Expression of Interest

Research proposal NWO VIEW 2005

1 Project

1a) **Title:** Expression of Interest

1b) **Project Acronym:** EOI

1c) **Principal Investigator:** prof.dr. Jarke J. van Wijk, TU Eindhoven

2 a) Summary

Data visualization concerns the visual presentation of data, such that users can easily obtain insight. A key ingredient for successful visualization systems is that they show interesting aspects in a clear way. This project studies how this can be done. The first key question is how users can be enabled to specify their (current) interest in an effective and efficient way. The second key question is how this user-defined interest can be used such that the visual attention for each element displayed corresponds to its importance. In the third place, visualization is an interactive and explorative process. How can we enable the user to change his interest continuously, and thereby to navigate through large data spaces efficiently and effectively?

The aim is to develop generic models and guidelines for this, which form a basis for a number of new methods and techniques. The methods will be applied and tested in the context of software visualization. Software visualization is a challenging field, where large and complex data-sets have to be handled, and where multiple views and multiple visualization methods have to be used. During the analysis of software systems, the interest of the user will change quickly, and also, just a small part of the data will be of interest. This urges a need for visualization methods where *interest* is given an explicit and central role.

2 b) Samenvatting

Visualisatie betreft het visueel presenteren van gegevens, zodanig dat gebruikers eenvoudig inzicht kunnen verkrijgen. Het is hierbij van belang dat interessante aspecten op een heldere wijze getoond worden. In dit project wordt onderzocht hoe dit kan worden gedaan. De eerste vraag hierbij is hoe gebruikers in staat kunnen worden gesteld om hun interesse te specificeren, de tweede vraag hoe dit kan worden gebruikt zodat de visuele aandacht die elk element trekt overeen komt met zijn belang. In de derde plaats, visualisatie is een interactief en exploratief proces. Hoe kunnen we de gebruiker in staat stellen om zijn interesse voortdurend bij te stellen, en zodoende snel en effectief door de gegevens te navigeren?

Het streven is om hiervoor generieke modellen en richtlijnen te ontwikkelen, op basis waarvan een reeks nieuwe methoden en technieken zal worden ontwikkeld. Deze methoden zullen worden toegepast en gevalideerd voor het visualiseren van software. Software visualisatie is een uitdagend veld, waar omvangrijke data-sets moeten worden verwerkt, en waarbij meerdere visualizatie methoden, al dan niet tegelijkertijd, moeten worden gebruikt. Tijdens de analyse van software systemen zal de interesse van gebruikers vaak snel wisselen, en verder is ook vaak slechts een klein deel van de gegevens werkelijk interessant. Dit geeft een sterke behoefte aan visualisatiemethodieken waarin aan *interesse* een expliciete en centrale rol wordt toegekend.

3 Classification

VIEW, theme Visualization

4 Composition of the Research Group

The following table shows the composition of the research group and their expected research effort. All persons mentioned are member of the Visualization group of the faculty of Mathematics and Computer Science of the Technische Universiteit Eindhoven (TU/e – W&I). The PI will be PhD supervisor of the PhD-students.

PI	Prof.dr.ir. J.J. van Wijk	0.2 fte
UD	Dr. K. Huizing	0.1 fte
UD	Dr.ir. A.C. Telea	0.2 fte
UD	Dr.ir. H.M.M. van de Wetering	0.1 fte
PhD	Vacancy	1.0 fte
PhD	Vacancy	1.0 fte
PhD	Vacancy	1.0 fte

5 Research School

The project will be carried out under the auspices of the research school ASCI (Advanced School for Computing and Imaging).

6 Description of the Proposed Research

6.1 Problem Statement

The intended research can be summarized in the following research question:

> *How can we enable users of visualization systems to specify their interest and visualize the data accordingly?*

Our aim is to define new models, taxonomies, methods and techniques for this. We argue that *interest* is a central issue in visualization, and we expect that more effective interactive visualization systems can be developed when interest is explicitly modelled and exploited in the visualization process. We study how to do this, and will validate the various solutions found. As application domain we use Software Visualization.

6.2 Background

Since the late eighties *visualization* has emerged as a new discipline, as a response to the fast growth of computing power and hence of the produced data sets. The aim of visualization in general is to provide insight in large amounts of data through visual methods. An enabling technology is the development of fast hardware for computer graphics, such that large geometric models can be shown and interacted with at interactive and even real-time speeds.

The field of visualization is usually split into two separate disciplines. *Scientific Visualization* is concerned with the visualization of scientific and technological data sets, such as flow data, molecular data, and bio-medical data. The data are typically continuous and have a natural geometric interpretation. *Information Visualization* (Card *et al.,* 1999; Spence, 2000; Ware, 2004) is concerned with discrete data, such as multivariate data, hierarchical data, and networks. In this proposal we mainly focus on Information Visualization, and especially Software Visualization.

The visualization pipeline is a standard and convenient way to describe visualization systems. The data is processed in the following sequence of steps:

- *Filtering*. The original data is prepared for visualization. A subset of the data is selected, and the data is transformed and possibly smoothed;
- *Mapping*. The abstract data is mapped to geometric objects and material properties, such as colour and texture. In information visualization this is often a non obvious and

challenging step, since abstract data by definition have no fixed geometric representation, requiring the invention of suitable metaphors;

- *Display*. The (2D or 3D) geometric objects are displayed on a screen with certain projection and lighting conditions. The recent advancements in available software libraries (OpenGL) and graphics hardware allow for high performance rendering, which can be achieved with limited implementation efforts.

Insight in the data can be achieved when *interesting things* are shown *in a clear way*. Almost by definition, this is often impossible to automate. Explorative visualization is used because it is a priori unknown what is interesting or not. Interaction is needed such that the user can select those aspects he finds interesting, and graphical attributes have to be tuned such that these pop out. Each processing step therefore typically has many parameters, which can be adjusted in an arbitrary order. Some examples of parameter changes are selection of a different subset of the data, the use of different colors and/or shapes to represent the data, and to zoom in on details.

Interest driven interaction plays a central role in visualization, but only few researchers have studied how *interest* can be modeled and exploited explicitly. We propose to give *interest* a central position in visualization. This gives rise to the following questions: how can we enable the user to specify his interest, how can we use this to obtain visualizations such that the interesting aspects are presented as clearly as possible, and how can we handle dynamic exploration? In the following we elaborate on this.

6.3 Specification

First, we consider how to enable the user to specify his interest in the data. Data comes in many forms. In Information Visualization, the data typically consist of tables (records with attributes), hierarchical data, and networks, and combinations thereof. Attributes come in different types. These can be numerical (weight, length), ordinal (small-medium-large), nominal (gender, name), or relational (parent, sibling). Also, they can be tuples or lists themselves.

We propose to associate with each object (record, node, edge, …) and each of its attributes a *Degree Of Interest* (DOI), which expresses the interest of the user and which is used to control the presentation. In the simplest case, each object has a Boolean value: interesting or not, which is set by the user. An example where this is used is for *selection*, for instance to select files in a file browser for a certain operation. In (Wills, 1996) selection is studied in detail. A taxonomy is defined, requirements are summed up, and a calculus for the combination of multiple selections is defined. Wills shows that just a few different combinations are useful. Instead of binary DOIs, they can also be ternary (corresponding to hidden, visible, selected), or higher valued; also, a continuous range can be used.

Another approach to derive DOIs per object is from user-defined DOIs on attributes. An early example are *dynamic queries* (Williamson and Shneiderman, 1992), well-known from the Spotfire display (Ahlberg and Shneiderman, 1994). Interactive exploration of multivariate data via brushes was first introduced by Becker and Cleveland (1988), and later generalized by Martin and Ward (1995). A brush consists here of a set of ranges on all attributes. Martin and Ward present how these can be presented, manipulated and combined, using various multivariate visualization methods. In the Influence Explorer (Tweedie *et al.* 1996), the Attribute Explorer (Spence and Tweedie, 1998), and the Prosection Matrix (Tweedie and Spence, 1998) again ranges on attributes are used. Objects denote for instance design alternatives, attributes denote performance indicators, ranges acceptable values. Next, for each alternative the number of attributes with acceptable values are determined, which number is visualized using color in a scatterplot, giving a strong image of acceptable alternatives, as well as the status of the other ones.

Zooming and panning also fall into this category. Here, the ranges denote 2D geometric space, the user can select an area of interest, which can be panned, zoomed in and out. The 2D space can denote data-dimensions, but also dimensions of the final image that visualizes the data. One challenge is to integrate a focus and context, well-known solutions are the Perspective Wall (Mackinlay *et al.,* 1991), fish-eye views (Furnas, 1986; Leung and Apperly, 1994) and hyperbolic browsers (Lamping and Rao, 1996; Munzner, 1998), where non-linear distortions are used such that areas of interest are shown larger.

Ranges are an easily understandable concept, but fall short when the user is for instance interested in objects where two different attribute values x and y are the same. User-defined values (here $z=x-y$) can be used to solve this, but this is not particularly intuitive. A third approach to define a DOI per object is to generalize on selections of the user. A user can indicate one or more (virtual) objects with interesting values, and next ask for similar ones. One way to implement this is to use a metric on the attribute space, and to use the reciprocal of the minimal distance to the selected objects as DOI. Related to this are multi-dimensional transfer functions to control opacity in volume rendering. Kniss *et al.* (2003) define such transfer functions via multi-dimensional Gaussians. Many other ways to define multi-dimensional functions can be thought of. For instance, the user could indicate a set of objects, and let the system search for objects within the convex hull. Also, the objects can be clustered first, and other objects within the same cluster can be shown, an approach we used in (van Wijk and Selow, 1999). Tzeng *et al.* (2003) showed that a neural network can be employed to generalize on user selections.

DOIs can also be based on the distribution of the data, in order to search for average or infrequent objects, in some respect. This is useful to search for average behaviour, removing noise, as well as for the detection of outliers.

One fundamental challenge is that in many occasions the patterns of interest will be hard to specify explicitly. Actually, this is the very reason that visualization is used, the visual system is excellent in pattern detection, also when the formal definition of a pattern is hard to give. We acknowledge this, but still we expect that the approaches proposed here will be beneficial, for instance to reduce the amount of information to be studied more carefully to a fraction of the original.

It is not difficult to define complex and sophisticated methods to specify and manipulate DOIs. Also, in for instance the data base and data mining community many such methods are available. However, it is a much harder challenge to make these accessible to and understandable for the user such that he is enabled to explore the data intuitively and effortlessly. The design of effective interaction metaphors and interface widgets is vital here. Some good examples in this respect are the alphaslider (Osada *et al.*, 1993; Ahlberg and Shneiderman, 1994) for the selection of alphabetic ranges, and the InfoCrystal, designed by Spoerri (1993), which enables users to visually define Boolean operations on selections.

This short and non-exhaustive overview shows that the selection of interesting objects has gotten attention already. Nevertheless, we expect that much can be gained by studying this topic separately in more depth. We intend to address the following issues:

- The approaches mentioned seem unrelated at first sight, we expect that unifying them will lead to more insight and also bring new inspiration. An extensive taxonomy on the specification of DOI's and operations upon them is therefore a first aim. This should lead to a compact and orthogonal set of basis functions and operations.
- Most current methods are defined just for multi-variate data. In software visualization, graphs and trees are a dominant data type, for which it is unclear if the same approaches apply.
- Only few visualization systems use non-visual analysis methods, such as descriptive statistics, clustering, Bayesian networks and neural networks. For very large data-sets, with more than tens of thousands of objects, we expect that the use of such methods is highly beneficial, if not unavoidable.
- Multiple DOIs can be defined and used, successively as well as simultaneously. The combined use of these provides additional power. One obvious use is for instance refinement of search via (fuzzy) Boolean operations. Another option is interpolation. In

(Van Wijk and Nuij, 2003; 2004) we showed how smooth interpolation can be defined for geometric zooming and panning, in (Van Wijk and Overveld,2003) we showed how multi-variate parameter settings can be defined easily with a widget for the interpolation of presets. Definition of suitable metrics on the data spaces is important here. For continuous variables this is relatively straightforward, for ordinal variables this is harder, and for nominal data it is not clear how to do this. In the latter case, a starting point is to use continuous DOIs and use smooth transitions between different settings of the DOIs.

- Development of intuitive interaction metaphors for the new methods for the specification of interest developed.

6.4 Presentation

In the previous section we discussed how to assign a Degree of Interest to objects, in this section we consider how these can be used to support the viewer, such that objects of interest clearly stand out. Visual perception has been studied intensively, and the results thereof can and should be used to obtain effective visualizations. An excellent overview on this is given by Ware (2004). Chapter 5 of his book is titled *Visual attention and Information that pops out*, which is the relevant topic here.

Graphic representations of data objects have attributes like color, shape, size, position, orientation, and texture. A simple approach to let interesting objects stand out is to parametrize a graphical attribute that is not used for other purposes to the DOI. For instance, if all objects are black, color interesting objects red; if all lines are thin, make interesting lines thick. Another approach is to add graphical elements, for instance by encircling interesting objects, or by assigning the background locally a different color. For simple cases, when all objects are homogeneous or when a single, static visualization is designed, these approaches work well.

However, in more complex cases problems arise. First, when the visualization contains a wide variety of objects, with varying attributes, it is often hard to see what is highlighted. An example is the search function in Internet Explorer. Here the colors in a rectangle around the text searched are changed, which falls short when the webpage is very messy or when the text is small. Also, when a user can define mappings himself, default solutions can fall short. A thin red line around selected items will not be effective if the interior of all items is set to red.

Interaction between different graphical attributes is another concern. Ideally, each object should get visual attention proportional to the DOI. However, when for instance objects vary in size, and color is used to emphasize objects, larger objects are likely to get more attention. Most research in this area has been done for icons. This is indeed an important aspect, however, in many visualizations lines, enclosing rectangles, and text also are important elements.

Guidelines on the relative strength of various graphical attributes are available (Mackinlay, 1986; Ware, 2004), but only few quantitative models exist that relate the visual attention attracted with graphical attributes. To deepen the insight, but also for direct use in applications, such models are welcome. These can be set up in various degrees of sophistication and complexity. Ideally, absolute measures should be derived, such that the use of different attributes can be compared; as well as measures that take multiple aspects simultaneously into account.

Standard cues for selection are color and thickness, but many other attributes can be used. A nice example is the Semantic Depth of Field (Kosara *et al.*, 2002), where important areas are shown sharp and lesser important areas blurred. Ware (2001) advocates the use of 2 ½ D design: Elements are laid out in two dimensions, the third dimension is used for additional cues. One example is the use of cushions in treemaps to emphasize hierarchical structure (Van Wijk and Van de Wetering, 1999). Three-dimensional icons give many options for extra cues, such as the gloss of the surface (Pellacini *et al.*, 2000), height and curvature. Dynamic cues, such as blinking and motion (Ware and Bobrow, 2004) are very strong, and provide solutions when more subtle options fail to attract attention. Finally, position is a strong and important cue. Often, position will be the result of a sophisticated lay-out algorithm, and it is not possible to modify it independently afterwards. Nevertheless, as for instance shown by Keim (1994), positioning items dependent on their relevancy gives a good insight in large amounts of data.

In our research we intend to address these issues, as follows

- Enumerate and categorize options for visualizing importance. Collect existing guidelines, models, and results;
- Explore the opportunities of novel methods;
- Derive quantitative models for measuring visual attention.

These steps will be iterated several times, with an increasing scope. We realize that the problem domain is large, and depends on for instance the particular visualization method used, the graphical elements used, and the data itself. We will focus first on simple, but generic methods (such as scatterplots), followed by more complex methods. In each iteration prototypes will be developed, which will be used for user tests and validation. Ideally, generic methods should come out. We see two possible directions here:

- Construction of a layer between the rendering and the final pipeline. Input to the layer are the graphical elements to be shown. For each element values of attributes and the DOI are given. Furthermore, the user can indicate which attribute(s) can/should be used for importance visualization. The layer selects suitable valuations for these attributes, based on the elements to be shown and quantitative models of attention. Attributes that can be varied are for instance colors, linewidths, size, and projection.
- Also, an image based approach can be used. First, the image is rendered in a standard way and stored. Second, a mask is constructed such that important areas are shown white, and lesser important areas black to grey. This can easily be realized by rendering the scene with color parametrized to importance. Optionally, this mask can be processed, for instance blurred. Finally, the original image is rendered again, using the mask for image manipulation. For instance, lightness, saturation, or other properties can be varied.

The first direction can exploit high-level knowledge on the image, but development of suitable models for this is difficult. Fortunately, it lends itself well to an incremental development. The second direction is easy to implement and flexible. It remains to be seen if it is indeed effective in practice. A benefit of both is that importance is modelled explicitly, and can therefore be modified separately, for instance to increase the visual attention of important aspects of the visualization.

6.5 Interactive Software Exploration

Visualization is an interactive and explorative activity. The overall pattern in a visualization session is summarized in Ben Shneiderman's Visual Information Seeking mantra: *Overview, zoom & filter, details-on-demand* (Shneiderman, 1996). Another key concept in Information Visualization is *focus+context*. Both stress that the user must be enabled to zoom in and focus, while on the other hand he should maintain an overview and understand the context of the data focused on.

Users should be supported in this process, not only by providing suitable widgets to make selections etc., but also support on a higher level could be useful. A typical area where this occurs is interactive software exploration for the purposes of reverse engineering and system evaluation. When a number of systems are evaluated, the same patterns occur in the way the analyst explores the software. To support interactive exploration and navigation, in general and in particular for Software Explortation, we intend to develop a taxonomy, followed by studies in depth. Some directions are:

- *Providing history browsing mechanisms.* Brodlie *et al.* (1993) have shown how parameter settings can be stored conveniently in a tree-structure, in order to simplify stepping back and forth. Van Wijk and Van Liere (1993) used traces of the path of the user to visualize navigation in high-dimensional space. Telea *et al.* (2000) have shown how history mechanisms can be used to facilitate the construction of data-flow networks.

- *Automated navigation.* Dennis and Healey (2002) have developed methods for automated camera motion along interesting areas, in the context of cartography. Generalization of these to more abstract spaces, such as software systems, is a challenge.
- *Automated scanning.* The Grand Tour method (Asimov, 1985) for the projection of high-dimensional information spaces onto planes is one example where a large space is automatically scanned. The aim of this method was to show a sequence of close projections, such that the space of all possible projections is covered quickly and densily. Would this principle apply also to non-cartesian spaces? I.e., spaces where the axes no longer denote scalar attributes, but for instance positions in the system and a variety of error metrics?
- *Adaptive scale.* During browsing and scanning a larger scale is often advantageous to maintain an oversight. Igarashi and Hinckley (2000) introduced automated zooming out when the user browses through documents, in Van Wijk and Nuij (2004) we gave an alternative solution for optimal paths in zoom-pan space. It is challenging if these concepts carry over to other, non-spatial dimensions as well.
- *Standard patterns.* When certain studies are done repeatedly, standard patterns for the navigation through the data will appear. Can we detect these, and can these be used to offer support to the user? One use case here is for instance analysis of large amounts of code, searching for anomalies.

6.6 Context

Information Visualization is the main focus of the TU/e Visualization group. Some examples of results are our work on the visualization of hierarchical data (van Wijk *et al.*, 2000; 2003), and graphs (Van Ham *et al.*, 2002, 2004). Recently, much of our work is related to the visualization of software, ranging from high-level descriptions, such as UML-diagrams and formal models, to the final code and test traces, as well as architectural aspects of software visualization systems (Telea *et al.* 2003; 2004). Software visualization is a challenging and attractive field, for various reasons. Software systems are large and complex. Systems consisting of multiple millions of lines of code are not uncommon. Often many different versions (in time, per model, per country) exist. Visualization can help to get a insight in these large systems, such that multiple stakeholders, with varying interests, benefit from this. Managers want an overview, architects are interested in aspects as the global structure and the impact of refactoring, programmers need deep insight in small parts of the system and how these fit in the overall system. Another important aspect is that almost all possible types of abstract data play a role in software visualization, and should ideally be shown simultaneously. Software is organized hierarchically, calling structures are graphs, components have high-dimensional attribute sets (especially metrics), on the lowest level text is important.

The department of Computer Science TU/e has started LaQuSo, our Laboratory for Quality Software. The aim of LaQuSo is to develop and test methods and tools for assessing and approving software systems, based on variety of sources, ranging from high-level models to the code itself. Software visualization fits naturally in this. Currently, one PhD student (Lucian Voinea) studies the visualization of dynamic aspects of software (Voinea *et al.*, 2004), and another one (Hannes Pretorius) has started with the visualization of large state spaces. Frank van Ham will finish his PhD on graph visualization in 2005, and part of his work also concerns software visualization (Van Ham, 2003). Furthermore, six master students study various aspects of software visualization.

Recently, we have started cooperation with experts on software analysis. In the Reconstructor project we cooperate with prof. A. van Deursen (TU Delft, CWI). The focus here is on reconstruction of software architectures. A post-doc and a PhD student will work on analysis methods, a PhD student at TU/e will work on the presentation of multiple views. Also, cooperation has started with prof. C. Lewerentz (University of Cottbus). He has made the Software Tomography system available to LaQuSo, and in the future we aim at expansion of this cooperation.

For various other aspects we can take advantage of the expertise of colleagues at TU/e. We have a good cooperation with prof. J.-B. Martens, Department of Industrial Design, who is an expert in perception (Martens, 2003), user studies, and the analysis of experimental results. For statistics, we rely on the broad knowledge and experience of dr. J. Dijkstra and other experts within the department.

7 Plan of Work

The work programme consists of three separate projects, with topics *Specification of Interest* (section 6.3), *Presentation of Interest* (section 6.4), and *Interactive Software Exploration* (section 6.5). The content of these projects is described in previous sections. Each project will be done by a PhD student, and should lead to a PhD-thesis. The structure of each project is more or less the same:

1. Collect and study related work (3 months);
2. Derive requirements (1 month);
3. Develop a taxonomy of the problem and solution space (2 months);
4. Define a generic model for the problem at hand (6 months in total);
5. Define new methods and techniques that implement (parts of) this model (12 months in total);
6. Apply to real-world case studies, verify and validate the results (12 months in total);
7. Write PhD thesis (9 months).

The development of successful visualization methods is currently a matter of design and invention. In the words of Spence (Spence, 2000): "This is not surprising in view of the complexity and unpredictability of typical tasks." We acknowledge this, and hence each project is structured as a design process. Step 4 to 6 will be iterated multiple times, leading to an increase in the strength and usefulness of the models developed. In the end, we hope to obtain more generic models, which provide a solid foundation for solutions to the problem addressed in this proposal.

Evaluation of new methods in practice is essential in visualization. We therefore propose to use a prototyping approach, characterised by fast implementation and evaluation of new ideas. We will evaluate our new methods via informal user tests first, followed by more extensive quantitative experiments. The definition and selection of suitable tasks and data-sets, based on the requirements defined, is a challenge on its own here.

The new methods will be applied for software visualization, typically within one of the systems for this we are developing or have developed in the past. Nevertheless, if one of the students has a bright idea for a new software visualization method, which allows a user to view interesting things clearly in short time, we will encourage to elaborate on this.

The results of the project will be disseminated in the form of contributions to conferences (such as IEEE Information Visualization, IEEE Visualization, and ACM Software Visualization) and journals (IEEE TVCG, ACM TOG, ACM TOCHI), three PhD-theses, as well as software tools and prototypes.

The three projects have their own focus, but are obviously strongly coupled. We hope and expect that the three PhD-students will closely cooperate and will stimulate and take advantage of each other.

8 Expected Use of Instrumentation

Standard computer equipment will suffice.

9 Literature

Five important publications of the team are indicated in bold.

Ahlberg, C. and Shneiderman, B. (1994) The Alphaslider: a compact and rapid selector. In *Proceedings CHI'94*, 365-371.

Asimov, D. (1985) The Grand Tour: A tool for viewing multidimensional data. *SIAM Journal on Scientific and Statistical Computing* 6(1), 128-143.

Becker, R.A., and Cleveland, W.S. (1988). Brushing Scatterplots. In Cleveland, W.S. and McGill, M.E. (eds.), *Dynamic Graphics for Statistics*, Wadsworth, 201-224.

Brodlie, K., Poon, A., Wright, H., Brankin, L., Banecki, G., and Gay, A. (1993) GRASPARC: A Problem Solving Environment Integrating Computation and Visualization. In *Proceedings IEEE Visualization 1993,* 102-109.

Card, S.K., J.D. Mackinlay, and B. Shneiderman (1999) *Readings in Information Visualization – Using Vision to Think.* Morgan Kaufman, San Francisco, 1999.

Dennis, B.M., Healey, C.G. (2002) Assisted navigation for large information spaces. In *Proceedings IEEE Visualization 2002,* 419-426.

Furnas, G. (1986) Generalized fisheye views. In *Proceedings of CHI'86*, 16–23.

Ham, F. van, Wetering, H. van de, Wijk, J.J. van, (2002) Interactive Visualization of State Transition Systems. *IEEE Transactions on Visualization and Computer Graphics* 8(4): 319-329.

Ham, F. van (2003) Using Multilevel Call Matrices in Large Software Projects. In *Proceedings IEEE Information Visualization 2003,* 227-232.

Ham, F. van, Wijk, J.J. van (2004) Interactive Visualization of Small World Graphs. In *Proceedings IEEE Information Visualization 2004,* 199-206

Igarashi, T., and Hinckley, K. (2000) Speed-Dependent Automatic Zooming for Browsing Large Documents, CHI Letters 2(2), 139-148.

Keim, D.A. and Kriegel, H.-P (1994) VisDB: Database Exploration Using Multidimensional Visualization. *IEEE Computer Graphics and Applications,* 14(5), 40-49.

Kniss, J., Premoze, S., Ikits, M., Lefohn, A., Hansen, C. and Praun, E. (2003) Gaussian transfer functions for multi-field volume visualization. In *Proceedings of IEEE Visualization 2003*, 497-504.

Kosara, R., Miksch, S., Hauser, H. (2002) Focus and Context Taken Literally. *IEEE Computer Graphics and Applications* 22(1), 22-29.

Lamping, J.S. and Rao, R. (1996). Visualizing large trees using the hyperbolic browser. In *ProceedingsCHI'96,* 388-389.

Leung, K. Y. and Apperly, M.D. (1994). A review and taxonomy of distortion-oriented presentation techniques. *ACM Transactions on Computer-Human Interaction*, 1(2), 126-160.

Mackinlay, J. D. (1986). Automating the Design of Graphical Presentations of Relational Information. *ACM Transactions on Graphics, 5*(2), 110-141.

Mackinlay, J.D., Robertson, G.G., and Card, C.K. (1991) The Perspective Wall: Detail and Context Smoothly Integrated. In *Proceedings CHI'91*, 173-179.

Martens, J. B. (2003). *Image Technology Design - A Perceptual Approach*. Kluwer.

Martin, A. and Ward, M.O. (1995) High dimensional brushing for interactive exploration of multivariate data. In *Proceedings IEEE Visualization 1995*, IEEE Computer Society Press, 271-278.

Munzner, T. (1998) Exploring Large Graphs in 3D Hyperbolic Space. *IEEE Computer Graphics & Applications*, 18(4), 18-23.

Osada, M., Liao, H., and Shneiderman, B. (1993) Alphaslider: development and evaluation of text retrieval method using sliders. In *Proc. 9th Symposium on Human Interface (Kobe, Japan),* 91-94.

Pellacini, F., Ferwerda, J.A., and Greenberg.D.P. (2000) Toward a psychophysically-based light reflection model for image synthesis. In *Proceedings ACM SIGGRAPH 2000*, 55-64.

Shneiderman, B. (1996) The eyes have it: A task by data type taxonomy of information visualizations. In *Proceedings of IEEE Symposium on Visual Languages '96*, 336–343.

Spence. R., and Tweedie, L. (1998) The Attribute Explorer: information synthesis via exploration. Interacting with Computers 11(2), 137-146.

Spence, B. (2000) *Information Visualization,* Addison-Wesley.

Spoerri, A. (1993) InfoCrystal: A Visual Tool for Information Retrieval. In *Proceedings IEEE Visualization'93,* 150-157.

Telea, A., and Wijk, J.J. van (2000) SmartLink: An Agent for Supporting Dataflow Application Construction. In *Proc. IEEE VisSym '00*, Springer, 2000, 190-198.

Telea, A., Maccari, A., and Riva, C. (2003) An open toolkit for reverse engineering data visualization and exploration. In *Technology of Object-Oriented Languages, Systems and Architectures (Proc. TOOLSEE)*, 76–89.

Telea, A., Voinea, L. (2004) A Framework for Interactive Visualization of Component-Based Software. In *Proceedings EUROMICRO 2004,* 567-574.

Tweedie, L., Spence, R., Dawkes, H., and Su. H. (1996) Externalising Abstract Mathematical Models, Proceedings CHI'96, 406-412.

Tweedie, L. and Spence, R. (1998) The Prosection Matrix: A tool to support the interactive exploration of statistical models and data. *Computational Statistics,* 13, 65-76.

Tzeng, F.-Y, Lum, E.B., Ma, K.-L (2003) A Novel Interface for Higher-Dimensional Classification of Volume Data. *In Proceedings IEEE Visualization 2003,* 505-512.

Voinea, L., Telea, A., Wijk, J.J. van (2004) EZEL: a Visual Tool for Performance Assessment of Peer-to-Peer File-Sharing Network. In *Proceedings IEEE Information Visualization 2004,* 41-48.

Ware, C. (2001) Designing with a 2 1/2D Attitude*. Information Design Journal* 10(3), 255-262.

Ware, C., and Bobrow, R. (2004) Motion to support rapid interactive queries on Node-Link diagrams. ACM Transactions on Applied Perception 1(1), 3-18.

Ware, C. (2004) *Information Visualization : Perception for Design*, 2nd edition, Morgan Kaufmann.

Wijk, J.J. van, Liere. R. van (1993) HyperSlice - Visualization of scalar functions of many variables. In Proceedings IEEE Visualization'93, 119-125.

Wijk, J.J. van, and H. van de Wetering (1999) Cushion Treemaps, *Proceedings IEEE Information Visualization'99,* 73-78.

Wijk, J.J. van, and Selow, E.R. (1999) Cluster and Calendar Based Visualization of Time Series Data. In *Proceedings IEEE InfoVis'99*, 4-9.

Wijk, J.J. van, J. Geerlings, F.J.J. van Ham, H.W.W. van de Wetering (2000) SequoiaView. http://www.win.tue.nl/sequoiaview .

Wijk, J.J. van, Overveld, C.W.A.M. van (2003) Preset based interaction with high dimensional parameter spaces. In: Post, F.H., Nielson, G.M., Bonneau, G.-P (eds.), *Data Visualization: The State of the Art 2003*, Kluwer, 391-406.

Wijk, J.J. van, Nuij, W.A.A. (2003) Smooth and efficient zooming and panning. In: *Proceedings IEEE InfoVis 2003,* 15-22. *Best paper award.*

Wijk, J.J. van, Ham, F. van, Wetering, H.M.M. van de (2003) Rendering Hierarchical Data. *Comm. ACM*, 46(9ve), 257-263.

Wijk, J.J. van, Nuij, W.A.A. (2004) A Model for Smooth Viewing and Navigation of Large 2D Information Spaces. IEEE Trans. Vis. Comput. Graph. 10(4), 447-458.

Wills, G.J. (1996) Selection: 524,288 Ways to Say "This is Interesting". In Proceedings IEEE Information Visualization '96, pages 54-61. IEEE Computer Society Press.

Williamson, C. and Shneiderman, B. (1992) The Dynamic Homefinder: evaluating dynamic queries in a real estate information exploration system. ACM, Proceedings SIGIR'92, 339-346.

10 Requested Budget

The following table gives a specification of the requested budget. In Visualization, conferences are the premier forum for the presentation of new results, and we therefore encourage our students to actively participate in these. The standard bench-fee of NWO falls short to cover the costs for this. The cost of attending two European conferences (1.500 euro or more) and three US conferences (2.500 euro or more) amount 10.500 euro per PhD student. We therefore request additional travel budget, within the constraint of the total overall budget.

(1) Personal costs	
a) PhD students: 3 × 160.026 euro	480.078 euro
(2) Other	
a) bench-fee: 3× 5.000 euro	15.000 euro
b) additional travel budget: 4922 euro	4.922 euro
Total	500.000 euro