

# Interactive Visual Analysis of Networked Systems: Workflows for Two Industrial Domains

*Fredrik Holmgren and Sverker Janson*

*Swedish Institute of Computer Science, Box 1263, SE-164 29 Kista, Sweden*

*fredrikh@sics.se, sverker@sics.se*

SICS Technical Report T2008:12, ISSN 1100-3154

## 1 Introduction

For each new generation, networked systems grow larger, receive more users, and have more dynamic services and configurations. Complexity is added both to the individual system states and to their evolution over time, making these systems increasingly difficult to understand, and costly to develop and manage.

By using Information Visualization methods to present complex information visually [4, 11, 16], we can significantly increase our ability to understand networked systems. Visualization tools could aid (1) developers in modelling, developing, debugging, and maintaining systems, (2) managers in conceptualizing proposed systems and understanding systems under development and in operation, (3) operators in configuring, optimizing, and recovering systems, (4) educators in presenting systems to students and practitioners, and (5) researchers in developing new algorithms and system architectures. Today, this potential remains largely untapped in the world of networked systems.

A main obstacle is the lack of visualization frameworks that support cost-efficient re-use and composition of components into powerful and flexible visualization tools suited for networked applications. A visualization framework supporting full system life-cycles must meet additional requirements on architecture structure and flexibility.

We report here on a first study of interactive visual analysis of networked systems. Working with ABB Corporate Research and Ericsson Research, we have created workflows which demonstrate the potential of visualization in the domains of industrial automation and telecommunications. By a workflow in this context, we mean a sequence of visualizations and the actions for generating them. Visualizations can be any images that represent properties of the data sets analyzed, and actions typically either change the selection of data visualized or change the visualization by choice of technique or change of parameters.

Workflows serve two purposes in this work. The first is to investigate and present the various techniques and actions needed for the specific problem scenarios. The second is to illustrate a set of visualisation techniques in a common context.

The workflows presented exploit a range of information visualization techniques: basic tables and charts, tree maps, node-link graphs, semantic substrates, and others. For basic tables and charts we have used a commercial tool, Tableau, which offers excellent support for these techniques and for integration with data sources. For other techniques, that better reflect the graphs and relations that occur in networked systems data sets, we have used research prototype software, and where no obvious suitable software exists we have simulated workflows using a drawing tool.

Through this work, we hope to incite users, tool providers, and other researchers to explore the considerable untapped potential of visual analysis in engineering domains such as networked systems, and encourage the development of more powerful visual analysis tools targeting these domains.

The remainder of this report is organized as follows. In Section 2, we present the challenge of Networked Systems and introduce Information Visualisation and concepts to be used in the sections that follow. Workflows for the industrial automation and telecommunications management domains are presented in Sections 3 and 4. Finally, in Section 5, we discuss of the relation between the visualisation workflows, the tools used, and our vision of visualisation as an integral part of the lifecycle of networked systems.

## **2 Background and Motivation**

In this section we discuss the challenges met by developers and managers of networked systems, introduce techniques from the Information Visualisation field that offer the hope and promise of significant impact on these problems, and discuss shortcomings of this body of work that limit industrial uptake and require further research.

### **2.1 Challenges of Networked Systems**

For each new generation, networked systems grow larger, receive more users, and have more dynamic services and configurations. Complexity is added both to the individual system states and to their evolution over time, making these systems increasingly difficult to understand, and costly to develop and manage.

Developers and network managers must deal with huge amounts of information, in a domain where even small data sets can be complex and difficult to interpret correctly. The problem is exacerbated when new events occur concurrently over time and extend or modify the data sets dynamically. The collected information may be incomplete when it comes to some particular detail that we want to study. We must be able to identify the missing information and collect it by interacting with the networked system, often by controlling the system towards the particular state that we want to study.

Tools that address these challenges could significantly lower the costs and increase the quality of development and management of networked systems.

### **2.2 Example Networked Systems**

Two networked system domains of interest are management of industrial control systems and management of telecommunication systems and services.

ABB offers the 800xA system [1], which includes all industrial automation functions in a single integrated operating and engineering environment, potentially encompassing thousands of units, with multiple underlying applications and protocols. In this environment, integration and engineering efficiency are major concerns.

In cooperation with ABB, we have explored aspects of visualization for new Ethernet-based networks for the 800xA system. It has been noted that it is important that support integrates various system aspects. Visualisation should not be limited to computer systems aspects; e.g., power supply availability and the location of devices in a process also need to be considered.

Ericsson explores the future of telecommunication networks and services, networked systems on a very large, even global, scale. With increasingly self-managing components, network and service managers will be able to focus on offering efficient business support.

With Ericsson, our visualization scenario is based around Service Introduction Business Support. A service manager may wish to explore the impact of services on resource consumption in the physical network, and how different services, or groups of users, interact with each other in their network use. Although this aims at visualisation at a high abstraction of the actual underlying network, it is considered important to still have the ability to "drill down" to the details of the underlying network. It is important that the visualisation approach is designed with this in mind, although this aspect has not been fully explored within the scope of this work.

At an abstract level, these are network management problem scenarios with significant synergies in the exploration of supporting tools.

### **2.3 Information Visualisation**

It has been observed that as the capacity of our computers increases, the volume of data that we produce increases even faster.

Visualisation is a process of transforming data into a visual form, using computer based tools and techniques [4]. Given a visualisation, our human visual system can efficiently perceive and process the information present in a data set. Visualisation helps us form a mental model of what the data represents and means [24]. Visualisation employs methods and techniques from many fields including interactive graphics, imaging, and visual design.

The field of Visualisation can be divided into Scientific Visualisation and Information Visualisation. In Scientific Visualisation, the underlying data is often spatial in nature and can be given natural 2D or 3D renderings, such as CT or MRI scans in medicine. In Information Visualisation, the field relevant for this project, the underlying data sets are more abstract, representing arbitrary entities and their relationships, which often lack spatial interpretations.

The Information Visualisation field has been defined as the use of computer-supported, interactive, visual representations of abstract data to amplify our cognition, the acquisition and use of knowledge [4]. In general, Information Visualisation can help us (1) increase resources, (2) reduce search, (3) enhance recognition of patterns, (4) enable perceptual inference, (5) enable perceptual monitoring and (6) enable manipulation and exploration of the visualisation.

For visualisation of networked systems, this could imply the following:

1. Information visualisation can extend our resources by visualising networked systems and their properties for which we are unable to form a mental picture.
2. Organizing a networked system and/or its properties according to a visual layout or structure can greatly reduce search compared to performing search in a textual representation.
3. Depending on the layout, particular features of networked systems may easily be recognized.
4. Repeated patterns or some combination of patterns can be associated with particular networked system states or topologies.
5. Visualised patterns change and recombine as the underlying "real" networked systems evolve.
6. To make full use of visualisation one should be able to interact both with the items of the visualised networked systems, their layouts and rendering, and with the underlying systems.

## 2.4 Tables and Charts

One of the oldest visual representations of data is tables. A table organizes data, but offers mainly support for lookup. For example, in general, extreme values have to be found by exhaustive search. But with added interaction we can obtain alternate views of our data that help us analyse additional aspects. A simple example of interaction is Excel's ability to construct pivot tables. More powerful visual analysis support is offered by Tableau, a table-oriented business intelligence tool [8, 15, 26]. Tableau tables have been extended with diagrams and charts for interactive exploration of trends, patterns, extremes and outliers.

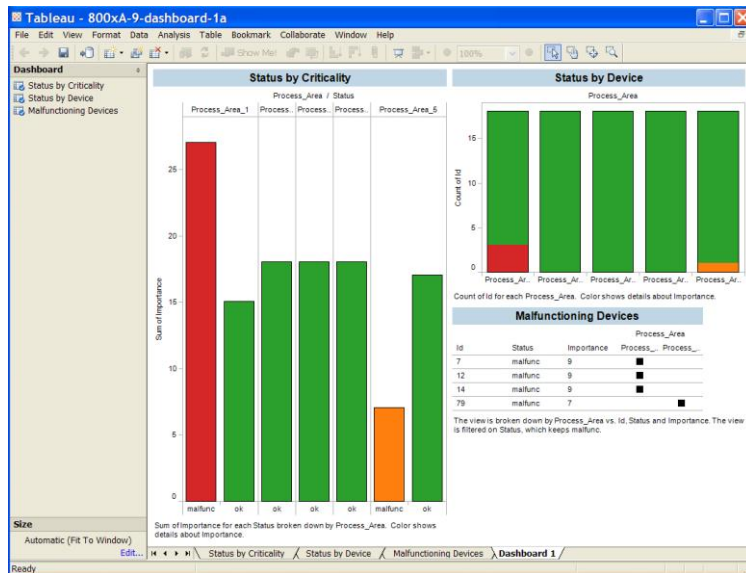
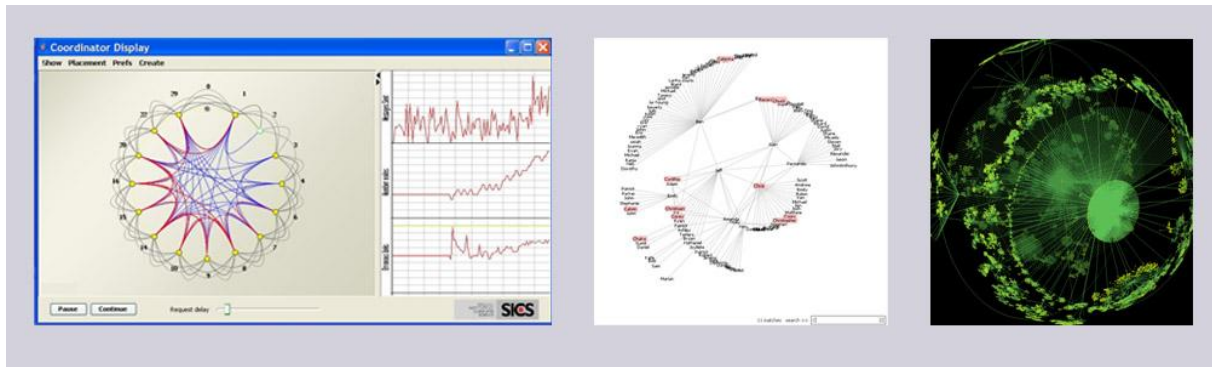


Figure 1. A Tableau dashboard constructed from tabular data

Figure 1 shows a dashboard for monitoring the status of an ABB 800xA industrial automation system. The dashboard is constructed in Tableau and shows three views of the status of the automation system: two bar charts illustrating the individual status of five process areas and one table listing malfunctioning devices. The first bar chart shows status by criticality or the ability of the system (or process area) to perform as intended. The second bar chart shows the proportions of functioning to malfunctioning devices for each process area.

## 2.5 Node-Link Graphs

A visual encoding that immediately comes to mind in the domain of networked systems is that of a node-link graph. Graphs of this kind can represent system nodes and links of communication between network nodes as well as other entities and relations, such as services, users, modules, and processes and their dependencies.



**Figure 2. Three node-link graph visualisations**

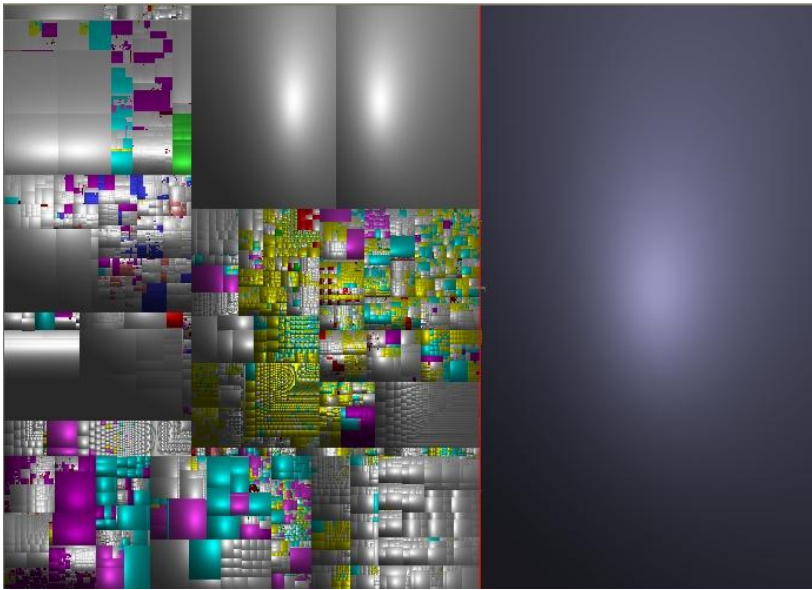
Figure 2 shows three examples of node-link graph visualisations. The left image shows a visualisation system developed as a part of a SICS Peer-to-Peer research project. The circle represents the identifier ring in a DHT overlay network, with nodes and corresponding links colour-coded by link type. The middle image shows a radial-tree layout generated by the Prefuse [9] system. The radial tree spans a social network of people and their relations. The graph highlights users matching a search query. The right image shows a hyperbolic view of a network with a very large number of nodes [3]. While the details of individual nodes are invisible, the link topology is quite clear.

Beyond these examples, the information visualisation field offers a wide range of algorithms and techniques for visualisation of node-link graphs [4, 11]. Techniques to avoid the limitations of visualisation of networks have been devised and examined [13, 16], along with applications that effectively prove both the usefulness of the algorithms and techniques and also shows the merits of visualisation in general.

## 2.6 Tree Maps

Tree structured node-link diagrams can grow too large to be practically useful. Treemaps are space-efficient layouts of trees, where nodes are drawn as space-filling nested rectangles [22]. The size and colour of a rectangle can be set to represent properties of leaf nodes.

The treemap is created by using a tiling algorithm that divides and transforms a rectangle into sub-rectangles of specified areas. Properties such as data ordering, aspect ratio and stability of the layout to changes of the underlying data are all affected by the choice of algorithm.

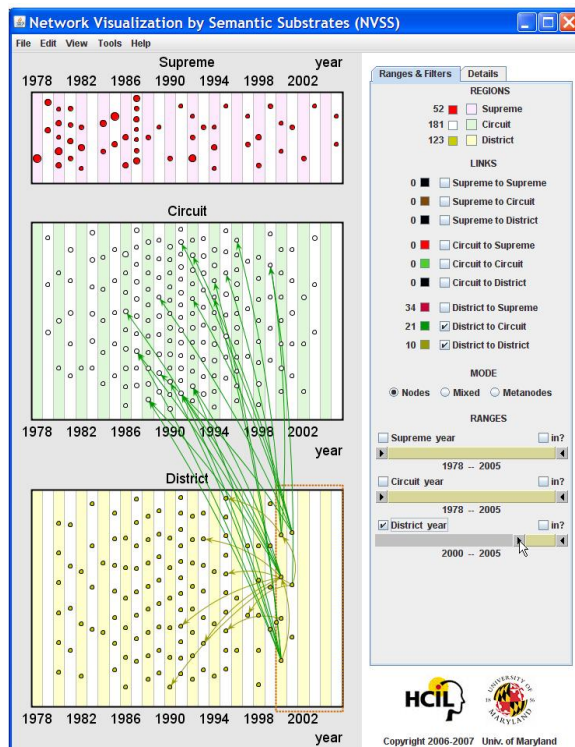


**Figure 3. SequoiaView**

A well-known example of the use of treemaps is to show the allocation of disk space, to help the user to find and identify files that are no longer needed. Figure 3 shows such a treemap generated by the tool SequoiaView [21]. The size of a rectangle indicates the size of the file or directory. The colour represents the type of a file. The large rectangle to the right corresponds to free disk space. Placing the cursor over a rectangle displays the file name. SequoiaView adds ridges to each rectangle during layout construction, to create a cushion-like appearance of individual directories and files, which makes them easier to identify.

## **2.7 Semantic Substrates**

To produce readable and aesthetically pleasing results most graph layout algorithms do not assign meaning to the placement of nodes. In Semantic Substrates [2], the placement of nodes is restricted to rectangular regions (substrates) according to some attributes chosen by the user. Thus semantic substrates can be regarded as user defined spatial templates for node-link graphs. By allowing the users to design their own substrates, semantic substrates offer highly flexible selection and filtering.

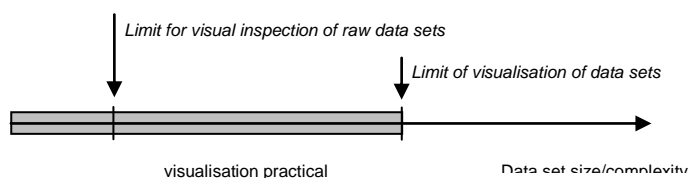


**Figure 4. Semantic Substrate**

The semantic substrate in Figure 4 consists of three regions set up by the user, illustrating the hierarchical court system of the United States in order of power [2]. A node within a region represents a legal court case. Links represents legal citations from one court case to another. By using the control panel to the right, the user may filter the links by the type of citation and the time period of citing or cited case.

## 2.8 Limitations of Visualisation

Information Visualisation has natural limitations. Consider Figure 5 below. At some size and complexity of data sets, inspection of raw data is feasible. At the other end, sufficiently large visualisations will either overload our perceptual system or run out of visualisation space (“no more pixels”). Visualisation may also fail due to lack of computational resources. Either way, visualisation fails at some data size or level of complexity.



**Figure 5. Intervals of practical visualization**

Much of the research aims to extend the interval where visualisation is practical or to develop techniques that will limit or project the visualised information to the practical interval. In interactive visual analysis, large data sets can also be handled by interactively zooming and filtering to focus on a smaller, and more relevant, subset of the data.

## 2.9 Visualisation workflows

When we speak of visualisation workflows, we will refer to sequences of visualisations and views and the interactions that trigger the transitions between views. Visualizations can be any images that represent properties of the data sets analyzed, and actions typically either change the selection of data sets visualized or change the visualization by choice of technique or change of parameters.

The Visualisation Mantra of Shneiderman, “Overview, zoom & filter, details-on-demand, overview, zoom & filter, details-on-demand ...”, describes the general task of a user exploring a data set [23]. The workflows reported here can be regarded as an instances or realisations of the Visualisation Mantra.

Visualisation workflows serve two purposes in this work. The first is to investigate and present the various techniques and actions that can offer support in the given problem scenarios. The second is to illustrate a set of visualisation techniques in a common context.

## 2.10 Visualisation Software

Besides the systems already mentioned, research in Information Visualisation has produced a wide range of software frameworks, toolboxes and libraries, such as JUNG [19], Prefuse [9], Infovis Toolkit [6], IVC [13], VTK [28], and GraphViz [7], to name a few.

The process of visualisation is often presented as a pipe-line of processes from source data to graphical items, the Visualisation Pipeline [12]. A further elaboration of the Visualisation Pipeline is the Visualisation Reference Model [4], which separates data and transformations. This model also takes human interaction into account. Further developments include refinements of human interaction [14] and design patterns for building blocks to facilitate reuse between applications [10]. Current frameworks and toolboxes implement the Visualisation Pipeline, and to various degrees also the Reference Model, in different ways.

Unfortunately, state-of-the-art Information Visualisation software tends not to be used to any large extent by practitioners in fields outside visualisation itself [14, 29]. So far, this has also been true in our own research into peer-to-peer and networked systems.

One part of the explanation is the gap between frameworks and applications. Frameworks aim to support efficient engineering of systems by offering reusable high quality implementations. Applications, developed by visualisation researchers, are often showcases of highly specialised algorithms and techniques. In their quest for generality, the frameworks often fall short of the requirements of specific applications. The implementations of algorithms and techniques in showcase applications may be tightly coupled with the application logic.

This gap makes integration and reuse of existing results difficult and costly, which in turn leads developers to resort to ad-hoc implementations, or non-reusable specialised solutions, re-inventing the wheel in the process.

## 2.11 Summary of Challenge and Opportunity

Networked systems form the foundation of major industrial products. Increasing size and complexity makes these systems difficult to understand, develop, and manage. The Information Visualisation field offers a wide range of algorithms and techniques to aid human understanding of the complex data sets of these networked systems.

A main obstacle is the lack of visualisation frameworks that support cost-efficient re-use and composition of components into powerful and flexible visualization tools suited for



networked systems applications. A visualization framework supporting full system life-cycles must meet additional requirements on architecture structure and flexibility.

## **2.12 Vision: Interactive Visual Analysis of Networked Systems**

We conclude the background and motivation section with our own vision for visual analysis of networked systems:

*Information Visualisation should serve an integrated role in all nontrivial networked systems, supporting all phases of its life cycle: modelling, prototyping, development, debugging, operations, and maintenance.*

Information Visualization should aid (1) developers in modelling, developing, debugging, and maintaining systems, (2) managers in conceptualizing proposed systems and understanding systems under development and in operation, (3) operators in analyzing, configuring, optimizing, and recovering systems, (4) educators in presenting systems to students and practitioners, and (5) researchers in developing new algorithms and system architectures.

The same visualisation support should be used throughout the lifecycle, easily adaptable to the evolution of the system and to changing requirements. Visualisation should not be a last minute add-on to a newly developed system. Visualisation can provide developers, technicians, and researchers with valuable information on a networked system from its initial design, to final operations, tuning, and maintenance.

## **3 Industrial automation use case and workflows**

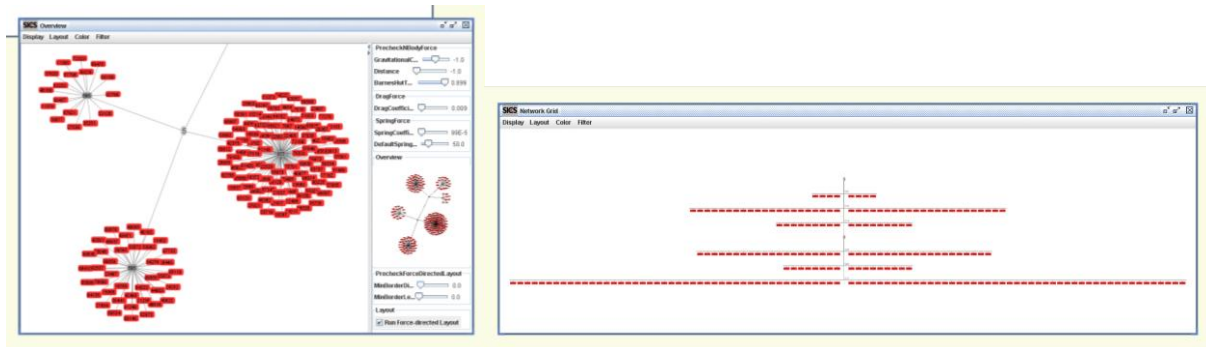
In this section we present three examples of workflows associated with an industrial automation system, the role of which is to control and support the management of industrial machinery and processes.

The first workflow explores the topology of and traffic in the field network of the automation system. The second focuses on the traffic in the field network, the protocols used and the amounts of communication. The third monitors the status of devices, individually and aggregated by subsystem.

### **3.1 Workflow: Field network hotspot**

This workflow explores the topology of and traffic in an automation system field network, loosely corresponding to an Ethernet-based 800xA field network. We can assume as background that several nodes have reported communication problems and that an operator or field engineer is looking for possible root causes.

A typical field network consists of controllers, sensors, and actuators, here interconnected with Ethernet links. For visualization purposes, there is a straight-forward interpretation of components as nodes and Ethernet connections as links in a node-link graph. Figure 6(a) shows a simple network visualized as a node-link graph using a force directed (FD) layout. FD layouts can be applied to any node-link graph and are particularly useful when the structure of a complex network is not known beforehand.



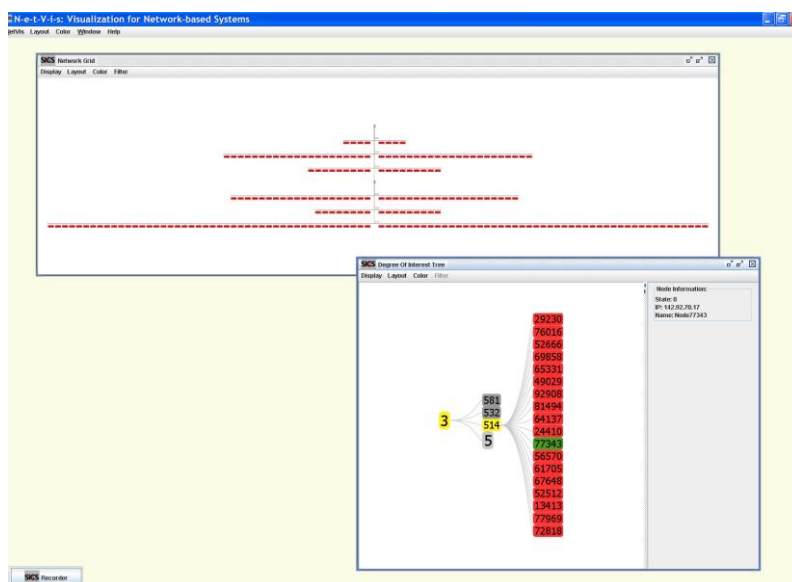
**Figure 6: (a) FD layout of field network, (b) traditional layout of field network**

In Figure 6(b) we see an alternative stylized view of the network, a simplified version of how a network of this kind is normally drawn on paper. The grey rectangles depict controllers and the red rectangles other devices, such as sensors and actuators. This is a view that automation network technicians and designers can relate to. There is not much detail apart from the tiny node identifiers. But it gives us an overview of the general structure of the network.

Both visualizations are generated using a platform designed and implemented around the Prefuse framework [18, 27]. Alternate layouts may be applied to the data in the current window by selecting an appropriate layout from the Layout menu of the floating window. A new window may be “cloned” from an existing window, applying a different layout.

For this application, the stylized view is a natural first overview of the network.

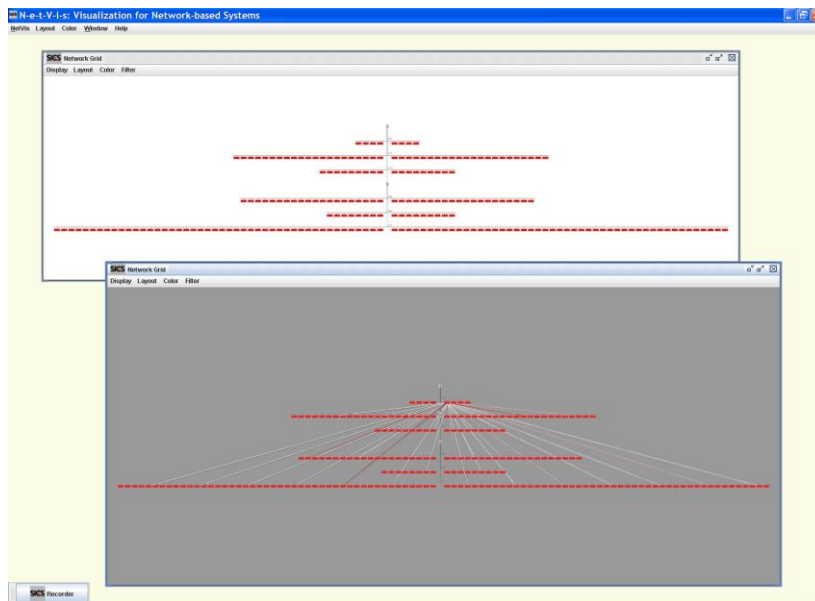
Figure 7 shows a new floating window “cloned” from the already existing window. We choose to inspect the field network as a regular tree applying a tree layout. Here, one branch of devices has been fully expanded. By double-clicking on a device it is selected and additional information is shown at the top of the right section of the new window.



**Figure 7. Complementary tree view and info**

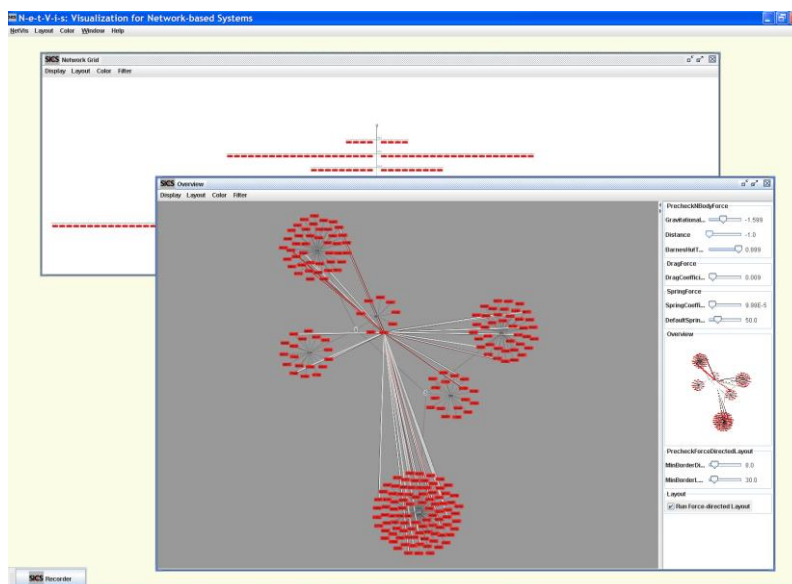
The visualisation platform is connected to a simulation of an 800xA field network, and subscribes to a set of network events. Each event represents a state or configuration change of the network. When a new event is received by the visualisation system, it updates its internal representation of the network and then any visual views of the network accordingly.

In the previous views, we have explored static aspects of the network, familiarizing ourselves with the network topology. Next, in Figure 8, we inspect the current traffic of the network. The traffic, subscribed communication events, may be interpreted and viewed as a second set of visually distinct links between sender and receiver nodes.



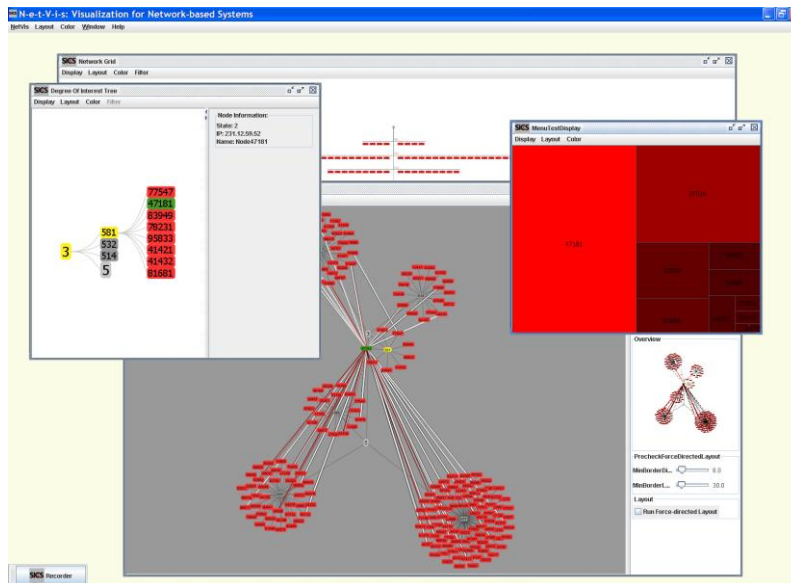
**Figure 8. Traffic patterns**

The background colour has been altered to highlight the traffic, shown as white and red arcs between network nodes. The red arcs illustrate bursts of high volume traffic during the time of the screenshot.



**Figure 9. Traffic patterns with alternate layout**

To get a clearer view, in Figure 9, we revert to the FD layout and see that it appears as a certain node is involved in most of the traffic. We then, in Figure 10, create two additional clones, applying a tree and a treemap layout respectively, to obtain more information about this node.



**Figure 10. Locating a hotspot and finding additional info**

We could have zoomed in on the FD layout for additional detail, but chose to inspect the network hierarchy as a treemap. In this treemap, the size of a node reflects the amount of traffic to and from the node. This gives us a clear picture of which nodes are generating or receiving what amounts of traffic, and a verification of which node is the most active, indicating a potential source of network problems.

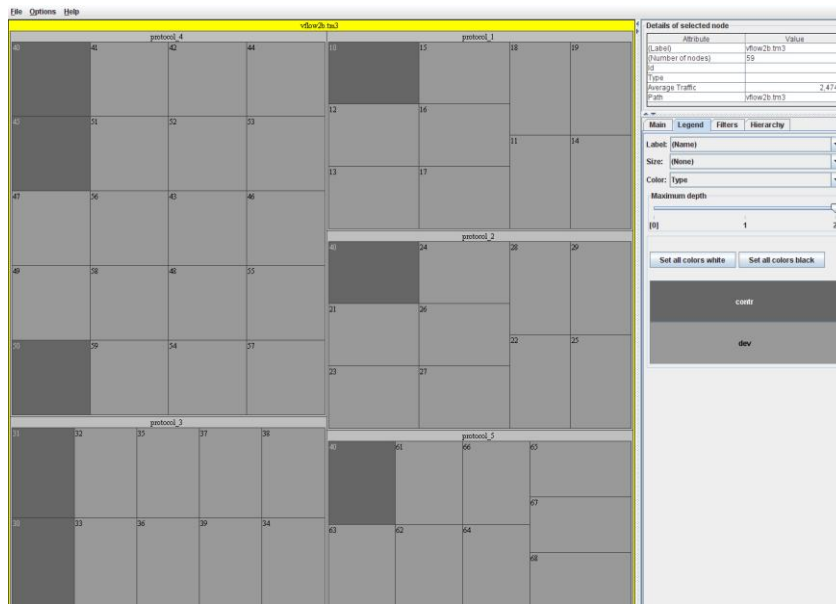
This series of visualisations makes up our first workflow, which was used to explore the root causes of communication problems reported. The workflow is constructed step-by-step by the user in a largely application independent fashion. In this workflow, the actions are mostly basic applications of different layout algorithms to obtain proper overviews followed by minor zooming, filtering and selection operations to get to the appropriate detail.

For this particular problem, finding a possible hotspot or congestion, this may not be an optimal workflow and a specialized tool may be desirable. Nonetheless, it illustrates well the vision of making complex and dynamic data sets accessible to human exploration, inspection, and understanding using a range of different visualisation algorithms and techniques.

### 3.2 Workflow: Protocol activity

Next, we present a workflow analysing traffic and the devices associated with a certain protocol. This extends the previous workflow by focusing further on questions such as ‘Which nodes are communicating?’ and ‘How are they communicating?’

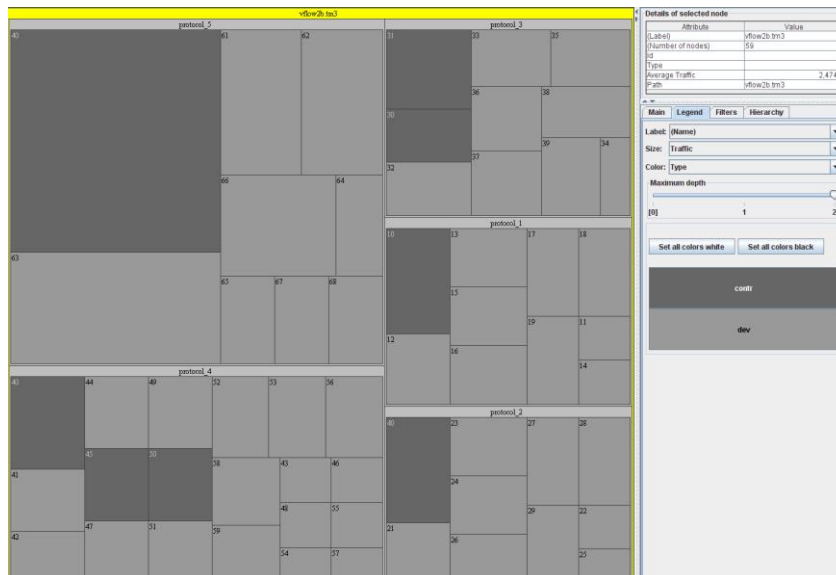
Our starting point, Figure 11, is a treemap overview of the set of protocols used in the network. The underlying data set represents the traffic for a given time interval and is represented as set of records. Each record states how much traffic a node handles during the interval given a specific protocol. There is one record for each protocol that a node supports. The treemap software used can form trees dynamically by branching on the values of properties. Here, we branch on the protocols supported by nodes. If a node supports several protocols, it will appear in several branches. In the treemap, each protocol forms a rectangular area divided into smaller rectangles that correspond to devices supporting this protocol, which are the leaves in the tree. The colour of leaves is set to reflect the type of device, with controllers set to a darker shade. Initially, the leaves are of equal size.



**Figure 11. Treemap of the protocols supported by the devices in a network**

In this first overview, some observations are immediate, e.g., that ‘protocol 4’ is supported by the most devices and that there is at least one controller involved in each protocol.

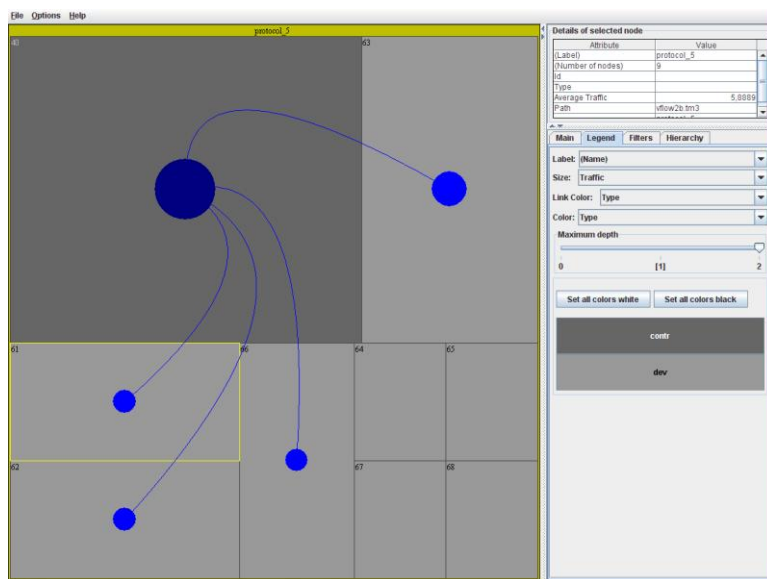
Each node has a traffic property recording the amounts of traffic per protocol in the given time-slice. In Figure 12, the size of leaves is set to be proportional to traffic. The areas corresponding to protocols will reflect the total amounts of traffic per protocol. Devices with no traffic will not be visible.



**Figure 12. Treemap of used protocols during a specific time-slice**

In Figure 13, we explore more closely the network formed by components communicating using protocol 5. We superimpose a set of links showing inter-node traffic. The software used does not support such links, which have been added using a drawing tool. We assume that we filter only traffic above a certain threshold so that we see the heaviest traffic only.

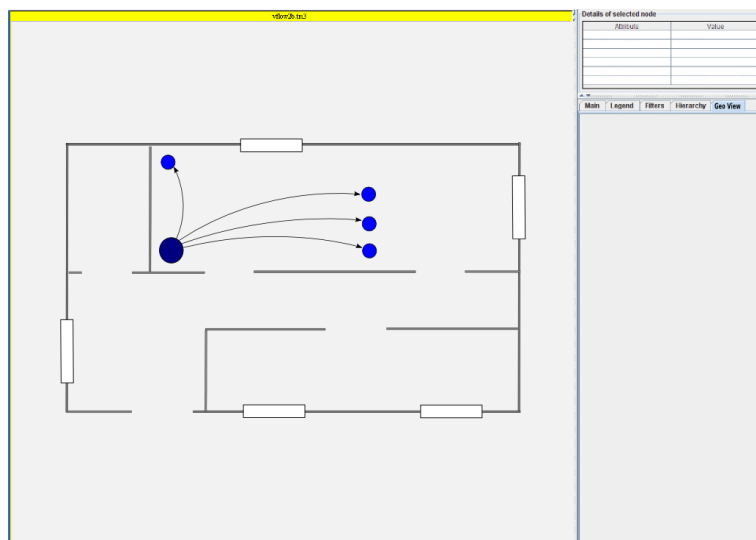
We see that five nodes are involved and that these are responsible for the anomalous traffic pattern in the network. We can now select each node and study its properties now listed in the “Details about the selected node” part of the control panel.



**Figure 13. For a certain protocol, which devices are communicating**

As the final step in the visual part of the workflow, we determine the locations of the components. We switch to a node-link graph view with a geographical layout, which uses available coordinate properties of the components, otherwise keeping the selection settings in the control panel.

While this change of view is not supported by the software used, Figure 14 illustrates one such possible geographical view based on the floor plan of a building. Having determined the locations, a next step could be to invoke other appropriate management tools to further analyze the problem.



**Figure 14. Geographical location of devices**

In the previous workflow we looked for anomalies in general. If we had known beforehand what kind of problem to look for, we could have chosen the treemap as our starting point.

Treemaps are quite powerful, illustrating hierarchy by layout and properties by relative size and colour. For this workflow we used treemap demo software from the Human-Computer Interaction Lab of the University of the Maryland [20], complemented by drawings. As required by this software, the input was a static file describing the properties of the network and the traffic during a particular time-slice.

### 3.3 Workflow: Status supervision

In this workflow, we explore a small 800xA plant and the status of its devices. The 800xA automation system organizes components into several different hierarchies, which can all be used as a basis for treemap views. We here use the location structure.

The plant is subdivided into five process areas. In Figure 15, a first treemap view, we display the aggregated status of each process area using a colour coding. Aggregation is done by taking the maximum of the problem severity levels. We assign no meaning to the area of the treemap and all current leaves should be of equal size.

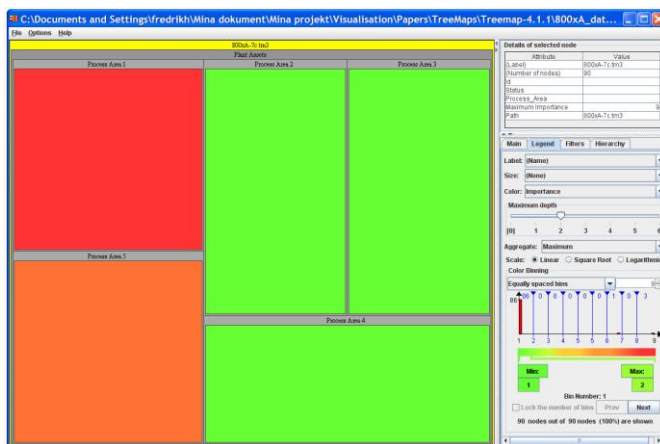


Figure 15. Aggregated view

Shades of red indicate that there is a malfunctioning device in the process area and green that everything is functioning as expected. When a process area experiences a problem, we need to determine what devices are involved. This is easily done, in Figures 16 and 17, by increasing the maximum depth of the treemap.

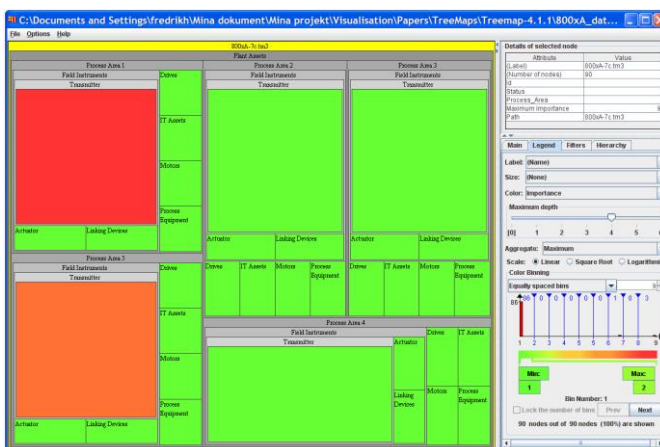


Figure 16. Hierarchy drill-down





Figure 17. Further drill-down into fully expanded view

In this workflow the analysis is less exploratory. The intended user is an operator monitoring and managing a plant. A quick overview is given by the initial treemap where visual clutter is avoided by showing only the aggregated state. The following steps illustrate the operator drilling into the underlying data to get further details on the failing devices.

Although in some sense trivial, this workflow was immediately useful in the 800xA context.

Today the supervision of an 800xA system involves many specialized and sometimes overlapping tools. A project at ABB explored the possibility of unifying these tools into a single framework for easier and more efficient use [25]. Based on the status supervision workflow, a prototype was developed and integrated into the 800xA environment. The unifying element is a treemap visualisation. The prototype was implemented in the Prefuse library and the colour states coded according to the industrial standard NES107 [17].

Figure 18 shows the treemap in the 800xA environment. The treemap shows the overall status of the system. By clicking on any of the nodes a list of appropriate specialized tools is to be shown. By choosing a tool from the list the corresponding tool is started. To cope with large numbers of devices, devices are partitioned into classes of criticality, depending on their overall importance for the system.

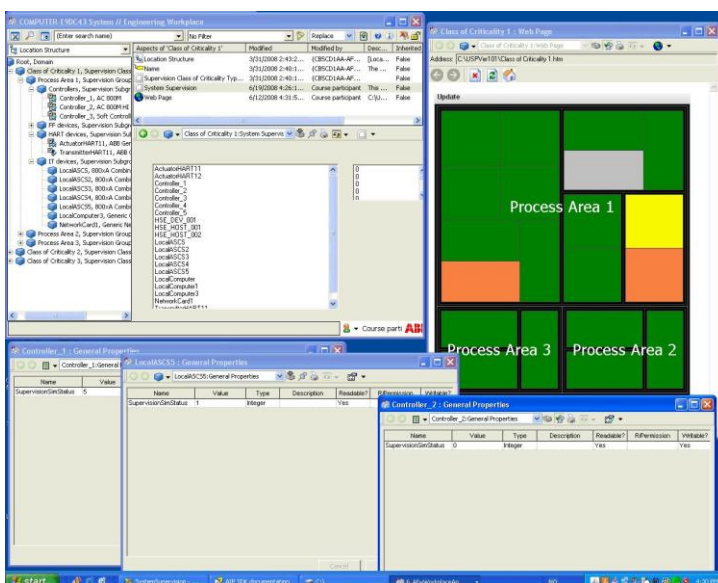


Figure 18. Unified Supervision System prototype



This prototype successfully demonstrated (1) the benefits of the treemap visualisation as an 800xA system overview, (2) a unifying overview from which an existing arsenal of special purpose tools for analysis could be invoked, thus producing an array of possible workflows, and (3) the integration of a visualisation tool into the 800xA environment.

## 4 Telecommunications management use case and workflows

In this section we present visual analysis workflows for three tasks in a telecommunications service introduction scenario: modelling the service components and their interactions, allocation of producer resources to the service, and post-hoc analysis of service log files.

The workflows are based on Virtual Flower (VF), an experimental use case for research purposes provided by Ericsson [5]. Customers use the VF service to order virtual flowers, and optionally real flowers, for their mothers on Mother's Day.

### 4.1 Workflow: Service Modelling

As a first step, we model the service components and their interactions. We will throughout refer to service components as *agents*. VF has three main categories of agents: users, provider, and producers. Users are the customers (children) who invoke the service and the mothers who receive it. The service provider orchestrates the introduction and operations of the service by contracting producers that provide elements such as software development, server operations, network services, and billing.

In Figure 19, we see a high-level service model showing the interactions between the users and the provider during ordering, delivery, and billing. Network provider agents carrying messages are indicated by a prefix before each message. All other agents are shown as nodes.

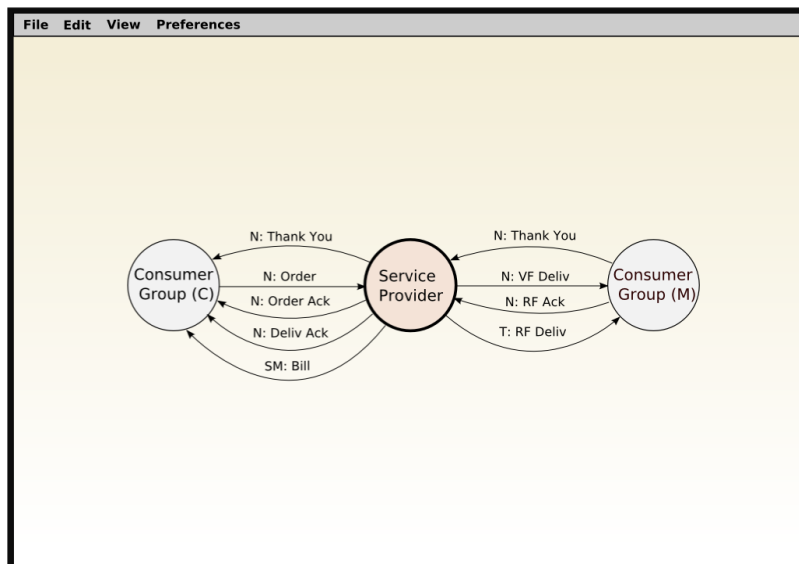
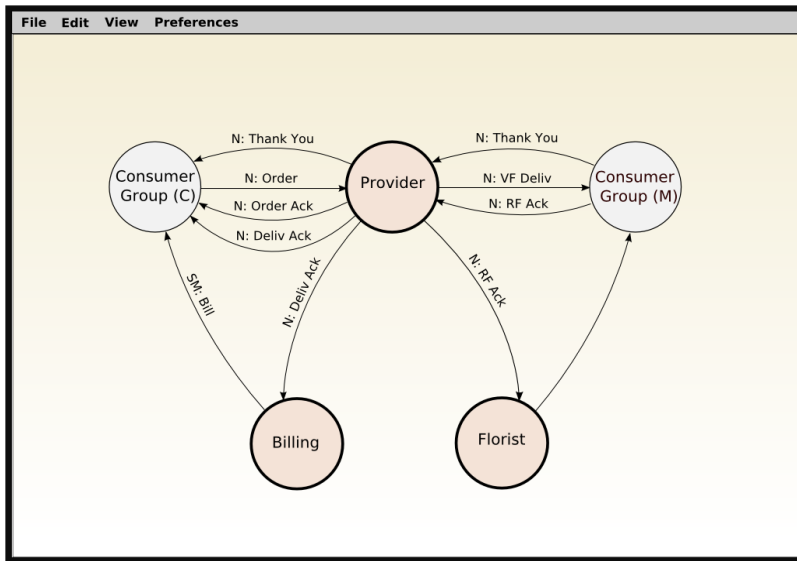


Figure 19. An overview of VF agents and their interaction

We see two user groups, children and mothers, and the provider agent. The links between the agent nodes correspond to messages during the three phases of the service.

In a next step, in Figure 20, we add further detail by introducing explicit agents representing billing and real flower producers.

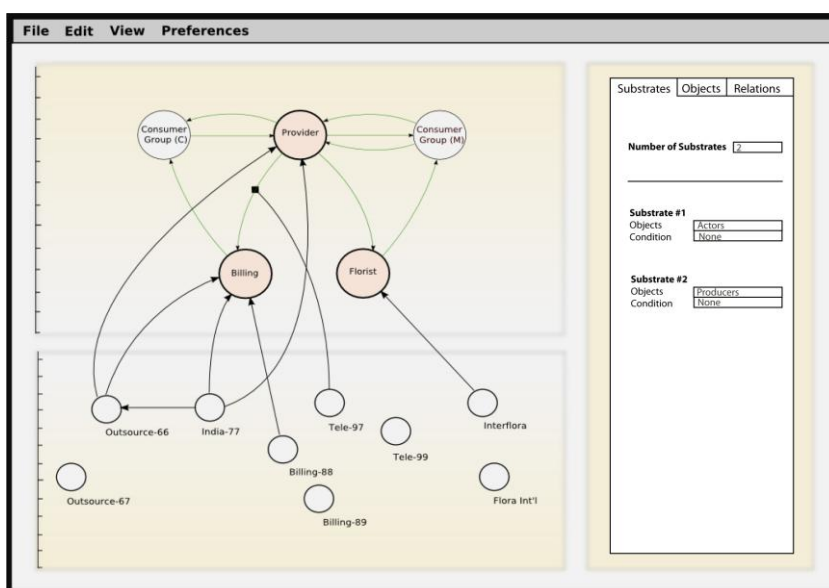


**Figure 20. Provider expanded to show producer agents**

This workflow could continue to refine the model further. The visualization capabilities required for this first workflow are available in most standard design tools: node-link graphs with user-edited layouts. Given more complex services, information visualization techniques such as additional layouts, and zooming and filtering techniques, would help explore the structure. We next extend the workflow using not commonly available techniques. The visualisations in both workflows have been drawn using Adobe Illustrator.

## 4.2 Workflow: Allocation of Resources

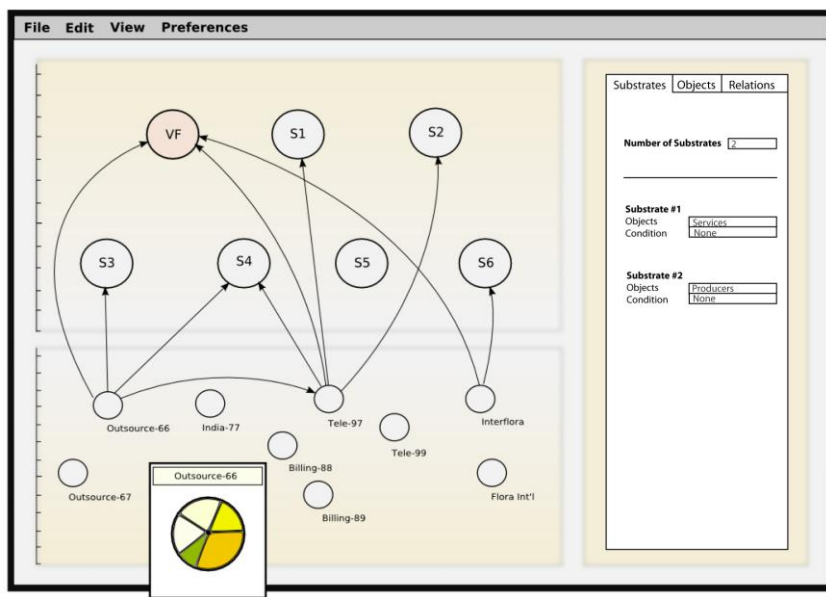
In Figure 21, we add concrete, potentially competing, producers that can provide the necessary resources to implement the roles of the abstract producer agents. We employ a semantic substrate to show the relationship [2].



**Figure 21. Associating producer agents**

The semantic substrate divides the nodes into two different regions. The upper region shows the producer agents and their communication links. The lower region shows the concrete producers. Links between the regions have been selected to show which producers have been contracted to provide resources to the service.

To get an overview of the resource allocation situation, in Figure 22, we display all services in the upper region (VF and S1-S6) and their associated producers in the lower region. The vertical placement of nodes in the regions show total resource use for services and degree of allocation for producers. The pie chart illustrates the relative allocation of a selected resource by each service, and any remaining unallocated resources.



**Figure 22. Services and producers**

### 4.3 Workflows: Post-Hoc Analysis of Log Files

In these workflows, we perform post-hoc analysis of a VF service log file. Previous workflows have been performed with experimental software or have been simulated using drawing tools. Here, we use the commercial state-of-the-art tool Tableau [8, 15, 26] and present the interactions required for each step in greater detail. We expect future visual analysis tools to provide all of the capabilities required for previous workflows in this highly interactive form.

The log file is a record of events generated by a simulator of the VF order phase. There is also a database file of subscribers who have received VF as an optional service.

The subscriber record includes an id, an associated region, and the age and gender of the subscriber. Events are subscriber generated or provider generated. A subscriber-generated event includes a time stamp, the subscriber id, the service requested and a calculated resource cost for the service execution. Provider events include the cost of resources that have been contracted and made available per time unit. Additional events in the simulation, which are

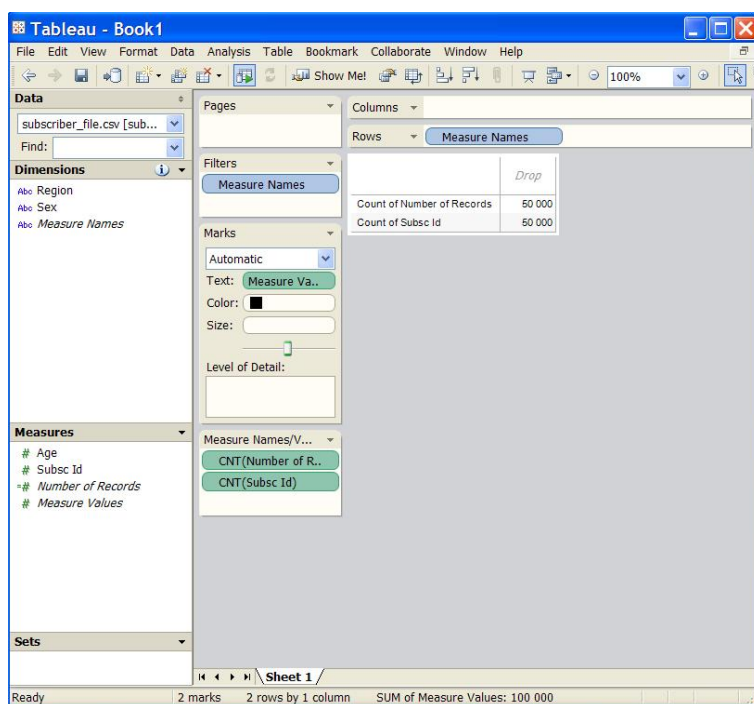
not logged, are advertising campaigns to subscribers about the availability of the VF service. Such events occurred on the 4<sup>th</sup>, 12<sup>th</sup> and 19<sup>th</sup> of May in our simulated scenario, where Mother Day was on the 25<sup>th</sup> of May 2008.

### *Subscriber File: Regional and Age Distribution*

We first check that the subscriber file does not contain any duplicate subscriber records, and compare the number of records and the number of subscriber id's in the listing.

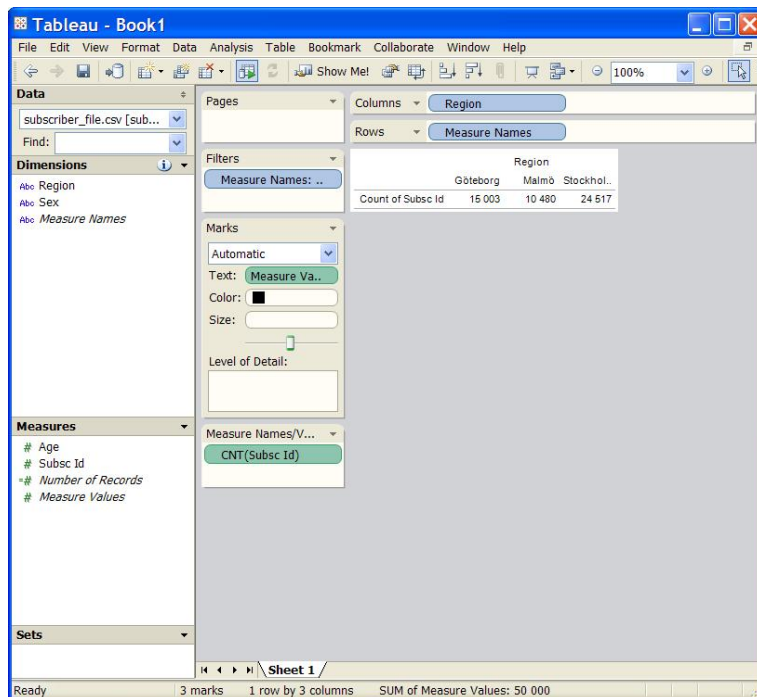
When loading the subscriber file into Tableau, the subscriber record fields appear in the left column in the Dimensions and Measures areas. The fields can be dragged into different parts of the table area in the right column, creating a table, graph or diagram. Between the data and table areas, in the middle column, we find view cards with initial default view settings.

In Figure 23, we have dragged the “Number of Records” and “Subsc Id” fields to the value/measurement field in the bottom view card. The values appear in the table area.



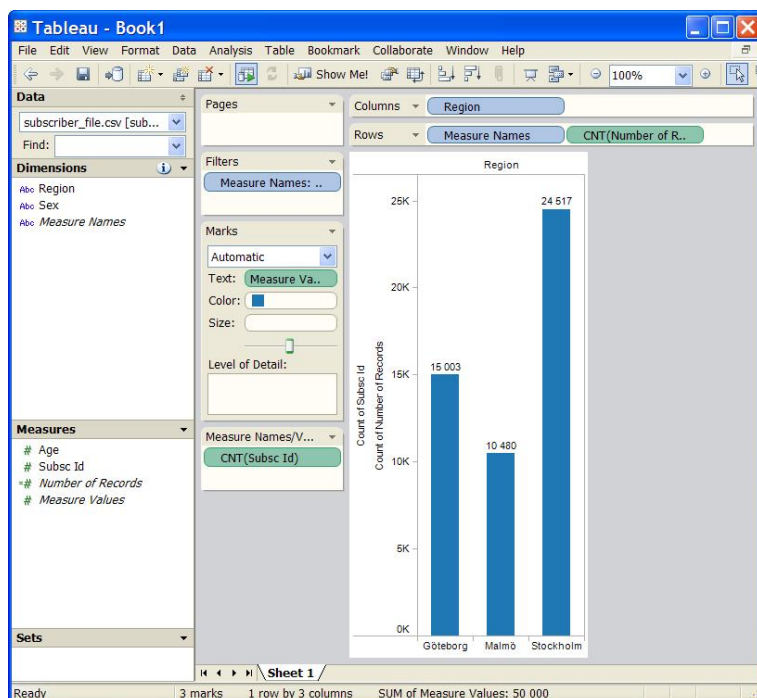
**Figure 23. Number of records and subscribers**

In Figure 24, we remove the “Number of Records” field and drag the “Region” to the column field, splitting the table over columns corresponding to different Region values in the file. We see how the subscribers are distributed over regions.



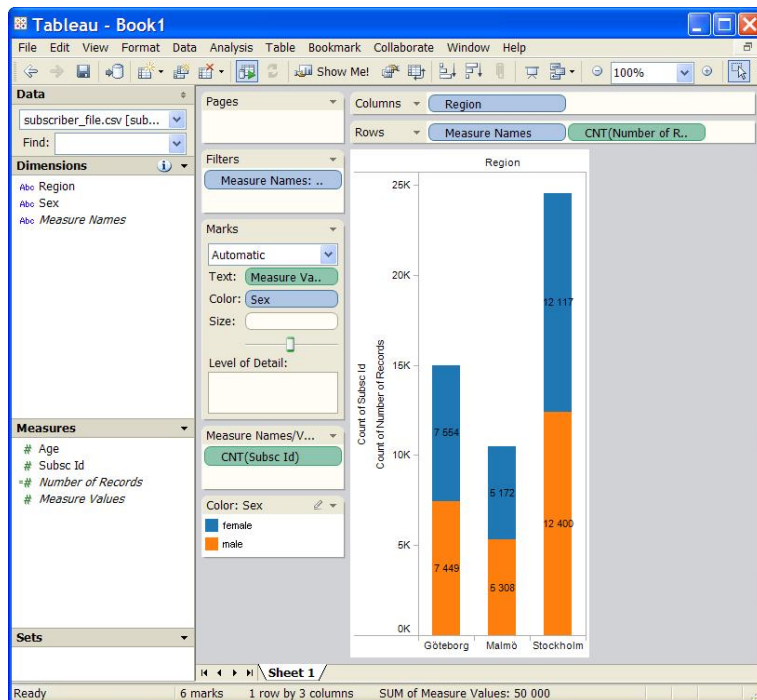
**Figure 24. Subscribers and regional distribution**

We make the proportions between regions more clear in Figure 25 by dragging “Number of Records” to the Rows area. As this is a Measures field, it will cause Tableau to switch from tabular to chart for this axis, by default a bar chart.



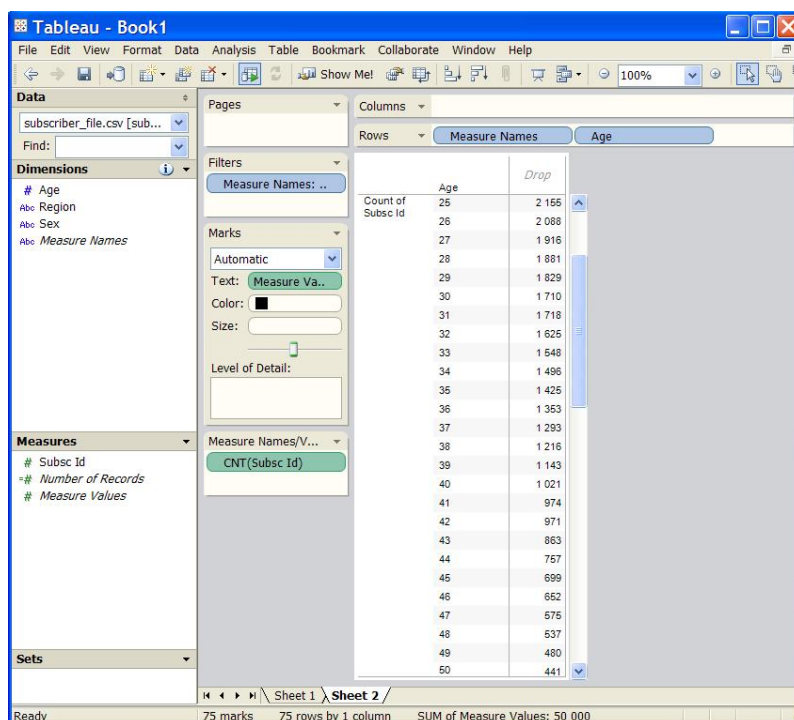
**Figure 25. Bar chart of regional distribution to emphasize proportions**

In Figure 26, we differentiate between men and women to see if we have an uneven distribution in any of the regions. This is achieved by dragging Sex to the Color field in the Mark area of the view cards.



**Figure 26. Proportion of men and women in each region**

In Figure 27, we explore the age distribution. We first backtrack to the empty view (using the left arrow in the tool bar), then change Age into a dimension by moving it from the Measures area into the Dimension area. Finally we drag Age to the Row field and the count of Subsc Id to the measure/value field. This will create one row per age value and display the number of unique subscribers per value.



**Figure 27. Table of the number of subscribers of a certain age**



In Figure 28 we switch to the much clearer bar chart representation by dragging Number of Records to the row field.

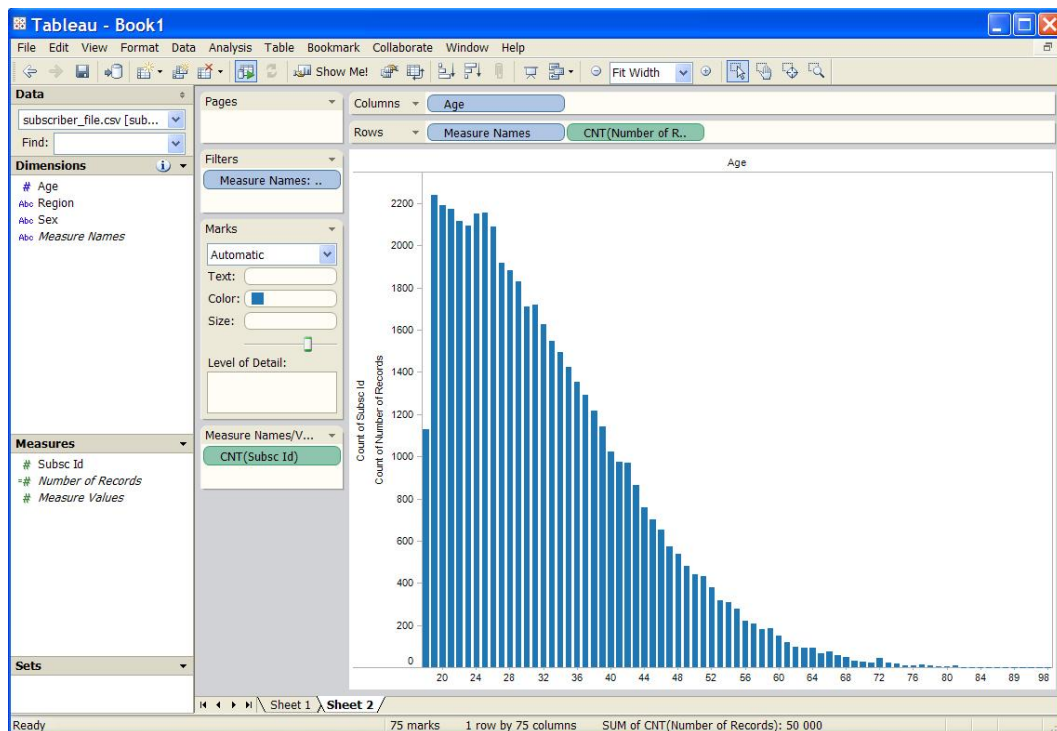


Figure 28. The table as a bar chart

Finally, in Figure 29, each bar is coloured to represent the distribution of a certain age over the set of regions by dragging Region to the Color field in the view cards.

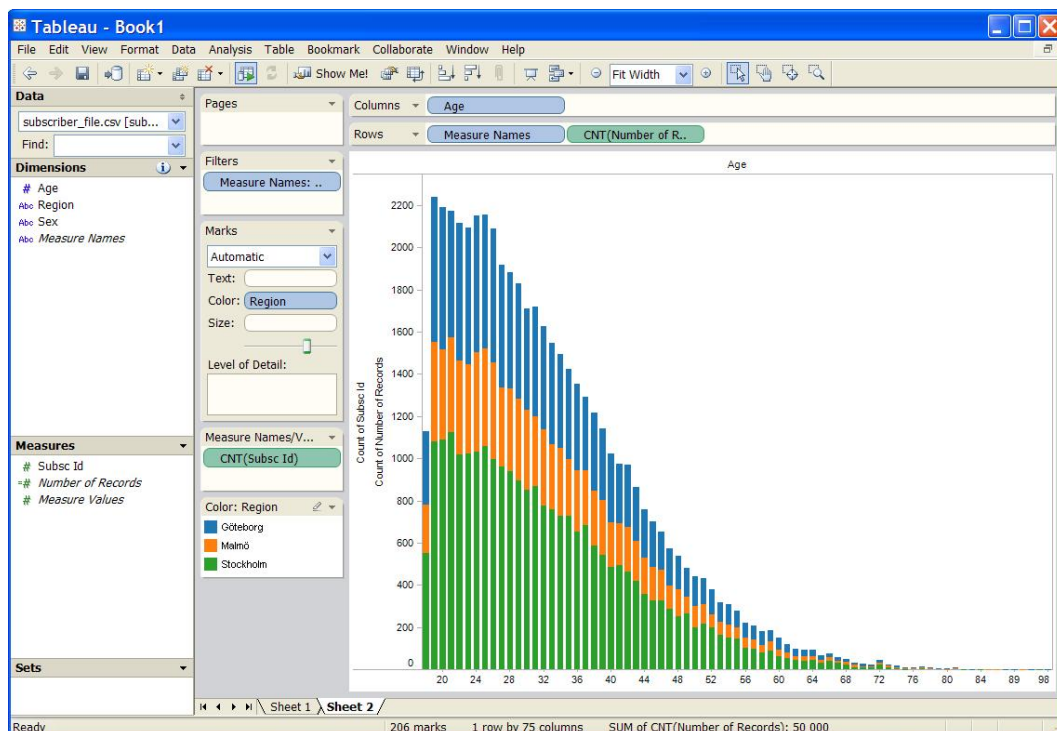


Figure 29. Subscriber age and region

## Subscriber File and Log File: Communication and Activity over Time

We next use the join facility of Tableau to join the subscriber file and the event log file on subscriber id.

In Figure 30, we first check the total number of event records (a). We then check the distribution of the recipients of the service (b). We finally include the region of the subscriber ordering the service to obtain a tabular representation of the communication in and between regions (c). The last table indicates that service recipients (mothers) are more evenly distributed over the regions than those initiating the service (children).

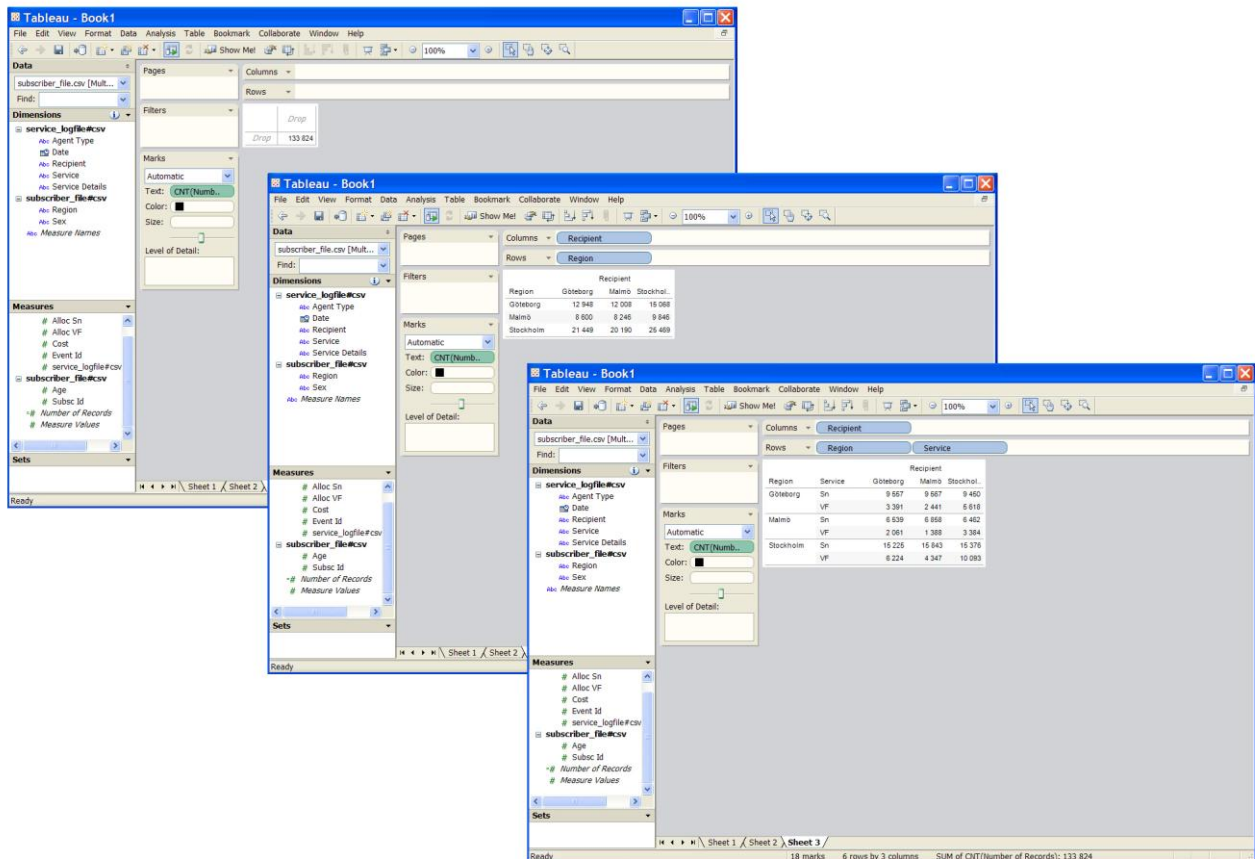
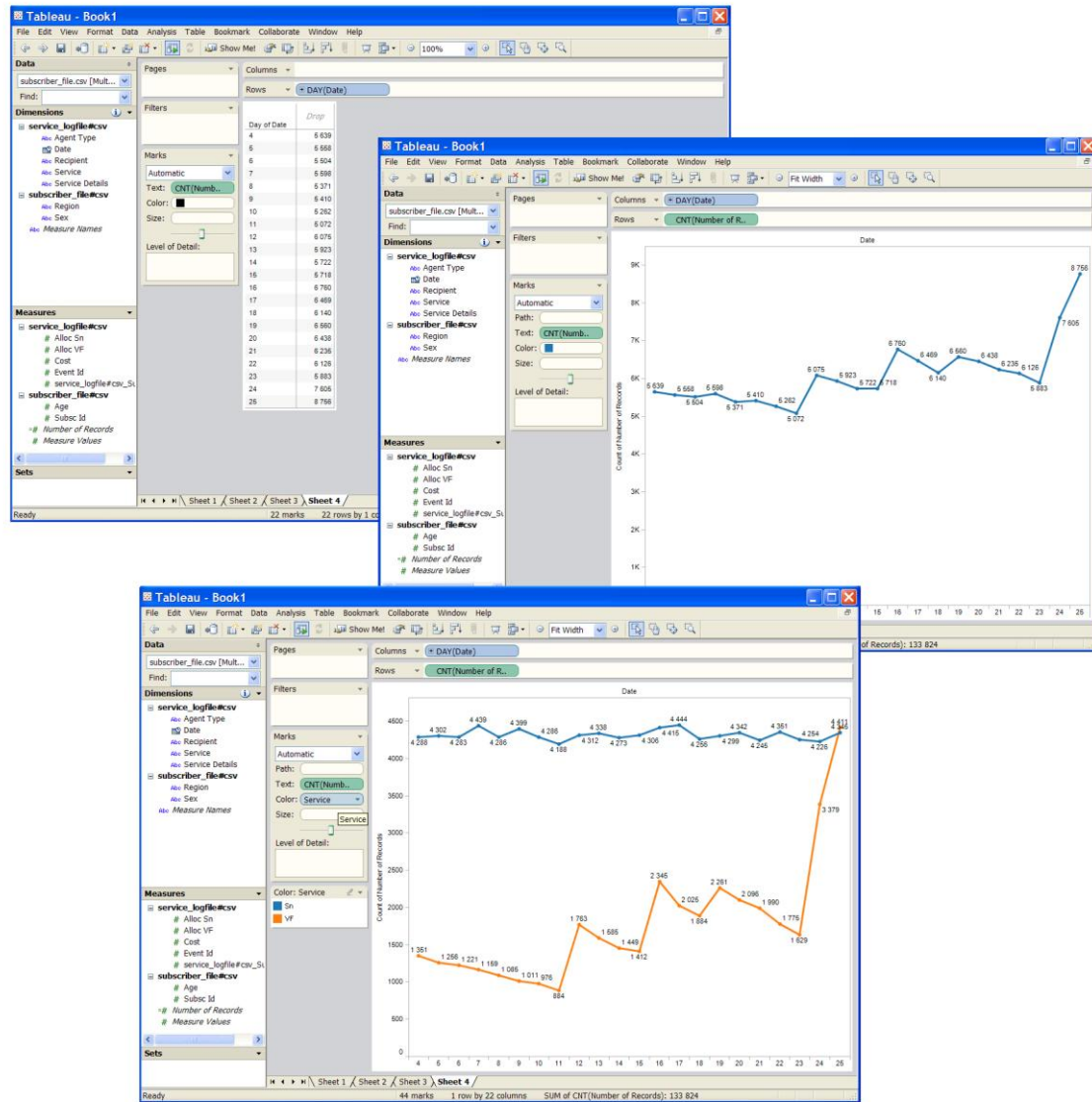


Figure 30: (a) Number of event records (after join with subscriber list), (b) order events per recipient and region, and (c) the number of order events by region, recipient and service

We next explore how the amount of ordering events varies over the days of the period. First we see a tabular representation in Figure 31(a) and graphical representation of the same information in Figure 31(b).



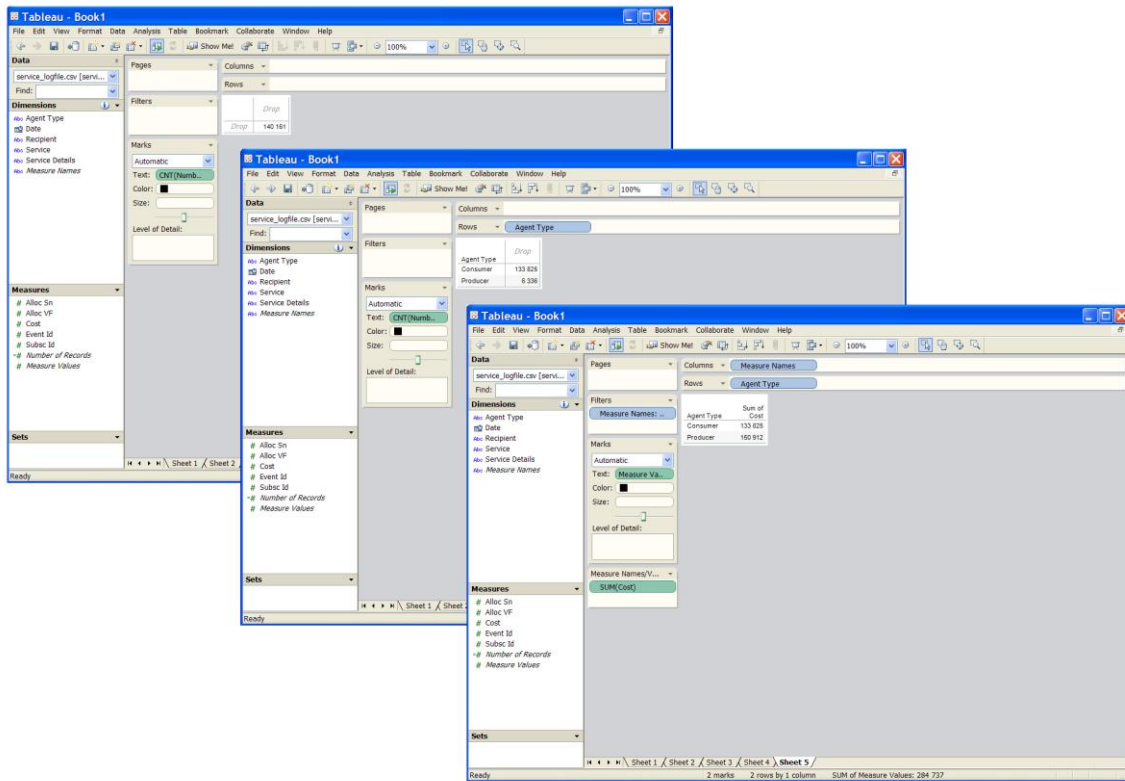


**Figure 31: (a) Events over time in tabular representation, (b) events over time in graphical representation and (c) the number of events over time differentiated by service type**

The log file includes information not only about the VF ordering events but also about other events, here collectively denoted 'Sn' (for n Services). In Figure 31(c) we have differentiated between the VF and Sn services. Here we see that the amount of Sn orders is more uniform than the amount of VF orders. The total amount of Sn orders exceeds the amount of VF orders but the amount of VF orders is more varied over time. We note that the VF graph reflects the influence of the advertisement campaigns and the time left until Mother's Day.

In Figure 32(a) we show the graphical representation on how the orders vary over the regions. Figure 32(b) shows how the orders vary over the set of recipients. We see a difference in distribution of subscribers ordering the service and the distribution of recipients. Recipients are distributed evenly while subscribers are distributed according to the size of the population of each region.

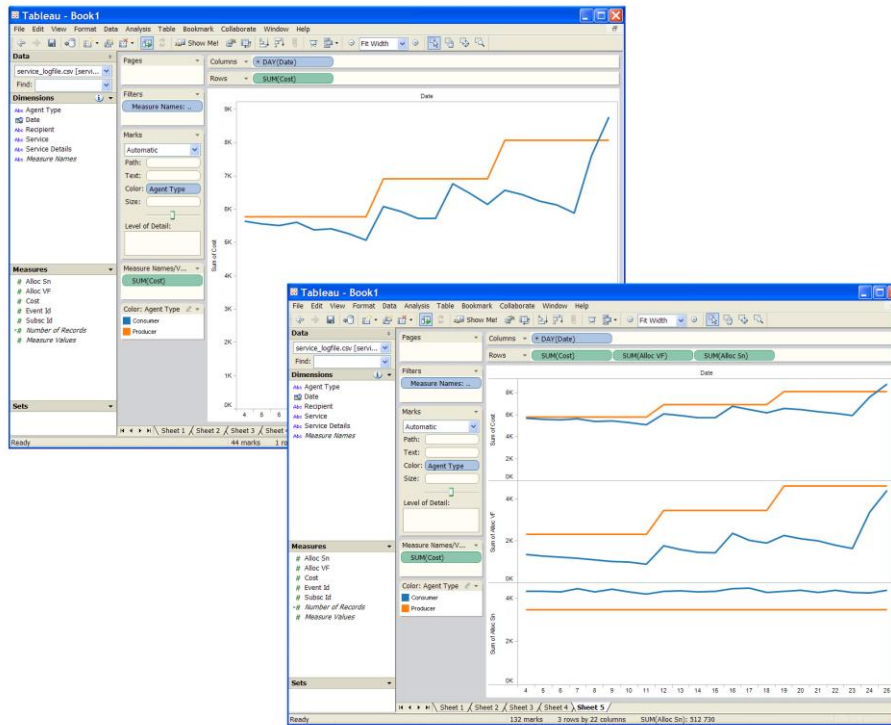




**Figure 33: (a) Total number of events, (b) number of events by consumers and producer, and (c) cost of resources produced and consumed**

In Figure 33(a) we see the total number of events during the period and in Figure 33(b) how these are divided between consumers and producers. In Figure 33(c) we see that the producer cost is a little bit higher which means that, overall, we have not exceeded contracted resources. In Figure 33 we see the total costs summarized over time. However it would be more useful to see how costs vary over time to see if there are any temporary overdrafts.

Figure 34(a) shows a graph illustrating the costs over time for both producer and consumer. Here we note a slight overdraft at the very end of the time period.



**Figure 34: (a) Costs of resources produced and (b) consumed over time shown graphically and costs of resources produced and consumed differentiated by service**

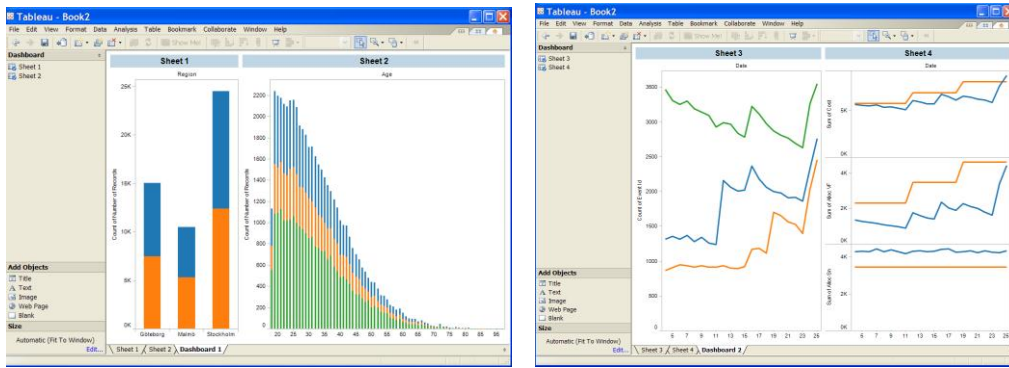
In Figure 34(b) we further distinguish between consumer costs and allocated producer cost by showing the costs for VF and Sn in two additional separate diagrams. From this we see that we have a constant overdraft for the Sn services while having plenty of reserve resources for the VF service. This means that we seem to have made a mistake when allocating resources to the Sn and VF services.

In this and the previous workflow we introduced time, an aspect that we have not studied in any of the preceding examples. This step is based around a simulation of a VF service design that could have been produced in an expanded and complete version of the previous two design steps.

Using Tableau we have shown three different workflows, one looking for the characteristics of our user base, one to check that the amount of traffic and the traffic patterns looked as expected and one to check that our contracted resources matched what the (simplified) cost function had reported.

### *Combining Visualisations into Dashboards*

It can be useful to collect visualisations into a dashboard, to reuse the efforts in a workflow and quickly observe changes, e.g., when updating the service design and performing a new simulation. Figure 35 shows two such dashboards, where the first gives an overview of the subscriber base, based on Figures 26 and 29, and the second, based on Figures 32(a) and 34(b), gives an overview of traffic and cost.



**Figure 35: (a) Subscriber dashboard, distribution over region and age (b) traffic dashboard, traffic over region and real versus allocated resources as costs**

## 5 Discussion

In this final section, we discuss current visualisation tools, the study of visual analysis workflows and topics for further research on visual analysis tools.

### 5.1 Visualisation tools

In the previous sections we have given examples of workflows showing the potential of visualisation in the domain of networked systems. In these, the Prefuse-based platform, the Treemap demo software, the Semantic Substrate concept and the Tableau tool all, in different ways, address the requirements on visualisation tools for networked systems.

The Prefuse-based platform has an array of different layout algorithms that can be interactively applied to a networked system, visualising both its structure and its traffic, and letting the user easily switch between different views of the current data selection.

The Treemap software does not have that ability, but in the workflow where it is used, the user task is less explorative in the sense that the user monitors the treemap for status changes and then drills down into the map for further details of malfunctioning devices. This interaction and the ability to reorganize the treemap according to the choice of tiling algorithm or tree branching rules offers functionality well beyond that of the Prefuse-based platform in its current incarnation.

The Semantic Substrate concept can be set up for visual analysis of a wide range of node-link graphs representing abstract objects and relations. The filtering functions that come with the definition of the substrates are highly useful when focusing the view.

Tableau on the other hand is limited to traditional tables and graphs, but has an extremely well-developed interactive functionality making it easy to quickly explore alternate views of complex log data sets with reasonably large number of variables.

The ideal framework or tool supporting visual analysis of networked systems would combine the abilities of these tools and still be easy to configure and customize. It is our intention that the workflows presented should suggest that such a tool is conceivable and desirable.

### 5.2 The Study of Workflows

We have here offered examples of workflows, and believe that further study of workflows will offer valuable insights. Our focus is to point the way towards future tools, and we have therefore discussed workflows in terms of the visualisation techniques used and the actions

taken by the user. However, workflows should also be analysed in terms of the characteristics of the visual analysis problem:

What is the structure of problem domain? What is its place in the networked system lifecycle? What types of data are we exploring? What are the objects and relations? What is common to networked systems, such as nodes, traffic, status, and events, and what is specific to the application domain?

What is the role of the user? Is the user a developer, exploring data to pinpoint the reason for some unwanted behaviour, an operator, monitoring an ongoing process using a preconfigured visualisation or a field engineer, configuring visualisations for an operator? What kind of answer, insight or verification is the user looking for?

Both the ABB and the Ericsson workflows focus on supporting the management of networks and their services but do so at different levels, supporting different user roles and tasks.

In the two first ABB 800xA examples the user has a more exploratory role, first by getting familiar with the network structure, then going stepwise into the details of the communication patterns, looking for anomalies and possibly failing devices and their place in the network and in geography. In the third ABB 800xA example, the role is less explorative and more monitoring, that of an operator overseeing industrial machinery. Structure is a key component of the data sets. Part of the data, traffic and status information is dynamically generated.

If the role of the user in the ABB examples was that of an engineer, maintainer or operator, the role of the user in the Ericsson Virtual Flower workflows is that of a designer. The workflows correspond to design steps, which, although incomplete, could be repeated in a design-test-refinement cycle in the design of a new service. In the first two steps, the user is supposed to provide information for the design of the service. Extending the standard role of a design tool, the visualisation serves as drawing board for pre-existing and added information about the service and its necessary sub-services and producers. Finally, the result of a simulation of the design is analysed to verify design assumptions.

### **5.3 Topics for Further Research on Visual Analysis Tools**

We have identified three main topics for further research on visual analysis tools for networked systems: handling dynamism in data sets and visualisations, open system architectures for visual analysis frameworks and tools, and integration of statistical data analysis techniques.

The ideal framework or tool supporting visual analysis of networked systems would combine the abilities illustrated in the examples and summarized in Section 5.1. What we have not studied are scenario examples where the data set to be visualised is dynamically changing over time. The tools we have used offer basic support for dynamic data sets and visualisations. Prefuse has mechanisms for animation and for table and display update. In Tableau a visualisation can be renewed periodically, reflecting a potentially updated data set.

One question in this context is how the change should be shown visually depending on the visual representation used. Dynamic data sets also imply the need and possibility for high level interactions like stop, replay and recording of visualisation sequences and settings. This further implies the need for an extended infrastructure analogous to the Prefuse based platform discussed in Section 3.1.

To be easily adaptable to various scenarios, such an infrastructure would be based on an open system architecture (to be defined). The core of such an architecture could consist of more detailed instances and variations of the visualisation reference model, as suggested in [10].

There would be several benefits from such an approach, as new interaction and layout algorithms may be easily incorporated and the overall configuration may be simpler and quicker to perform. Having dynamic data would also affect the core architecture and having a tabular representation of the initial and filtered data may not be suitable for all scenarios. At a higher level such a system architecture would include support for management and coordination of parallel visualisation views, subscription of dynamic data, and the high level interactions previously mentioned.

## Acknowledgements

We would like to thank our industrial partners Marius Petru-Stanica, Rolf Setthammar, Henrik Johansson, and colleagues at ABB Corporate Research; and Susanne Engberg, Martin Johnsson, and colleagues at Ericsson Research, for their contributions to this work, including detailed information on the industrial domains, input and feedback on use cases, and discussions on visualisation workflows.

Thanks also to Pär Törnberg and Kalle Ojaniemi, who implemented the Prefuse-based visualisation platform as part of their Master's Thesis projects, and to Tableau Software for giving us access to Tableau Professional Desktop Edition.

The work presented herein was supported by the Knowledge Foundation (KK-stiftelsen, Dnr: 2007/0155) and by SICS Center for Networked Systems.

## Bibliography

- [1] ABB, Industrial IT System 800xA, <http://www.abb.com/product/se/9AAC115756.aspx>
- [2] A. Aris, B. Shneiderman. Designing Semantic Substrates for Visual Network Exploration, HCIL Technical Report, HCIL-2007-27
- [3] CAIDA, the Cooperative Association for Internet Data Analysis, <http://www.caida.org/tools/visualization/walrus/gallery1>
- [4] S. K. Card, et al. Readings in Information Visualization, Morgan Kaufmann, 1999
- [5] S. Engberg, J. Mellberg, Virtual Flower Use-case, Ericsson Research, internal report extract, 2008
- [6] J-D. Fekete, The InfoVis Toolkit, IEEE Symposium on Information Visualization 2004, October 2004
- [7] Graphviz - Graph Visualization Software, <http://www.graphviz.org/>
- [8] P. Hanrahan, C. Stolte, J. Mackinlay, Visual Analysis for Everyone: Understanding Data Exploration and Visualization, Tableau white-paper, <http://www.tableausoftware.com/>
- [9] J. Heer et al., The Prefuse visualization toolkit, <http://prefuse.org/>
- [10] J. Heer and M. Agrawala, Software Design Patterns for Information Visualization, IEEE Transactions on Visualization and Computer Graphics, Vol. 12, No. 5, September/October 2006



- [11] I. Herman et al, Graph Visualization and Navigation in Information Visualization: a Survey, 2000 IEEE, <http://citeseer.ist.psu.edu/herman00graph.html>
- [12] InfoVis:Wiki, Visualization Pipeline, [http://www.infovis-wiki.net/index.php/Visualization\\_Pipeline](http://www.infovis-wiki.net/index.php/Visualization_Pipeline)
- [13] IVC Software Framework, InfoVis CyberInfrastructure- Software, <http://iv.slis.indiana.edu/sw/>
- [14] B. Lorensen, On the Death of Visualization, [http://visual.nlm.nih.gov/evc/meetings/vrc2004/position\\_papers/lorensen.pdf](http://visual.nlm.nih.gov/evc/meetings/vrc2004/position_papers/lorensen.pdf)
- [15] J D. Mackinlay, P. Hanrahan, C. Stolte, Show Me: Automatic Presentation for Visual Analysis, <http://www.tableausoftware.com/files/081027infovis-showme-vfinal-fix.pdf>
- [16] T. Munzner, 15 Views of a Node Link Graph: An Information Visualization Portfolio, <http://video.google.com/>
- [17] NAMUR Recommendation, NE 107, Self-Monitoring and Diagnosis of Field Devices, 2006
- [18] K. Ojaniemi. Network Visualisation: Animation, interaction techniques and work towards a general purpose platform for visualisation of network based systems, KTH Masters Thesis, 2009 (forthcoming)
- [19] J. O'Madadhain, D. Fisher, P. Smyth, S. White, Y-B. Boey, Analysis and Visualization of Network Data using JUNG, [http://jung.sourceforge.net/doc/JUNG\\_journal.pdf](http://jung.sourceforge.net/doc/JUNG_journal.pdf)
- [20] C. Plaisant, B. Shneiderman et al., Treemap, University of Maryland/Human-computer Interaction Lab, <http://www.cs.umd.edu/hcil/treemap/#download>
- [21] SequoiaView, [http://w3.win.tue.nl/nl/onderzoek/onderzoek\\_informatica/visualization/sequoiaview](http://w3.win.tue.nl/nl/onderzoek/onderzoek_informatica/visualization/sequoiaview)
- [22] B. Shneiderman, Treemaps for space-constrained visualization of hierarchies, <http://www.cs.umd.edu/hcil/treemap-history/>
- [23] B. Shneiderman, Crossing the Information Visualization Chasm, 1999, <http://www.cs.umd.edu/hcil/pubs/presentations/info-viz-chasmslides/index.htm>
- [24] R. Spence, Information Visualization, Design for Interaction, second edition, Prentice Hall 2007
- [25] M-P. Stanica, System Supervision Unified Software Architecture Concept, ABB Technical report, ABB Research, 2008
- [26] Tableau, available at: <http://www.tableausoftware.com/>
- [27] P. Törnberg. Network Visualisation: Layout algorithms and work towards a general purpose platform for visualisation of network based systems, KTH Masters Thesis, 2009 (forthcoming)
- [28] Visualization ToolKit (VTK), <http://www.vtk.org/>
- [29] J.J. van Wijk, The Value of Visualization, <http://www.win.tue.nl/~vanwijk/vov.pdf>