Open Graphic Evaluative Frameworks

Authors
Steinfeld, Kyle
Schiavon, Stefano
Moon, Dustin

Publication Date
2012-06-19

Open Graphic Evaluative Frameworks

A climate analysis tool based on an open web-based weather data visualization platform

Kyle Steinfeld[1], Stefano Schiavon[2], Dustin Moon[3]

[123] *University of California, Berkeley*

[1] *ksteinfe@berkeley.edu,* [2] *stefanoschiavon@berkeley.edu,* [3] *drmoon@berkeley.edu*

Abstract. Buildings are the world's largest consumer of energy, accounting for 34% of total use. In the United States residential and commercial buildings are responsible for 72% of electricity use and 40% of CO2 emissions. In order to reduce the impact of buildings on the environment and to utilize freely available environmental resources, building design must be based on site climate conditions, e.g. solar radiation and air temperature. This paper presents a web-based framework that enables the production of user-generated visualizations of weather data. The Open Graphic Evaluative Framework (Open GEF) was developed using the Graphic Evaluative Frameworks (GEF) approach to authoring design-assistant software, which is more appropriate than the now dominant 'generalized design tool' approach when supporting design processes that require a high level of calibration to the cyclic and acyclic shifting of environmental resources. Building on previous work that outlined the theoretical underpinnings and basic methodology of the GEF approach, technical specifications are presented here for the implementation of a Java driven web-based visualization platform. By enabling more nuanced and customizable views of weather data, the software offers designers an exploratory framework rather than a highly directed tool. Open GEF facilitates design processes more highly calibrated to climatic flows that could reduce the overall impact of buildings in the environment.

Keywords. Visualization; Sustainable architectural design; Climate analysis; Weather data

Problem Statement

The most widely used software platforms developed for climate data visualization employ a generalized design tool approach, and pursue an intended genericism. Seeking the widest possible impact, these tools present themselves as "easy to use" and appropriate across a range of climates, micro-climates, building typologies, building activities, material systems and human needs. Rather than providing specialized visualizations appropriate to each of these cases, they produce generic graphics that seek to apply to all of them. This genericism is an appropriate response considering the diversity of the built environment and the massive number of visualizations that would be required to address this diversity with adequate specificity.

Two important factors have shifted in recent years, rendering the generalized design tool approach to weather data visualization inadequate for a growing number of cases.

First, the intended audience for climate visualization tools, architectural designers, has undergone a cultural shift, with many architects regularly engaging in scripting, parametric modeling, and other low-level computational techniques. This shift has lessened the requirement for "easy to use tools" and increased the desire for "customizable tools" that enable a low-level engagement with data and programmatic structures.

Second, as awareness of the importance of climate-responsive design has increased, experimental designs and novel climate-responsive design typologies have gained prominence. These new typologies often require idiosyncratic evaluative methods that are not supported by existing visualization tools. Design for the kinetic-responsive building typology (buildings containing large portions or systems that actuate or change properties in response to environmental dynamics), is one example of an unsupported case of increasing interest in architectural design.

Seeking to address these changing factors, we propose a new software platform for the visualization of weather data in the service of climate-responsive design. As defined in previous work (Steinfeld, et al, 2010), software developed under the GEF approach endeavors to enable designers to:

- Produce data visualizations as a part of (as opposed to in anticipation of) an actively evolving design process.
- Programmatically represent and compare datasets from a variety of sources (published data, output of simulation models, Geographic Information Systems, and other user-defined sources) at a range of scales (climate, microclimate, building zone, material).
- Construct structures for evaluation specific to the unique requirements of the climate and built response with which they are working.
- Responsively evolve the selection of data-points most salient to their investigations alongside an evolving design position, thereby producing customized lenses through which to describe and develop design ideas.
- Compare climatic response strategies for a given set of building specifications.
- Directly manipulate and evaluate weather data, in contrast to existing tools that offer high-level evaluative heuristics.

Precedent Work

Climate Consultant, produced by Murray Milne, Robin Liggett et al. at University of California Los Angeles, is a stand-alone software design tool (Milne and Liggett, 2012) that allows for the visualization and analysis of weather data, and provides recommended design guidelines. It is an excellent example of well-implemented climate visualization tool, but employs the design-tool approach to software authoring and is thereby rendered inappropriate for the cases discussed above.

Weather Tool, produced by Autodesk, is a visualization and analysis program for hourly climate data and a companion application to Ecotect (a widely used climate, solar and thermal analysis tool targeting the architectural design community). Using hourly data recorded in various file formats, such as TMY3, EPW, etc., it allows its user to browse a set of views of these data. When compared to Climate Consultant, Weather Tool does not offer climatic responsive strategies based on the analysis of the climatic data.

Dview [1] is an hourly time series data visualization software developed by Mystaya Engineering for the National Renewable Energy Laboratory. DView focused on the visualization of typical meteorological data (e.g. TMY2, TMY3, EPW etc.). DView displays data in a variety of graphical formats: a) hourly, daily and monthly profiles; b) colorbars; c) boxplots; and d) probability and cumulative density functions. DView can compare two or three dataset in its data series, colorbar and boxplot graphs.

A custom-built weather data parser (Steinfeld, et al, 2010), written as a Java library for the open source graphics environment Processing, has been created which is capable of reading and manipulating weather and simulation data from a variety of sources and multiple formats (e.g. CSV, TMY3, EPW, etc.). The resultant open-source, graphic toolkit brings all the data together onto one platform providing designers with an extensible framework through which interactive data visualizations may be created as part of an evolving, evaluative and comparative design process. This previous work did not include the development of graphic interfaces that support the display of weather data, nor did it include the web-based acquisition of weather data.

Graphic Interfaces

All software, no matter how low-level, presents its user with abstractions for manipulating data through pre-defined routines. While Open GEF seeks to enable a lower-level engagement with weather data, some level of abstraction is required. Rather than manipulating the data more directly, for example via a spreadsheet application, architectural designers prefer to view weather data through lenses created by a set of programmatic graphic interfaces.

Detailed below are the graphic interfaces that the authors speculate are the most useful for innovative climate-responsive design processes, and are currently offered by Open GEF. All of these interfaces allow the user to easily display data using a default set of graphic conventions by employing the draw() function, as detailed below. Some of these interfaces can be described as *graphic spaces*, as they associate a graphic dimension in 2d (typically as an x-axis, y-axis, or polar coordinates) with a weather data variable (ex: dry-bulb temperature, enthalpy, solar position, hour-of-day, day-of-year, hour-of-year). In this type of interface, the user is provided with a function that returns a position on the graph (see the plot() function below), allowing the user to create graphics in a flexible and open-ended way. The plot functionality is not relevant for all the graphic interfaces listed below.

Properties and methods common to most of the interfaces described below are:

- **x,y**: Sets the position of the graphic on the screen
- **width, height**: Sets the size of the graphic on the screen
- **dYr**: A custom data object (dynamic-year, referred to as dYr below) containing 8760 hours (see dHr, below).
- **dHr**: A custom data object (dynamic-hour, referred to as dHr below) each of which contains a value for a pre-defined set of variables. Typical variables loaded from a weather file include: *DryBulbTemp*, *RelHumid*, and *WindSpeed*. Variables such as this are referred to below as “*key values*”.
- **plot()**: Interface-dependent method used in the plotting of data to the screen. Usage and returned values vary.
- **draw()**: Interface-dependent method that displays the graphic interface with default values.

Colorbar

The Colorbar graphic interface, diagrammed in Figure 1 and shown in Figure 2, is a graphic space wherein the x-axis corresponds to days of the year and the y-axis corresponds to hours of the day. This type of graph may be found in a wide range of precedents and is typically used to provide an overview of the dynamics of a limited set of variables across diurnal and annual cycles.

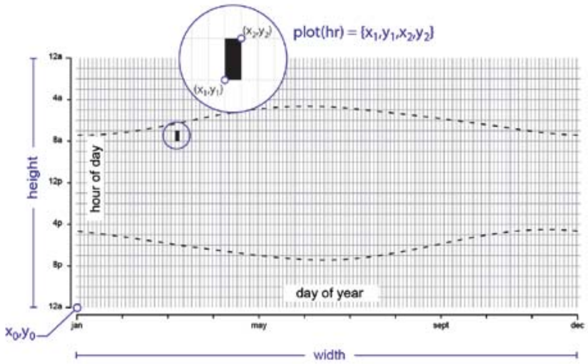


Figure 1
A Diagram of the Colorbar Graphic Interface

The Colorbar draw() method takes a key value and a dYr and draws a colorbar graph with default parameters to the graphic space. In this default version of the graph, a greyscale representation of the key value for each hour is plotted, the range of which is set by the minimum and maximum values present for the year.

The Colorbar plot() method gives the designer more fine-grained control, allowing the representation of multiple key values and user-defined graphics at the individual hour level. The plot() method takes a dYr and the index of the hour to plot, and returns a vector representing the relevant X,Y coordinates of the corner points of a rectangle representing the hour. From these coordinates, the user may draw shapes of the their choosing (*rectangle, ellipse, etc.*) to the graphic space, and perform a color interpolation based on any key values they wish.

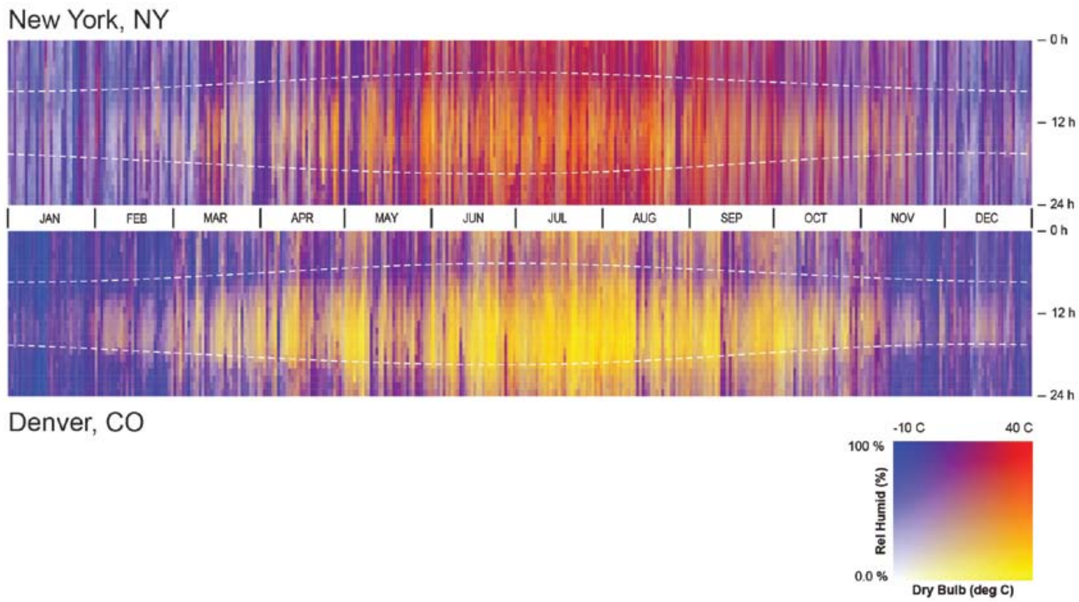


Figure 2
Colorbar Graphics Comparing Dry-Bulb Temperature and Relative Humidity Values for Denver, CO and New York, NY.

Solargraph

The Solargraph graphic interface, diagrammed in Figure 3, is a graphic space for describing the position of the sun, and largely follows the existing conventions of solar plots. In this graphic, the radius from a given center point corresponds to the solar altitude and the angle measured clockwise from vertical corresponds to the solar azimuth. Charts of this type are used to uncover correspondences between solar position and other variables impactful on building design, and are essential to the design of building elements such as solar shading masks. Unlike conventional solar plots, users may choose to represent night-time conditions by plotting solar positions below the horizon, depicted as “greyed-out” regions in the sample graphic below.

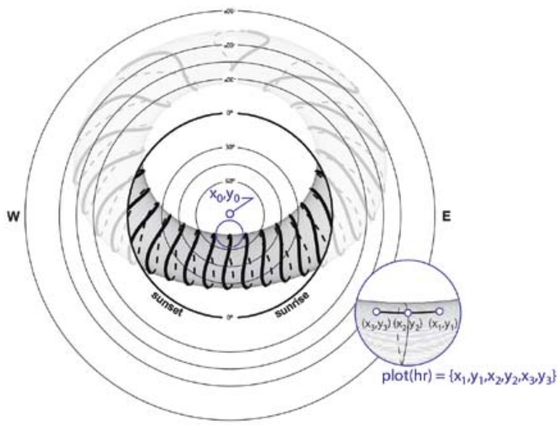


Figure 3
A Diagram of the Solargraph Graphic Interface

The Solargraph draw() method takes a latitude, longitude a dYr and a key value and draws a solar graph with default parameters to the graphic space. In this default version of the graph, a grey-scale representation of the key value for each hour is plotted in a position that corresponds to the sun's location in the sky at the selected hour.

The Solargraph plot() method takes a dYr and the index of the hour to plot, and returns a vector representing three X,Y coordinates, corresponding to the path of the sun over the course of an hour, centering on the hour in question. From these coordinates, the user may construct a line with graphic qualities of their choosing, and perform a color interpolation based on any key values they wish.

Psychrometric graph

The Psychrometric graph, diagrammed in Figure 4, is a graphic space for plotting information according to the standard ASHRAE-style psychrometric chart. In this graphic, the x-axis corresponds to the dry-bulb air temperature (°C) and the y-axis corresponds to the humidity ratio (grams of water/kg of dry air, gw/kgda). The utility of psychrometric charts in climate analysis is undisputed, as this graphic representation simultaneously captures the thermodynamic parameters of air and vapor that are very relevant to human comfort, energy use and building design.

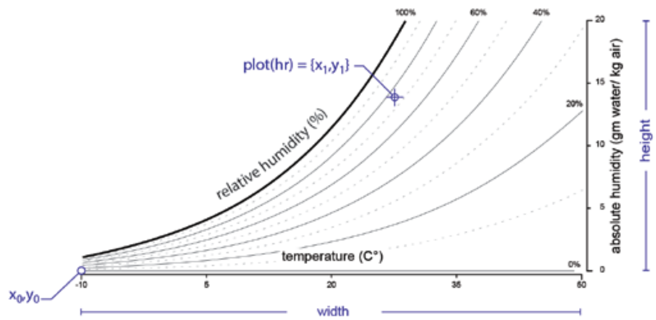


Figure 4
A Diagram of the Psychrometric Graphic Interface

The Psychrograph draw() method takes a dYr and draws a psychrometric graph with default parameters to the graphic space. In this default version of the graph, a black ellipse is plotted, corresponding to the dry-bulb temperature and humidity ratio values, for each hour of the dYr.

The Psychrograph plot() method takes a dYr and the index of an hour to plot, and returns a vector representing the relevant X, Y coordinate, corresponding to the dry-bulb temperature and humidity ratio, for the specified hour. From this coordinate, the user may draw various shapes of the their choosing (*rectangle, ellipse, etc.*) to the graphic space, and perform a color interpolation based on any key values they wish.

Miscellaneous Other Graphs

A number of miscellaneous other graphic interfaces are provided by OpenGEF at this time, as described in Tables 1, 2 and 3 and depicted in Figures 5, 6 and 7, below.

Linegraph	
Description	Provides a simple set of methods for generating standard time-value graphs.
Plot()	Takes a dYr and the index of an hour to plot. Returns a vector(s) representing the relevant X, Y coordinate(s), corresponding to the hour along the X-axis and the key value(s) along the Y-axis.
Draw()	Takes a dYr and a key value. Draws a line with default parameters to the graphic space. In this default version of the graph, a black line is plotted, with the X-axis corresponding to the year and the Y-axis to the specified key value.

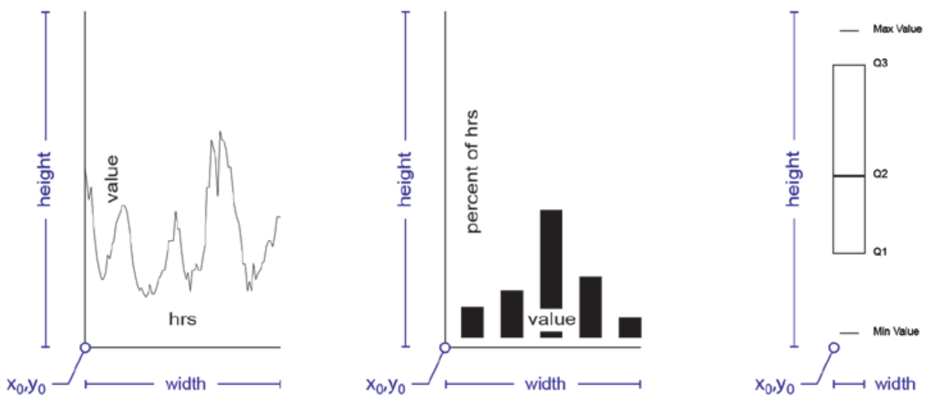
Table 1
Definition of the Linegraph Graphic Interface

Histogram	
Description	The Histogram graphic interface provides a simple set of methods for generating histograms, a standard graphic type in descriptive statistics.
Draw()	Takes a dYr, a key value, an array of interval ranges and an optional pair of colors. Draws a histogram with the key value data plotted according to the defined intervals and shaded with the interpolated color. The defined intervals may be represented by user-defined geometry (*rectangle, ellipse, etc.*)

Table 2
Definition of the Histogram Graphic Interface

Boxplot	
Description	Box plots are a standard tool in descriptive statistics for depicting distributions of values across a sample set, summarizing this sample set as five numbers: the largest observed value (max), the upper quartile value (q3), the median (q2), the lower quartile value (q1), and the smallest observed value (min).
Draw()	Takes a dYr and a key value. Draws a black and white box plot with the X-axis corresponding to the user-selected width and the Y-axis corresponding to the distribution of the specified key value for the year mapped to the height.

Table 3
Definition of the Boxplot Graphic Interface



Figures 5,6,& 7
Diagrams of the Linegraph, Histogram and Box Plot Graphic Interfaces

Web-Based Infrastructure

Procedures required by existing tools for acquiring and processing weather data present obstacles to designers and contain some computational inefficiencies that can be overcome through a web-based approach. Most existing tools, including the previous iteration of the GEF framework, require users to identify weather data files from the appropriate online sources, download them to the proper directories on their local computers, and load them into memory before producing an evaluative graphic. Some of these tools provide limited support for this procedure. A simpler approach is to direct the evaluative tool to load the appropriate data directly from a universally accessible online location, bypassing the need for the user to manipulate data files. One tool, Autodesk’s Climate Server, which is integrated into the Revit platform, has implemented a similar approach [2]

A web-based approach also offers computational efficiencies. The standard EPW weather file format includes 35 values for each of the 8760 hours of the described year, which totals to 306,000 individual pieces of information in each data file (not including header data). Any given evaluative graphic requires only a small fraction of these values, rarely needing any more than four values for each hour. A web-based approach allows individual graphics to request only the values required for a specific evaluation, eliminating the processing and storing of extraneous data. Moreover a web-based tool allows the user to perform his/her analysis directly from the most up to date version without the need to update the tool to the lastest available version.

Implementation

At the time of writing, remote weather data acquisition for generating dYr data objects has been fully implemented, and data formatting has been completed on a select group of weather data files.

In previous implementations, the GEF framework provided methods for producing dYr data objects by parsing local EPW files via the ParserEPW() method, which required only the file path to the local file. In the current implementation, this method has been replaced by the LoadDyr() method, which requires the weather station identifier (WMO) and an array of key strings identifying the values to be loaded. Rather than processing and storing all the data values contained in a local EPW file, this command retrieves information from the OpenGEF webserver, and processes only those values specified. At the time of writing, remote values are not stored in a formal web database, but are simply organized into a publically accessible folder structure containing comma-separated text files. This new implementation allows access to a range of values commonly required for climate evaluation but not included in the standard EPW format. These values have been pre-calculated and stored for efficient retrieval, and include: PPD, PMV, Wet-bulb Temperature, Dew Point, Humidity Ratio and Enthalpy.

For example, to load dry-bulb (included in a standard EPW file) and humidity ratio data (not included in a standard EPW file, but pre-calculated and stored on OpenGEF servers) from the EPW file for New York - LaGuardia, one may execute the following command:

“LoadDyr(725030,{”dryBulbTemp”,”humidRatio”});”

Future Work

The infrastructure for storing and retrieving climate data over the web is an important step toward a fully web-based visualization platform. Future steps toward this goal will include:

- Replacing the current file-folder implementation with a more formal web database, which will allow for faster data retrieval.
- Pre-calculating a more extensive range of weather data values, and include values that will require remote calculations based upon user-definable variables (ex: PPD, which is dependent upon variables that the user may want to define).
- Implement a web-based user interface, such that users can create evaluative graphics by entering their scripts into a web form, enabling more effective data analysis.

Sample Results

To assess the features of Open GEF, a comparison with the thermal comfort analysis in the psychrometric chart of Climatic Consultant v5.3 (CC) has been performed. New York LaGuardia Airport EnergyPlus weather file has been used as input. Figure 8 shows the output of Climate Consultant with the bioclimatic chart deleted, because it is not yet part of Open GEF. CC calculated that 977 hours out of 8760 are within the comfort range according to ASHRAE standard 55-2004 using the PMV method. The assumptions of the PMV model (Metabolic activity = 1.1 met, summer clothing = 0.5 clo and winter clothing = 1 clo) are input on a separate pane previous to the psychrometric display pane. Changing these parameters requires switching back and forth between these two panes. The thermal comfort range is reported in blue. The data points that lay within the comfort range are plotted in green and the others in red, thereby representing the level of comfort for each hour as a binary condition: “comfortable” or “uncomfortable”.

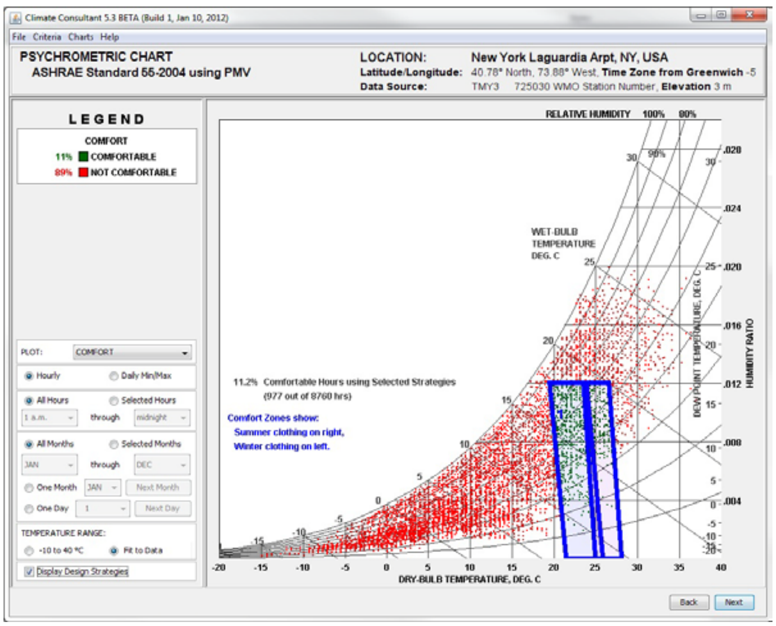


Figure 8

Climate Consultant v5.3 output of thermal comfort analysis in the psychrometric chart of the weather data for New York-LaGuardia.

Figure 9 shows the output of Open GEF with the psychrometric chart on the right, showing the weather data points, a colorbar chart on the bottom of the page, reporting the calculated Predicted Percentage of Dissatisfied (PPD) people, a histogram on the left, showing the frequency distribution within relevant PPD bins, and a series of parameters to be controlled to the top left page. The thermal comfort analysis is reported simultaneously in multiple ways, in the psychrometric chart, in the colorbar, and in the histogram. The colorbar allows the user to detect hourly and daily patterns usable in sustainable design approaches (e.g. night purge ventilation). The histogram gives the user a finer assessment of the thermal comfort distribution, the divisions of which follow the European standards for thermal comfort. Climate Consultant and the ASHRAE standard 55 represent thermal comfort compliance as a dichotomous variable, comfortable or uncomfortable people, without specifying how comfortable or how far from comfort people are and which are the predicted percentage of dissatisfied (for ASHRAE when PPD is less than 10% then the zone could be define as comfortable). The background of the psychrometric chart shows the PPD value for each point in the graph. The PPD values and the comfort range depend on the clothing insulation and the metabolic activity. It is particularly useful to instantaneously see how these parameters affect the comfort range and the comfort evaluation. This ability of Open GEF underlines a potential sustainable strategy that is not yet widely adopted, to allow users to adapt their clothing to the outdoor/indoor conditions. As in Climate Consultant, the Open GEF allows the user to select the hours/days of the year to be plotted. Boxplots of the dry bulb temperature and of the humidity ratio are shown close to the axes giving a quick summary of the data.

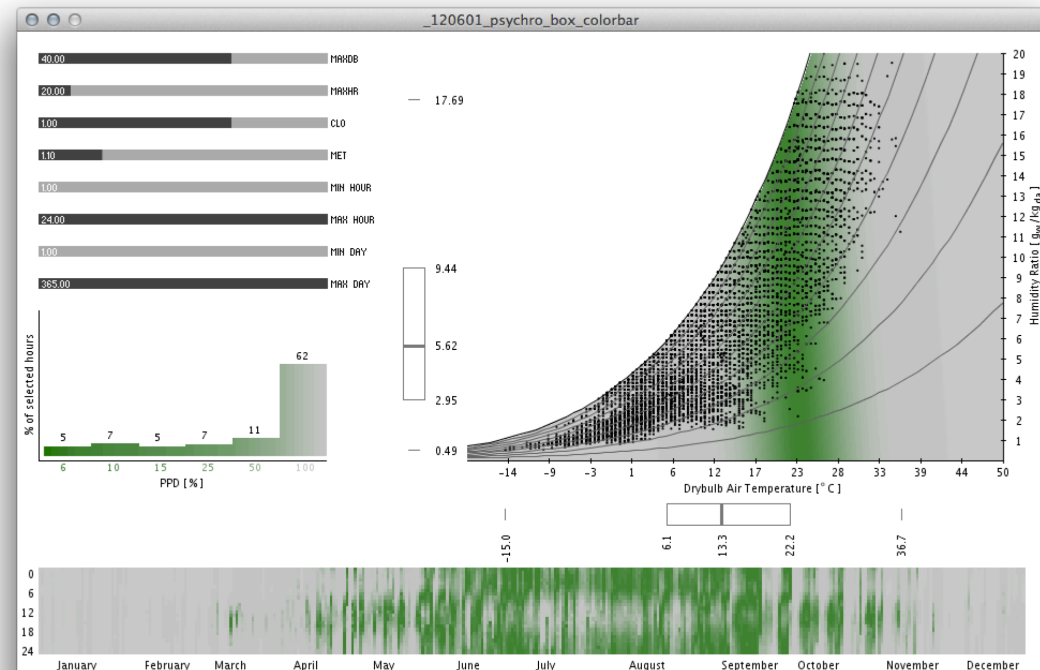


Figure 9
Open GEF output of thermal comfort analysis in the psychrometric chart of the weather data for New York-LaGuardia.

References

Marsh, A., 2000. Playing Around with Architectural Science. In ANZAScA Conference Proceedings. Annual Conference of the Australian and New Zealand Architectural Science Association. Adelaide, South Australia. Available at: http://andrewmarsh.com/andrew/publications [Accessed June 2, 2010].

Milne, M., 2010. Energy Design Tools. Energy Design Tools Developed at the UCLA School of Architecture. Available at: http://www.energy-design-tools.aud.ucla.edu/ [Accessed July 28, 2010].

Steinfeld, K. et al., 2010. Situated Bioclimatic Information Design: a new approach to the processing and visualization of climate data. In Proceedings of the ACADIA ’10 Conference. ACADIA ’10. New York, NY: Association for Computer Aided Design in Architecture.

[1] www.mistaya.ca/software/dview.htm

[2] Autodesk Climate Server: http://www.youtube.com/watch?v=LpjxXOZELu8
http://usa.autodesk.com/green-building-studio/