Designing a Weather Service

Daniel J. Lacks, PhD[1], Mel Pelchat[2] and Samuel Lee[3]
[1]Chief Scientist, Cole Engineering Services, Inc. (CESI), Daniel.Lacks@cesicorp.com, Orlando, FL, USA
[2]Software Intern, CESI, Mel.Pelchat@cesicorp.com, Orlando, FL, USA
[3]Software Intern, CESI, Samuel.Lee@cesicorp.com, Orlando, FL, USA

Abstract

This work describes the design approach for building a weather service research and development project called Clime. The purpose of Clime is to provide real time, present, historical, and future predictive weather information derived from authoritative international sources as well as fabricated or modified weather data for training, testing, experimentation, and other modelling and simulation use cases. The challenges for designing Clime included determining weather data needs, finding the data amongst authoritative sources, mapping or formulating the data, closing data gaps, and providing useful information for future capabilities. In addition to data needs, the design approach includes capabilities to share data, generate graphics, tap into live feeds, and host imagery. Our initial intent was to align data with legacy capabilities, but our result was to instead generate new data models due to the breadth, depth, quantity, and quality of newer data sources with better relevancy and higher fidelity information.

1 Introduction

This work describes the design approach for building a weather service research and development project called Clime. The purpose of Clime is to provide real time, present, historical, and future predictive weather information derived from authoritative international sources as well as fabricated or modified weather data for training, testing, experimentation, and other Modelling and Simulation (M&S) use cases. Much of the internationally used weather simulation data in use today derives from the Synthetic Environment Data Representation & Interchange Specification (SEDRIS) Meteorological and Oceanographic (METOC) specification which is still relevant to industry, academia, and government implementers [1]. The METOC specification is feature rich and includes climatology, wind, solar energy, humidity, weather, clouds, atmospheric forecasts and observations, ocean observations, bathymetry, and more for a variety of land, sea, air, and space simulation applications. However, a major drawback to using SEDRIS is not so much with SEDRIS, but many of the data sources in use are dated perhaps from the peak of SEDRIS development about 15 years ago to as far back as 50 years. The advance of technologies to sense, collect, and record the data available open source data solutions has higher accuracy and better fidelity than what is available in data products generated by SEDRIS mainly due to source data not being updated consistently or timely.

The challenges for designing Clime as a completely data driven tool included determining weather data needs, finding the data amongst authoritative sources, mapping or formulating the data, closing data gaps, and providing useful information for future capabilities. M&S weather data requirements not only ask for temperature but want to know particulars for a region like biomes. For example, a 0 deg C temperature reading does not provide regional information being a tundra which affects survivability, instrumentation, and trafficability. Amongst the thousands of worldwide weather instrumentation devices conveniently accessible from publicly available files and data feeds on the Internet, another challenge presents itself because the devices vary in age and functionality which results in providing inconsistent data sets varying by region. In order to overcome these gaps, we find ourselves searching for more data, formulas, and procedures for deriving and classifying what is typically abstract information such as climate zones.

2 Approach

We presented the Terrain Web Service (TWS) at ITEC 2017 [2]. TWS converted the One Semi-Automated Forces (OneSAF) Environment Runtime Component (ERC), based on SEDRIS [3], into service called Glade which is provisioned on-demand, sitting behind a single Application Programming Interface (API) web service called Waymate. TWS hosts terrain and weather (METOC) data which is accessed via a question and answer protocol, for example, "What is the temperature in this region?" The question and answer are formatted in JavaScript Object Notation (JSON) [4], and in this case the answer indicates the temperature value and units (degrees Celsius) as it would be obtained by directly calling the ERC API.

The intent of building TWS was to provide a loosely-coupled mechanism for legacy simulators to interact with web services to get consistent answers to synthetic environment related questions in a manner that promotes fair-fight (where each simulator gets the same answer). The market research at the time indicated that simulators that were not users of Synthetic Environment Core data products, the data feeding ERC and SEDRIS libraries, needed a place to get this information without having to change their terrain format or libraries to use ERC natively. However, that market research was obviated by the concurrently maturing concept of the Synthetic Training Environment (STE) which is opting to build its own One World Terrain (OWT) [5] synthetic environment. OWT is not guaranteed at this point to comply with SEDRIS data formats, thus from a design perspective, the desire for simulators to use or reuse SE Core or SEDRIS related data products is on-hold until the

OWT concept matures to see if it adopts or replaces SE Core or SEDRIS.

Due to the uncertainty of STE's direction, and the desire for STE to also have a weather service capability, Clime proposes an alternative method to TWS or ERC for providing weather data which is not dependent on SEDRIS or SE Core related data. The design approach for Clime is to be data driven and flexible to accommodate information from, particularly, readily available open source data in addition to any other data formats desired. Clime's design will accommodate connecting, querying, and streaming weather data from reliable sources in addition to importing data from files to accommodate open and closed network deployments. Also, unlike TWS which API complies with the ERC API data structures and method names, the Clime API is designed from the ground up to provide *raw* weather data as collected from sensors, compiled and consolidated weather data to provide meaningful history and forecasts; and *abstract* weather data which provides context to the numbers (i.e., "it is hot and muggy outside"). Another key feature for Clime is to repose pure data and allow collaborative data editing, with the abilities to revert and share data.

2.1 Data Mapping

Since SEDRIS or SE Core are not considered to be primary providers of weather data for Clime, we researched viable international open source data sources that could meet or exceed the capabilities available from legacy technologies. The expectation is to meet or exceed the data quality, and the timeliness and ease to acquire the data. The approach accommodates two use cases. The first is when Clime is connected to the Internet and can directly download and update weather data periodically or on-demand. The second use case involves a casual computer user getting some guidance from Clime on accessing weather data from a data source, going to a website, choosing the data they want, downloading it, and then uploading it into Clime. The following web sites provide a wealth of information:

- https://openweathermap.org/history - provides an API to get current and historical weather data by location to get temperature max and min, pressure, humidity, wind speed, etc. in a JSON format for free or fee based on usage. Provides open source tools to display data on maps. We prototyped retrieving live data to include lat/lon; city ID and name; time; sunrise time; sunset time; temperature low, high, and current; humidity; sea and ground pressure; wind speed, gust, and direction; cloudiness; general weather conditions; rainfall; and snowfall.
- https://aa.usno.navy.mil/data/docs/RS_OneYear.php - provides sun and moon data such as sunrise, sunset, transit times, civil twilight, lunar phase information as a data table for free.
- https://darksky.net/dev/docs#response-format – provides an API to get historical, current, or forecast weather data to include temperature, pressure, snowfall, wind speed and direction in JSON for free or fee depending on usage.
- https://www.ncdc.noaa.gov/cdo-web/ - NOAA provides worldwide weather datasets for over 100,000 historical sensor data feeds (30+ years) and over 2,000 active sensor data feeds.
- https://www.cpc.ncep.noaa.gov/products/global_monitoring/temperature/global_temp_accum.shtml - NOAA Climate Prediction Center (CPC) data provides global max and min temperatures files and was used for our initial prototype.

From a design and implementation perspective, the level of effort to ingest the data sources varies. OpenWeatherMap and DarkSky appear to have polished data, but there are daily data caps to access data for free. The US Navy solar/lunar illumination data is polished, but text-based data tables must be parsed to get the necessary data. NOAA datasets are sometimes very high quality, but other times data is missing or of partial quality, even when querying the same area over time. CPC data is highly compressed, and upon decompression consumes a lot of storage space. The assumption is modernization or malfunction of equipment, or varying data collection and reporting capabilities. Our initial estimate for replacing the METOC API data sources based on open source data is 71%, this number may be improved with data abstraction or estimation. Some missing data may be calculated, such as accumulated precipitation. Other values like percentage of fog coverage may not be available and thus pose a data gap. OpenWeatherMap lists, for example, mist and fog which may be manually mapped to different fog coverage values. Other options include using SEDRIS, finding another data source, or calculating fog coverage based on weather patterns. While data gaps exist between open source and METOC data, alternative approaches exist to get or generate data.

2.2 Data Abstraction

While weather data sources provide a wealth of information, the information is generally presented as raw numbers. Raw data is sometimes needed, for example, to feed simulated sensors models. However, abstract data is sometimes needed as well either for models or human comprehension. For example, knowing there is 8.6mm of rain in a pan may be useful for a calculation, but a human simply wants to know there is heavy rain in a town.

Weather measurements tend to come in two formats: periodic (typically daily) measurements and aggregate measurements summarizing data over a period (of days). Either or both formats may be given by the same station for different periods. This resulted in developing a coherent system within the Clime design to unify the data. When the weather is queried for a specific day, the query does not need to first check for a daily measurement, and then start searching for a multiday measurement that overlaps that day if a daily information is not present. The technical solution is to create a unified data model to ease the querying process across different data sources. The data will include accurate historical data that averages the multiday measurements, alongside statistical data. Statistical data yields approximations when data is not present, and it may also be used for forecasting. For example, if a weather station recorded 19 inches of rain over a 30-day period, a 0.63-inch approximation may be made for rain every day. Alternately, a normal distribution may be used, applying mean and standard deviation values for each of those days. These and other approaches are varied by the user.

The Clime design approach considers querying for this information in a relational database using Structured Query Language (SQL) [6]. Frameworks to interact with the SQL database are also design considerations. Pivotal develops the Spring [7] framework for enterprise applications in Java. Oracle also has an existing architecture for representing databases called the Java Persistence Architecture (JPA) [8], and another architecture for building queries and interacting with databases called Hibernate (H2) [9]. The JPA treats database interactions as manipulations on a collection of objects, and H2 will translate what the JPA needs from the database into a query in SQL. Queries may be created in SQL or in the Java Persistence Query Language (JPQL) [10] which is a Java viewpoint on SQL.

To unify these capabilities under the Spring framework, Pivotal developed the Spring Boot framework. It utilizes the H2 in-memory database for testing and allows loosely coupled connectivity to any relational database so that databases may be interchanged, and the program would never know. The Spring Web module [11] facilitates easy REST adapter setup for service communication over a network. For authentication there is Spring Security as well [12].

There are legacy METOC queries that are currently identified as data gaps based on the data mapping process. Some climate zones cannot be computed by a pure weather service like, "Ocean/Body of Water." This is not a kind of weather, it is a terrain. A separate system is needed (such as TWS) to handle terrain queries. But a weather database identifies if any underbrush is healthy and wet, or if it's ready to ignite with the slightest spark.

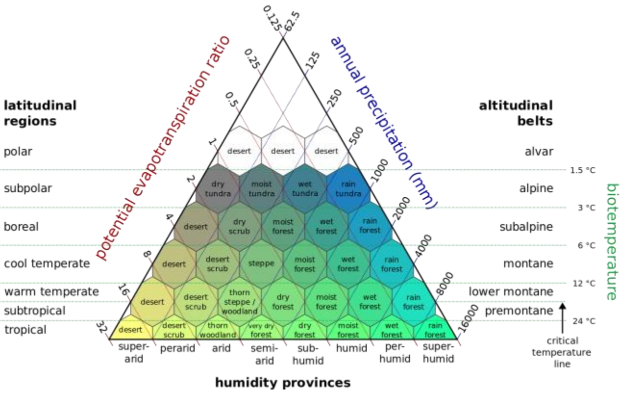

Fig. 1. Leslie Holdridge's Life Zone Classification System

Some regional information may be abstracted based on available information. The climate zone "Tropical Rainforest" may be classified based on average rainfall, temperature, and ratio of evaporation to rainfall using the Life Zone Classification System (Fig. 1) [13]. Such assumptions are not without error. An area with high heat and rain is generally not considered to be a tropical rainforest if all the trees have been cut down by humans. However, the Life Zone Classification System identifies the general conditions for a tropical rainforest at that location which is anticipated to cover Earth accurately.

Other data gaps stem from weather service sensors not located at every location on Earth. Weather will be approximated based on nearby sensor data. This is done by querying a region around the requested location and pooling the data from all the stations in the region. Then, the data computed is based on the square distance from the target location, with closer stations receiving a stronger bias.

Finally, there are data gaps because one weather system's data set may include stations that another system does not. The Clime design includes a method to import datafiles from a weather system into its own local database, and then convert that data into a unified query-able data source for our database. Adding new data sources to our system does not require a complete overhaul of our system, just a new module developed for importing data from that source.

3 Conclusion

The Clime design approach includes capabilities to share and edit data, generate graphics, tap into live feeds, and host imagery. The functionality and performance factors for those features influenced us to experiment with techniques to best balance querying a database versus working with a software framework such as Spring. The approach involves importing pure datasets into the configuration managed database for data at rest. We expect open source data to meet 71% of our existing requirement to meet or exceed current METOC capabilities. For data gaps, the data may need to be manually entered, we may revert to using SEDRIS, find another data source, or calculate data based on weather patterns. We prototyped bringing data into the Spring framework, and then wrote normalized data tables back to the physical database as needed. Our initial intent was to align data with legacy capabilities such as SEDRIS METOC, but our result was to instead generate new data models due to the breadth, depth, quantity, and quality of newer data sources with better relevancy and higher fidelity open source information. Even though there are some data gaps, the quality, ease of access, and concurrency of open source data exceeds our ability to get the equivalent function from SE Core or SEDRIS.

References

[1] P. Foley, T. Gifford. "Introduction to SEDRIS for Managers." Accessed on November 12, 2018 at http://www.sedris.org/stc/2004/tu/intro/sld001.htm (2004)

[2] D. Lacks. *Terrain Web Services – Building the Backend*. ITEC (2017)

[3] J. Logsdon, R. Wittman. *Standardization, Transformation, & OneSAF*. MP-MSG-056-20 accessed on December 5, 2018 at https://www.sto.nato.int/publications/STO%20Meeting%20Proceedings/RTO-MP-MSG-056/MP-MSG-056-20.pdf (2007)
[4] D. Crockford. *Introducing JSON*. How JavaScript Works, accessed on December 5, 2018 at https://www.json.org/ (est. 2007)
[5] D. J. Durall. "Synthetic Training Environment and the digital revolution in the Army." Accessed December 5, 2018 at https://www.army.mil/article/211619/synthetic_training_environment_and_the_digital_revolution_in_the_army (2018).
[6] International Business Machines (IBM). *Relational Database.* https://www.ibm.com/ibm/history/ibm100/us/en/icons/reldb/, accessed on December 11, 2018 (2018)
[7] Pivotal. *Spring: the source for modern java*. https://spring.io/, accessed December 11, 2018 (2018)
[8] Oracle. *Java Persistence API.* https://www.oracle.com/technetwork/java/javaee/tech/persistence-jsp-140049.html, accessed on December 11, 2018 (2018)
[9] H2. *H2 Database Engine.* http://www.h2database.com/html/main.html, accessed on December 11, 2018 (2018)
[10] Oracle. *JPQL Language Reference.* https://docs.oracle.com/html/E13946_01/ejb3_langref.html, accessed December 11, 2018 (2018)
[11] Pivotal. *Introduction to Spring Framework – Modules – Web.* https://docs.spring.io/spring/docs/3.0.0.M4/reference/html/ch01s02.html, accessed December 13, 2018 (2018)
[12] Baeldung. *Spring Security Authentication Provider.* https://www.baeldung.com/spring-security-authentication-provider, accessed December 11, 2018 (2018)
[13] Holdridge, L.R. *Determination of world plant formations from simple climatic data*. Science. 105 (2727): 367–8, accessed on December 5, 2018 at http://www.sciencemag.org/content/105/2727/367 (1947)

Author/Speaker Biographies

Dr. Daniel Lacks is the Chief Scientist at Cole Engineering Services, Inc and lead of the Clime project. Dr. Lacks spent the last 17 years as an engineer on US DoD programs to include the Synthetic Natural Environment (SNE) team on the WARfighter's SIMulation and JSIMS programs. **Mel Pelchat** is a software engineering intern at Cole Engineering Services, Inc. and has a background in US Army Finance Corps. He anticipates graduating from the University of Central Florida in December 2018 with a Bachelor's Degree in Computer Science. Mel contributes to the Clime effort focusing on prototyping. **Samuel Lee** is a software engineering intern at Cole Engineering Services, Inc. He is currently enrolled as a freshman at the Seminole State College as a Computer Engineering major and works on Computer Engineering and Game Development projects. Sam contributes to the Clime effort focusing on data research and mapping.