

DATACUS:

Multi-Touch Data Explorer

Design

November 8, 2010

Ryan Daubert
Buck Heroux
Adam Jackson
Ian Smith
Matt Smith

CSCI 4308-4318. Software Engineering Project 1 & 2
Department of Computer Science
University of Colorado at Boulder
2010-2011

Sponsored By:
Nicholas Goodman

Dynamo Business Intelligence Corporation
Seattle, WA

PROJECT PROPOSAL

The goal of this project is to build a system using the latest hot technologies! Use multi-touch input devices and advanced analytic graphics and build the most advanced, most enjoyable data exploration application ever. Multi-touch, made popular by the iPhone and iPad, allow for the most natural interactions with applications ever conceived. Data exploration, as a natural task, explore, visualize and repeat is a natural fit and one that fits brilliantly with multi-touch.

Think of this project as building the cool interface from the movie "Minority Report" for exploring business data.

The project team will build:

- An interactive data explorer that will explore multi-dimensional data.
- Users will be able to graph, chart, and explore data with different multi-touch gestures.
- Target audience is business analysis users.
- Client will be written in Silverlight *or* Flash.
- Windows 7 multi-touch devices are a requirement, but other mouse/TUIO devices (large screens) are a design target.
- Data, including hierarchical data navigation, results, and querying available via web services (XMLA). In other words, this is a UI/query app, not a data crunching application.

TABLE OF CONTENTS

1	INTRODUCTION	1
2	REQUIREMENTS	2
2.1	SOFTWARE ENVIRONMENT	2
2.1.1	DEVELOPMENT ENVIRONMENT	2
2.1.2	SOFTWARE RUNTIME ENVIRONMENT	2
2.2	HARDWARE ENVIRONMENT	2
2.3	PERFORMANCE	3
2.3.1	START-UP TIME	3
2.3.2	SPEED	3
2.3.3	CAPACITY	3
2.3.4	RELIABILITY	3
2.4	FUNCTIONAL	4
2.4.1	SERVER INTERACTION EVENT DATA	4
2.4.2	SERVER INTERACTION EVENT DISPLAY	4
2.4.3	SERVER INTERACTION SCREEN UPDATES	4
2.4.4	TOUCH INPUT EVENT DATA	4
2.4.5	TOUCH INPUT EVENT DISPLAY	4
2.4.6	TOUCH INPUT EVENT SCREEN UPDATES	4
2.4.7	STATES	5
2.4.8	CHART MOTION	5
2.5	USER INTERFACE	6
2.5.1	GENERAL	6
2.5.2	GESTURES	7
2.5.3	SELECTION	7
2.5.4	OPTION MENUS	7
2.5.5	TRASH CAN	7
2.5.6	FOUR PANELS	8
2.5.7	PICKER	8
2.5.8	“SETS” (IDEAL STRETCH FOR PROJECT)	9
2.6	DOCUMENTATION	9
2.6.1	DOCUMENTS	9
2.6.2	PRESENTATIONS	10
2.6.3	WIKI SYSTEM	10
2.7	RELEASE	10
3	SUMMARY	11
4	REFERENCES	12

TABLE OF FIGURES

Figure 1	Datacus Conceptual Overview	1
Figure 2	User Interface Mock-Up	6

1 INTRODUCTION

DynamoBI is start-up with eight employees spread between Seattle, San Francisco, and India. Their core product is an open source business intelligence (BI) database built for analytics, charting, and dashboards. DynamoBI's goal is to create purpose built tools that make business intelligence easy and enjoyable. They are interested in applying new UI technology to discover ground breaking improvements in the experience of data exploration through multi-touch display interfaces. DynamoBI intends to open-source the Datacus project and use it as a basis for potential future products. DynamoBI's core product and speciality is an open source BI database. This database's purpose is 100% about analytics, charts and dashboards. Placing the Datacus package in front of their core product makes for a perfect demonstration environment for the databases they provide.

A conceptual diagram provided by DynamoBI of the proposed Datacus system can be seen in Figure 1. The diagram shows Datacus accepting touch input from a touchscreen, which is interpreted by Windows 7 and the TUIO protocol. Datacus also sends and receives data to a Mondrian server, which then retrieves information from an OLAP database selected by the user. The touch events produced by the user manipulate the content viewed and the visual representation of the data in the user's database. By doing so Datacus visually presents the selected information and allows the user to explore data through the interface.

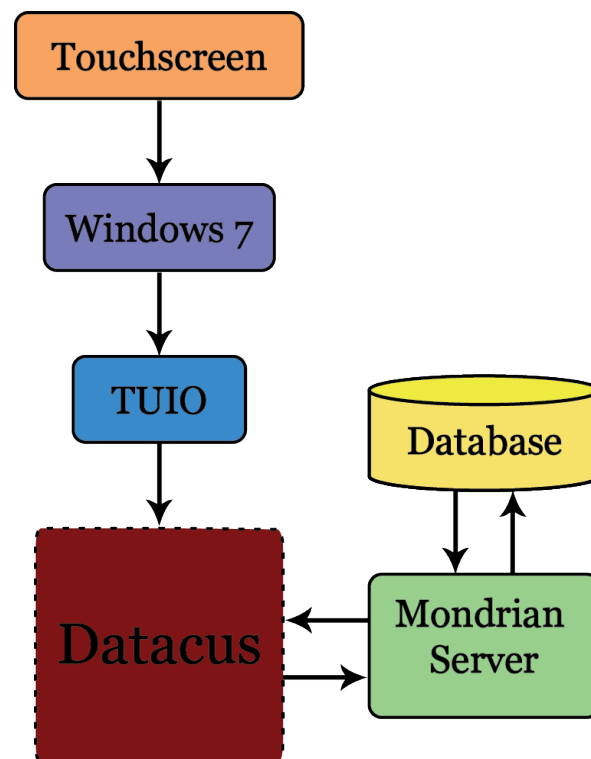


Figure 1: Datacus Conceptual Overview

2 REQUIREMENTS

The requirements for Datacus have been broken up into the following sections: Software Environment, Hardware Environment, Performance, Functionality, User Interface, Documentation, and Release. These sections elaborate on the specific requirements necessary for the completed Datacus software project.

2.1 Software Environment

The Software Environment includes both the software required to develop Datacus and the requirements for a user to run the software.

2.1.1 *Development Environment*

Datacus must be developed using the following environments and APIs:

- Windows 7 Operating System – For development and testing. Also provides the Windows Touch API
- ECLIPSE 3.1.2 – For use with the Adobe Flash Builder 4 plug-in
- Adobe Flash Builder 4 – Adobe’s Flash development environment
- Mondrian 3.2.0 – Pentaho’s open source OLAP engine
- TUIO 2.0 – Touchscreen interface protocol
- 3D visualization APIs (must include at least one of the following):
 - Papervision3D beta 2.0
 - Away 3D V3.5
 - Flash Builder 4 Charts
- Graphing APIs (must include at least one of the following):
 - Flash Builder 4 Charts
 - Degrafa beta 3.1
 - Axiis beta 1.0

2.1.2 *Software Runtime Environment*

Datacus must execute correctly on these platforms:

- Windows 7
- Flash Player 10.1 and above

2.2 Hardware Environment

The Hardware Environment is a list of the minimum system requirements that Datacus must perform optimally on

- Processor: 2GHz or higher Intel or AMD processor
- RAM: 2GB or higher RAM
- Hard Drive: 1GB available disk space
- Monitor: Minimum 1920x1080 pixel 24-bit color multi-touch display of any size (21” or larger recommended)

- LAN or broadband internet connection to any available OLAP database

2.3 Performance

The Performance section lists issues specified for Datacus to perform well on any user's machine.

2.3.1 *Start-Up Time*

The time it takes for the program to fully load and become operational

- Less than 30 seconds

2.3.2 *Speed*

Reacts instantly to touch

- Gesture recognition and response:
 - Objects, menus, and items respond immediately to user touch input
 - Calculates data changes and input touches seamlessly with little to no delay
- Server (database) interaction:
 - Although server connection determines speed, a graphical indicator informs the user of server communication and processing
- Graphs:
 - Reloads within a second of receiving the new modified information from the server

2.3.3 *Capacity*

How much data and how many users Datacus can support at one time

- Only one user supported
- Automatically fetches additional members from the server asynchronously.
- Potentially can support as much data that is available in the cubes on the server. Typically the capacity/memory requirements in Datacus will be heavy on the members / levels / dimensions but since there is only one chart displayed at a time the actual data segments (numerics) should be rather small.

2.3.4 *Reliability*

How well the software will react to "normal" use

- Will not crash from normal use (normal use defined as a series of tests designed to ensure the overall functionality of the software)
- Restarts without causing a fatal error after being used for a set period of time and closed by the user (time to be determined later)

2.4 Functional

The Functional requirements explain how Datacus handles different events. It also explains how the charts (graphs) move and what states within Datacus are.

2.4.1 *Server Interaction Event Data*

Server Events are defined as Datacus communicating with a database using Mondrian for data extraction

- Data pulled in from any OLAP server database specified by the user
- Handles as much data as specified by the user

2.4.2 *Server Interaction Event Display*

An event display is an indication of the software interacting with a database

- A server interaction notification icon revolves and changes color as communication with server is occurring
- Stretch Goal: involves some sort of comet animation that when the return updates instead of just changing color back it “shoots” the things on the screen that are going to change.

2.4.3 *Server Interaction Screen Updates*

Timing and notification of formation of charts and graphs from information stored in a database

- Needs to update information in less than a second after data is retrieved from the OLAP database and loaded into the software
- Visualization indicates when the system is communicating with the server and notifies the user when updates are complete.
- The current chart data goes modal and either dissapears, is greyed out, or has some “refreshing” animation happening while the update is happening.

2.4.4 *Touch Input Event Data*

Touch events are defined as a user interacting with the software interface using their hand

- Recognizes touch input and responds to different gestures (to be determined as prototype develops)

2.4.5 *Touch Input Event Display*

- A cursor follows single point touches to indicate the system is reacting to the user interaction
- Multiple Point touches indicate gestures that react differently depending on items selected by the user

2.4.6 *Touch Input Event Screen Updates*

Graphical reactions to touch by user including gestures and selection

- Actions responding to gestures and touch manipulate the system and change the visualization on the user interface

- Items react differently to touch by users such as
 - Expanding
 - Zooming
 - Selecting

2.4.7 States

Preferences determined by the user of how to represent data selected from the database

- Ability to save and upload “states” of options chosen by the user for viewing data that the user wishes to use later on different data sets
- The outcome is that returning to the previously serialized state object is functionally equivalent to the user having repeated the same set of steps and the outcome/output/visualization should be identical.

2.4.8 Chart Motion

- Allows user to grab either the categories panel or series panel and switch their locations on UI to change values from horizontal chart (graph) to vertical chart (graph)
- Able to switch between chart types with gestures (gestures to be determined later)
- Pinch and expand gestures expand the bottom line of chart into multiple or fewer divisions
 - If nothing selected then the system expands the entire chart over time
 - If one item is selected then the system only expand that item

2.5 User Interface

The User Interface requirements include lists and descriptions of different aspects of the Datacus software that a user will be able to interact with and explains methods how a user should be able to interact. Figure 2 (below) shows a bare mock-up of the user interface including the “sets”, server status, trash can (delete), an expandable picker, the home menu, the graphing area, and the four panels that the user will interact with to manipulate the graphical representations of their data from the user’s database.

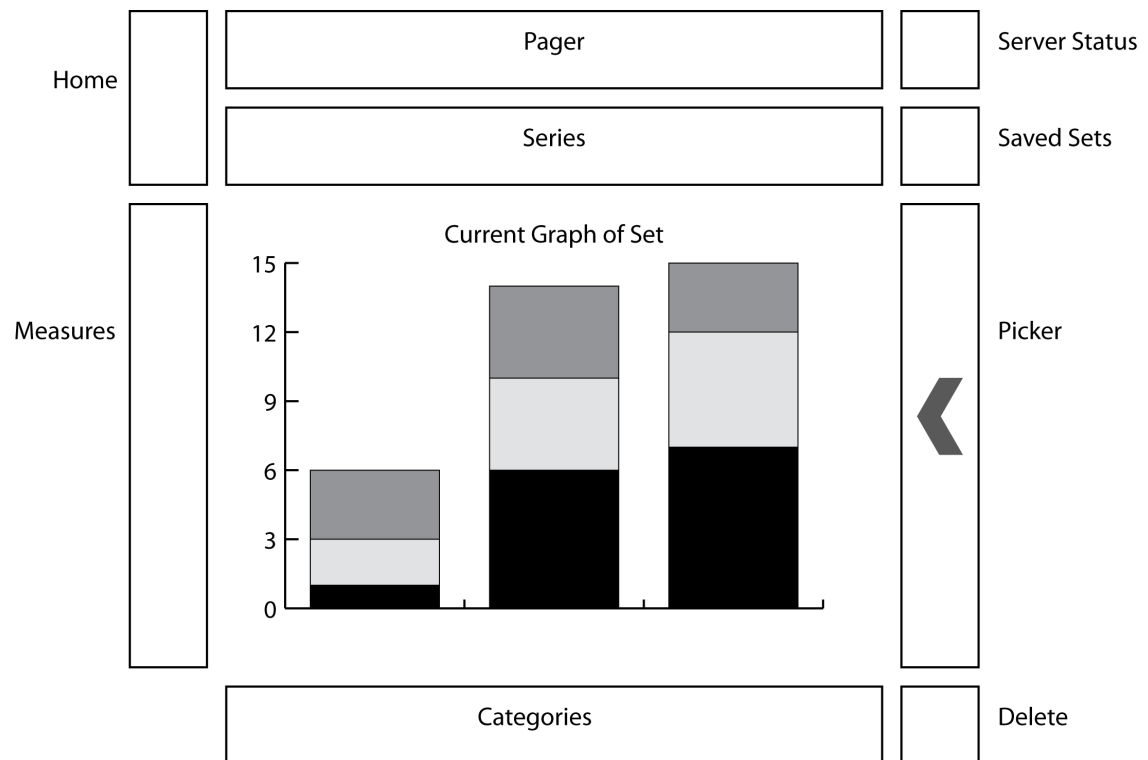


Figure 2. User Interface Mock-Up

2.5.1 General

- English supported as the primary language
- Targeted toward business users
- Users have a visual reference indication communication between software and database
- Users notified of any diagnostic or run time errors such as:
 - Cannot connect to database
- Home menu will include options that are typically associated with a home menu in a software’s interface (exact items to be determined as development occurs)

2.5.2 *Gestures*

- Input method through multi-touch touchscreen monitors
- Takes advantage of previous gestures implemented by other multi-touch devices such as:
 - Pinch
 - Swipe
 - Drag & Drop
 - Double Tap
 - Flick
- Includes new gestures that are intuitive to users for manipulating graphical data (gestures to be determined later and require extensive user input and testing to be considered “intuitive”)

2.5.3 *Selection*

- Ability to select or deselect single items by tapping on them or tapping multiple items in a sequence
- Ability to grab an entire selected group by dragging one of the selected items

2.5.4 *Option Menus*

- A menu that gives the user the ability to choose different options of a certain object
- Activated by the right click gesture (an extended push, touch and hold)
- Graphically represented by a radial menu to give users options such as:
 - Color change
 - Zoom
 - Drill Down (through data)
 - Drill Up (through data)
- All options to be determined throughout the development process

2.5.5 *Trash Can*

- Allows users to drag any item, set of items, or panel of items into the trash to remove the information from the visualization
- Accomplished by “dragging and dropping” selected items into a trash can icon (location to be determined through user testing)
- Should not delay between finger drag and icon (item) drag. Motion should appear fluid on screen
- After removal the server reloads information into visualization to represent a new presentation of the data

2.5.6 *Four Panels*

There are four primarily panels required for the user to interact with the system which are described below

2.5.6.1 *Paging Panel*

- Roulette wheels
 - Can include a single level, or multiple levels from same or different dimensions
 - Can be Time Dimensions, or other dimensions
 - If multiple levels in the same dimension are selected (say, Year and Month) then moving the Year wheel to 2008 will adjust the Month wheel so it has only the months under 2008. This is similar to the Toolbar picker.
- Play (go) Button (ideal stretch for scope of project)
 - Enabled only if one level is selected on paging panel
 - Pressing play grabs all information from the database for the time period and product selected (next 30 objects)
 - Represented as a pause button during processing
 - Include option to go forward and back (frame by frame)

2.5.6.2 *Series Panel*

- Represents information comparable by users (time, profit, etc.)
- Picked from wheel on the “picker”
- These are the items retrieved from the OLAP database

2.5.6.3 *Categories Panel*

- Comparable items can be dragged onto this bar
- Picked from child-parent chart inside the “picker” (see picker description below)
- These are the items retrieved from the OLAP database

2.5.6.4 *Measure Panel*

- Represents quantity of numbers
- Not able to drag items onto this panel
- Should have some way to select which measure to visualize (such as Total Sales, or Average Sale Price)

2.5.7 *Picker*

- Expands over the graphing area to give the user a central location for deciding what data should be included in the visualization
 - Collapsible animation for both expanding and closing
 - Be expanded and closed by a “flicking” gesture

- Semi-transparent to allow user to still be able to view the chart/graph under it
- When expanded have a circular “dimension” selection attached to one side (top or side)
 - Allows users to select what categories they want to see about their data
 - Drag these options into the “Series Panel”, “Category Panel”, or “Paging Panel”
 - Able to flip through dimension options by rotating a wheel
 - The categories are attached to the Toolbar
 - When flipping through dimensions, 2D tree updates to new available data
- 2D trees used to allow for choosing what data from the user’s database will be added to the report.
 - Only items at the same level of the trees can be chosen for comparison (can’t compare parent and child nodes together)

2.5.8 “Sets” (Ideal stretch for project)

- “Sets” Creator
 - Folders of commonly selected items for visualization data
 - Drop items from panels or from picker into a folder icon on toolbar to create a set
 - Keyboard (virtual or physical) required to save the set as a name
- “Sets” Manipulator
 - Expandable menu to allow choosing a previously saved set
 - Editable based on right click equivalent gesture (to be determined later)
 - Paging panel and series panel flip to display all items in the set and the “Picker” expands to allow for adding new items into the set

2.6 Documentation

These requirements specify the documentation, both documents and presentations, that will be provided and how it will be created.

2.6.1 Documents

- Specific content of each document will be determined later
- Development documents: Requirements, Design, Test Plan, Release Notes
- User documents: User References, User Tutorials (Videos and Interactive Walkthroughs)
- Provide PDF, Word Documents, and printed versions of all documents

2.6.2 Presentations

- Overview, State of the Project, Final Demo
- All presentations created in MS PowerPoint or Keynote
- Provide PDF and printed versions of all presentations

2.6.3 Wiki System

- English
- Hosted Documentation and Help Files
- Audiences for documentation: Troubleshooting and Tutorials

2.7 Release

Release requirements explain issues related to the release and delivery of the final product.

- End-user releases provided as well as open-source code for developer release
- Files made available as .tgz and .zip files via Github
- Documentation distributed via Wiki System and downloaded through Github

3 SUMMARY

This document describes the requirements set for the Datacus software package. This includes requirements related to the development environment in which the software is to be developed, the hardware environment in which Datacus runs, the functionality that the software must provide, performance standards required of the software, user interface requirements specified by the sponsor DyanamoBI, supporting documents that must be written, and the manner in which the software package is to be delivered including the code, the application, and all documentation. This document will evolve as the project progresses and new requirements are specified.

4 REFERENCES

There are a few documents related to this paper that are useful for further reading on certain topics.

[BlazeDS 10]

"BlazeDS." Wikipedia. June 7, 2010. Wikimedia Foundation, Inc. (October, 2010)
<<http://en.wikipedia.org/wiki/BlazeDS>>

[Degrafa 08]

"Degrafa : Declarative Graphics Framework." 2008. Degrafa Team. (October, 2010)
<<http://www.degrafa.org/>>

[Flash App 10]

"Flash Application Design: Understanding Flash applications." adobe.com. Adobe Systems, Inc., 2010. (October, 2010)
<http://www.adobe.com/support/flash/applications/app_design/app_design04.html>

[Flash Server 10]

"Flash Application Design: Using an application server." adobe.com. Adobe Systems, Inc., 2010. (October, 2010)
<http://www.adobe.com/support/flash/applications/app_design/app_design06.html>

[Flex 10]

"Source of DynamicChart." 2010. Flex Blog. (October, 2010) <<http://www.flex-blog.com/samples/sample13/srcview/index.html>>

[olap4j 10]

"olap4j: Open Java API for OLAP." 2007-2010. olap4j Team. (October, 2010)
<<http://www.olap4j.org/>>

[Tuio 10]

iTuio as3 library. 2010. Tuio Flash Blog. (October, 2010)
<<http://bubblebird.at/tuioflash/tuio-as3-library/>>