Visual Thinking Design Patterns

Colin Ware
Center for Coastal and Ocean Mapping
University of New Hampshire
Durham, NH, USA
cware@ccom.unh.edu

William Wright
Oculus Info Inc.
Toronto
Ontario, Canada
bill.wright@occulusinfo.com

Nicholas J. Pioch
Systems & Technology Research LLC, Suite 5850, 400 W. Cummings Park, Woburn, MA 01801
npioch@STresearch.com

***Abstract*—In visual analytics, interactive data visualizations provide a bridge between analytic computations, often involving "big data", and computations in the brain of the user. Visualization provides a high bandwidth channel from the computer to the user by means of the visual display, with interactions including brushing, dynamic queries, and generalized fisheye views designed to select and control what is shown. In this paper we introduce Visual Thinking Design Patterns (VTDPs) as part of a methodology for producing cognitively efficient designs. We describe their main components, including epistemic actions (actions to seek knowledge) and visual queries (pattern searches that provide a whole or partial solution to a problem). We summarize the set of 20 VTDPs we have identified so far and show how they can be used in a design methodology.**

Keywords—design patterns, visual thinking, data visualization, visual analytics.

I. Introduction

Visualizations are tools for reasoning about data and to be effective they must support the activities of visual thinking. Part of this is ensuring that data is mapped to the display in such a way that informative patterns are available to resolve visual queries concerning the cognitive task. This requires matching the graphic representation with the capabilities of human visualization. For example, correlations between variables should be visually easy to see and commonly searched for symbols should be more distinctive than those that are rarely sought out. In addition, interactions must be designed to support an efficient visual thinking process. Visual analytics is an example of distributed cognition and cognitively efficient interactions require that perceptual and cognitive processes in the brain of the analyst must be efficiently linked to computational processes in a computer. For example, data points representing companies can be shown simultaneously in a map view and in a scatter plot view; the technique of brushing can be applied so that points on the map, when selected, are highlighted in both views. This can support reasoning about the growth of industries related to geographic regions, but to be cognitively efficient the brushing effect should ideally appear in less than a tenth of a second.

In this paper we introduce the concept of Visual Thinking Design Patterns (VTDPs) as a tool to help with the construction of cognitively efficient visualization designs. VTDPs are based partly on a prior construct developed by Ware [1] and called visual thinking algorithms (VTAs). VTDPs represent a broadening of this original concept with a change in emphasis. VTDPs are a method for describing the combined human-machine cognitive processes that are executed when interactive data visualizations are used as cognitive tools.

First we describe the characteristics of VTDPs followed by a brief description of the set of 20 we have identified to date. Two of the VTDPs are described in somewhat greater detail to show how they combine machine computation with perceptual and cognitive processes. Finally we show how VTDPs can be used in an agile design process.

VTDPs take their inspiration from Alexander's design patterns [2] intended for architects as well as designed patterns as used by software engineer [3]. Although considerable research has shown that perceptual and cognitive principles can be applied usefully to the design of interactive visualization, this knowledge is only applied in practice if a particular designer has taken an interest in the relevant research. VTDPs are intended to provide an accessible structured method for combining knowledge about interaction methods and visualization designs together with cognitive and perceptual principles.

- Like their precedents, VTDPs are intended to describe best practice example solutions to design problems where interactive visualization is an intended component.
- VTDPs provide a method for taking into account perceptual and cognitive issues especially key bottlenecks in the visual thinking process, such as limited visual working memory capacity. They also provide a way of reasoning about semiotic issues in perceptual terms via the concept of the visual query.
- VTDPs incorporate the common set of interactive techniques used in visualization and suggest how they may be used separately or in combination.

This is not to say that there are no prior methodologies for incorporating cognitive principles into design. About three decades ago the GOMS (Goals, Operators, Methods and Selection Rules) model [4] was introduced and more sophisticated approaches have followed in the form of the

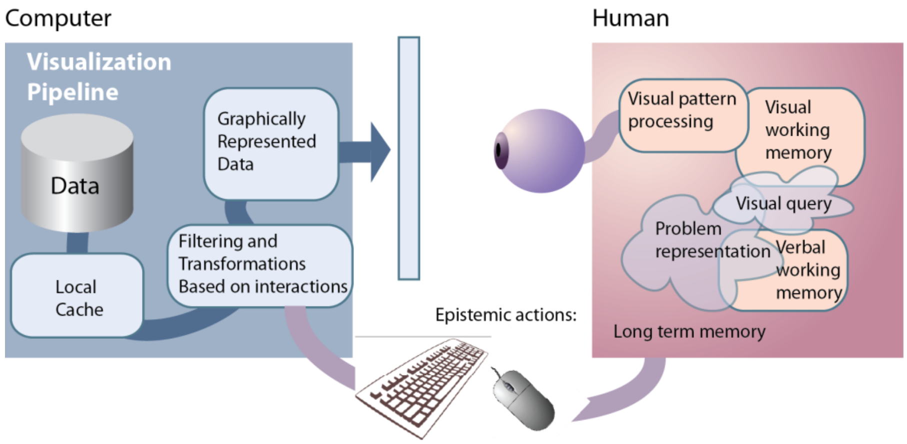

Fig. 1. Key elements of the VTDP virtual machine

ACT-R [5] model and SOAR cognitive modeling systems. These modeling systems provide executable cognitive models, containing timings for cognitive computations and as well as for common interactions such as mouse movements. But to use these models as part of a design process a proposed interface must be designed in detail and executed in a simulation that includes both simulations of cognitive operations and simulations of how the computer application will behave. This is beyond the capabilities of all but a few specialized centers. Set against this is the need for agile design based on rapid prototyping.

VTDPs should be understandable with a modest amount of training, and it should be possible to incorporate them into common design practices. VTDPs are potentially applicable to all visualization design problems, including design of presentation materials, interactive training materials and the design of analytic tools. In the following section we will concentrate on a single application domain — the design of visualization tools used in visual analytics.

Like software engineering and architectural design patterns, VTDPs are not modules and not re-usable. The demands of analysis are almost infinitely varied due to the enormous variety of data types and analytic problems. As a result, any modularization of VTDPs would necessarily restrict the domain. Nevertheless, implementations of VTDPs can take advantage of modular components. For example, map display software may allow for extra magnifying windows which can be used to support the pattern comparisons in a large information VTDP space.

II. The Components of a VTDP

A VTDP begins with a statement of the problem it is designed to solve, together with one or more examples. In most cases it will contain a pseudo-code description of the combined cognitive process and this will typically incorporate many the following components.

A. *Perceptual and Cognitive Operations*

These include converting some part of a problem into a visual query, mentally adding both imagery and attributes to a perceived symbol or other feature. Cognitive operations include decisions such as terminating a visual search when an item is found. A key bottleneck in cognitive processing is the capacity of visual working memory, so situations where visual working memory limits can lead to cognitive inefficiently are elucidated, and solutions described.

B. *Visual Queries*

A visual query is cognitive transformation of some aspect of a problem so that progress can be made using a visual pattern search. For example, finding a relationship in a social network diagram will involve visually searching for a line linking two nodes. Visual queries provide a method for reasoning about key problems of data representation..

C. *Visual Pattern Processing*

A visual pattern is the target of a visual query. The goal of efficient graphic design is to ensure that data is mapped into graphical form in such a way that all probable visual queries can be efficiently executed.

D. *Working memory load*

Working memory capacity is assumed to be at most 4 simple patterns or shapes if the patterns are previously unknown. If patterns are well known, a skilled analyst can hold more complex patterns in working memory.

E. *Epistemic Actions*

These are any actions designed to seek information. They include eye movements to focus on different parts of a display, and mouse movements to select data objects or navigate through a data space.

F. Externalizing

These are instances where someone saves some knowledge gained by putting it out into the world, for example, by adding annotations to a visualization, or checking boxes to indicate that certain information is deemed important or irrelevant.

G. Interaction Computation

This includes all parts of a visual thinking algorithm that are executed in a computer. Of particular relevance to VTDPs are computations involved in rapidly changing how information is displayed. This can be as simple as zooming or changing the range of the data that is displayed because of user interaction via a time-slider.

H. Display budget

The display budget is the amount of visual information that can be usefully displayed at a given time. For example, there is no point in displaying a graph with 5000 nodes if very few of them can be visually resolved.

I. The VTDP virtual machine

When thinking about interactive systems, it is useful to consider the simple virtual machine, illustrated in Figure 1. This contains a simple visualization pipeline as well as the key components of perceptual and cognitive processing. Very often, a gain in cognitive efficiency can be achieved by shifting part of the workload from visual working memory by means of simple computations and interactive methods.

III. A SUMMARY LIST OF VTDPS

We have identified a set of 20 VTDPs to date and the following section gives a brief summary of 11 of them, listing the remainder. In two cases, Drill Down with Aggregation, and Lateral Exploration, we provide somewhat more detail in order to show how they combine perceptual operations with machine operations. The first two design patterns, Visual Query and Reasoning with a Hybrid of a Visual Display and Mental Imagery are components of almost all visualizations. These are followed by two high level patterns, Analytic Framework and Visual Monitoring. The remaining patterns are mid-level. VTDPs that have already appeared as visual thinking algorithms in [1] are indicated.

A. VTDP: Visual Query

Most visualization involves a cognitive operation where some aspect of the problem is turned into a visual search for a pattern that can provide part of the solution. For example, someone may wish to know how a piece of information got from Jack to Jane and they have a social network diagram showing communications as lines connecting nodes representing individuals. The visual query is to find a path of lines between the node representing Jack and the node representing Jane.

B. VTDP: Reasoning with a Hybrid of a Visual Display and Mental Imagery (IV3 CH11:A3)

Almost all visual thinking is a process or reasoning where external imagery is combined with mental imagery. Mental imagery is used to represent alternative interpretations or possible additions to an external visualization. Visual queries are executed on the combined external/internal image.

C. VTDP: Analytic Framework

This provides a broad framework for analytic tasks (as opposed to monitoring tasks or routine processing tasks)[6,12]. In some cases a framework can be configured and the user can set up a system to suit a particular set of tasks and a particular data set. The analytic framework has the overall goal of 'sense-making' and as such includes a set of activities such gathering and organizing information. Most of the VTDPs in this list can be part of the analytic framework.

D. VTDP: Drill Down-Close Out with Hierarchical Aggregation

In many big data applications it is not possible to show all entities at once. A common approach for dealing with this is to aggregate items hierarchically to provide a visual overview [7]. The basic interaction is clicking on an object of interest whereupon it opens revealing the visual representations of the objects that it stands for. A reverse operation closes the object. For aggregated objects to be meaningfully used they must visually portray sufficient information scent for decision making. This is done by causing key information to propagate up the hierarchy using task relevant rules, such as a statistical summary (sum, median average), exceptions (anomalies, faults), or temporal changes A screen budget combined with the number of data objects can be used to calculate the degree of aggregation necessary. Drill down-close out works best with balanced hierarchies. For example, a ten way tree requires eight clicks to get to leaf nodes with 100 million items in the data base, but only if the tree is balanced. A 40 way tree only requires 5 clicks and a 450-way tree only requires 3 clicks to access the same amount of data. If the visual cue being searched for is pre-attentive then the time to click may be close to constant, but if it is not then a slow serial search will occur. There can be a large burden placed on working memory if multiple drill down operations are to be compared.

Process: Drill Down, Close Out with Aggregation

Display Environment: Symbols representing aggregations of data.

1. Based on a screen budget and the quantity and architecture of the data, the computer creates a hierarchy of entities, each having a visual representation that reveals some aspect of its constituent parts.

2. The visualization begins with a display of the top level object.

3. The analyst conducts a visual search for informative objects based on some aspect of their visual appearance (visual scent).

4. The analyst clicks on the object judged to have the highest probability of yielding useful information.

5. If useful information is acquired, the analyst saves the object in human or machine memory.

6. The analyst closes the object.

7. Repeat from 3.

E. VTDP: Visual Monitoring (IV3 CH11:A10)

This VTDP is applied in situations where analysts must monitor a set of measured values or instruments. Usually monitoring is interspersed with other work. The basic visual thinking algorithm involves setting up a schedule of interrupts, according to which users will periodically stop what they are doing and conduct a visual scan of a set of displays looking for anomalous patterns that may require action. Key patterns will depend on the monitoring goals. A key decision is whether interrupts should be cognitively or system generated.

F. VTDP: Cognitive Reconstruction

Most analysts will have their work frequently interrupted. They also often need to extend or redo prior analytic work as new data becomes available. Successfully resuming a prior analysis requires that the entire cognitive system be reconstructed and this involves rebuilding both the machine state and the operator state. All long term memory is more a process of reconstruction than a process of retrieval. Any full featured analytic support system will contain features to support cognitive reconstruction such as the ability to add annotations and many packages support the resurrection of a prior analytic system state through the use of scripts.

G. VTDP: Lateral Exploration

Often analysts start with a piece of information and follow links outward [8]. For example, in social networks we may follow a chain of social relationships. Or we may trace linkages between people through an organization. Lateral exploration is an alternative to the 'overview first/details on demand' approach advocated by Shneiderman. Perhaps in most cases, analysts begin by finding a lead, in the form of a piece of information that seems relevant, then follows links laterally to find related information, assembling what is relevant and discarding what is not. One of the simplest instances of this is a search for related information within a large network.

H. VTDP: Find Local Patterns using Degree-of-Relevance Highlighting (IV3 CH11:A7)

Sometimes information objects in a display are interrelated in ways that are highly task relevant. Degree-of-relevance highlighting can be useful when it is possible to display a substantial amount information on the screen at once but because of its density it cannot all be made legible. A simple interaction solves the problem; touching an object causes both it and other task relevant data objects to be highlighted. The highlighted objects may also reveal addition detail. As with the hierarchical aggregation, graphical information scent is needed to provide a starting point for visual search. As a first order approximation, degree-of-relevance highlighting is useful for between 30 and 500 graphical symbols representing data.

I. VTDP: Pattern integration across views using brushing (IV3 CH11:A5)

Brushing [9] is often useful to represent data in several different views. In brushing, selecting a data object in any one of the views causes those same data objects to be highlighted wherever they appear in all of the other views, thereby visually linking them. For brushing to be effective is it important that the highlighting technique used takes advantage of pre-attentive visual cues.

J. VTDP: Pattern Comparisons in a Large Information Space (IV3 CH11:A6)

A need to represent detailed information in a larger context is a common problem for data visualization. The most common example is a map display, where we want to compare small scale features on the map. The same problem occurs with more abstract data, for example, in large network diagrams. Any pattern comparison involves loading some aspect of one pattern into visual working memory, to be later compared to some other pattern. Pattern comparisons are far more efficient if the transfer of attention between one pattern and another can be made using eye movements, because in this case the information only needs to be held for a fraction of a second. The working memory burden is far greater when other techniques, such as zooming, are required for the comparison. Requiring more than a few simple shapes to be held in working memory will result in high error rates. Adding extra magnifying windows is a common way of ensuring that the task can be carried out using eye movements.)

One way of dealing with the pattern comparison problem is to use a method called the Generalized Fisheye View [10]. This method relies on a degree of interest function whereby the computer attempts to show only task relevant information and hides or shrinks other information. The success of this method depends entirely on the predictability of related information.

K. VTDP: Multidimensional Dynamic Queries (IV3 III CH11:A9)

With multidimensional discrete data all entities have the same set of attributes. The attributes define the data dimensions and each entity can be thought of as a point in a multidimensional space. A set of sliders is provided that can narrow the range on each of many attributes. Each slider adjustment is an epistemic action, narrowing the range of what is displayed. Ideally, feedback is very rapid (<100 msec) [11] If we assume that each dynamic query slider can be used to reduce the range selected to 10% of the original, then the number of objects that be interactively queried is ~ 10^d where d is the number of dimensions. The method has most often been used with scatter plots and to a lesser extent time series plots, but it can work with ranges displayed on maps and with node link diagrams.

L. VTDP: Table Data - Sort and Compare

Tables of data can contain glyphs instead of just numbers, in which case they become interactive visualizations. A common analytic strategy is to use sorting to bring out certain kinds of relationships visually. Sorting on one value can be useful in revealing correlated anomalies. For example, suppose we have a data table with system failures represented by a glyph in one of the columns. If failures tend to occur when a particular variable is low, sorting from low to high on that variable will bring marks representing faults to the top of the table where they can be visually compared to values in other columns of the table.

Other VTDPs are the following.

- VTDP: Pathfinding on a Map or Diagram (IV3 CH11:A2)
- VTDP: Discovering Novel Temporal Patterns
- VTDP: Table Data: Compute, Chart and Find Patterns
- VTDP: Model-Based Interactive Planning
- VTDP: Design Sketching (IV3 III CH11:A4)
- VTDP: Task List
- VTDP: Query by Example
- VTDP: Presentation linking images and words.

IV. VTDPs in the Design Process

Visual thinking design patterns are intended as tools for the designer to use in reasoning about visualization design. In the following paragraphs we sketch out the design process as a series of steps. In most cases these steps should be part of a spiral design methodology, with multiple iterations.

A. Step 1: High level cognitive task analysis

At the start of the design process the team members must establish, in a general way, the problem to be solved by the final product. Initially the description should not specify the implementation method so as not to prejudge the solution. Refinement will come later.

B. Step 2: Data inventory

It is important to know early on in the process what data is available that bears on the cognitive goals, what is readily accessible, and what is likely to be accessible only with time delays. Sometimes confidentiality can be a stumbling block. The structure of the data will be important in defining the visualization and so will the semantics. Data can be enormously varied in its properties, but the following list contains some of the attributes that are likely to appear in any inventory.

- Quantity: Very big data requires different approaches beyond what applies to medium or small data.
- Structure: E.g. hierarchical, map layers, network, multi-dimensional discreet.
- Time to access: Sometimes real-time or near real-time is critical.
- Ease of access: Some data is readily accessible; some data may require new infrastructure; some data has explicit costs; some data has security issues.
- Interrelationships and interdependencies.
- Quality and reliability.
- Potential for derived data products – processing infrastructure.

The data inventory process will almost certainly start to blend into step 3, task refinement, because data is only important as it relates to task requirements.

C. Step 3: Refinement of cognitive task requirements

As the data is understood, the set of tasks can be refined. Knowing the specifics of particular data objects will suggest additional questions that may be addressed. As a generally strategy it is useful to work top down, breaking down the overarching goal into subgoals that relate to specific data sets. It is also important to establish interdependencies between tasks, such as the order in which they must be performed. At some point specific computer based tools may be invoked, but this should be held off as long as possible.

Only part of an analytic problem is likely to be amenable to solutions that use visualization and it is important to divide the problem into aspects that can use visual thinking and aspects that use other forms of thinking or computation. The following of aspects of a problem suggest a visualization solution.

- There must be a way of transforming the problem so that solutions are discoverable through a visual pattern search.
- The task should not be so repetitive and standardized that an automatic computer pattern search is more appropriate.

Once a set of subtasks that is amenable to visual thinking solutions has been identified, they can be matched to the VTDPs and their associated visualizations.

D. Step 4: Identification of VDTPs and visualizations that can bear on the task

This step is the key creative stage in the design process. It is best led by an experienced designer who knows both the types and nature of visualization and understands the VTDPs that can be used to make them cognitively efficient.

The rapid design and development of visualizations would be almost impossible if it were necessary to come up with a radical new design and a radically new interaction method for every problem. Fortunately, there are only a small number of basic types of visualizations that have widespread use in practice, namely charts, maps, node link diagrams and tables. These are can be provided in component libraries. The key development issue is that the components should support the relevant set of VTDPs.

E. Step 5: Design decision rules

Since VTDPs incorporate rules regarding when they are most effective, it is possible to construct a set of rules for visualization problems that provide guidelines for when the different methods can be applied. For example, we have constructed the system of rules below regarding techniques that can be applied to the visualization of graphs as node link diagrams.

- Small graph (<30 nodes) static representation
- Medium graph (>30,<600) degree of relevance VTDP

- Large Graph (>600 < 5m) use dynamic queries VTDP attributes to reduce graph size to get to degree of relevance VTDP (<600).
- Very large graph (>5m) Query by example VTDP may be needed. Start with example, computer finds similar – user selects to refine query.

F. Step 6: Prototype development.

In order to test the cognitive affordances of design alternatives, it is necessary to have some form of prototype. This can be design sketches of key screens, or a rapidly developed prototype. The purpose is to provide a basis for reasoning about cognitive efficiency when the visualization is applied to the cognitive tasks identified in step 3.

G. Step 7: Evaluation through cognitive walkthrough

The cognitive walkthrough should focus on critical or frequent tasks to ensure that the cognitive execution will be efficient using the VTDPs. Small groups should talk through the steps, attempting to identify bottlenecks, such as unreasonable memory load, or instances where repetitive work can be offloaded to the computer.

It is likely that new cognitive tasks will emerge during the process and the list of cognitive tasks can be refined; links between cognitive tasks may be revealed.

H. Step 8: Alpha and beta products.

The final stages of design should apply the common techniques of developing alpha and beta product versions that can be tested and refined through use in problem solving.

V. CONCLUSION

It is often said that designers work intuitively. But they still bring all the knowledge and skills they have acquired to a design problem. Using intuition simply means that the creative reasoning process is largely unconscious and not explicitly analytical. Designers are explicitly analytical at certain stages in the design process, such as when they critique design solutions. VTDPs provide a way of training designers in the way interactive visualization techniques work in the visual thinking process, together with the limitations and strengths of each method. If the designer is familiar with them, VTDPs can become part of the mental landscape of the designer, providing a starting point for intuitive design solutions.

VTDPs are not intended to be rigidly prescriptive; instead they provide a framework for thinking about key parts of the interaction design in terms of their perceptual and cognitive efficiency. Design is an optimization process. The goal is to maximize the value of whatever is produced and to minimize the effort required to produce it. Complete information about the set of tasks is rarely available, and software tools are rarely used in isolation.

We have introduced the concept of VTDPs in this paper and described how they can be applied in a design context. The set described in the present paper is not complete or definitive and will be extended as our work progresses.

The ideal context for the VTDPs to be applied is in an agile design environment such as the one being developed around Oculus Aperture studio [13]. Aperture provides an open and extensible Web 2.0 visualization framework; it has an extensive library of customizable components, including maps, node-link diagrams and charts of various kinds such as scatter plots, bar charts and time series plots. Aperture already provides support for many of the VTDPs listed in this paper and as it develops more will be added.

ACKNOWLEDGMENTS

This study was supported by the DARPA XDATA Program under Contract No. FA8750-12-C-0317 to Oculus Info Inc.; also under STTR Contract No. W31P4Q-12-C-0207 to Systems Technology Research. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA. The ideas presented here have benefitted from discussions with many people, especially the Oculus design team, as well as Tamara Munzner, and Sheelagh Carpendale.

REFERENCES

[1] C. Ware, Information Visualization, Perception for Design, Third Edition. Morgan Kaufman. 2012.

[2] C. Alexander, S. Ishikawa, and M. Silverstein, A pattern language. Oxford University Press. 1977.

[3] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley. 1995.

[4] S.K.. Card, T.P Moran, and A. Newell, The Psychology of Human-Computer Interaction, London: Lawrence Erlbaum Associates. 1983.

[5] J. R Anderson, M. Matessa, and C. Lebiere,. ACT-R: A theory of higher level cognition and its relation to visual attention. Human-Computer Interaction, 12(4), 439-462. 1997.

[6] P. Pirolli, and S. Card, The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In Proceedings of International Conference on Intelligence Analysis (Vol. 5). 2005.

[7] N. Elmqvist, and J.D. Fekete, Hierarchical Aggregation for Information Visualization: Overview, Techniques,and Design Guidelines. IEEE Transactions on Visualization and Computer Graphics. 16(3) 439-454. 2010.

[8] F. van Ham, , and A. Perer. Search, Show Context, Expand on Demand": Supporting Large Graph Exploration with Degree-of-Interest. Visualization and Computer Graphics, IEEE Transactions on 15, no. 6 953-960. 2009.

[9] R. Becker, and W.S. Cleveland, Brushing scatterplots. Technometrics, 29, 127-42.

[10] G.W. Furnas,. Generalized fisheye views. Vol. 17, no. 4. ACM, 1986.

[11] C. Ahlberg, and B. Shneiderman. Visual information seeking: Tight coupling of dynamic query filters with starfield displays." In Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence, pp. 313-317. ACM, 1994.

[12] R. Amar and J. Stasko. A knowledge task-based framework for design and evaluation of information visualizations. In Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on, pp. 143-150. IEEE, 2004.

[13] D. Jonker, S. Langevin, N. Bozowsky and W. Wright, Aperture: An Open Web 2.0 Visualization Framework, 46th Hawaii International Conference on System Sciences, 201.