# Cluster and Calendar based Visualization of Time Series Data

Jarke J. van Wijk
*Eindhoven University of Technology*
*Dept. of Mathematics and Computing Science*
*P.O. Box 513, 5600 MB Eindhoven*
*The Netherlands*
*vanwijk@win.tue.nl*

Edward R. van Selow
*Neth. Energy Research Foundation ECN*
*P.O. Box 1, 1755 ZG Petten*
*The Netherlands*
*vanselow@ecn.nl*

## Abstract

*A new method is presented to get insight into univariate time series data. The problem addressed here is how to identify patterns and trends on multiple time scales (days, weeks, seasons) simultaneously. The solution presented is to cluster similar daily data patterns, and to visualize the average patterns as graphs and the corresponding days on a calendar. This presentation provides a quick insight into both standard and exceptional patterns. Furthermore, it is well suited to interactive exploration. Two applications, numbers of employees present and energy consumption, are presented.*

## 1 Introduction

Time series data are ubiquitous. The aim of time series analysis is to obtain insight into phenomena, to discover repetitive patterns and trends, and to predict the future. We focus here on the analysis of univariate time series data. Suppose, we have collected energy consumption or air pollution data at short time intervals during one year, then how can we extract information from these data?

In the next section we discuss the problem and consider various solutions. Current methods fall short in the analysis of time series data at the various time scales, such as years, weeks, and days. Our new approach is based on a combination of two methods: The use of cluster analysis (section 3) and the visualization of the result on a calendar (section 4). Several applications are presented. In section 5 the strengths and limitations are discussed.

## 2 Background

Time series data consist of a sequence of $N$ pairs $(y_i, t_i), i = 1, \cdots, N$, where $y_i$ is the measured value of a quantity at time $t_i$. They are the simplest type of data to be analyzed, much simpler than for instance flow data, which consist of a mix of scalar and vector quantities at a multi-dimensional grid. Visualization is trivial: just draw a graph. So what's the problem?

The first is that $N$ can be very large. For instance, measurement at 10-minute intervals during a year yields 52,560 values. The second is that repetitive data patterns often have different time scales. For our applications we usually distinguish three time scales: seasons, weeks, and days. Human activities can vary strongly for these time scales, and hence also the related measured quantities. The third is that clear a priori hypotheses are rarely available. Hence, the user wants to have an overview first, subsequently he may want to zoom in on data and detect peculiar patterns or subsequences, and so on.

How can we analyze time series data? The first approach is to use mathematical models. A well-known method is the ARIMA model of Box and Jenkins [1]. This stochastic model can be used to predict future values, and for an expert, its coefficients give some insight into its time-dependent behavior. But in general, the multi-scale aspect is not addressed, and, the counterpart of the very high compression, details are lost.

Transformation from the time domain to a scale space directly addresses the multiple scales that are present in the data. Fourier transforms, Wavelet transforms, and fractal analysis [2] are conceivable approaches. They are most suited when the dominant frequencies or time scales are unknown. However, for the type of applications discussed here it is often known a priori that patterns will have a scale of days or weeks, hence such methods are too general. Furthermore, the result after transformation, defined over a frequency or scale-space domain, is much harder to interpret.

Another approach is to use the dependency on time scales explicitly, by considering the data as two-dimensional, for instance as $f(day, hour)$. The data can then be displayed as a so-called fingerprints. The days and hours are mapped on different axes, data is visualized via color [3]. In addi-

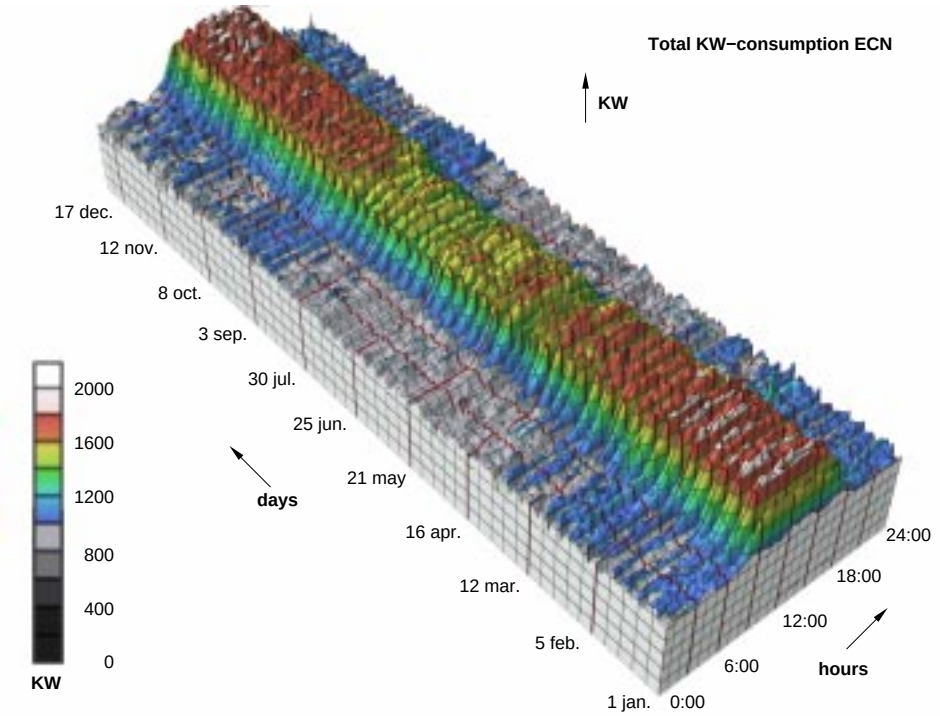


**Figure 1. Power demand by ECN, displayed as a function of hours and days**

tion to color, the third dimension can be used to display the data, yielding a mountain landscape. As an example, figure 1 shows the power demand data of a research facility (i.e. ECN). Such images show all data simultaneously. Seasonal trends can be discerned, as well as the typical day pattern. Yet, the variation over the week is harder to discern and the day-patterns of Saturdays and Sundays are obscured. Furthermore, in order to see the trends smoothing had to be used, but this eliminates fine details.

A simple way to get an overview is to average the data. For instance, temperature data over a year can be displayed as a graph of the average daily temperature, combined with a grey-shaded band to show the variation over the day [4]. However, if the data follow a weekly pattern, this technique is less useful, and any pattern within a day is not shown.

This can be overcome by showing multiple graphs. We could show the average day pattern for each month, for each day of the week, and so on. However, information is lost here too. As an example, many data patterns on holidays show strong similarities to data patterns on Sundays. If the data for each weekday is averaged separately, the holidays will disturb the results. To get more precise information and to eliminate cross-over of the various effects, we could make graphs for combinations of time scales: ranging from Sundays in January to Saturdays in December. As a result, the number of graphs to investigate becomes overwhelmingly large, and the difficulty arises how to combine graphs properly and how to extract information.

Let's make a step backward. What do we want? We want to elucidate which standard day patterns occur, and how they are distributed over the year and over the week. Furthermore, we want to detect days with patterns that strongly deviate from these standard patterns. If we use multiple graphs, as suggested before, it is implicitly assumed that there is a fixed relation between the distribution of patterns over the months and weekdays. In general, this assumption can not be tested a priori. An alternative is to drop this assumption, and let the analysis tool decide which daily patterns are similar and show their distribution over the year. This is the basis of our approach: cluster analysis, combined with a calendar based visualization.

## 3 Cluster analysis

Our aim is to merge similar day patterns into clusters, such that the day patterns within a cluster are more similar than the day patterns in other clusters. Each cluster contains an average day pattern. To this end, a simple and straightforward bottom-up clustering algorithm suffices [5]. We split the time series data into a sequence of $M$ day patterns. Each day pattern $Y_j$, $j = 1, \cdots, M$ consists of a sequence of pairs $(y_i, t_i)$, $i = 1, \cdots, N$, where $y_i$ denotes the measured value

and $t_i$ the time that has elapsed since midnight.

We start with $M$ clusters, each cluster containing one day pattern. Next, we compute the mutual differences between all clusters, and merge the two clusters which are most similar into a new cluster. As a result, we have $M-1$ active clusters. This step of merging small clusters into larger clusters is repeated until a single cluster results, which contains the average of all day patterns. To speed up the clustering procedure, the calculated differences between clusters are stored in a table, which only has to be updated for new clusters. The result of this algorithm is a binary tree of $2M-1$ clusters.

Various distance measures can be used. Suppose that we have two day patterns $y_i$ and $z_i, i = 1, \cdots, N$. A simple measure is the average geometric distance, or root-mean-square distance:

$$d_{rms} = \sqrt{\sum (y_i - z_i)^2}/N.$$

This measure is robust and usually yields good results. If we want to cluster patterns with similar shapes, a normalized version can be used:

$$d_{nm} = \sqrt{\sum (y_i/y_{\max} - z_i/z_{\max})^2}/N.$$

Here the measured values are normalized via division by the maximum value in the sequence. If we want to eliminate slow trends, we have to subtract the average difference. This means that we consider two patterns as equal if they are the same, except for an offset:

$$d_{sh} = \sqrt{\sum (y_i - z_i - \Delta)^2}/N,$$

with

$$\Delta = \sum (y_i - z_i)/N.$$

If we are interested in peak values, we can use:

$$d_{ma} = \mid y_{\max} - z_{\max} \mid .$$

We have experimented with several other measures, and found that the preceding measures gave the best results, and provide an easy-to-use toolset.

## 4 Cluster visualization

Now we have grouped the day patterns, how can we get insight into the result? A standard way to display the result of clustering is the use of dendrograms as is shown in figure 2. The bottom row shows the initial elements, each next row shows how two clusters are combined. This works fine if the number of elements is small. For more than, say, hundred clusters such images are much harder to grasp. Figure 3 shows a full clustering tree for 365 day patterns, which was generated by the well-known graph visualization package *dot* [6] without additional directives for the lay-out. Such an automatic tool yields the best result when the user does not supply lay-out directives that constrain the search for an optimal lay-out. Additional lay-out directives must be used to generate a dendrogram, with the days in their original order on the same row, and with each new cluster on a next row. This yields a highly cluttered image.

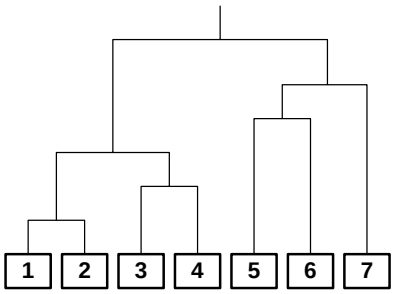


**Figure 2. Dendrogram**

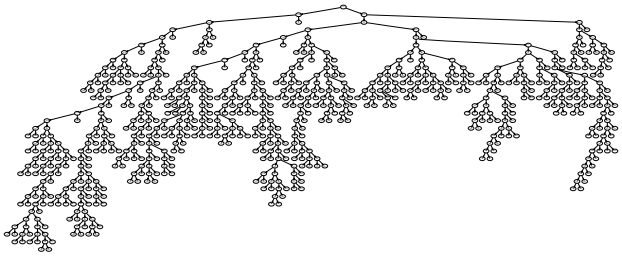

**Figure 3. Full clustering tree**

This can be improved if only a selection of all clusters is displayed, where the user can browse through and zoom in and out on the clustering tree, in the same style as with a file browser. What we still lack then is insight into the distribution of the elements of the cluster over the year. What is needed is a visual representation such that the viewer can effortlessly determine whether similarity is due to the season, the day of the week, or that some other correlation exists. Fortunately, such a representation already exists: a calendar.

### 4.1 Visualization

We have developed a combined representation of daily patterns and clusters. Patterns are shown as graphs, clusters are shown on a calendar. Colors indicate corresponding clusters and patterns. As an example, figure 4 shows a result of a cluster analysis of time series data on the number of employees present at ECN. The most significant seven clusters are shown. On the right, the average value per cluster is shown as a colored graph; on the left, each day in the calendar is colored according to the cluster to which it belongs.

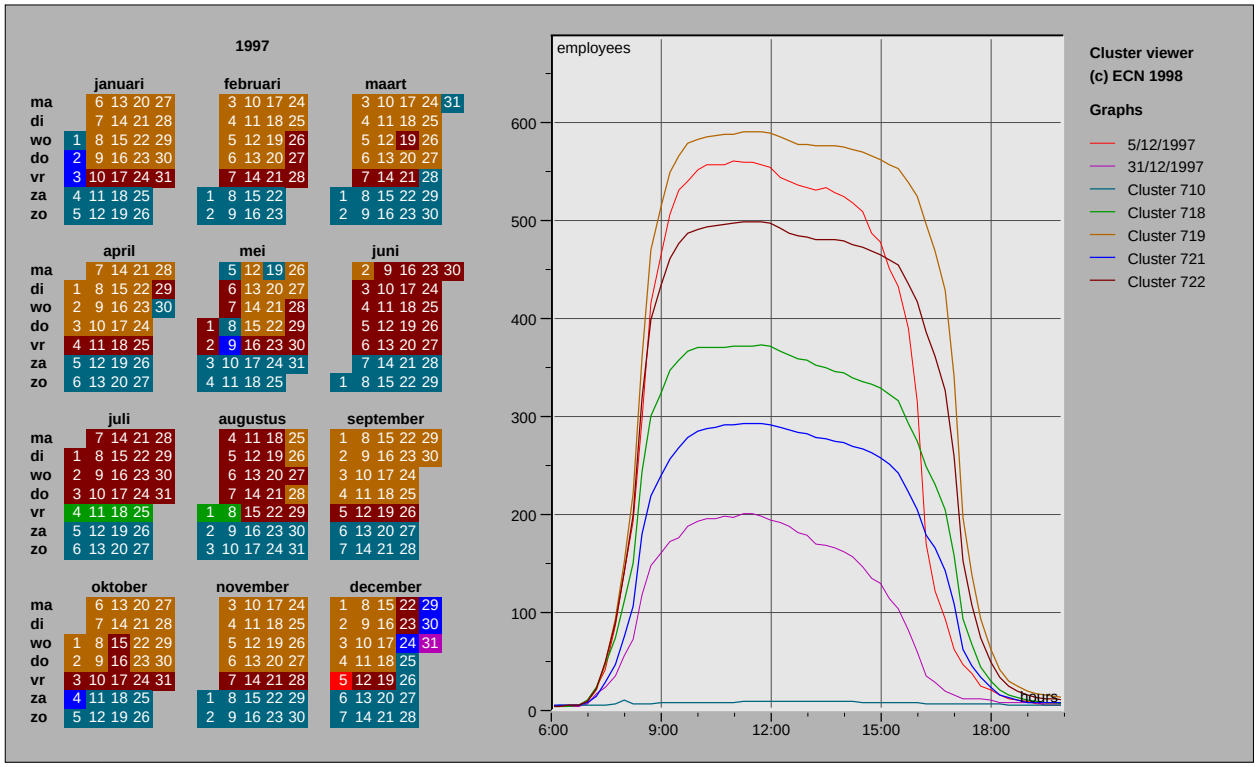


**Figure 4. Calendar view of the number of employees**

Several conclusions can be drawn from this image. We see that:

- Office hours are followed strictly. Most people arrive between 8:30 and 9:00 am, and leave between 4:00 and 5:00 pm. Furthermore, in the morning the number of employees present is slightly higher than in the afternoon.
- On Fridays and in the summer fewer people are present (cluster 722);
- On Fridays in the summer even fewer people are present (cluster 718);
- In the weekend and at holidays only very few people are working (cluster 710): security and fire brigade;
- Holidays in the Netherlands in 1997 were January 1st, March 28th, March 31st, April 30th, May 5th, May 8th, May 19th, December 25th and 26th.
- School vacations are visible in Spring (May 3rd to May 11th), in Autumn (October 11th to October 19th), and in Winter (December 21th to December 31st);
- Many people take a day off after a holiday (cluster 721);
- On December 5th many people left at 4:00 PM. Dutch people will immediately know the explanation: On this day we celebrate Santa Claus and are allowed to leave earlier!

We see that for this distribution of patterns quite plausible explanations exist. The advantage of clustering is that none of these explanations have to be inserted a priori, such as separating working days and holidays, and all effects are elucidated automatically. The combined representation of average graphs and clusters enables a user to quantify these effects easily. Another strong point is that standard patterns (cluster 719) as well as exceptional patterns (December 5th) are detected automatically.

## 4.2 Interaction

For effective data exploration, user interaction is as important as presentation. The combination of cluster analysis with a calendar representation provides good opportunities for interaction. We have embedded our presentation in an interactive system for the analysis of time series data, such that the user can interact with the image presented to him (such as fig. 4) in many ways.

Selection of the data to be displayed can be done easily. Initially, no days are selected for display. The user can tog-

gle days for display via point-and-click on a single day, on the label of a month, or on the label of the year. All days are then displayed as separate graphs. The user can point-and-click on a graph, upon which the corresponding day on the calendar is highlighted. Exceptional patterns are thus easy to locate.

When the user selects a day, a typical next question is which other days have a similar pattern. This is where the cluster analysis comes in. The user can select a day, and ask for more similar days via a single button press. The system determines the parent cluster, shows the average graph of this cluster and highlights the days within the cluster via color. This step can be repeated and reversed, so that the user can interactively enlarge and shrink the cluster to be displayed. Also, the user can select other days, and inspect several clusters simultaneously.

In addition to this bottom-up approach, the user can show clusters top-down. The user can select the number of clusters to be displayed, upon which the system generates a partioning of the year as shown in figure 4. Via two *more/less* buttons, the user can add and remove clusters, until a meaningful decomposition is made.

The full clustering process itself is done initially and later on request of the user. Our non-optimized version takes about 5 seconds on a PC with a Pentium 100 MHz processor, which is quite acceptable for interactive use. The clustering tree is stored and re-used upon each query. Reclustering has to be done if the user wants to use a different distance measure. As an additional option, the user can reduce or enlarge the time interval upon which the comparison has to be made. For instance, if he finds in a graph a strange peak occurring between 9:00 am and 10:00 am, he can select this interval graphically, and ask for a reclustering using only this time interval and the $d_{sh}$ measure. As a result, all days with a similar peak in this time interval are clustered.

Many standard options are further provided for the display of the graphs. The user can zoom-in and out, show the standard deviation for a cluster, and show each member of the cluster individually. Smoothing, with different filters and a user-controllable width, can be applied, which is useful if noisy data have to be processed. Clusters can be generated from these smoothed data. Some straightforward additional options could be fit easily within our framework. The use of the following distance measure:

$$\begin{aligned} d_{mn} = \quad & 5000 \mid y_{mon}/6 - z_{mon}/6 \mid + \\ & 1200 \mid y_{mon}/3 - z_{mon}/3 \mid + \\ & 400 \mid y_{mon}/3 - z_{mon}/3 \mid + \\ & \mid y_{day} - z_{day} \mid; \end{aligned}$$

where $y_{mon}$ and $y_{day}$ are the number of the month and the day respectively, gives a balanced clustering of the year in half-years, quarters of a year, months, etcetera. This enables a user to view standard averages and slow trends, using the same methods for interaction as with the content based clustering. With a slightly modified measure, also a separation into weekdays can be made.

Also, a simplified clustering method was implemented: Starting at a selected day, all other are added one after each other in order of their distance to an initial day. This option is useful to determine stepwise whether certain patterns are exceptions or not, again using the same methods for interaction.

### 4.3 Application

The background of our interest in time series data is the liberalization of the energy markets. In the Netherlands, customers with very high energy consumptions are recently allowed to choose their gas and electricity supplier and negotiate a tailor-made tariff. Other customers will follow in the next few years. This will strongly enhance competition between the energy distribution companies, which will have to transform themselves from utility companies into market-oriented companies. Insight into consumption patterns is essential for the segmentation of their customer markets. But also customers themselves need insight into their energy consumption patterns in order to lower consumptions, avoid peak rates and to negotiate a lower tariff.

Our aim is to develop methods, techniques, and tools that enable customers to analyze their energy consumption patterns easily and effectively. We started with a study of the electricity consumption at ECN itself. After collection of data several analysis and visualization methods were tried. The time series analysis data tool described proved to be very helpful.

Figure 5 shows a cluster analysis of the power consumption. The five main clusters are shown here. During week-ends power consumption was fairly constant. Furthermore, four clusters with about the same patterns but different plateau levels emerge. The correlation with the seasons is clearly visible. Finally, in the morning of February 4th a high peak demand occurred.

## 5 Discussion

We have presented a new method for the exploration and analysis of extensive time series data. The combination of cluster analysis and calendar based visualization turned out to be highly effective. Almost effortlessly images such as figure 4 and 5 can be generated, which provide a good insight.

The next step to be made is the extension to the interactive visualization and analysis of several variables simultaneously. This enables a user to study correlations between variables, either manually or automatically. Detected correlations can lead the user in the direction of a suitable

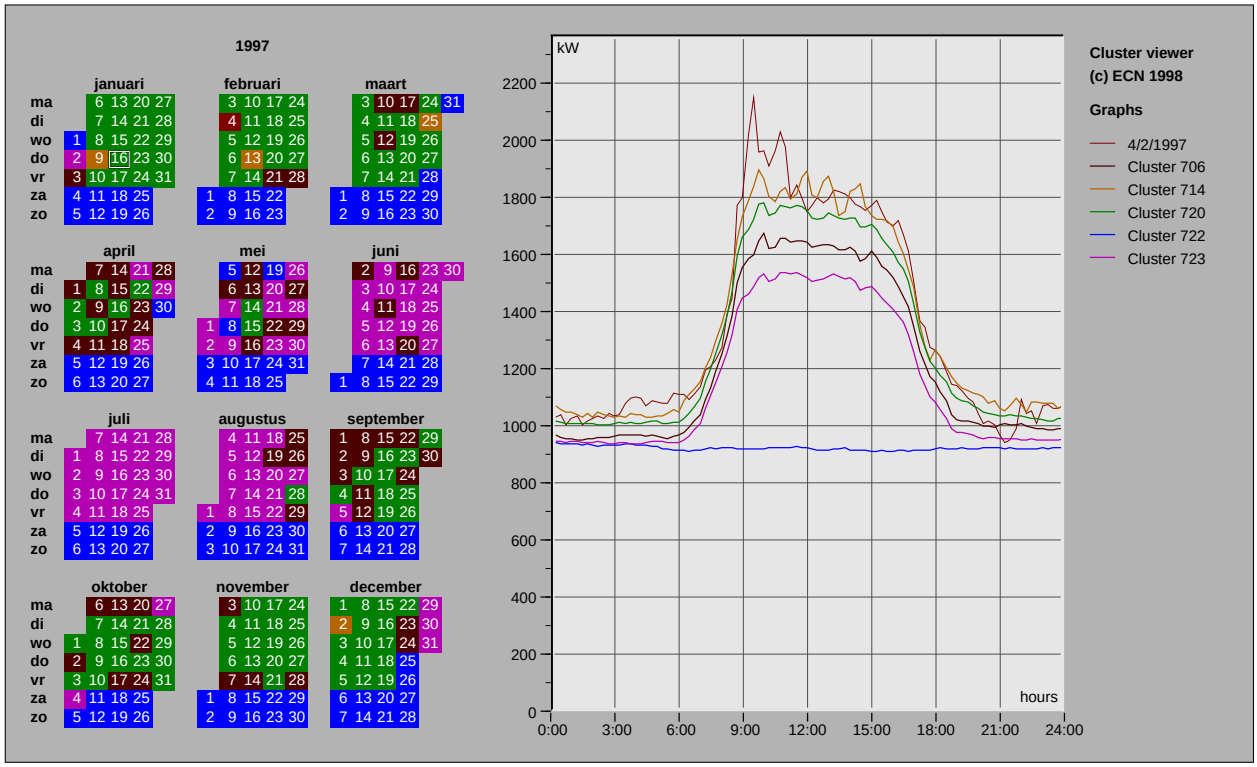


**Figure 5. Cluster analysis of power demand by ECN**

model. Model parameters can subsequently be estimated by a regression method, and a statistical analysis of the model residuals will indicate the validity of the model. Adopting this procedure in the study of ECN energy consumption, a linear model was identified which could accurately predict the power consumption from the sunlight intensity and the number of employees [7]. We used different packages for this, integration of such methods in a single tool would be highly effective.

In conclusion, we think that our cluster and calendar based analysis is a useful method to explore and visualize large quantities of univariate time series data, and provides a sound basis for a general analysis tool.

## References


[1] Box, G.E.P. and Jenkins, G.M. *Time Series Analysis: Forecasting and Control*, 2nd edition, Holden-Day, 1976.

[2] Evertsz, C.J.G. Fractal Geometry of Financial Time Series. *Fractals* **3** (3), pp. 609-616, 1995.

[3] Keller, P.R. and Keller, M.M. *Visual Cues*, IEEE Press, Piscataway, NJ, USA, 1993, p. 53.

[4] Tufte, E.R. *The Visual Display of Quantitative Information*, Graphics Press, 1983.

[5] Kaufman, L. and Rousseeuw, P.J. *Finding Groups in Data: An Introduction to Cluster Analysis*, John Wiley, 1990.

[6] Gansner, E.R., E. Koutsofios, S, North, and K-P. Vo. A Technique for Drawing Directed Graphs. *IEEE Transactions on Software Engineering* **19** (3), pp. 214-230, 1993.

[7] Selow, E.R. van, Wijk, J.J. van, Jehee, J.N.T. Identification and Visualization of Energy Consumption Patterns. In: *Proceedings of DistribuTECH DA/DSM Europe*, Pennwell, London, October 1998.