DESIGN OF WEB-BASED GIS TOOLS TO SUPPORT ROUTINE QUALITY MONITORING OF REMOTE SENSED PRODUCTS FROM NEXT GENERATION LEO AND GEO MISSIONS

T. Scott Zaccheo, Michael Sze and David B. Hogan

AER, Inc, Lexington, MA 02421-3126

Outline

- **Introduction**
  - What role might open-source Geographic Information System (GIS) applications, e.g. Google Earth™/Google Map™, technologies play in dissemination of data from next generation LEO and GEO weather remote sensing systems?
- **Design Objective**
  - Develop modular prototype environment to assess applicability of these technologies
- **Example Prototype Discussion**
  - Google Earth Applications
  - Web-browser embedded Google Earth/ Google Maps applications
- **Observations**

Introduction

- **Current and next generation operational satellite observing system are migrating towards providing standard products for**
  - Routine satellite operations
  - Product quality assessment and monitoring
  - Access to common data product for applications
  - Education outreach programs
- **Question**
  - To what extent can open source publically available GIS technology provide tools and applications to meet these needs?
- **Assessment Method**
  - Develop prototype three tier infrastructure to assess applicability of open source GIS technology

Prototype Design Objectives

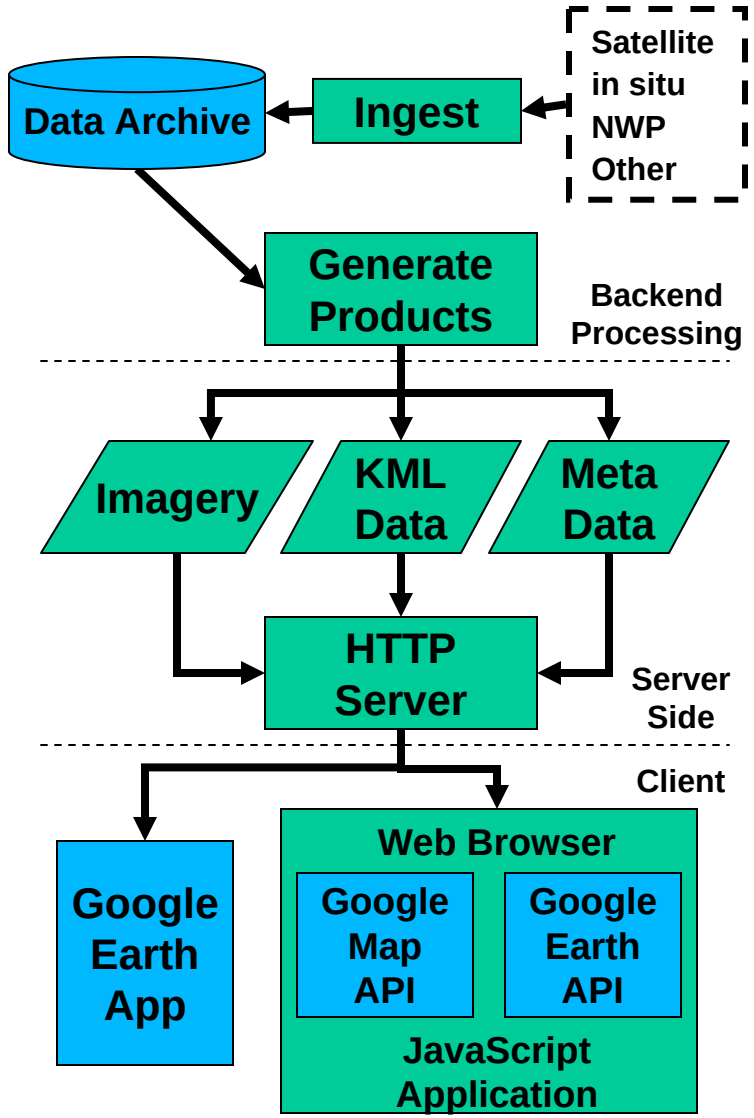
- **Objectives**
  - Modular approach that supported multi-tier implementation
  - Support
    - Satellite remote sensed imagery/products
    - Analysis and Forecast data
    - In situ measurements/observation data
      - METAR, SYNOP, BOUY
    - Incorporate “traditional” analysis methods
      - Comparison of model and estimates derived from satellite data with “ground truth”
      - Provide user selectable graphical and text base analysis
- **Design Decision**
  - Google Earth™/ Google Maps™ chosen as user diplay tool
    - Publically availability
    - Open source interface description Keyhole Markup Language (KML)
    - Web browser embedding permits JavaScript APIs
    - Ease of implementation

Keyhole Markup Language (KML) Relevance to Remote Sensing Applications

- **What is KML?**
  - Open source XML schema or ASCII-based file format for describing temporally and geo-spatially located data and information.
- **Attributes Most Relevant to Satellite Remote Sensing and Meteorology Data**
  - **Placemarks**
    - Descriptive feature containing geometric elements points, lines, polygons – 3 dimensions (+ time see below)
  - **Overlays**: Geo-referenced Imagery
  - **Time Attributes**: Start and stop interval for animation
- **KMZ Extension**
  - Defines methods for compressing KML files
  - Provides compressed multilayered file structure based on standard “zip” technology
  - Facilitates net-centric distributed data architectures

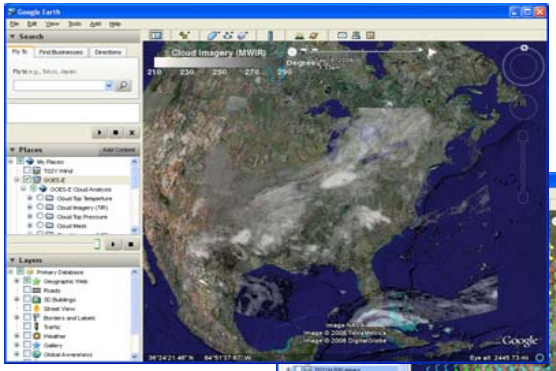
Prototype Three Tier Architecture

- **Backend Processing**
  - Convert products to PNG Imagery, KML/KMZ data
  - Generate KML/KMZ and ancillary metadata
  - Data types
    - Satellite imagery in standard scientific formats, e.g. netCDF and HDF.
    - NWP analysis and forecast data in WMO GRIB format
    - Point source observation data in WMO BOUY/METAR/SYNOP and netCDF formats
- **Server Side**
  - Standard data store and HTTP services
  - Potentially provides “on demand” access
- **Client Side**
  - Standalone Google Earth
  - Browser Embedded Applications
    - Exploit web scripting language to integrate Google frameworks with traditional graphical and text based analysis tools

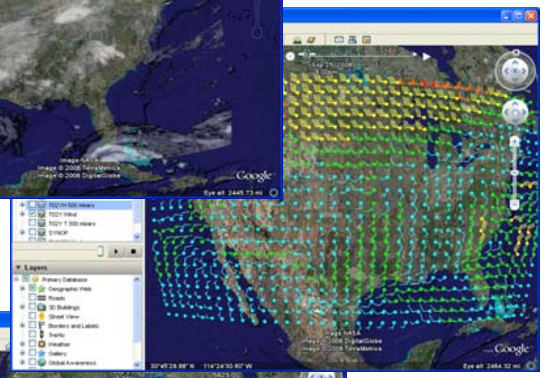


Google Earth™ Applications

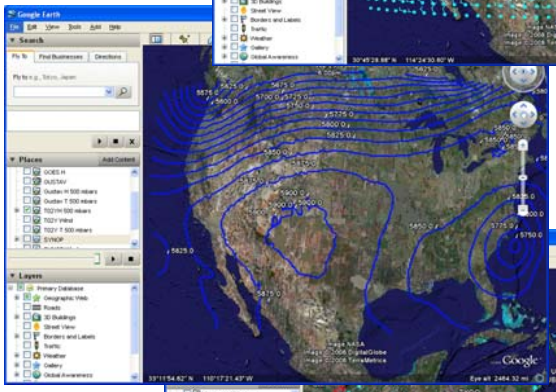
- **Sample Applications**
  - Animation of multi-spectral satellite imagery weather symbols or NWP field contour lines
    - Implementation: KMZ containing mix of imagery and KML meta data
  - Worldwide Station Browser provides current weather graphics for user selectable stations
    - Implementation: KML
- **Benefits**
  - Ease of implementation
  - Programmable multiple level of detail
  - Spatial and temporal data registration
- **Limitations**
  - Lacks ability to extend user interactions beyond basic application functionality, e.g. callbacks



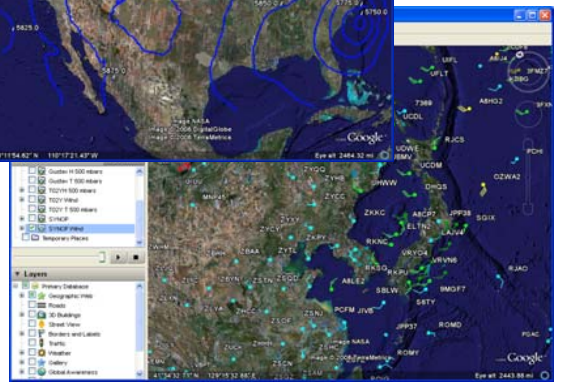
Animated Satellite Imagery



Animated Wind Barbs

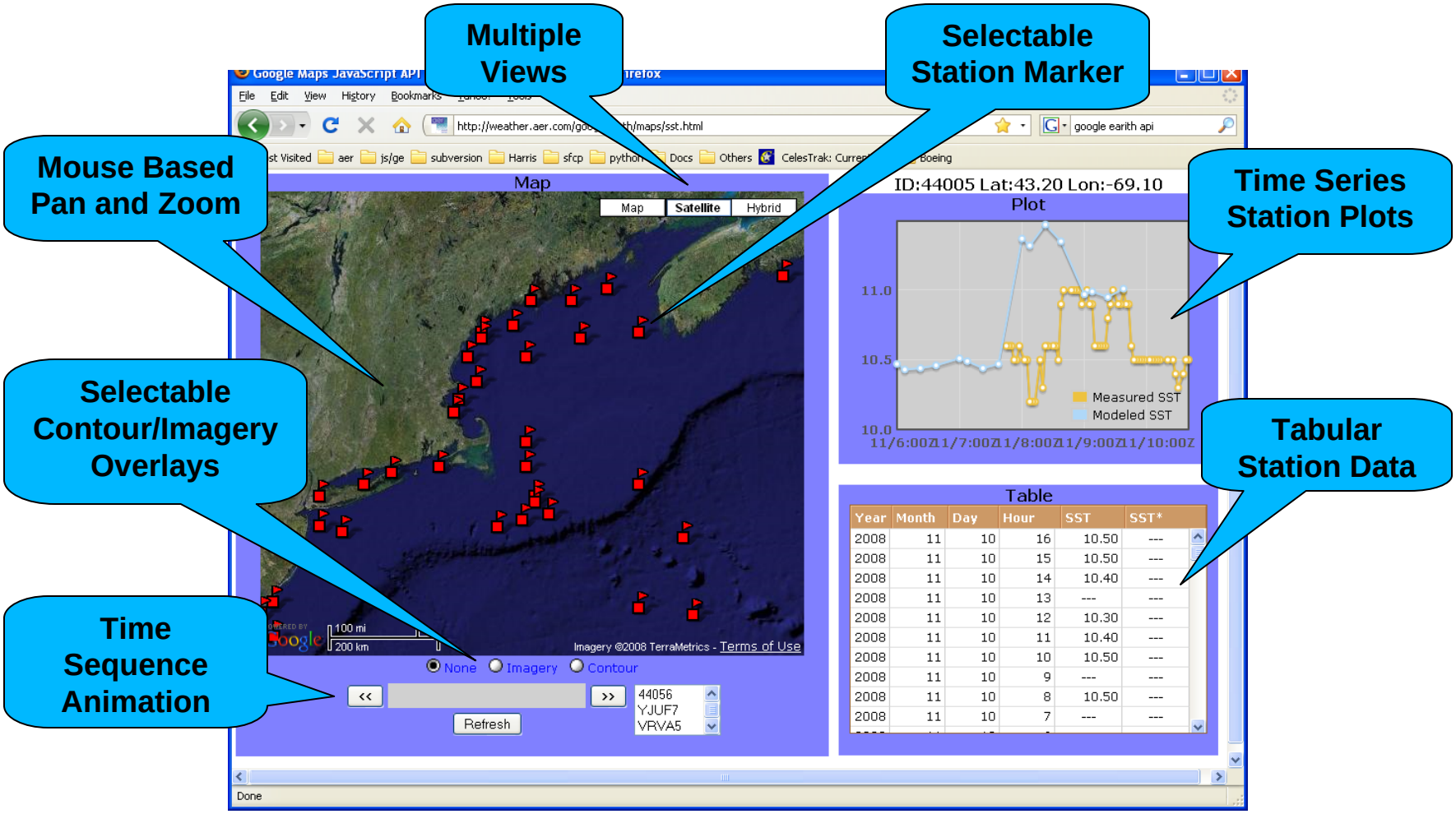


Animated NWP Contour



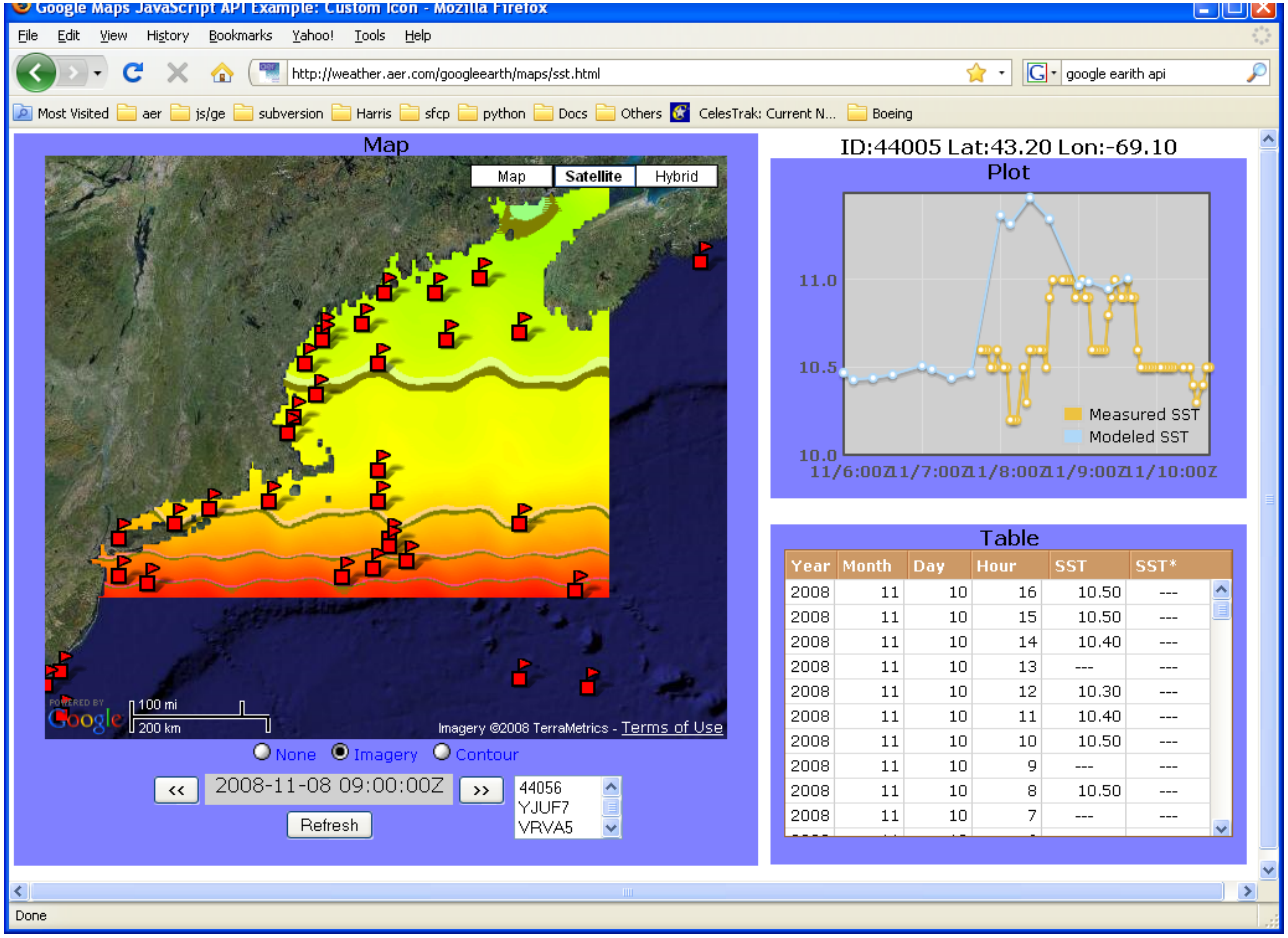
Worldwide SYNOP Station Information

Browser Embedded Google Map™



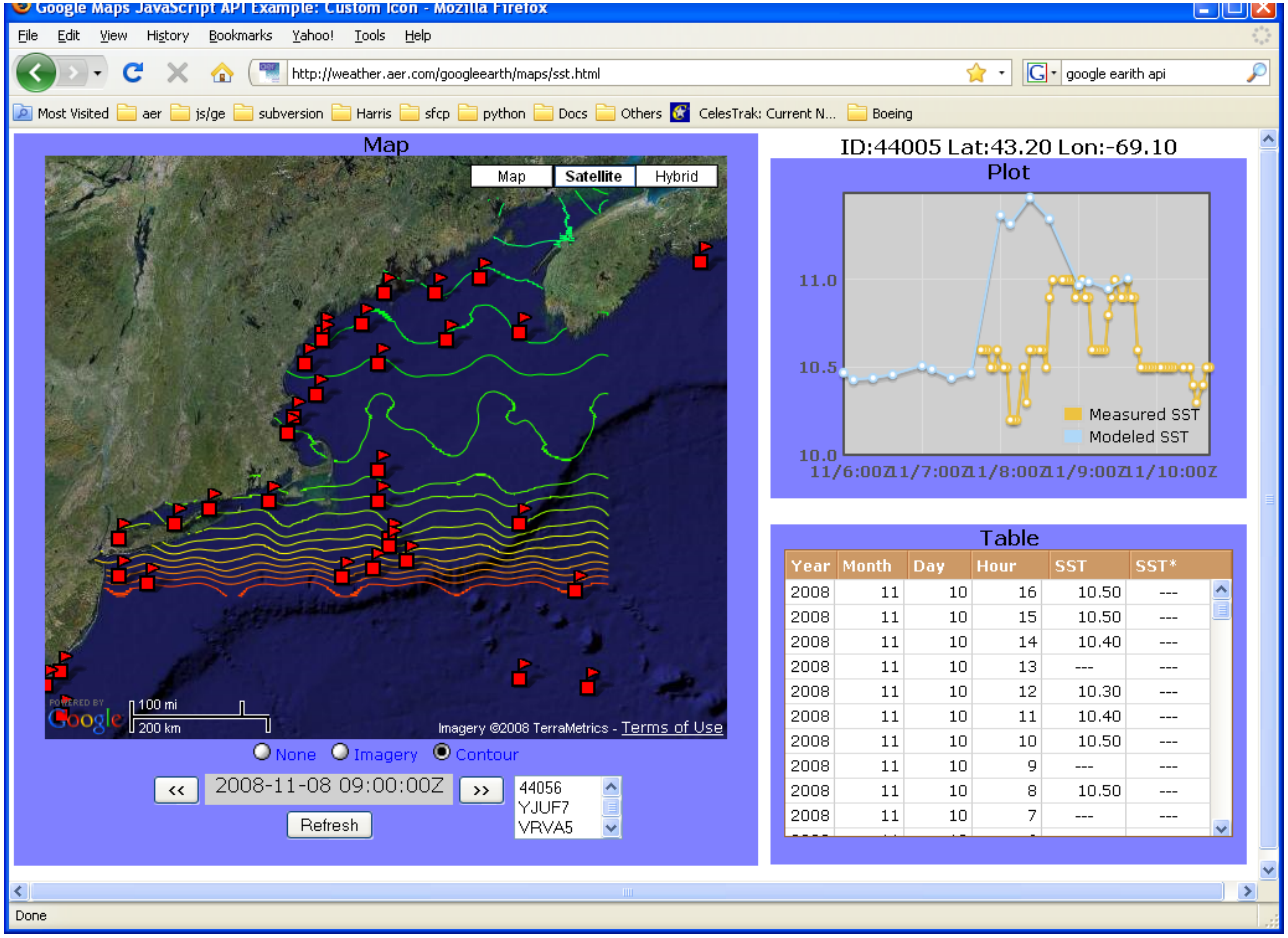
Google Map/Earth APIs combined JavaScript/PHP/HTML can provide GIS-like functionality to interactive web-based tools

Prototype Implementation
Animated SST Analysis Imagery Overlay



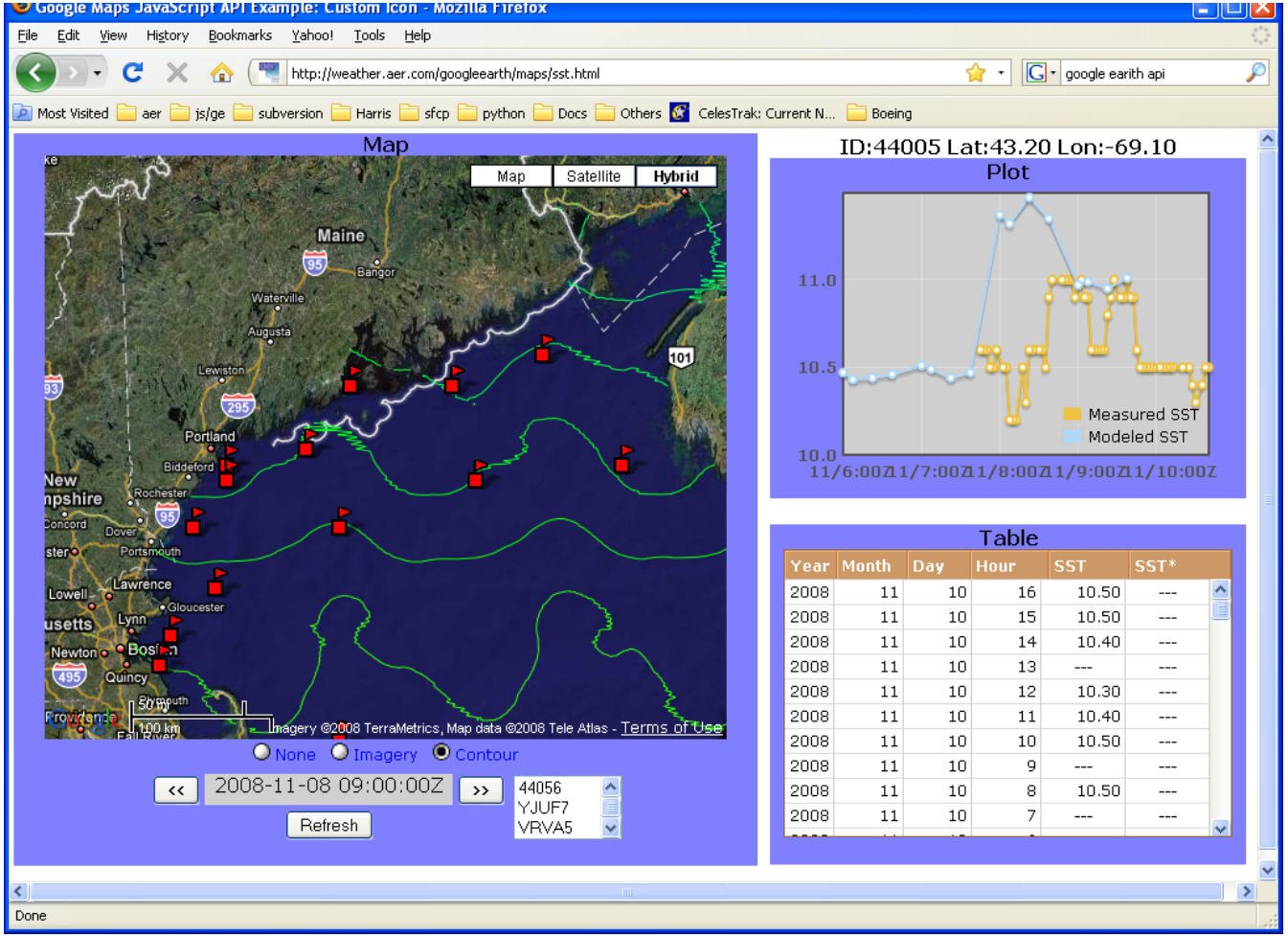
Integrated time animated geo-referenced imagery implemented using standard PNG/GIF imagery and JavaScript API methods

Prototype API Implementation
Animated SST Analysis Contour



Time animated KML analysis contour plots implemented using KML data and JavaScript methods

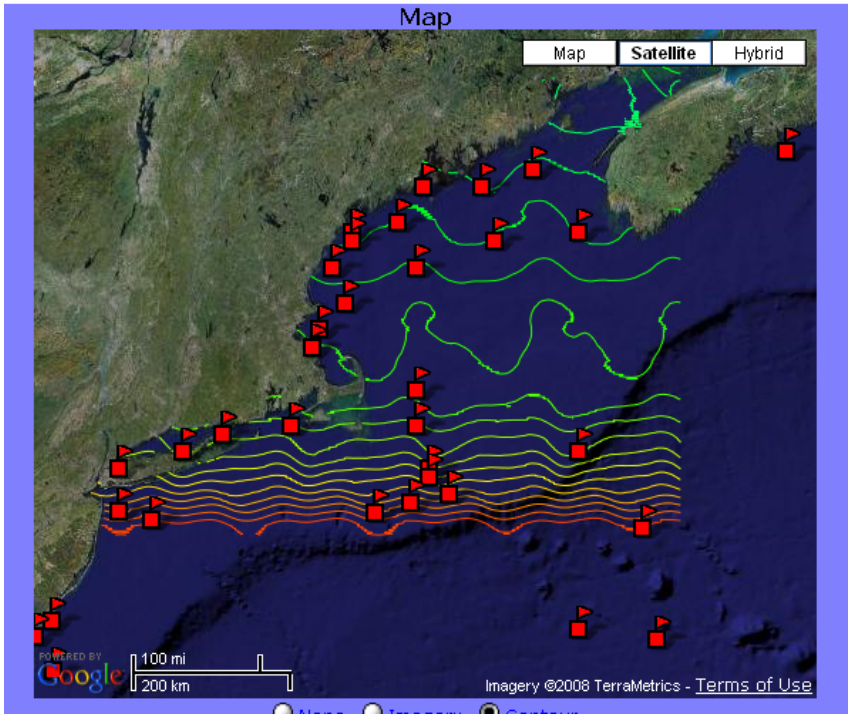
Prototype API Implementation
GIS Display Functionality



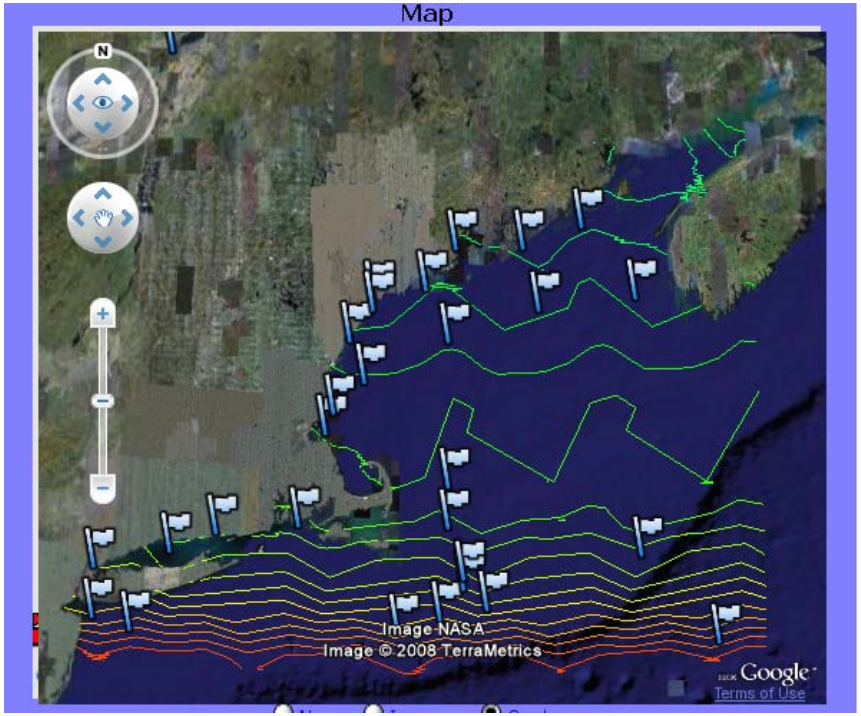
Google APIs provide varying level of detail including geopolitical boundaries, transportation routes and meter resolution imagery

Browser Embedded Google Earth

Google Maps API Implementation



Google Earth API Implementation



- **Google Map API replaced with Google Earth API**
  - Provide full feature KML support
- **Modifications**
  - KML layer for map-based imagery

Summary

- **General**
  - Probably not a current choice for complex data analysis tasks or for GIS-workflow applications.
  - Excellent adjunct for standardized data exploration capabilities.
  - Subscribing to the standards used by these tools (especially KML), increases flexibility, capabilities and ease of future migration.
- **Google Earth™ Stand-Alone Application**
  - Simple approach for providing access to wide user base.
  - Requires only providing KML-formatted data.
  - Inherently network enabled and permits access local and remote data transparently.
  - Does not permit customization of the user interface.
  - Time sequencing and animation are provided but are inflexible.
- **Google Earth™ or Google Maps™ API Approach**
  - Provides excellent way to provide a wide user community with customized user interfaces to geographic diverse data sets.
  - High degree of customization is available through Javascript/ ECMA programming.
  - Although very sophisticated, there is a cost associated with developing custom client
  - Embedded Google Earth™ currently is computationally and memory intense.