**Big Climate Data Analytics: Effective Knowledge Discovery from Colombia’s Weather Data**

By

Roland Hudson

A DISSERTATION

Submitted to

The University of Liverpool

in partial fulfilment of the requirements

for the degree of

MASTER OF SCIENCE

24/11/2018

ABSTRACT

**Big Climate Data Analytics: Effective Knowledge Discovery from Colombia’s Weather Data**

By

Roland Hudson

The goal for the project is a big climate data analytic system that enables knowledge-discovery (KD) and provides recommendations for construction strategies given a geographical location and the associated weather data.

In Colombia the weather is massively varied due to high altitude mountains, coastlines and effects of phenomena such as el Niño. Tropical weather is unlike the weather in the Northern and Southern latitudes as there are no seasons instead daily variations dominate.

Recommended content:

* Introduce into the research or technology area addressed in the dissertation
* Identify specific problems and challenges identified in your work
* Describe what is achieved and proposed in the project
* Briefly describe the structure of the report
* Mention what is the application area and benefits of the proposed solutions

DECLARATION

[check with the recent templates]

I hereby certify that this dissertation constitutes my own product, that where the language of others is set forth, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions, or writings of others.

I declare that the dissertation describes original work that has not previously been presented for the award of any other degree of any institution.

Signed,

ACKNOWLEDGEMENTS

[any acknowledgement you want to express]

TABLE OF CONTENTS

# Chapter 1. Introduction

## Scope

This dissertation examines how a big weather data framework combined with knowledge discovery techniques can help define localized approaches to building design and construction that improve living conditions and reduce energy consumption in Colombia.

## Problem statement

The construction industry is responsible for massive amounts of greenhouse gases (Dimoudi and Tompa, 2008). Through their lifecycle buildings consume 50% of all total energy demand and contribute 50% of all CO2 emissions.A primary goal for environmental construction is to reduce the energy consumed by buildings, estimated to be around 40% of the total global energy consumption (Omer, 2008). Much of this energy is expended on heating lighting and cooling. Energy efficiency can be defined as the minimising the amount of energy consumed to achieve thermal comfort for occupants. Currently the energy required to maintain thermal comfort accounts for 60-70% of energy consumed in non-industrial buildings (Omer, 2008). Better understanding of human response to climatic context (bioclimatic design) can result in buildings that require less energy for heating and cooling (Olgyay and Olgyay, 2015, p11).

Low-energy construction strategies exist that can minimize or remove the need for heating and cooling in buildings. For example; orientation of buildings, sizing and positioning of openings, choice of materials and use of passive heating and ventilation. These techniques require an understanding of local and regional climate conditions across different time frames.

Detecting climate patterns in the Colombian context is challenging due to weather variations caused by extreme changes in altitude over short distances, proximity to different ocean bodies and phenomena such as el Niño and la Niña. This underlying complexity is complicated further by subtle and inconsistent seasonal patterns associated with tropical latitudes.

The Colombian context contrasts with Northern and Southern latitudes where consistent seasonal variation dominates the climate making it easier to identify what design strategy to apply. Colombia’s complex climate patterns make identifying appropriate, localized, low-energy construction strategies difficult. Typical construction is often unable to cope with regional and daily variations in weather, people live and work in uncomfortable conditions often too hot and too cold. To correct these issues heating and cooling need to be installed which is expensive, costly to run and produces emissions.

### Complexity of existing workflow

For an architect or engineer the current workflow to determine what low-energy construction strategies to use is a multi-step approach:

* Analysis of climate data - analysis of historical weather data compiled into files representing typical meteorological years (TMY) with a range of variables stored for each hour of the year.
* Biological evaluation – data is plotted on a psychrometric chart (physical and thermal properties of moist air) to diagnose the hours of the year lying outside a predefined zone of thermal comfort.
* Identifying technological solutions (design strategies) – in response to the biological evaluation technological approaches can be identified to ensure more time within the comfort zone. For example; site selection and orientation to gain or minimise solar radiation.
* Developing the architectural application – the designer synthesises the previous three steps into a design proposal.
* Simulation may be undertaken to confirm the design approach or optimise a chosen strategy.

The designer must also consider usage patterns of the building making the process more complex. Buildings are rarely occupied constantly, depending on use, occupancy can vary daily (residential buildings are often occupied evenings and night time), weekly (office buildings not in use at weekends) and through the year (schools and universities have seasonal holiday periods). The design approach for two buildings with different uses in the same location will not be the same. Zones within certain buildings will have different orientations, usage patterns and activity types, each zone will have varied design needs.

To address these issues the project proposes an application based around Colombian weather data that combines datamining techniques with expertise of the low-energy building construction domain to link specific design strategies with a specific location and time frame.

## Approach

The project methodology begins with a literature review, then an IT artefact is developed following a specific development model. Finally, the artefact is evaluated qualitatively and quantitively.

### Literature review:

The literature review surveys low-energy architectural design strategies and how these relate to weather and climate conditions and design criteria. The spatiotemporal nature of climate data is examined and techniques for data mining meteorological data are discussed. Knowledge discovery as a process model is defined and its application to pattern seeking with climate data is addressed. The review also includes big data tools, components, applications and architecture in climate science. With a focus on the use of workflow management for scientific big data systems and appropriate analytics methods.

### Artefact

The IT artefact is an application that facilitates big weather data analytics for Colombia. By integrating analytics and visualisation, the application enables data exploration and knowledge discovery by linking construction strategies with geographical location and related historical weather data.

### Application development methodology:

Development follows a UML based agile model driven design method with a distribution of development phases and activities show in Figure 1. Each activity involves specific techniques and can be linked to deliverables and specific diagramming or modelling methods (Table 1).

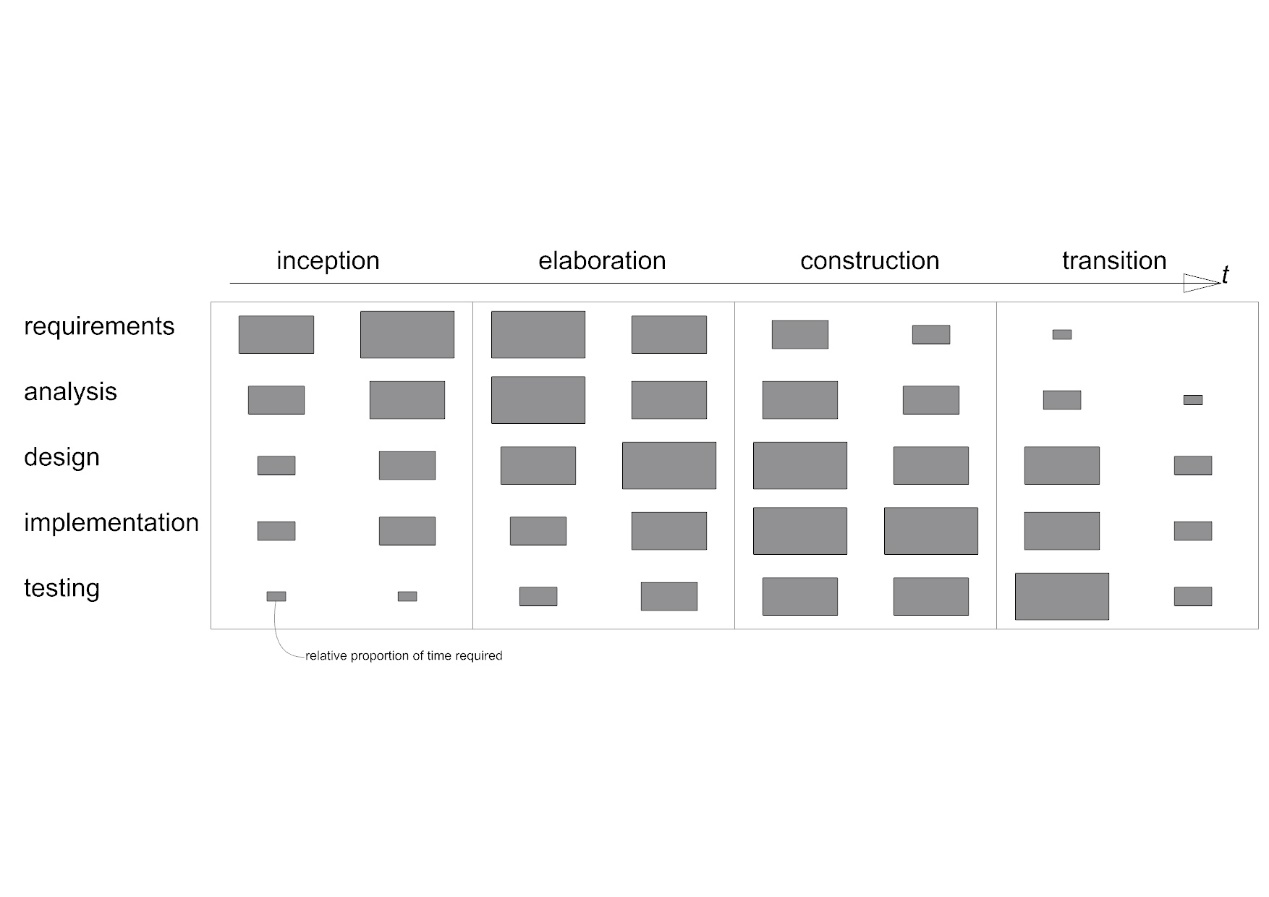
****

Figure 1 Agile model driven design phases

|  |  |  |  |
| --- | --- | --- | --- |
| **Activity** | **Techniques** | **Deliverables** | **Diagrams** |
| **Requirements capture and modelling** | Text descriptions of use cases and requirements  Use case modelling  Architectural modelling, prototypes | Use case model, requirements list, initial architecture | Use case, Package |
| **Requirements analysis** | Use cases analysed to extract required objects. Interactions between objects identified – communication diagrams developed | Analysis models | Class, Object, Communication |
| **System and architecture design** | Design patterns identified | Overview design and implementation architecture | Package, Component, Deployment, Class |
| **Class design** | Class and object modelling, Interaction modelling. State modelling, Design Patterns  prototypes | Design models | Class, Object, Sequence, State machine, Package |
| **Interface design** | Class and object modelling, Interaction modelling, State modelling, Design Patterns, prototypes | Design models, interface specification | Class, Object, Sequence, State machine, Package |
| **Data management design** | Class and object modelling, Interaction modelling, State modelling, Design Patterns, prototypes | Design models, data specification | Class, Object, Sequence, State machine, Package |
| **Construction** | Programming, component reuse | Constructed system, documentation |  |
| **Testing** | Programming, test planning and design, testing | Test plans, test cases, tested system |  |

Table 1 Development activity details

### Qualitative evaluation of artefact by domain experts:

Qualitative evaluation is based on software walkthroughs, presentations and interviews with domain experts. Expert will be presented with a series of studies and results from the application, opinions of experts will be captured and summarised. With expert review it will be possible to evalauate if useful localised construction approaches can be generated.

### Quantitative evaluation of output from artefact results:

Quantitative evaluation involves statistical comparison of different analytic methods, and different configurations of those methods. Quality measures for methods applied (distance metrics for clustering). The application is also evaluated through verification, validation and testing – tests identified during the requirements specification and revisited through the prototyping stages.

## Outcome

The goal for the project is a big climate data analytic system that enables knowledge-discovery to support decision making in the design and construction of buildings in Colombia. Specifically, the project aims to enable the search for patterns in climate data that can be linked to localized, climate-responsive design and construction strategies. Application of these strategies can lead to buildings that perform better in terms of production costs, life-time running costs (reduced heating and cooling) and occupant comfort.

The project proposes that by combining a big data workflow management infrastructure with spatiotemporal data mining techniques localized approaches to building design and construction that respond to the unique weather conditions in Colombia can be identified.

To achieve this goal a big data system is proposed that follows current best practices for the storage, processing, analysis, management and visualization of the data. Specific focus will be on enabling the analytics and visualization that enables knowledge discovery through data mining. Knowledge from the data will support decision making for the design and construction of buildings to potentially improve living conditions (quality of life and wellbeing) and reduce energy consumption in buildings.

# Chapter 2. Background and review of literature

## Background

The literature review spans various themes, first low-energy environmental design strategies are examined, what they are and how they are represented is defined. The spatiotemporal nature of climate data is described, this is contrasted with classical data mining and key differences are determined. Data mining methods applicable to spatiotemporal are discussed and challenges for these techniques identified. Data mining is positioned as one of three key elements in knowledge discovery processes, the relationship with the other two; domain expertise and data management is described.

Relevant big data concepts are explored, this includes the use of workflow management systems for big data science applications and how Infrastructure as a Service offered by cloud service providers may be an applicable service model. Applications that have addressed climate data tasks using big data analytics tools are identified. Common to many of these applications is Apache Spark’s machine learning library, this is identified as a key tool for this project and is described.

## Literature review

### Low Energy Environmental Design Strategies

#### Human thermal comfort

Can be understood as a combination of temperature, relative humidity, air movement and radiant temperature, giving a state of mind where a person requires no change in current conditions (ASHRAE, 2013) or a state where minimal extra energy is require to maintain the human balance (Manzano-Agugliaro *et al.*, 2015).

#### Psychrometric chart

Is used to map interrelationships of thermal conditions of the environment (Figure 2). A zone of human thermal comfort can be plotted following standard guidelines (ASHRAE, 2013). Hourly data points can be plotted on the chart, where they fall outside of the comfort zone the design of the building and/or services must be adapted to provide comfort.

The psychrometric chart was adapted by Givoni (1992) as the building bioclimatic chart and later Manzano-Agugliaro *et al.* (2015) to include specific zones representing strategies that can be applied to a design to extend the zone of comfort. Conventional heating and air conditioning are recommended only at extremes. The strategies include what Lechner (2009, p9) describes as tier 1: basic building design (building orientation, position and size of openings, material specification) and tier 2: passive systems (passive solar heating, night-time flush cooling). Correct design decision making at these levels can reduce building energy consumption by up to 80% (Lechner, 2009, p9).

Each design strategy can be further specialised into a series of instrumental techniques (Manzano-Agugliaro *et al.,* 2015) which can include regionally specific and traditional construction and more experimental methods. Climate Consultant software (Milne, Liggett and Benson, 2009) generates a prioritised subset of recommendations from a set of 68 design guidelines each associated with a zone on the psychrometric chart.

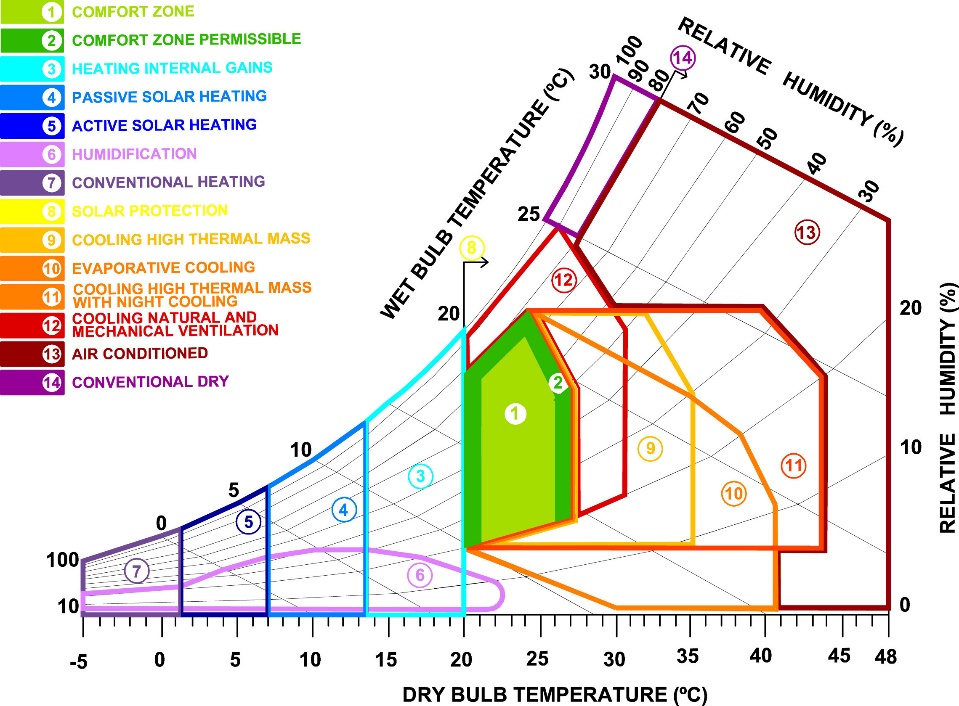


Figure 2 Manzano-Agugliaro *et al.* (2015) adapted version of the psychrometric chart

#### Comfort Indices

Colombia’s national institute of hydrological, metrological and environmental studies (INSTITUTO DE HIDROLOGIA, METEOROLOGIA Y ESTUDIOS AMBIENTALES (IDEAM)) has proposed a method to calculate climatic comfort (IDEAMCI) in Colombia (González, 1998). Based on a system (by unreferenced authors Leonardo Hill and Morikofer – Davos) for refrigeration power it assumes when the human body is surrounded by air temperatures lower than 36.5C and the air is constantly renewed by wind a cooling effect will be experienced. IDEAMCI extends this by including relative humidity at three discrete ranges of altitude and produces an index on a scale of zero to fifteen or more (hot to cold). IDEAMCI seeks to resolve an understanding at both a spatial and temporal resolution (hourly) across an entire Country.

Most recently the Universal Thermal Comfort Index (UTCI) has been proposed citing the shortcomings of preceding indices and limits in terms of confinement to specific applications (Jendritzky and Höppe, 2017).

Indices exist to help those not familiar with the domains of thermos-physiology or biometeorology better understand the implications of climate, activity and clothing on the human body (Höppe, 1999) and are not necessarily designed to support decision making for construction. Notably UTCI is the result of a multi-disciplinary effort the development team did not include practitioners from the built environment. Indices are useful in later design stages to fine tune heating, cooling and ventilation systems or to refine material specifications (insulation and glazing) ensuring certain thermal comfort levels are achieved. Indices convert a multivariate problem into a single value removing the visibility of the underlying data making it difficult to determine design strategies that respond to the cause of discomfort.

Architects concerned with designing comfortable environments need to be aware of the full set of climate variables early in the design process to develop appropriate strategies. Strategies inform siting and orientation of the building, size and number of openings, materials and integration of technological systems to assist passive cooling and heating. Such early stage design decisions are more probable to have an impact on the cost and function of the finished building and are cheaper to implement (CURT, 2004, p4). Despite their short comings these indices can be helpful to compare and evaluate clustering solutions.

### Climate data is spatiotemporal

Data collected for climate science is classed as spatiotemporal (ST) data (Atluri, Karpatne and Kumar, 2017). Mining and knowledge discovery with ST data differs from classical data mining due to its properties and the variety of data types (Faghmous and Kumar, 2014) . ST data is heterogenous, it is not identically-distributed, instead ST data demonstrates non-stationarity in space and time. Auto-correlation exists in ST data two nearby locations are not independent but are correlated. ST data can be categorised into four types; event data (start and end of heavy rainfall), trajectory data (path of a cyclone), point data (temperature measured in a moving set of weather balloons) and raster data (temperature measured across a fixed set of weather stations).

Classical data mining uses features with labels (Atluri, Karpatne and Kumar, 2017) in ST instances can be defined as points, trajectories, time-series, spatial maps and raster. Search for similarities between these instances involves clustering, classification, pattern discovery and relationship mining.

Various data mining methods applicable to climate data are described by Atluri, Karpatne and Kumar (2017). *Relationship mining* involves linking changes in one variable to other phenomena. *Clustering* on instances and

ST-DBSCAN is recommended for finding anomalies. *Frequent pattern mining* includes searching *motif patterns* in time-series and in networks for sets of distant locations experiencing similar climatic conditions with consistent temporal activity. Change detection can identify transitions or deviations in behaviour. Faghmous and Kumar (2014) add *network-based analysis* to undertake relationship and pattern mining in gridded and non-gridded climate data sets. *Pattern mining* for Faghmous and Kumar includes searching for user defined patterns, empirical orthogonal functions and spatiotemporal clustering.

### Examples of applied ST data mining methods applied to climate data

* Clustering methods have been successfully applied to climate classification (Forsythe, Blenkinsop and Fowler, 2015) (Netzel *et al.*, 2016)
* Self-organising maps (SOM)’s have been used to extract features from data (Liu, Weisberg and Mooers, 2006) and applied to metrology and oceanography (Liu and Weisberg, 2011)(Liu and Weisberg, 2005).
* Delta-maps (Fountalis, Bracco and Dovrolis, 2014) (Bracco *et al.*, 2017) group nodes in a network according to homogeneity, these have beenapplied to precipitation and sea surface temperatures. Robustness analysis of networks generated can be evaluated using link maps, area strength and s-core decomposition.

##### K-means clustering applied to climate classification

Recent research suggests that unsupervised learning methods are applicable to climate analytics and data mining. Fovell & Fovell (1993) studied hierarchical clustering in the US and sought a best method based on the minimising of bias in terms of method, latent and information. Redundancy problem is discussed when two or more highly corelated variables are included (little unique information added and repeats data magnifies the) and PCA used scaling variables, withholding then adding variables to observe the impact

Degaetano's (1996) study sought to develop an ecosystem management and planning guide by defining mesoscale climate zones in the north-eastern US. K-means was compared to Ward’s clustering technique and improvements were found in spatial distribution and homogeneity of clusters. The findings suggested k-means could produce stable clusters with minimal information bias.

Rhee *et al.* (2008) used k-means as part of a multi-step approach to delineate climate regions in the Carolinas that combined in-situ (weather station data) with remotely senses and spatially distributed data. K-means was integrated within a more complex workflow (hierarchical followed by non-hierarchical then decision trees trained on results that then classify remotely sensed data) and the study demonstrated the validity of the method for establishing clusters that were subsequently used for supervised classification of data. PCA is not used as it is lose important information in monthly time series, interpretable distance measure can be use and truncation of PCAs is not considered.

All three-above use pseudo f and CCC and suggest a consensus approach – first with hierarchical to define centroids and then non-hierarchical to determine the clusters

Clustering applied to assess climatic influences on water resources and food security in the Himalayas (Forsythe, Blenkinsop and Fowler, 2015). It showed clustering could be used to characterise the bias of gridded datasets and undertake meteorological reanalyses of climate models. K-means defined eight sub-regional climate classifications and further increases in cluster numbers defined subdivisions to each macro zone.

Netzel *et al.*, (2016) studied 32 different clustering methods and compared them to the KG Classification. The study concluded that clustering could find 50% of the climate types defined by the KG classification. The remaining classes differed in climatic character and spatial distribution but were shown to be more homogeneous and more distinct than KG climate types.

Zscheischler, Mahecha and Harmeling (2012) used k-means clustering with subsets of 5 normalized variables. When k-means was used with climate and vegetation variables similar clusters to the KG zones could be generated.

### Challenges for ST data mining

Dealing with interdependencies at multiple scales within climate data is complex and means global studies cannot be used to understand long-term local impacts (Faghmous and Kumar, 2014). Relationships in climate data may be long range and multivariate, many space-time-variable subsets exist where relationships may be found. This spatiotemporal variability makes clustering with ST data challenging. Similarly, anomalies and extremes in climate data need to be understood as multivariate cumulative extremes. Faghmous and Kumar suggest better methods are needed for validation of ST data mining. Significance testing needs randomization tests that do not break the inherent autocorrelation and performative measures are required to compare unsupervised STDM.

For Atluri, Karpatne and Kumar (2017) key challenges involve finding methods for combining multi-modal data sets and controlling granularity of partitioning to ensure substructures are not overlooked.

The need to integrate domain theory and expertise is acknowledged (Karpatne *et al.*, 2017) as a key strategy that could accelerate knowledge discovery in data science particularly where complex physical phenomena are involved.

### Knowledge discovery

Knowledge discovery integrates data mining, domain theory and data management. Knowledge discovery in data is defined as (Begoli and Horey, 2012) 1. Collection, storage and organisation of data. 2. Understanding and application of analytic methods. 3. Understanding the problem domain. To enable knowledge discovery Begoli and Horey recommend applications are made to allow researchers easy ways to interact, explore and analyse data. A variety of analysis methods should be supported inclung statisitical, data mining, machine learning, visualisation and visual analysis. Different data storage and processing mechanisms should be provided to support a variety of intermediate data structures (structured and semi-structured) required by different ananlysis methods. Data should be made as accessible as possible by using open standards, lightweight architecture and APIs to expose results. The potential for knowledge discovery in climate science has not yet been fully realised (Bracco *et al.*, 2017).

### Cloud computing

Cloud computing is defined by NIST (Mell and Grance, 2011) as a model consisting of the following characteristics; on demand service, board network access, rapid elasticity and a measured service. Three key service models are offered by cloud providers; Software as a Service, Platform as a Service and Infrastructure as a service (IaaS). IaaS provides access to cloud-based computing resources that allow the deployment and execution of arbitrary software. This service model offers three key functions of interest to this dissertation; dynamic provisioning and configuration of processing resources to run cloud-based systems, scalable storage capacity that can be used for applications, backups, archival, and file storage and Content Delivery Networks to store content and files to improve the performance and cost of delivering content for web-based systems (Liu *et al.*, 2011).

Amazon is a cloud provider offering all the different cloud service models. The Amazon IaaS includes Elastic Map Reduce (EMR) a hosted Hadoop framework that includes Apache Spark and other distributed frameworks. EMR automatically configures another Amazon product, Elastic Compute Cloud (EC2) that provides virtual cloud-based servers. With EMR clusters of virtual machines are launched on a Virtual Private Cloud (VPC). Specific analytic jobs are described as a Step on AWS, which is a distinct work unit that can run on an EMR cluster and a single cluster can have several Steps. EMR is designed to handle node provisioning, Hadoop configuration, cluster setup and tuning and automatically replaces poorly performing machine instances. EMR provides a file system, EMRFS that allows reading and writing files to Amazon’s Simple Storage Service (S3). S3 is Amazon’s object storage service providing scalability and automatic data replication is distributed across three facilities in a region.

### Big data tools

#### Workflow management for scientific big data systems

An approach for supporting scientific data analysis on large data sets in the cloud is workflow management systems (WMSs) (Buyya *et al.*, 2016). These processing tools enable acquisition of resources, scheduling of tasks, execution of data analysis and visualisation on distributed resources. Workflows are defined as a series of linked tasks in the form of directed acyclic graph (DAG).

Specific WMS platforms exist (see Askalon, Kepler, Taverna and Pegasus) some provide a graphical interface to assemble workflows costing of loops, conditionals and graph constructs. Originally developed for grid computing many have been extended to take advantage of the cost-effectiveness of cloud platforms and applicable to climate science (Figure 4) (Rodriguez and Buyya, 2017).

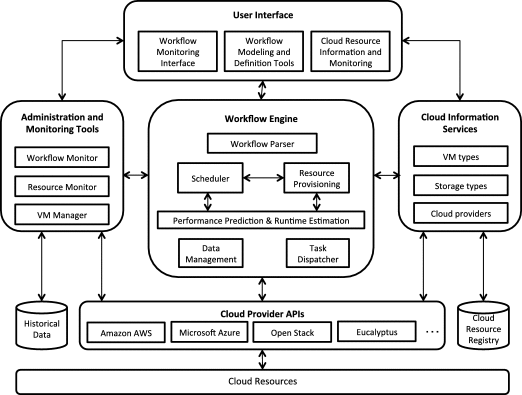


Figure 3 Reference architecture of a WMS

#### Big data application architecture and components

Avci Salma, Tekinerdogan and Athanasiadis (2017) define a big data feature model and a generic reference architecture that can be developed into an application architecture using domain-driven design. Features and architecture components can be selected based on design rules determined by the domain (Figure 4).

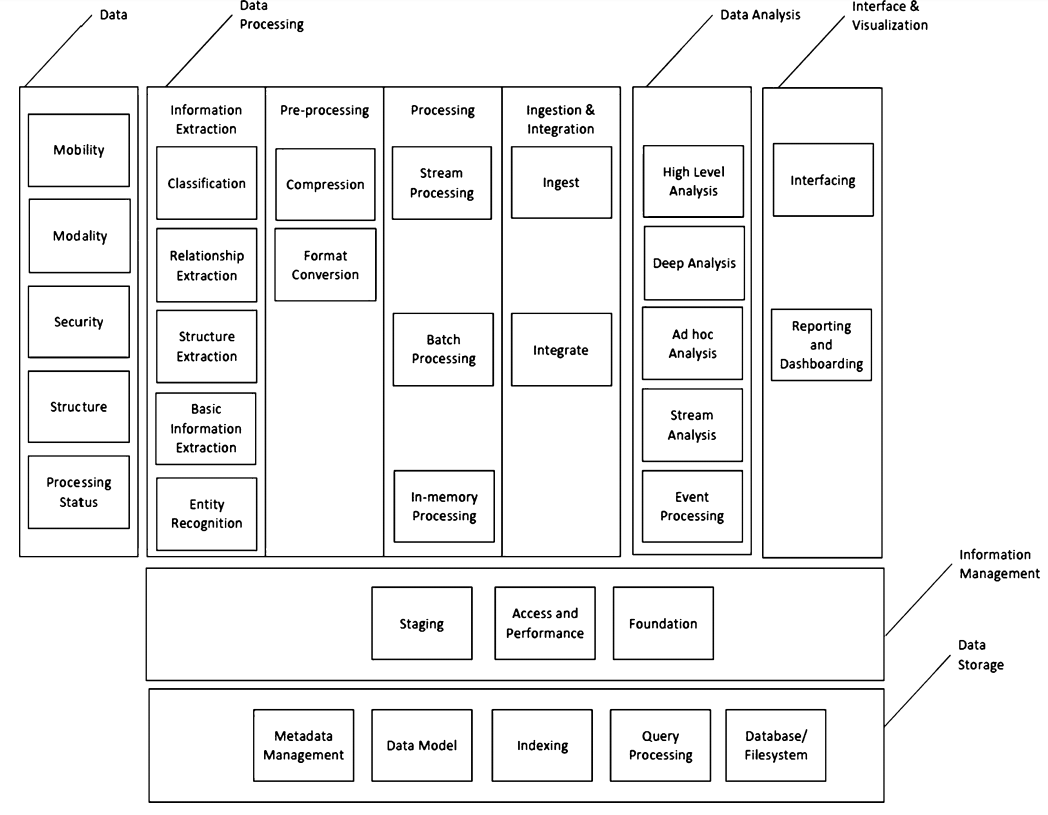


Figure 4 Big data reference architecture (Avci Salma, Tekinerdogan and Athanasiadis, 2017)

#### General applications on Weather Data

Several precedents exist describing the application of big data tools to process and undertake simple analysis on climate data. Three tools frequently occur in the literature Hadoop, Apache Spark and MapReduce.

* Apache Spark was used to process at-rest weather data to determine averages for a range of variables across a set of weather stations (Jayanthi and Sumathi, 2017).
* Hadoop and Spark were used to find points with similar weather conditions using Euclidean distances based on weather attributes using the NOAA hourly land based data set (Rodenburg and Maria Fiore, 2017).
* MapReduce was compared to Spark to calculate minimum, maximum and average values of weather parameters using data from NOAA (Chouksey and Chauhan, 2017). For smaller datasets differences in performance were less pronounced, above 8GB Spark was faster.
* Hadoop and MapReduce were used on NOAA data (Dagade *et al.*, 2015) to compare the performance of Pig and Hive to average data for each station for a single variable.
* MapReduce was used to process NOAA data (Mariam Varghese, 2015).
* K-means with MapReduce was used to cluster weather data from China (Fang *et al.*, 2014). Different size datasets were tested to evaluate processing speeds (250mb-2GB) and compared against another clustering algorithm.
* A self-organising map (SOM) (a type of artificial neural network trained using unsupervised learning) was implemented using Apache Spark and analysed IoT data (Jayaratne *et al.*, 2017).
* MapReduce enabled prediction using an artificial neural network and k-means clustering with air quality data stored with HBase, in an implementation designed to support decision making in traffic regulations in Marrakesh (El Fazziki *et al.*, 2015).

#### Apache Spark

[Apache Spark](https://aws.amazon.com/big-data/what-is-spark/" \t "_blank) is an open-source, distributed processing system commonly used for [big data](https://aws.amazon.com/big-data/what-is-big-data/) workloads. Apache Spark utilizes in-memory caching and optimized execution for fast performance, and it supports general batch processing, streaming analytics, machine learning, graph databases, and ad hoc queries

Key features of Spark (Zaharia *et al.*, 2013) are:

* fault tolerance through a parallel recovery mechanism
* tolerance of stragglers (slow nodes)
* a processing model based on discretised streams (D-Streams)
* integration with batch and interactive query models such as MapReduce
* dynamic load balancing (Das, Zaharia and Wendell, 2015)

Fault recovery in other streaming systems is based on times data replication which is expensive in terms of time and hardware and results in long recovery times and problems for handling stragglers. Systems such as Storm, TimeStream MapReduce Online use continuous operators where operators receive each record, internal states are updated, and new record sent. The Spark Streaming / D-Streams model contrasts with the long-lived operators approach by structuring the computation as a sequence of stateless, batch processes issued at short time intervals. Resilient Distributed Datasets (RDDs) are used to keep data in memory and by tracking the graph of operations used to produce each RDD it can be recovered without on disk replication. Faults on nodes are handled using parallel recovery which means on failure of a node all other nodes in the cluster work to rebuild the lost RDDs.

Clustering options with spark

Clustering evaluation with spark WSSSe and BDDunn and BDSilhouette (Luna-Romera *et al.*, 2016)

within set sum of squared errors (WSSSE) (sum of squared distances of points to their nearest centre) that is provided by Spark’s KMeansModel.computeCost() method (*KMeansModel (Spark 2.3.2 JavaDoc)*, no date)

On the selection of numbers of clusters

Silhouette and Sum of Squared Errors (Thinsungnoen *et al.*, 2015)

Elbow method (Kodinariya and Makwana, 2013) and (Nikolaou *et al.*, 2012)

#### Challenges for big data analytics applied to climate science and construction

The use of big data analytics is underutilised in the domain of weather data (Jain and Jain, 2017), better big data weather forecasting could benefit the construction industry by helping build more efficient buildings and provide an “*improved quality of life”.*

## Summary

Key aspects to emerge

Low-energy design methods are concerned with construction techniques that allow environmental conditions within a building to be adjusted to try an achieve human comfort.

Proposed application should be regarded as a Knowledge discovery tool as it will combine data collection and storage combines domain expertise, data mining and

To facilitate the KD a workflow management system is required which needs to be integrated with an

IaaS system that provides flexible computing and analysis with Apache Spark

Clustering Datamingin techniques that appear particular applicable to climate data – clustering – Spark includes several clustering methods – the review demonstrates its popularity in the domain

# Chapter 3. Analysis and Design

## System actors

The primary actor in the system is an Environmental Designer (ED), an architect or engineer whose aim is to develop energy-efficient designs for buildings in specific geographical locations. The ED’s goal is to minimise the need for heating and cooling systems while maintaining thermal comfort for occupants reduce energy consumption.

To achieve these aims the ED needs to understand the interrelationship of occupant’s thermal comfort, topography, local weather conditions, annual solar path and larger scale climatic patterns and examining how these interact with the anticipated usage patterns of the building.

Equipped with this knowledge the ED can apply a hierarchy of design strategies to develop the design. The first level relates to designing the form and fabric of the building (for example the location, orientation, materials of the building). The next level involves identification of potential passive strategies (for example, designing direct solar heating and natural ventilation into a building). The final level is the specification of mechanical systems to heat and cool, potentially using renewable energy sources. The first two levels are the concern of this project.

Other actors identified are a system administrator and data scientist. The administrator’s role is the configuration of systems settings such as access and security. The data scientist working in the field of environmental design is also identified as a system user. In this case the use is like the ED but with additional goals of management of the datasets, developing and managing new analytical techniques, visualisation and statistical methods that the ED has access to.

## General Use Case Analysis

The ED’s goal is to run an analytic or data mining technique on a set of climate data to identify patterns that can be visualised and linked to with specific design strategies. An example analysis / data mining problem could be framed as:

*For the coastal regions of Northern Colombia what design strategies can be identified for afternoon weather conditions during the first three months of the year, using a k-means clustering approach. Visualise the clusters on a map and indicate the design strategies and considerations applicable to each cluster.*

To address this broad aim data storage, analytical processing and visualisation should be developed as a proof-of-concept using a public cloud-based infrastructure as a service (IaaS) to take advantage of the low-cost capital investment, flexible infrastructure, performance and the potential for collaboration.

The ED should be able to define this process via an online interface that allows the storage, editing and reuse of previously defined processes. The ED requires the ability to start, pause, stop and cancel the analysis process. Prediction of runtime and monitoring of workflow progress is also desirable. Results should be stored and accessible for online visualisation and statistical summary and comparison. Resulting graphics and summaries should be available for download in formats for use in excel and as high-quality images and vector graphics for use in reports once the process is complete.

This general use case can be subdivided into four phases:

### 1. Define workflow

To define a process or work-flow (Figure 4) the ED must be able to specify a dataset or collection of datasets selected from a set of preloaded data. From the data the ED needs to define a geographical region or single point to study. The ED requires a high level of control of the temporal dimensions of the data. They will define the start and end dates and may need to specific that analysis takes place on recurring time-periods within the data (such as an afternoon in a specific season). The ED needs to specify which variables from the data to analyse. Once the dataset and spatial and temporal subset has been defined and the dimensionality specified, the ED will select an analytic or data mining technique from a set of predefined (and described) methods. At any point in this process the ED may need to save, save as and or edit the defined work-flow.

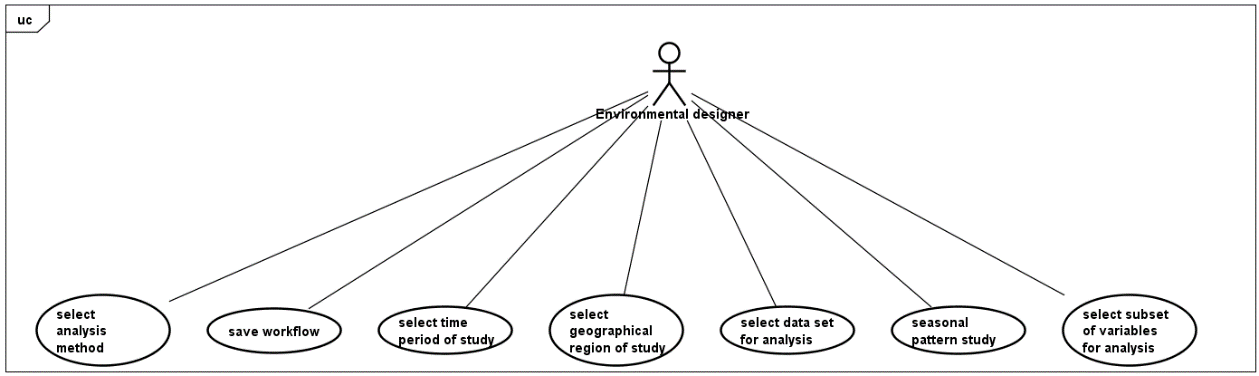


Figure 5 Define workflow

### 2. Run workflow + monitor resources

Once the work-flow is defined the ED will submit it for processing and its progress will be monitored in terms of its status (ready, executing, staging, completed) (Figure 5). The ED may also need to monitor the state of processing resources. During processing the should be able to stop or cancel the workflow.

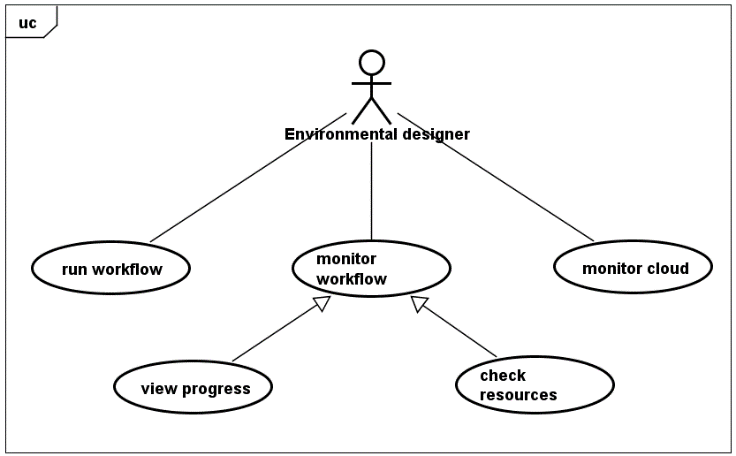


Figure 6 Run and monitor workflow

### 3. Output + visualise results

Following analysis results from the workflow shall be stored and accessible to the ED (Figure 6). Numerical and statistical summaries of the results will be generated by the ED using one of a range of predefined methods. These results would be stored and available for download in formats that can be further analysed or shared in spreadsheets. The ED may need to use simple visualisation of results (histograms, pie charts, line charts). Visualisation tools should include the ability to represent results by geospatial mapping. The graphics generated will be downloaded as high-quality images or vector graphics and used in reports. Within the interface the ED may need to compare two or more visualisations side-by-side to undertake a visual analysis. The ED also needs to share visualisation results with collaborators by providing a secure link to a webpage where the graphics can be viewed and download.

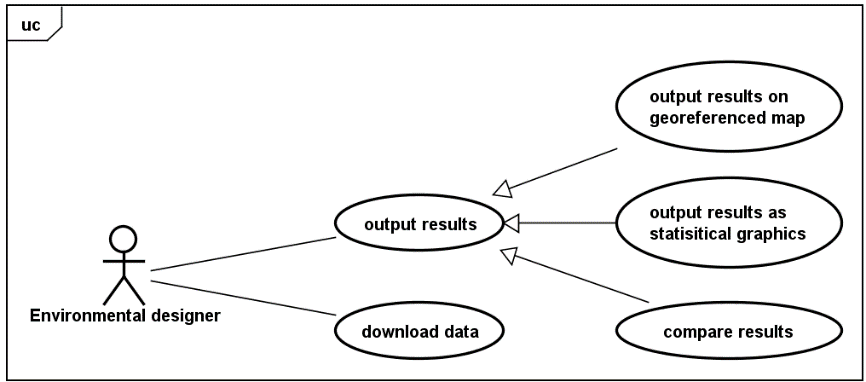


Figure 7 Output and visualise results

### 4. Manage design strategies

The design strategies specific to individual ED’s, it is necessary, therefore, to manage the predefined general design strategies used by the system (Figure 7). Some EDs will wish to add new strategies, removing unwanted strategies and editing existing ones. A standard unambiguous method for describing a strategy is required.

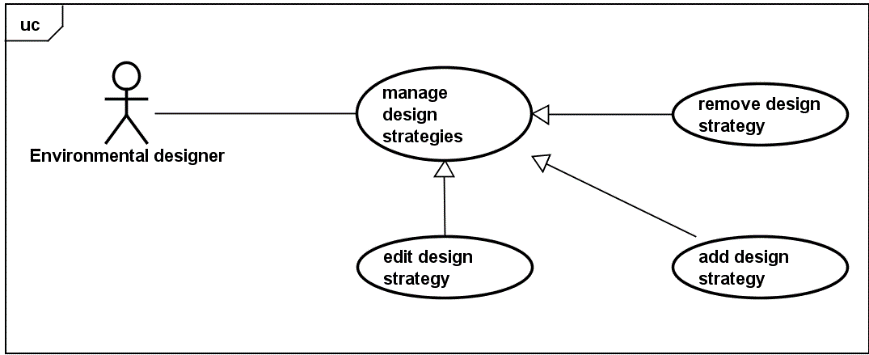


Figure 8 Manage design strategies

### Proposed System Architecture

Figure 8 shows the high-level system architecture based around Infrastructure as a Service (IaaS) provided by AWS cloud infrastructure. The prototype focuses on the items shown in red: workflow and design strategy management system, visualisation and analysis. Each of which communicate with elements of the AWS Cloud via the API, S3 for reading and writing data and EMR for submitting and controlling analytics. AWS Physical hardware on Elastic Compute is indirectly via configurations specified for EMR. Full security for a range of users is considered beyond the scope of the project, however for local prototype application development AWS Credentials are required to access the AWS API. Credentials are stored locally and automatically instantiated using the AWS SDK toolkit within the development environment. Each of the key elements in the overall architecture can be decomposed into lower level subcomponents shown in Figure 9.

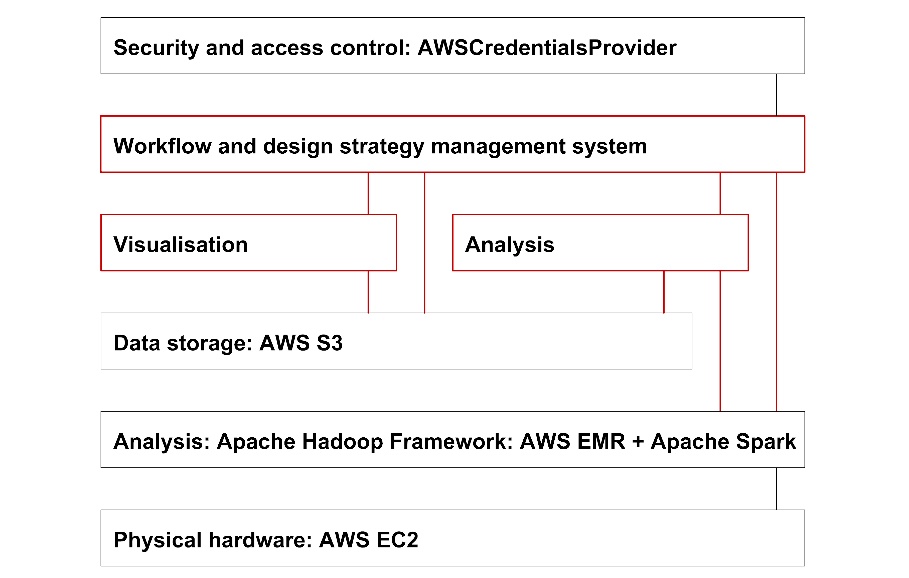


Figure 9 Proposed architecture (extended from: Lopes, Palmer and O’Sullivan, 2017)

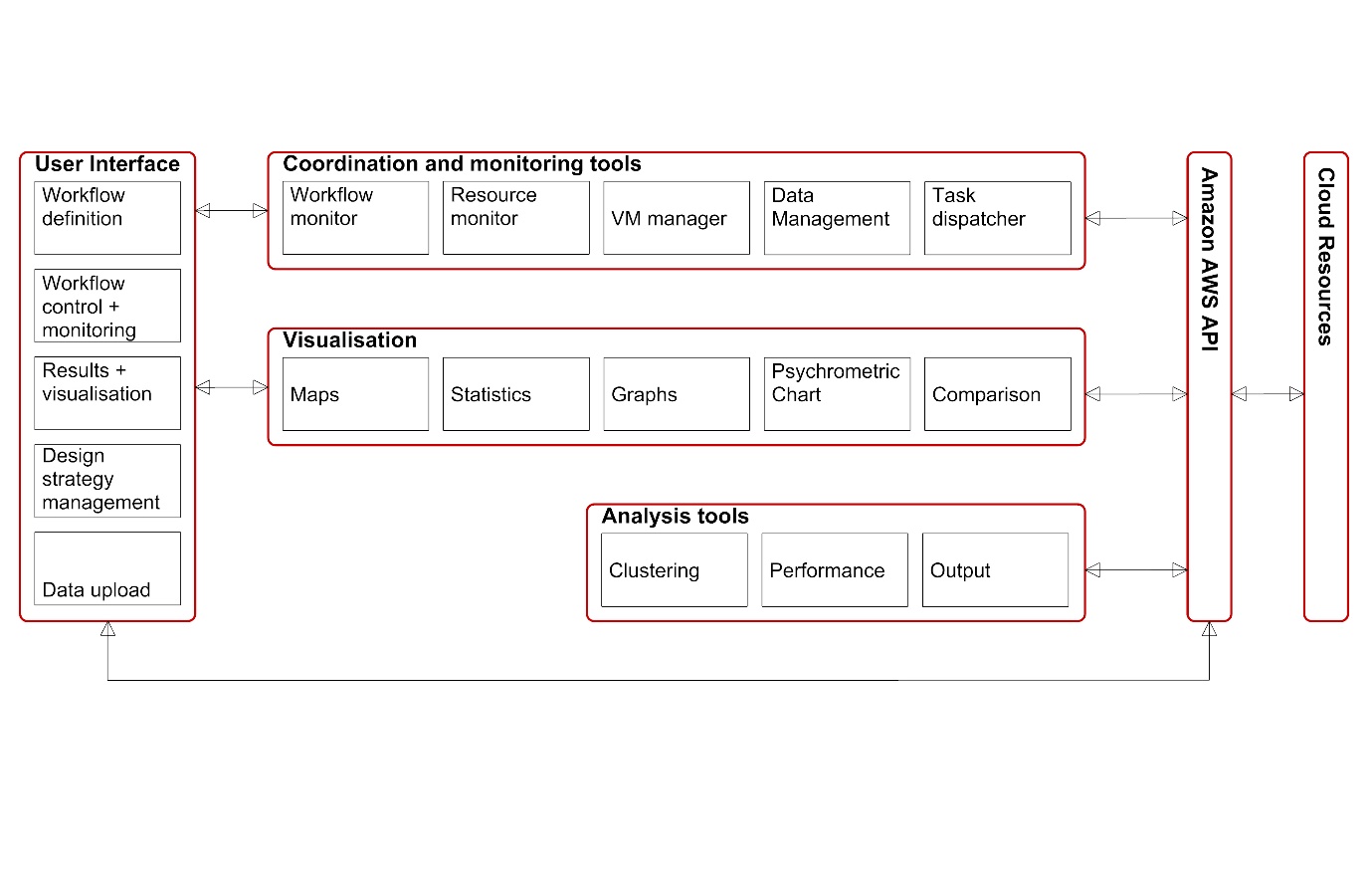


Figure 10 Architecture for the system.

Key architectural elements define the proposed system packages (Figure 11). The sub-package within the user interface package, Workflowbuilder, contains domain specific components for climate-driven building design and is regarded as an interchangeable element if the system was applied to another domain.

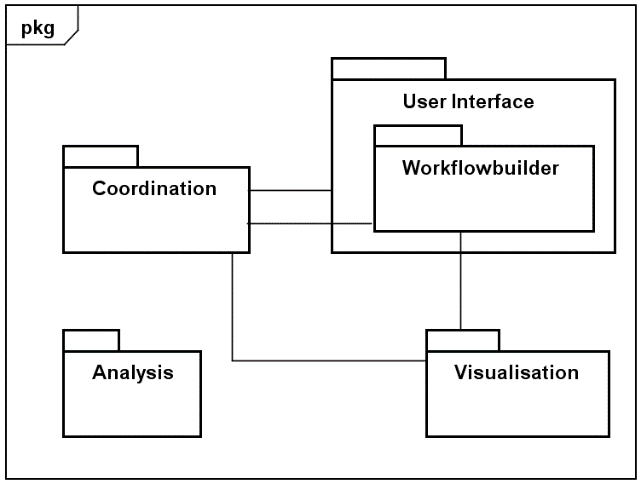


Figure 11 Simplified package diagram.

## Sequence Diagrams for General use cases

The following sequence diagrams (Figure 12, Figure 13, Figure 14, Figure 15) show how the use cases described in (ref to section) have been implemented in the final prototype using the classes and packages described in the previous section.

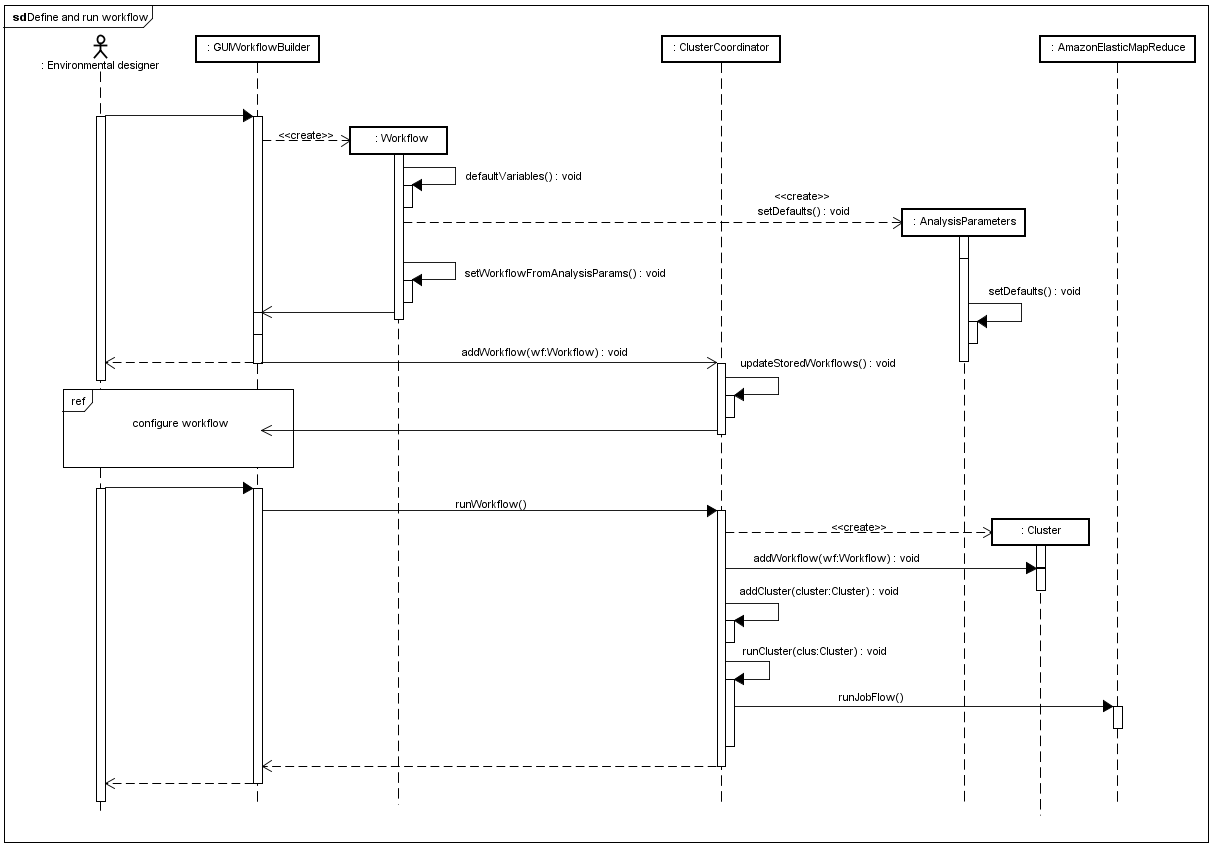


Figure 12 Sequence diagram for defining and running a workflow

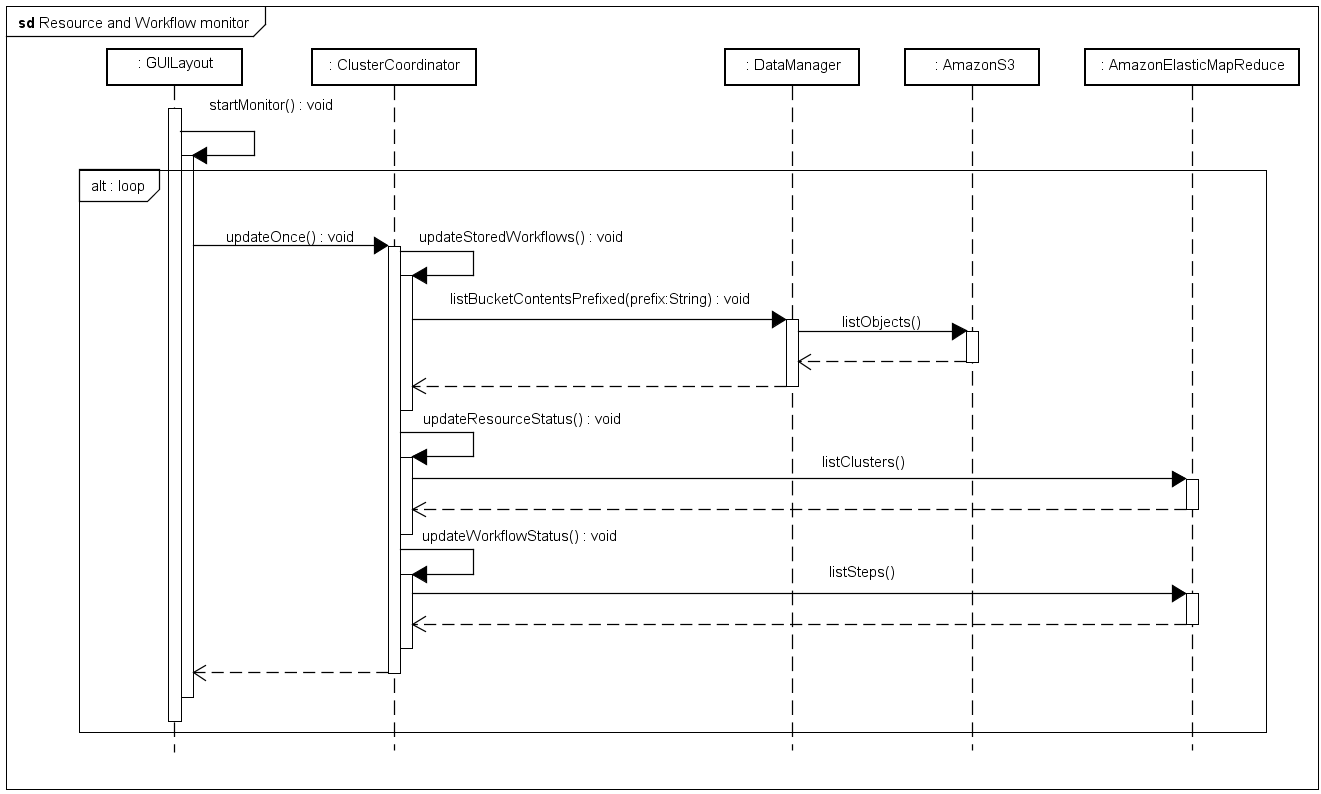


Figure 13 Sequence diagram for monitoring workflows and resources.

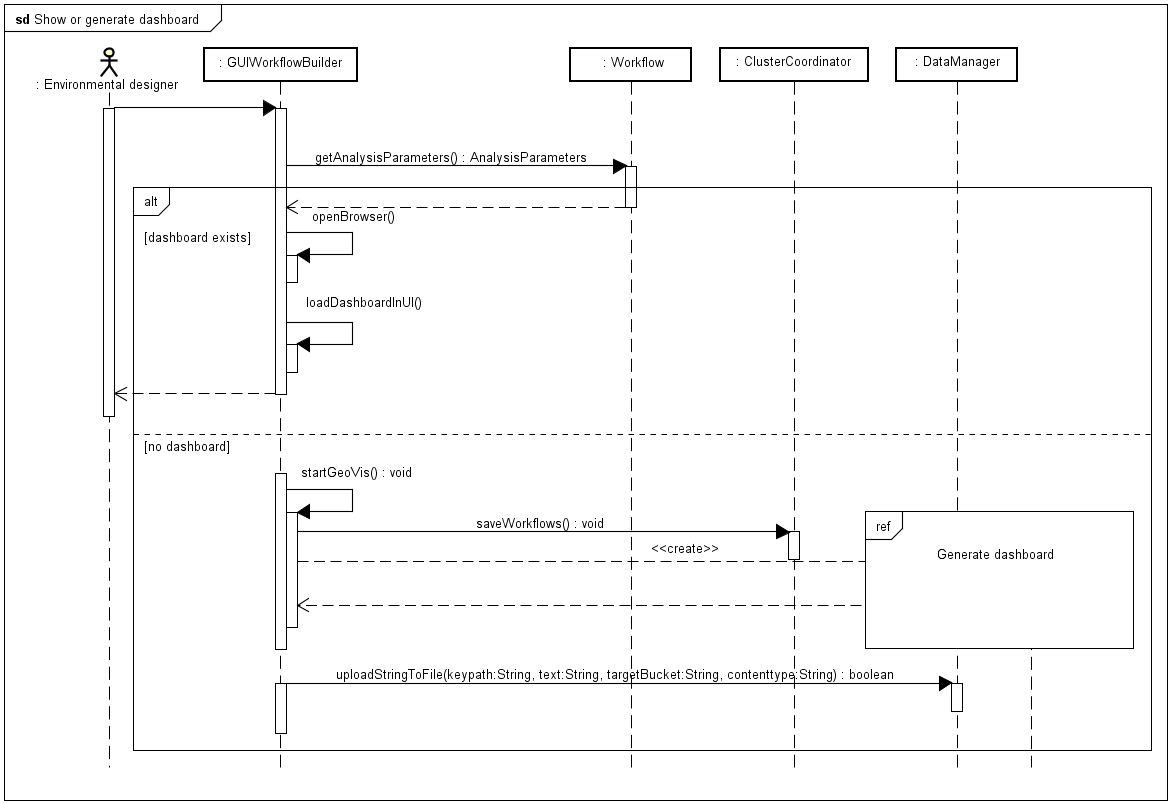


Figure 14 Sequence diagram for show or generate dashboard.

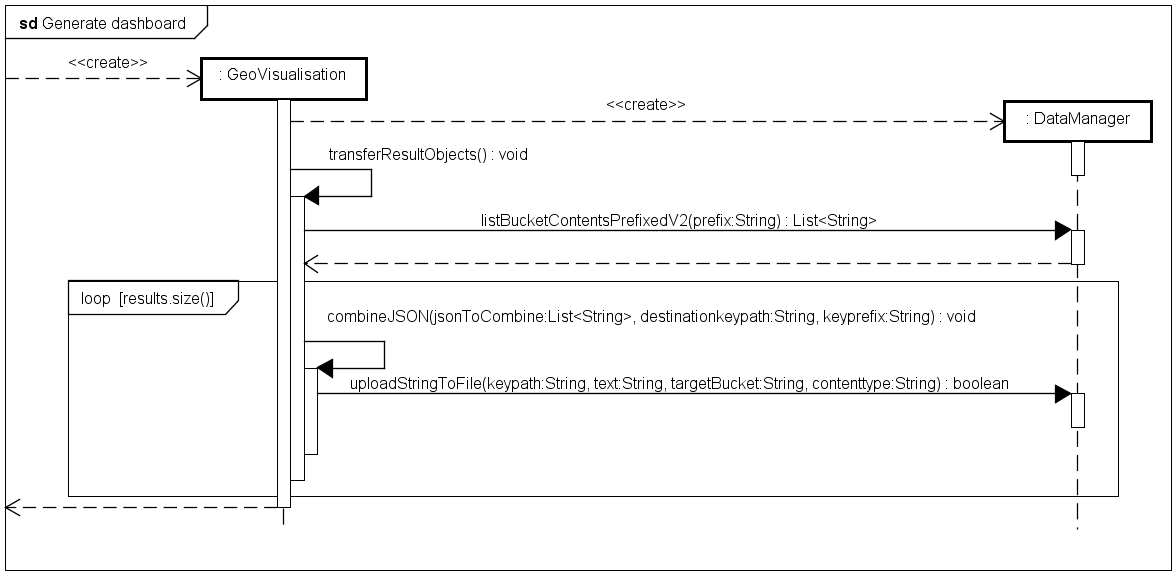


Figure 15 Sequence diagram for generate dashboard also see Figure 14.

## Design methodology

The design process was broken into a series of major development increments (see Figure 16). Each increment lasted approximately 25 days and followed Agile Model Driven Development (AMDD) with UML (Ambler, 2004). First, over several days, high-level modelling was used to understand the scope, requirements and potential architecture of the system. This was followed by a series of construction iterations, each of which began with a planning phase. Requirements were ranked by priority and the highest priority implemented first. Over several hours, UML models were produced to explore what should be built for the iteration and to estimate the time required. Issues identified in the planning models were then developed in more detail using just-in-time models, created in less than thirty minutes involving hand-sketched flow diagrams, sequence diagrams and class diagrams. Using these modelled details code was written during the following hours or days using a test-first and refactor approach.

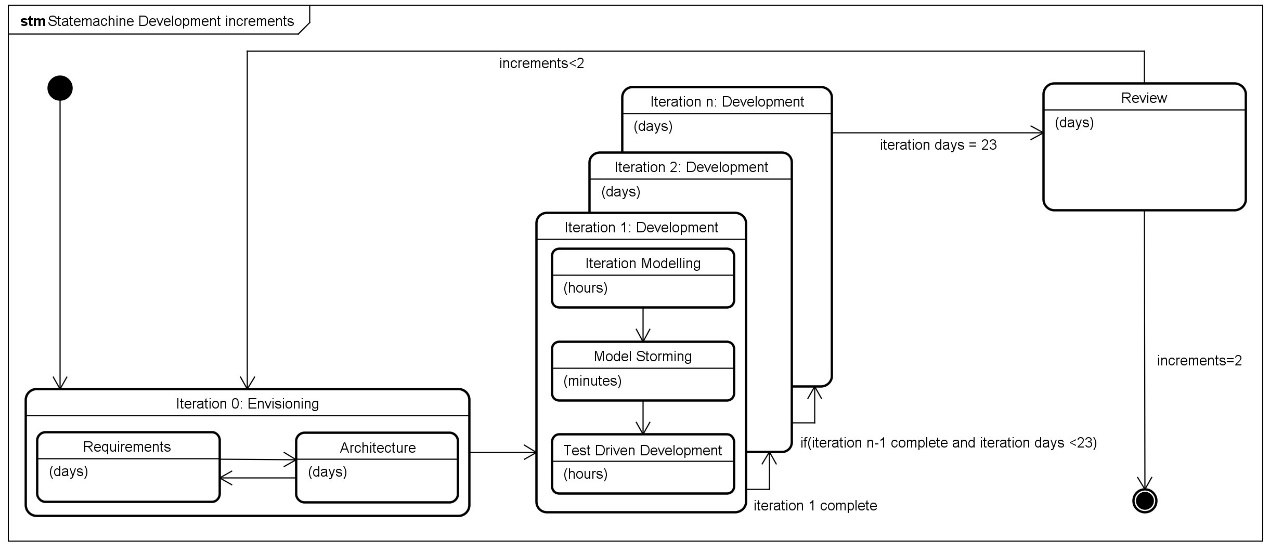


Figure 16 Development increments (extended from Ambler (2004), p119).

# Chapter 4. Implementation

## Development iterations

Implementation of the software prototype followed the AMDD methodology over five main iterations each including prioritising requirements, UML modelling, code development, integration testing and refactoring:

1. Key challenges across all packages were addressed, with the goal of a prototype that could perform all basic functionality. Functions were developed to use key aspects of the AWS API, reading and writing data in S3 and programmatically running clusters with simple Spark analysis routines. A rudimentary user interface was defined using Java Swing. This enabled predefined analysis routines and cluster configurations to be submitted to EMR, results written to S3 and visualised on an online map using HTML, JavaScript and D3.js.
2. Development focused on the Workflowbuilder package, monitoring of resources and workflow status within the Coordination Package and structuring the Analysis Package. UI development switched to JavaFX and the interface for user configuration of analysis workflows was developed and tested. An analysis pipeline was developed to provide parsing and filtering of the data set based on analysis parameters associated with workflows. Basic performance metrics were implemented with Apache Spark. Output from analysis was structured in preparation for developing a graphical visualisation framework.
3. Visualisation package was developed including routines to read, reformat and write output from analysis to a public server space. A dashboard framework was developed to read and display output data and provide user interaction.
4. Refactoring the analysis package to provide additional mechanisms to evaluate results.
5. The dashboard framework in the visualisation package was restructured with more code reuse, allowing multiple controllable instances of graphic tools simultaneously. Improving the graphical representation and user access to the evaluation metrics generated through analysis.

## Environment

The prototype operates locally on a Java Virtual Machine (JVM) analytics take place within AWS Cloud using the Elastic Map Reduce framework running spark-core\_2.30. Hardware on EMR is configured by the user via the local application as part of the workflow definition, users have access to all classes of AWS EMR instances and can run up to 20 instances (1 master and 19 slaves).

The system was developed and tested on Windows 10 Pro running on 64-bit Operating System with Intel Core i7-8700k CPU and 16Gb ram. Java development took place using Eclipse photon IDE with AWS Toolkit for Eclipse (*AWS Toolkit for Eclipse*, no date). UML models were developed using Astah Community with some additional diagrams modelled in Rhino3d. JavaScript, html and CSS used to define the dashboard web framework was developed using Sublime Text 3 and tested with http-server (*http-server: a command-line http server*, 2018) to debug locally within Chrome 69. Github.com was used for version control and the latest version of code and documentation is available: <https://github.com/rolyhudson/climacolombia.git>.

## Architecture

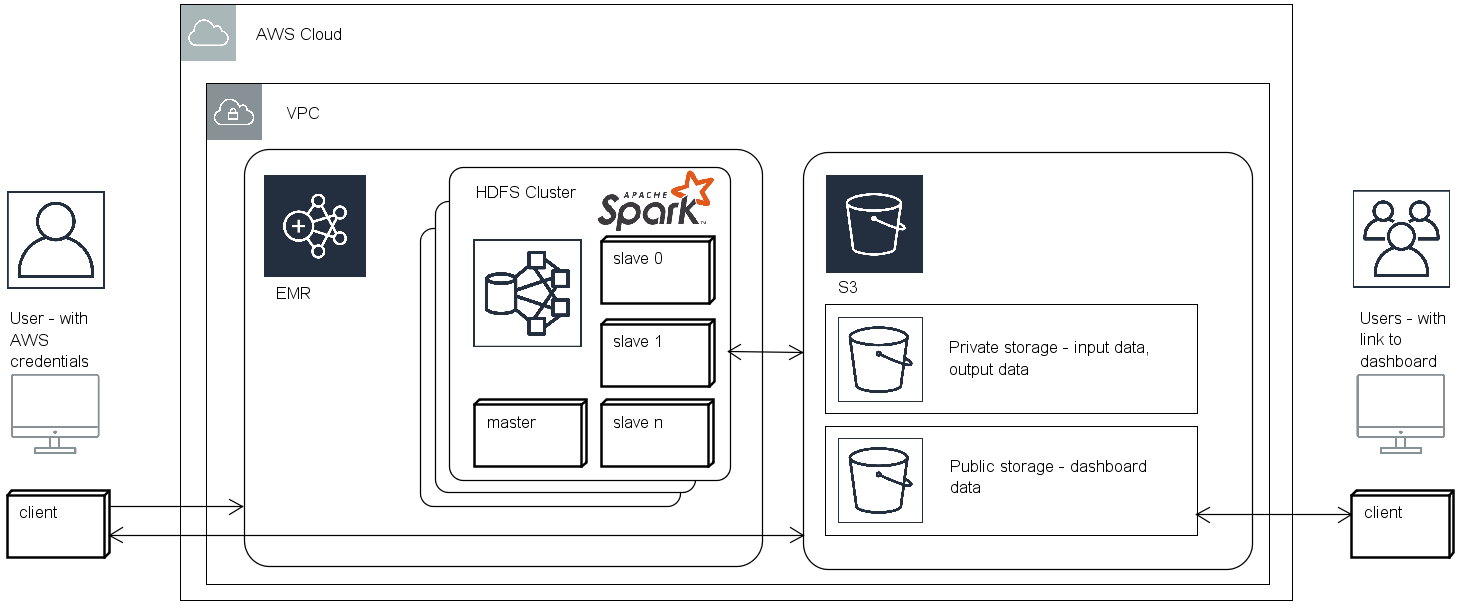


Figure 17 Deployment on AWS

The application uses IaaS on Amazon Web Services (AWS) cloud infrastructure. Users access the local application (with AWS login credentials) and can launch resources within Amazon’s Virtual Private Cloud (VPC). A graphical interface enables users to create workflows (analytic jobs) and configure the hardware that will be used to compute them. These analytic jobs are launched as Hadoop distributed file system (HDFS) clusters on Amazons’ Elastic Map Reduce (EMR) each running Apache Spark, multiple clusters be can be configured and launched simultaneously. Clusters have access to private buckets within Amazon’s Simple Storage Service (S3), where they access input data and store results from the analytic jobs. When analytics on a cluster completes, output files are transformed and stored in a publicly accessible bucket with a web framework that defines a graphical dashboard for visualising results. A Uniform Resource Locator ([URL](https://en.wikipedia.org/wiki/URL)) is created for the dashboard that can be shared by the original user so others can access the visualisation and results.

## Climate data

The data set is based on ten years of multivariate, historical, monthly averages of gridded climate data from three sources. First, climatic data from 1901-2009 formatted as ESRI ASCII raster by CGIAR CSI (Cgiar-csi.org., 2012) based on original data from CRU (Jones and Harris, 2008). Secondly, wind speeds from the CCMP gridded surface vector winds (Wentz *et al.*, 2015). Thirdly, elevation data is extracted from a hole-filled DEM of SRTM (Jarvis, A., H.I. Reuter, A. Nelson, 2008). Figure 13 illustrates the data preparation steps. A C# dot net program was written that takes a topojson (Bostock, 2017) format file as input, this describes the boundary (or collection of boundaries) that define the zone of interest. A point grid is generated at half degree latitude and longitude intervals filling the area(s) of study. Cross-referencing the grid to the DEM determines altitudes for each point.

The grid is used to extract climate data from a ten-year period (2000-2009) from the CRU and CCMP datasets. CRU data is formatted as ASCII ESRI raster format at the same resolution as the grid. CCMP data is in netCDF format, this was pre-processed with an independent Java program, written using the Unidata (2012) netCFD Java library. The netCDF data was converted to the ASCII ESRI raster format. Each ASCII raster file represents a single month of a year and contains data for earth’s surface. Data points in the raster files that coincide with our grid vertices are found and stored with the georeferenced grid in arrays. Relative humidity derived using the ratio between vapour pressure and saturation pressure. The data is averaged for the 2000-2009 period providing a typical year with monthly values. The prepared climate data can be saved to a Comma Separated Value (csv) file.

Version control: <https://github.com/rolyhudson/griddedClimateDataProcessing.git>

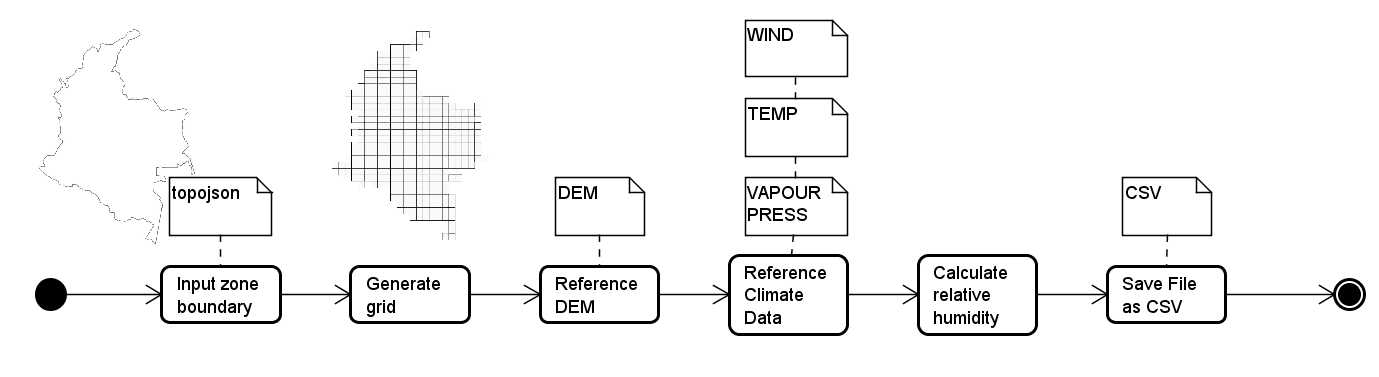


Figure 18 Flow diagram showing the stages of the data preparation process.

## Packages

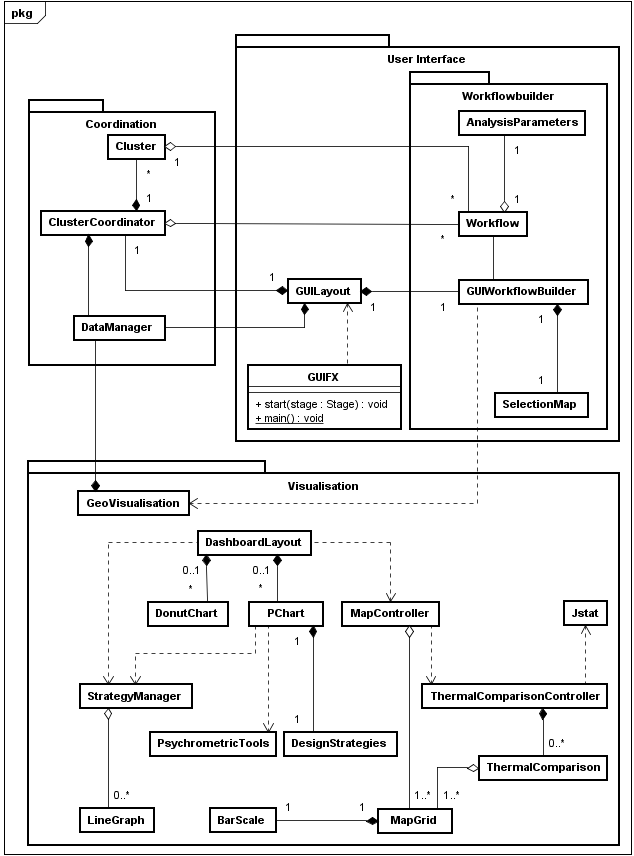


Figure 19 Coordination, User Interface and Visualisation packages overview

The system is implemented with four key packages and one sub-package (Figure 17). The User Interface package provides functionality that presents the status of cloud resources and running analysis jobs. New analysis routines and data sets can be uploaded from the UI as .jar packages. The UI also provides access to dashboard of results via an embedded web engine and representation (and a mocked-up editor) of the design strategies editor.

The main component in the User Interface package is a sub-package, Workflowbuilder, which concerns all classes involved with the definition and of a workflow which include defining spatial zones for analysis and specifying a series of parameters concerning the data, analysis method and temporal scales. The coordination package interfaces with the EMR client via the ClusterCoordinator class and S3 via the DataManager class both of which use locally stored credentials and the AWSCredentialManager for secure access to AWS services.

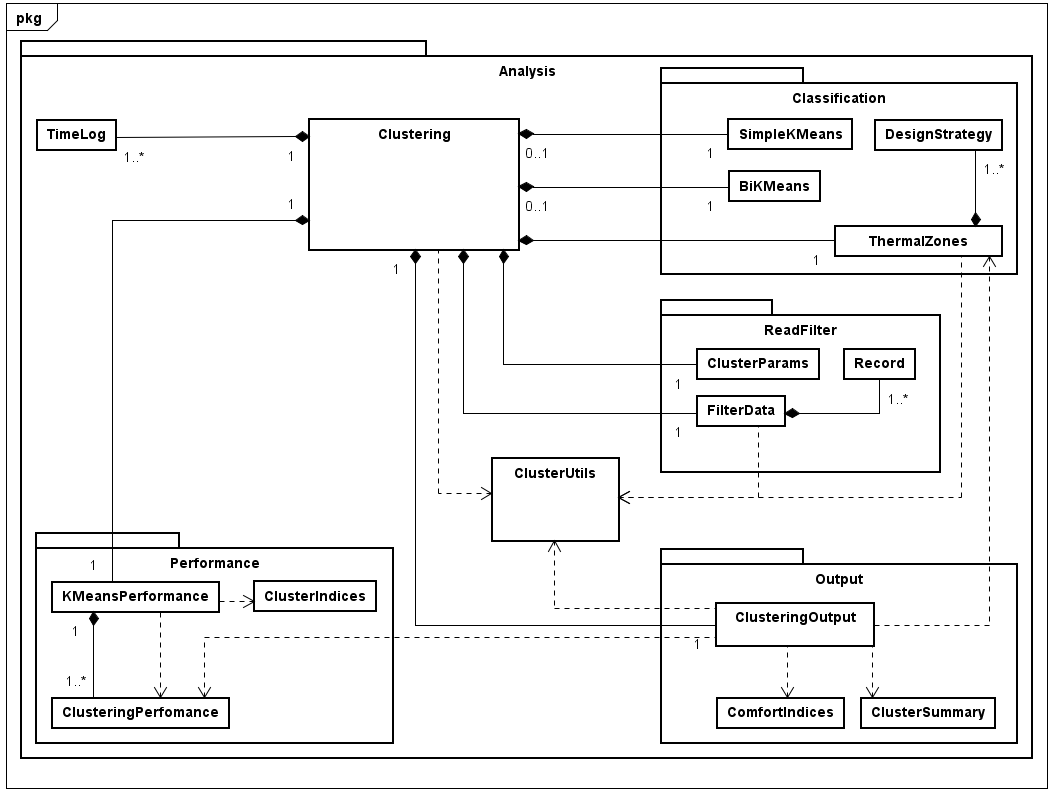


Figure 20 Analysis package overview

The Analysis package is independent of other packages and defines the analytical procedure used by Apache Spark running on AWS EMR. It includes four sub-packages each handling different aspects of the analytic routine. Analysis parameters defined by the User Interface are parsed and filtered by classes within the Read Filter sub-package. The Performance sub-package computes the performance of clustering solution using a series of standard indices, additionally the number of clusters can be optimised base on one of these indices. Classification sub-package implements two clustering methods from Spark’s Machine Learning library MLLib (*MLlib: Main Guide - Spark 2.3.2 Documentation*, no date) K-Means clustering and Bisecting K-means clustering and a third method that is a hybrid of the previous two. Data vectors are submitted to the Classification sub-package, clustering models are constructed, and the data is classified and assigned applicable design strategies. Finally, the Output sub-package organises the analytics results and the meta-data related to the performance of the process and writes these in a logical storage structure to S3.

Once analysis completes the GeoVisualisation class in the Visualisation package is used to post- process analytic output and move files to public buckets on S3 with a unique URL that is provided to the user. A default homepage is created which references the dashboard framework components also hosted within a public area of S3 providing an interactive, online interface for visualising the results.

## Workflow management system

### Workflowbuilder and Coordination package class diagrams

At the core of the application is the workflow management system consisting of classes defined with the Workflowbuilder package (Figure 20 and Figure 21) which communicate with the classes in the Coordination Package (Figure 22).

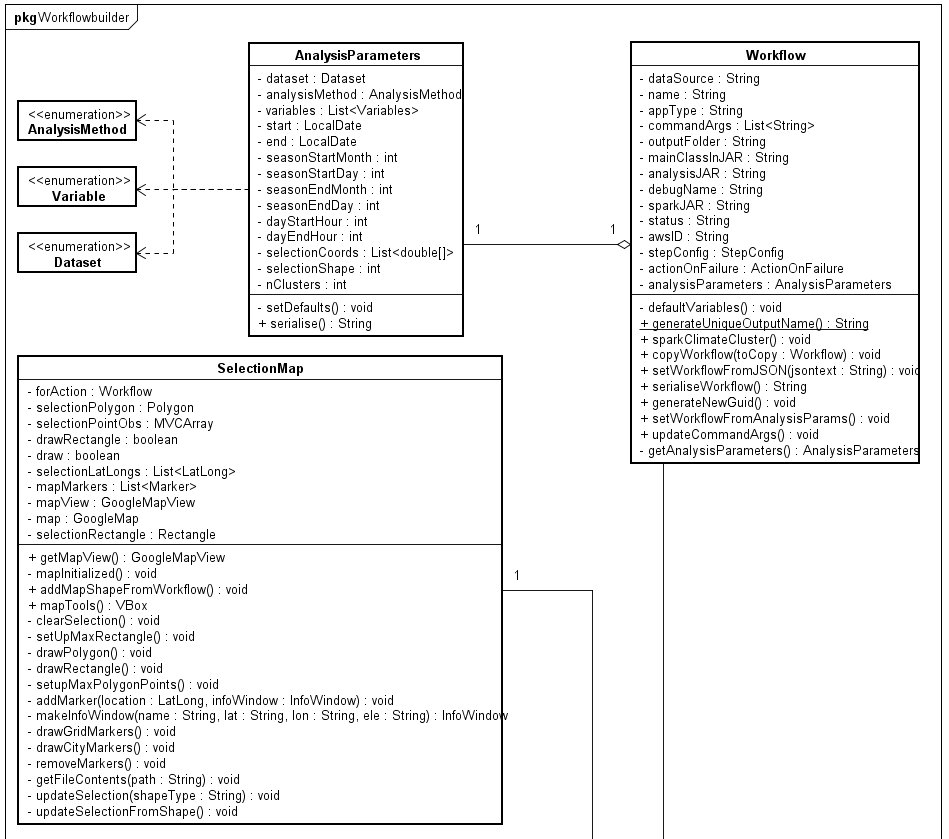


Figure 21 Workflowbuilder package class diagrams a



Figure 22 Workflowbuilder package class diagrams b

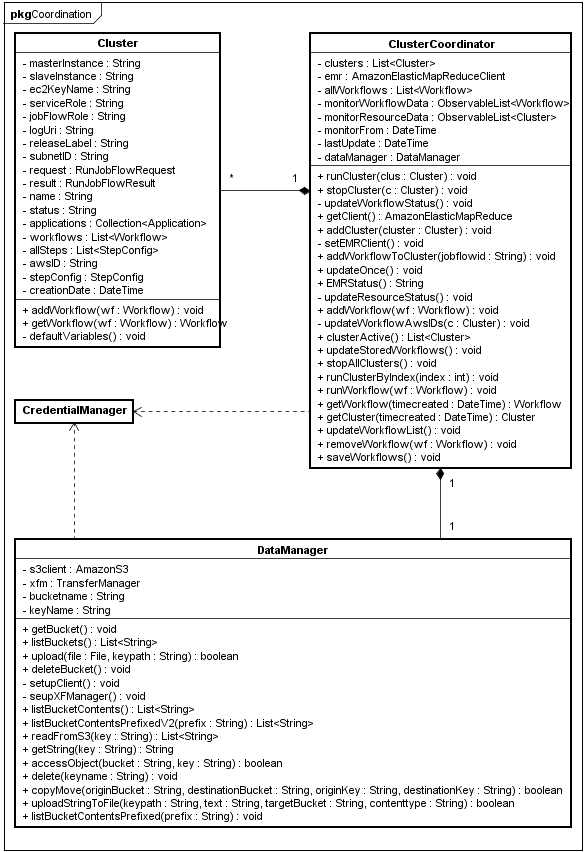


Figure 23 Coordination package class diagrams

### Workflow creation

The Workflow class manages all the parameters for an AWS Step, additionally, the Workflow class includes a single instance of the AnalysisParameters as the attribute analysisParameters. The AnalysisParameters is a description of the domain specific configuration of the workflow. The user can configure this class instance via the GUIWorkflowbuilder (Figure 23) class which provides access to hardware configuration options, choice of data set, selection of variables for clustering, clustering method, number of clusters and control of the temporal range of the analysis. Spatial specification of the analytics is via the map an instance of the SelectionMap class that implements GoogleMaps in JavaFX using GMapsFx (Terpilowski, no date). The map shows the selected data set, locations as red points (grid-based or cities) and a selection polygon or rectangle can be drawn and edited to define the region for analysis.

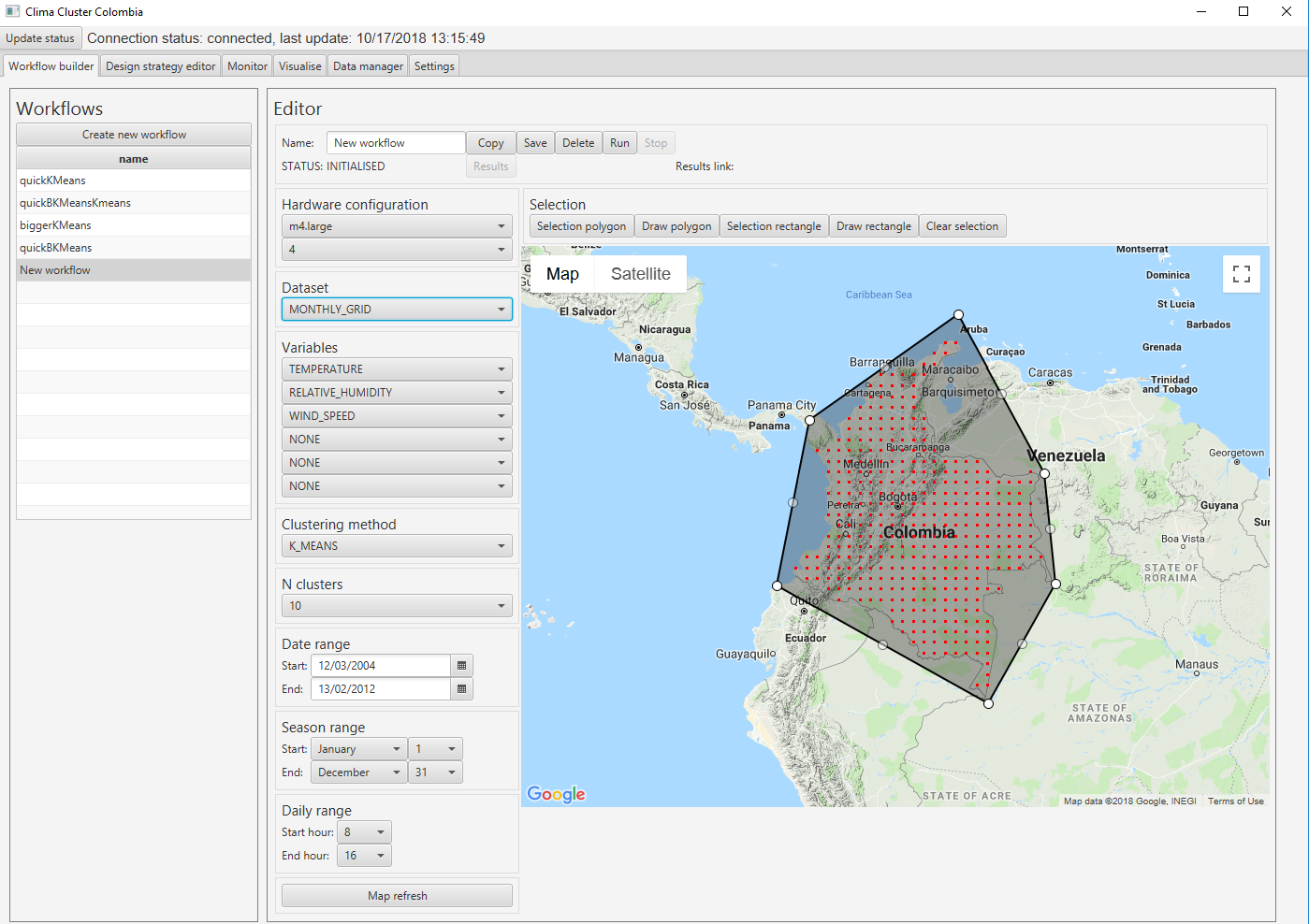


Figure 24 GUI for Workflowbuilder

User edits to the workflow via the UI are stored in a list of workflows maintained by the ClusterCoordinator class. Multiple workflows can be created and saved in one session. On exiting the application session all workflows are stored in S3 in a private bucket in JSON format. Each workflow instance has a GUID which determines the S3 object name, the object contains a string in JSON format that is returned by the class’ serialise method. Workflow objects are pushed to S3 by the UI’s instance of the DataManager class datamanager.

### Monitoring resources and workflows

The ClusterCoordinator maintains a list of all EMR clusters created after a datetime parameter – monitorFrom (set by the user). Each EMR cluster is represented locally by an instance of the Cluster class. This manages all the parameters required to create a new EMR cluster (Figure 22). While the application is running the ClusterCoordinator is constantly monitoring the AWS resources (see Figure 13 Sequence diagram for monitoring workflows and resources.) The goal of the monitoring process is to keep the ClusterCoordinator’s locally stored lists of clusters and workflows up to date with the status of the EMR Clusters and associated steps found on the cloud. The implemented monitoring algorithm (Figure 24) allows users to break between work sessions and shut down the application, on returning the status of any running clusters or workflows is reported and the user can continue to edit, copy or execute a workflow created in a previous session.

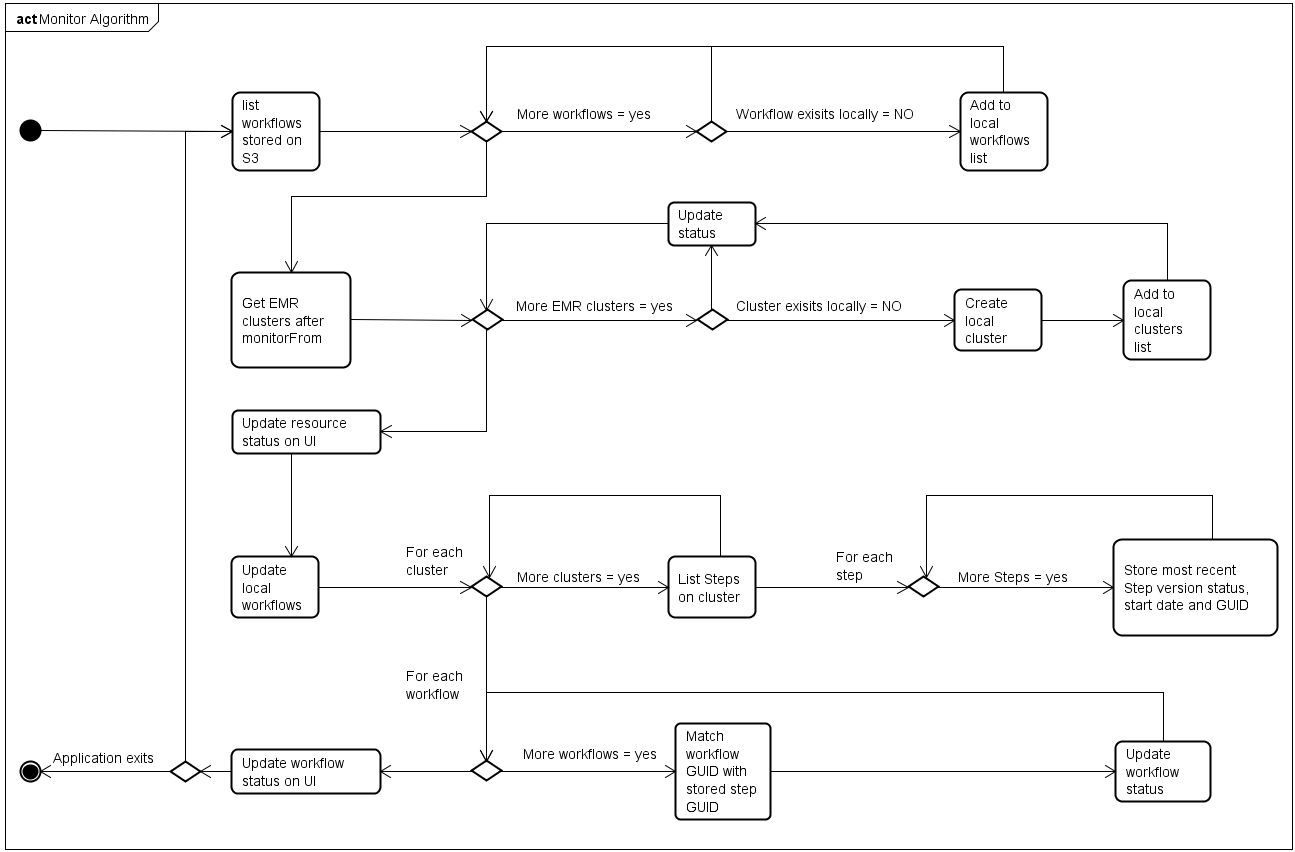


Figure 25 Resource and workflow monitoring algorithm

Resource monitoring takes place on a separate thread to the UI and is executed every minute to avoid AWS’s throttling limits. Key to this process is the workflow class GUID and the method, setWorkflowFromJSON, which creates a new workflow by de-serialising the JSON string stored in the S3 object. Once locally stored lists are updated they are added to an ObservableList that is used to populate the UI’s monitor tab (Figure 25). To work with a workflow, it must be selected from the left-hand panel in the Workflow builder tab (Figure 23). The status of a workflow is show in the UI, its status or state (Figure 26) determines the availability of controls on the UI. For example, a COMPLETED workflow only allows the user to copy, delete or generate the dashboard Figure 27. Whereas an INITALISED work flow provides access to all workflow configuration controls but not the buttons to stop or generate the dashboard. The activity diagram in Figure 28 shows the different processing paths a user can follow when working with a workflow.

### Running a workflow

Once the user has prepared their workflow they click the run button on the UI and the workflow is sent to EMR by the ClusterCoordinator. The workflow can either be added to an existing cluster or a new cluster can be created if none exist or the user requires different hardware configuration. The activity diagram in Figure 29 illustrates the steps involved in running a workflow.

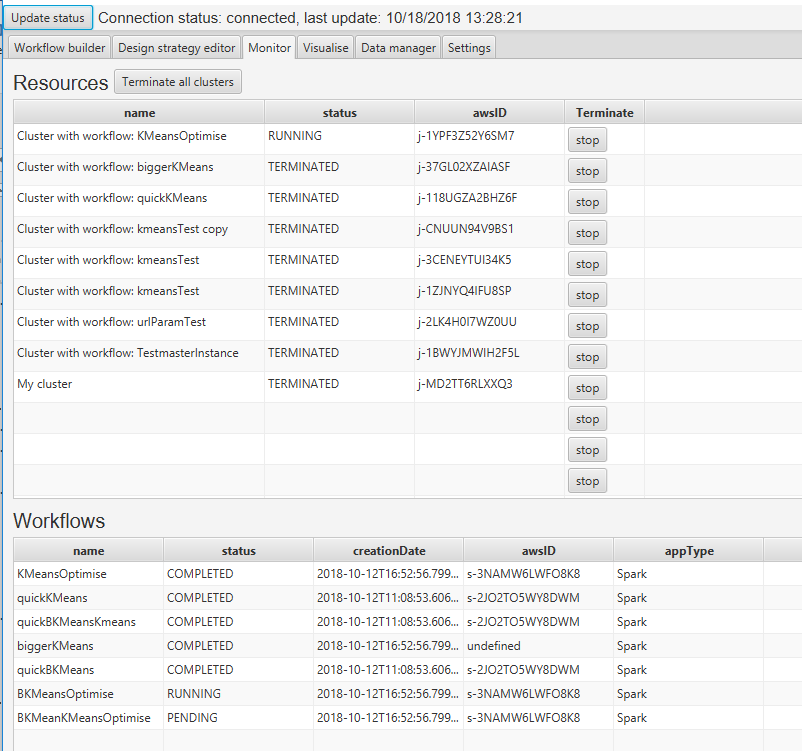


Figure 26 UI Monitor resources and workflows tab

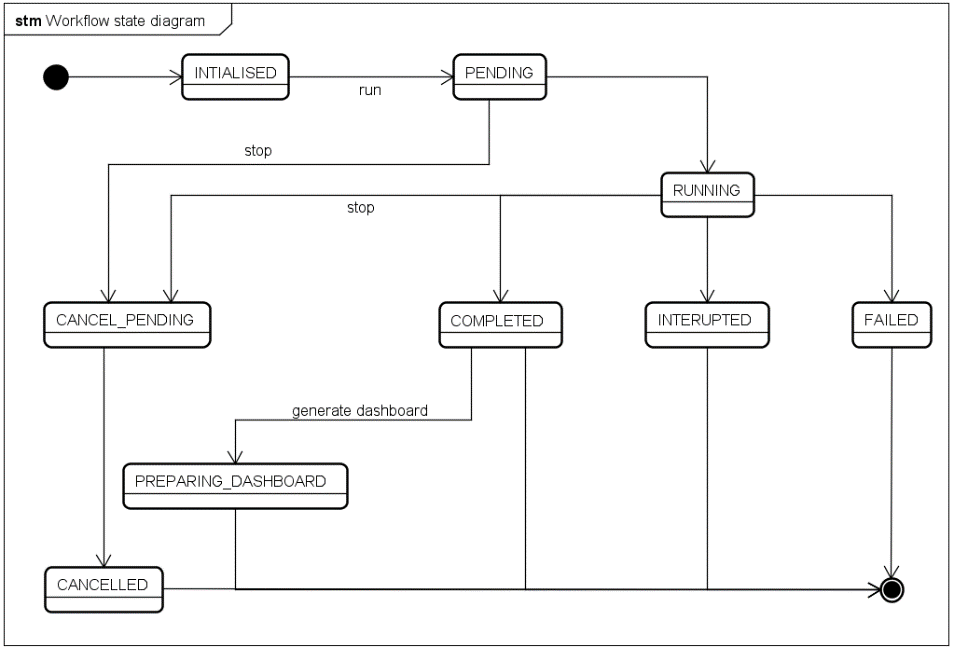


Figure 27 Workflow state machine

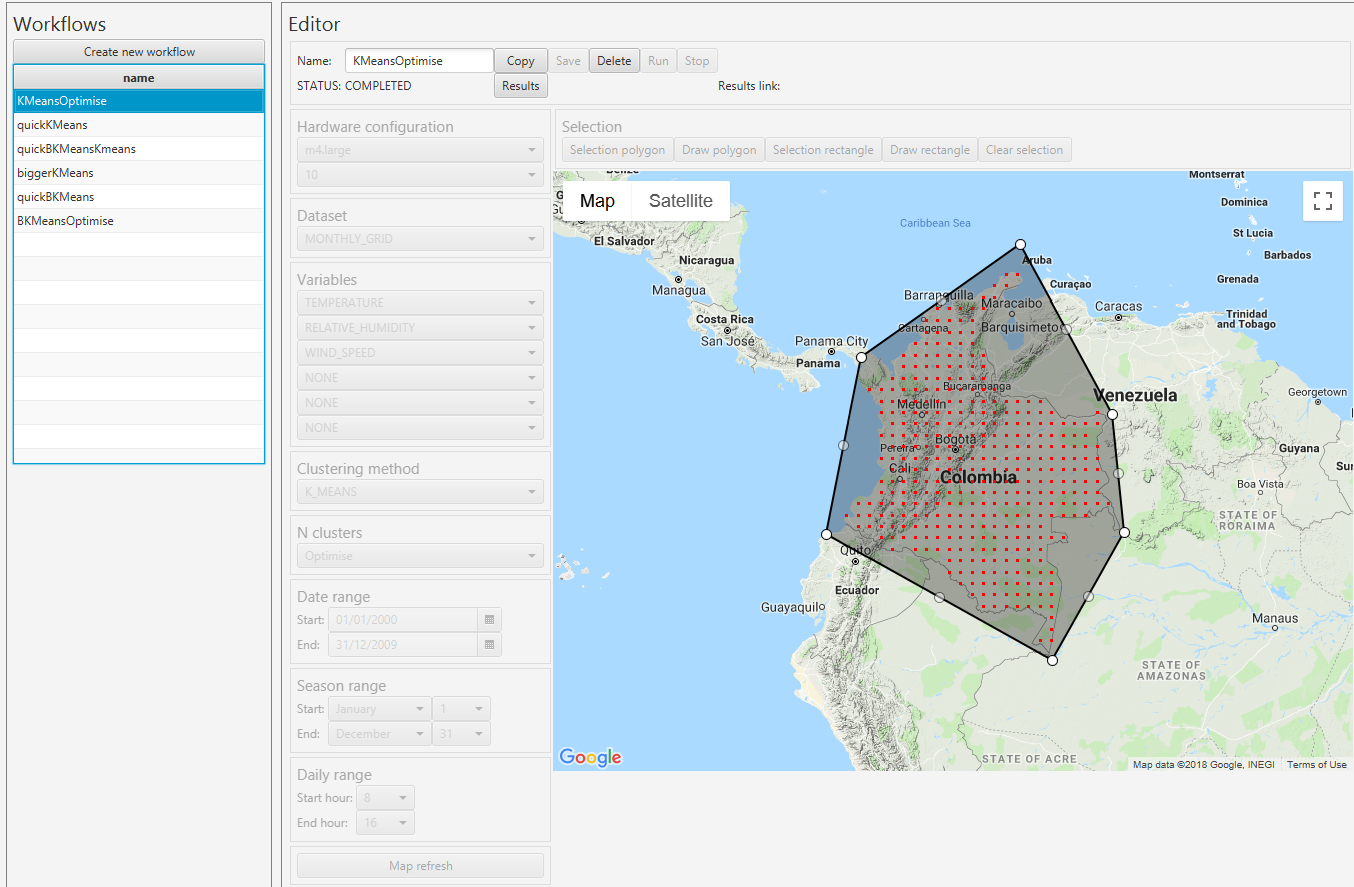


Figure 28 Workflow with status COMPLETED

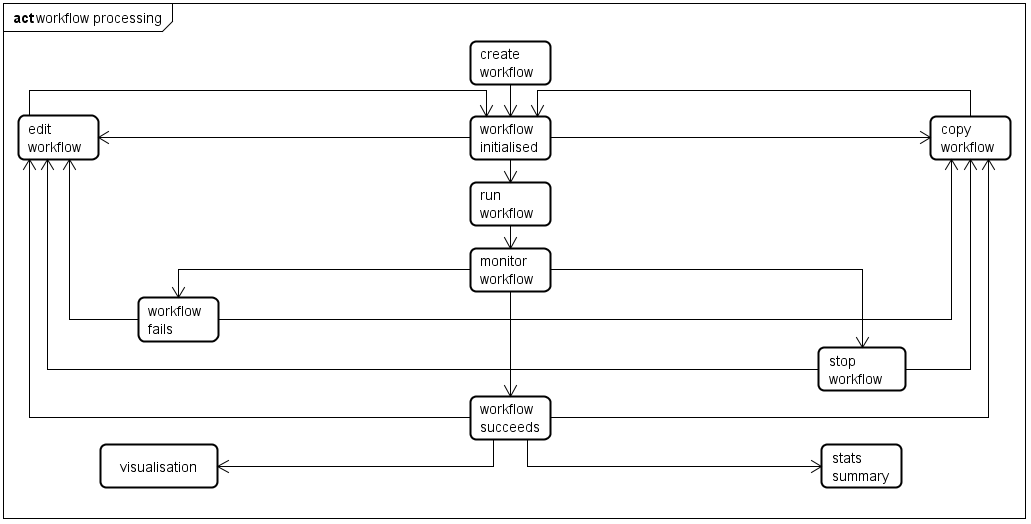


Figure 29 Processing workflows activity diagram

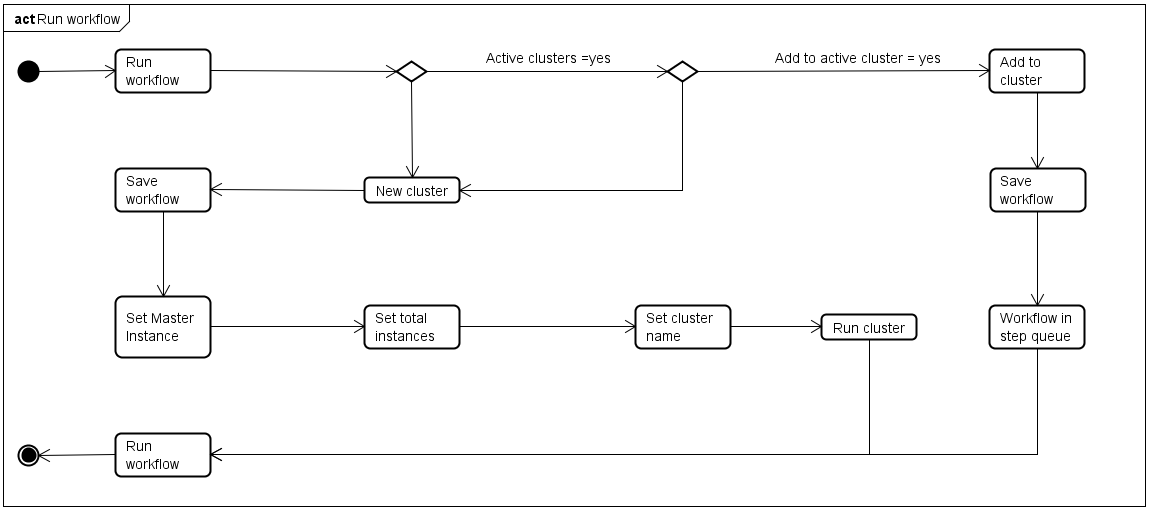


Figure 30 Run workflow activity diagram

### Management of design strategies

The management of design strategies has not been fully developed for the prototype. The UI includes a tab for editing strategies, however, this is a mock-up of a simple drawing interface that would allow users to draw and edit strategies over a psychrometric chart (Figure 30). The intention is that each strategy is represented as closed polygon with a name, an array of these polygons can be stored, one per line, in a simple comma separated text file where the first field is the name followed by x and y coordinates of the vertices. For the purposes of the prototype the strategies described by Manzano-Agugliaro *et al.*, (2015) have been stored in this format and are used for the analytic process described in the following section

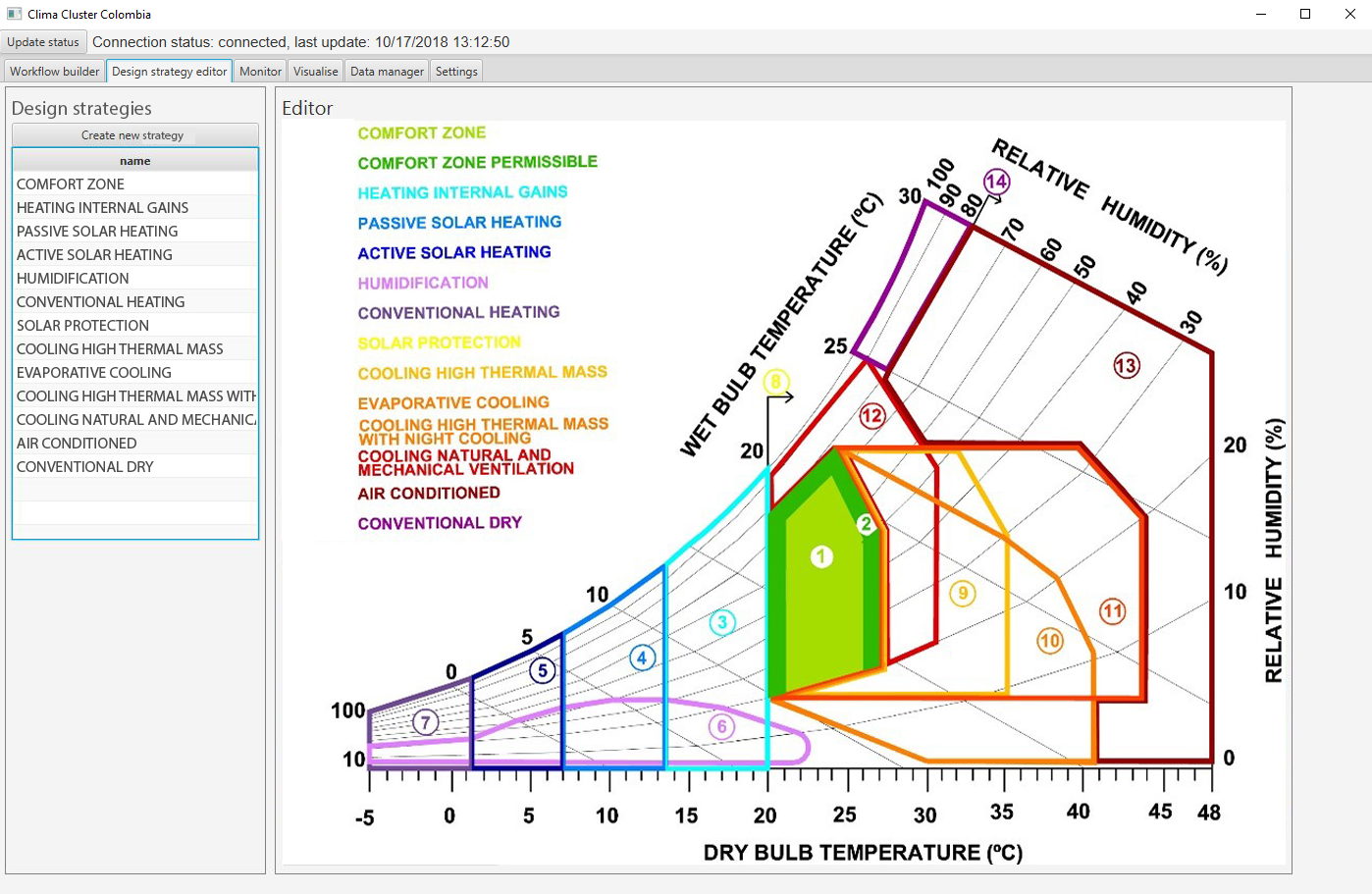


Figure 31 Design strategy editor mock-up.

## Analytics

The analytic pipeline is a Spark application defined as an independent Java project which is exported locally as a .jar file and uploaded to S3 via the UI with the keypath “sparkJAR”. When a workflow is submitted to EMR it defines parameters for the configuration of an EMR Step. These parameters include its name, the action that the cluster takes on failure, the name of the analysis jar and a set of arguments. The analysis jar parameter refers to Amazon’s command-runner.jar that enables spark-submit script. The spark-submit script is defined in the arguments that specify:

1. the deployment mode
2. the class in the application that contains the main method
3. the location of the spark application jar
4. Data source
5. Output folder
6. Location of the workflow file on S3
7. Location of the design strategy file on S3

These seven arguments can be seen in the EMR management console within the cluster details (Figure 31). Once the EMR cluster is running arguments 4-7 are passed to the class in the application that contains the main method which takes an array of Strings as its only argument. The Clustering class (Figure 32) includes the main method and acts as the controller for the analytics process, it initialises the SparkSession and parses the clustering parameters, reads and filters the data set and the design strategies. Depending on the selected clustering method, performance is evaluated before the selected data is classified and design strategies appropriate to each cluster, data-point are identified and summarised. Finally, the results are written to S3. Figure 20 provides an overview of the Analysis package and its class structure, Figure 33 illustrates the sequence of processes used to implement the analysis.

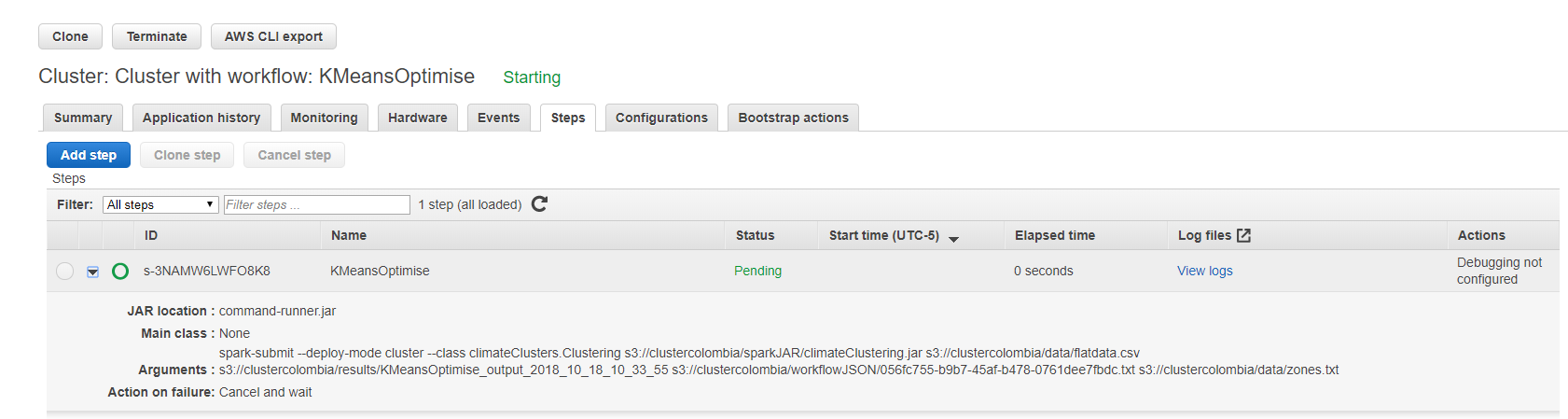


Figure 32 Parameters for a Step on EMR

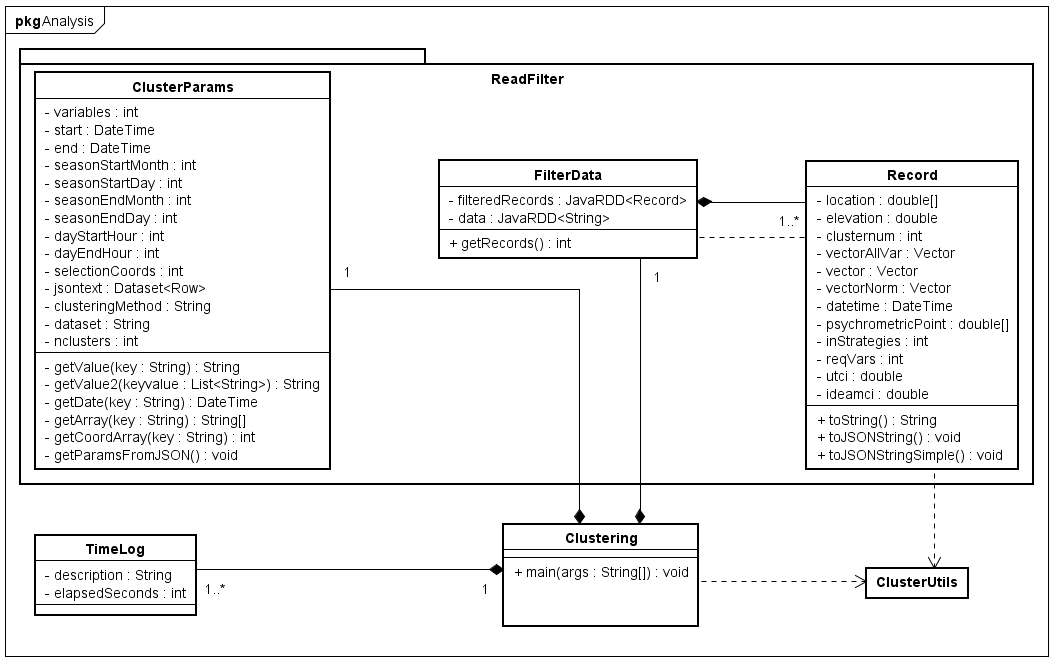


Figure 33 Analysis package and ReadFilter sub-package.

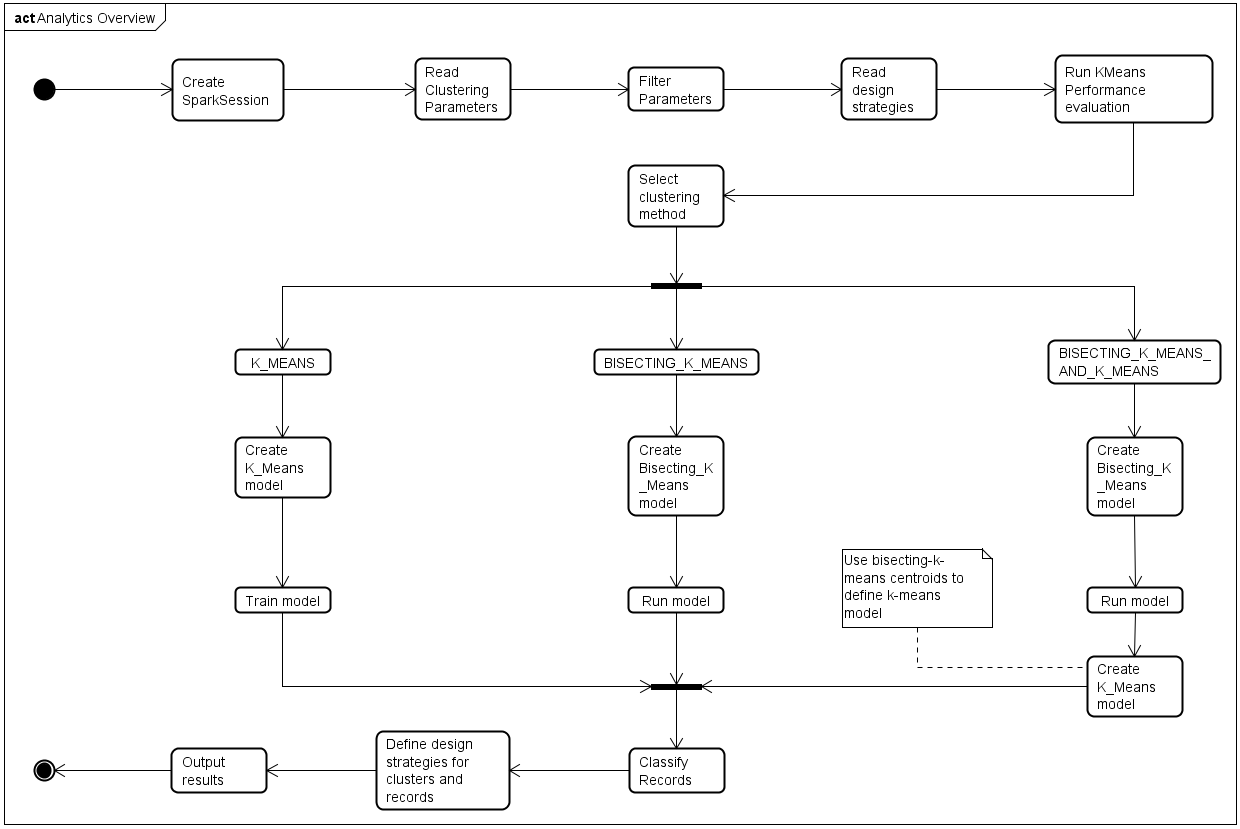


Figure 34 Analytics overview activity diagram

### Read Clustering Parameters and Filter Data

Deserialization of the Workflow class and its analysisParameters attribute is undertaken by the ClusterParams class (Figure 33) by reading from a string in JSON format stored on S3. The JSON file is read directly to a Dataset<Row> and from that the clustering method, number of clusters, temporal and spatial configuration of the workflow can be selected.

The FilterData class (Figure 33) takes the parsed clustering parameters, reads the input file containing all the climate data in csv format. Using a combination of the RDD operations filter and map and the helper methods defined in the ClusterUtils class a subset of the original data is defined as a JavaRDD of Record objects. This dataset contains only data points within the required spatial and temporal ranges.

Each instance of the Record class (Figure 33) includes attributes to stores all the variables contained on each line of the required input data and a list of variables that will be used for clustering. In addition, the Record object stores a psychrometricPoint, a 2d array defined by the temperature and relative humidity, a list of design strategies that can be applied to the Record, the comfort indices UTCI and IDEAMCI for the record. The Record class includes a method that returns a normalised Vector of the variables that is used for the clustering processes. Design strategies stored as a text file on S3 are read and parsed by the ThermalZones class (Figure 37) and stored as a list.

### Clustering Performance Indices

The KMeansPerformance (Figure 35) class is responsible for defining performance indices for the selected clustering method. A workflow can be configured with a user-defined number of clusters or set to optimise the number of clusters. Performance metrics for each solution are stored in a list of ClusteringPerformance objects (Figure 35) in the KMeansPerformance class. The list either contains results for a single clustering solution or if optimisation has been selected it stores clustering indices for 58 solutions with two to sixty clusters. ClusteringPerformance objects store cost as WSSSE plus Silhouette and Dunn indices. These metrics are calculated by the ClusterIndices class (Figure 35), which includes different methods for each index depending on the selected clustering method.

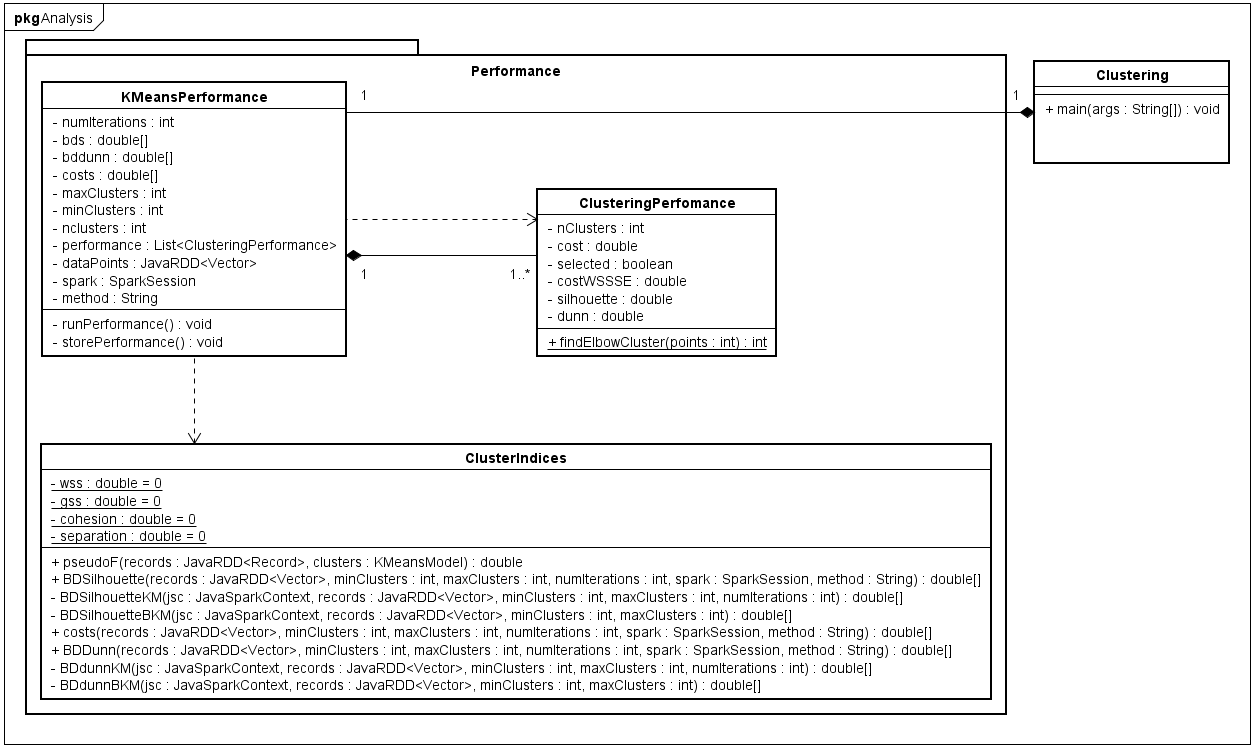


Figure 35 Analysis package and Performance sub-package

Optimisation is based on the WSSSE for each solution and the optimal number of clusters is determined computationally using the elbow method implemented as the static method findElbowCluster in the ClusteringPerformance class. The method uses a geometric algorithm that creates the performance graph for all solutions and finds the point on the curve (clustering solution) furthest from a line between the first and last solution. Figure 36 shows a scenario where twelve clusters is found to be the most optimal solution with a WSSSE of 3.7.



Figure 36 Elbow method

### Clustering classification and design strategy assignment

Once performance metrics have been generated, and the optimal number of cluster defined the analytics proceeds to defining the chosen clustering model. This is based on two classes SimpleKMeans and BiKmeans (Figure 37) which are used independently or combined to provide a third, hybrid method. Each Record is classified (assigned a cluster number) using the selected clustering model and the ThermalZones class tests the Record’s psychrometricPoint to find applicable design strategies and assigns these to the Record. The DesignStrategy class (Figure 37) stores the boundary of a strategy as a list of vertex coordinates defined in terms of temperature and relative humidity. To test if a design strategy is applicable to a Record its psychrometicPoint (a 2d point also defined by temperature and relative humidity) can be containment tested against the strategies boundary vertices, if the point is inside the polygon the DesignStrategy is added to the Records list.

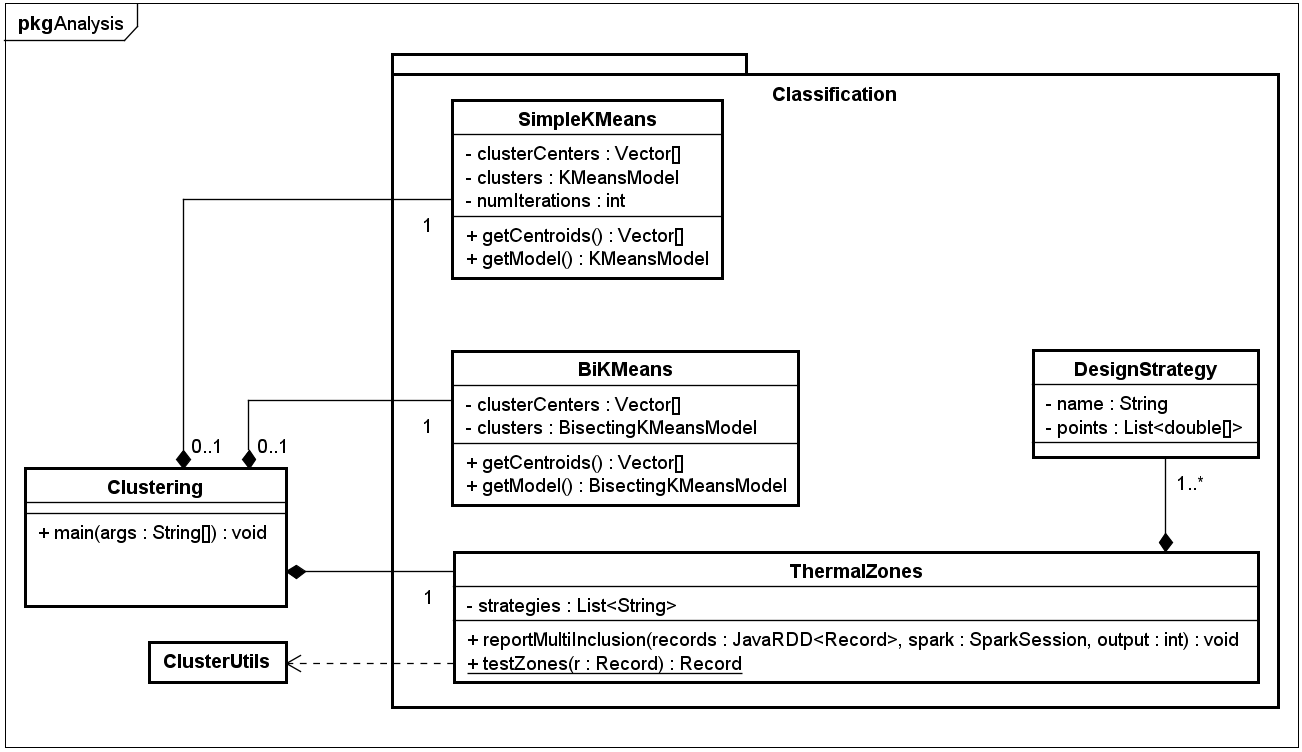


Figure 37 Analysis package and Classification sub-package

### Output of results

In anticipation of querying and quickly visualising the results according to different spatial and temporal ranges Spark is used to pre-process and structure the output. Meta-data related to the clustering performance and configuration of each cluster is stored. Three key spatiotemporal queries are anticipated. The first, averaging all temporal results to provide a single dataset with no temporal dimensions creating a single typical period for the original temporal scope across the entire spatial range. The next query type creates twelve datasets each representing a typical month, creating a single typical period but with monthly granularity across the entire spatial range. The third anticipated query type provides the finest granularity generating datasets for each year and month and summary information per individual cluster. Lastly to accommodate users that wish to create custom queries or create their own visualisation processes the full set of results is stored as a single document.

#### Meta-data

Performance metrics stored as a list of ClusteringPerformance objects in the KMeansPerformance class are converted to a Dataset<Row> and then written to a file in JSON format. Using the getComfortIndicesClusters method in the ComfortIndices class (Figure 39), average temperature, relative humidity, thermal indices are defined for each cluster. The comfort indices and the centroids of each cluster are passed to the reportClusterSummary method in the ClusterSummary class (Figure 39). A list of ClusterSummary objects, one for each cluster in the solution, is generated describing centroids, the number of data points contained, thermal indices, average temperature and average relative humidity. The ClusterSummary object also contains design strategies associated with the cluster. Strategies are determined by using the cluster’s temperature and relative humidity to define a single point and containment testing against the list of DesignStrategy objects in the ThermalZones class. The list of ClusterSummary objects are converted to a Dataset<Row> and then written to a file in JSON format.

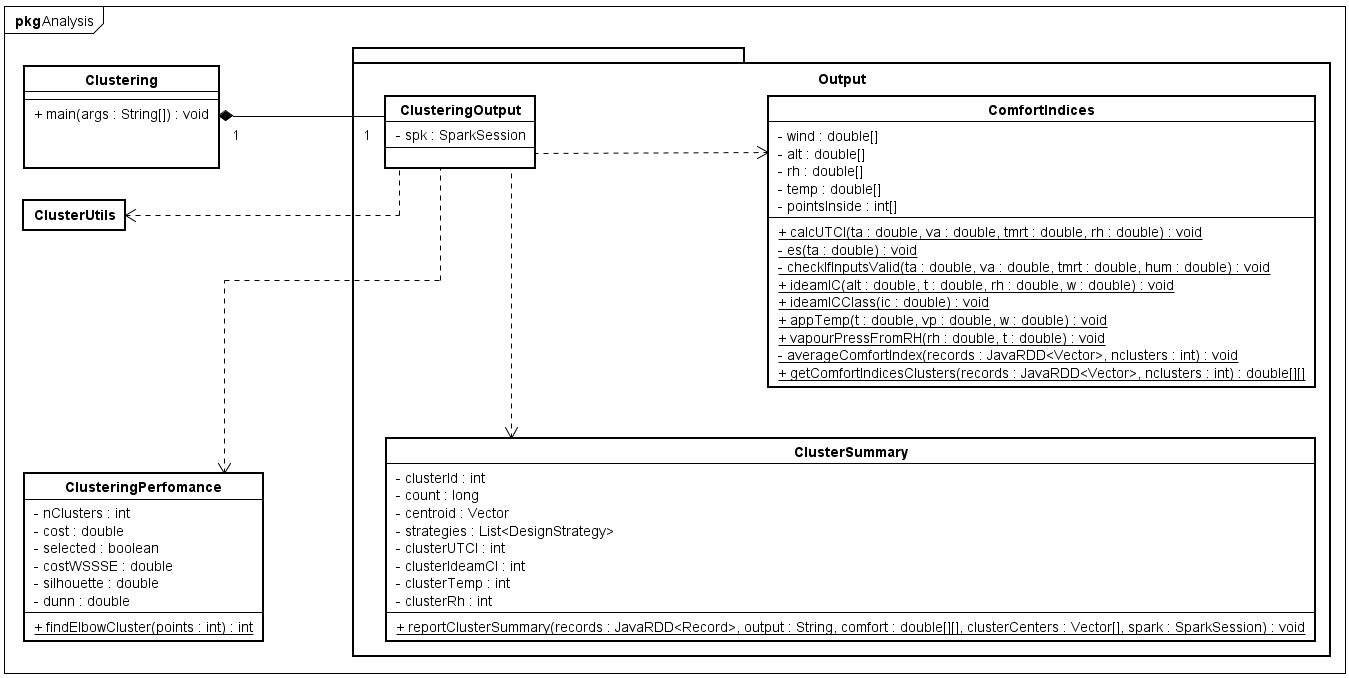


Figure 38 Analysis package and Output sub-package

#### Typical period

Generating a typical period involves grouping the data first by geographical location. Then for each location finding the most commonly occurring cluster number, that clusters average temperature and relative humidity used to define which design strategies are applicable.

#### Typical period with monthly granularity

Creating a typical period with monthly granularity first requires iterating the months of the year and filtering the data by each month. A typical period for each month is created by grouping by location and then finding the most frequent cluster number and assigning appropriate strategies and storing the results.

#### Year + month + cluster

This form of query results in one data set for each year and each month of the original data, the process iterates each year and each month (Figure 38) storing the filtered data and a summary of the frequency of design strategies at each iteration. Additionally, for each cluster the frequency of found design strategies is also stored.

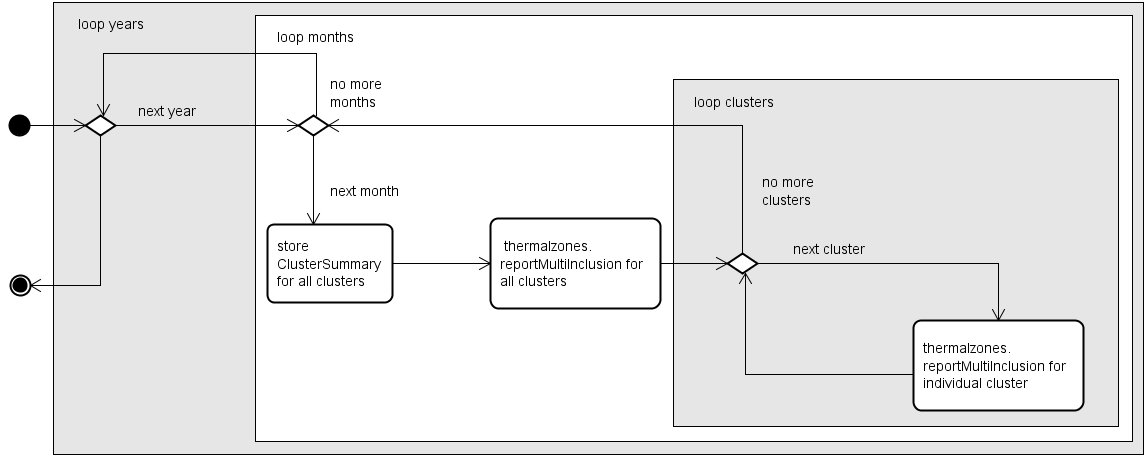


Figure 39 Year + month + cluster

## Dashboard + visualisation

Once the analysis successfully completes the Step on the EMR cluster is given the status of COMPLETED. The UI’s resources monitoring function will detect the new status and change the controls available on the interface, making the results button available and other controls disabled. Clicking the dashboard button triggers an event in the GUIWorkflowBuilder class that checks if a dashboard has been generated, if it has the web engine in the UI’s visualise tab is updated with the workflow’s dashboard URL. The application also opens the default browser and points to the same URL.

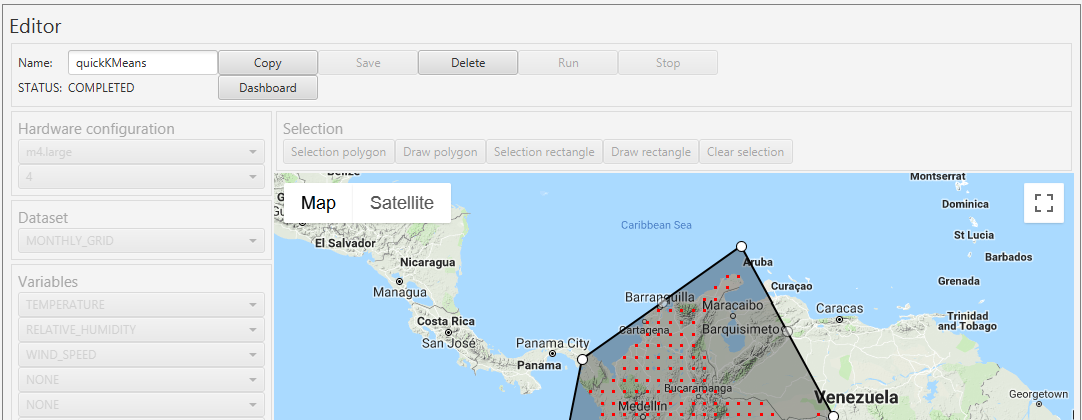


Figure 40 Dashboard button available

If a dashboard does not exist a new folder name is created which will be located within a public bucket on S3 and contain the dashboard data files and index.html. The GUIWorkflowBuilder creates a new thread that instantiates the GeoVisualisation class (Figure 41). The role of the GeoVisualisation class is to combine files output by the Spark application and transfer the merged files to a newly created space on a public bucket on S3. The Spark application is distributed across several machine instances on EMR specified by the user, within the application any process that requires writing output is also distributed so multiple parts are generated. The GeoVisualisation class iterates all objects with a keypath prefix corresponding to the results location on S3 passed to the analysis application. All files with the string “part” in the name that share the same root keypath are combined and transferred to the public bucket with the keypath comprised of the dashboard folder name and their root keypath. The text file containing the serialised description of the original work flow is renamed parameters.txt and transferred. A template index.html referencing a set of JavaScript files is also copied to the new location. Once the file combine-and-transfer is complete the user can click the dashboard button and examine the dashboard either within the visualise tab in the application on the default browser.

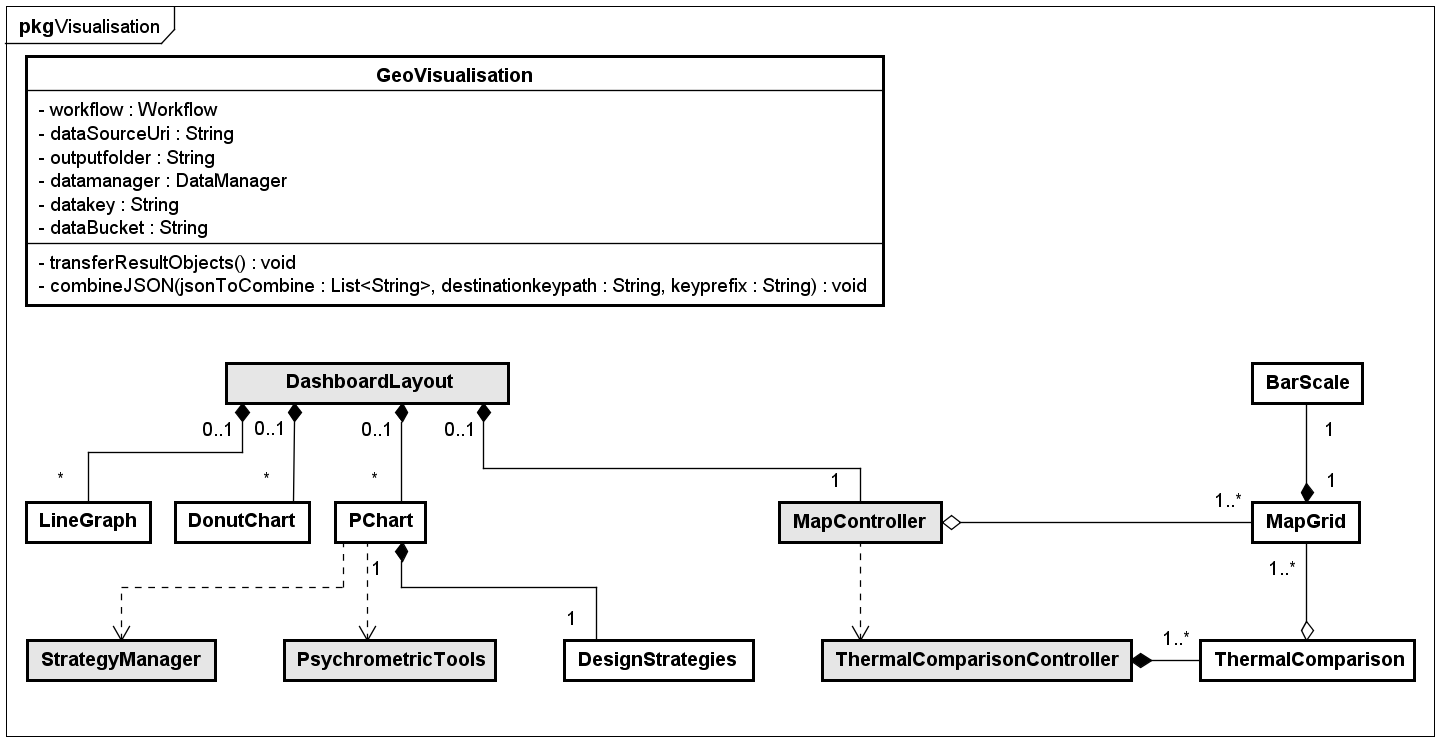


Figure 41 Visualisation package and classes.

The index.html is a landing page that references a set of shared JavaScript files that define the dashboard framework. Some of these are JavaScript classes while others (shown in grey in Figure 41) contain functions defining the behaviour of the interface. The classes define charts and maps that are instantiated several times on the dashboard to visualise the results according to the structure defined by the analytic process. Figure 42 is an activity diagram that describes the sequences of functions called on opening the page and where object instances are created. The zone shaded grey indicates a subset of the behaviour that is triggered when the user interacts with a control on the dashboard. Three sections are provided for examining the results; overview, cluster explorer and comfort comparison.

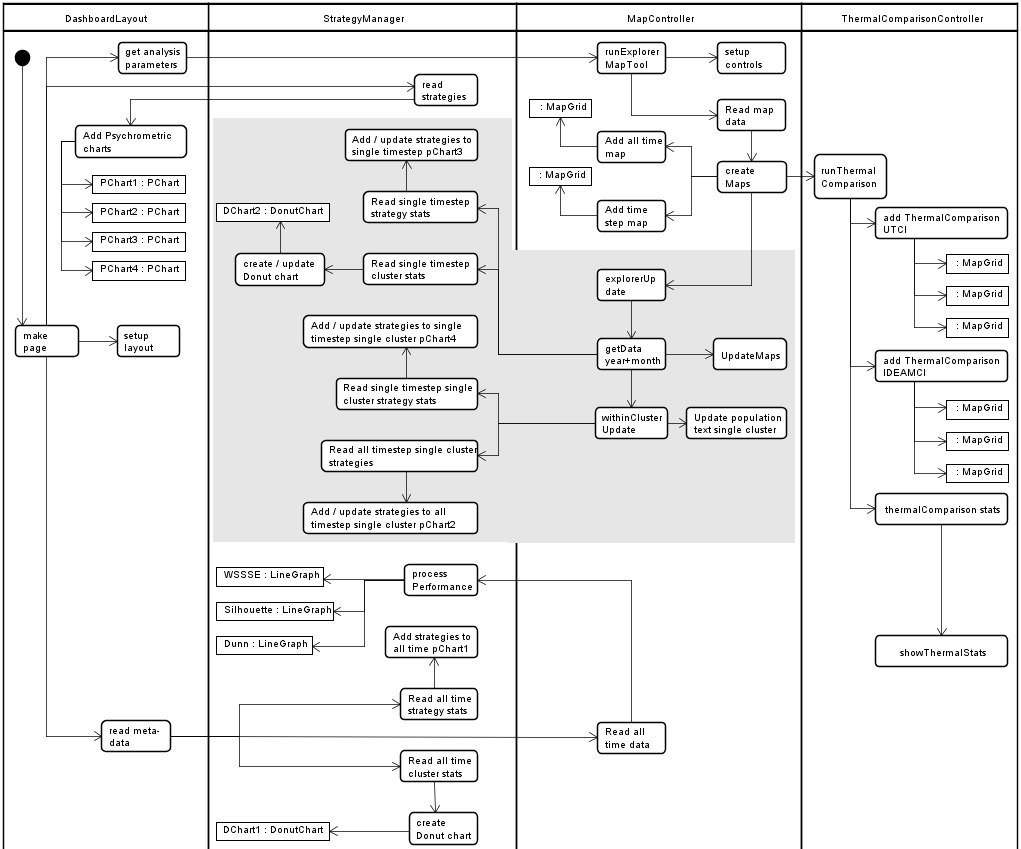


Figure 42 Dashboard behaviour model.

#### Overview

The overview section (Figure 43) displays the input parameters and performance indices plus the URL of the page. If the analysis involved the optimisation routine LineGraph objects are created for each of the three indices are shown, each with a red dot showing the optimal number of clusters. Scores for each index are shown on the left-hand panel. If the user selected the number of clusters the score for each index is shown in each panel (Figure 44).



Figure 43 Dashboard overview section for optimisation

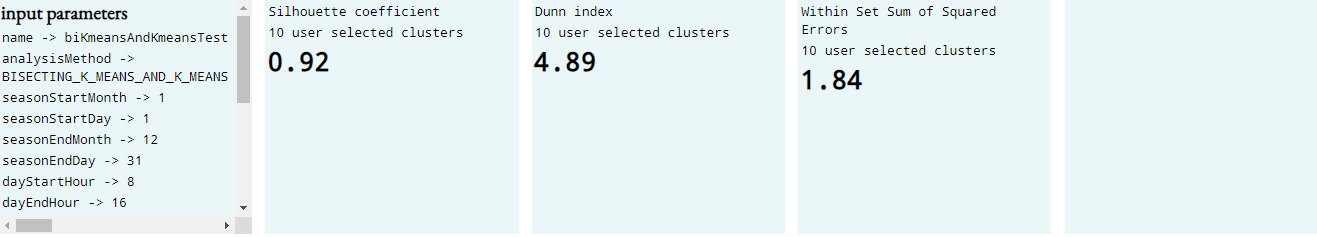


Figure 44 Dashboard overview section for single clustering solution

#### Cluster explorer

The cluster explorer section provides four ways of looking at the results. These are accessed by clicking on a tab to select the time step and then on the nested tab to select one of the following time-cluster combinations:

1. all time steps / all clusters
2. all time steps / single cluster
3. single time step / all clusters
4. single time step / single cluster

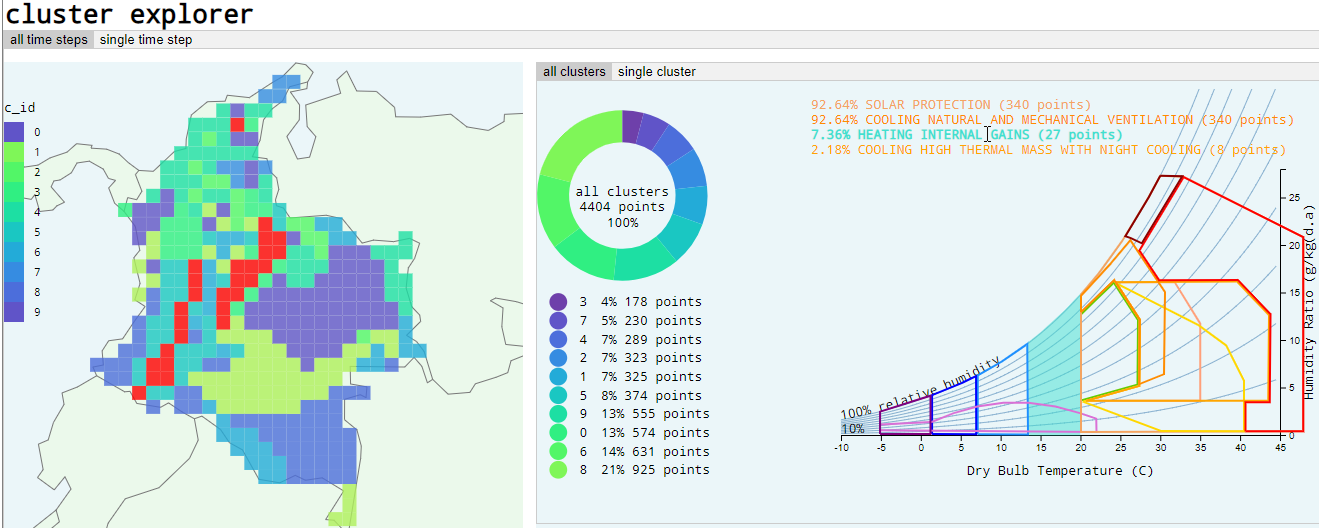


Figure 45 Dashboard cluster explorer all time steps / all clusters with interaction

On opening the page all time steps and clusters are presented this corresponds with the output described in Typical period. Instances of the MapGrid, DonutChart and PChart objects are used to provide the user can interact with these to explore the data. The PChart includes DesignStrategy class boundary zones as shown in the UI (Figure 31), listed on the chart are the strategies that correspond with the clusters shown on the MapGrid. Placing the mouse over the text highlights locations on the map where they are applicable and highlights the corresponding boundary on the psychrometric chart (Figure 45). The distribution of points between clusters is shown by the donut chart, placing the mouse over this highlights the corresponding cluster on the map and scale. Similarly, mouse over the map or scale highlights the corresponding clusters on the donut and scale or map. Selecting the single cluster tab shows provides a range control to select an individual cluster and examine its population, associated the strategies and their spatial distribution.

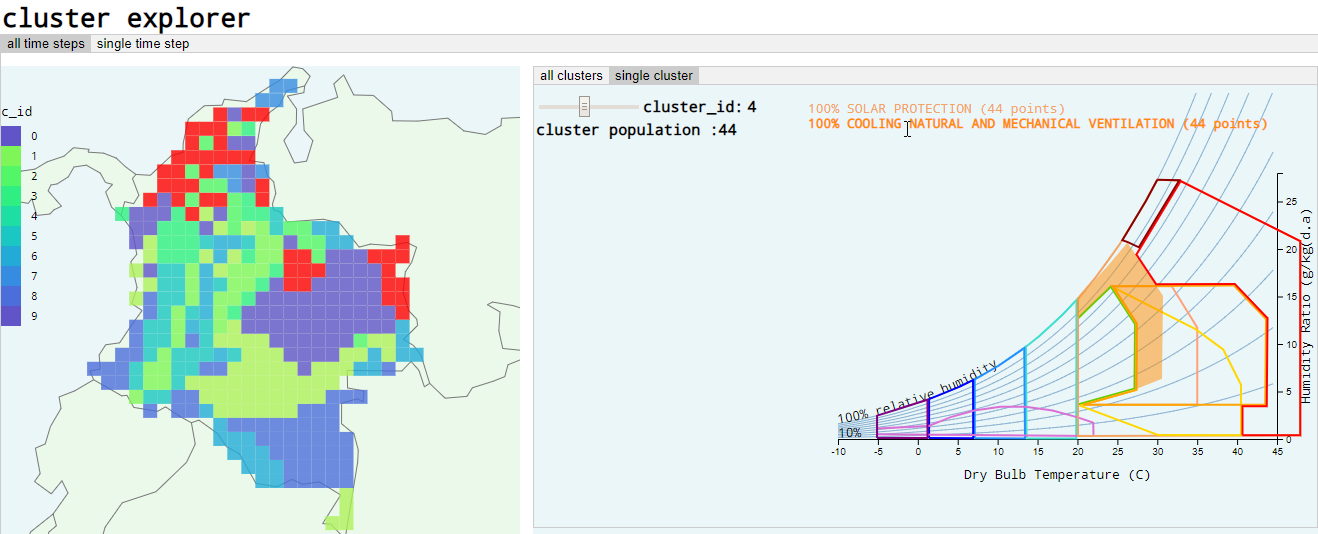


Figure 46 Dashboard cluster explorer all time steps / single cluster with interaction

Single time step / all clusters mode provides the option of exploring a specific year and month and examining the design strategies associated with that period. The interface includes MapGrid, DonutChart and PChart objects plus range controllers for year and month. This corresponds with the output described in Year + month + cluster. Selecting the single cluster tab in this mode provides more detailed study of a single cluster at a specific time (Figure 48).

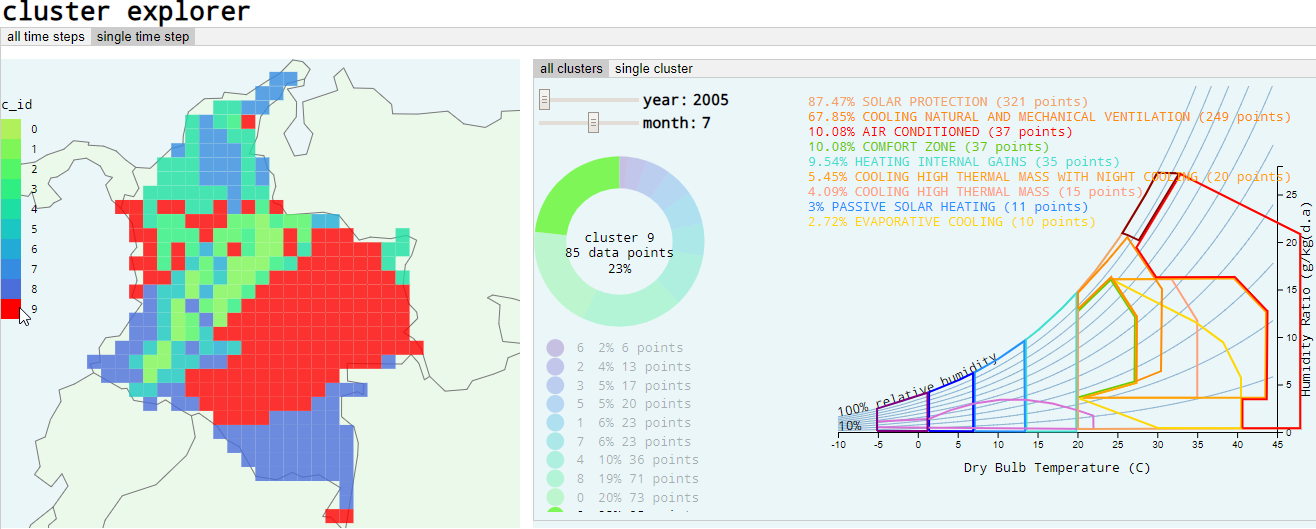


Figure 47 Dashboard cluster explorer single time step / all clusters with interaction

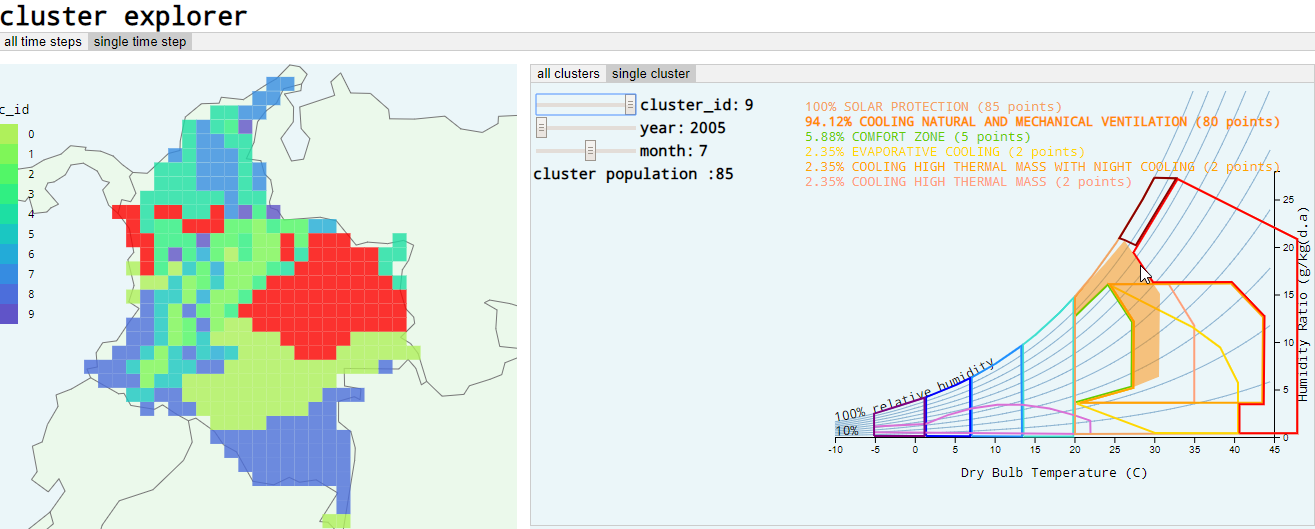


Figure 48 Dashboard cluster explorer single time step / single cluster with interaction

#### Comfort comparison

The last section on the dashboard compares comfort indices predicted by clusters to those calculated by the variables for each grid cell or city. Two comfort indices were defined during the analysis UTCI and IDEAMCI Figure 49. For each index a brief description of each index and link to more detail information is provide. A ThermalComparison object is created for each index, this includes three MapGrid objects showing the predicated index, calculated and delta. On the left-hand panel summary statistics of differences between predicated and calculated values is show as mean, standard deviation and root mean squared error.

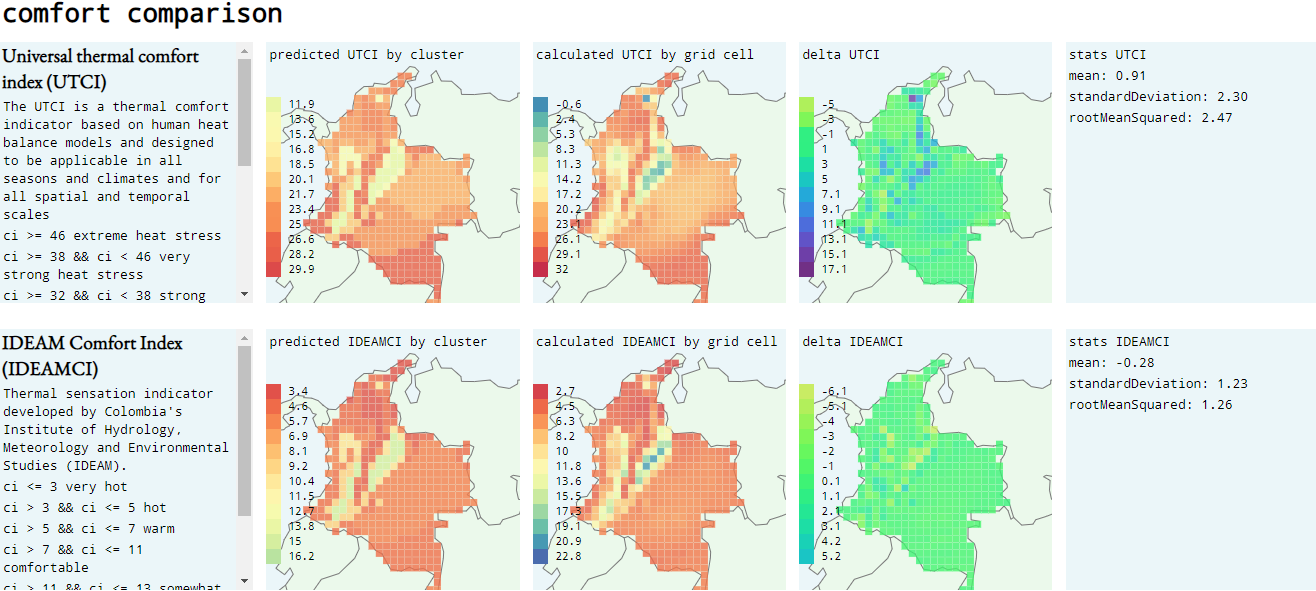


Figure 49 Dashboard thermal index comparison

## Class diagrams

### User Interface package

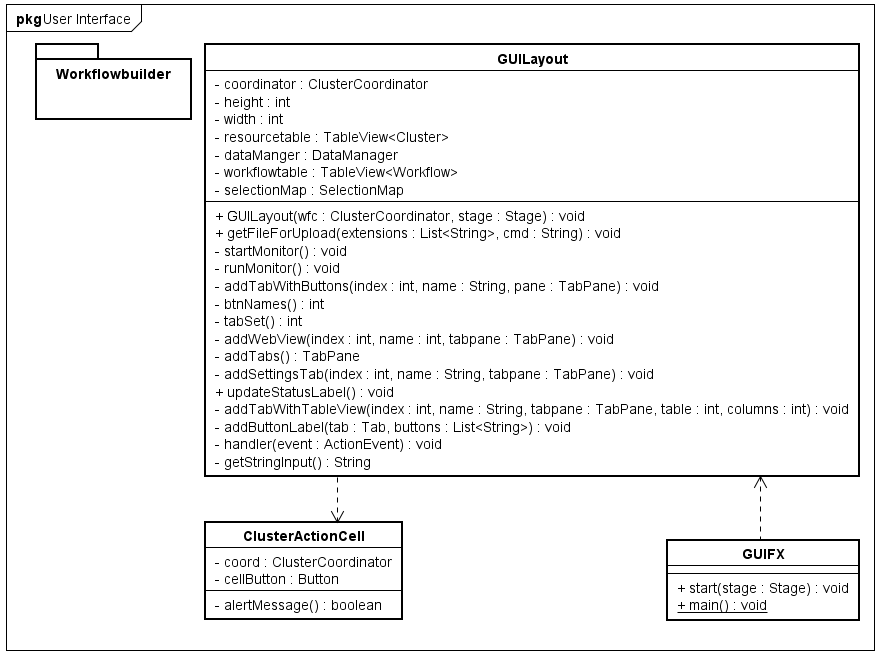


Figure 50 User Interface package

Key parts:

Sets up the UI with a series of tabs handles all non-workflow builder interaction

JavaFX Table views update whenever the Observable list that they contain is updated provide

Cluster action cells are used to provide buttons in the resource monitor to stop a cluster at any time

### Visualisation package

# Chapter 5. Results and evaluation

## Evaluation

The system is intended to produce recommendations for environmental design strategies based on the analysis of a set of weather data. The hypothesis to be tested is that design strategies can be linked with patterns discovered in weather data at various spatiotemporal scales and with different subsets of variables. It is proposed that this can be tested using a big data architecture that enables data analytics over large sets of weather data. Colombia is chosen for a test case as the tropical climate combined with the extreme topography and proximity to oceans.

Products of the system will be decomposed for evaluation; weather patterns will be assessed both quantitively and qualitatively and matching of recommended design strategies and the patterns evaluated qualitatively.

Several metrics exist to express the completeness and homogeneity of clusters discovered through analytics. Validity-measure or V-measure expresses how well both completeness and homogeneity are satisfied (Rosenberg and Hirschberg, 2007). Other metrics capture one or the other and include Purity, Entropy, Rand Index, misclassification index, f-measure, silhouette coefficient and cluster distortion of clusters can be calculated.

For results of all analysis methods visual inspection of graphical output will play an important role in evaluation of the system. Including interactive graphical representations will allow dynamic exploration of spatiotemporal results. Use of correlation matrices, 2D scatter plots and plotting georeferenced zones on maps will amplify knowledge discovery and allow products of the artefact to be presented to domain experts. Qualitative evaluation of the artefact and the products (patterns and recommendations) will be undertaken by identifying very specific use cases and developing walkthroughs.

### Statistical comparison of the different knowledge discovery methods used

K-Means vs Bisecting K Means vs Bisecting Kmeans + Kmeans

Performances – optimese to wssse and compare

### Statistical comparison between individual knowledge discovery methods

### Opinion by domain experts(s) – presentation of results and analysis

See interview question format in Smith dissertation example

Cognitive walkthrough, heuristic evaluation, review based see Shneiderman, B. (1998) Designing the user interface: Strategies for effective human computer interaction (3rd ed.). Reading, MA: Additon-Wesley Publishing

# Chapter 6. Conclusions

## Findings

## Future research

# References

Ambler, S. W. (2004) *The object primer : agile modeling-driven development with UML 2.0*. Cambridge University Press.

ASHRAE (2013) *2013 ASHRAE Handbook: Fundamentals*, *ASHRAE*. doi: 10.1163/ej.9789004155947.i-937.23.

Atluri, G., Karpatne, A. and Kumar, V. (2017) ‘Spatio-Temporal Data Mining: A Survey of Problems and Methods’, *ACM Comput. Surv*, 1(1). doi: 10.1145/nnnnnnn.nnnnnnn.

Avci Salma, C., Tekinerdogan, B. and Athanasiadis, I. N. (2017) ‘Chapter 4 – Domain-Driven Design of Big Data Systems Based on a Reference Architecture’, in *Software Architecture for Big Data and the Cloud*, pp. 49–68. doi: 10.1016/B978-0-12-805467-3.00004-1.

*AWS Toolkit for Eclipse* (no date). Available at: https://aws.amazon.com/eclipse/ (Accessed: 3 October 2018).

Begoli, E. and Horey, J. (2012) ‘Design principles for effective knowledge discovery from big data’, in *Proceedings of the 2012 Joint Working Conference on Software Architecture and 6th European Conference on Software Architecture, WICSA/ECSA 2012*, pp. 215–218. doi: 10.1109/WICSA-ECSA.212.32.

Bostock, M. (2017) *TopoJSON*, *Github*. Available at: https://github.com/topojson/topojson (Accessed: 3 May 2017).

Bracco, A. *et al.* (2017) ‘Advancing climate science with knowledge-discovery through data mining’, *npj Climate and Atmospheric Science*, 1(1), p. 4. doi: 10.1038/s41612-017-0006-4.

Buyya, R. *et al.* (2016) ‘Chapter 18 – eScience and Big Data Workflows in Clouds: A Taxonomy and Survey’, in *Big Data*, pp. 431–455. doi: 10.1016/B978-0-12-805394-2.00018-0.

Cgiar-csi.org. (2012) *CRU-TS v3.10.01 Historic Climate Database for GIS | CGIAR-CSI.* Available at: http://www.cgiar-csi.org/data/uea-cru-ts-v3-10-01-historic-climate-database (Accessed: 5 November 2017).

Chouksey, P. and Chauhan, A. S. (2017) ‘Weather Data Analytics using MapReduce and Spark’, *International Journal of Advanced Research in Computer and Communication Engineering*, 6(2). doi: 10.17148/IJARCCE.2017.6210.

Dagade, V. *et al.* (2015) ‘Big Data Weather Analytics Using Hadoop’, *International Journal of Emerging Technology in Computer Science & Electronics*, 14(2), pp. 976–1353. Available at: https://pdfs.semanticscholar.org/f2e4/918444be9b30f29132e93ce02d29ccf26eda.pdf (Accessed: 29 May 2018).

Das, T., Zaharia, M. and Wendell, P. (2015) *Spark Streaming*. Available at: https://databricks.com/blog/2015/07/30/diving-into-apache-spark-streamings-execution-model.html (Accessed: 21 February 2018).

Degaetano, A. T. (1996) ‘Delineation of Mesoscale Climate Zones in the Northeastern United States Using a Novel Approach to Cluster Analysis’, *Journal of Climate*, 9(8), pp. 1765–1782. doi: 10.1175/1520-0442(1996)009<1765:DOMCZI>2.0.CO;2.

Dimoudi, A. and Tompa, C. (2008) ‘Energy and environmental indicators related to construction of office buildings’, *Resources, Conservation and Recycling*, 53(1–2), pp. 86–95. doi: 10.1016/j.resconrec.2008.09.008.

Faghmous, J. H. and Kumar, V. (2014) ‘Spatio-temporal Data Mining for Climate Data: Advances, Challenges, and Opportunities’, in Chu, W. (ed.) *Springer Berlin Heidelberg*, pp. 83–116. doi: 10.1007/978-3-642-40837-3\_3.

Fang, W. *et al.* (2014) ‘Meteorological data analysis using MapReduce.’, *The Scientific World Journal*, 2014, p. 646497. doi: 10.1155/2014/646497.

El Fazziki, A. *et al.* (2015) ‘A multi-agent framework for a hadoop based air quality decision support system’, in *CEUR Workshop Proceedings*, pp. 45–59.

Forsythe, N., Blenkinsop, S. and Fowler, H. J. (2015) ‘Exploring objective climate classification for the Himalayan arc and adjacent regions using gridded data sources’, *Earth System Dynamics*, 6(1), pp. 311–326. doi: 10.5194/esd-6-311-2015.

Fountalis, I., Bracco, A. and Dovrolis, C. (2014) ‘Spatio-temporal network analysis for studying climate patterns’, *Climate Dynamics*, 42(3–4), pp. 879–899. doi: 10.1007/s00382-013-1729-5.

Fovell, R. G. and Fovell, M. Y. C. (1993) ‘Climate zones of the conterminous United States defined using cluster analysis’, *Journal of Climate*, pp. 2103–2135. doi: 10.1175/1520-0442(1993)006<2103:CZOTCU>2.0.CO;2.

Givoni, B. (1992) ‘Comfort, climate analysis and building design guidelines’, *Energy and Buildings*, 18, pp. 11–23. Available at: https://ac-els-cdn-com.liverpool.idm.oclc.org/037877889290047K/1-s2.0-037877889290047K-main.pdf?\_tid=67644909-7d5e-4718-b47f-80732d024251&acdnat=1527603110\_130b2a2e77cd97d48361202940532374 (Accessed: 29 May 2018).

González, O. (1998) *Metodología para el Calculo del Confort Climático en Colombia: NOTA TECNICA DEL IDEAM.* Bogota. Available at: http://documentacion.ideam.gov.co/openbiblio/bvirtual/007574/Metodologiaconfort.

Höppe, P. (1999) ‘The physiological equivalent temperature - A universal index for the biometeorological assessment of the thermal environment’, *International Journal of Biometeorology*. doi: 10.1007/s004840050118.

*http-server: a command-line http server* (2018) *npmjs*. Available at: https://www.npmjs.com/package/http-server (Accessed: 3 October 2018).

Jain, H. and Jain, R. (2017) ‘Big data in weather forecasting: Applications and challenges’, in *2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC)*. IEEE, pp. 138–142. doi: 10.1109/ICBDACI.2017.8070824.

Jarvis, A., H.I. Reuter, A. Nelson, E. G. (2008) *Hole-filled SRTM 90m for the globe Version 4 Database*. Available at: http://srtm.csi.cgiar.org (Accessed: 5 November 2017).

Jayanthi, D. and Sumathi, G. (2017) ‘Weather data analysis using spark — An in-memory computing framework’, in *2017 Innovations in Power and Advanced Computing Technologies (i-PACT)*. IEEE, pp. 1–5. doi: 10.1109/IPACT.2017.8245142.

Jayaratne, M. *et al.* (2017) ‘Apache spark based distributed self-organizing map algorithm for sensor data analysis’, in *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, pp. 8343–8349. doi: 10.1109/IECON.2017.8217465.

Jendritzky, G. and Höppe, P. (2017) ‘The UTCI and the ISB’, *International Journal of Biometeorology*. doi: 10.1007/s00484-017-1390-5.

Jones, P. and Harris, I. (2008) *Climatic Research Unit (CRU) time-series datasets of variations in climate with variations in other phenomena*, *NCAS British Atmospheric Data Centre*. Available at: http://catalogue.ceda.ac.uk/uuid/3f8944800cc48e1cbc29a5ee12d8542d (Accessed: 5 November 2017).

Karpatne, A. *et al.* (2017) ‘Theory-Guided Data Science: A New Paradigm for Scientific Discovery from Data’, *IEEE Transactions on Knowledge and Data Engineering*, 29(10), pp. 2318–2331. doi: 10.1109/TKDE.2017.2720168.

*KMeansModel (Spark 2.3.2 JavaDoc)* (no date). Available at: https://spark.apache.org/docs/latest/api/java/org/apache/spark/mllib/clustering/KMeansModel.html (Accessed: 18 October 2018).

Kodinariya, T. M. and Makwana, P. R. (2013) ‘Review on determining number of Cluster in K-Means Clustering’, *International Journal of Advance Research in Computer Science and Management Studies*.

Liu, F. *et al.* (2011) *NIST Cloud Computing Reference Architecture Recommendations of the National Institute of Standards and Technology*. Available at: https://ws680.nist.gov/publication/get\_pdf.cfm?pub\_id=909505 (Accessed: 3 October 2018).

Liu, Y. and Weisberg, R. H. (2005) ‘Patterns of ocean current variability on the West Florida Shelf using the self-organizing map’, *Journal of Geophysical Research: Oceans*, 110(6), pp. 1–12. doi: 10.1029/2004JC002786.

Liu, Y. and Weisberg, R. H. (2011) ‘A Review of Self-Organizing Map Applications in Meteorology and Oceanography’, in Igadwa Mwasiagi, J. (ed.) *Self Organizing Maps - Applications and Novel Algorithm Design*. www.intechopen.com. doi: 10.5772/13146.

Liu, Y., Weisberg, R. H. and Mooers, C. N. K. (2006) ‘Performance evaluation of the self-organizing map for feature extraction’, *Journal of Geophysical Research: Oceans*, 111(5). doi: 10.1029/2005JC003117.

Lopes, D., Palmer, K. and O’Sullivan, F. (2017) ‘Chapter 10 – Big Data: A Practitioners Perspective’, in *Software Architecture for Big Data and the Cloud*, pp. 167–179. doi: 10.1016/B978-0-12-805467-3.00010-7.

Luna-Romera, J. M. *et al.* (2016) ‘An approach to silhouette and dunn clustering indices applied to big data in spark’, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. doi: 10.1007/978-3-319-44636-3\_15.

Manzano-Agugliaro, F. *et al.* (2015) ‘Review of bioclimatic architecture strategies for achieving thermal comfort’, *Renewable and Sustainable Energy Reviews*. Pergamon, 49, pp. 736–755. doi: 10.1016/J.RSER.2015.04.095.

Mariam Varghese, S. (2015) ‘Leveraging Map Reduce With Hadoop for Weather Data Analytics’, *IOSR Journal of Computer Engineering Ver. II*, 17(3), pp. 2278–661. doi: 10.9790/0661-17320612.

Mell, P. and Grance, T. (2011) *The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology*. doi: 10.6028/NIST.SP.800-145.

Milne, M., Liggett, R. and Benson, A. (2009) ‘Climate Consultant 4.0 develops design guidelines for each unique climate’, *American Solar Energy Society Meeting*. Available at: http://www.energy-design-tools.aud.ucla.edu/papers/ases09-milne.pdf (Accessed: 10 April 2018).

*MLlib: Main Guide - Spark 2.3.2 Documentation* (no date). Available at: https://spark.apache.org/docs/latest/ml-guide.html (Accessed: 16 October 2018).

Netzel, P. *et al.* (2016) ‘On Using a Clustering Approach for Global Climate Classification’, *Journal of Climate*, 29(9), pp. 3387–3401. doi: 10.1175/JCLI-D-15-0640.1.

Nikolaou, T. G. *et al.* (2012) ‘On the application of clustering techniques for office buildings’ energy and thermal comfort classification’, *IEEE Transactions on Smart Grid*. doi: 10.1109/TSG.2012.2215059.

Olgyay, V. and Olgyay, A. (1963) *Design With Climate: Bioclimatic Approach to Architectural Regionalism*. Princeton University Press.

Omer, A. M. (2008) ‘Energy, environment and sustainable development’, *Renewable and Sustainable Energy Reviews*, 12, pp. 2265–2300. doi: 10.1016/j.rser.2007.05.001.

Rhee, J. *et al.* (2008) ‘Delineation of climate regions using in-situ and remotely-sensed data for the Carolinas’, *Remote Sensing of Environment*, 112(6), pp. 3099–3111. doi: 10.1016/j.rse.2008.03.001.

Rodenburg, B. and Maria Fiore, M. (2017) *Detecting Weather Twins using Apache Spark*, *LSDE: Large Scale Data Engineering 2017*. Available at: https://event.cwi.nl/lsde/2017/showcase\_n2.shtml (Accessed: 29 May 2018).

Rodriguez, M. A. and Buyya, R. (2017) ‘Chapter 18 – Scientific Workflow Management System for Clouds’, in *Software Architecture for Big Data and the Cloud*, pp. 367–387. doi: 10.1016/B978-0-12-805467-3.00018-1.

Rosenberg, A. and Hirschberg, J. (2007) ‘V-Measure: A conditional entropy-based external cluster evaluation measure’, pp. 410–420. Available at: http://www.aclweb.org/anthology/D07-1043 (Accessed: 6 June 2018).

Terpilowski, R. (no date) *GMapsFX*. Available at: https://rterp.github.io/GMapsFX/ (Accessed: 18 October 2018).

Thinsungnoen, T. *et al.* (2015) ‘The Clustering Validity with Silhouette and Sum of Squared Errors’, in *The Proceedings of the 2nd International Conference on Industrial Application Engineering 2015*. doi: 10.12792/iciae2015.012.

Unidata (2012) ‘NetCDF-Java library and TDS version 4.6.9’. Boulder. CO: UCAR/Unidata. doi: http://doi.org/10.5065/D6RN35XM.

Wentz, F. *et al.* (2015) *Remote Sensing Systems Cross-Calibrated Multi-Platform (CCMP) 6-hourly ocean vector wind analysis product on 0.25 deg grid, Version 2.0, [subset: CCMP V2.0 Level-3.5].* Santa Rosa, CA: Remote Sensing Systems. Available at: www.remss.com/measurements/ccmp (Accessed: 5 November 2017).

Zaharia, M. *et al.* (2013) ‘Discretized Streams: Fault-Tolerant Streaming Computation at Scale’, *Sosp*, (1), pp. 423–438. doi: 10.1145/2517349.2522737.

Zscheischler, J., Mahecha, M. D. and Harmeling, S. (2012) ‘Climate classifications: The value of unsupervised clustering’, in *Procedia Computer Science*, pp. 897–906. doi: 10.1016/j.procs.2012.04.096.

# List of Tables

[Table 1 Key Research methods used in the dissertation 7](#_Toc526246822)

[Table 2 System development process 8](#_Toc526246823)

# List of Figures

[Figure 1 process diagram 7](#_Toc527463438)

[Figure 2 Manzano-Agugliaro et al. (2015) adapted version of the psychrometric chart 10](#_Toc527463439)

[Figure 3 Reference architecture of a WMS 13](#_Toc527463440)

[Figure 4 Define workflow 17](#_Toc527463441)

[Figure 5 Run and monitor workflow 18](#_Toc527463442)

[Figure 6 Output and visualise results 18](#_Toc527463443)

[Figure 7 Manage design strategies 19](#_Toc527463444)

[Figure 8 Proposed architecture (extended from: Lopes, Palmer and O’Sullivan, 2017) 19](#_Toc527463445)

[Figure 9 Architecture for the system. 20](#_Toc527463446)

[Figure 10 Simple package diagram. 20](#_Toc527463447)

[Figure 11 Development increments (extended from Ambler (2004), p119). 21](#_Toc527463448)

[Figure 12 Deployment on AWS 22](#_Toc527463449)

[Figure 13 Flow diagram showing the stages of the data preparation process. 23](#_Toc527463450)

[Figure 14 Packages 25](#_Toc527463451)

[Figure 15 Sequence diagram for defining and running a workflow 27](#_Toc527463452)

[Figure 16 Sequence diagram for monitoring workflows and resources. 27](#_Toc527463453)

[Figure 17 Sequence diagram for show or generate dashboard. 28](#_Toc527463454)

[Figure 18 Sequence diagram for generate dashboard. 28](#_Toc527463455)

[Figure 19 User Interface package 29](#_Toc527463456)

[Figure 20 Workflowbuilder package 30](#_Toc527463457)

[Figure 21 Analysis package 33](#_Toc527463458)

[Figure 22 Coordination package 35](#_Toc527463459)

[Figure 23 Visualisation package 36](#_Toc527463460)

[Figure 24 User interface 24](#_Toc527463461)

[Figure 25 Results dashboard 24](#_Toc527463462)

[Figure 26 Results dashboard 25](#_Toc527463463)