Defence Research and Development Canada | Recherche et développement pour la défense Canada

DEFENCE R&D DÉFENSE

Design of a Maritime Domain Awareness Visual Analytics Prototype (DMVAP) Task 2: Concept Exploration

Chris Venour, David Roodnick

Prepared By:

Salience Analytics Inc.
87 West 41 Ave
Vancouver, BC, Canada, V5Y 2R8
Tel: 604-790-3771
www.SalienceAnalytics.ca

PWGSC Contract No.: W7701-101843
DRDC Scientific Authority: Valérie Lavigne, (418) 844-4000 ext. 4114
Salience Document Number: SAI-11-04R

Defence R&D Canada – Valcartier

Contract Report

DRDC Valcartier CR 2011-222

July 2011

Canada

SAI-11-04R

(UNCLASSIFIED)

Design of a Maritime Domain Awareness Visual Analytics Prototype (DMVAP)

Task 2: Concept Exploration

2011 July

Chris Venour   David Roodnick

Salience Analytics Inc.

DRDC Valcartier CR 2011-222

PWGSC Contract Title:	Design of MDA Visual Analytics Prototype
PWGSC Contract No.:	W7701-101843
DRDC Document Number:	DRDC Valcartier CR 2011-222
DRDC Project Manager:	Valérie Lavigne, (418) 844-4000 ext. 4114
DRDC Scientific Authority:	Valérie Lavigne, (418) 844-4000 ext. 4114
Project Period:	Dec 2010 – July 2011
Salience Document Number:	SAI-11-04R

Salience Analytics Inc.
87 West 41 Ave
Vancouver, BC, Canada, V5Y 2R8
Tel: 604-790-3771
www.SalienceAnalytics.ca

Salience

(UNCLASSIFIED).

Design of a Maritime Domain Awareness Visual Analytics Prototype (DMVAP)

Task 2: Concept Exploration

2011 July

Chris Venour David Roodnick

Salience Analytics Inc.

DRDC Valcartier CR 2011-222

PWGSC Contract Title:	Design of MDA Visual Analytics Prototype
PWGSC Contract No.:	W7701-101843
DRDC Document Number:	DRDC Valcartier CR 2011-222
DRDC Project Manager:	Valérie Lavigne, (418) 844-4000 ext. 4114
DRDC Scientific Authority:	Valérie Lavigne, (418) 844-4000 ext. 4114
Project Period:	Dec 2010 – July 2011
Salience Document Number:	SAI-11-04R

Salience Analytics Inc.
87 West 41 Ave
Vancouver, BC, Canada, V5Y 2R8
Tel: 604-790-3771
www.SalienceAnalytics.ca

Salience

Written by:	Chris Venour Analyst	______________________ signature and date
	David Roodnick Programmer	______________________
Checked by:	Mike Davenport Principal Investigator	______________________
Approved by:	Mike Davenport Project Manager	______________________

Indexing Information

Salience Document Number	SAI-11-04R
Client Name	DRDC Valcartier
Client Reference Number	DRDC Valcartier CR 2011-222
Keywords	Maritime, DRDC, Visual Analytics, Google Earth, Processing, Android, Tablet

A Note on Trademarks

Abstract

This document summarizes the techniques used to mock-up and showcase new Visual Analytics concepts in support of the design of a Maritime Domain Awareness Visual Analytics Prototype. Static mock-ups were primarily done using the drawing tool in MS Word and standard Graphical User Interfaces were mocked up in MATLAB. Dynamic map-based mock-ups of Word Clouds and a Time Slider on Ship Tracks were done in Google Earth and Google Earth Plugin. A collaborative version of Google Earth was also demonstrated. To mock up the more complex Predictive Rendezvous app, code was written in Processing, a Java-based rapid prototyping tool. The mock-up runs as a Android app as well as on a laptop computer. A simple Situational Facts Knowledge Base was mocked up in OWL using Protégé.

Résume

Ce document résume les techniques utilisées pour la création de maquettes et la présentation de concepts nouveaux d'analyse visuelle en support aux travaux de conception du prototype d'analyse visuelle pour la connaissance de la situation maritime. Les maquettes statiques ont été principalement réalisées à l'aide des outils de dessin de MS Word et les maquettes d'interfaces graphiques conventionnelles ont été créées avec MATLAB. Les maquettes dynamiques centrée sur une carte pour les nuages de mots et le curseur temporel pour les pistes de navires ont été réalisées avec Google Earth et le plugin Google Earth. Une version collaborative de Google Earth a également été démontrée. La maquette plus complexe de rendez-vous prédictifs a été programmée en Processing comme outil de prototypage rapide en Java. Les maquettes s'exécutent en tant qu'applications Android, ainsi que sur des ordinateurs portables. Une maquette simple de la base de connaissances des faits situationnels a été créée en utilisant OWL Protégé.

Executive Summary

The design of a Maritime Domain Awareness Visual Analytics Prototype contract seeks to explore and demonstrate how Visual Analytics (VA) and collaborative technologies can help Canadian Forces (CF) and Maritime Security Operations Centre analysts monitor the maritime approaches to North America. Analysts currently struggle under a deluge of data from sensors, voluntary reports, and intelligence sources.

This document summarizes the techniques used to mock-up and showcase new and innovative Visual Analytics "apps" in support of the prototype design. Particular attention was paid to mock-ups made in Google Earth (GE) and Google Earth Plug-In, because GE provides a simple, free, and widely-used visualization platform for map-based VA, and because it is already used in the CF operations centers. Google Earth Plug-In was found to work better because it provides better support for dynamic visualizations and more control over the user interface.

Word Clouds and Time Slider on Ship Tracks apps were mocked up in Google Earth Plug-In and are available for hands-on evaluation with the project deliverables. GE was flexible and visually rich, but proved to be slow and jerky when displaying the movement of large numbers of ship contacts. A collaborative version of Google Earth was also demonstrated, in which geographically remote users share a view and can jointly insert information, change the view, and chat.

To assess a novel Predictive Rendezvous App for visualizing threats a smooth and very responsive mock-up was required, so code was written in *Processing*, a Java-based rapid prototyping tool. Tests were done on a laptop, using a mouse interface, and on a Motorola Xoom tablet personal computer, using a touch screen interface. User comments were positive in both contexts.

This document also briefly reviews a simple maritime ontology that was created in OWL using Protégé to represent the Situational Facts Knowledge Base. The ontology was plotted using Graphic OWL and integrated into static mock-ups of the Facts on the Map and Visual Rationale apps.

Table of Contents

Table of Figures

List of Tables

Acronyms and Abbreviations

ASDK	Android Software Development Kit
API	Application Programming Interface
CF	Canadian Forces
DMVAP	Design of a Maritime Domain Awareness Visual Analytics Prototype
DOS	Disk Operating System
DRDC	Defence Research and Development Canada
GE	Google Earth
GPW	Global Position Warehouse
GrOWL	Graphic OWL
GUI	Graphical User Interface
HTML	Hypertext Markup Language
KML	Keyhole Markup Language
MATLAB	Matrix Laboratory (software)
MDA	Maritime Domain Awareness
MSOC	Maritime Surveillance Operations Centre
MS Word	Microsoft Word
OWL	Web Ontology Language
PC	Personal Computer
RJOC	Regional Joint Operations Centre
SFKB	Situational Facts Knowledge Base
USB	Universal Serial Bus
VA	Visual Analytics
VAMDA	Visual Analytics for Maritime Domain Awareness

1 Introduction

This is the report on Task 2 of contract W7701-101843, "Design of a Maritime Domain Awareness Visual Analytics Prototype". This contract is part of a larger DRDC Valcartier advanced research project ARP 11hm, "Maritime Domain Analysis through Collaboration and Interactive Visualization" also known as "Visual Analytics for Maritime Domain Awareness" (VAMDA).

Canadian Forces (CF) and other government analysts at the Regional Joint Operations Centres (RJOCs) and the Maritime Security Operations Centres (MSOCs) are tasked with finding rare but critical evidence with a deluge of mundane data. Research project ARP 11hm seeks to explore and demonstrate how Visual Analytics (VA) and collaborative technologies can [10]:

- Help RJOC and MSOC analysts rapidly grasp and identify key information elements and thus gain insight into current operational situation,
- Improve the visualization of the recognized maritime picture,
- Help analysts better comprehend a situation and anticipate how it could develop,
- Help analysts identify what is not known (data gaps and uncertainty),
- Support detection, alerting, and visualization of anomalies,
- Enable collaborative team work.

1.1 Objectives

The objective of this contract is to produce a high-level design for a Visual Analytics prototype to support Maritime Domain Awareness (MDA). Specific activities include:

- **Design Study:** identify the scope, required functionality, and applicable concepts and techniques for the four VA prototype capabilities listed in the Design Study [3];
- **Exploration of VA Concepts:** showcase or mock-up a subset of the proposed VA concepts and techniques (this document);
- **System Architecture:** identify the VA system components that will be required to build and deploy the prototype (see [1]); and
- **Scenario and Dataset Development:** assemble or create datasets that could be used to validate a future prototype (see [2]).

1.2 Mock-Ups and Concept Exploration

This document provides an overview of how information representation techniques and Visual Analytics concepts were showcased or mocked up as part of the Design Study, and describes how the interactive mock-ups can be run. Four approaches have been used:

- **Static Screen Mock-Ups:** the majority of mock-ups were drawn using MS Word's drawing tools, with image elements (e.g. icons) brought in from public sources. These Mock-Ups can be copied and modified in the digital versions of the Design Study [3]. Whenever large image elements are used in a mock-up, the source is referenced in the caption of that figure.
- **Graphical User Interface (GUI) Mock-Ups:** conventional GUI screens in the Design Study [3] were mocked up using MATLAB's GUI Development Environment (GUIDE). Source code files for these mock-ups have file extension ".m" and are included in the Design Study document directory of the distribution disk.
- **Google Earth (GE) Mockups:** dynamic and interactive mock-ups of some of the map-based apps and pop-ups were produced using Google Earth. Sections 2 and 3 describe how the mock-ups were implemented, and lessons learned.
- **Predictive Rendezvous Concept Exploration:** the novel VA concept behind the Predictive Rendezvous App is fundamentally interactive, so static mock-ups (e.g. Figure 4.22 of [3]) proved to be inadequate. We therefore mocked up an interactive proof-of-concept using *Processing*, as described in Section 4.
- **Ontology Mock-Up of the Situational Facts Knowledge Base (SFKB):** the simplest way to illustrate how the Facts on the Map app and the Visual Rationale app will interact with the SFKB was to mock-up a small maritime ontology, as described in Section 5.

2 Google Earth Mockups

The possibility of implementing some VA apps using Google Earth (GE) is appealing for at least the following reasons:

- GE is free and very widely used;
- Data can be passed to GE efficiently through KML files; and
- The RJOCs already used GE.

Subsections 2.1 through 2.3 describe Google Earth mock-ups of the following apps:

- Word Clouds (Section 2.1)
- Time Slider on Ship Tracks (Section 2.3).

Subsection 2.4 examines how Google Earth might be used to enable collaboration.

2.1 Creating the Google Earth Mock-Ups

The Google Earth demo data was created as follows:

- Vessel information was extracted from the Global Positioning Warehouse (GPW) and saved it as a comma separated value file (xyz.csv);
- Perl script `create_fast_time_shipping_kml.pl` (available on the `GoogleEarth` directory on the mock-ups disk) was used to generate the KML data for the Time Slider on Ship Tracks demo. The script was executed using the following DOS command line:

```
C:> create_fast_time_shipping_kml.pl xyz.csv > 200shipswordclouds.kml
```
(2-1)

- Perl script `create_word_clouds_kml.pl` (also available on the `GoogleEarth` directory on the mock-ups disk) was used to generate the KML data for the Word Clouds demo. The script was executed from the DOS command line as follows:

```
C:> create_word_clouds_kml.pl xyz.csv > demo2.kml
```
(2-2)

- A few lines of KML code was inserted by hand at the top of both kml files.
- The files wordcloudsfinal.html and fastshipping.html, which include specialized JavaScript code, were written by hand based on a Dutch website [12].

Table 2-1 illustrates the translation done by the Perl scripts. The top row shows GPW data regarding the vessel “British Trader” and the bottom row shows the corresponding KML used by the GE plugin to display the vessel’s movement. Names and positions of vessels were taken from the GPW file but other data such as cargo, flag and destination information, were fabricated for the purpose of the demo.

Table 2-1 Transforming GPW data into KML

GPW data	0,BRITISH TRADER,IM,MER,,453698,031908ZNOV09,Y,040859ZNOV09,235496000,T536 8,,UNEQUATED,,,18,CMN623949370,ZIPR7,23.2,-69.5,031800ZNOV09,180,15,OSWEX, OTR,08:00.0,N,,,0,0,IMO 9238038 MMSI 235496000,NOT USING ICS,,,, 0,BRITISH TRADER,IM,MER,,454693,041913ZNOV09,Y,050427ZNOV09,235496000,T494 5,,UNEQUATED,,,18,CMN623951867,ZIPR7,17.9,-67.6,041800ZNOV09,180,15,OSWEX, OTR,13:00.0,N,,,0,0,IMO 9238038 MMSI 235496000,NOT USING ICS,,,, 0,BRITISH TRADER,IM,MER,,463001,161906ZNOV09,Y,170809ZNOV09,235496000,T409 8,,UNEQUATED,,,18,CMN626507670,ZIPR7,17.1,-51.5,161800ZNOV09,45,15,OSWEX,O TR,06:00.0,N,,,0,0,IMO 9238038 MMSI 235496000,NOT USING ICS,,,,
KML equivalent	<Placemark id="BRITISH TRADER"> <name>BRITISH TRADER</name><description>BRITISH TRADER,military,US,St John</description> <styleUrl>#icon0</styleUrl> <gx:Track> <when>2009-11-03T18:00:00Z</when><gx:coord>-69.5 23.2</gx:coord> <when>2009-11-04T18:00:00Z</when><gx:coord>-67.6 17.9</gx:coord> <when>2009-11-16T18:00:00Z</when><gx:coord>-51.5 17.1</gx:coord> </gx:Track> </Placemark>

2.2 Google Earth Mockup: Word clouds

The Word Cloud demo can be run from a browser by opening the following file:

- http://salienceanalytics.ca/chris/wordcloudsfinal.html

The demo plots the movement of 200 vessels over a period of a month and displays information about these vessels in the form of word clouds, as shown in Figure 2-1. The word clouds float above the ship icons, and move with the ships as the simulated time changes.

Radio buttons at the top of the demo allow the user to display different information about a vessel – its name, cargo, flag or destination. A time slider at the bottom of the page allows the user to skip to a certain time in the animation and the buttons below the slider start and stop the demo and set the rate of the animation. The animation can run in real time – where 1 second in the animation corresponds to 1 second of real time – or the user can speed up the animation to as much as 4 hours per second by clicking on the “faster” button.

This demo illustrates how dynamic the Google Earth plugin can be. When the main file `wordcloudsfinal.html` is clicked in a web browser, the KML file of vessel information is loaded into the browser. Javascript code in the HTML page makes use of a library of functions provided by the plugin to parse and alter the KML data whenever the user selects a radio button to display different information about vessels.

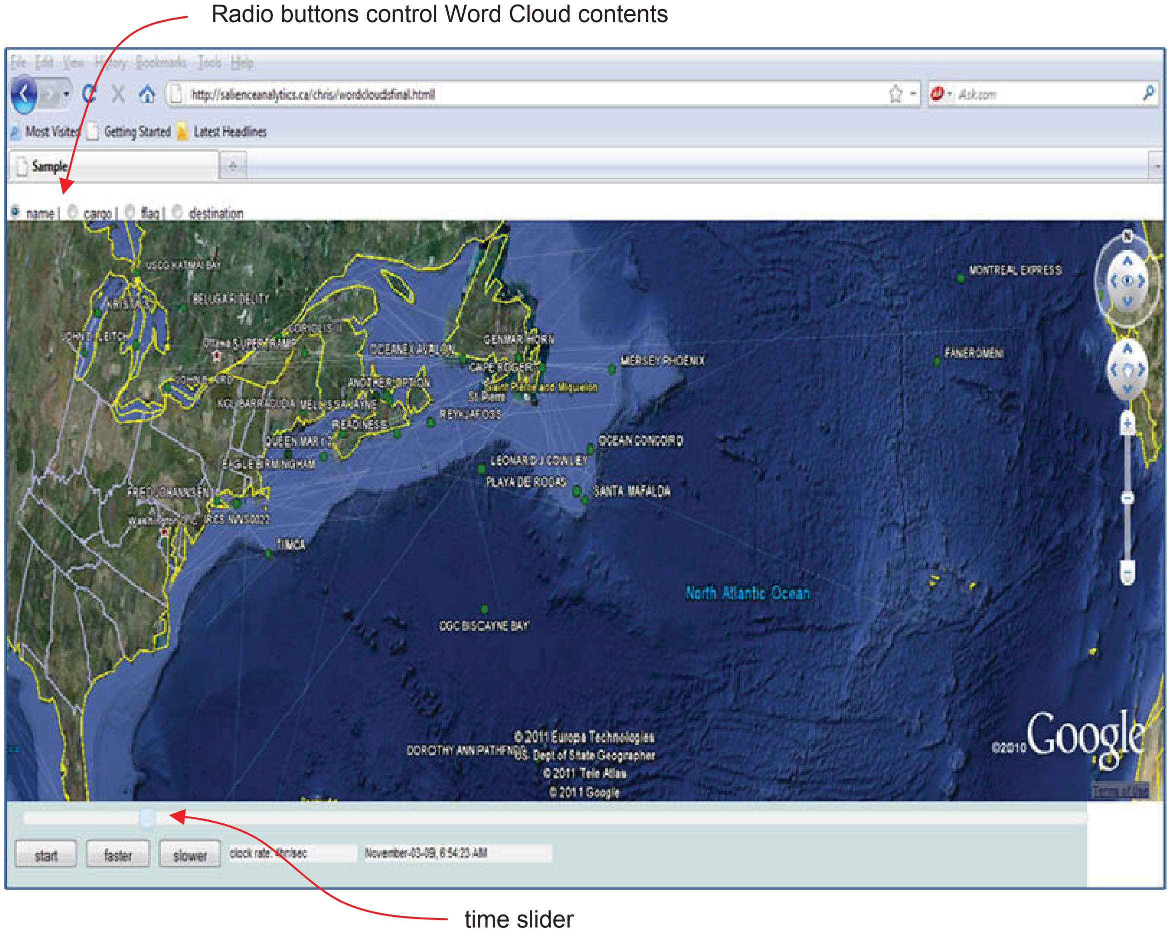

Figure 2-1 Word clouds demo

Word clouds float near vessel icons in the Google Earth view of the North Atlantic. Ship names are shown because the operator has clicked on “name” in the top left bank of radio buttons. The operator has also asked for an animation speed of 4 h/sec.

2.2.1 Description of files

The webpage wordcloudsfinal.html contains all the necessary code for the demo. It makes use of two css files to define the layout and style of the HTML page, and two JQuery libraries are used to help create the time slider functionality. All the necessary files and libraries are listed in Table 2-2, and they have been copied to the distribution disk.

Table 2-2 Files for Word Clouds demo

Function	File(s)
The main file of the demo	`wordcloudsfinal.html`
The KML file containing all the vessel information (positions, names, cargo, flag, destination)	`kml/200shipswordclouds.kml`
Files pertaining to the look of the webpage wordcloudsfinal.html	`css/stylesheet.css` `css/jquery-ui-1.8.6.custom.css`
JQuery library files used by time slider code	`java/jquery-1.4.2.min.js` `java/jquery-ui-1.8.6.custom.min.js`
An image representing vessels	`images/green.png`

2.3 Google Earth Mockup: Time Slider on Ship Tracks

Hundreds of vessels travel the Atlantic at any given time and so a way of flagging seasonal or daily patterns, or patterns pertaining to ship class or some other variable, might be useful. The demo at http://salienceanalytics.ca/chris/fastshipping.html aims to demonstrate how the Google Earth plugin might be used to visualize maritime activity and perhaps detect patterns in traffic.

Vessels are represented only by coloured dots – in this example the colour of a vessel is based on the type of cargo it is carrying – and these dots are unlabelled in order to reduce clutter that might distract from the detection of patterns. Figure 2-2 shows a snapshot of the mock-up.

The HTML and Javascript code for this demo are almost identical to that of the Word Clouds demo except that the programming enabling different word clouds to be displayed has been removed.

The dataset in the Time Slider on Ship Tracks demo is much larger than the data displayed in the Word Clouds demo, however: half a million lines of GPW data, which have been transformed into KML, are plotted into the plugin compared to the 60,000 lines of data processed by the Word Clouds demo. As the large (34 Mb) data file for the Fast Time Shipping demo is loaded into the instance, a warning message appears in one of the windows at the bottom of the web page, informing the user to wait until the file has fully loaded before starting the demo.

All the files and libraries used by this program are listed in Table 2-3, and have been delivered with the mock-ups disk.

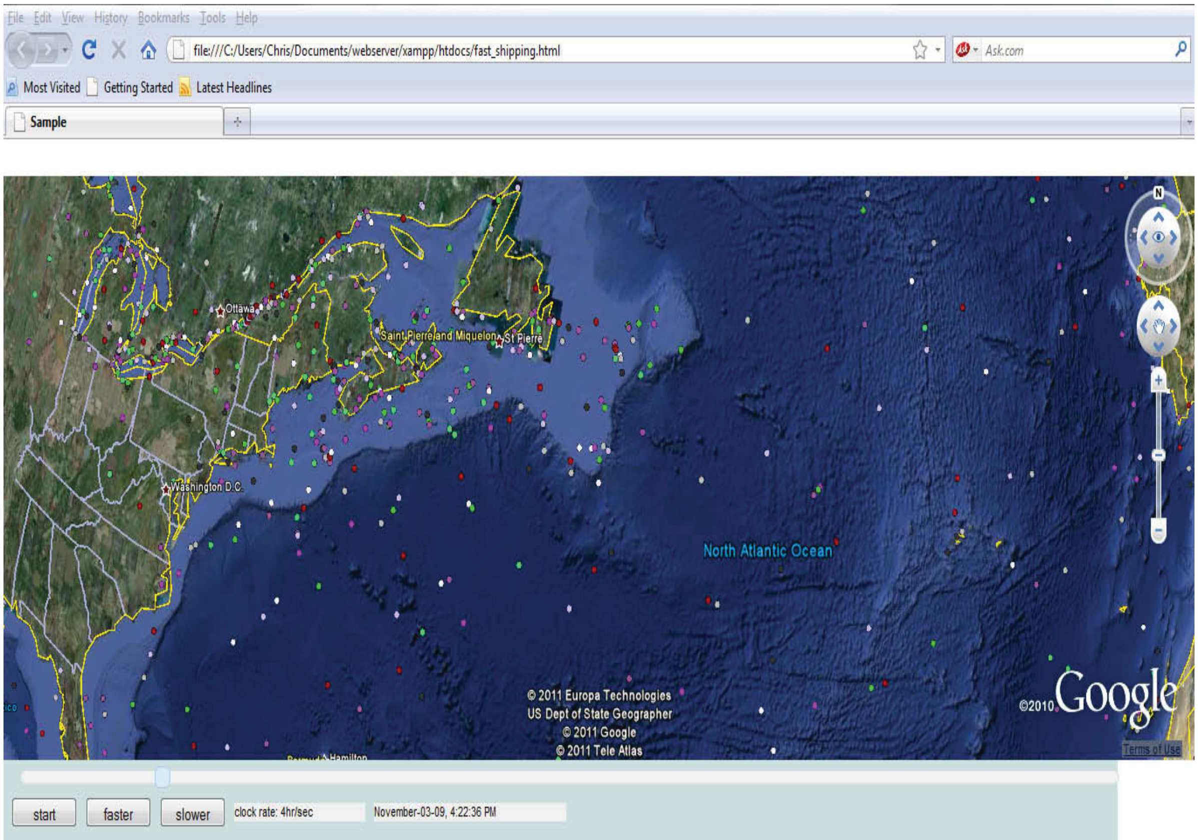

Figure 2-2 Time Slider on Ship Tracks

Vessels are represented by coloured dots and are unlabelled to reduce clutter and thus help reveal patterns in maritime traffic.

Table 2-3 Files for the Time Slider on Ship Tracks demo

Function	File(s)
The main file of the demo	`fastshipping.html`
The KML file containing all the vessel information (positions, names, cargo, flag, destination)	`kml/fast_time_shipping.kml`
Files pertaining to the look of the webpage wordcloudsfinal.html	`css/stylesheet.css` `css/jquery-ui-1.8.6.custom.css`
JQuery library files used by time slider code	`java/jquery-1.4.2.min.js` `java/jquery-ui-1.8.6.custom.min.js`
An image used in the demo	`images/white.png`

2.4 Google Earth Concept Exploration: Collaboration

The GE plugin might be used to create a space for collaboration and the following demo, written by a Google employee, is an example of what this might look like:

https://www.google.com/accounts/ServiceLogin?service=ah&continue=http://earthpad.appspot.com/_ah/login%3Fcontinue%3Dhttp://earthpad.appspot.com/&ltmpl=gm&ahname=Earth+Pad&sig=9ecd6519fbe98be83eafe1e25754f013

(Note that a Google account is required to view this demo. Figure 2-3 displays a snapshot of the demo).

A user logs in to his Google account and creates a GE instance when he clicks on the link above. Other users can then join this same instance when they are provided with a url which uniquely identifies the instance. Any event that occurs in a user's view gets replicated across the network to the other instances, so if a user changes the view from New York to London, for example, the view of all the other users sharing the instance will also change to London.

Similarly placemarks added to the instance by a user are viewable to everyone sharing the space. Analysts can also send messages to each other using a chat facility that has been provided.

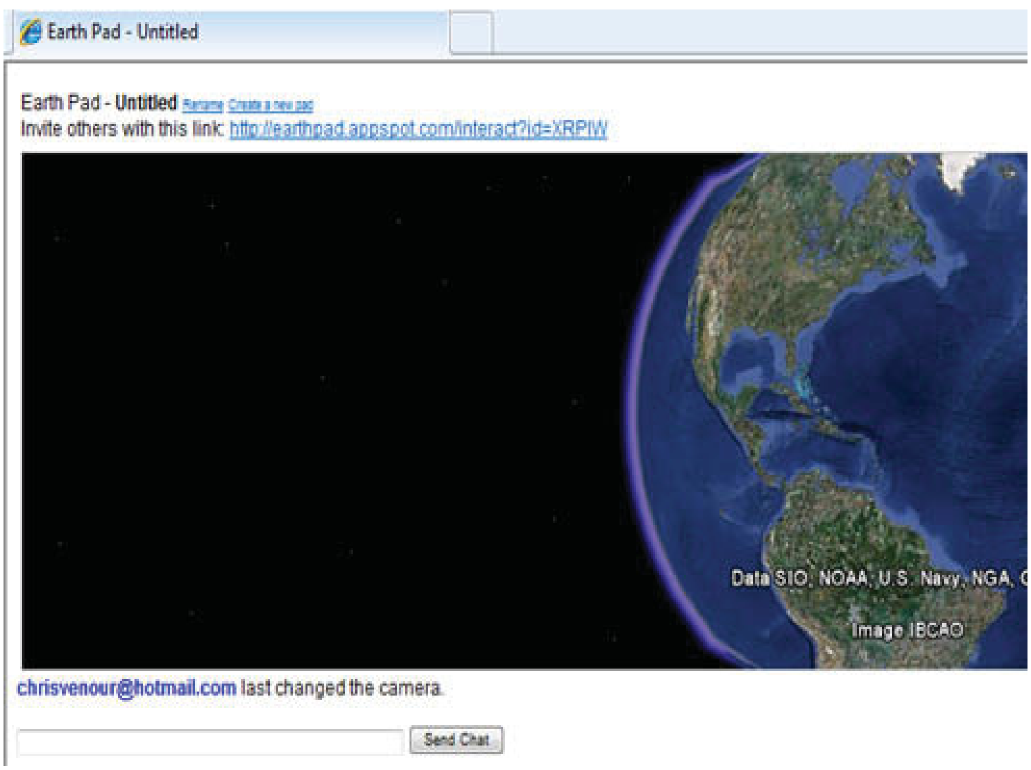

Figure 2-3 Collaborative Google Earth demo
Multiple users can share the same GE plugin instance. When a user makes a change to his instance, that change is propagated to all the other users' instances. Users can communicate using the Chat button at the bottom.

Different instances of the Google Earth plugin, each of which providing different graphical interfaces and tools to analysts, could appear on a single web page, as shown in Figure 2-4. A particular instance, such as one providing a space for collaboration, could easily be maximized to fill the entire web page and minimized, as shown in Figure 2-4, when the analyst has completed using the tool.

Figure 2-4 Multiple Plugin Instances on a Single Web Page
Multiple GE plugin instances, which provide separate tools and interfaces, can appear on a single web page. One of the instances pictured above could be a space for collaboration.

2.5 Limitations of Google Earth

The main limitations of Google Earth for the MVAP are:

- Updates are slightly too slow – the time slider for example does not smoothly move the ship tracks, and the animated view is jumpy;
- It is necessary to be connected to the internet to make it work, and it thus may be blocked by DND servers; and
- There is too little flexibility in how data is displayed.

3 Guide to the Google Earth Plugin

The Google Earth plugin embeds the functionality of the Google Earth application into a web page. The plugin provides an application programming interface (API) that a Javascript developer can use to create a powerful Google Earth application. The plugin API is divided into two main classes [7]:

- a KML class containing the KML elements that run in the regular Google Earth application. Thus all of the functionality available in the regular Google Earth application is available in the plugin: KML elements and objects such as placemarks, lines, paths, overlays, 3D models, tours and animation sequences are supported in the plugin environment. In fact a KML file which runs in the regular Google Earth application can be loaded and displayed in the plugin without alteration.
- a plugin-specific class that allows programmers to, for example, set the rate at which the clock in an animation ticks, change the position of the Google Earth camera or even parse a KML file that has already been loaded into the browser and make changes to it.

Extension libraries which expand the functionality of the API have also been built for developers [8]. These libraries provide additional functions which, for example, enable users to edit lines and polygons that have been drawn on the screen. Other functions perform useful mathematical operations such as calculating the distance between two points, computing a bearing from point A to point B, determining the area of a polygon drawn on the screen or determining if a point is within a polygon.

3.1 How to Create an Instance of Plugin in a Web Page

In order to create an instance of the Google Earth plugin, you first of all need to install the plugin from http://code.google.com/apis/earth/. The regular Google Earth application will also need to be installed on your machine.

Create an instance of the Google Earth plugin in a web page as follows:

- Load the Google Earth API.
- Create a DIV element to hold the plugin.
- Create functions to initialize the plugin.
- Call the initialization functions once the page has loaded.

Appendix A provides a complete example of HTML code performing these steps and Figure 3-1 shows the simple instance that results from this code.

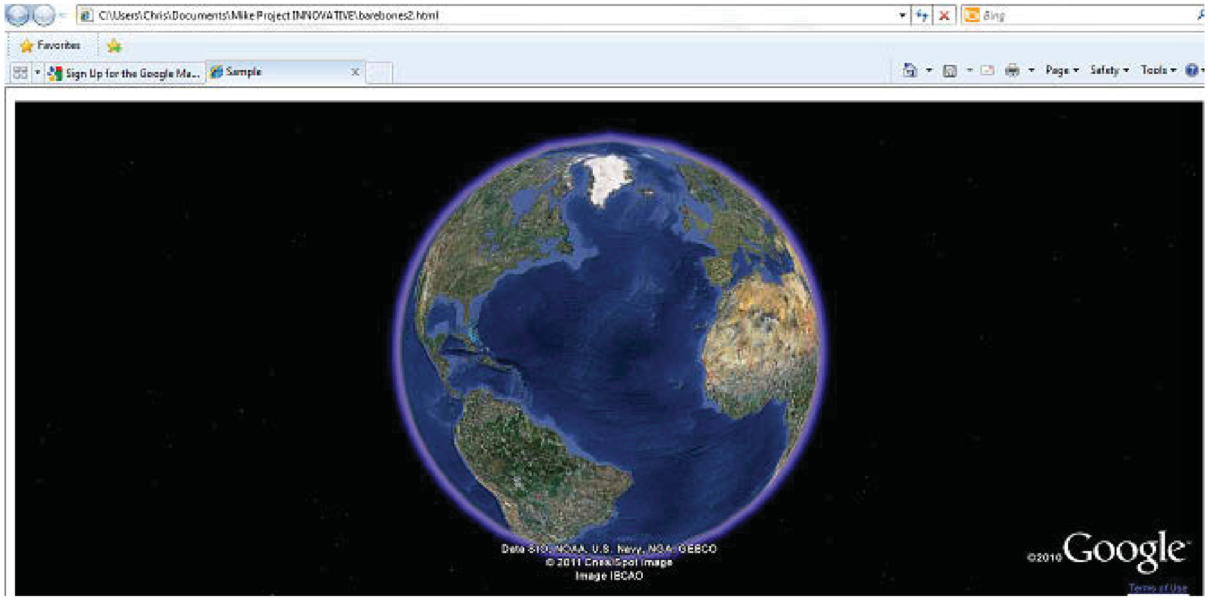

Figure 3-1 Bare bones version of the Google Earth Plugin
This is the simplest instance of the Google Earth plugin display.

In this simple instance, a user can only move to different locations and zoom in or out – the left sidebar and top menu of functions available in the regular GE application are not provided, although the functionality found in these menus could be added to the instance using Javascript and the plugin's API.

3.2 Advantages of Plugin over regular Google Earth

The advantages of the Google Earth Plug-In over web-based Google Earth include:

The Developer Defines the User Interface: The Google Earth Plugin gives the developer complete control over what kind of interface and tools he would like to provide to users whereas the interface and functionality provided in the regular Google Earth application are fixed and cannot be altered or expanded upon.

Much Richer and Interactive Applications**:** The plugin can provide a much richer and more dynamic experience for users than the regular Google Earth application. Making use of the plugin API, a Javascript programmer can, for example, write code so that if a user hovers the mouse over a vessel, a geodesic is drawn from that vessel to its destination port. Javascript code could also enable, for example, a user to add an overlay to a group of moving vessels and have the boundaries of that overlay stretch or shrink as vessels in the group move.

Simpler Implementation: To change, delete, or add KML elements to a previously loaded file in regular Google Earth is extremely cumbersome. No fewer than four separate KML files (plus a program which determines which updates should be made), would be required and all of this code would need to work in tandem and communicate to each other across a network. All that is required to perform these actions in the plugin, however, is some Javascript code running in the browser (see Chapter 17 of [13]).

3.3 Weaknesses of Google Plugin

The plugin is relatively new technology - it was introduced in May 2008 - and there are important omissions in the library. For example:

Position of a Moving Placemark: Google Earth will move a placemark smoothly along a specified path in response to time changes specified by the time slider or the animation. Unfortunately there is no API function to access the current location of a moving placemark. This can be worked around by having the Java Script do its own interpolation based on the current time (which can be accessed).

Confusing Time Objects: Google Earth uses two different objects - *timestamp* and *timespan* to indicate time. Strangely, one cannot anticipate which of these two objects will be returned when a request for the current time is made and so code such as that displayed in Table 3-2 has to be written to handle both possibilities. The plugin documentation provides an example on how to retrieve the current time (using code similar to that shown in Table 3-2) but it offers no explanation as to why this bizarre ritual has to be performed.

Mediocre Documentation: Google has provided an excellent gallery [5] of nearly 100 demos complete with source code. Unfortunately some important bugs are not mentioned in the Google Earth Plugin documentation[6]. Those were found only after posting a question in the Google Earth Plugin forum and receiving responses from other developers that they too had encountered the problem.

Table 3-1 Oddly Complicated Code Required to Extract Time in GE Plugin.

```
var geTimestamp = ge.getTime().getTimePrimitive();
if (geTimestamp.getType() == 'KmlTimeSpan')
{
   var timeStringUTC = geTimestamp.getEnd().get();
}
else
{
   var timeStringUTC = geTimestamp.getWhen().get();
}
```

3.4 Limitations of Google Earth Plugin

Although Google Earth Plugin is a much more powerful visualization environment, it inherits two key limitations of Google Earth:

- Updates are still too slow – the time slider for example does not smoothly move the ship tracks, and the animated view is jumpy;
- It is still necessary to be connected to the internet to make it work, and it thus may be blocked by DND servers.

4 Guide to the Predictive Rendezvous Mockup

The Predictive Rendezvous mock-up and concept exploration was implemented using *Processing*, a rapid prototyping language based on Java that is available for free [4].

4.1 User Guide

Once *Processing* has been installed, launch the Predictive Rendezvous mock-up as follows, and as shown in Figure 4-1:

a) Click on the Processing icon. This will open a new window with a menu.

b) Use the menu to browse to file Intercept.pde in the distribution disk.

c) Open Intercept.pde to open a new Processing window with the demo code in it.

d) Click on the "play" icon to run the demo.

When the demo runs it immediately launches the main screen shown in Figure 4-2(a). Two elements of the display can be controlled:

- Use mouse or finger to smoothly slide the time and see where the three ships might be at some future time.
- Click on the aircraft icon to determine when an aircraft would have to leave Comox to rendezvous with the Alaskan Explorer at the designated time.

Click on the standard MS Windows top-right-corner "x" to close the window.

4.2 Porting to a Tablet Computer

It proved to be very easy to run Intercept.pde or any *Processing* apps on an Android powered device following the clear instructions on the Processing Wiki[14]. We tested and demonstrated Intercept.pde on a Motorolla Xoom tablet PC running Android.

The procedure can be summarized as follows:

- Install and run the Android software development kit (ASDK) on the PC;
- From the ASDK, download a special version of *Processing* with Android support;
- Run *Processing* and select "Android" mode from the drop-down box in the toolbar (note in Figure 4-1(e) that *Processing* is still in "Standard" mode);
- Load Intercept.pde
- Attach the Tablet using a USB cable and click "Run on device" in the *Processing* menu.

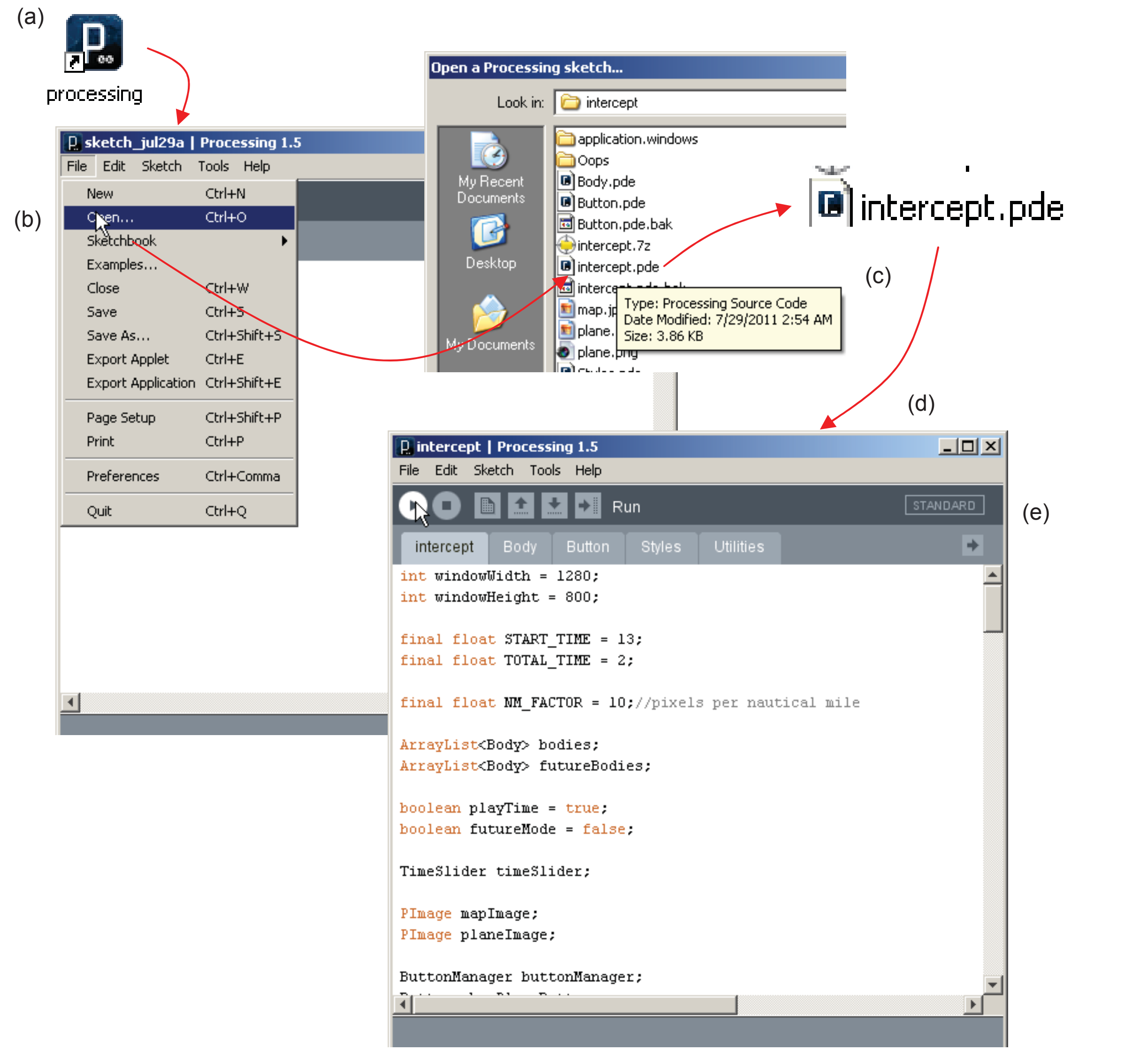

Figure 4-1 Launching the Intercept.pde Mock-Up

Click on (a) the *Processing* icon to launch the *Processing* GUI, then (b) click File/Open to get a file selector. Navigate to choose (c) Intercept.pde, which then opens the required code (d). *Processing* is shown here in "Standard" mode (e).

For the simple Predictive Rendezvous demo, the controls switched seamlessly from mouse to touchscreen. We did other experiments drawing lines and creating regions with two fingers, and had no problems with the touchscreen interface.

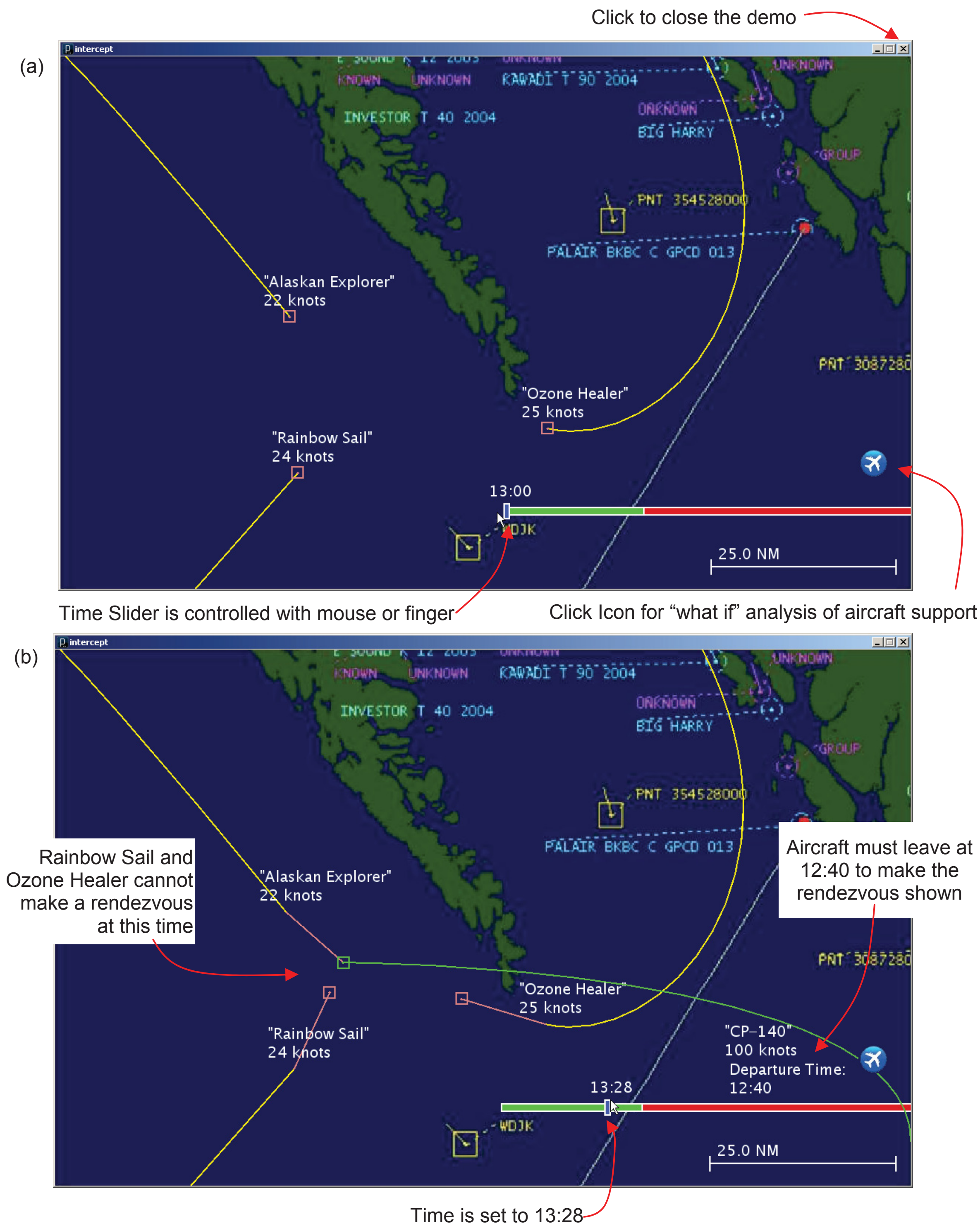

Figure 4-2 The Predictive Rendezvous Mock-Up Screen

Screen (a) appears when intercept.pde first runs. Slide the time slider as shown in (b) to view where the vessels might be at that new time. Click on the aircraft icon to do a "what if" analysis of launching an aircraft to observe the rendezvous.

5 Small Maritime Ontology

Although the Situational Facts Knowledge Base (SFKB) is not just an ontology, like an ontology it can be portrayed using a node-link graph portrayal. It was therefore easiest, when creating mock-ups of the Facts on the Map app ([3] Section 7.3) and the Visual Rationale app ([3] Section 7.4) to create a very small maritime ontology as shown in Figure 5-1. That ontology is included in the distribution disk as `QuickShips.owl`.

The ontology was created using Protégé 3.4.4[11]. Visualization of the ontology was done using Graphic OWL (GrOWL) [9].

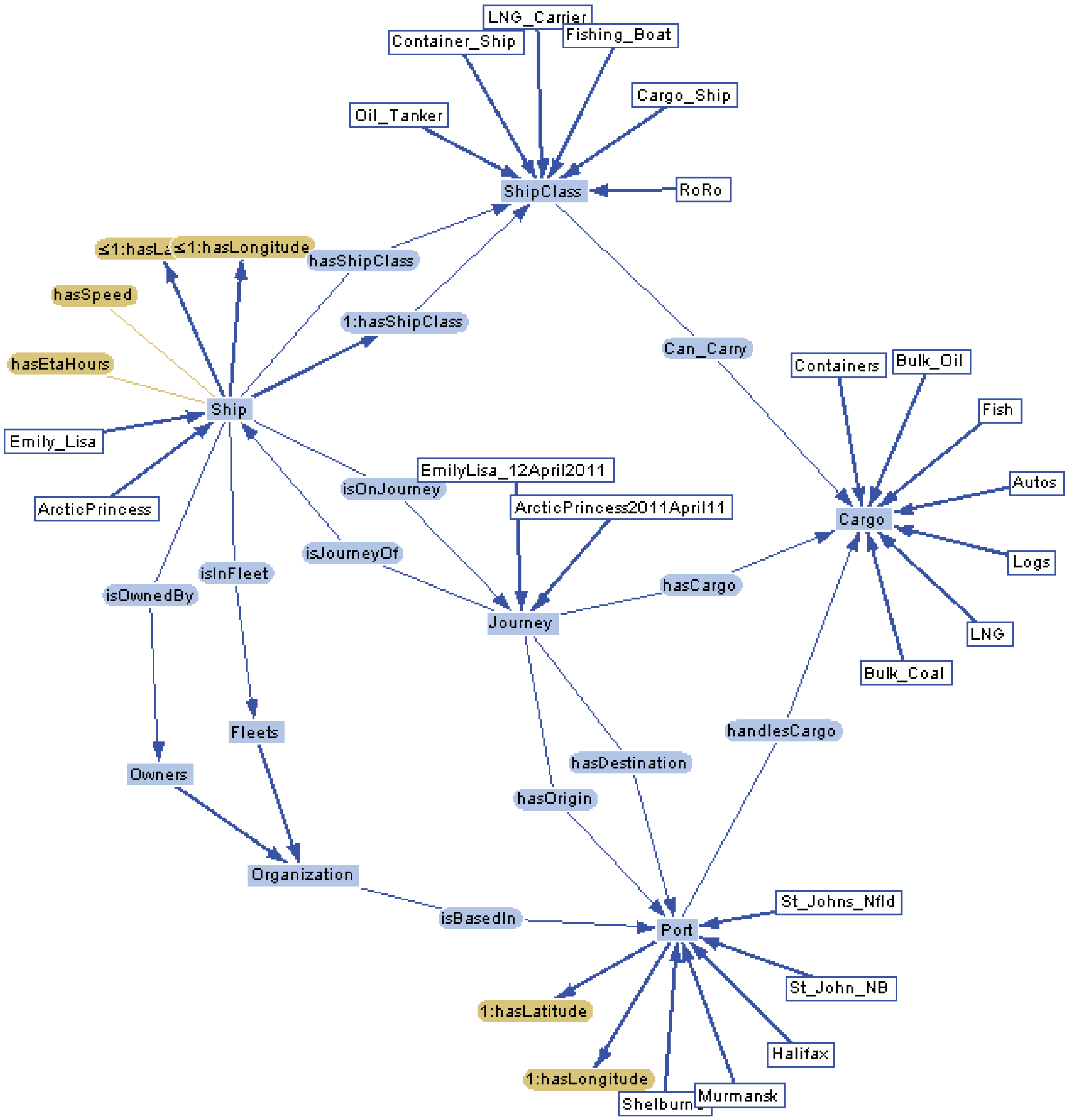

Figure 5-1 The Quick Ships SFKB Mock-Up

This diagram shows the complete ontology that was used to mock-up the Situational Facts Knowledge Base. The visualization was done using GrOWL[9], where round blue boxes are object properties, brown boxes are datatype properties, square blue boxes are classes, and white boxes are individuals.

Appendix A – HTML code for creating Plugin instance

```
<html>
<head>
   <title>Sample</title>
   <script src="https://www.google.com/jsapi?key=ABCDEFG"> </script>
   <script type="text/javascript">
      var ge;
      google.load("earth", "1");

      function init() {
         google.earth.createInstance('map3d', initCB, failureCB);
      }

      function initCB(instance) {
         ge = instance;
         ge.getWindow().setVisibility(true);
      }

      function failureCB(errorCode) {
      }

      google.setOnLoadCallback(init);
   </script>

</head>
<body>
   <div id="map3d" style="height: 400px; width: 600px;"></div>
</body>
</html>
```

You will need to sign up for a Google Maps API key at http://code.google.com/apis/maps/signup.html and insert your unique key here.

A. References

[1] J. Bouchard, and M. Davenport, "Design of a Maritime Domain Awareness Visual Analytics Prototype (DMVAP): Task 3: Preliminary Architecture," DRDC Valcartier Technical Report: CR 2011-223, July 2011.

[2] M. Davenport, "Design of a Maritime Domain Awareness Visual Analytics Prototype (DMVAP): Task 4: Scenarios and Use Cases," Technical Report: DRDC Valcartier CR 2011-224, May 2011. [A copy was saved as file: DmvapUseCases.docx]

[3] M. Davenport, V. Lavigne, and D. Gouin, "Design of a Maritime Domain Awareness Visual Analytics Prototype (DMVAP): Task 1: Design Study," Technical Report: DRDC Valcartier CR 2011-221, July 2011.

[4] B. Fry, and C. Reas, "Download Processing," web page at: http://processing.org/, 2011.

[5] Google, "Google Earth API Demo Gallery," web page at: http://code.google.com/apis/earth/documentation/demogallery.html, 2011.

[6] Google, "Google Earth API Developer's Guide," web page at: http://code.google.com/apis/earth/documentation/index.html, 2011.

[7] Google, "Google Earth API Reference," web page at: http://code.google.com/apis/earth/documentation/reference/#, 2011.

[8] Google, "KML Extension Namespace and the gx prefix," web page at: http://code.google.com/apis/kml/documentation/kmlreference.html#kmlextensions, 2011.

[9] S. Krivov, R. Williams, and F. Villa, “GrOWL: A tool for visualization and editing of OWL ontologies,” Journal of Web Semantics, vol. 5, no. 2, www.websemanticsjournal.org/index.php/ps/article/download/111/109, 2007 [A copy was saved as file: KrivovGrowl_2011.pdf]

[10] PWGSC, "Conception Prototype Analytique," document W7701-101843/001/QCL, 2010. [A copy was saved as file: W7701-0-1843 C.pdf]

[11] Stanford University, "Welcome to Protege," web page at: http://protege.stanford.edu/, 2011.

[12] Stichting Zeilvaart Warmond, "Zeilvaart Warmond (Sailing Warmond)," web page at: http://2010sr.zeilvaartwarmond.nl, 2011.

[13] J. Wernecke, The KML Handbook: Geographic Visualization for the Web Addison-Wesley Professional, 2008.

[14] Wiki Processing, "About Processing for Android," web page at: http://wiki.processing.org/w/Android, 2011. [A copy was saved as file: Android Processing.mht]

B. Contractor Distribution List

Name	Company	Copy
Michael Davenport	Salience	X

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified.)

1. ORIGINATOR (Name and address of the organization preparing the document.)

   Salience Analytics Inc.
   87 West 41 Ave
   Vancouver, BC, Canada, V5Y 2R8

2. SECURITY CLASSIFICATION (Overall security classification of the document, including special warning terms if applicable.)

   Unclassified
   (NON-CONTROLLED GOODS)
   DMC A REVIEW:GCEC JUNE 2010

3. TITLE (The complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S, C or U) in parentheses after the title.)

   Design of a Maritime Domain Awareness Visual Analytics Prototype (DMVAP), Task 2: Concept Exploration (U)

4. AUTHORS (Last name, followed by initials – ranks, titles, etc. not to be used.)

   Venour, C., Roodnick, D.

5. DATE OF PUBLICATION (month and year of publication of document.)

   July 30, 2011

6. NO. OF PAGES (Including Annexes, Appendices and DCD sheet.)

   31

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

   Contract Report

8a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant)

   11hm

8b. CONTRACT NO. (If appropriate, the applicable number under which the document was written)

   W7701-101843

9a. ORIGINATOR'S DOCUMENT NUMBER (Official document number by which the document is identified by the originating activity. Number must be unique to this document.)

   DRDC Valcartier CR 2011-222

9b. OTHER DOCUMENT NOS. (Any other numbers which may be assigned to this document either by the originator or the sponsor.)

10. DOCUMENT AVAILABILITY (Any limitation on further distribution of the document, other than those imposed by security classification.)

   (x) Unlimited distribution
   (   ) Distribution limited to defence departments
   (   ) Distribution limited to defence contractors
   (   ) Distribution limited to government
   (   ) Distribution limited to Defence R&D Canada
   (   ) Controlled by Source

11. DOCUMENT ANNOUNCEMENT (Any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (10). However, where further distribution (beyond the audience specified in (10) is possible, a wider announcement audience may be selected.)

   Unlimited

12. ABSTRACT (Brief and factual summary of the document. May also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), or (U). May be in English only).

A design study has been done into visual analytic strategies for improving Maritime Domain Awareness. The study provides a functional specification for 23 possible software applications ("apps") that respond to specific operational requirements and analytical challenges at Canada's Regional Joint Operations Centers (RJOCs). It recommends that the apps be transient, self-contained, and single-purpose so that they can be integrated as add-ons rather than replacements for the current systems. Shared resources, including a tagging infrastructure, an Analysis Set of vessels, an activity log, and a mind map environment for recording analytical insights, facilitate collaboration and the sense-making process.

About half of the proposed apps use the 2D geographical map currently familiar to the RJOCs. The Current Plot still plots ship locations and tracks on a chart, but analysts can now smoothly slide the contacts backward and forward in time to understand past behaviour and anticipate future threats. The plot can be enhanced with thematic views, bubble sets, and what-if analysis. Pop-up overlays provide evidence for close encounters, odd routes, missing contacts, and overall reliability of each ship.

Time-domain displays are used to visualize patterns of vessel behaviour, news feeds, chat text, and collaboration activities. Visualizations include timelines, graphs, and ThemeRivers™. A Link Analysis App based on off-the-shelf tools is included, but with extensions for this specific domain. Two of the apps provide visual analytics for browsing facts and reviewing the reasoning process of the Situation Facts Knowledge Base (SFKB).

13. KEYWORDS, DESCRIPTORS or IDENTIFIERS (Technically meaningful terms or short phrases characterizing a document and could be helpful in cataloguing it. Should be **Unclassified** text. If not , the classification of each term should be indicated as with the title. Equipment model designation, trade name, military project code name, and geographic location may be included. If possible, should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified.)

Visual Analytics, Domain Awareness, Maritime, Recognized Maritime Picture, Vessel of Interest, Situation Awareness, Anomaly Detection, Visualization, Collaboration, Prototype Design.

Defence R&D Canada

Canada's Leader in Defence
and National Security
Science and Technology

R & D pour la défense Canada

Chef de file au Canada en matière
de science et de technologie pour
la défense et la sécurité nationale

DEFENCE DÉFENSE

www.drdc-rddc.gc.ca