# The 2006 IEEE/WIC/ACM International Conference on Web Intelligence

## Session A4 (WI): Social Networks

## *Mining and Visualizing the Evolution of Subgroups in Social Networks*

**Tanja Falkowski, Jörg Bartelheimer and Myra Spiliopoulou**
Otto-von-Guericke-University Magdeburg
School of Computer Science
Technical and Business Information Systems
Research Group on Knowledge Management and Discovery



## Motivation

- Effective collaboration is critical to strategic success of organizations see, e.g. [Cross & Parker, 2004], [Lesser & Storck, 2001], [Rodan, S. & Galunic, C., 2004], [Valente, T. W., 1995 ])
  - Decreasing learning curve of new employees
  - Reducing Rework
  - Generating new ideas for products and services and diffusion of innovations
- Collaboration takes place in networks that are usually not homogeneous but have structures
- These networks are not static but evolve over time
- To foster interactions between people, organizations are thus interested to
  - provide an appropriate organizational support as well as
  - a convenient technological infrastructure

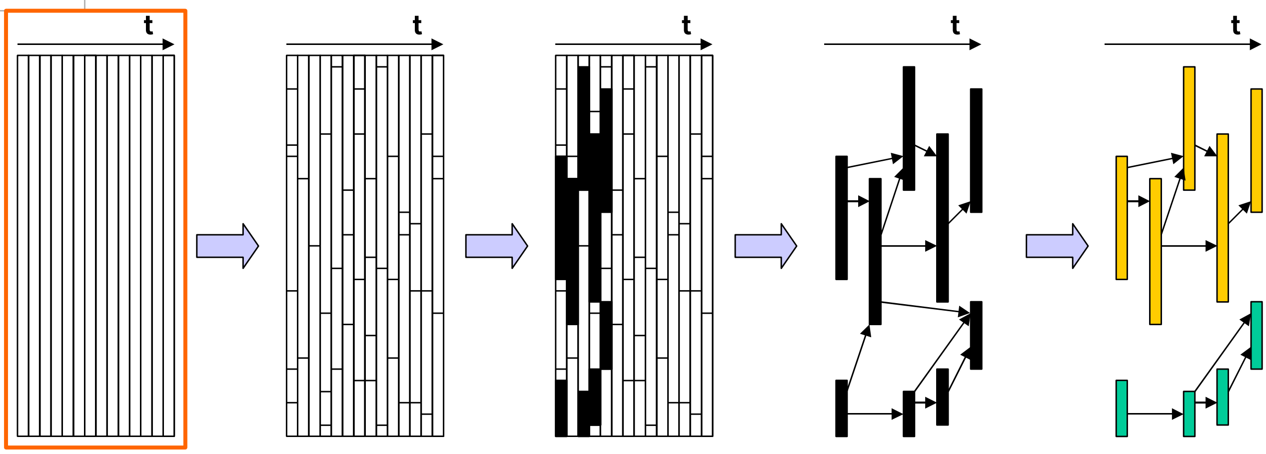
➢ Providing a method and tool to **analyze the temporal development of collaborations in the form of interaction in organizations**

## Road Map

- Graph-Representation of the Interaction Network
- Partitioning the Time Axis (Sliding Window Approach)
  - Dynamic view of the graph by splitting the network into equidistant slices that are defined by a time window
- Subgroup Detection in Partitions
  - Hierarchical divisive clustering
  - Quality measure to determine subgroups
- Observing Dynamics of Subgroups (Community Instances) and Communities
  - Display the subgroups along the time axis and allow for a variety of analysis settings
  - Temporal view of the network evolution



## Overview




**Step 1.** Partitioning the time axis

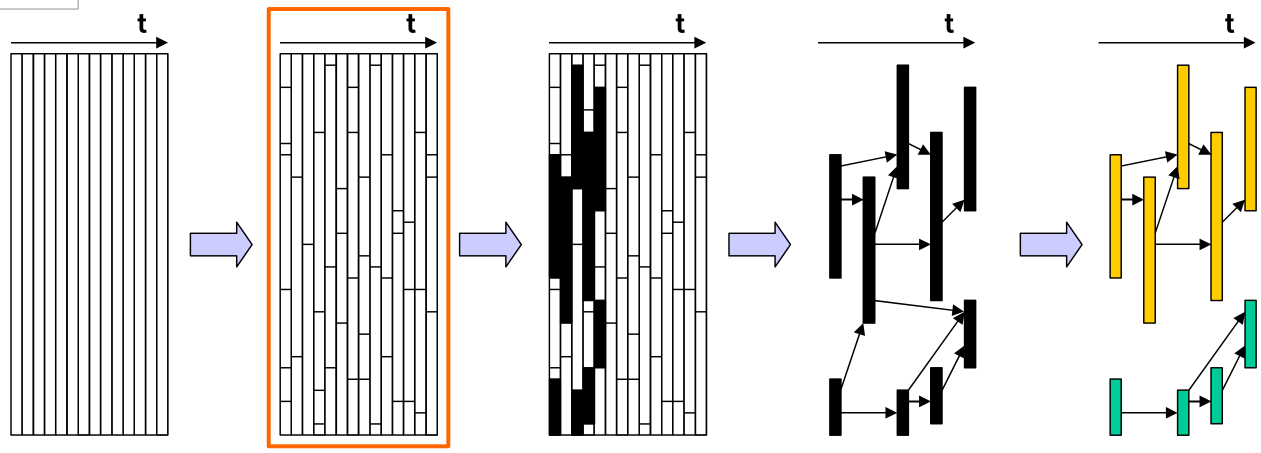
**Step 2.** First clustering to find subgroups (community instances) in time windows

**Step 3.** Detecting similar community instances in time windows

**Step 4.** Visualization of similar community instances

**Step 5.** Second clustering to find clusters of similar community instances

# Overview




**Step 1.** Partitioning the time axis

**Step 2.** First clustering to find subgroups (community instances) in time windows

**Step 3.** Detecting similar community instances in time windows

**Step 4.** Visualization of similar community instances

**Step 5.** Second clustering to find clusters of similar community instances

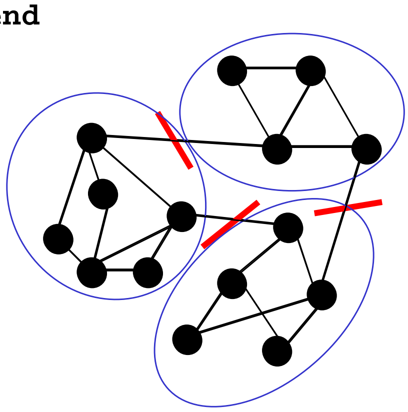


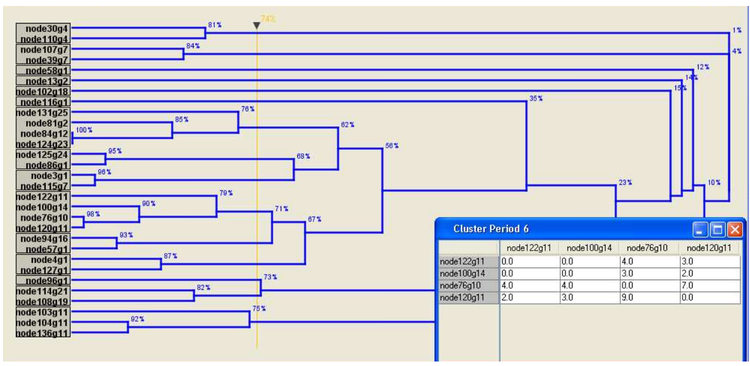
# Subgroup Detection
## Hierarchical Divisive Edge Betweenness Clustering

Hierarchical Divisive Edge Betweenness Clustering [Girvan & Newman, 2002]

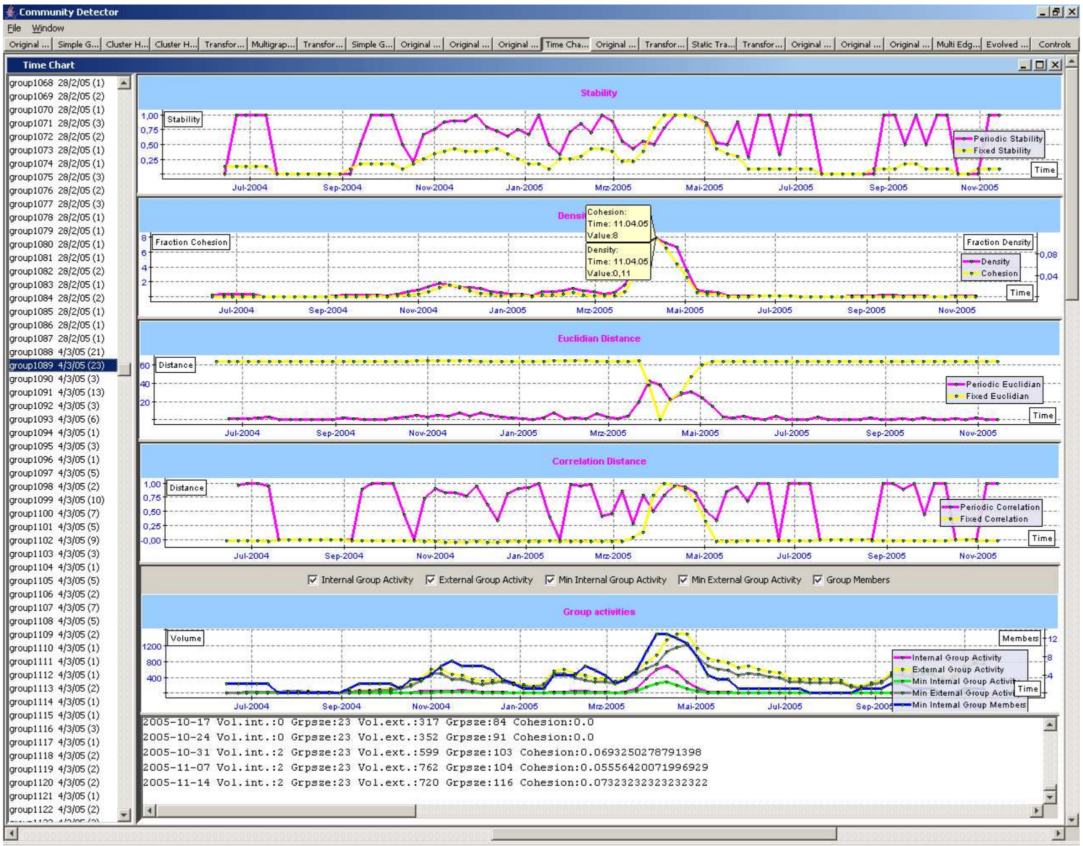
- Basic Idea: When a graph is made of tightly bound clusters, loosely interconnected, all shortest paths between clusters have to go through the few intercluster connections

```
repeat until no more edges in graph g_t
    Compute edge betweenness for all edges
    Remove edge with highest betweenness
end
```



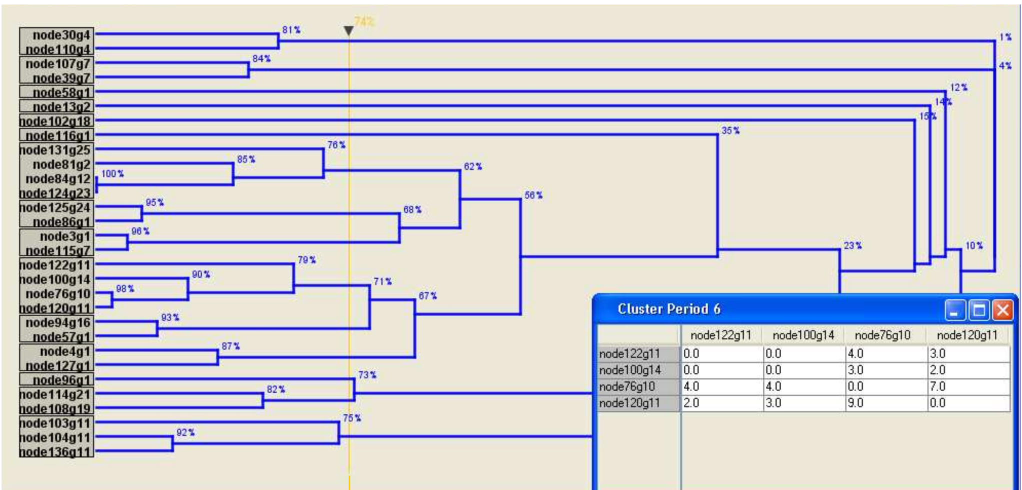


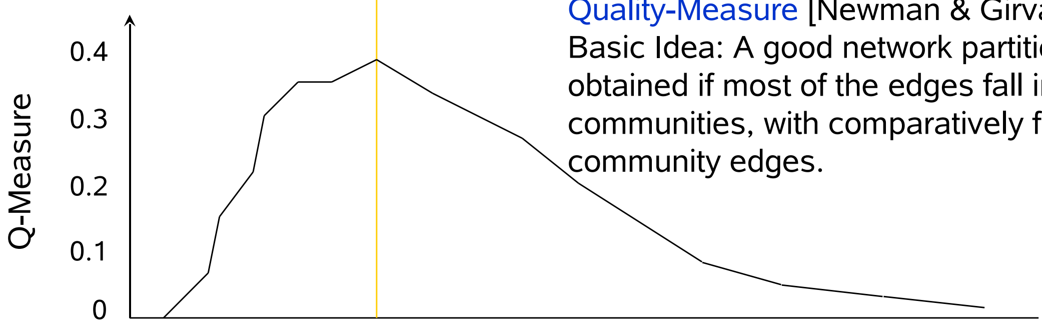
# Temporal Development of Subgroups





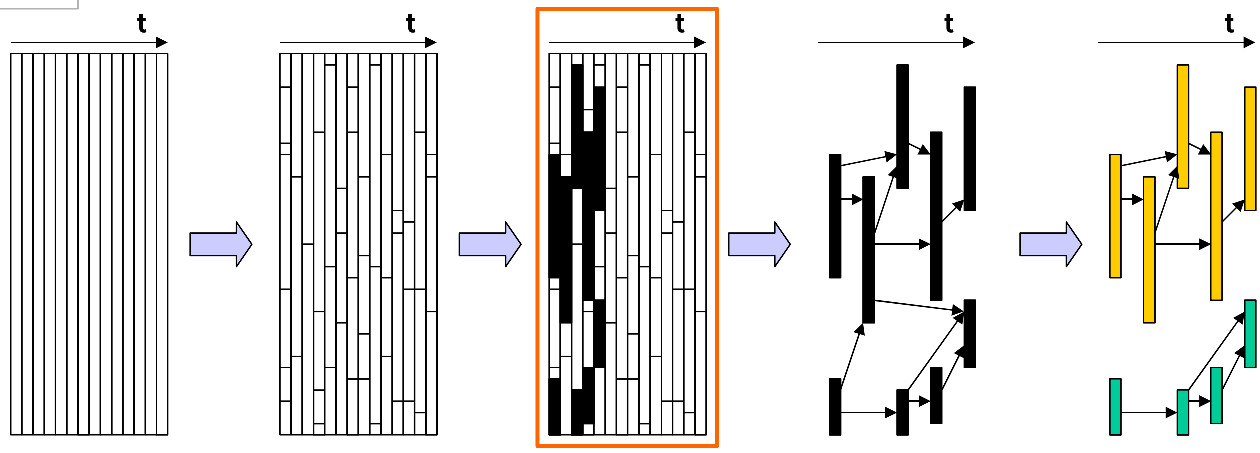

# Community Detection
## Quality Measure




Quality-Measure [Newman & Girvan, 2004]. Basic Idea: A good network partition is obtained if most of the edges fall inside the communities, with comparatively few inter-community edges.

# Overview




**Step 1.** Partitioning the time axis

**Step 2.** First clustering to find subgroups (community instances) in time windows

**Step 3.** Detecting similar community instances in time windows

**Step 4.** Visualization of similar community instances

**Step 5.** Second clustering to find clusters of similar community instances



# Finding similar community instances

- Detecting similar community instances over time
  - Defining the overlap of two community instances

$$overlap(x, y) = \frac{|x \cap y|}{\min(|x|, |y|)}$$

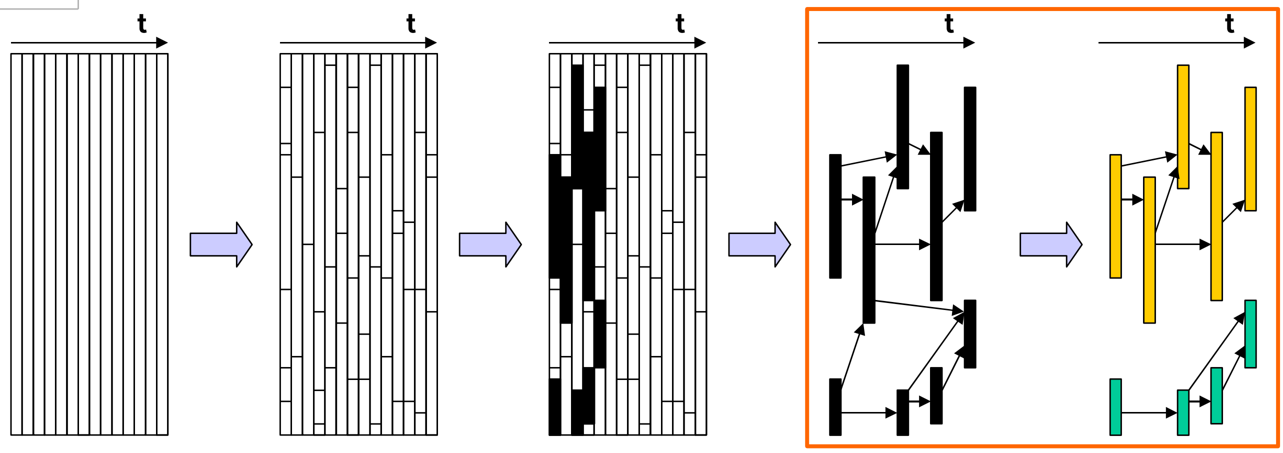
  - Defining the similarity of two community instances

$$sim(x,y) = \begin{matrix} 1 & overlap(x,y) \quad \tau_{overlap} \\ 0 & otherwise \end{matrix}$$

  - Defining the similarity of two community instances with upper boundary to the number of periods that separate the instances

$$similarity(x^{G_i}, y^{Gj}) = \begin{cases} 1 & t_j - t_i ¿ \tau_{periods} ¿ overlap(x, y) \geq \tau_{overlap} \\ 0 & otherwise \end{cases}$$

# Overview




**Step 1.** Partitioning the time axis

**Step 2.** First clustering to find subgroups (community instances) in time windows

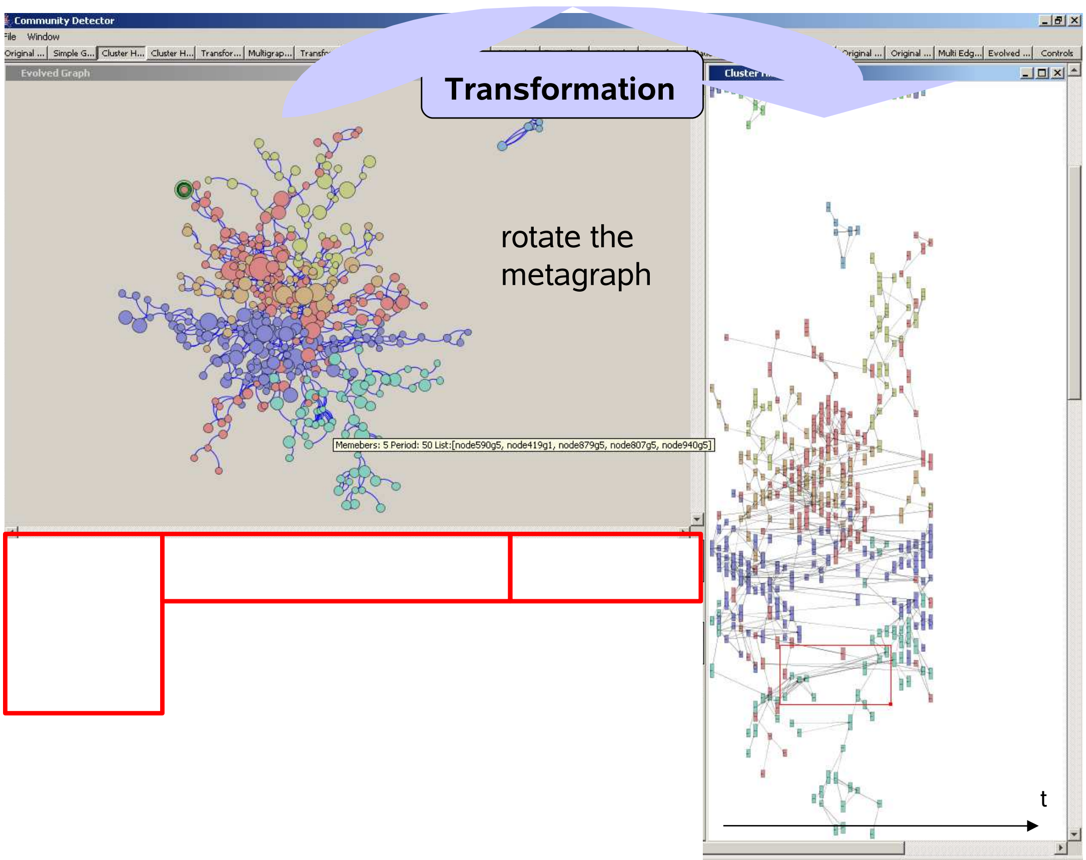
**Step 3.** Detecting similar community instances in time windows

**Step 4.** Visualization of similar community instances

**Step 5.** Second clustering to find clusters of similar community instances



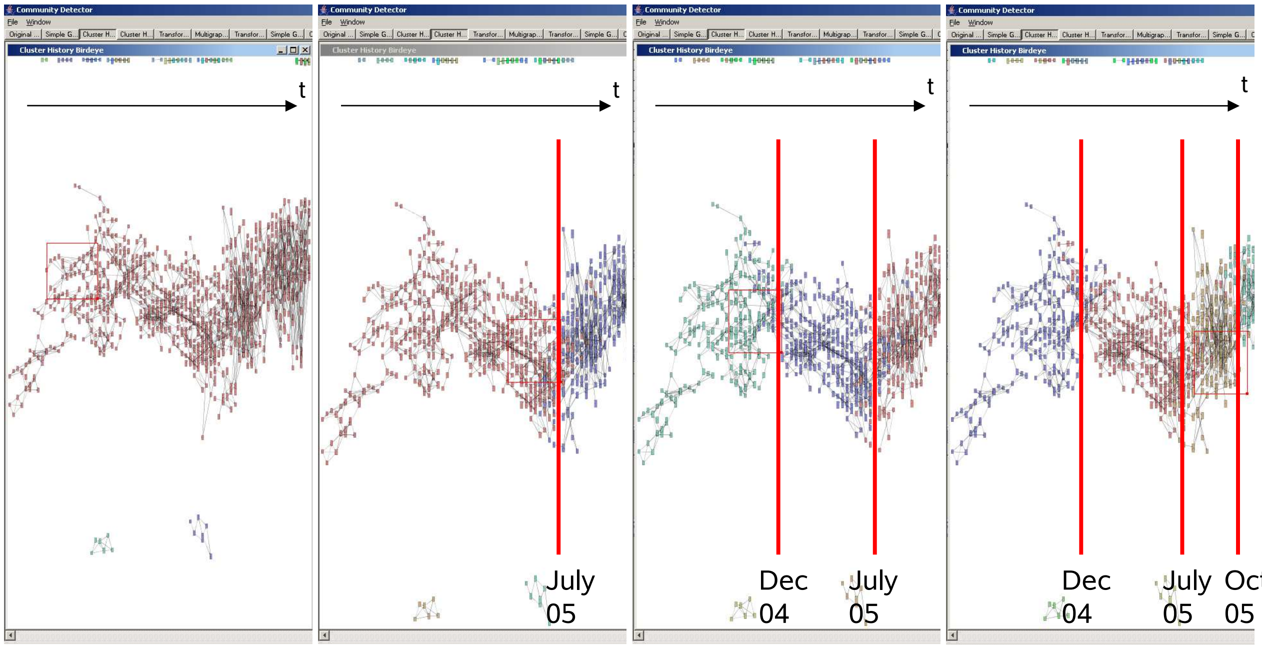
# Community Dynamics
## Community History View



# Community Dynamics
## Experimental Results

Number of clustering iterations (= number of edges removed):

0 27 38 48






# Outlook

Conclusion

- Tool to detect communities and to analyze communities over time
- Several forms of visualization and analysis settings

Outlook

- Analyze the interplay of different measures to assess the temporal development of subgroups
- Evaluate the impact of different parameter settings ($\tau_{periods}$, $\tau_{overlap}$, Window length) on results
- Community transition types (quantitative and qualitative assessment) and their implications (→ new data set)

**Tanja Falkowski, Jörg Bartelheimer and Myra Spiliopoulou**
Otto-von-Guericke-University Magdeburg
School of Computer Science
Technical and Business Information Systems
Research Group on Knowledge Management and Discovery