2006

Community dynamics mining

T. Falkowski
Otto-von-Guericke-University Magdeburg, falkowski@iti.cs.uni-magdeburg.de

J. Bartelheimer
joerg.bartelheimer@student.uni-magdeburg.de

M. Spiliopoulou
myra@iti.cs.uni-magdeburg.de

COMMUNITY DYNAMICS MINING

Falkowski, Tanja, Otto-von-Guericke-University Magdeburg, Universitätsplatz 2, 39106 Magdeburg, Germany, falkowski@iti.cs.uni-magdeburg.de

Bartelheimer, Jörg, Otto-von-Guericke-University Magdeburg, Universitätsplatz 2, 39106 Magdeburg, Germany, joerg.bartelheimer@student.uni-magdeburg.de

Spiliopoulou, Myra, Otto-von-Guericke-University Magdeburg, Universitätsplatz 2, 39106 Magdeburg, Germany, myra@iti.cs.uni-magdeburg.de

Abstract

In this paper we propose a model to analyze community dynamics. Recently, several methods and tools have been proposed to extract communities from static graphs. However, since communities are not static, but change over time, it is necessary to provide methods to determine and observe the community transitions and to extract the factors that cause the development. We regard a community as an object that exists over time and propose to observe community transitions along the time axis. For this we partition the time axis under observation by time windows. In each time window, a set of interactions between community participants is aggregated. These static networks are analyzed for subcommunities by applying community detection mechanisms. Through this we detect communities in each interval and can observe if communities persist over time or undergo a transition. We present community transitions and the observable indicators for the respective development. We furthermore present a software environment that incorporates several community detection and analysis methods to analyze community transitions. It supports a dynamic temporal community analysis and provides several forms of visualizations and analysis settings thus providing an interactive tool to observe community dynamics.

Keywords: Communities, Community Dynamics, Social Network Analysis, Visualization.

1 INTRODUCTION

In organizations people interact with each other for different purposes. Groups of people inside organizations who share a concern or a passion about a topic, and who interact to expand and exchange their knowledge and their expertise are called communities of practice (Wenger et al. 2002). Communities of practice can often be found in organizations where people usually know each other in person and face-to-face communication is the predominant form of knowledge exchange. However, besides meetings in person, communities of practice usually make use of technological infrastructure to coordinate, collaborate and communicate (Eales 2003, Cross et al. 2002). Communities that interact via a (Web-based) community platform are referred to as online communities (Preece 2000) or Web communities (Flake 2000). Besides the improvement of intra-organizational knowledge sharing, organizations are keen to extent communication with customers to increase customer loyalty (Manville 2004).

Irrespective of the type of communities and the members involved, organizations are interested to support community building to facilitate knowledge sharing. The success of communities depends on internal factors (e.g., a community leadership change) as well as external factors (e.g., publicity in mass media). The organization is, at least partially, able, to influence these factors by designing technological and organizational instruments that have a positive impact on the community.

In order to support organizations in this process, it is necessary to provide tools to determine community transitions since relations in communities are dynamic and change over time. Several tools such as SoNIA (Moody et al. 2005) and TeCFlow (Gloor & Zhao 2004) allow temporal social network visualization. Both tools visualize temporal social graphs by creating movies of graphs. However, since both tools work on the vertex and edge level and thus visualize changing behavior between single actors, it is not possible to explore the dynamics of groups. Turner et al (2005) proposed an ego centric visualization tool to analyze the range of variation in communication found in newsgroups. In contrast, we propose to visualize the temporal changes on the community level to allow for an exploration of sub-group dynamics. We present a dynamical temporal observation of communities along the time axis using time windows and regard a community as an object that exists over time. Whether a community exists is first determined statically at each time window, as described below. Then, its evolution is observed by comparing the communities appearing in each time window, according to a model of community transitions.

Furthermore, so far, community detection was mainly done on graphs based on aggregated data. This approach has two major shortcomings: i) since all interactions in time are treated equally, big aggregates dominate the results and not those that are most current and ii) transitions in the interaction behavior such as a merging communities or a periodically active community can not be observed. We thus regard a community as a dynamic object that exists over time. We partition the community interactions along the time axis into intervals and determine the appearance of communities (more precisely sub-communities[1]) in each time window. Thus, we model a community as a static instance and apply methods to detect the community structure in each interval. We obtain a dynamic view by comparing the communities in different time windows and assess in which period which communities are active. The changes in the community structure are than visualized and the user can choose different settings to further explore the community dynamics according to his analysis objectives.

In the next section we discuss how we model the community interactions and how sub-communities can be detected in static graphs. We furthermore present how the changes in the interaction over time can be modeled to explore community dynamics. In Section 3 we briefly present community transition types, how they can be observed and possible triggers for the development. In Section 4 we discuss

[1] Some researchers observe the entire graph as a (Web) community. Sub-structures in this graph are thus sub-communities. However, those sub-graphs are also often referred to as communities.

our community dynamics miner and present first analysis results of an experiment with community data. Section 5 contains the conclusion and an outlook.

2 MODELLING COMMUNITY DYNAMICS

The formation of sub-groups of actors can be observed in many networks. To find these community structures is therefore of interest for many research fields. Examples are social sciences such as citations networks (Jeong et al. 2003, Newman 2001), biology such as genetic networks (Wilkinson & Huberman 2004) or food webs (Dunne et al. 2002), or in computer science such as the WWW (Kleinberg & Lawrence 2001) or email log files (Tyler et al. 2003). To observe communities in large networks, we need a community detection method to find dense network partitions that are oft interest. The problem of detecting communities has also been discussed, e.g., in (Cortes et al. 2001, Radicchi et al. 2004, Aggarwal & Yu 2005). In the following Section 2.1 we briefly discuss which detection method we apply to find the communities in a given network. In Section 2.2 we describe how this method is extended to allow for a detection and analysis of dynamics in communities.

2.1 Community Representation and Detection

So far, communities have been regarded as a static phenomenon and aggregated data over longer periods has been used to determine communities. The aggregation of the interactions between community members has two main drawbacks: i) big aggregates dominate the results and not those that are most current since all points in time are treated equally and ii) transitions in the interaction behavior can not be observed. We therefore propose a dynamical temporal observation of communities along the time axis using time windows and regard a community as an object that exists over time. Whether a community exists is first determined statically at each time window, as described below. Then, its evolution is observed by comparing the communities appearing in each time window, according to a model of community transitions. The necessary steps (network modeling, graph decomposition and analysis of results) are described in the following.

First, we model the network of interactions in a way suitable to find communities. We do so by defining a graph G = (V, E), in which V denotes the set of vertices (nodes) and E the set of edges (i, j), with $i,j \in$ V. Each community member i is denoted a distinct vertex and an interaction between two members i and j, e.g., an e-mail exchange, is represented as an edge (i, j). We quantify the interaction between two members by assigning a weight $w(i, j)$ to the edge (i, j). Appropriate weights are "number of messages exchanged" or the "total length of all messages measured in characters". We stress here that this weighting scheme would favor "old" members of the community over newcomers if weights are aggregated over time. However, we build the graph and assign the weights for each time window, thus avoiding this caveat.

Next, we decompose the graph into communities. We define a (sub) community as a subset of vertices within a graph with a high degree of interaction among the participants. We apply a hierarchical divisive clustering approach that divides the graph by the iterative removal of edges. The edges that are removed should be those that do not contribute to a community. We must therefore define a good criterion of interaction between vertices. This criterion must be good in the sense that it gives an indication for the membership of one vertex to a community. Depending on the size of the network we apply the following two measures to find the edges to be removed:

a. Our first measure is the *edge betweenness* score proposed by Girvan and Newman (2002). The betweenness of an edge is the number of shortest path between pairs of vertices that run along it. It is based on the assumption, that the few edges between communities have more "traffic", as, e.g., an information flow between vertices in two communities has to travel along these edges. The hierarchical clustering algorithm iteratively removes the edges with the highest edge betweenness score. We apply this method to a multigraph as described by Newman (2004) to include weighted edges. Each edge betweenness value is divided by the edge weight. Therefore, the edge betweenness value between two very connected pairs is lowered so that rather weak connected pairs are separated faster than strong connected ones. Due to the high complexity to calculate the edge betweenness – $O(m^2n)$, where m is the number of edges and n the number of vertices – it is only applicable for small networks with up to a few thousand vertices.

b. Our second measure is proposed by Radicchi et al. (2004) and has a lower complexity: The *edge clustering coefficient* which is based on counting short loops of edges. The basic consideration of this method is that edges between communities belong to less short loops, e.g., with lengths 3, as the completion of short loops requires a third vertex that also runs between the same communities. Inter-community edges thus have a low number of loops. The edge clustering coefficient C_{ij}, for an edge between vertex *i* and *j*, is defined as

$$C_{ij}^{(3)} = \frac{z_{ij}^{(3)} + 1}{\min(k_i - 1, k_j - 1)},$$

where z_{ij} is the number of triangles to which the edge belongs and k_i the degree of vertex *i*. $min(k_i-1, k_j-1)$ is the maximal possible number of triangles including that edge. To avoid penalizing edges that belong to no triangles the numerator is increased by 1. For edges that are not strongly connected in a community, C_{ij} will be small, as they do not belong to many triangles. The edge with the lowest value will be removed and afterwards C_{ij} will be recalculated for the remaining edges. Since the clustering coefficient is a local measure it can be calculated faster – $O(m^4/n^2)$. This algorithm is particularly suitable for dense networks with many triangles, such as social networks (Watts & Strogatz 1998).

The results of the hierarchical clustering are presented in a dendrogram, a tree diagram, which illustrates the community structure of the graph. Since we have no a priori knowledge about the number of communities that exist in a network, we need an indicator on where to partition the dendrogram to obtain a meaningful network partition. For this purpose, we use the quality function proposed by Newman and Girvan (2004) to determine the best dendrogram cut which is based on the concept of modularity. The quality function Q is defined as:

$$Q = \sum_i e_{ii} - \sum_{ijk} e_{ij} e_{ki} = Tr(e) - \left\| e^2 \right\| ,$$

where e_{ii} is the fraction of edges in the original network that connect two vertices inside the community *i* and e_{ij} the fraction of edges that connect vertices in community *i* to those in community *j*. $||x||$ indicates the sum of all elements in x. Q has a value between 0 and 1. Values approaching Q = 1 indicate strong community structure. Modularity values that are greater than 0.3 appear to indicate significant community structure (Newman & Girvan 2004). The higher the value the more well-defined the communities are and the most accurate the partition is. To find the best partition, Q needs to be optimized over all possible network partitions.

We validated the selection of measures and the hierarchical clustering algorithm by applying them on the datasets presented in Newman & Girvan (2004). For these datasets, we have obtained comparable modularity measures.

2.2 Modelling Community Dynamics

Since the interactions between participants and the set of participants are not static but change over time, we use the representation of the network but consider the graph as dynamic. Vertices as well as edges appear and disappear from the graph through time. We define the dynamic graph g_t as a graph in interval *t*. g_t consists of all vertices and edges that are active in an interval *t*. If all interactions would be aggregated over time to G by summing up all g_i where $i = 1, \ldots, t$ all information about the temporal development would be lost. Therefore, we define g_t as a sliding window over time interval *t* that spans a set of interactions. In other words, all interactions that take place in this interval t are aggregated to g_t.

To find the right length of the interval is very challenging. Each graph should show enough change, but not too much in order to give information about network dynamics. The question concerning the right rate of change can be described with respect to different levels (e.g., fast, medium, slow). The meaning of these terms depends on the context since the scale will vary across different relations. For example, the rate of change for the relation “e-mail-exchange” will be much shorter than for “trade relationship”. Thus, chunks of time that capture the nature of events must be identified. The interval size for the respective relation under observation is not computed automatically. We assume that the intervals are known and provide a mechanism that allows the user to choose the interval size.

After defining the interval we would intuitively partition the graph over time into equidistant time slots, each slot starting when the last slot finished. This modus is called a non-overlapping sliding window. An overlapping window partially overlaps with the prior window. The degree to which it overlaps must be defined. We apply an overlapping window since it smoothes out the gaps that sometimes occur between two intervals. It can be interpreted as a moving average (Moody & McFarland & Bender-deMoll 2005).

Each window is considered a static representation of the network in the chosen interval. At first we apply the community detection mechanism as described in Section 2.1 to obtain a community structure for g_t. $C_t = \{c_{t1}, c_{t2}, \ldots, c_{tn}\}$ is the set of communities in g_t. To determine whether a community persists over time we must be able to assess if a community c_{xn} in g_x is the same as a community c_{ym} in the interval g_y, where n, m ≥ 1 and y > x. Qualitatively we would define that a community in a subsequent interval is the same, if the characteristic features are similar. First of all, this would be the set of participants. We therefore define that community c_{xn} and community c_{ym} are the same if a given percentage of members in community c_{xn} in g_x is also a member of community c_{ym} in the interval g_y. The appropriate level of the percentage depends on the community type, the type of relations and the intent of the observer. If a community consists of a small set of very active core members and a high number of less connected members that often change, the percentage should be rather small. Otherwise, the community might not be considered the same just because many of the other “uncharacteristic” members changed, even though the most active core members are still detected as a community in different intervals. The percentage can therefore be chosen by the user of our system according to his needs.

The temporal developments of the communities are then visualized in a community history view. The procedure is described in more detail in Section 4. In the next section we discuss observable community transitions and their possible causes.

3 OBSERVABLE COMMUNITY TRANSITIONS AND TRIGGERS

Taking effective methods to support communities implies that we are able to recognize the community’s actual status as well as ongoing transitions. For this, it is necessary to know about transition types, how to observe them and the triggers that evoke these changes. Ideally, real changes should be distinguished from accidental changes. Wenger, McDermott and Snyder (2002) view communities as a continually evolving “living being” and observed five stages of development: potential, coalescing, maturing, stewardship and transformation. Along this lifecycle, they pointed out necessary activities that need to be taken to support communities in their development. Communities may develop according to this lifecycle, but often they stay at a certain stage for extended periods and then suddenly evolve or they move back- and forward between the stages. Furthermore, it is often difficult to determine the stage as some characteristics are fulfilled, others not. However, the lifecycle offers some indications about possible types of transitions in communities.

In the following we briefly discuss four transition types, observable indicators to assess the change and internal as well as external factors that influence the development of the community.

3.1 Community Transitions and Transition Indicators

3.1.1 Community persists

A community must be established in order to say if it actually undergoes a development. In an organizational context, a community is established if it maintains a coherent, ongoing identity known both to itself and the environment, e.g., the larger organization, and if it functions as a goal-directed, self-managing system. To observe the temporal development of a specific community we must therefore be able to find corresponding communities at different points in time (Hopcroft et al. 2004). As described in Section 2.2 we define that a community persists over time if a given percentage of participants is detected in several periods as one community.

However, the periods the community shows up must not necessarily be consecutive. It might happen that a community is detected in interval *t*, not detected in *t+n* but shows up again in *t+m* where $m > n \geq 1$. Thus, we define a community still as the same community even though it was inactive for some time. The length of the period of "allowed" inactiveness can be chosen by the user.

The phenomenon that a community persists can be divided into the sub-transitions: The community grows or the community declines. We define that a *community grows* if in the period under observation its number of participants increases. This evolution is considered more evident if at the same time the number of edges in relation to the number of vertices in the community increases. To identify a significant development this observation should be made over a certain number of periods. The actual number of observed time steps depends on the length of the intervals. If the number of edges in relation to the number of vertices inside a community increases in several consecutive periods we observe that the community evolves. Parallel, the average shortest path between nodes decreases. If the number of vertices remains constant at a high level but the number of edges increases, we observe a community that matures. A *community declines* if, over several periods, the community members become less active (number of edges or edge weights decrease) and/or members leave the community (number of vertices decreases). If the number of edges and the number of vertices decreases significantly, the community might disappear.

3.1.2 Community disappears

„When we say that a community disappears, what we mean is that the community is no longer recognized as a separate, functioning system with a known, ongoing identity." (Gongla & Rizzuto 2004) There are many reasons why communities disappear. An abrupt disappearance without any prior indication is unlikely. Rather, we expect that it takes quite some time to fade. Members gradually leave the community, slowly decreasing their participation. During this drift phase, new members rarely join the community since the community is not visible to them and any overtures may get little response. Core members leave and are not replaced. Members stop identifying with the community and less and less activity can be observed. Since this is usually a very slow process it is hard to determine when exactly the community finally ceases to exits.

3.1.3 Community merges

Communities merge with other communities. We can distinguish between mergers of equal and unequal communities (Gongla & Rizzuto 2004). In the former case, communities find that they have a lot in common and individual members find themselves in both communities. The knowledge domain of each community is either similar or complementary to the other community. The two communities disappear and a new community is established in their place. A merger of un-equal communities may be observed, if a community constitutes a specialized sub-domain of a larger community. It may join willingly the broader community or it may be absorbed by it. We can observe a merger of multiple communities by examining the historic view of the communities as shown in Section 4.

3.1.4 Community splits

Analogous to a merger of communities, a community can split into multiple communities. A split can be determined if we observe the time axis from the opposite direction. A community split might happen, e.g., if a sub-group with a more specialized knowledge domain developed which at some point separates from the community. Further transition triggers are briefly discussed in the following.

3.2 Causes of Community Transitions and their Indicators

The reasons why the structures of communities change are complex. Examples for triggers are a change in the organizational or technological structure or in the knowledge domain of the community. Both can have a high impact on the behavior of the community members. The challenge is to detect the change that triggers the development and the development itself to be able to relate triggers and transitions. We discuss two important transition triggers and how they can be observed.

3.2.1 Community Leadership Change

The activity in communities is to a big part determined by core members who have a high influence on the development of the community. Sometimes this is a single person. However, often the core consists of a (small) group of people that has established the community and that feels responsible for its continued existence. Several studies show, that only a small minority of the users posts the majority of the messages and thus stimulate activity. Organizations should identify influential members and try to prevent leadership changes or take necessary actions in case a leadership change is necessary.

A leadership change can be best observed on the vertex level. Each vertex has a unique label that it retains all along. The variation of certain properties indicates the change in personal behavior. The properties that give indications about a changing behavior of one member are the degree, the edge clustering coefficient and the vertex betweenness centrality.

- The *degree* of a vertex is one indicator for its centrality in a graph. An increasing degree, compared to the average degree, over several periods is an indication for an increasing integration into the network. If the majority of neighbors belong to the same community, we can assume that the vertex is more central. Otherwise, if many neighbors belong to other communities a vertex with a high degree may act as a bridge between two or more communities.
- As described in Section 2.1, the *edge clustering coefficient* is a local measure that gives indication about the transitivity of the edge. An increasing coefficient indicates a higher probability that one or both vertices, which are connected by this edge, become a member or a more central member of a community, depending on the previous scores. A decreasing coefficient indicates that the nodes move to the border of the community. To distinguish between a node at the border of a community and a node, which acts as a bridge between two or more communities one has to observe the edge betweenness scores. If the edge clustering coefficient is low, but the edge betweenness is high, the probability that the node acts as a bridge is higher.
- It has been reported that vertices with high *vertex betweenness centrality* values tend to play a more important role compared, e.g., to vertices with a high degree in keeping communities connected (Baur & Benkert 2005). If vertices have established a high betweenness value, it is very likely that they have an important role in the network. A decreasing betweenness may indicate that a core member becomes less active, which might result in a less active community. However, this measure makes no distinction between vertices in different groups or between geodesics, which remain in one group, and those, which cross to different groups.

It should be noted, that if a leadership change is observed as a change in the community structure, it most likely invokes further changes in the community that might be observable in preceding time windows. Thus, a change in the community structure it not only in itself an interesting transition but it can also be an indicator for a possible transition in the future.

3.2.2 External Influences

Advertising or publicity in the mass media may influence the development of a community positively as well as negatively. A positive report about a community in the media or an award can be very motivating, a negative publicity or a slating review can have devastating effects. Furthermore, seasonal variations can be observed, depending, e.g., on the domain of the community. A political community might be more active in the run-up to elections, a community of sailors during summer time.

External influences as advertising usually have an immediate effect on the community. A dramatic increase in new members might be a result of a positive publicity campaign. These changes are observable by comparing global properties of the graph in different intervals and give indications for structural changes. These are the number of vertices and edges, the average shortest path, the diameter of the graph, the density of the graph, the modularity of the graph (Q-measure) and the number of components, which indicates how connected a graph is. Furthermore, the graph clustering coefficient, a measure for the network transitivity, can provide good indications for structural network changes.

4 VISUALIZING COMMUNITY DYNAMICS

To track the development of established communities and to visualize these transitions we developed a software environment that supports temporal graph analysis based on the community detection methods described in Section 2. The transitions that we consider are displayed in Table 1. The main design goals of our tool are i) to decompose the network into communities (cf. 4.1), ii) to provide an interactive visualization of the decomposition (cf. 4.2) and iii) to provide an interactive visualization of the community development (cf. 4.3). In the following section, we describe the functionality of the software by describing how a user does proceed when performing a community dynamics analysis.

Transition	**Description**
Community Persists	c_i in g_t is matched by a community c_j in g_{t+n}, $n \geq 1$
Community Grows	c_i in g_t is matched by a larger community c_j in g_{t+n}, $n \geq 1$
Community Declines	c_i in g_t is matched by a smaller community c_j in g_{t+n}, $n \geq 1$
Community Disappears	c_i in g_t is not matched by any community c_j in g_{t+n}, $\forall\ n \geq 1$
Community Merges	c_i in g_t is not matched by a single community in g_{t+n}, $n \geq 1$, but the combination of two or more communities in g_t match c_j in g_{t+n}
Community Splits	c_i in g_t is not matched by a single community in g_{t+n}, $n \geq 1$, but the combination of two or more communities in g_{t+n} match c_i in g_t

Table 1. Community transition types and their description

4.1 Community Detection in Static Graphs

At first the relationships between community members that are stored in a database are transferred in a graphical representation where actors are represented as vertices and the relation between members, e.g., the communication activity is represented as a weighted edge between two actors. We implemented a Kamada-Kawai graph layout which positions the vertices so that the Euclidean distance between them is as close as possible to the graph-theoretic (path) distance (Kamada & Kawai 1989) (see left screenshot in Figure 1). The user can move vertices and zoom into the graph to investigate the structure in more detail.

The structural changes in the aggregated graph as well as the graph over specific periods can be analyzed by observing the curves that display the graph clustering coefficient and the average shortest path length (see right screenshot in Figure 1). Further statistics that provide indications of the network status for each period are displayed in a table. Currently, we present for each period the number of interactions, of vertices and of edges, minimum, maximum and average degree, the diameter, the edge clustering coefficient and the average shortest path. However, other community change indicators can be included easily, e.g., the vertex clustering coefficient, vertex betweenness or closeness centrality.

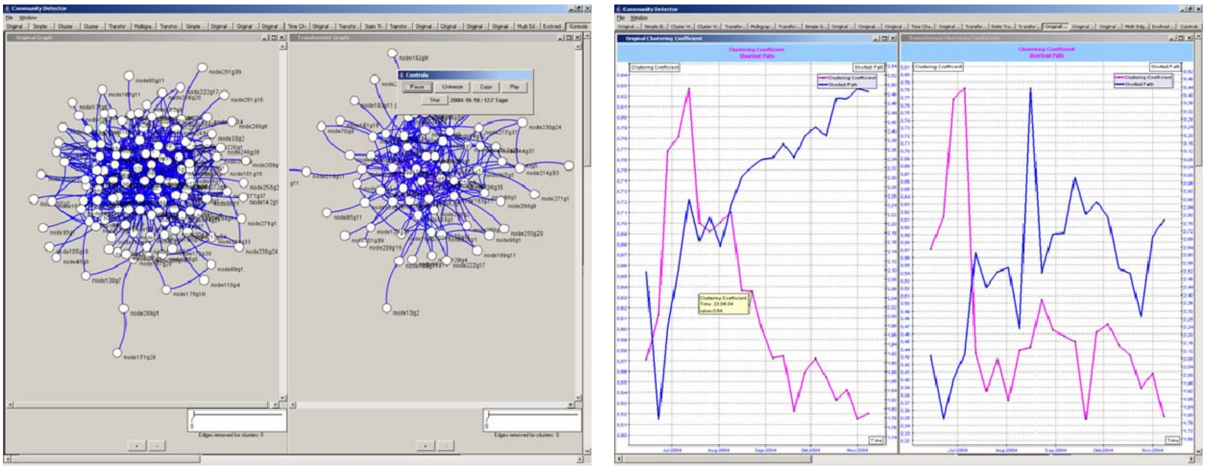


Figure 1. *Left side: Left graph displays all interactions in aggregated form as a graph. The right graph shows all interactions for one interval. Right side: The diagrams show the corresponding curves for the graph clustering coefficient and the shortest path over time for the aggregated graph (left diagram) and the graph on the right (right diagram).*

4.2 Interactive Visualization of Community Structure

We find the communities in each interval by invoking the hierarchical divisive clustering algorithm described in Section 2.1. The results of the clustering are displayed in a table and a dendrogram. Again, each row in the table represents one period. For each period the results for the best clustering according to the modularity measure are presented. Further statistics include: The modularity (Q-measure), the number of communities found, the size of the smallest and largest community, the number of edges to be removed to obtain the maximum modularity and some other statistics.

The user can experiment on the impact of the Q-measure threshold and tune it to the most appropriate value for him: To do so, he only needs to move a slider in the dendrogram (see description of Figure 2) and observe how the community structures change, both in the graph visualization, where edges are added resp. removed, and in the table entries, which are updated accordingly. Besides the dendrogram on the left, the right screenshot in Figure 2 has two other major areas: a list of all detected communities in the small middle window and the curves on the right. The right area has three sub-areas, uppermost, middle and lowermost. The horizontal axes represent the respective time windows. The vertical axes are described in the next paragraph. In the section underneath the three diagrams on the right, a list of all members of the chosen community is displayed.

The lowermost curve displays the total number of interactions between the chosen group (*Internal Group Activity*) and the total number of interactions of all group members with other participants of other communities (*External Group Activity*). The *Min Internal Group Activity* und *Min External Group Activity* represents the number of reciprocal interactions between two actors. It can be seen that the chosen group is only active for about 6 weeks (Internal Group Activity), but some members have an active relation with external participants. The middle and the uppermost diagram show how similar the internal community interaction behavior is over time. In the middle diagram the vertical axis depicts the correlation distance as a similarity measure for the groups in different periods by transforming the matrices of two periods into two vectors to calculate the correlation. It can be seen that the group shows up in two time windows with almost the same members but the structure changes very quickly and the group disappears. In the uppermost diagram the y-axis displays the Euclidian distance as a similarity measure. The more similar a group interaction in two periods, the lower is the value of the Euclidian distance. For both measures we compare the similarity between the chosen interval and all other intervals (*Fixed Correlation* or *Fixed Euclidian* respectively) and for two succeeding intervals (*Periodic Correlation* or *Periodic Euclidean* respectively).

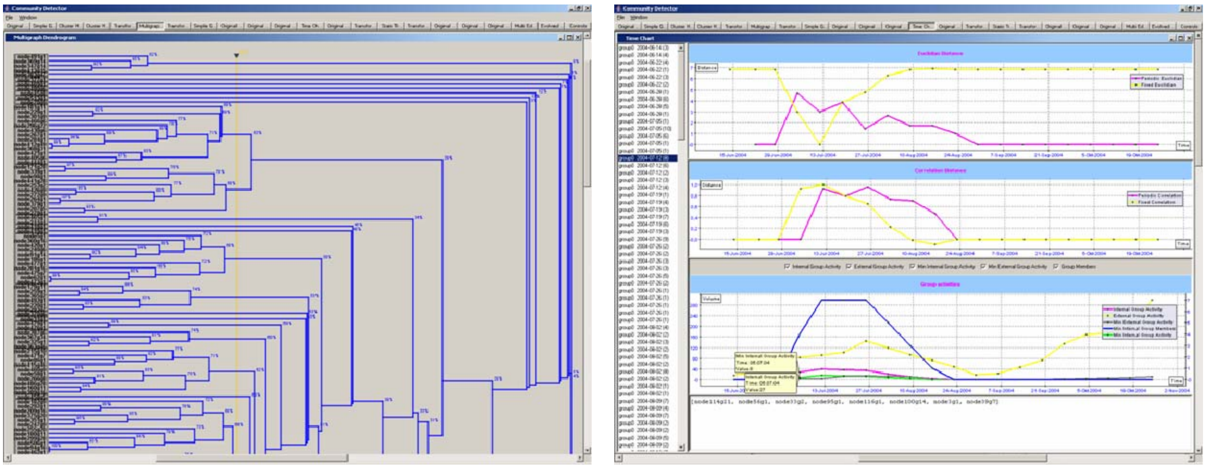


Figure 2. Left side: Results of the clustering are displayed in a dendrogram. The best dendrogram cut is calculated using the modularity measure and represented by the yellow slider which can be moved. Right side: Temporal development of a chosen community.

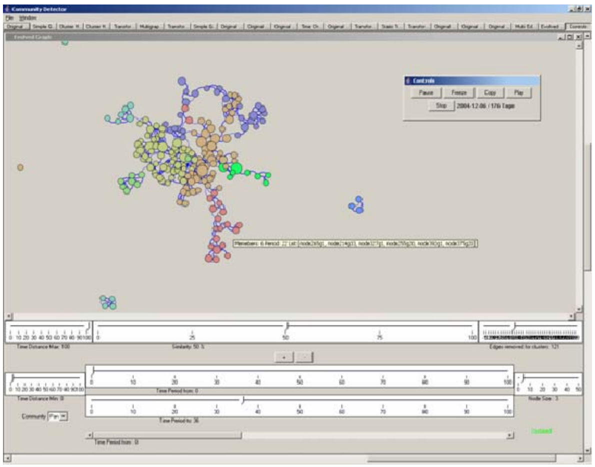
4.3 Interactive Visualization of Community Dynamics

In Figure 3 on the left side we see a static representation of a temporal community evolution. In this visualization, each detected community is represented as a vertex. The size of the vertex corresponds to the size of the community. The minimum community size is defined by the user. Edges are used to represent similarity of communities. Thus, vertices that are connected by an edge are similar. In the presented screenshot, communities are connected if at least 50 percent of the members belong to both communities. Communities with the same members over several periods are positioned closer in the graph whereas communities with no members in common should be more separated from each other. Furthermore, the different colors help to distinguish between similar communities and those that are not. By this, the user can observe how communities stand to each other over a period of time.

The user can choose for how many periods the community must at least exist to be displayed. If a long period is chosen, the user obtains only long-term community whereas in another case it might be of interest to find only short-term communities. Another slider for the time distance defines how continuous the communities are connected, separating communities by a maximum distance. Furthermore, one can define the observation period and filter the vertices so that the communities are displayed only in a selected period. The described properties can be used to filter communities and their connections so that the graph only shows data that is useful for a particular analysis.

Note, that in the obtained graph in Figure 3 on the left, the temporal development can not be observed, as the communities are only displayed according to their similarity. In a next step, the filtered and clustered data is copied to a community history view, which allows seeing temporal developments by using the coordinates from the graph and putting the vertices on the horizontal axis according to the period they appear in (see Figure 3 right side). The position transformation allows tracking the development along the time axis. Each community is now represented as a rectangle where the height of the rectangle corresponds to the size of the community. All communities that are considered as similar according to the actual setting are connected by edges and have the same color.

The left side of the screenshot on the right in Figure 3 shows all communities over time in an overview window. The right side displays a cutout to allow for a more detailed view. In the lower part of the overview window we can see a community in light blue and one in red that exist over just a few periods. In the cutout view we can furthermore observe a community in yellow that existed over a longer period. We can see that smaller communities merge to one big but split very fast to several small communities. However, we can also observe, that the used data unfortunately does not exhibit a stable community structure as the interactions between members are fluctuating very fast.



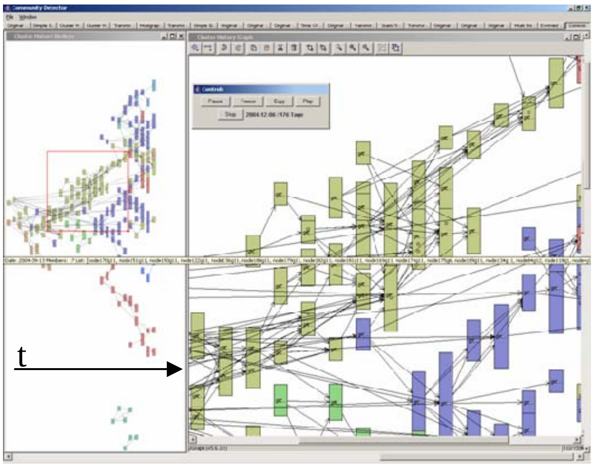


Figure 3. *Left screenshot: Visualization of communities based on the Euclidian distance. Right screenshot: Community history view along the time axis.*

5 CONCLUSION AND OUTLOOK

In this paper, we discussed a model to analyze community dynamics. We presented our interactive software tool that visualizes the community dynamics by partitioning the graph defined by a time window. Each window is clustered and the discovered communities are displayed. The obtained visualization allows for a better temporal overview in order to see how the communities are related and how they develop over time. The software tool is interactive and flexible: It offers several visualization forms and provides the user with many analysis settings to detect and explore community dynamics. The findings of a community dynamics analysis can be used to support organizations in designing technological or organizational measures to improve the community platform and its environment.

However, as in any observation of online communities, we are only able to visualize a certain amount of communication in the organization. The interpretation of the results could be enhanced, if information about "offline" communication could also be incorporated in the analysis.

Further research could focus on how to apply analysis findings in dedicated actions. Thus, community designers could improve/enhance the user experience and thus foster the community platform. The results of past changes might also be used to predict future community developments. Our aim is furthermore to improve the quantification of the weight of the edges. Counting the number of interactions may not represent the quality of the interaction. For example, the incorporation of the content of the interaction could provide helpful indications on the quality of the interaction.

Furthermore, the sliding time window algorithm has still problems to handle strongly fluctuating community activities. This makes it complicated to find the right length of the time window. When the number of actors and interactions increases quickly, the graph becomes large and dense making it hard to analyze the community structure. Therefore, it seems necessary to develop a more appropriate similarity function, which scales better with changing activity and density. So far, we have no quality measure for the clustering of the community graph. We investigate whether we can also apply the modularity measure.

References

Aggarwal, C. C. and Yu, P. S. (2005). Online Analysis of Community Evolution in Data Streams. In: Proceedings of SIAM International Data Mining Conference (SDM 2005).

Baur, M. and Benkert, M. (2005). Network Comparison. In: Brandes, U. and Erlebach, T. (Eds.), Network Analysis, Springer, Berlin.

Cortes, C., Pregibon, D. and Volinsky, C. (2001). Communities of Interest. In: Proceedings of the 4th International Conference on Advances in Intelligent Data Analysis, 105-114.

Cross, R., Borgatti, S. P. and Parker, A. (2002). Making Invisible Work Visible: Using Social Network Analysis to Support Strategic Collaboration. California Management Review, 44(2), 25-46.

Dunne, J. A., Williams, R. J. and Martinez, N. D. (2002). Food-web structure and network theory: The role of connectance and size. PNAS, 99(20), 12917-12922.

Eales, R. T. J. (2003). Support Informal Communities of Practice within Organizations. In: Ackerman, M. S., Pipek, V. and Wulf, V. (eds.), Sharing Expertise: Beyond Knowledge Management, MIT.

Flake, G. W., Lawrence, S. and Giles, C. L. (2000). Efficient Identification of Web Communities. In: Sixth International Conference on Knowledge Discovery and Data Mining 2000, 150-160, ACM.

Girvan, M. and Newman, M. E. J. (2002). Community structure in social and biological networks. PNAS, 99(12), 7821-7826.

Gloor, P. A. and Zhao, Y. (2004). TeCFlow - A Temporal Communication Flow Visualizer for Social Networks Analysis. In: CSCW'04 Workshop on Social Networks, ACM.

Gongla, P. and Rizzuto, C. R. (2004). Where Did That Community Go? - Communities of Practice That "Disappear". In: Hildreth, P. and Kimble, C. (eds.), Knowledge Networks: Innovation through Communities of Practice, Idea Group Publishing, Hershey, PA, USA.

Hopcroft, J., Khan, O., Kulis, B. and Selman, B. (2004). Tracking evolving communities in large linked networks. PNAS, www.pnas.org/cgi/doi/10.1073/pnas.0307750100.

Jeong, H., Néda, Z. and Barabási, A.-L. (2003). Measuring preferential attachment in evolving networks. Europhysics Letters, 61(4), 567-572.

Kamada, T. and Kawai, S. (1989). An algorithm for drawing general undirected graphs. Information Processing Letters, 31(1), 7-15.

Kleinberg, J. and Lawrence, S. (2001). The Structure of the Web. Science, 294, 1849-1850.

Manville, B. (2004). Building Customer Communities of Practice for Business Value. In: Hildreth, P. and Kimble, C. (Eds.), Knowledge Networks: Innovation through Communities of Practice, Idea Group Publishing, Hershey, PA, USA.

Moody, J., Mc Farland, D. and Bender-deMoll, S. (2005). Dynamic Network Visualization. American Journal of Sociology, 110(4), 1206-1241.

Newman, M. E. J. (2001). The structure of scientific collaboration networks. PNAS, 98(2), 404-409.

Newman, M. E. J. (2004). Analysis of weighted networks. Physical Review, E 70.

Newman, M. E. J. and Girvan, M. (2004). Finding and evaluating community structure in networks. Physical Review, E 69(026113).

Preece, J. (2000). Online Communities, Wiley, Chichester, UK.

Radicchi, F., Castellano, C., Cecconi, F., Loreto, V. and Parisi, D. (2004). Defining and identifying communities in networks. PNAS, 101(9), 2658-2663.

Tyler, J. R., Wilkinson, D. M. and Huberman, B. A. (2003). Email as Spectroscopy: Automated Discovery of Community Structure within Organizations. In: Huysman, M., Wenger, E. and Wulf, V. (Eds.), Communities and Technologies, Kluwer Academic Publishers, Dordrecht.

Turner, T. C., Smith, M. A., Fisher, D., and Welser, H. T. (2005). Picturing Usenet: Mapping computer-mediated collective action. Journal of Computer-Mediated Communication, 10(4).

Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of "small-world" networks. Nature, 393, 440-442.

Wenger, E., McDermott, R. and Snyder, W. M. (2002). Cultivating Communities of Practice, Harvard Business School Press, Cambridge.

Wilkinson, D. M. and Huberman, B. A. (2004). A method for finding communities of related genes. PNAS, 10(1073).