

A web-enabled visualization toolkit for geovisual analytics

Quan Van Ho, Patrik Lundblad, Tobias Åström and Mikael Jern

Information Visualization

11(1) 22–42

© The Author(s) 2011

Reprints and permissions:

sagepub.co.uk/journalsPermissions.nav

DOI: 10.1177/1473871611425870

ivi.sagepub.com



Abstract

A framework and class library (GAV Flash) implemented in Adobe's ActionScript is introduced, designed with the intention to significantly shorten the time and effort needed to develop customized web-enabled applications for geovisual analytics tasks. Through an atomic-layered component architecture, GAV Flash provides a collection of interactive geographic and information visualization representations for exploring high-dimensional spaces that are extended with motion behaviour. Versatile interaction methods are drawn from many data visualization research areas and optimized for dynamic web visualization of spatio-temporal and multivariate data. Based on layered-component thinking and the use of programming interface mechanism, the GAV Flash architecture is open and facilitates the creation of new or improved versions of existing components so that ideas can be tried out or optimized rapidly in a fully functional environment. GAV Flash is not only a tool for interactive visualization, it also supports storytelling around visual analytics in which visual representations serve not only as a discovery tool for individuals but also as a means to share stories among users fostering a social style of collaborative data analysis. A mechanism 'snapshot' for saving the explorative results of a reasoning process is introduced, which aids collaboration and publication of gained insight and knowledge embedded as dynamic visualizations in blogs or web pages with associative metadata or 'storytelling'.

Keywords

web-enabled visualization toolkit, geovisual analytics, storytelling, layered component architecture, multiple-linked views

Introduction

The rise of 'Web 2.0' has become undisputedly linked with developments such as blogs, wikis, social networking and collaborative software development. While the benefits of visual analytics tools are many, it has been a challenge to adapt these tools to the Internet and reach a broader user community scaling not only the size of the audience but also the data. Research has so far focused more on tools that explore and present data,^{1–3} whereas tools that support the creation and publishing of gained knowledge with integrated storytelling have not been granted the same attention.

In this context, we introduce a class-library GAV Flash programmed in Adobe's ActionScript with a large collection of interactive web-enabled visualization components facilitating an emerging interdisciplinary field, named geovisual analytics, which integrates perspectives from visual analytics, geographic and information visualization, geospatial semantics and knowledge management, and spatial analysis. Tools using cognition and perception as foundations for

visualization help identify relevant geospatial information and knowledge through analytical reasoning that melds natural human abilities of vision and cognition with interactive visual interfaces are specifically designed to provide support for storytelling. Geovisual analytics represents a distinct synergy between geographic and information visualization and associated application areas (Figure 1) such as public statistics, environmental, logistics, meteorology, social science, education, energy, decision support and visual business intelligence. The name emphasizes the link with the discipline of visual analytics and can be

Department of Science and Technology, National Center for Visual Analytics (NCVA), Linköping University, Linköping, Sweden.

Corresponding author:

Quan Van Ho, Department of Science and Technology, National Center for Visual Analytics (NCVA), Linköping University, Linköping, Sweden

Email: ho.van.quan@itn.liu.se

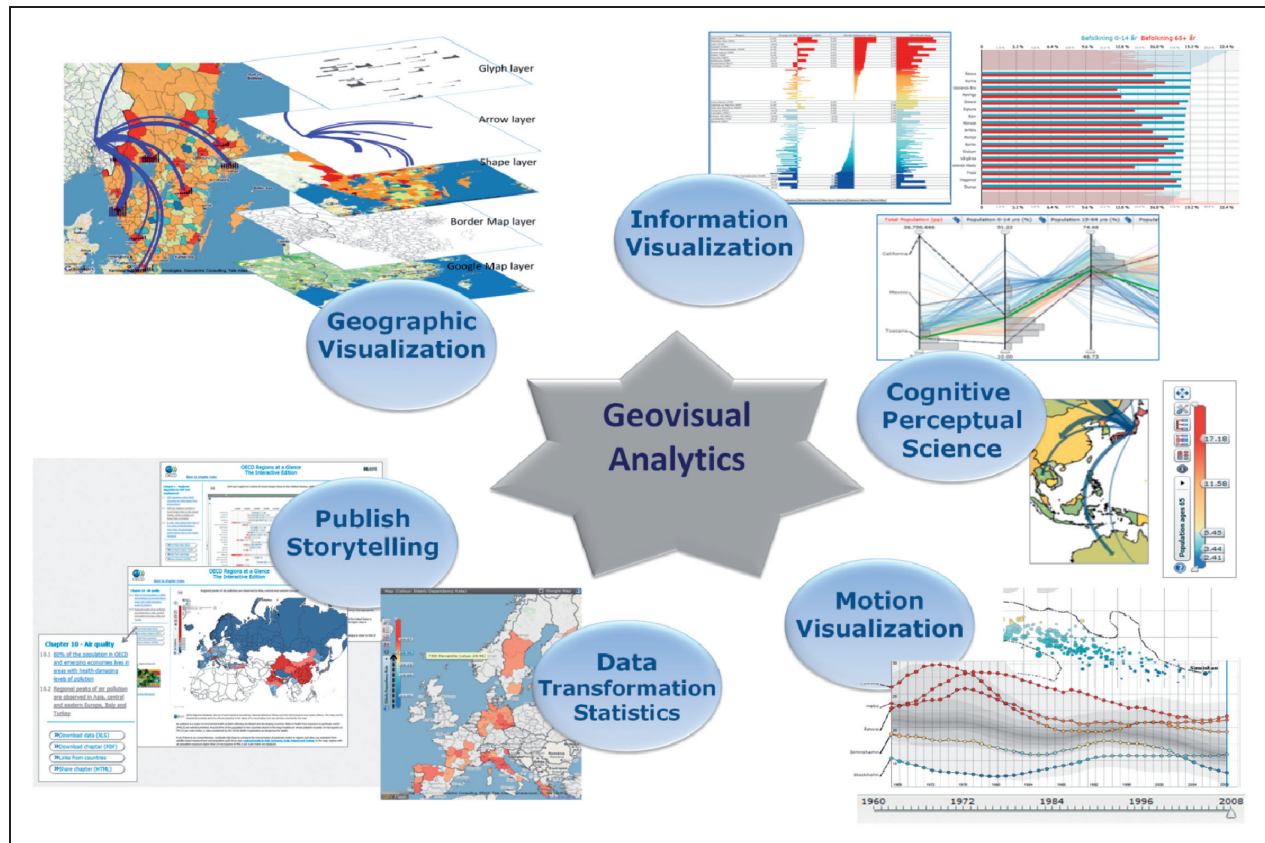


Figure 1. Geovisual analytics represents a distinct synergy between geographic and information visualization, interactive visual user interfaces using perceptual methods, storytelling, time and dynamic data transformation.

viewed as sub-area with specific focus on spatial and temporal dimensions.

Geovisual analytics represents a distinct synergy between geographic and information visualization, interactive visual user interfaces using perceptual methods, storytelling, time and dynamic data transformation.

Why use Flash? Originally developed for .NET,⁴ GAV was converted into the ActionScript programming language in 2008 after serious demands from the statistics user domain, for example the Organisation for Economic Co-operation and Development (OECD), European Commission, World Bank, and many national statistics bureaus interested in our geovisual analytics technology. This growing interest in geographic and information visualization was related to a widespread introduction of Gapminder,⁵ a statistics bubble animation tool that was later acquired by Google. Flash became a de facto standard within the statistics community because of its widespread use and easy accessibility to all citizens.

Based on layered-component architecture, a developer builds an application from bottom to top by either using already-created GAV Flash functional

visualization components, creating their own or modifying one of the atomic layers to create a modified functional component (Figure 2).

GAV Flash views are time-linked, which is important in the synthesis of animation within explorative data analysis. The user can stop the time animation and start interacting with the data at any time step.

Multivariate and temporal data are analysed through the use of time-linked views controlled by a time slider (Figure 2). Trends are detected through several visual representations simultaneously, each of which is best suited to highlight different patterns and can help stimulate the analytical visual thinking process so characteristic for geovisual analytics reasoning. Of particular interest is our Flash implementation of motion-graph components through a dynamic histogram with embedded focus and context technology, a scatter matrix coordinated with a scatter plot, a table lens, an extended parallel coordinates plot (PCP), a choropleth-layered map structure, map glyphs and flow maps. Interactive features include tooltips, brushing, highlight, visual inquiry and conditioned statistics filter mechanisms that can discover outliers and simultaneously update all views.

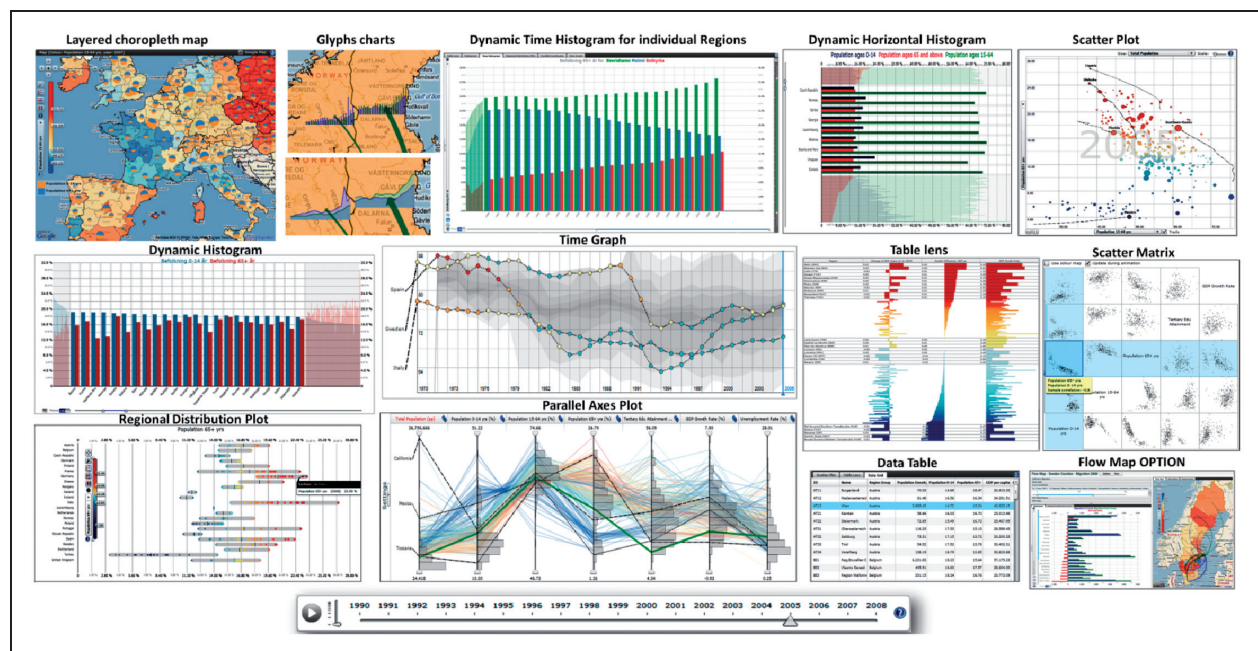


Figure 2. GAV Flash views are time-linked, which is important in the synthesis of animation within explorative data analysis. The user can stop the time animation and start interacting with the data at any time step.

These components are applied in three characteristic case studies that are included to demonstrate and evaluate the tools: statistics data visualization (OECD), self-organizing mobile networks (Ericsson Research, Sweden) and road emergency (SMHI), developed in close collaboration with our domain experts.

In addition to developing a visualization framework based on the Flash Web platform, an integrated snapshot with an associated storytelling mechanism was implemented in each component that will enable collaboration and transition of spatio-temporal and multivariate data into communicative sense-making news entities with contextual metadata. The analyst uses tools (Figure 3) to (1) import data, (2) explore and discern trends, (3) orchestrate with snapshots and metadata, (4) collaborate with colleagues to confirm results and (5) finally publish essential gained insight and knowledge embedded as dynamic visualization ‘Vislet’ in a blog or web page with associative metadata.

A scenario in World eXplorer,⁶ a statistics geovisual analytics application developed with the GAV Flash toolkit for accessing statistical data from the world databank through an API, importing user data, exploring and publishing statistical data on the web.

In summary, this paper introduces a web-enabled visualization and publication toolkit supporting geovisual analytics tasks, facilitating:

- customizable interactive motion visual representations for Web publishing;
- a framework for the creation of user components such as data transformers and also for making changes to existing low-level visualization components so that ideas can be tried out rapidly in a fully functional useful environment;
- coordinated multiple, time-linked views;
- component-embedded interactions including brush, pick, highlight, filter, dynamic sliders, focus and context, and other special interaction facilities;
- a data cube model for fast access to spatio-temporal and multivariate attribute data required for time animation;
- data uploading using standard importers such as SDMX or PCAXIS or implementing your own database access;
- the use of online maps such as Google maps or Bing maps as background maps and putting glyph layers on top of the online maps;
- an integrated snapshot mechanism for saving and packaging the results of a geovisual analytics reasoning process – the foundation for Storytelling; and
- the publication of (HTML code) discoveries and knowledge in blogs or web pages as embedded dynamic visualizations.

The remainder of the paper is organized as follows. The ‘Related work’ section gives an overview of related work. The ‘GAV Flash framework’ section presents our GAV Flash framework. Concepts of snapshots and storytelling are presented in the

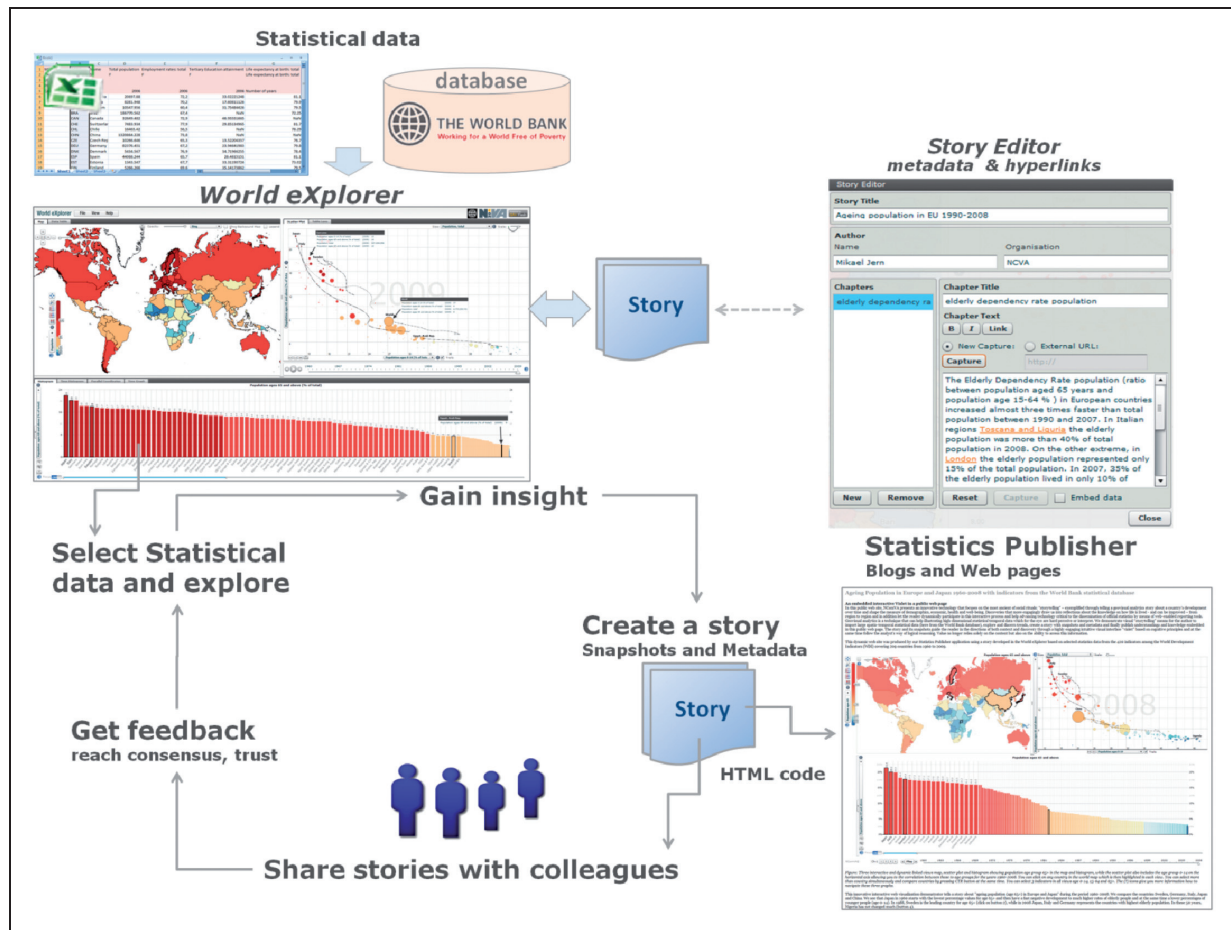


Figure 3. A scenario in World eXplorer [6], a statistics geovisual analytics application developed with the GAV Flash toolkit for accessing statistical data from the world databank through an API, importing user data, exploring and publishing statistical data on the web.

‘Storytelling’ section, then three case studies are presented in the ‘Case studies’ section with user feedback and the conclusion in the ‘conclusion and user feedback’ section.

Related work

We divided generic tools for explorative data analysis into two categories: tools supporting for visualization and tools extended with built-in snapshot and storytelling capabilities.

Toolkits for spatio-temporal multivariate data

Visual exploration of spatio-temporal and multivariate data has been the subject of many research papers. Andrienko and Andrienko^{1,7} have illustrated several motivating approaches in earlier papers. Muller and Schumann² and Guo et al.³ are other appealing examples. Many papers⁸ emphasize the advantages of multiple linked views.

InfoVis Toolkit,⁹ CommonGIS,¹⁰ GeoVista,¹¹ VIS-STAMP,³ GAV⁴ and CGV¹² are examples of exploratory data analysis (EDA) tools that all have evolved from research and can use visualization and computational methods to search for space-time and multivariate patterns. While the benefits of geovisual analytics tools are many, it has been a challenge to adapt these tools to the Internet and reach a broader user community, which is one of our goals. Web-enabled tools are therefore needed for applications explicitly designed with the purpose of visualizing, exploring and communicating spatio-temporal and multivariate data through web environment. Such tools should also employ data transformers and data providers, layout mechanisms, interaction, time animation and storytelling suited for a geovisual analytics’ task. Protovis^{13,14} and prefuse flare¹⁵ are examples of web-enabled tools. They include a collection of interactive visualization components. Nevertheless, they do not seem to support linking mechanisms, for example through selection or filtering, to create multiple linked views. They also do not

support time-animation and storytelling mechanism which we want to support in our toolkit.

Tableau¹⁶ and its predecessor Polaris¹⁷ are other examples of a popular web-enabled tool applied to business analytics. In addition to a collection of interactive visualizations, they support multiple linked views, storytelling mechanisms and geographical maps.

Although some of the web-enabled tools above have geographical map components, they do not seem to support working with online maps that support users in more easily identifying geographical locations when working with geographical data, and allow putting visualizations, for example glyph layers, on top of the online maps, which we want to support in our toolkit. They also do not support focus and context features for geographical maps.

Tools for storytelling

The importance of a capacity to snapshot EDA sessions¹⁸ and then reuse them for presentation and evaluation was early demonstrated by MacEachren et al.¹⁹ and Jern²⁰ and incorporated features to capture and reuse interactions and integrate them into electronic documents. CCMaps²¹ presents a mapping tool that allows users to save snapshot events and reuse them for presentation purposes. Another effort was made by Visual Inquiry Toolkit³ that allows users to place pertinent clusters into a ‘pattern-basket’ to be reused in the visualization process. Robinson²² describes a method they call ‘Re-Visualization’ and a related tool ReVise that captures and reuses analysis sessions. Keel²³ describes a visual analytics system of computational agents that supports the exchange of task-relevant information and incremental discoveries of relationships and knowledge among team members commonly referred to as sense making. Many Eyes²⁴ is an interesting storytelling approach implemented for a public website where novice users can upload their own data, create dynamic visualizations and participate in discussions. Nevertheless, Many Eyes does not seem to support creating multiple-linked views and time animation.

To our best knowledge, there is no publication of a web-enabled visualization toolkit which includes a collection of interactive visualizations and at the same time supports multiple-linked views, storytelling and working with geographical and online maps. Therefore, we build such a visualization toolkit based upon previous research²⁵ and extend our previous toolkit and framework GAV for dynamic web visualization, which supports both the expert and public user.

GAV Flash framework

The GAV Flash framework is developed based on a recommendation from the visual analytics (VA) research programme:²⁶ ‘support seamless integration of tools so that data requests, visual analysis, note-taking, presentation composition and dissemination all take place within a cohesive environment’ addressing the need for integrated exploratory, analytical reasoning and communicative tools. Common geovisualization and information visualization components are included which support interactive features such as tooltips, brushing, highlight, visual inquiry and conditioned filter mechanisms that can discover outliers and methods supporting time-linked multiple views. Also included are tools that support data-analysis algorithms, tools that connect the components to each other and data providers that can load data from various sources (Figure 4).

The GAV architecture allows new or existing components/classes to be incorporated with the already existing components, for example special methods used in our case studies (‘Case studies’ section). Means are also provided for a developer to extend and further customize the popular information visualization methods by breaking them into lower-level ‘atomic’ components (Figures 5 and 6).

Framework design principle

The core philosophy of GAV Flash is modularity, we want application developers to be able to pick and choose from a wide range of visualizations, data providers and data transforms and combine them in various ways. This puts a high demand on each component of the framework to be generalized so it can receive and communicate data with others. At the same time, each component should be self-contained so that the advanced functionality is always present, no matter which components are combined. The generalization is achieved through the definition of interfaces, which detail only the necessary functions and properties in assets shared by components. An example of this is the data set, whose interface is limited to functions that supply data and metadata – all other functionality is encapsulated in the implementation. As the components are only aware of the interfaces, we can easily replace the data set with some other structure, for example a direct database connection, without reimplementing any visualizations or data processors. Apart from the data sets, GAV Flash applications are built using a combination of visualization components and linking modules that control selection, filtering, colour and animation. Other components handle

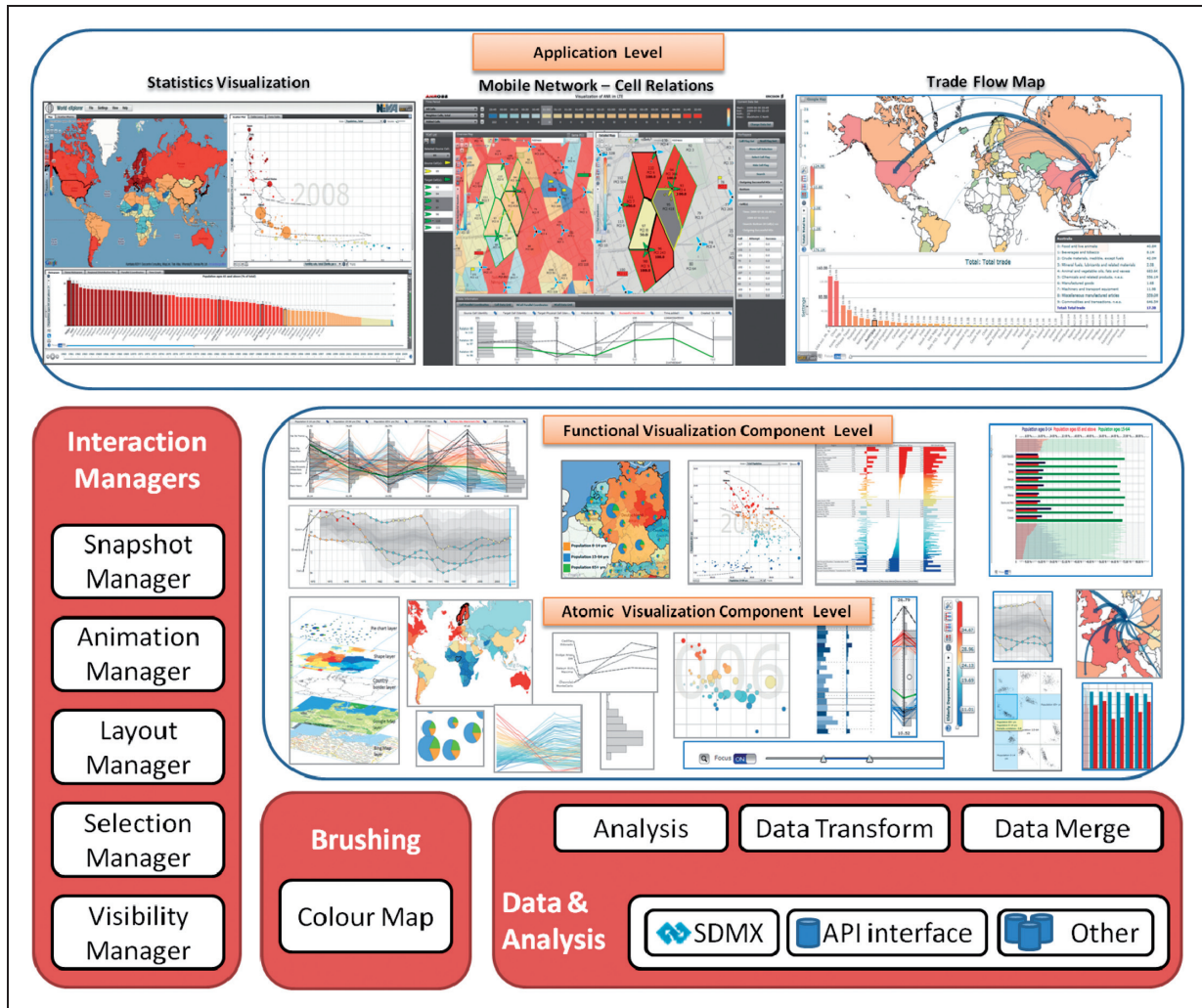


Figure 4. GAV Flash Framework.

application-level events such as menus and a module for the snapshot mechanism, which is described further in the ‘Storytelling’ section. The abstraction into interfaces also allows others to extend the framework with new functionality, be it new visualizations or data providers. They simply have to follow the framework definitions of how to access data and shared assets and then implement their own ideas.

Atomic and functional component architecture

The generalization of components coupled with advanced features can make it hard to encompass all data scenarios in a component. It could be faced with a large multivariate data set but also with a highly dense temporal set. These two types of data sets often require different solutions in terms of the data processing, the element drawing and also the end-user experience of the visualization. To facilitate this need

for dynamic components, we break them down into small blocks called atomic components. These atomics are used together to form a fully functional component, but they are not dependent on each other and so they can be combined in any way. This concept can take many forms depending on the parent component, the clearest example being how the map uses different layers to display different levels of data as presented below (Figure 6). The same type of concept is used in other components as well, although not as obviously as the map example.

The combination of a component base with one or several atomic parts forms a functional component. They are generally encapsulated together with the required GUI elements needed to control the visualization so that only the combined properties of all atomic parts are exposed to the surrounding application. The atomic parts allow the creation of custom functional components that can differ extensively

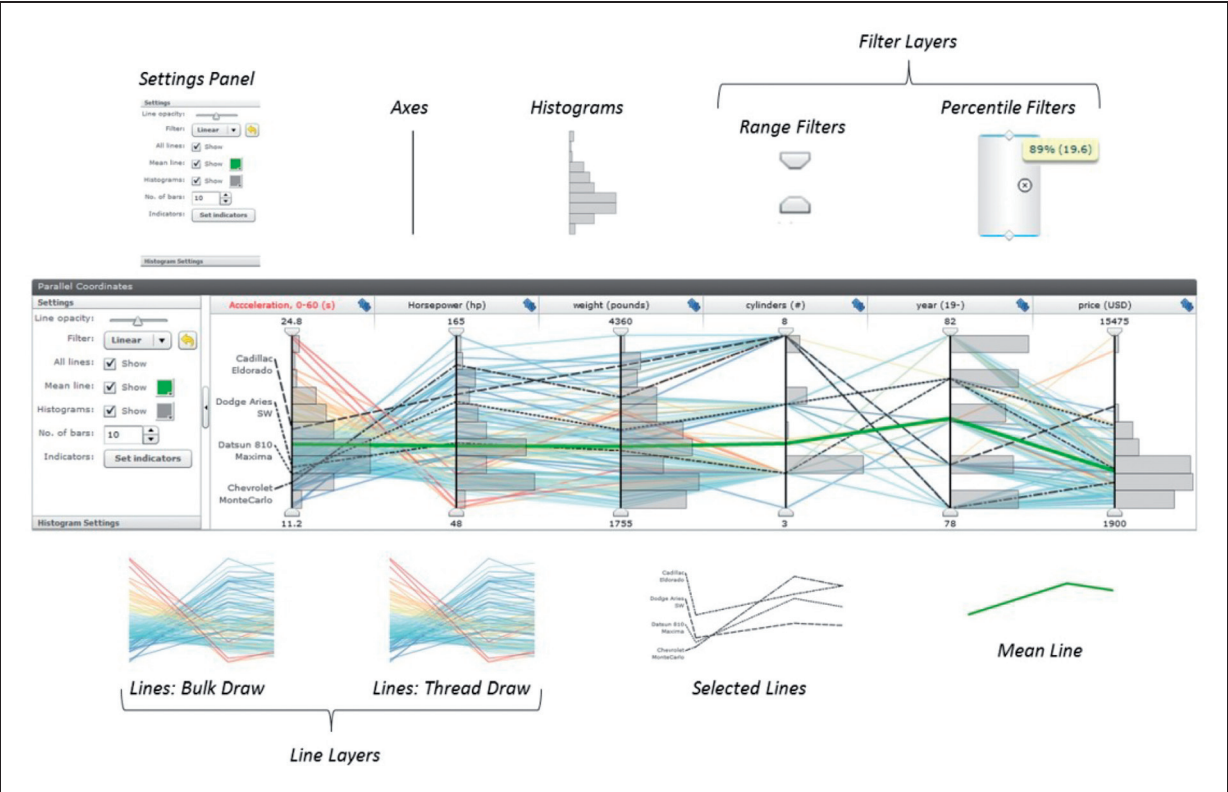


Figure 5. Functional PCP component and its layers.

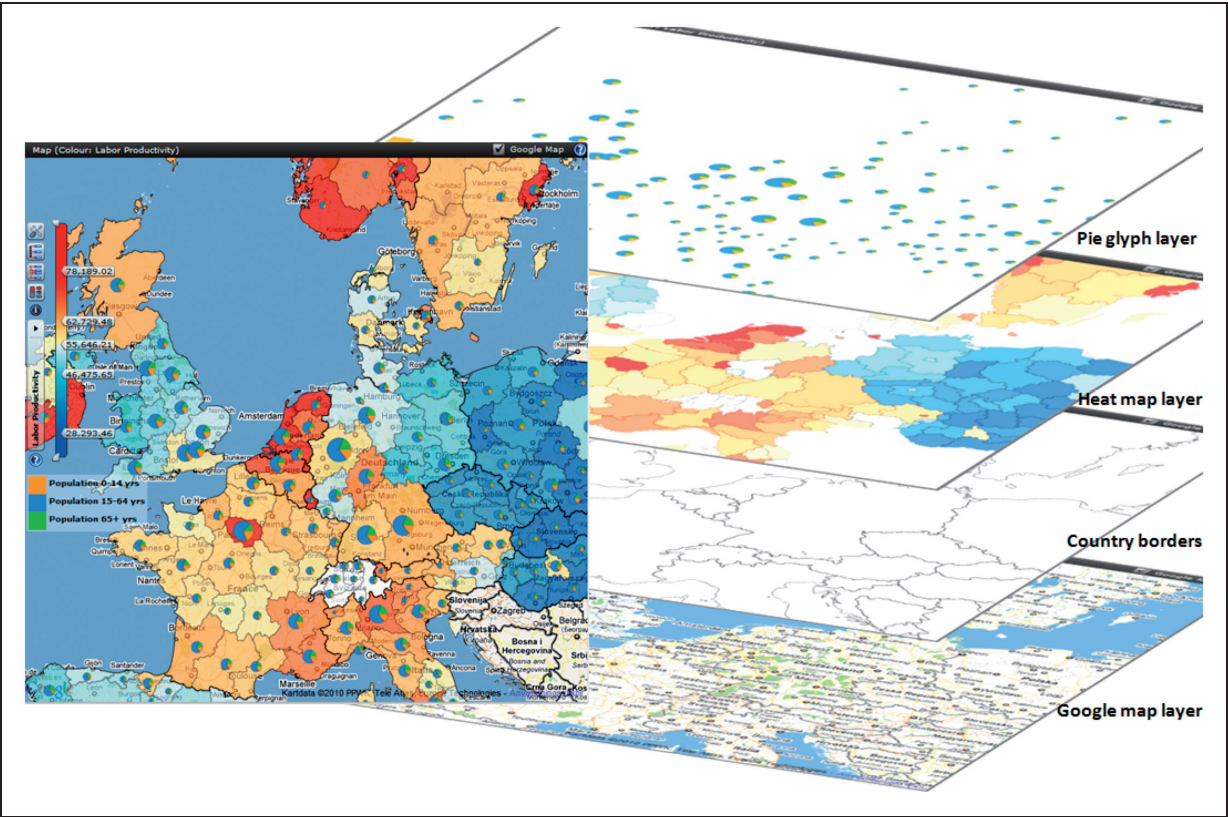


Figure 6. Choropleth map component and its layers.

depending on the end users' needs. A number of atomic parts can be reused in several components. For example, a circle glyph layer can be used in both the scatter plot and in the map, or a range filter can be used in both the PCP and the colour legend. The functional components can in turn be used by application developers and linked to each other through the use of linking components such as a selection manager, a visibility manager, and an animation controller to create quick prototypes and get a first look at their data. That first prototype can then determine which way the visualization needs to go and whether some kind of special atomic and/or functional component needs to be developed.

Two example functional components

In this section we will describe two functional components, a parallel coordinates plot and a choropleth map, to demonstrate the concept of building functional components from atomic components or layers. A demonstration of functional components can be seen at the toolkit website.²⁷

Parallel coordinates. The building blocks for a standard PCP are axes and lines. It is common to have the ability to filter along each axis and the possibility to see statistical analysis on the data. The GAV Flash PCP consists of a base plot model that keeps track of all common axis-related information, such as order, visibility, positions and orientation. This model is available to all atomic layers added to the plot so that they can draw their contribution in the correct coordinate system. Each layer works independently of each other, but they can all affect the common controls such as filter and selection lists by accessing the model. Figure 5 shows a number of typical layers of our PCP: axis layer, header layer, line layer, selected line layer, mean line layer, range filter layer, percentile filter layer and histogram layer. The header layer allows users to select which column is used for colouring lines. The range filter layer allows users to filter out items of no interest and then focus on items of interest. In addition to the layers mentioned above, the PCP also includes a setting panel which allows users to turn on or off layers optionally, adjust the opacity of lines or change the number of bins of histograms, which should be displayed. A demonstration of our PCP can be seen at <http://ncomva.se/flash/projects/Guide/?chapter=Visualizations§ion=Parallel%20Axes%20Chart>.²⁸

Choropleth map. The heart of our choropleth map is a map base class which contains no visual parts at all,

but controls everything needed to display something within its context. It creates transformations to deal with projections and keeps track of the user's input as he or she zooms or pans. The visuals are instead created in layers, for example a polygon layer for coloured regions and a glyph layer for region data. By making the map independent of its layers, we can combine them in any way we choose, or create new ones as long as they adhere to the basic principles of the map.

To support users working with geographic regions (e.g. identifying the geographical positions of cells in a mobile phone network as mentioned in 'RoadVis section') as well as exploiting the advantages of online maps, we built a background map layer which allows using Google maps or Bing maps as a background map. The users can switch between these two online maps or turn them off. Figure 6 shows a number of typical layers of our choropleth map: background map layer (Google map in this case), border layer, coloured region layer, glyph layer (a pie chart glyph layer in this case).

To avoid redrawing layers (which normally include a collection of geometry objects) every time the map is zoomed or panned, which is normally time consuming, we developed a special coordinate system and plot layers within this coordinate system so that every time the map is zoomed or panned, we calculate the transformation matrix and use the built-in transformation operations of Flash to transform map layers as images to the target display window. This process is much faster than redrawing the layers by common techniques and at the same time avoids a flickering effect. A demonstration of our choropleth map can be seen at <http://ncomva.se/flash/projects/Guide/?chapter=Visualizations§ion=Choropleth%20Map>.²⁹

Data model

GAV Flash uses a simple data set model as a base for storing and communicating data. It is designed to manage data in three dimensions, represented by attribute, space and time, and can communicate the boundaries and content of these dimensions to the visualizations. To make it independent of actual storage structure, we have used a static and simple interface for data access. This allows different storage structures to be implemented to serve for different purposes. We have implemented a storage structure for GAV Flash that simply is an array optimized for fast access and can handle large spatio-temporal, multivariate data sets.

This data model also facilitates the implementation of VA applications that typically have a pipeline architecture of three modules, which is similar to the

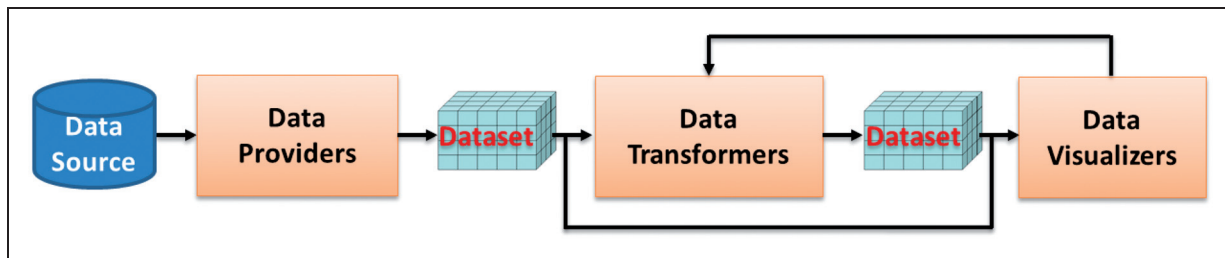


Figure 7. A typical pipeline architecture of VA applications.

pipeline of Card et al.,³⁰ as illustrated in Figure 7. In this architecture, data are first loaded from data sources into the data-provider module and then passed to the data-transformation module for analysis and/or processing before being passed to the visualization module for visualization. The transformation is optional, and an application can even combine visualizations that display the data both before and after transformation. Visualizations can be linked to control the transformation, thus giving the user direct access to how the data are manipulated. This model also allows developers to integrate their own components into our framework, for example, a new data-transformation component facilitating a needed data-analysis algorithm that first processes the data before visualization can be made. GAV Flash can then focus on visualization, leaving data analysis to the experts.

Custom data providers

The basic application uses a single source of data which is fed into components, the source commonly being a text or an XML-based file. This is a fine approach for prototype building and smaller studies, but when building for more complex systems we need to be able to extend the data provider to encompass more sources and types of data. A typical example is regional statistics where the user accesses a database for general data and combines it with their own data. The challenge for the developer here is to allow this kind of direct data feed, from any number of different types of data sources, while at the same time enabling the user to upload their own data and combine them together. The solution used in GAV is to create a collection of common data loaders, with a GUI and components to merge the data together into a single data set usable by the visualizations. If the developer needs to load specific file formats or access specific databases, they can easily extend the framework with their own data provider, which merges into the rest. Examples include csv and Excel readers, and REST

web-service consumers such as one for the World Databank.

Component linking

To facilitate linking components through selection, visibility, colouring and animation, we introduce the concept of program context. A program context contains a data set, a selection manager, a visibility manager, a colour map and an animation manager. The selection manager controls the list of selected items, called the selection list. The visibility manager controls the list of the filtering status of items, called the visibility list. The colour map controls the colouring scheme and the animation manager controls the change of time. Components sharing the same program context will share the same selection list, visibility list, colouring scheme and time step. Whenever a change happens in the program context, all components using this context will be informed about this change and each of them will respond to this change by its own way. For example, a click in a choropleth map will change the selection list and this change will be informed to a PCP and choropleth map itself, which share the same context. Then both the choropleth map and PCP will highlight the selected items by their own way.

Extendibility

To ensure that a specific VA task can be solved by the framework, the framework is designed so that components are extendable by developers. For example, developers can add new features or new layers into an existing component, rewrite a component to improve its performance or even replace an existing component by a new component. To make this possible, we have employed interface mechanism extensively in our design of components. Components are based only on interfaces that are necessary to make them work when connected to other components. Nevertheless, to make the implementation of components easier, the interfaces are kept as simple as possible. This means that they only contain definitions necessary for communicating

with the components. For example, in the data model above, data sets use a simple interface, called *IData set*, which only includes definitions of a basis function *getValue* and some extra functions such as *getMax*, *getMin*, *getMean* to supply necessary data to other components as well as to avoid recalculation of the same operation in different components.

Performance and interactivity

By choosing Adobe Flash as a programming platform for our toolkit, we face two major challenges, performance and interactivity, when dealing with large data sets and advanced components such as PCP and a choropleth map. There is no publication addressing these challenges, therefore in this section we will present our approaches to these challenges. For a PCP, the performance largely depends on the number of lines, the number of line segments of each line, line thickness, line opacity and the view size. For small data sets, of less than 2000 items over about 10 attributes, our approach is to represent a line by a Flash Shape object. This approach facilitates the implementation of line filtering and line colour/opacity transforms through employing the abilities of Flash Player to show or hide objects and transform their colour/opacity without redrawing. Nevertheless, for larger data sets the rendering time increases quickly. This is due to the pixel colour blending process taking a longer time as the number of lines through a single pixel increases. To speed up this process, we divide lines into groups, where each group includes a subset of lines and the number of lines in each group can be adjusted by users according to their hardware. Each group is then drawn into a Shape object and then cached as a bitmap.

We test this approach with a data set of 9020 Sweden zip codes,³¹ each with 13 attributes (population, purchase power, etc.). The window size for plotting lines is 1000×500 pixels. The test is done on a Dell Latitude E6410 (Dell (China) Company Limited) notebook with a graphics card of NVIDIA NVS 3100M and 8 GB of RAM. As mentioned above, there is no publication of PCP implemented in Adobe Flash with high performance, therefore we show here, in Table 1, the results of time and memory consumed of two approaches, the normal approach (or non-grouping approach) and our grouping approach, in a number of different configurations. Table 2 shows the rate of time consumed between the normal approach and our approach and the rate of memory consumed between our approach and the normal approach. It shows that the approach with 200 lines in a group is from 6 to 51 times faster than the non-grouping approach but uses twice as much

memory. It also shows that putting 100 lines in a group does not improve rendering time but takes one and half times as much memory as putting 200 lines in a group.

For choropleth maps, the performance depends on the number of regions, the number of polygons per region, the number of vertices of a polygon, the view size and the zoom level of the maps. We choose to represent a region by the aforementioned Flash Shape object, which, similar to the PCP case, facilitates the implementation of filtering, selection and colour/opacity transforms. This works well with normal maps of about 2000 regions, 5000 polygons and 250,000 vertices, and even with larger maps that include 9000 regions, 24,000 polygons and 1,100,000 vertices in total when we use our threading technique presented below. The map does not suffer from the colour-blending problem of the crossing PCP lines. Nevertheless, for even larger maps we are planning to use a tiled approach, common to web-map services like Google maps, and employ the shader technology recently supported by Flash Player to enhance the performance further.

The second challenge comes from the fact that Flash Player does not support multithreaded ActionScript, which makes the application unresponsive during the execution of tasks that sometimes can take 10 seconds, such as drawing a large number of items. Unresponsiveness for a long time can distract the user and make the exploration process of the user less efficient. To address this problem in a similar manner to Piringer et al.,³² we have implemented our own library to simulate a multithreading environment. Our approach is to assign each large task, for example drawing 10,000 lines, to a so-called microthread or pseudothread and to divide the large task into a number of smaller tasks, for example drawing 100 lines at a time, so that each small task can be executed in a reasonable time period. Then, small tasks of threads are scheduled into frames so that the total execution time of each frame is less than what the given frame rate allows, typically 30 ms per frame. After each frame, the control is returned to the Flash Player and allows the user to interact with the application. Threads can be cancelled early as a result of user interaction and new threads can be scheduled and executed to reflect user updates.

Although used to enhance interactivity during execution of large tasks, it is important to realize that our threads are, owing to the nature of Flash Player, still executed in a single ActionScript thread, and thus do not employ multiple core processors. A demonstration of the aforementioned techniques can be seen in our grid map demonstration application (Figure 8).³³

Table 1. The results of time and memory consumed of non-grouping approach and grouping approach in several different configurations

		6		9		12	
No. attributes rendered		Time (ms)	Memory (MB)	Time (ms)	Memory (MB)	Time (ms)	Memory (MB)
Line opacity = 1 Thickness = 1 pixel	Non-grouping approach	19039	40	113388	51	202060	64
	No. items in a group = 100	1910	241	2923	254	4176	289
	No. items in a group = 200	1687	110	2815	116	3986	127
Line opacity = 0.5 Thickness = 1 pixel	Non-grouping approach	20529	65	119222	65	228838	68
	No. items in a group = 100	3082	179	4616	191	6435	226
	No. items in a group = 200	3203	115	4690	119	6553	129

Table 2. Time rate and memory rate between the non-grouping approach and the grouping approach

		6		9		12	
No. attributes rendered		Time rate	Memory rate	Time rate	Memory rate	Time rate	Memory rate
Line opacity = 1 Thickness = 1 pixel	No. items in a group = 100	10	6	39	5	48	5
	No. items in a group = 200	11	3	40	2	51	2
Line opacity = 0.5 Thickness = 1 pixel	No. items in a group = 100	7	3	26	3	36	3
	No. items in a group = 200	6	2	25	2	35	2

Interaction using the focus and context approach

Interaction with large regional data sets and reducing the amount of data to zoom in on a smaller area in order to see details while still maintaining the overview was introduced by Shneiderman.³³ Overview gives the user an idea of where to find regions of interest and then analyse in more detail what is important to the user. This can help us to avoid the performance issue mentioned above. We support this feature in GAV Flash by implementing the concept of sub-data sets in GAV Flash. A sub-data set is a list of references to data items in a large data set. As a data item normally corresponds to a region in a map and vice versa, a sub-data set corresponds to a sub-area of the whole map and vice

versa. Then, each component works based on its sub-data set input. It visualizes only items in the sub-data set. If we wish to visualize the whole data set, the sub-data set will be set to the whole data set. Figure 8 shows an example of this approach applied to 8000 Italian municipality regions. An interactive region selection is applied to the context map (left view) to focus on an area of interest in the two right views which interact only with a reduced regional data set in this focus area for a better performance.

Examples of perceptual methods

Focus and context is only one example of an interactive method that takes advantage of our knowledge of human perception and cognition. Other examples of

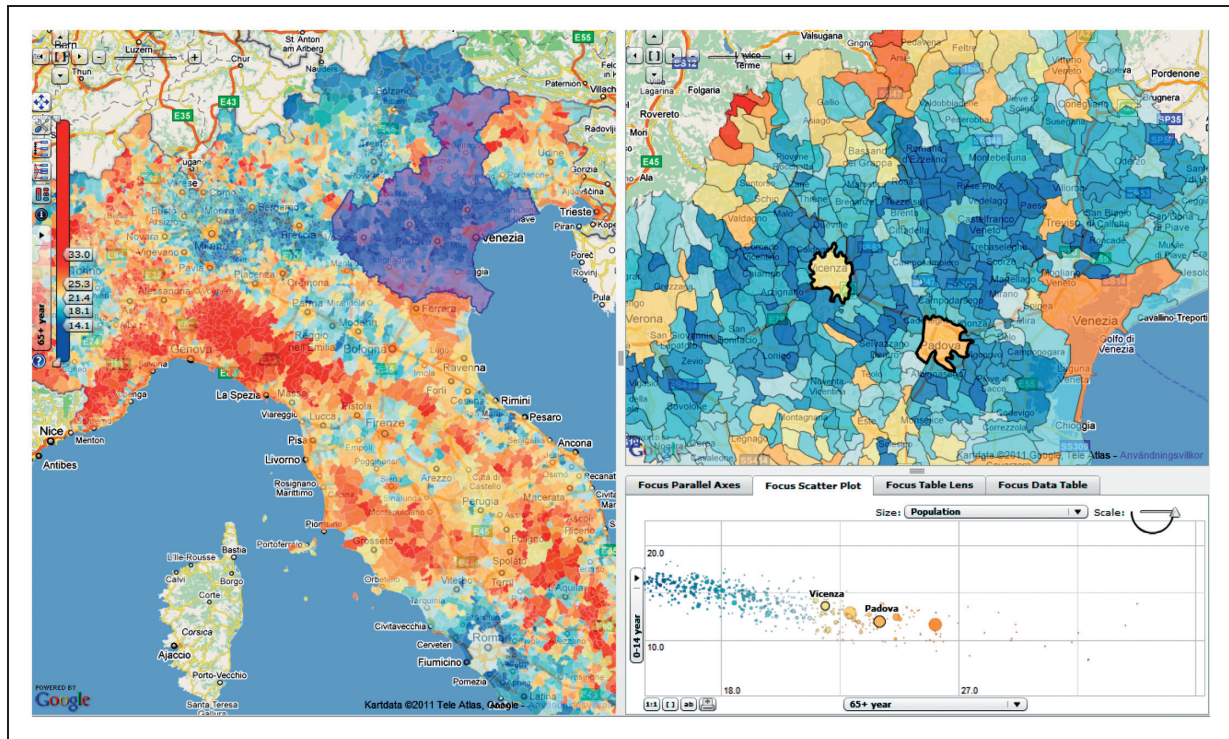


Figure 8. An illustration of focus & context approach applied to a large regional dataset of 8000 Italian municipality regions. The left view shows the context map, the top right view shows the focus map and the bottom right view shows only the items in the focus map (<http://vitagate.itn.liu.se/GAV/gridmap/Italy2008/>).

well-known perceptual methods supported by GAV Flash include dynamic filtering attached to the colour legend to see only relevant data, support for divergent colour scales to highlight regions with low or high values, moving bubbles with trace lines for temporal data (Figure 13), profiles in parallel coordinates to compare two or more regions for multiple indicators over time (Figure 14), geographical map layers to identify location of interest and focus and context (Figure 8).

Storytelling

GAV Flash is not only a medium for presentation, but it also supports storytelling in which visualizations serve not only as an exploration and discovery tool but also as a means to collaborate stories among users fostering a social style around visual analytics. Storytelling^{25,34} is achieved through a mechanism in GAV Flash (Figures 9 and 10) that supports the storage of interactive events in an analytical reasoning process through ‘memorized interactive visualization views’ or ‘snapshots’ that can be captured at any time during an explorative analytics process and becomes an important task of a web-based storytelling authoring analytical reasoning process where discoveries can be linked into any online document using

hyperlinks and applied in combination of the enormous amount of collaboration infrastructure available on the web. Figure 9 shows an integrated storytelling process developed with GAV Flash and applied to national world statistics and based around importing data, authoring tool, snapshots and metadata, and finally publishing interactive educational documents for public statistics.

An example of the integrated storytelling process based on importing data, authoring tool, snapshots, metadata and publishing interactive documents.

Snapshot

When exploring and making sense of, for example, comprehensive statistics data, we need a coherent cognitive workspace to hang our discoveries on for organizing and navigating our thoughts. The GAV Flash toolkit includes such means by capturing, saving and packaging the results of an exploration ‘gain insight’ process in a series of ‘snapshots’ that could help the analyst to highlight views of particular interest and subsequently guide other analysts to follow important discoveries. The snapshot tool creates a single or a continuous series (story) of visualization captures during the exploration process. In a typical scenario, the analyst has selected relevant attributes, time step

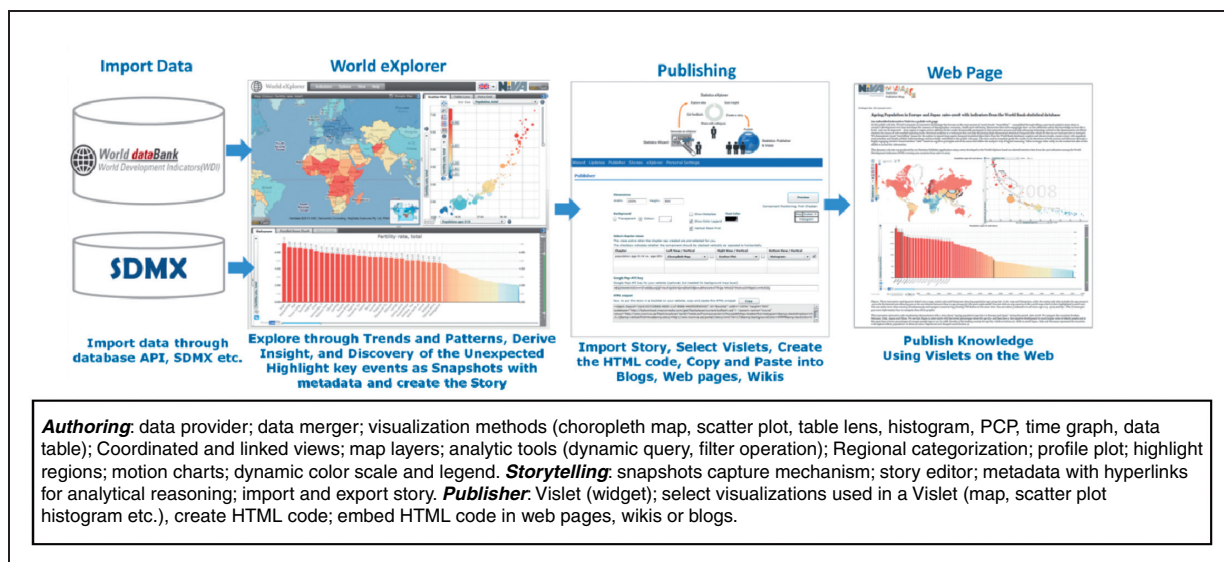


Figure 9. An example of the integrated storytelling process based on importing data, authoring tool, snapshots, metadata and publishing interactive documents.

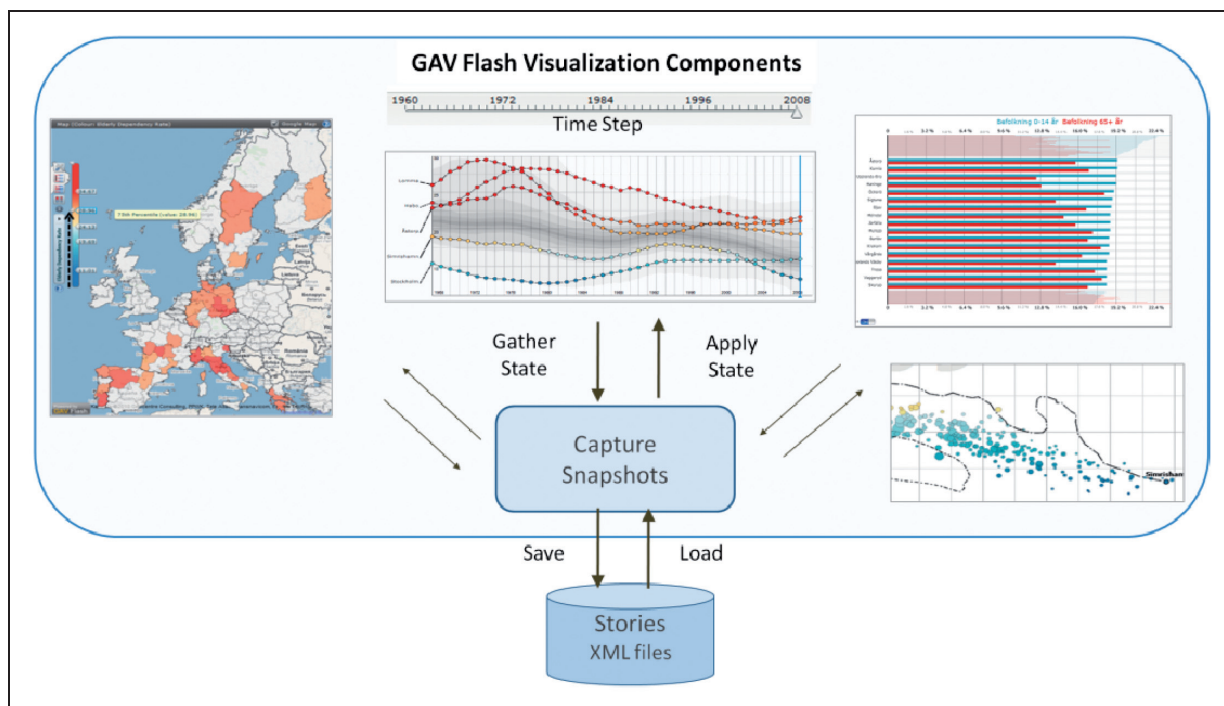


Figure 10. The snapshot system scans through all active views to gather and apply states.

(temporal data), data items of interest, colour class values and filter conditions for selected attributes and finally highlights the 'discovery' from a certain angle (viewing properties).

The analyst requests a snapshot with the Capture function that results in a snapshot class operation scanning through all its connected GAV Flash components

for properties to be captured. Each of these properties will then be parsed into XML and written to a file that also contains details on which data and variables were used and a unique name for each component. When a snapshot is activated, the saved state of the Snapshot class will be read from the XML file and parse its nodes back into component properties again. The previously

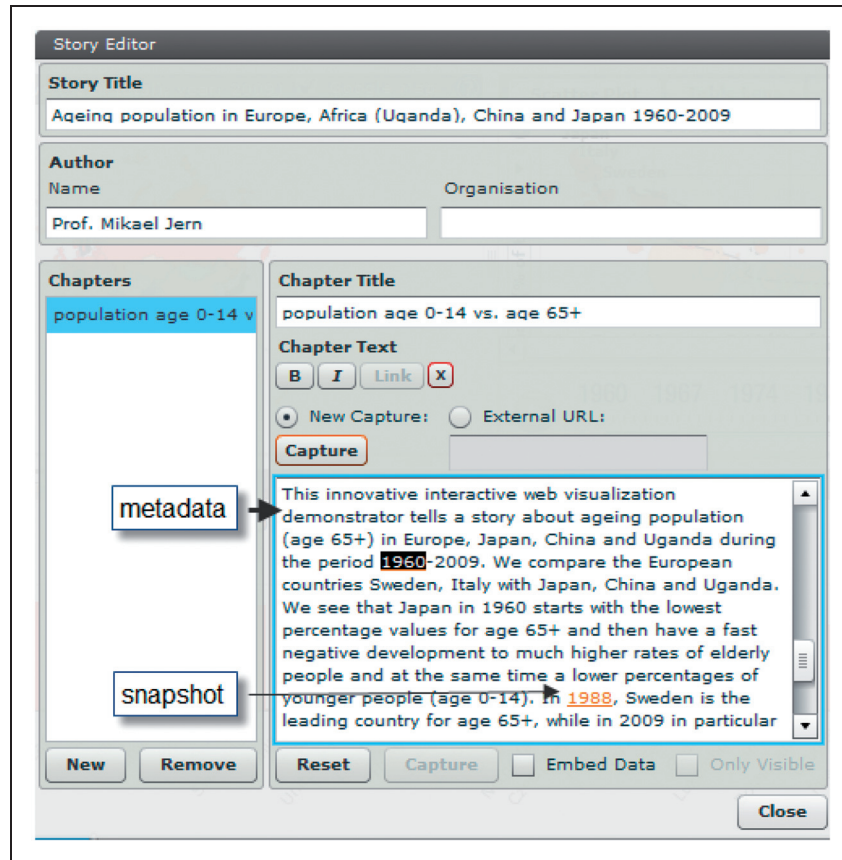


Figure 11. An illustration of the story editor of GAV Flash.

marked properties will then be applied and set the state of the application.

Storytelling

Storytelling, in our context, is about telling a story on the subject of explored data and related analytics reasoning about how knowledge was gained. Snapshots that instantiate a GAV Flash state (Figure 10) are a central feature of our storytelling mechanism together with associated descriptive text that could guide the reader in the analyst's way of thinking. The author creates a single or a discrete series of captures during the explorative process by electing relevant indicators, regions of interest, colour schema and filter conditions focusing on the data of interest or a time step for temporal statistics.

Hypertext, meaning 'more than just text', provides a richer functionality than simple metatext by allowing the reader to click on key words and learn about topics in the story. A story hyperlink is here a reference in the story metatext that links to an external URL website or a captured snapshot. A hyperlink in the metatext can be related to a snapshot or an external URL.

Before the actual capture is done, the user navigates, for example, the map view to a particular country, selects indicators for the scatter plot and selects a time step. A new view, such as PCP, can be added to the story. A 'capture' is made and all preferred states are saved. When the story is read later, hyperlinks can be initiated and the application will display the state-of-the-art snapshots.

Hyperlinks that instantiate snapshots and are associative descriptive text represent a central feature in our storytelling mechanism (Figure 11). These could guide the reader in the analyst's way of thinking. While it is true that a picture is often worth a thousand words, sometimes a few words and a snapshot provide the difference between a pretty picture and understanding. This focus on publishing through assisted content creation with emphasis on visualization and metadata represents a novel advantage of our storytelling.

Publish geovisual analytics results

A Vislet is a standalone Flash application (widget) assembled from low-level GAV Flash components in

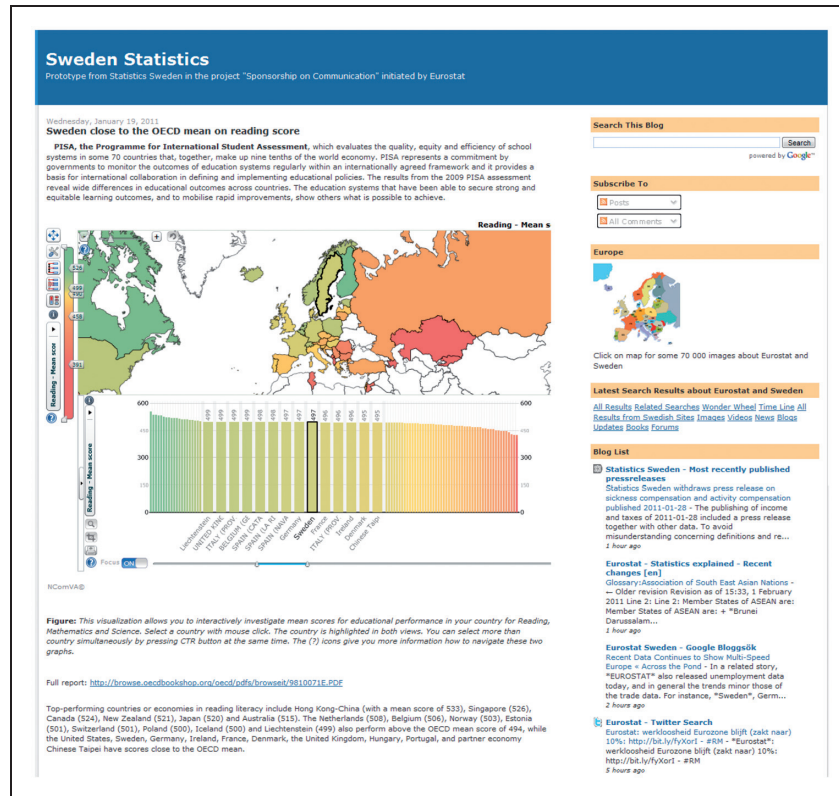


Figure 12. An example of a Vislet embedded in a web page (blog) (see <http://swedeneurostat.blogspot.com/>).

a class library and Adobe Flex GUI tools and is represented by, for example, a single map view or a composite time-linked map and scatter-plot view (Figure 12). A Vislet facilitates the transition of selected tedious statistics data into heterogeneous and communicative sense-making news entities with integrated metadata and dynamic embedded animated visualization that could engage the user.

An example of a Vislet embedded in a webpage (blog) (see <http://swedeneurostat.blogspot.com/>).

Publisher is the server tool that imports a story (Figure 9) and generates the HTML code that represents the Vislet and metadata. First, the user selects an appropriate visual representation for the Vislet, for example a map, scatter plot, parallel axes, table lens or time graph. Then, the size of the Vislet window with metadata is set, and, finally, Publisher generates the HTML code. This code snippet can then be copied and pasted into any webpage system, such as a blog. The Vislet can now be opened in the reader's Web browser and dynamically communicate the story. A Publisher server maintains the Vislet flash (swf) files together with a story repository, statistical data and regional shape maps. The Vislets run locally in the client's Flash Player and can thus achieve dynamic interactive performance.

Interactive features in a Vislet are exposed to all visualizations including tooltips, brushing, highlight, dynamic filtering which can discover outliers and dynamic multiple-linked views. Several specialist colour legend tasks are supported, for example showing outliers based on the 5th and 95th percentiles in certain colours or dynamic sliders that control, for example, class values.

Storytelling with snapshots can support schemes such as:

- communication of knowledge and insights;
- supporting learning and education activities;
- transfer of insights between individuals;
- communication of experience, values, expectations, opinions, predictions, etc.;
- making knowledge visible to improve process how knowledge can be identified; and
- supporting decision making through collaborative graphics annotation.

Case studies

In this section we present three of our case studies to demonstrate how GAV Flash is used in the

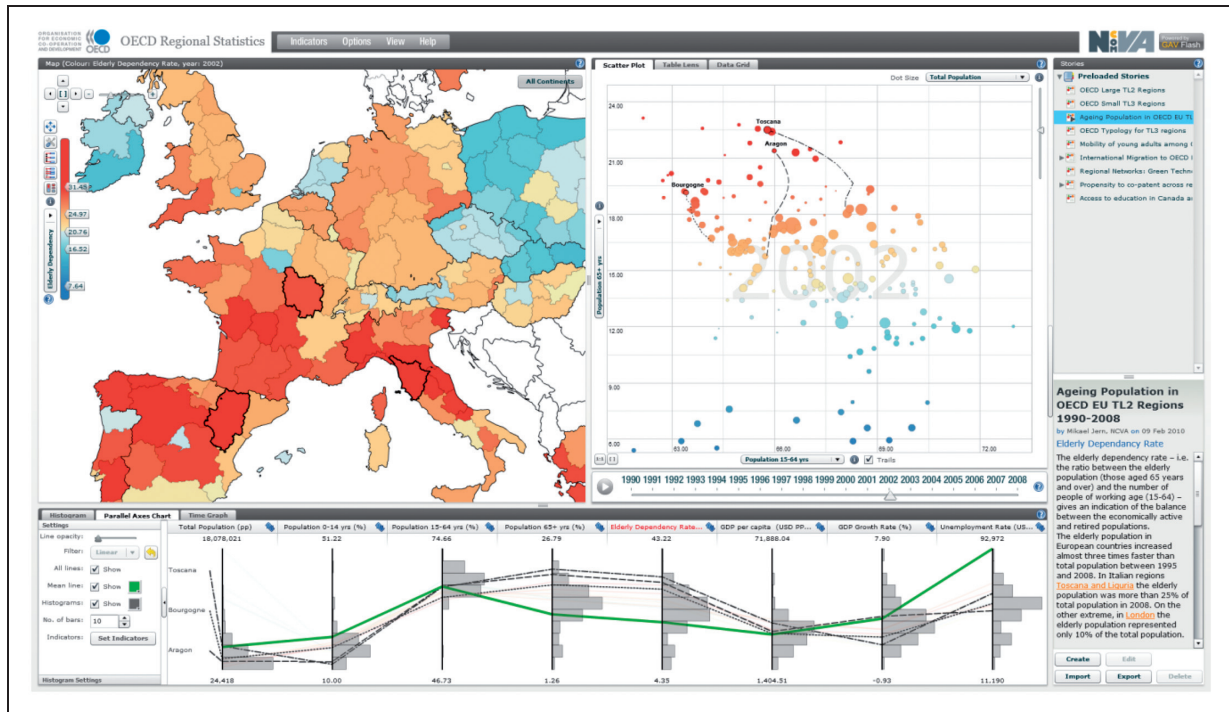


Figure 13. The OECD eXplorer layout.

development of visual analytics applications. These case studies have been conducted in collaboration with our partners within NCVA to produce applications used by our partners within statistics data visualization, self-organizing cellular radio network analysis and emergency scenarios.

OECD eXplorer

In Nov 2008, 'OECD eXplorer' was introduced (<http://stats.oecd.org/OECDregionalstatistics/>), developed with GAV Flash in collaboration with OECD.

It introduced common information visualization motion tools to the statistics community, such as PCP, scatter plots, table lenses and time graphs, to illustrate complex statistical data. OECD explorer's user interface is divided into three main dynamic, linked views (Figure 13): a choropleth map (top left view), a scatter plot (top right view) and a PCP (bottom view). These views are separated by interactive splitters, allowing the user to scale the sizes of individual views. They are all linked to each other and can be animated through a time-animation controller.

The scatter plot allows users to analyse the relationship between two indicators and, when in animation mode, it allows users to see how the indicator values change over time through trace lines (Figure 13, top right view).

The PCP is another component which is introduced to the statisticians to support a number of tasks, for example to analyse the relationships between indicators and to see a profile for selected regions. Each region is represented by a string passing through the parallel axes where each axis represents a single indicator. Differences between highlighted regions can be found by visually comparing the profiles representing them, a highly appreciated method to the statistics community.

The PCP has been extended with special features that are important to statistical exploration, such as histograms and filter operations based on percentile statistics. Histograms attached to each axis are used to visualize the distribution of indicator data, splitting the axes into a user-defined number of equally high rectangular areas (bins). The width of a rectangle indicates the frequency of regions intersecting that bin. Statistical filter methods based both on ranges chosen by the user and on percentile calculations are embedded in the PCP attached to an indicator. Figure 14 shows only regions which fulfil the condition below the 10th percentile and above the 90th percentile, for example only regions representing outliers are displayed. Two regions, Liguria and Paris, that comply with these conditions are highlighted.

This case study also demonstrates the storytelling process from creating a story with snapshots and meta-text, saving the story and, finally, using Publisher to

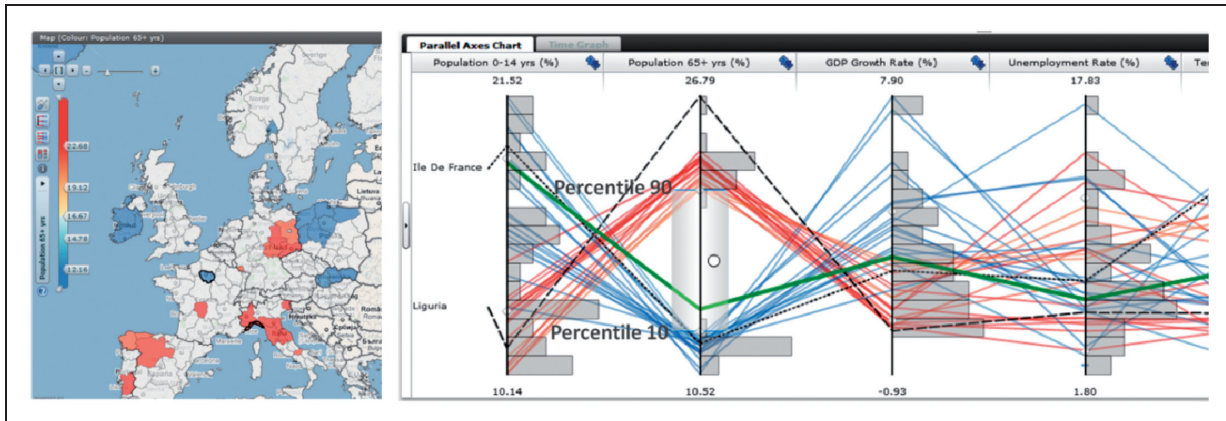


Figure 14. Regions below 10th percentile and above 90th percentile (bottom).

load the story and generate the HTML code that can be placed in a website. A video showing the whole process can be found at <http://vitagate.itn.liu.se/GAV/videos/VDA>³⁵ and examples of Vislet can be evaluated at <http://swedeneurostat.blogspot.com/> and <http://swedeneurostat.blogspot.com/2010/03/education-in-eu-example-with-vislets.html>.³⁶

Reviews from eXplorer partners who have evaluated the platform and available tools highlight the following eXplorer characteristics as the most effective:

- It enables the statisticians to (1) simultaneously analyse relations among different indicators, (2) explore trends over time and for different geographical boundaries and (3) use different map layers for better locating places. It provides functions for analysing data and benchmarking regions, presenting stories about the statistics and combining metadata and maps status.
- It captures the complexity of multidimensional regional data through dynamic multiple and time-link views.
- The structure of eXplorer encourages collaboration between statistics analysts and users of statistics.
- It encompasses data visualization and the possibility to capture, save and open discoveries (snapshots) with attached analytics reasoning metadata, for example storytelling, which can better support a more educational use of official statistics.
- The publishing approach (through Vislets) is regarded as very attractive to the general public as it does not require IT expertise. This publishing technique may become a strategic tool for news media to publish statistics news on the web.
- It is today a worldwide-recognized tool for better understanding the socio-economic structure of OECD regions and their performance over time.

ANROSS

ANROSS, a new version of VoSON,³⁷ is a visual analytics prototype developed in close collaboration with domain experts from the Swedish telecom company Ericsson to show how ANR³⁸ works. ANR is an algorithm invented by Ericsson to automatically detect and resolve physical cell identity (PCI) conflicts; therefore, it can self-configure cellular radio networks that are typically mobile phone networks. To show how ANR works, it is important to visualize and analyse the network data that are output by the process in which ANR configures a network.

The network in this context mainly includes two kinds of objects: cells and cell relations. A cell is a device that covers a geographical area and serves mobile phones in its area as well as hand over phone calls to other cells when necessary. Each cell has two identifiers: a globally unique cell identifier (CGI) and a PCI. CGIs are unique and constant over time but are more difficult and time consuming to detect for mobile terminals. In contrast, PCIs are easier to detect, but they are not unique as there are, at most, 504 different PCIs in LTE.³⁹ The PCI of a cell can vary over time as it needs to be unique in a region and may need to change. A cell relation is a neighbour relation between two neighbour cells. Cell relations are used to hand over phone calls from one cell (source cell) to a neighbour cell (target cell).

The network data in this context mainly include three kinds: cell data, relation data and event data. All of them are time varying and recorded for each time period, which normally is 15 minutes or 1 hour. Cell data include cell position, cell coverage area, cell PCI, cell CGI, number of call drops and number of outgoing/incoming handover successes or failures. Relation data include the number of handover successes/failures. Event data include cell added/removed,

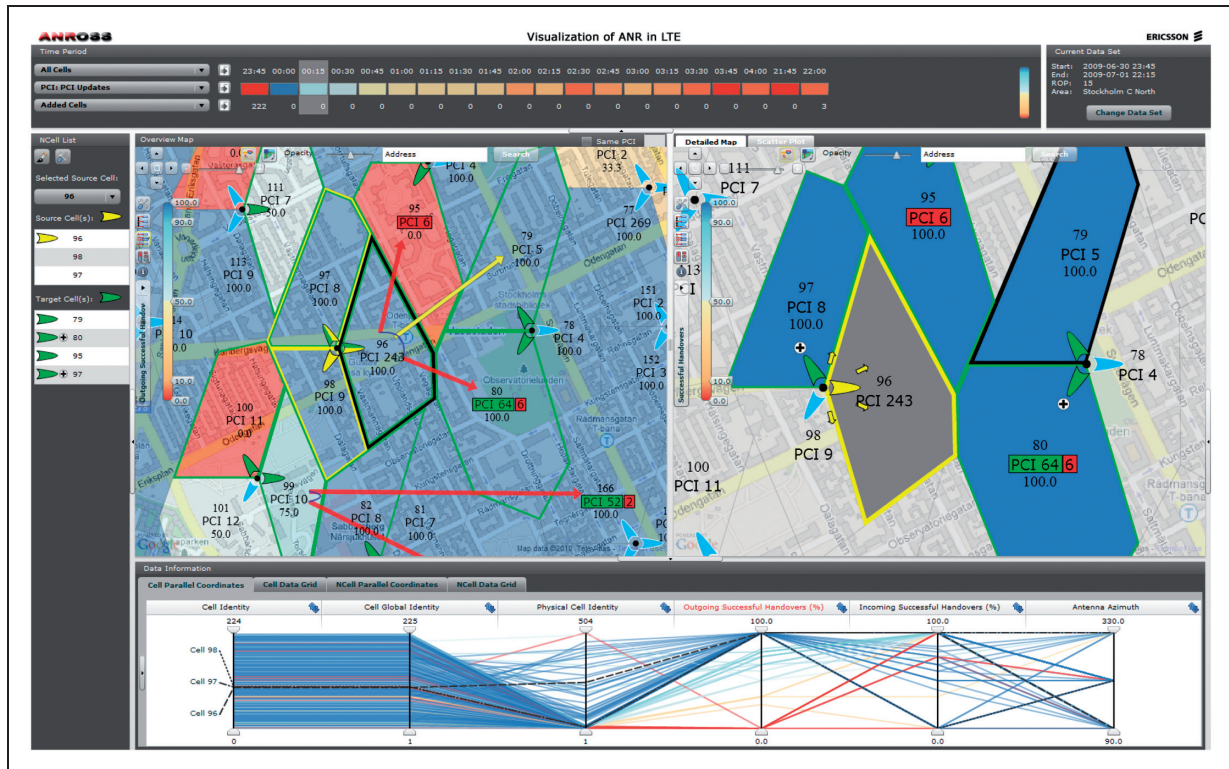


Figure 15. ANROSS layout including multiple linked and interactive views.

relation added/removed and PCI conflict detected/resolved.

To visualize all three kinds of data, ANROSS includes multiple-linked and interactive views (Figure 15): (1) an advanced time slider (top view), which displays aggregation data of each time period such as number of cells added, number of cells removed, number of relations added, number of relations removed, number of PCI conflicts detected, number of PCI conflicts resolved; through using a carefully designed colour map it highlights time periods in which some indicator has an extreme value; (2) a cell map (middle left view) presenting cell data and cell event data in a time period such as number of call drops, cell PCIs, cell PCI conflicts, cell PCI resolutions through map layers and glyph layers; (3) a relation map (middle right view) presenting relation data and relation event data in a time period such as number of handover failures of each relation, relations added and relations selected also through map layers and glyph layers; both these maps use online map services (Google maps and Bing maps) as background maps to support users in more easily identifying the geographical position of cells as well as performing analyses relating to terrain; for example, they can help users to identify objects between two cells (e.g. a lake or a high building), which can affect

the performance of the relation between these two cells; (4) two PCPs (bottom view), one visualizing cell data and one visualizing relation data; (5) two data tables (or data grids) (bottom view), one displaying cell data and one displaying relation data; (6) a data-selection panel (left) highlighting cells and relations being selected in cell map and relation map; (7) a search panel (right, hidden) which allows finding low/high performance cells or relations, which then will be highlighted in the cell map or relation map, respectively. The views support various interactive features such as filtering, selection, brushing, zooming and panning. When one or many cells or relations are selected in one view, other views will be updated to reflect this change. For example, when one cell is selected in the PCP for cells, the cell map will automatically zoom on the selected cell. At the same time, the relation map will show all relations of this cell. Through these linked and interactive views, ANROSS allows users to see how the network evolves from different initial configurations under the control of ANR and to answer various questions, such as:

- Why does a cell change its PCI, how is a PCI conflict detected and what is the difference in the performance of a cell (e.g. number of call drops) before and after a PCI change?

- Why is a relation added or removed, and what is the performance (e.g. number of handover failures) of a relation?

In addition, through using the filtering ability of PCPs, the searching tool and the linked views, ANROSS allows users to find and supervise ‘problem’ cells and relations such as cells/relations having a large number of call drops/handover failures, or cells having a potential PCI conflict in future.³⁷ To develop ANROSS we have employed many existing components in the GAV Flash framework, such as choropleth maps, a colour legend, PCP, a data grid, a visibility manager and a selection manager, and we have also added new layers and new components to the GAV Flash framework such as cell glyph layers, cell-relation glyph layers and an advanced time slider component.

By employing the GAV Flash framework, we have shortened the prototyping process. Using existing GAV Flash components is straightforward and quick. The most time is spent on collecting data, discussing and analysing the problems, determining the functionalities of the application, designing the interface, building new layers and components for ANROSS and testing the application.

ANROSS has been evaluated by network experts from Ericsson and has also been shown to Telia, one of the biggest network operators in Sweden. The overall feedback from both the network experts and the operators is very positive. They like the time slider, the two maps and the ability to see how the network changes over time. The ability to find ‘problem’ cells and relations were also considered as highly positive ‘...it looks useful and very good to me’. As a result, they would like to collaborate to develop this prototype into a product. Two videos showing the layout and functionality of ANROSS, as well as how it is used to present ANR, can be found at <http://vitagate.itn.liu.se/GAV/videos/VDA>.³⁵

RoadVis

RoadVis is another application developed using the GAV Flash framework that is used to analyse and make decisions, often in time-critical situations, on the large and ever-increasing amounts of time-varying and geospatial digital weather information related to emergency scenarios. It has been developed and customized in close collaboration with domain experts from the Swedish Meteorological and Hydrological Institute (SMHI) and the Swedish road administration and is today used on a daily basis for analysing and communicating information about road weather

conditions, particularly during the Swedish winter months.

RoadVis consists of a multiple-linked visual user interface consisting of a collection of dynamic maps, weather and information visualization methods that enable the user to simultaneously analyse relations among several different attributes with the aim of augmenting analyst and decision-maker capabilities to assimilate complex situations and reach important knowledge. The data visualized by RoadVis comprise weather observations from 770 automated observation stations around Sweden collected every 30 minutes, as well as a long 24-hour forecast and a short 6-hour forecast for each station that is made every hour. Using threading and dynamic queries, the application is constantly updated with the latest data in the background and the analyst can continue to use the application during their vital work in keeping the roads safe.

The visual representation for RoadVis, as shown in Figure 16, is based on a map of Sweden, where each individual station is marked and coloured depending on requested attribute value. The user can select relevant road condition, such as ice slipperiness, frost, condensation or temperature. RoadVis then calculates both present and forecasted scenarios for the coming 6 hours as well as the next 24 hours using the short and the long forecast. The PCP is used to view, analyse and make visual inquiries for the multivariate weather data. Relations and trends are observed and the analyst can also filter data and discover areas with a higher risk for accidents unless preventative methods are taken immediately. Meteorological diagrams visualize both the observed values and the forecasted weather for the selected stations. A demonstration video of RoadVis can be found at <http://vitagate.itn.liu.se/GAV/videos/VDA>.³⁵

RoadVis uses a multilinked visual user interface, where a station, represented by a circle, is coloured according to temperature. Meteorological diagrams are shown for two selected stations in the map. A parallel coordinates plot makes dynamic enquiries and filters the multivariate weather data. A time slider controls time steps for the time-linked views.

Conclusion and user feedback

The authors expect that the three case studies introduced, which are now in full operation, will enhance the use and understanding of geovisual analytics applied to spatio-temporal and multivariate data. It will enable the analyst and operator to take a more active role in the discovery process of exploring. The tool will increase the interest in and knowledge of structures and development patterns among specialist as well as non-specialist users. Feedback from domain

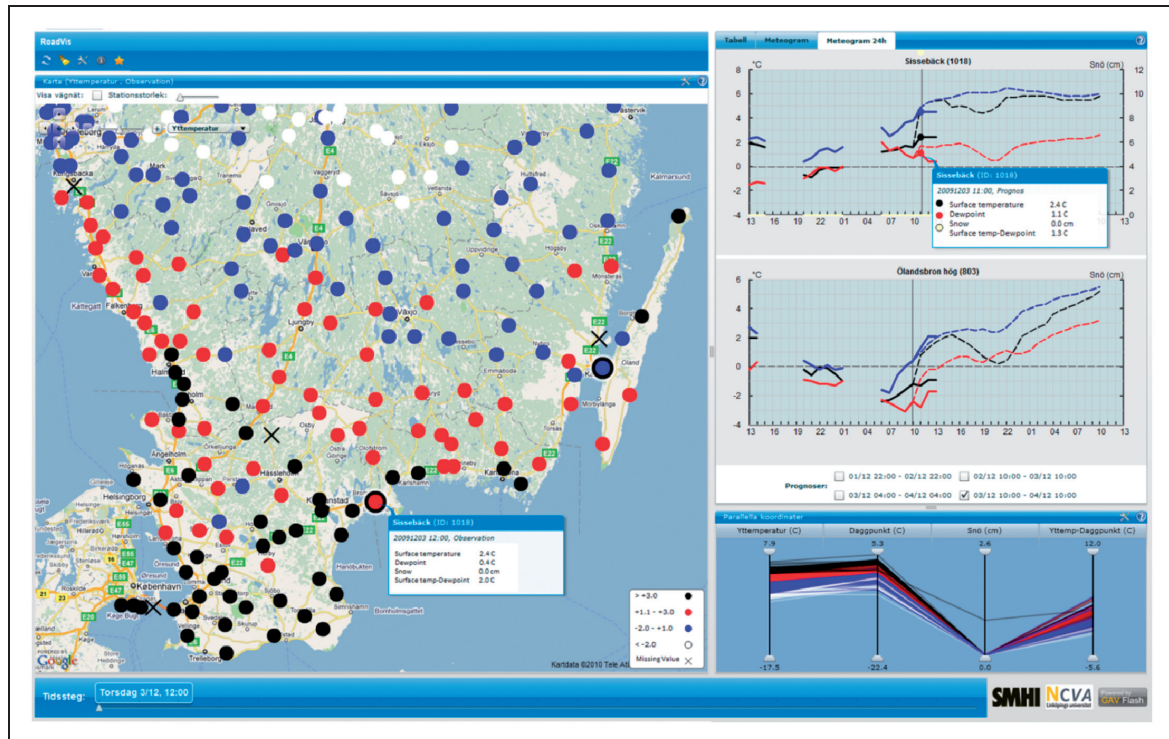


Figure 16. RoadVis multi-linked visual user interface, where a station, represented by a circle, is here colored according to temperature. Meteorological diagrams are shown for two selected stations in the map. A parallel coordinates plot makes dynamic inquiries and filter the multivariate weather data. A time slider controls time steps for the time-linked views.

experts from Ericsson, SMHI and OECD, who have started using the tool, shows that a sense of analytical reasoning and speed-of-thought interaction is achieved. Here are some features highlighted by our partners:

- a component-based geovisual analytics toolkit programmed in ActionScript for Web access;
- ability to have dynamic time-link views and see the multidimensionality of complex data;
- IT expertise is not required to publish interactive visualization embedded in blogs and wikis;
- Flash implementation of practical information visualization methods such as PCPs, table lenses and time graphs;
- PCP with embedded fundamental statistics based on dynamic percentile inquiry and filtering and attached histograms; and
- an architecture facilitating the analyst to explore data and simultaneously save important snapshots of discoveries or create a continuous story of snapshots (storytelling) to be communicated and shared with team or public users.

At the same time, GAV Flash will encourage the practical use of collaborative geovisual analytics

through an integrated exploration, collaboration and publication process addressing editorial storytelling aimed at producing geovisual analytics-related news content in support of an automatic authoring process. The author simply presses a button to publish gained knowledge that clearly visualizes the result of an explorative data-analysis process. It is a storytelling technology with the goal of advancing research critical to, for example, official statistical production based on Vislets – embedded dynamic visualization with the analytics sense-making metadata joined together and publishable in HTML webpages such as blogs and wikis.

Acknowledgements

The applied research case studies were carried out in close collaboration with OECD, SMHI and Ericsson Research, who supplied data and comprehensive evaluation of the application. The research is in part supported by funding from the Swedish Knowledge Foundation and the Swedish Agency for Innovation Systems (VINNOVA). The authors thank their colleague Markus Johnsson.

References

1. Andrienko G and Andrienko N. Visual exploration of spatial distribution of temporal behaviors. *Proceedings of the 9th*

- International Conference on Information Visualisation (IV05)*; 2005 Jul 6–8; London, England. Washington DC: IEEE Computer Society; pp. 799–806.
2. Muller W and Schumann H. Visualization methods for time-dependent data – an overview. *Proceedings of the 2003 Winter Simulation Conference*; 2003 Dec 7–10; New Orleans, Louisiana, USA. New York: Association for Computing Machinery; pp. 737–745.
 3. Guo D, Chen J, MacEachren AM and Liao K. A visualization system for space–time and multivariate patterns (VIS-STAMP). *IEEE Trans Vis Comput Graph* 2006; 12(6): 1461–1474.
 4. Jern M, Åström T and Johansson S. Geoanalytics tools applied to large geospatial data sets. *Proceedings of the 12th International Conference Information Visualisation (IV08)*; 2008 Jul 9–11; London, England. Washington DC: IEEE Computer Society; pp. 362–372.
 5. Gapminder, <http://www.gapminder.org/>
 6. OECD eXplorer, <http://www.oecd.org/gov/regional/statistics/indicators/explorer>.
 7. Andrienko N and Andrienko G. Interactive visual tools to explore spatio-temporal variation. *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI 2004)*; 2004 May 25–28; Lecce, Italy. New York: Association for Computing Machinery; pp. 417–420.
 8. Dykes J, MacEachren AM and Kraak MJ. *Exploring Geovisualization*. Oxford: Pergamon Press, 2005, pp. 159–180.
 9. Fekete J and Plaisant C. The InfoVis toolkit. *Proceedings of the 10th IEEE Symposium in Information Visualization (InfoVis '04)*; 2004 Oct 10–12; Austin, Texas, USA. Washington DC: IEEE Computer Society; pp. 167–174.
 10. Voss H, Andrienko N and Andrienko G. CommonGIS – common access to geographically referenced data. *ERCIM News* 2000; 41: 44–46.
 11. GeoVista Studio, <http://www.geovistastudio.psu.edu>.
 12. Tominski C, Abello J and Schumann H. CGV – An interactive graph visualization system. *Comput Graph* 2009; 33(6): 660–678.
 13. Bostock M and Heer J. Protovis: A graphical toolkit for visualization. *IEEE Trans Vis Comput Graph* 2009; 15(6): 1121–1128.
 14. Protovis, <http://vis.stanford.edu/protovis/>
 15. Flare, <http://flare.prefuse.org/>
 16. Tableau, <http://www.tableausoftware.com/>
 17. Stolte C, Tang D and Hanrahan P. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *IEEE Trans Vis Comput Graph* 2002; 8(1): 52–65.
 18. Shrinivasan YB and Wijk JJ. Supporting the analytical reasoning process in information visualization. *Proceedings of the 26th Annual SIGCHI Conference on Human Factors in Computing Systems*; 2008 Apr 5–10; Florence, Italy. New York: Association for Computing Machinery; pp. 1237–1246.
 19. MacEachren AM, Brewer I and Steiner E. Geovisualization to mediate collaborative work: Tools to support different-place knowledge construction and decision-making. *Proceedings of the 20th International Cartographic Conference*; 2001 Aug 6–10; Beijing, China. Beijing: Chinese Society of Geodesy Photogrammetry and Cartography; pp. 6–10.
 20. Jern M. Smart documents for web-enabled collaboration. In: Vince JA and Earnshaw RA (eds) *Digital Content Creation*. New York: Springer-Verlag, 2001, pp. 140–162.
 21. Carr DB, White D and MacEachren AM. Conditioned choropleth maps and hypothesis generation. *Annals of the Association of American Geographers* 2005; 95(1): 32–53.
 22. Robinson A. Re-visualization: Interactive visualization of the progress of visual analysis. *Proceedings of GIScience Workshop Visual Analytics & Spatial Decision Support*; 2006 Sep 20; Muenster, Germany; pp. 1–21.
 23. Keel P. Collaborative visual analytics: Inferring from the spatial organisation and collaborative use of information. *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST)*; 2006 Oct 31 – Nov 2; Baltimore, Maryland, USA. Washington DC: IEEE Computer Society; pp. 137–144.
 24. Viégas FB, Wattenberg M, Ham FV, Kriss J and McKeon M. Many eyes: A site for visualization at Internet scale. *IEEE Trans Vis Comput Graph* 2007; 13(6): 1121–1128.
 25. Jern M, Rogstadius J, Åström T and Ynnerman A. Visual analytics presentation tools applied in HTML documents. *Proceedings of the 12th International Conference Information Visualisation (IV08)*; 2008 Jul 9–11; London, England. Washington DC: IEEE Computer Society; pp. 200–207.
 26. Thomas J and Cook K. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. Chicago, IL: IEEE Computer Society Press, 2005, p. 13.
 27. GAV Flash functional component guide, <http://ncomva.se/flash/projects/Guide/>
 28. GAV Flash parallel coordinates plot component, <http://ncomva.se/flash/projects/Guide/?chapter=Visualizations§ion=Parallel%20Axes%20Chart>.
 29. GAV Flash choropleth map component, <http://ncomva.se/flash/projects/Guide/?chapter=Visualizations§ion=Choropleth%20Map>.
 30. Card SK, Mackinlay JD and Shneiderman B. *Readings in Information Visualization: Using Vision to Think*. San Francisco, CA: Morgan Kaufman Publishers, 1999.
 31. Data set of 9020 Sweden zip codes, <http://vitagate.itn.liu.se/GAV/gridmap/swedenzip/SwedenZipCodeData.xls>.
 32. Piringer H, Tominski C, Muigg P and Berger W. A multi-threading architecture to support interactive visual exploration. *IEEE Trans Vis Comput Graph* 2009; 15(6): 1113–1120.
 33. Shneiderman B. The eyes have it: A task by data type taxonomy for information visualization. *Proceedings of the 1996 IEEE Symposium on Visual Languages*; 1996 Sep 3–6; Boulder, Colorado, USA. Washington DC: IEEE Computer Society 336–343.
 34. Wohlfart M and Hauser MH. Story telling for presentation in volume visualization. *Proceedings of EuroVis 2007*; 2007 May 23–25; Norrköping, Sweden. Washington DC: IEEE Computer Society; pp. 91–98.
 35. Demonstration videos of GAV Flash applications, <http://vitagate.itn.liu.se/GAV/videos/VDA>.
 36. Examples of vislet in blogs, <http://swedeneurostat.blogspot.com/> and <http://swedeneurostat.blogspot.com/2010/03/education-in-eu-example-with-vislets.html>.
 37. Ho Q, Åström T and Jern M. Geovisual analytics for self-organizing network data. *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST)*; 2009 Oct 11–16; Atlantic, New Jersey, USA. Washington DC: IEEE Computer Society; pp. 43–50.
 38. Amirijoo M, Frenger P, Gunnarsson F, Kallin H, Moe J and Zetterberg K. Neighbor cell relation list and physical cell identity self-organization in LTE. *Proceedings of the 2008 IEEE International Conference on Communications Workshops*; 2008 May 19–23; Beijing, China. Washington DC: IEEE Computer Society; pp. 37–41.
 39. Dahlman E, Parkvall S, Sköld J and Beming P. *3G Evolution: HSPA and LTE for Mobile Broadband*. Oxford: Academic Press, 2007.