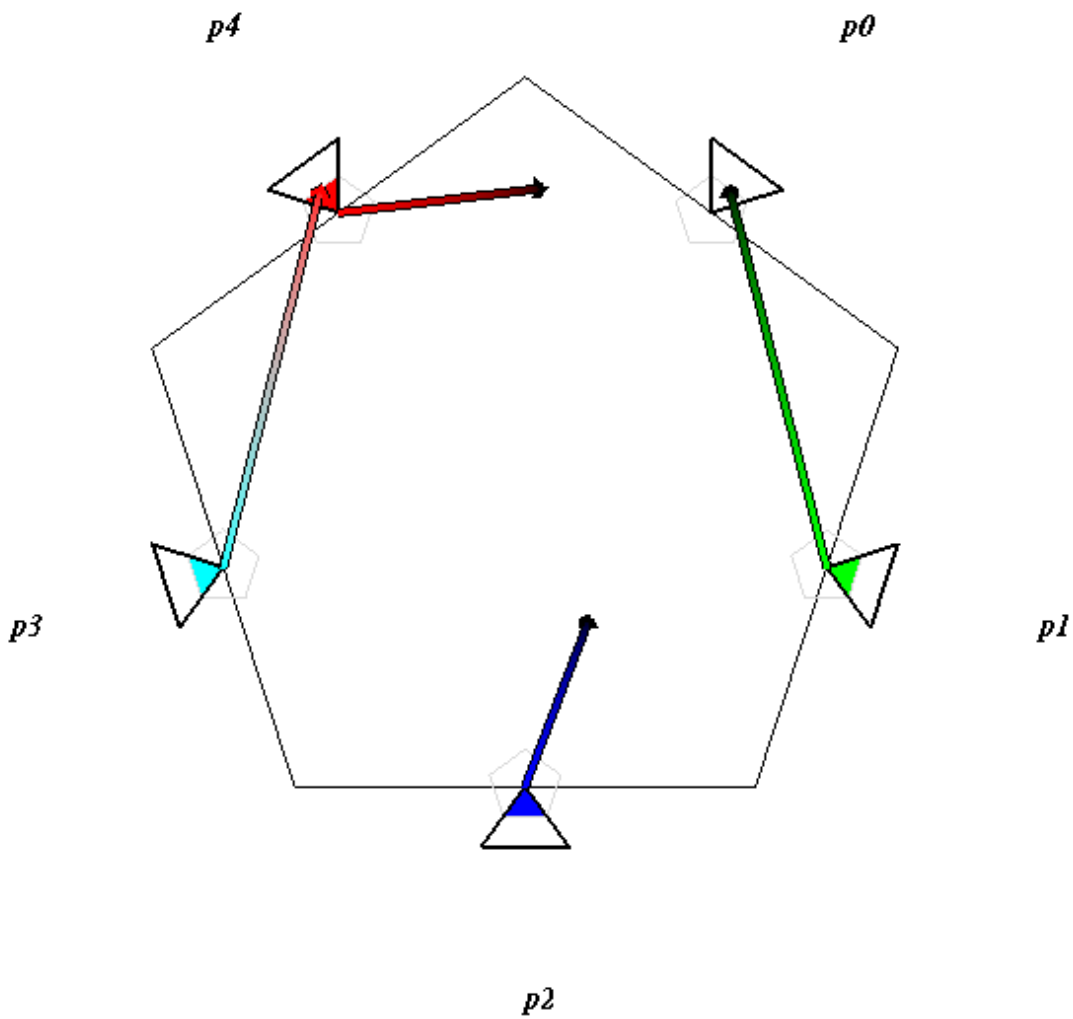
- Die Prozesse liegen auf einem n-seitigen Layout-Polygon
- Jedes Polygon wächst mit jedem Zeitzyklus
- Nachrichten werden als Pfeile dargestellt

Wachsende Polygone - Growing Polygons

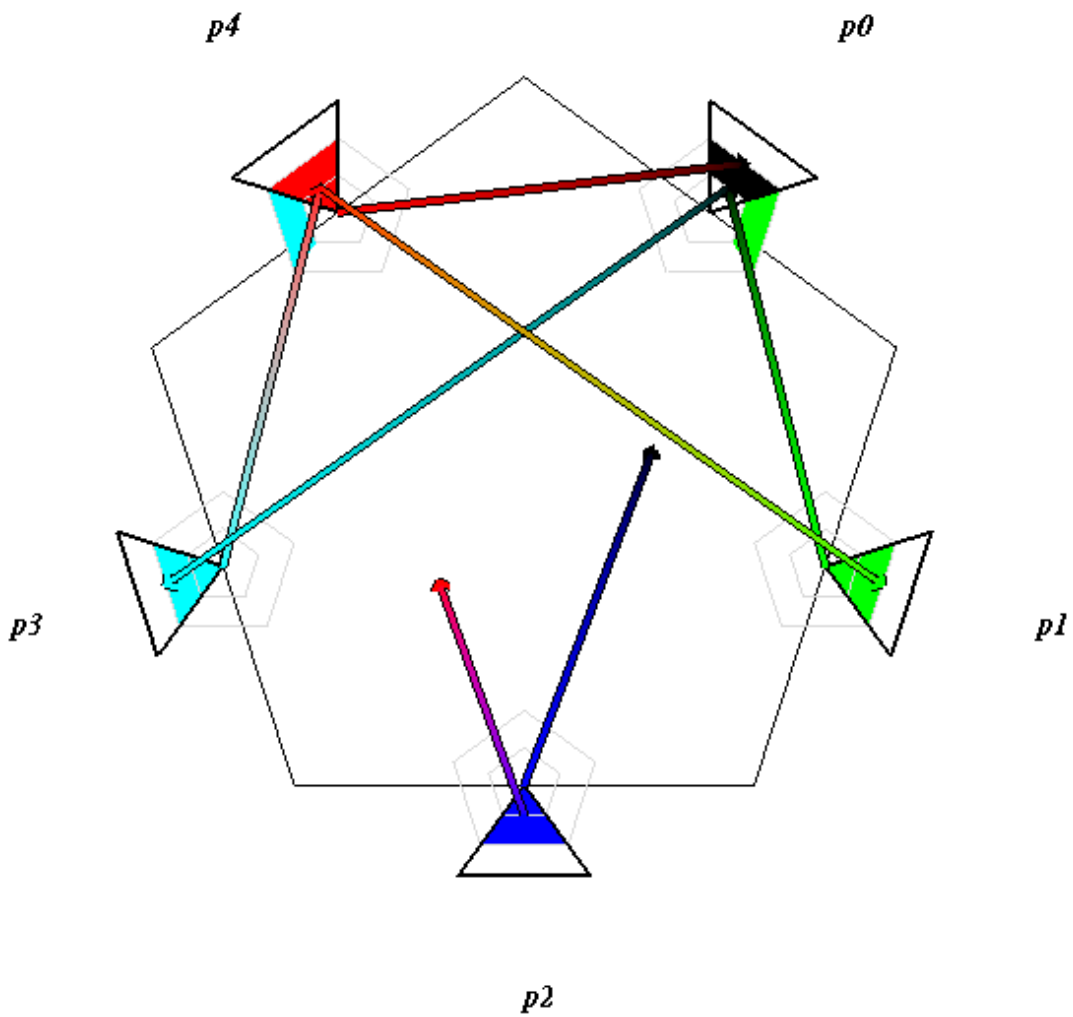
- Die Nachrichten tragen auch hier die kausalen Abhängigkeiten weiter
  - (Nachricht von P_1 nach P_2)
  - somit kann auch hier Transitivität dargestellt werden



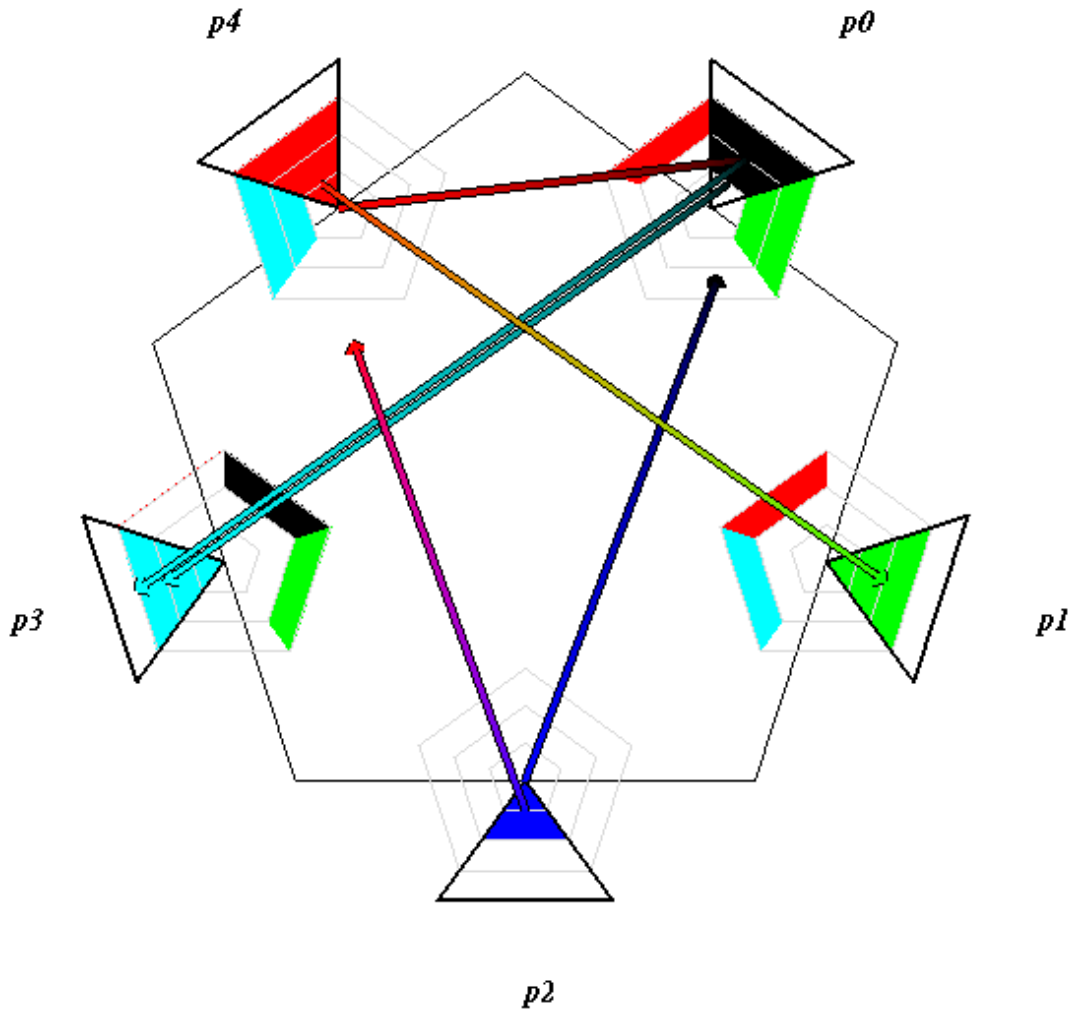
Nachrichten Austausch 1



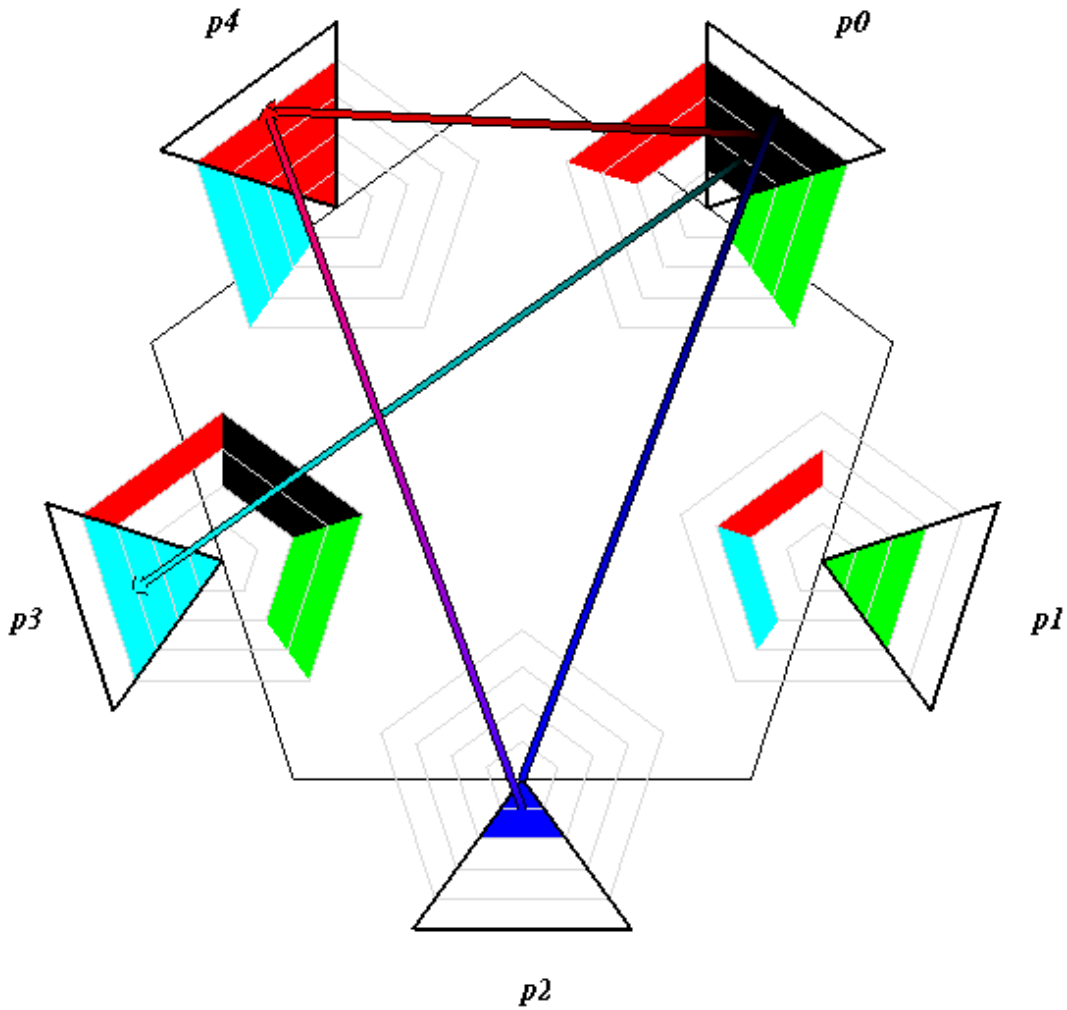
Nachrichten Austausch 2



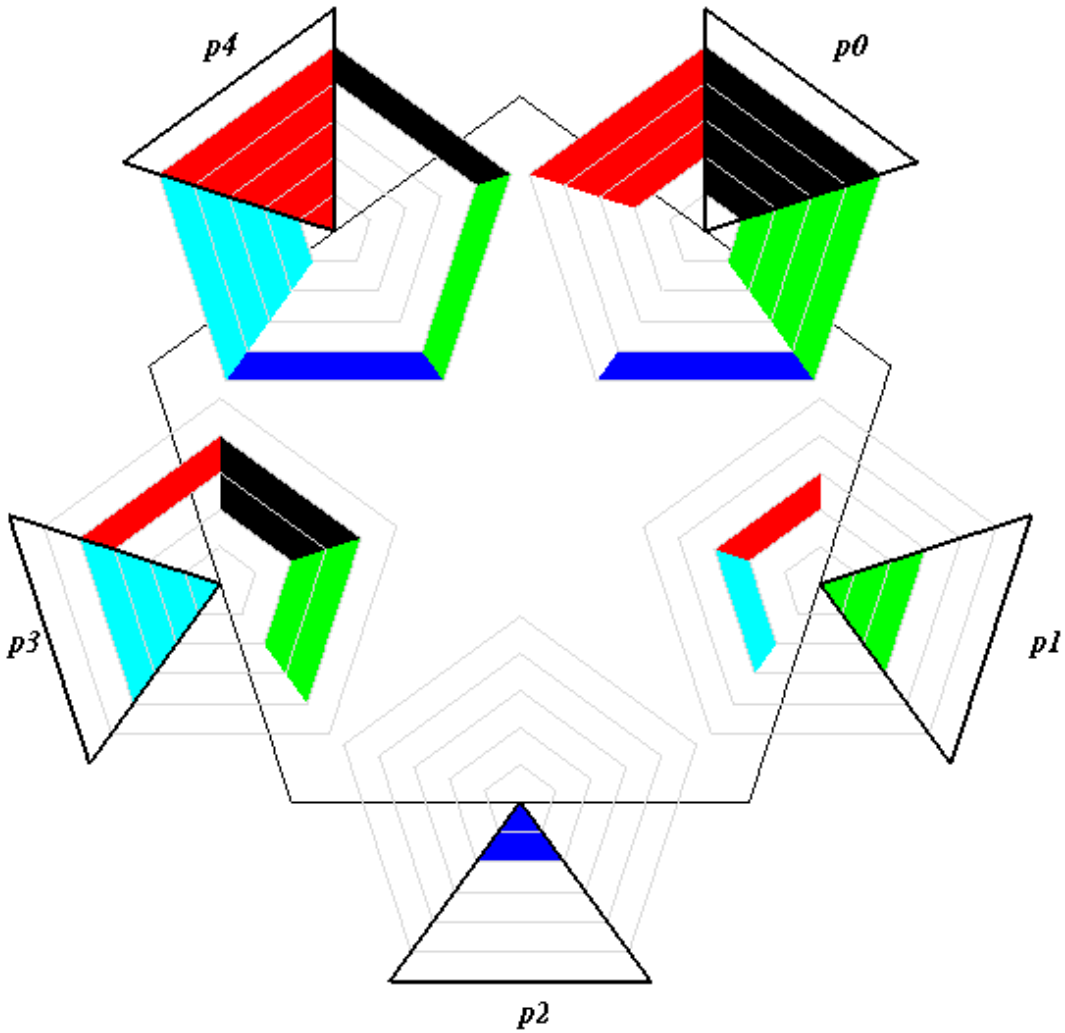
Nachrichten Austausch 3



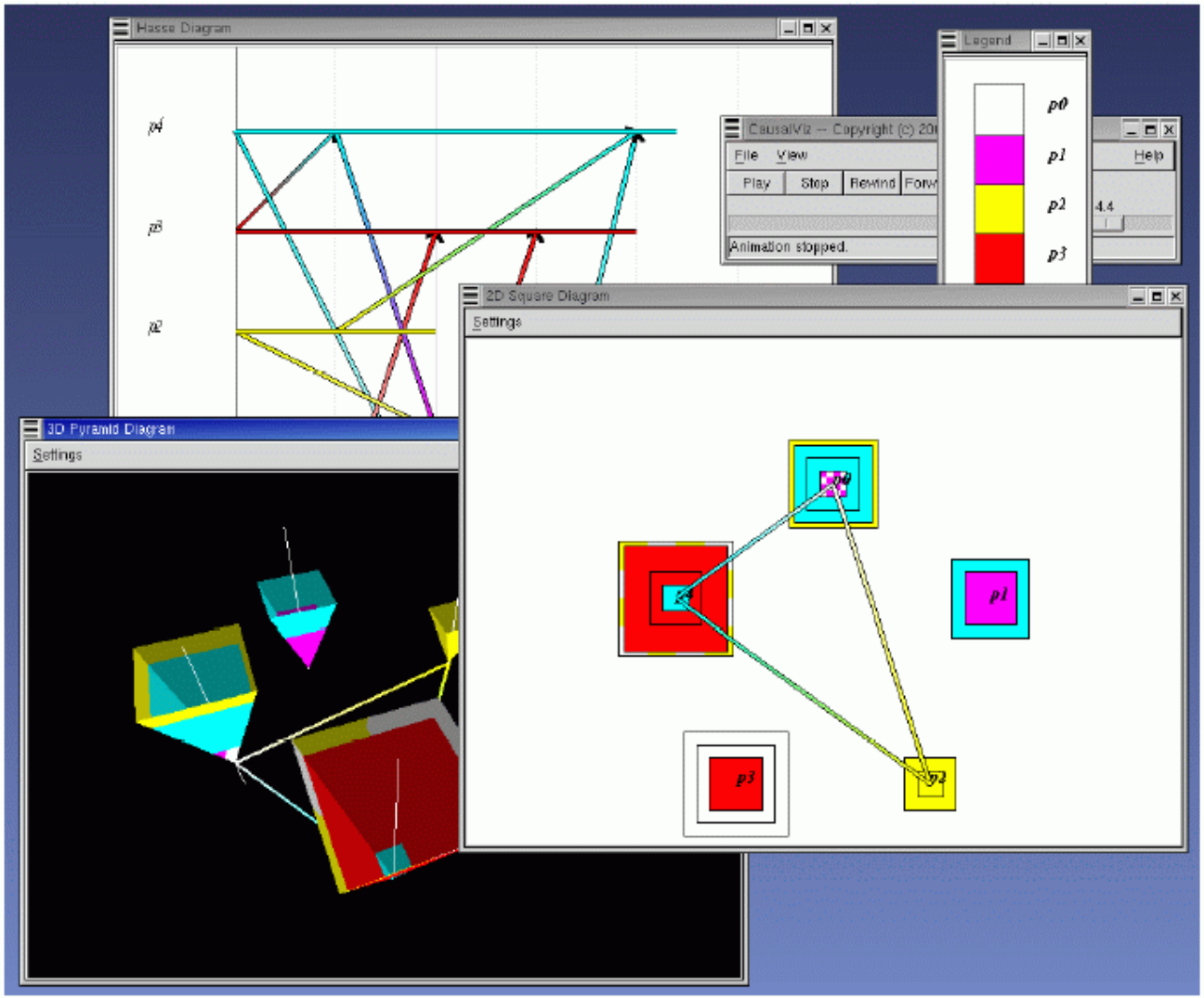
Nachrichten Austausch 4



Nachrichten Austausch 5 (Endprodukt)



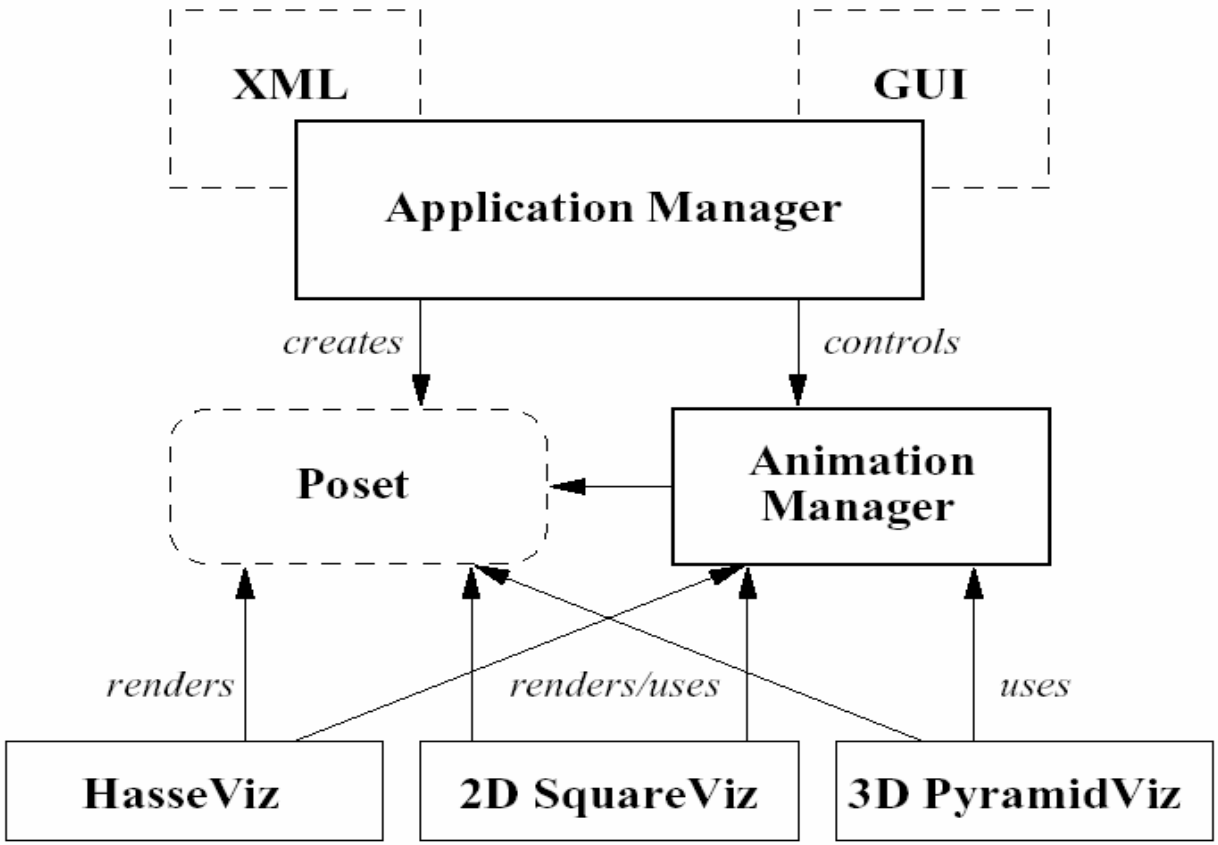
Das Animationsprogramm



Das Animationsprogramm

- CausalViz
  - Liest Szenarien aus einer XML-Datei und stellt diese dann dar, als:
  - 2D Quadrate
  - Weg-Zeit-Diagramm (Hasse-Diagramm)
  - 3D Pyramiden
  - 2D Polygone (nicht in Version 0.1.0 enthalten)

Systemarchitektur von CausalViz



```
  <?xml version="1.0" standalone="yes" ?>
- <poset>
  - <set>
    - <subset id="0" x="95" y="0">
        <element id="e0" time="0" />
        <element id="e1" time="10" />
        <element id="e11" time="30" />
        <element id="e12" time="30" />
      </subset>
    + <subset id="1" x="32" y="50">
    + <subset id="2" x="-12" y="50">
    </set>
  - <ordering>
      <order before="e1" after="e2" />
      <order before="e5" after="e6" />
      <order before="e7" after="e8" />
      <order before="e10" after="e11" />
    </ordering>
  </poset>
```

//Prozess mit Ereignissen

//Nachrichten

//send = before

//recv = after

Kommentar zum Programm

- Das Programm „sollte“ unter Linux laufen, mit den entsprechenden Gtk+/Gtk-- und OpenGL Bibliotheken und dem richtigen Compiler.

Fazit

- Vorteile
  - Die Abhängigkeiten zwischen den Prozessen sind leichter zu erfassen
  - Viele Informationen auf einen Blick
- Nachteile
  - Konzept macht sich nicht gut auf Papier oder Tafel
  - Starre Animation (feste Anzahl von Prozessen, starre Zuteilung von Zeitschlitzen)

Links

- Webseite des Projektes:

  Visualization of Causal Relations

  http://www.cs.chalmers.se/~elm/projects/causalviz

  ➔ Publikationen, Bilder und das Programm Causalviz stehen dort zum Herunterladen zur Verfügung

Ende

Oder gibt es noch Fragen?