ZVTM - Developer's Guide

April 6, 2009

Contents

List of Figures

1 Introduction

Graphical user interface toolkits such as Java Swing or Qt are very powerful, generic and portable, but are limited to conventional interface widgets of the WIMP model (*Windows Icons Menus and Pointers*). Implementing advanced visual interfaces for applications that work with complex, structured graphics such as graph editing applications, geographical information systems, visual language environments or information visualization tools to name a few (see Figure 1 for a few examples) requires developers to use lower-level APIs such as Java2D or OpenGL. GUI toolkits tailored for the WIMP model also do not allow for the easy implementation of new interaction techniques coming from HCI (*Human-Computer Interaction*) research. Low-level APIs such as Java2D and OpenGL give developers much more flexibility and feature a higher expressive power, but at the cost of a much higher development and maintenance effort.

Our goal with ZVTM is to give developers a solution situated at an intermediary level of abstraction between low-level graphics APIs such as Java2D and high-level GUI toolkits such as Swing. ZVTM aims at:

- making the development of interface components involving complex structured graphics easier by hiding most of the complexity (performance, memory management, UI events, geometrical transformations, animations, concurrent access);
- ease the robust and generic implementation of new interaction techniques coming from HCI research, and their integration in existing applications;
- make it possible to quickly prototype such new techniques;
- offer researchers a generic and efficient framework for the evaluation of these new techniques through, e.g., controlled experiments.

ZVTM provides developers with the basic components required for implementing multi-scale (or *zoomable*) interfaces based on a set of rich graphical instructions. ZVTM focuses on high quality visual rendering while maintaining good performance, and on the user experience by promoting foundational concepts such as *perceptual continuity* in graphical interfaces [10] through a very powerful yet simple-to-use animation module (see Section 4). ZVTM can be used to quickly implement new multi-scale navigation techniques, bi-focal representations, new pointing techniques, etc. It also features an interface with the Jung[1] graph layout framework, and facilities to easily load graphs generated by GraphViz[2] as demonstrated in ZGRViewer[3].

ZVTM is the continuation of a project that Emmanuel Pietriga initiated with Jean-Yves Vion-Dury at Xerox Research Centre Europe while preparing my PhD. It was then called XVTM (Xerox Visual Transformation Machine). ZVTM builds upon the XVTM and is distributed under the LGPL license as was XVTM. ZVTM is now developed and maintained by INRIA project-team In Situ[4], and his hosted on SourceForge.net.

Some recent examples of use of ZVTM in our research work include a new technique that facilitates pointing at small objects, called DynaSpot [1]; techniques for navigating in large networks [3]; the evaluation of multi-scale navigation techniques in the context of the visual search for objects with particular characteristics [7]; the development of new types of magnification lenses [6]. The toolkit has also been used to implement various applications, including a visual authoring tool[5] for Semantic Web data[4, 5], visualization components for the WebContent platform[6], a visual programming language for authoring XML transformations [9], a universal Gene Ontology annotation, visualization and analysis tool for functional genomics research [2], Blast2GO [7]; a viewer for SALI networks [8], a visual environment for querying RDF graphs[9], and more.

[1] http://jung.sf.net
[2] http://www.graphviz.org
[3] http://zvtm.sf.net/zgrviewer.html
[4] http://insitu.lri.fr
[5] http://www.w3.org/2001/11/IsaViz
[6] http://www.webcontent.fr
[7] http://bioinfo.cipf.es/blast2go/
[8] http://sali.rguha.net/
[9] http://rdqlplus.sourceforge.net/

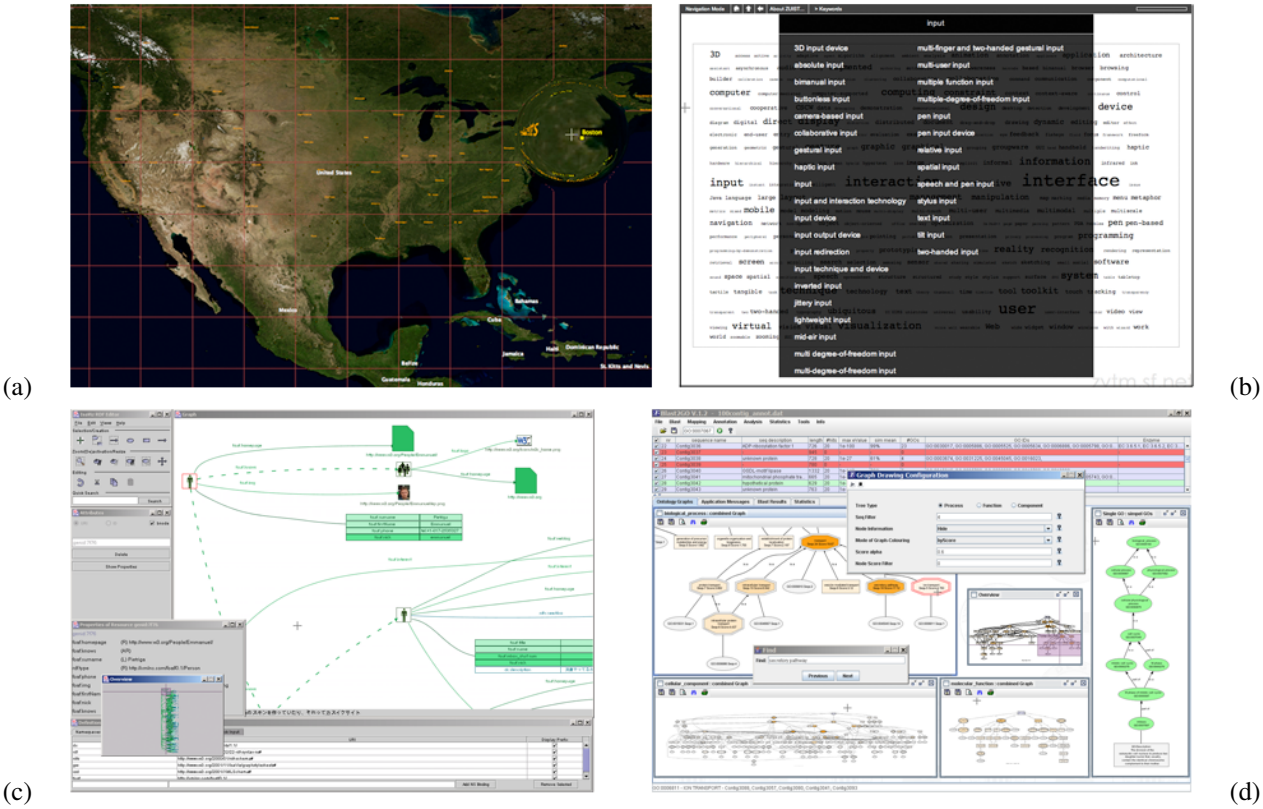

Figure 1: Various applications using ZVTM : (a) navigating in a high-resolution satellite picture (86 400 x 43 200 pixels) from NASA [11] with data from `http://geonames.org` superimposed on it; (b) navigating in a set of 570 research papers (PDF documents) using a zoomable interface, from metadata associated with documents down to the pages of each document [8] ; (c) Semantic Web data visual authoring tool (*Massachusetts Institute of Technology & World Wide Web Consortium*) [4] ; (d) universal Gene Ontology annotation, visualization and analysis tool for functional genomics research (*Universidad Politécnica de Valencia*) [2].

2 Getting and building ZVTM

3 Basic Concepts

4 Animations

5 ZUIST - a Multi-Scale Scene Manager for ZVTM

ZUIST is a generic multi-scale scene engine/API for ZVTM which makes it easier for developers to create multi-scale interfaces à la Google Earth (though in 2.5D) in which objects representing a scene at varying levels of detail are loaded/unloaded dynamically to/from memory depending on the region of the virtual space scene through the camera(s). Multi-scale scenes can be built through the ZUIST API (javadoc) or can be loaded from files that describe them declaratively using a dedicated, simple XML vocabulary.

For now, only the Javadoc API documentation is available, along with code examples from various applications: the basic viewer/debugger, UIST Archive Explorer, and Blue Marble Next Generation Explorer.

5.1 Applications

5.1.1 UIST Archive Explorer

UIST Explorer is a ZUIST-based application that lets users navigate in 20 years of papers published at the ACM UIST conference from 1988 to 2007. It was one of the 20th Anniversary Interactive Visualization tools demonstrated during the 2007 conference. Users have access to 578 research papers, typically 4-to-10 page long. Papers can be browsed by year, by author, or by keywords. Overall, the multi-scale scene is composed of more than 80,000 graphical objects, most of them being 1224x1584 pixels compressed bitmap images (representing about 2GB of compressed data), loaded dynamically into memory. Running UIST Explorer typically requires less than 512MB of RAM. During the conference, the application ran on an Apple MacBook Pro laptop hooked to a SmartBoard 3000i (see pictures and videos below), with a Java heap size set to 1GB, of which it never used more than 50

5.1.2 LRI Explorer

A similar application was developed to navigate in all research documents published by members of LRI, the computer science lab at Université Paris-Sud between 2005 and 2008, ranging from book chapters, journal and conference papers to PhD dissertations and tech reports. Documents can be browsed by author or by team, and are then organized by type of publication and year of publication, down to the level where the pages of each document appear (when a PDF version is available). The total number of documents is about 1,500. A PDF version of the document is available for 753 of them, amounting to about 15,000 pages actually readable directly within the application.

5.1.3 Blue Marble Next Generation Explorer

Navigating in a multi-scale version of NASA's Blue Marble Next Generation world map: 86400 x 43200 pixels decomposed into 2730 tiles (1350x1350 each) arranged in a pyramid. The map is enriched with geographical data taken from the Geonames database and boundaries for countries and administrative regions publicly available as ESRI shapefiles.

6 Tree and Graph Layout

7 Displaying PDF documents

References

[1] O. Chapuis, J.-B. Labrune, and E. Pietriga. Dynaspot: Speed-dependent area cursor. In *CHI '09: Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems*. ACM, 2009. To appear (10 pages).

[2] A. Conesa and S. Götz. Blast2go: A comprehensive suite for functional analysis in plant genomics. *International journal of plant genomics*, 2008, 2008.

[3] T. Moscovich, F. Chevalier, N. Henry, E. Pietriga, and J.-D. Fekete. Topology-aware navigation in large networks. In *CHI '09: Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems*. ACM, 2009. To appear (10 pages).

[4] E. Pietriga. IsaViz, a visual environment for browsing and authoring RDF. The 11th Int. World Wide Web Conference (WWW 2002) - Developer's Day: The Semantic Web - `http://www.w3.org/2001/11/IsaViz/` (W3C), 2002.

[5] E. Pietriga. Semantic web data visualization with graph style sheets. In *SoftVis '06: Proceedings of the 2006 ACM symposium on Software visualization*, pages 177–178. ACM Press, 2006.

[6] E. Pietriga and C. Appert. Sigma lenses: focus-context transitions combining space, time and translucence. In *CHI '08: Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems*, pages 1343–1352. ACM, 2008.

[7] E. Pietriga, C. Appert, and M. Beaudouin-Lafon. Pointing and beyond: an operationalization and preliminary evaluation of multi-scale searching. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1215–1224. ACM Press, 2007.

[8] E. Pietriga and M. Beaudouin-Lafon. ZUIST - UIST 2.0 Celebrating 20 Years of UIST. `http://www.acm.org/uist/archive/uist2.0/ZUIST.html`, 2007.

[9] E. Pietriga, J.-Y. Vion-Dury, and V. Quint. Vxt: a visual approach to xml transformations. In *DocEng '01: Proceedings of the 2001 ACM Symposium on Document engineering*, pages 1–10. ACM Press, 2001.

[10] G. G. Robertson, S. K.Card, and J. D.Mackinlay. Information visualization using 3D interactive animation. *Communication of the ACM*, 36(4):56–71, 1993.

[11] R. Stockli, E. Vermote, N. Saleous, R. Simmon, and D. Herring. The Blue Marble Next Generation - A true color earth dataset including seasonal dynamics from MODIS. Published by the NASA Earth Observatory, 2005.