**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

03/12/2018

ABSTRACT

**BIG CLIMATE DATA ANALYTICS: EFFECTIVE KNOWLEDGE-DISCOVERY FROM COLOMBIA’S WEATHER DATA**

By

Roland Hudson

The aim of this dissertation is to develop a distributed, machine-learning application that classifies Colombian climate data and provides decision support to environmental designers seeking to understand the spatial and temporal use of low-energy design strategies. These strategies can help provide more comfortable living and working conditions for people using the buildings and reduce the need for heating and cooling, lowering emissions and energy consumption.Implementing these strategies requires understanding the local and regional climate conditions over different periods. In Colombia, a lack of seasons, extreme topographical variations and subtle tropical patterns make identifying localized, low-energy construction strategies complex.

The dissertation provides an overview of literature covering previous relevant research publications and theory, this includes examination of low-energy design strategies and spatiotemporal nature of climate data. The literature review summarises big data tools and systems, relevant to climate data and identifies recent applications of machine learning to classify climates. The analysis and design chapter describes use cases, identifies system requirements and proposes a general system design. The implementation chapter documents how Agile Model Driven Development is used to model, implement and document a software artefact, which integrates a local application with analysis and visualisation in a distributed, cloud-based environment.

The completed system allows users to explore a climatic dataset with the aim of finding spatiotemporal patterns and linking these to low-energy design techniques. A graphical user interface provides tools to configure analytic jobs, create and edit design strategies on a psychrometric chart and monitor the status of cloud resources. Apache Spark provides a distributed framework for processing the data using hierarchical and non-hierarchical clustering techniques. The system links clusters to design strategies and calculates domain and data-centric indices that indicate clustering performance.

The evaluation chapter describes the system appraisal process involving a series of analytic experiments that capture clustering specific and domain centric performance metrics. Both metrics indicate that the artefact can classify the Colombian climate into distant classes that represent climate patterns. Visualisation of the results of the evaluation support this finding by graphically representing known patterns in space and time and linking these with appropriate design strategies. Finally, software walkthroughs with domain experts suggest that with modification of certain controls and re-evaluation of the primary users the application is useful in industry.

DECLARATION

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 another.

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,



Roland Hudson

**Student, Supervisors and Classes:**

|  |  |
| --- | --- |
| Student name: | ROLAND HUDSON |
| Student ID number: | H00056278 |
| GDI name: | TALY SHARON |
| CRMT class ID: | LAUR-906-2-201843-2 |
| DA name: | YURI DEMCHENKO |
| CAC class ID: | UKL1.CKIT.702.H00028508 |

ACKNOWLEDGEMENTS

The dissertation was developed as a spin off project to research conducted during 2017 with Rodrigo Velasco at the Universidad Piloto de Colombia, the author is grateful for the opportunity to collaborate on the original project. The author wishes to thank Instituto de Hidrología, Meteorología y Estudios Ambientales for supplying Colombian weather data. Thanks to the domain experts that patiently sat through software walkthroughs and provided broader insight on the applicability and usefulness of this research. The author thanks Taly Sharon for applying pressure to define the proposal and Yuri Demchenko for advising the dissertation. Finally, thanks Elisa for positively encouraging continuous education and for tolerating endless weekends doing ‘homework’.

**TABLE OF CONTENTS**

Page

[LIST OF TABLES 3](#_Toc531594836)

[LIST OF FIGURES 3](#_Toc531594837)

[Chapter 1. Introduction 6](#_Toc531594838)

[1.1 Scope 6](#_Toc531594839)

[1.2 Problem statement 6](#_Toc531594840)

[1.2.1 Complexity of existing workflow 7](#_Toc531594841)

[1.3 Approach 8](#_Toc531594842)

[1.3.1 Literature review 8](#_Toc531594843)

[1.3.2 Artefact 8](#_Toc531594844)

[1.3.3 Application development methodology 8](#_Toc531594845)

[1.3.4 Qualitative evaluation of artefact by domain experts 9](#_Toc531594846)

[1.3.5 Quantitative evaluation of output from artefact results 10](#_Toc531594847)

[1.4 Outcome 10](#_Toc531594848)

[Chapter 2. Background and review of literature 11](#_Toc531594849)

[2.1 Background 11](#_Toc531594850)

[2.2 Literature review 12](#_Toc531594851)

[2.2.1 Low-energy environmental design strategies 12](#_Toc531594852)

[2.2.2 Climate data is spatiotemporal 14](#_Toc531594853)

[2.2.3 Examples of applied ST data mining methods applied to climate data 15](#_Toc531594854)

[2.2.4 Challenges for ST data mining 15](#_Toc531594855)

[2.2.5 Cloud computing 16](#_Toc531594856)

[2.2.6 Big data tools 16](#_Toc531594857)

[2.3 Theory 22](#_Toc531594858)

[2.3.1 Data: 22](#_Toc531594859)

[2.3.2 Analytics: Clustering applied to climate classification 22](#_Toc531594860)

[2.3.3 Expertise: 26](#_Toc531594861)

[2.4 Terms 27](#_Toc531594862)

[Chapter 3. Analysis and design 29](#_Toc531594863)

[3.1 General use case analysis 29](#_Toc531594864)

[3.1.1 System actors 29](#_Toc531594865)

[3.1.2 Define workflow 30](#_Toc531594866)

[3.1.3 Run workflow + monitor resources 31](#_Toc531594867)

[3.1.4 Output + visualise results 31](#_Toc531594868)

[3.1.5 Manage design strategies 32](#_Toc531594869)

[3.2 General requirements 32](#_Toc531594870)

[3.2.1 Sequence diagram for defining and running a workflow 33](#_Toc531594871)

[3.2.2 Sequence diagram for monitoring workflows and resources 33](#_Toc531594872)

[3.2.3 Sequence diagrams for visualising results 34](#_Toc531594873)

[3.3 General System Architecture 36](#_Toc531594874)

[3.4 Design methodology 37](#_Toc531594875)

[3.5 Development iterations 38](#_Toc531594876)

[Chapter 4. Implementation 40](#_Toc531594877)

[4.1 Environment 40](#_Toc531594878)

[4.2 Architecture for AWS cloud infrastructure 40](#_Toc531594879)

[4.3 Packages 41](#_Toc531594880)

[4.4 Climate data 44](#_Toc531594881)

[4.5 Workflow management system 45](#_Toc531594882)

[4.5.1 Workflowbuilder and Coordination 45](#_Toc531594883)

[4.5.2 Workflow creation 46](#_Toc531594884)

[4.5.3 Monitoring resources and workflows 46](#_Toc531594885)

[4.5.4 Running a workflow 49](#_Toc531594886)

[4.5.5 Management of design strategies 50](#_Toc531594887)

[4.6 Analytics 51](#_Toc531594888)

[4.6.1 Overview 51](#_Toc531594889)

[4.6.2 Parsing parameters and filtering Data 52](#_Toc531594890)

[4.6.3 Performance Indices 53](#_Toc531594891)

[4.6.4 Classification and design strategy assignment 54](#_Toc531594892)

[4.6.5 Output of results 54](#_Toc531594893)

[4.7 Dashboard + visualisation 56](#_Toc531594894)

[4.7.1 Overview 58](#_Toc531594895)

[4.7.2 Cluster Summary 59](#_Toc531594896)

[4.7.3 Cluster explorer 59](#_Toc531594897)

[4.7.4 Comfort comparison 61](#_Toc531594898)

[4.8 Summary 62](#_Toc531594899)

[Chapter 5. Results and evaluation 64](#_Toc531594900)

[5.1 Evaluation 64](#_Toc531594901)

[5.2 Quantitative evaluation 64](#_Toc531594902)

[5.2.1 Metrics used for evaluation 64](#_Toc531594903)

[5.2.2 Evaluation approach 64](#_Toc531594904)

[5.2.3 Statistical evaluation of analyses 66](#_Toc531594905)

[5.2.4 Graphical evaluation 70](#_Toc531594906)

[5.3 Qualitative evaluation 72](#_Toc531594907)

[Opinion by domain experts 72](#_Toc531594908)

[5.4 Hypothesis Validation 75](#_Toc531594909)

[Chapter 6. Conclusions 77](#_Toc531594910)

[6.1 Lessons Learned 77](#_Toc531594911)

[Process and technology 77](#_Toc531594912)

[6.2 Industrial Application and Limitations 78](#_Toc531594913)

[6.3 Knowledge-discovery 79](#_Toc531594914)

[6.4 Future research 80](#_Toc531594915)

[REFERENCES CITED 82](#_Toc531594916)

LIST OF TABLES

Page

[Table 1 Development activity details 9](#_Toc531594917)

[Table 2 Evaluation workflows 66](#_Toc531594918)

[Table 3 Ranked results 67](#_Toc531594919)

LIST OF FIGURES

Page

[Figure 1 Agile model driven design phases 9](#_Toc531594920)

[Figure 2 Adapted version of the psychrometric chart (Manzano-Agugliaro *et al.,* 2015) 13](#_Toc531594921)

[Figure 3 Reference architecture of a WMS 17](#_Toc531594922)

[Figure 4 Big data reference architecture (Avci Salma, Tekinerdogan and Athanasiadis, 2017) 18](#_Toc531594923)

[Figure 5 Spark unified stack (Zaharia et al., 2015) 19](#_Toc531594924)

[Figure 6 Spark cluster components (“Cluster Mode Overview - Spark 2.4.0 Documentation,” n.d.) 20](#_Toc531594925)

[Figure 7 Hierarchical clustering steps 23](#_Toc531594926)

[Figure 8 Non-hierarchical k-means steps: (a) Original data. (b) Random initial centroids. (c-f) Two iterations of: assign data points to closest cluster, recalculate cluster centroid. Adapted from Piech (2013). 24](#_Toc531594927)

[Figure 9 Define workflow 31](#_Toc531594928)

[Figure 10 Run and monitor workflow 31](#_Toc531594929)

[Figure 11 Output and visualise results 32](#_Toc531594930)

[Figure 12 Manage design strategies 32](#_Toc531594931)

[Figure 13 Sequence diagram for defining and running a workflow 33](#_Toc531594932)

[Figure 14 Sequence diagram for monitoring workflows and resources. 34](#_Toc531594933)

[Figure 15 Sequence diagram for showing or generating a dashboard. 35](#_Toc531594934)

[Figure 16 Sequence diagram for generate dashboard also see Figure 15. 35](#_Toc531594935)

[Figure 17 High-level system architecture based around IaaS provided by AWS 36](#_Toc531594936)

[Figure 18 Architecture for the user interface, coordination, visualisation and analysis elements, including subcomponents 37](#_Toc531594937)

[Figure 19 Simplified package diagram. 37](#_Toc531594938)

[Figure 20 Development increments (extended from Ambler (2004), p119). 38](#_Toc531594939)

[Figure 21 Deployment on AWS 41](#_Toc531594940)

[Figure 22 Coordination, User Interface and Visualisation packages overview 43](#_Toc531594941)

[Figure 23 Analysis package overview 44](#_Toc531594942)

[Figure 24 Flow diagram showing the stages of the data preparation process 45](#_Toc531594943)

[Figure 25 GUI for the Workflowbuilder 47](#_Toc531594944)

[Figure 26 Resource and workflow monitoring algorithm 47](#_Toc531594945)

[Figure 27 UI for monitoring resources and workflows 48](#_Toc531594946)

[Figure 28 Discrete behavioural states for a workflow 49](#_Toc531594947)

[Figure 29 UI showing workflow with status COMPLETED 49](#_Toc531594948)

[Figure 30 Logical steps in processing workflows 50](#_Toc531594949)

[Figure 31 Logical steps in running a workflow 50](#_Toc531594950)

[Figure 32 UI showing mock-up for the design strategy editor 51](#_Toc531594951)

[Figure 33 AWS interface showing input parameters for a Step on EMR 52](#_Toc531594952)

[Figure 34 Logical steps in the analytics process 52](#_Toc531594953)

[Figure 35 Graph showing the elbow method 54](#_Toc531594954)

[Figure 36 Sequence for extracting data for each cluster, each month and each year 56](#_Toc531594955)

[Figure 37 UI showing the dashboard button available 57](#_Toc531594956)

[Figure 38 Behaviour model for the JavaScript dashboard system 58](#_Toc531594957)

[Figure 39 Section of dashboard UI showing the results of an optimised clustering analysis 59](#_Toc531594958)

[Figure 40 Section of dashboard UI showing the results for a single clustering solution 59](#_Toc531594959)

[Figure 41 Section of dashboard UI showing summary of each cluster’s population, design strategies, variable ranges and comfort indices 59](#_Toc531594960)

[Figure 42 Section of dashboard UI showing cluster explorer with all time-steps and all clusters displayed. The user’s mouse is over the design strategy “heating internal gains” and corresponding map cells are highlighted red 59](#_Toc531594961)

[Figure 43 Section of dashboard UI showing cluster explorer with all time steps and single cluster displayed 60](#_Toc531594962)

[Figure 44 Section of dashboard UI showing cluster explorer with single time step and all clusters displayed 61](#_Toc531594963)

[Figure 45 Section of dashboard UI showing cluster explorer with single time step and single cluster displayed 61](#_Toc531594964)

[Figure 46 Section of dashboard UI showing thermal index comparison. Top row UTCI comparison, predicted by clusters (left), calculated (centre) and error (right). Bottom row compares IDEAMCI 62](#_Toc531594965)

[Figure 47 Results from optimised studies. Left: k =6 at first Dunn maximum of 3.3. Right: inflexion is at k = 11 for WSSSE of 20. 65](#_Toc531594966)

[Figure 48 WSSSE dissimilarity matrix 68](#_Toc531594967)

[Figure 49 Dunn index dissimilarity matrix 68](#_Toc531594968)

[Figure 50 UTCI rmse dissimilarity matrix 69](#_Toc531594969)

[Figure 51 IDEAMCI rmse dissimilarity matrix 69](#_Toc531594970)

[Figure 52 Silhouette index dissimilarity matrix 69](#_Toc531594971)

[Figure 53 Comfort comparison for KM11TrRh 70](#_Toc531594972)

[Figure 54 Comfort comparison for BKM11TRhWs 70](#_Toc531594973)

[Figure 55 Cluster mapping. Left: KM11TrRh. Right: BKM11TRhWs 71](#_Toc531594974)

[Figure 56 Annual cluster mapping for BKM11TRhWs 71](#_Toc531594975)

[Figure 57 Design strategies and geographic location (red cells) for BKM11TRhWs. A: solar protection. B: cooling by natural and mechanical ventilation. C: comfort zone. D: heating from internal gains. E: cooling by high thermal mass and night cooling. 72](#_Toc531594976)

[Figure 58 Workflowbuilder package class diagrams for AnalysisParameters, Workflow SelectionMap classes 94](#_Toc531594977)

[Figure 59 Workflowbuilder package class diagrams for GUIWorkflowbuilder class 95](#_Toc531594978)

[Figure 60 Coordination package class diagrams for Cluster, ClusterCoordinator and DataManager classes 96](#_Toc531594979)

[Figure 61 Analysis package and ReadFilter sub-package class diagrams for ClusterParams, FilterData, Record and TimeLog classes 97](#_Toc531594980)

[Figure 62 Analysis package and Performance sub-package class diagrams for KMeansPerfomance, ClusteringPerformance, and ClusterIndices classes 97](#_Toc531594981)

[Figure 63 Analysis package and Classification sub-package class diagrams for SimpleKMeans, BiKMeans, DesignStrategy and ThermalZones classes 98](#_Toc531594982)

[Figure 64 Analysis package and Output sub-package class diagrams for ClusteringOutput, ComfortIndices, ClusteringPerformance and ClusterSummary classes 98](#_Toc531594983)

[Figure 65 Visualisation package and class diagram for GeoVisualisation class 99](#_Toc531594984)

# 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 & 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). Buildings expend energy mainly on heating, lighting and cooling. Energy efficiency is defined as 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 periods.

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. Subtle and inconsistent seasonal patterns associated with tropical latitudes add complexity.

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 Colombian 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. Heating and cooling systems are required to correct these conditions, which are expensive to install, costly to run and produce 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, active or passive solar heating to gain heat energy from solar radiation.
* Developing the architectural application – the designer synthesises the previous three steps into a design proposal.
* Simulation tools can confirm or optimise the selected design approach.

The designer must also consider usage patterns of the building, increasing the complexity of the process. Buildings are rarely in use constantly. Depending on use, occupancy can vary daily (residential properties are most used during evenings and night-time), weekly (office buildings are 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. Usage patterns, activity types and orientation of spaces within buildings can also vary. Each space can therefore require a different design response.

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 period.

## Approach

The project methodology begins with a literature review, then, following a specific development model an IT artefact is developed. Finally, using qualitative and quantitative methods are used to evaluate the artefact.

### Literature review

The literature review surveys low-energy architectural design strategies, describing how these relate to climate conditions and design criteria. This background discusses the spatiotemporal nature of climate data and techniques for data mining meteorological data. 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. The literature defines knowledge-discovery as a process model and describes applications that seek patterns in climate data.

### 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 shown in Figure 1. Each activity involves specific techniques linked to deliverables and specific UML models (Figure 1).

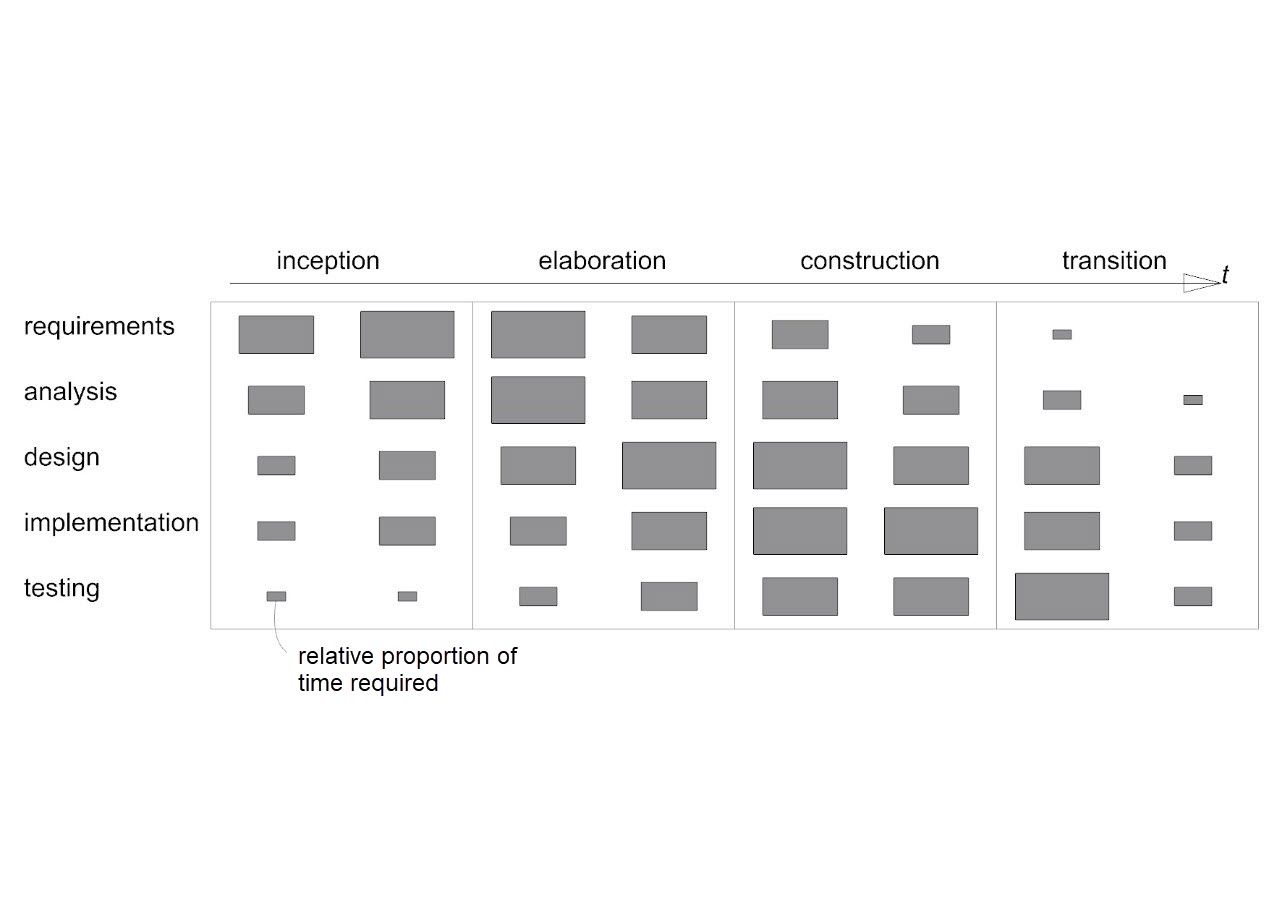


Figure 1 Agile model driven design phases

Table 1 Development activity details

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |  |

### Qualitative evaluation of artefact by domain experts

Software walkthroughs, presentations and interviews with domain experts form the basis of the qualitative evaluation. Experts review studies and results from the application and their opinions captured and summarised. With expert review, it will be possible to evaluate if the artefact offers a useful tool for the industry and if the artefact can identify localised construction approaches.

### 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 the methods applied include distance metrics for clustering. Evaluation of the application also includes verification, validation and testing– tests are 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, and link them to localized, climate-responsive design and construction strategies. Application of these strategies can lead to buildings that perform better in terms of production costs, lifetime running costs (reduced heating and cooling) and occupant comfort.

The project proposes that by combining a workflow management system with spatiotemporal data mining techniques in a big data infrastructure, it will be possible to identify localized design and construction approaches that respond to the unique weather conditions in Colombia.

The system integrates best practices for the storage, processing, analysis, management and visualization of data big data. Specific focus is on enabling analytics and visualization to facilitate 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.

Code and documentation for the artefact area available: <https://github.com/rolyhudson/climacolombia.git>. Results from the evaluation process, demonstrating the online visualisation of results are available: <http://lacunae.io/>.

# Background and review of literature

## Background

The use of big data analytics is underutilised in the domain of weather data, better big data weather forecasting could benefit the construction industry by helping build more efficient buildings and provide an “*improved quality of life”* (H. Jain & Jain, 2017)*.* The knowledge-discovery (KD) process (Begoli & Horey, 2012) offers the potential to employ big data analytics to support decision making in the design and construction industry with the aim of improving comfort of building occupants and reducing the need for energy. KD requires integration of three elements; data management, data mining and domain expertise.

Data management concepts related to big data and climate, include the use of workflow management systems for big data science applications (Buyya et al., 2016) and how Infrastructure as a Service (IaaS) offered by cloud service providers is an applicable service model (F. Liu et al., 2011). Apache Spark’s machine learning library provides functional tools applied by existing applications that implement climate data processing tasks in a big data analytic context report.

In the context of climate, data mining requires working with spatiotemporal (ST) data, which contrasts with classical data mining (Faghmous & Kumar, 2014). ST data mining involves specific methodologies each of which imply a set of unique challenges. Clustering, an unsupervised machine learning technique has proved advantageous in studies aiming to classify climate patterns (Fovell & Fovell, 1993). Both hierarchical clustering (HC) and non-hierachical clustering (NHC) methods have proven utility in identifying patterns in climate and weather data.

In the context of design and construction, domain expertise involves the concept of human comfort, a state where given a set of combined environmental conditions minimal extra energy is required to maintain the human balance (Manzano-Agugliaro, Montoya, Sabio-Ortega, & García-Cruz, 2015). A psychrometric chart is a graphical tool that represents environmental conditions relating to comfort (Givoni, 1992). Designers can plot a zone of human comfort on the chart, and around this, the designer can represent zones of different low-energy environmental design strategies. These strategies describe instrumental methods that the design can use to adjust environmental conditions to improve comfort (Manzano-Agugliaro *et al.,* 2015), strategies that require minimal energy input are prioritised. The aim of integrating the three elements of KD is to automatically link these strategies with weather patterns found with data analytics.

## 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 required to maintain the human balance (Manzano-Agugliaro et al., 2015).

Psychrometric charts map interrelationships of thermal conditions of the environment (Figure 2). Standard guidelines (ASHRAE, 2013) define how to plot a zone of human comfort. When a designer plots weather data on the chart, they can identify periods when a proposed building does not provide sufficient thermal comfort and understand how and when they need to adapt the design.

Givoni (1992) defined the psychrometric chart as the building bioclimatic chart and this was adapted by Manzano-Agugliaro *et al.* (2015) to include specific zones representing strategies that can be applied to a design to extend the zone of comfort. Manzano-Agugliaro *et al*. recommend conventional heating and air conditioning only at extremes. The strategies include what Lechner (2014, 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%.

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.

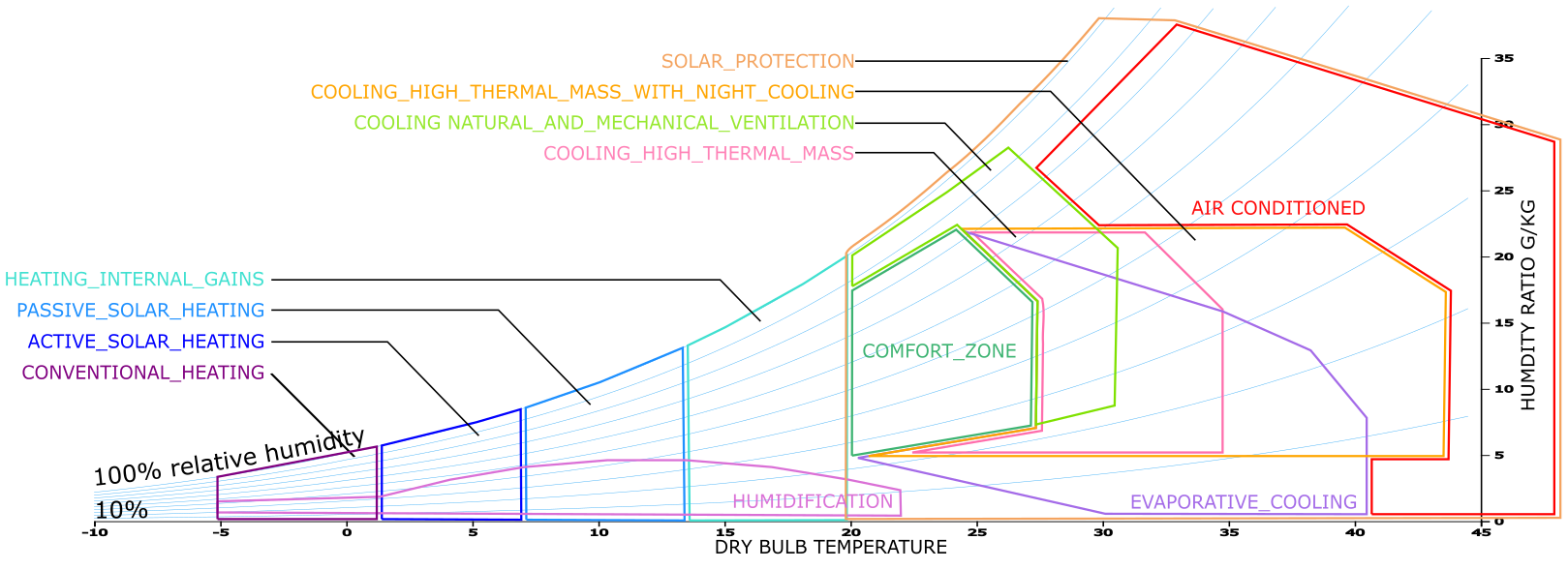


Figure 2 Adapted version of the psychrometric chart (Manzano-Agugliaro *et al.,* 2015)

#### 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 IDEAMCI assumes that when the human body experiences cooling sensations when surrounded by air temperatures lower than 36.5C and wind constantly renews the air. 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 spatiotemporal understanding of comfort at an hourly resolution across Colombia. More 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 & Höppe, 2017).

Indices exist to help those not familiar with the domains of thermo-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, which did not include practitioners from the built environment. Indices are useful during later design stages to fine tune heating, cooling and ventilation systems or to refine material specifications (insulation and glazing) ensuring certain thermal comfort levels. However, 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. Early stage design decisions are more probable to influence the cost and function of the finished building and are cheaper to implement (CURT, 2004, p4). Despite their shortcomings, 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, & 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 & Kumar, 2014) . ST data is heterogeneous, 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 et al., 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. *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 using empirical orthogonal functions and ST clustering.

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

* Self-organising maps (SOM)’s have been used to extract features from data (Y. Liu, Weisberg, & Mooers, 2006) and applied to metrology and oceanography (Y. Liu & Weisberg, 2011)(Y. Liu & Weisberg, 2005).
* Delta-maps (Fountalis, Bracco, & Dovrolis, 2014) (Bracco, Falasca, Nenes, Fountalis, & Dovrolis, 2017) group nodes in a network according to homogeneity, these have beenapplied to precipitation and sea surface temperatures. Link maps, area strength and s-core decomposition are used to evaluate the robustness of networks generated by delta-maps.

### 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 & 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.

### Cloud computing

Cloud computing is defined by NIST (Mell & Grance, 2011) as a model consisting of the following characteristics; on demand service, broad network access, rapid elasticity and measurable service. Cloud providers offer three key service models; 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 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 (F. Liu et al., 2011).

Amazon Web Services (AWS) is a cloud provider offering all the different cloud service models. The AWS 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) to provide virtual cloud-based servers. EMR launches clusters of virtual machines on a Virtual Private Cloud (VPC). A Step on AWS is a distinct work unit that can run on a cluster and describes an analytic job. A single cluster can have several Steps. EMR handles 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 AWS’s Simple Storage Service (S3). S3 is an object storage service providing scalability and automatic data replication by distributed data across a minimum of 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. A workflow is 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 consisting 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 3) (Rodriguez & Buyya, 2017).

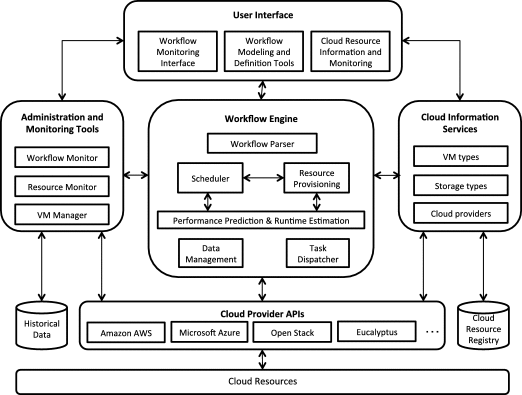


Figure 3 Reference architecture of a WMS

#### 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 used to develop application architecture using domain-driven design. Features and architecture components can be selected from the model based on design rules determined by the domain (Figure 4).

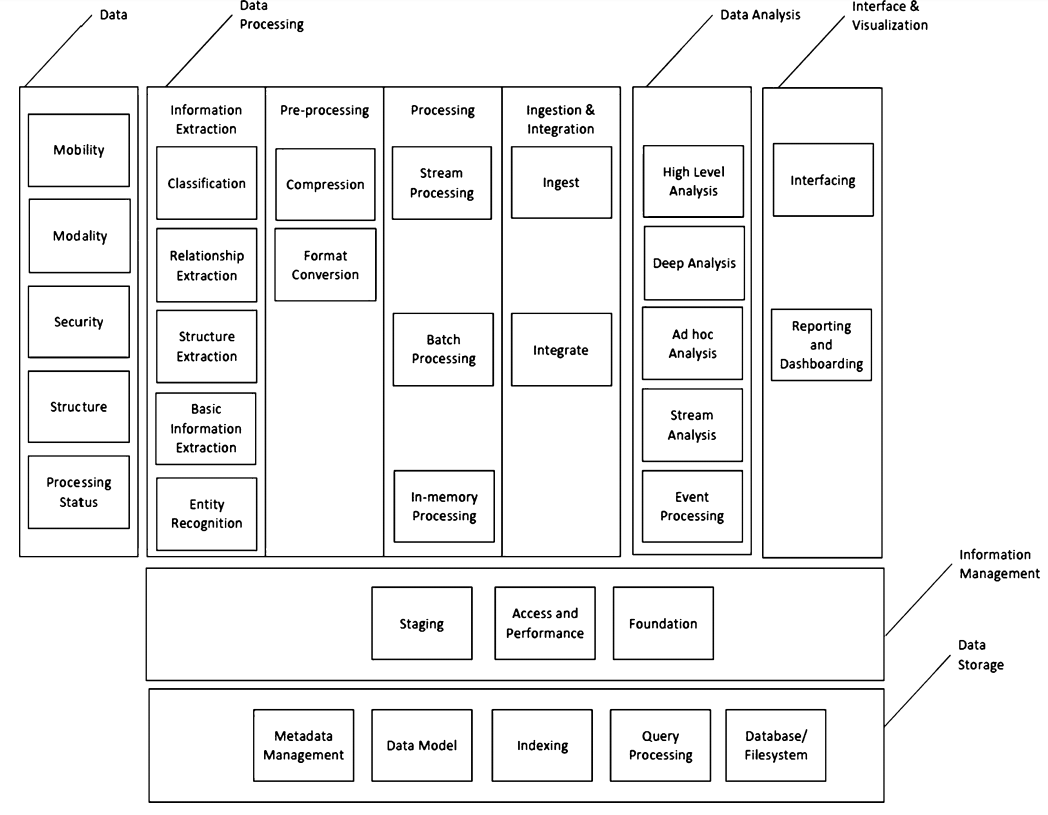


Figure 4 Big data reference architecture (Avci Salma, Tekinerdogan and Athanasiadis, 2017)

#### Big data and Weather Data

Several precedents exist describing the application of big data tools to process and undertake simple analysis on climate and weather 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 & Sumathi, 2017).
* Hadoop and Spark were used to find points with similar weather conditions using Euclidean distances based on weather attributes using the US National Oceanic and Atmospheric Administration (NOAA) hourly land based data set (Rodenburg & Maria Fiore, 2017).
* MapReduce was compared to Spark to calculate minimum, maximum and average values of weather parameters using data from NOAA (Chouksey & 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, Lagali, Avadhani, & Kalekar, 2015) to compare the performance of Pig and Hive to average data for each station for a single variable.
* MapReduce was used to build an analytical engine for processing National Climatic Data Center (NCDC) temperature data from automated sensors (Mariam Varghese, 2015).
* K-means with MapReduce was used to cluster weather data from China (Fang, Sheng, Wen, & Pan, 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 and found to reduce processing time compared to a serial method (Jayaratne, Alahakoon, Silva, & Yu, 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, Sadiq, Ouarzazi, & Sadgal, 2015).

#### Apache Spark

Apache Spark is a distributed processing system and unified analytics engine designed for processing large datasets in-memory. Spark is designed to be fast and general purpose and integrates with Hadoop processing clusters. Spark is accessible via application programming interfaces (API) in Python, Java, Scala and SQL and offers a set of built-in libraries (Zaharia, Wendell, Konwinski, & Karau, 2015, p1). The core element of the Spark stack (Figure 5) is an engine that schedules, distributes and monitors applications defined by multiple computational tasks across many worker nodes within a computing cluster. Integrated with the Spark Core are libraries for processing structured data with SQL, streaming real-time data, machine learning and graph processing.

A screenshot of a cell phone

Description generated with very high confidence

Figure 5 Spark unified stack (Zaharia et al., 2015)

Spark’s Machine Learning Library MLlib is of interest for this dissertation as it offers several different clustering methods including k-means and bisecting k-means. Both methods provide access to a clustering “cost” the *within set sum of squared errors* (WSSSE) (sum of squared distances of points to their nearest centre) (“KMeansModel (Spark 2.3.2 JavaDoc),” n.d.). Luna-Romera *et al.* (2016) suggest that Spark’s WSSSE does not capture cluster consistency or distance between clusters and describe Spark based implementations of the Dunn index and Silhouette index as alternatives.

To run a Spark process on a cluster of distributed nodes a SparkContext object defined in the driver program coordinates a set of independent processes (Figure 6). Spark uses one of a series of cluster mangers to allocate resources across applications (Spark jobs on AWS’s EMR use Yarn as the cluster manager). The cluster manager defines and connects the cluster nodes and creates executors on each, each executor can run application processes and store data. Once the nodes are running SparkContext acquires the executors, based on the application code SparkContext sends tasks to run on executors.

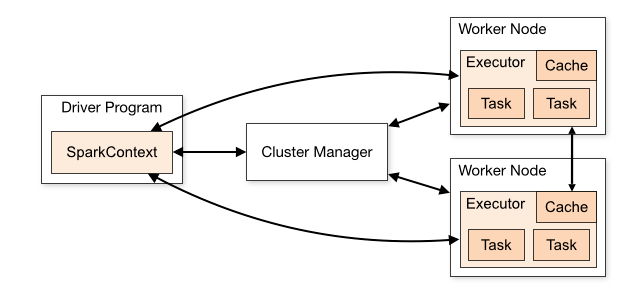


Figure 6 Spark cluster components (“Cluster Mode Overview - Spark 2.4.0 Documentation,” n.d.)

Spark Core includes the API for Resilient Distributed Datasets (RDDs) a parallel programming abstraction that keeps data in-memory and tracks a graph of operations used to produce each RDD, enabling recovery without replication. Using RDDs Spark structures the computation as a sequence of stateless, batch processes issued at short time intervals. Spark handles faults using parallel recovery whereby on failure of a node all other nodes in the cluster work to rebuild the lost RDDs. In contrast other distributed systems rely on data replication for fault recovery which is costly in terms of time and hardware, and can result in long recovery times and problems for handling slow compute nodes (Zaharia et al., 2013).

RDDs can be defined by parallelising an existing collection of data in the driver program or by referencing a dataset stored in an external system such as AWS’s S3 (“RDD Programming Guide - Spark 2.4.0 Documentation,” n.d.). RDDs support transformation operations that return new RDDs and action operations that return a value to the driver program. The following code snippet shows how an RDD is created from an external file. A transformation operation “map” is passed as a lambda expression and returns a new RDD. Lastly, by passing the operation action “reduce” as a lambda expression a single value is returned to the driver program.

JavaRDD<String> lines = sc.textFile("data.txt");

JavaRDD<Integer> lineLengths = lines.map(s -> s.length());

int totalLength = lineLengths.reduce((a, b) -> a + b);

The driver program implements operations on the RDDs by passing functions that Spark runs on nodes within the cluster. When programming with Java, Spark offers two ways to create functions, first using lambda expressions (as in the code snippet above) or using Spark’s Function interfaces, programmers can create user defined classes.

Spark’s Java API allows user to define function classes as anonymous inner classes inline:

RDD<String> errors = lines.filter(new Function<String, Boolean>() {

public Boolean call(String x) { return x.contains("error"); }

});

Alternatively, users can define a named function class and then pass it to the RDD:

class ContainsError implements Function<String, Boolean>() {

public Boolean call(String x) { return x.contains("error"); }

}

RDD<String> errors = lines.filter(new ContainsError());

#### Knowledge-discovery

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

## Theory

The three integrated elements that define the concept of KD provide a concise theoretical framework for defining goals of the dissertation.

### Data:

Key considerations involving the data include the growing quantities of climate data and its spatiotemporal nature. Large amounts of this form of data are available and the sources are growing which suggests the logic in seeking a solution that is scalable in terms of data capacity. AWS’s IaaS provides flexible, storage for data in S3, and is included in AWS’s free usage tier and has a well-documented Java SDK. Understanding the spatiotemporal nature of climate data is important for manipulating and organising climate data especially in the variety of ways patterns may exist.

### Analytics: Clustering applied to climate classification

Recent research suggests that clustering, an unsupervised machine learning technique, is particularly applicable to climate analytics, classification and data mining. Studies focus on common clustering themes; seeking best performing clustering methods for a specific goal, comparisons between hierarchical and non-hierarchical clustering methods, selecting the number of clusters, pre-processing variable prior to clustering and developing hybrid workflows from different clustering methods. The methods described in the following sections work with vectors, each a set of *n* features. Within this n-dimensional parameter-space. Clustering methods calculated Euclidian distances between vectors and cluster centroids to define determine proximity and define which vector belongs to which cluster.

#### Hierarchical clustering

Hierarchical clustering (HC) methods organise data into a sequence of nested groups. HC begins by treating all observations of vectors as individual clusters. HC repeats two steps; first identify the two closest clusters and then merge these into a single cluster (Figure 7). Fovell & Fovell (1993) studied HC methods to define climate zones in the US and found average linkage method performed best to minimise bias in terms of method, latent and information.

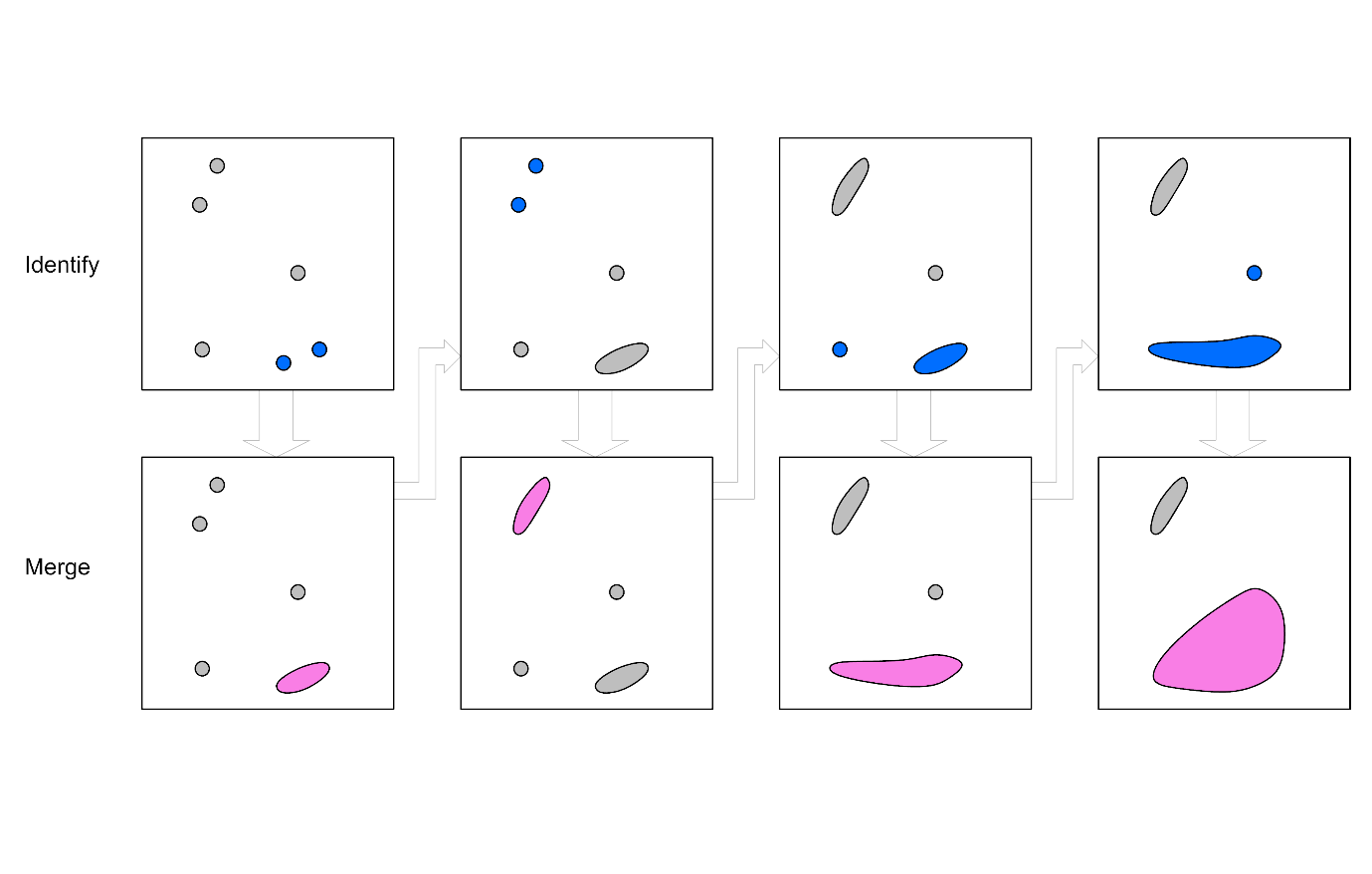


Figure 7 Hierarchical clustering steps

#### Non-hierarchical clustering

Non-hierarchical clustering (NHC) methods provide a single partitioning of the data aiming to define natural groups within the data (Jain and Dubes, 1988, p89). K-means is a NHC method that was defined in 1955 (A. K. Jain, 2010), despite its age and alternatives it remains popular in the literature of clustering with climate data. The k-means algorithm requires *K,* the number of clusters and begins by creating *k* initial cluster centroids. K-means repeats two steps; assigning vectors to a cluster with the closest centroid, redefining cluster centroids based on the data points in the cluster (Figure 8).

Degaetano (1996) compared K-means NHC to Ward’s HC technique in a study that sought to develop an ecosystem management and planning guide by defining mesoscale climate zones in the north-eastern US. The study showed k-means improved the spatial distribution, homogeneity of clusters and produced stable clusters with minimal information bias.

Climatic influences on water resources and food security in the Himalayas were assessed using clustering (Forsythe, Blenkinsop, & Fowler, 2015). The study showed clustering could can 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.

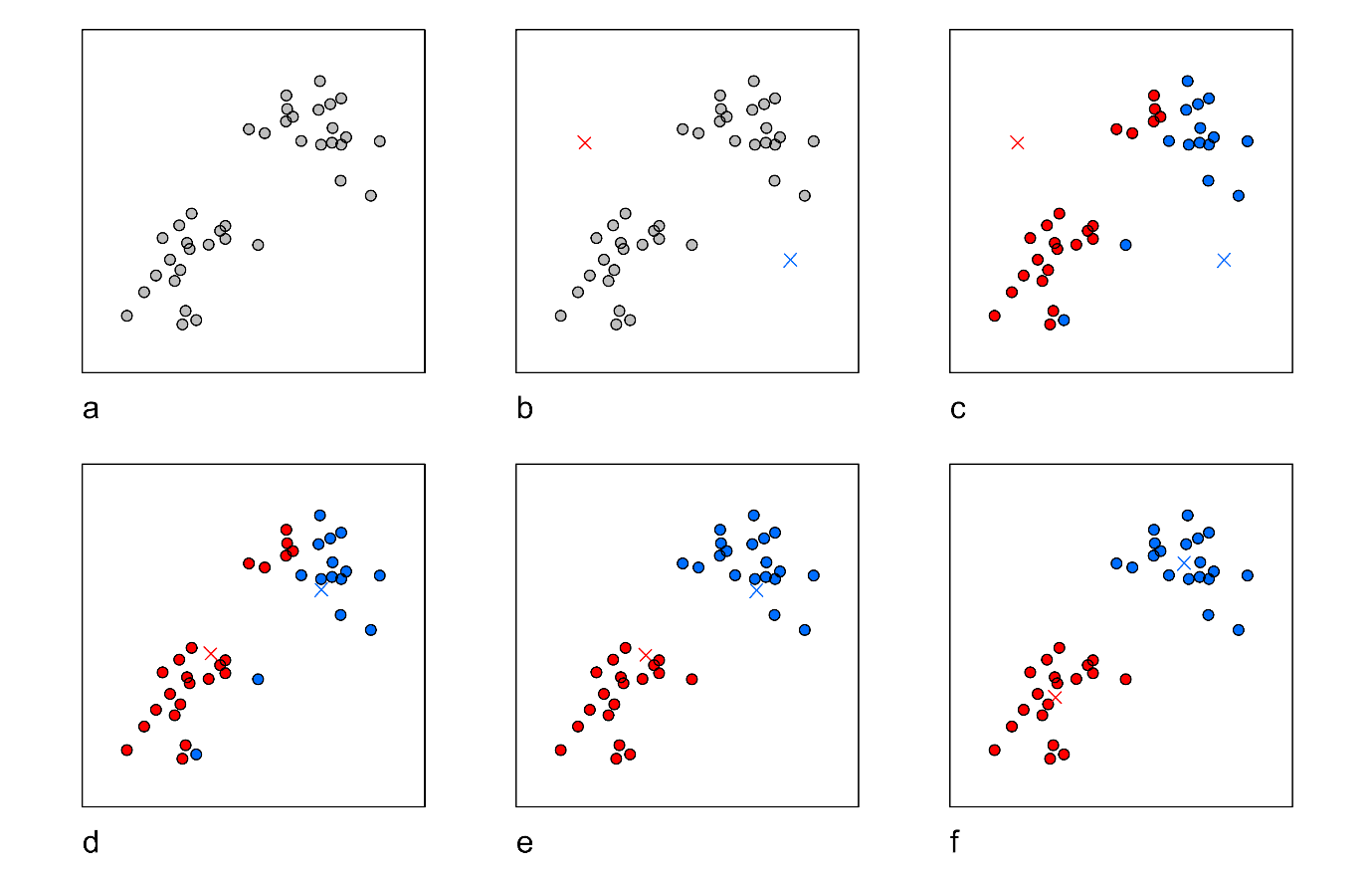


Figure 8 Non-hierarchical k-means steps: (a) Original data. (b) Random initial centroids. (c-f) Two iterations of: assign data points to closest cluster, recalculate cluster centroid. Adapted from Piech (2013).

Zscheischler, Mahecha and Harmeling (2012) used k-means clustering with subsets of five normalized variables. Combining k-means with climate and vegetation variables generated clusters like the Köppen Geiger Climate Classification (KGCC) zones.

Netzel and Stepinski (2016) undertook 32 different clustering experiments comprised of HC Ward linkage or NHC k-means with different input variables and similarity measures and compared the results to KGCC. The study concluded that clustering could find 50% of the climate types defined by the KGCC. The remaining classes were more homogeneous and more distinct than KGCC types but differed in climatic character and spatial distribution. Conclusions from the study indicate that NHC gave better results than the HC.

#### Hybrid approach

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 remote sensors and spatially distributed data. This method integrated k-means within a more complex workflow (hierarchical clustering followed by non-hierarchical then decision trees, trained on the results classified remotely sensed data).

#### Choosing the numbers of clusters

A recurring theme in the literature is how to choose the number of clusters, *k*. Many indices exist to describe the homogeneity and completeness of clustering solutions (Arbelaitz, Gurrutxaga, Muguerza, Pérez, & Perona, 2013), but Fovell and Fovell (1993) insist informed decisions over *k* must also include domain knowledge. Increasing *k* serves to become more specific but at the cost of generality. Lower cluster numbers represent a loss of detail but, they can enhance interpretation and generality. Climate data varies smoothly and hard edges between clusters do not exist, the choice is partly subjective and based on an adequate subdivision.

Silhouette index represents dissimilarity (Wiwie, Baumbach, & Röttger, 2015) and is found to produce good results in comparison to other indices (Arbelaitz et al., 2013). Silhouette is the ratio between the difference of the inter-mean (average of distances from each cluster centroid to the global centre) and the intra-mean (the average distance between each vector and its cluster’s centroid) and the maximum of the inter or intra means.

Dunn index defines quality of clustering as a ratio of cohesion (the maximum cluster diameter) and separation (the distance to the nearest neighbour). Luna-Romera *et al.*, (2016) suggest the first maximum found with the Dunn index indicates the optimal number of clusters. Higher values in both Silhouette and Dunn indices indicate better performance.

Sum of Squared Errors (SSE), a measure of internal clustering cohesion (Thinsungnoen, Kaoungku, Durongdumronchai, Kerdprasop, & Kerdprasop, 2015), is used for validity and is calculated using the square of the Euclidian distance between each vector and its cluster centroid. TheElbow method can find a close-to-optimal value of *k* by plotting SSE against the *k.* The inflexion (elbow) in the graph indicates a good candidate for *k* (Kodinariya and Makwana, 2013 and Nikolaou *et al.*, 2012)

#### Pre-processing and choosing the variables

Netzel and Stepinski (2016) and Zscheischler, Mahecha and Harmeling (2012) are clear that normalisation of variables is essential to remove the effects from different scales. Principle component analysis (PCA) is thought to be important in identifying and removing highly corelated variables (Fovell & Fovell, 1993), but simultaneously questioned. PCA required subjective decisions to define where truncation of the features should occur and this can lead to information bias. Rhee *et al.* (2008) chose to avoid of truncation entirely and not use PCA fearing loss of information in their monthly time series data.

Domain knowledge also needs to factor in selection of variables. Forsythe, Blenkinsop and Fowler (2015) suggest the usefulness of classification is dependent on the extents that it reflects the constraints that determine the physical processes of interest. This indicates that understanding the domain and existing problem approaches to the problem can provide valuable insight into the selection of variables.

The literature suggests that clustering is a key machine learning technique for datamining spatiotemporal data. Spark emerges as a main contender for undertaking this kind of climate data analysis, it offers NHC k-means and HC bisecting k-means as built in functions. Spark provides methods for evaluating the clustering performance and further validation metrics with Spark implementations were identified in the literature. EMR, AWS’s analytics IaaS that provides hosting of virtual machines for distributed processing includes Spark. EMR is part of AWS’s free usage tier and has a well-documented Java SDK.

### Expertise:

The domain includes human biology, its relation to climate and the construction of buildings. Understanding domain goals and concerns will provide insight into the types of analytic processes and required variables. The concept of the workflow management system should be tailored to the domain but integrate with the chosen big data services. Understanding the relationship of low-energy design methods and construction techniques to the climate data is crucial for analytics methodology. The connection between environmental conditions and design strategies, which aim to achieve human comfort in energy efficient buildings, is dependent on how strategies are represented and manipulated.

## Terms

Key terminology defined in the literature review, and used throughout the dissertation are summarised in the list below:

**Apache Spark:** a distributed processing system and unified analytics engine designed for processing large datasets in-memory.

**AWS:** Amazon Web Services, a cloud service provider.

**Clustering:** an unsupervised machine learning technique.

**Comfort Index:** value expressing the suitability of the environment for physical activity.

**Dunn Index:** defines quality of a clustering solution as a ratio of cohesion to separation, high values are better.

**EC2:** Elastic Compute Cloud. AWS’s service providing virtual cloud-based servers.

**EMR:** Elastic Map Reduce, AWS’s hosted Hadoop framework.

**HC:** Hierachical clustering.

**IDEAM:** Instituto de hidrología, meteorología y estudios ambientales, Colombia’s meteorological office.

**IDEAMCI:** Comfort Index defined by Colombia’s meteorological office IDEAM.

**KD:** Knowledge-discovery, the integration of data mining, domain theory and data management.

**Low-energy design strategy:** a description of an instrumental approach to improving thermal comfort in a building given a set of environmental conditions, methods.

**MLlib:** Spark’s Machine Learning Library.

**NHC:** Non-hierarchical clustering.

**Psychrometric chart:** a graphical system for mapping the thermal conditions of the environment.

**RDD:** Resilient Distributed Datasets, a collection of records spread over one or more partitions.

**S3:** Simple Storage Service, AWS’s object storage service.

**Silhouette index:** a value in the range zero to one,representing dissimilarity in a clustering solution, high values are better.

**ST:** Spatiotemporal, a term used to characterise data with dimensions in both space and time.

**UML:** Unified Modelling Language, a general-purpose language for visualising system design.

**UTCI:** Universal Comfort Index, defined by the International Society of Biometeorology Commission to have wide ranging utility and application.

**Workflow management system:**  An approach for supporting scientific data analysis on large data sets in the cloud.

**Workflow:** a series of linked tasks.

**WSSSE:** within set sum of squared errors, used to express internal cost of a clustering solution, low values are better.

# Analysis and design

## 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 specific design strategies. The following two sentences describe a typical analysis / data mining problem:

*Using k-means clustering what design strategies are suitable for afternoon weather conditions during the first three months of the year in the coastal regions of Northern Colombia. 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 are developed as a proof-of-concept using a private 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 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 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.

### 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 similar to 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.

The general use case divides into four phases:

### Define workflow

To define a process or workflow (Figure 9) the ED must be able to specify a dataset or collection of datasets selected from a set of preloaded 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 specify that analysis is recurrent 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 ED has defined the dataset and spatial and temporal subset and the dimensionality specified, the ED will select an analytic or data mining technique from a set of predefined methods. During the process, the ED may need to save, save as and edit the defined workflow.

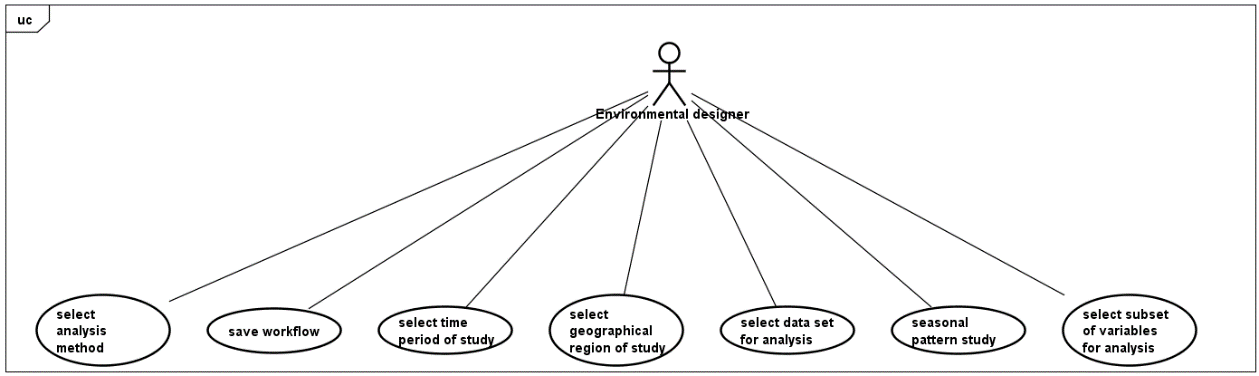


Figure 9 Define workflow

### Run workflow + monitor resources

Once the ED has defined the workflow he can submitted it for processing and monitor its progress in terms of its status (ready, executing, staging, completed) (Figure 10). The ED may also need to monitor the state of processing resources. During processing, the ED should be able to stop or cancel the workflow.

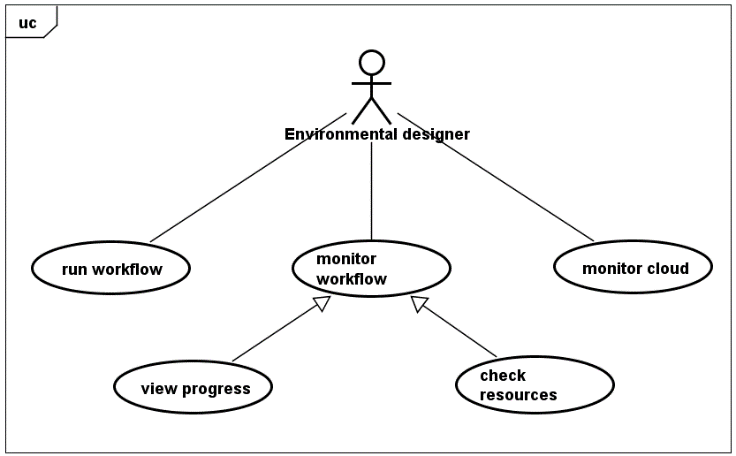


Figure 10 Run and monitor workflow

### Output + visualise results

Following analysis, the system should store results from the workflow and make them accessible to the ED (Figure 11). The ED needs to generate numerical and statistical summaries of the results using one of a range of predefined methods. The system should store these results and make them 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. High-quality images or vector graphic versions of the visualisation should be available for download and use in reports. 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, the graphics should be available for viewing and download from a webpage.

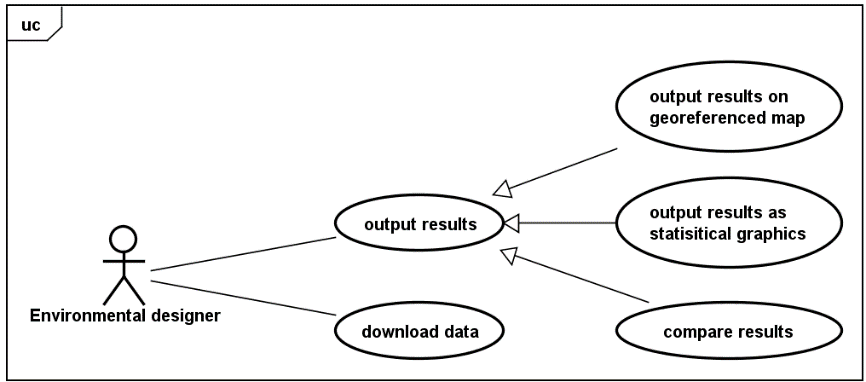


Figure 11 Output and visualise results

### Manage design strategies

The design strategies are specific to individual ED’s, it is necessary, therefore, to manage the predefined general design strategies used by the system (Figure 12). Some EDs will wish to add new strategies, removing unwanted strategies and editing existing ones. The system requires a standard unambiguous method for describing a strategy.

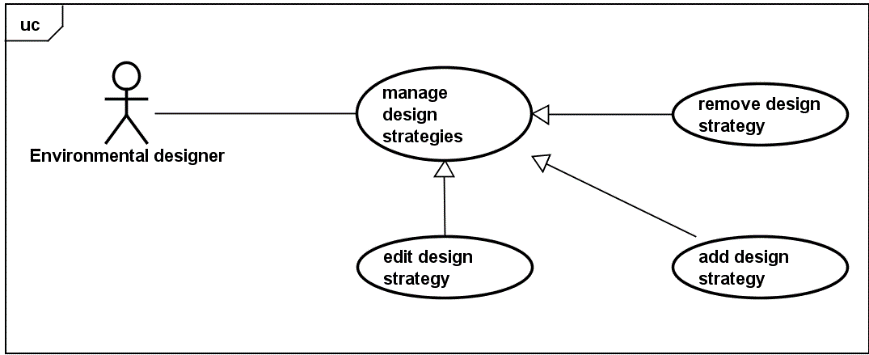


Figure 12 Manage design strategies

## General requirements

The following sequence diagrams, show how the use cases described in section 3.1 are implemented in the final prototype.

### Sequence diagram for defining and running a workflow

Figure 13 shows the interactions that take place when the ED defines and runs a workflow. The GUI workflow builder class creates a new instance of workflow which defines default variables and creates an analysis parameters instance. The cluster coordinator class stores the new workflow object and the user can configure the parameters. When the ED submits the workflow for analysis the cluster coordinator creates, stores and assigns the workflow to a new cluster object. The cluster coordinator then submits the cluster and workflow to Amazon’s Elastic MapReduce (EMR).

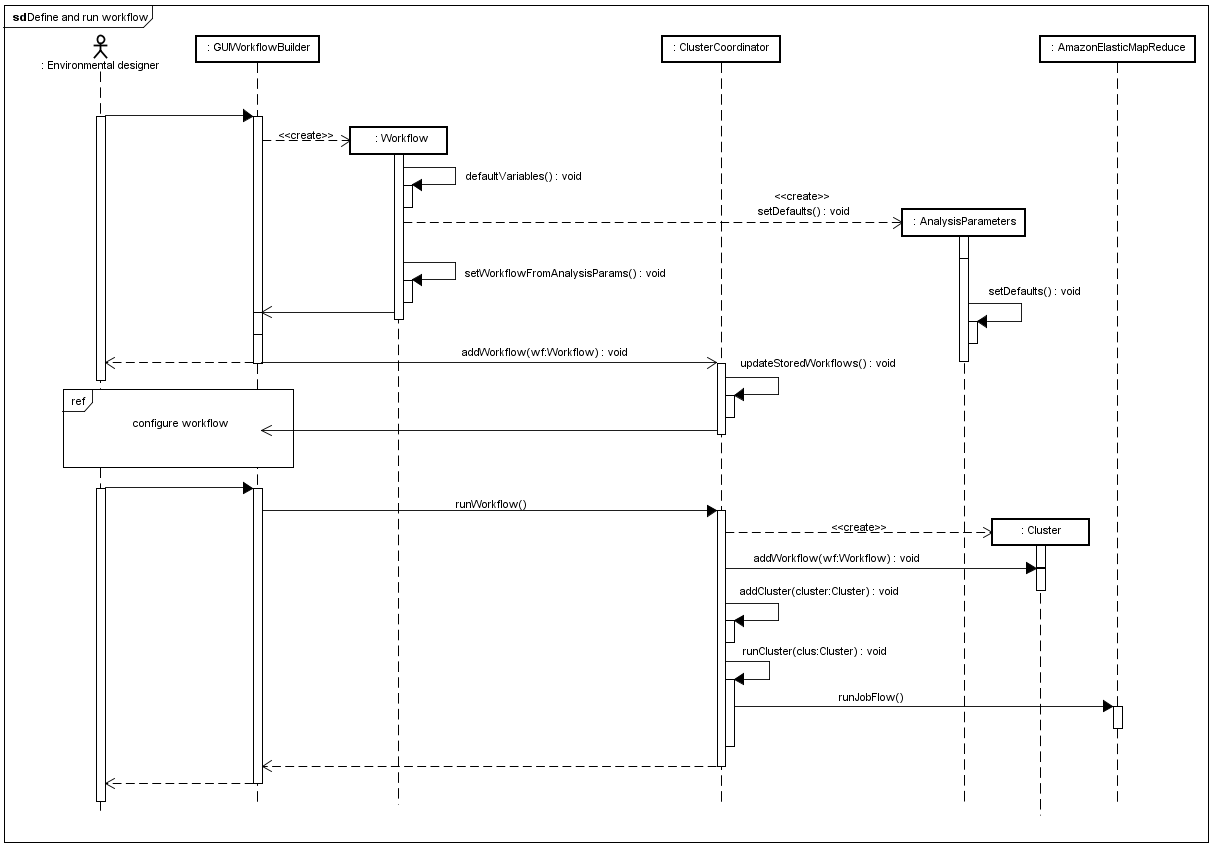


Figure 13 Sequence diagram for defining and running a workflow

### Sequence diagram for monitoring workflows and resources

Figure 14 describes the interchange of messages to monitor the status of the cloud-based clusters and workflows. The sequence runs at sixty second intervals, via the data manager the cluster coordinator retrieves a list of all workflows stored on Amazon’s S3 and creates an instance of the workflow class to store new workflows locally. Within a user defined period, the cluster coordinator retrieves clusters from EMR. For each EMR cluster the cluster coordinator updates (or creates) an instance of the cluster class stored locally and displays its status in the GUI. The cluster coordinator retrieves the associated steps (each EMR step corresponds with a workflow) on each cluster and updates the status of each locally stored workflow to match the step status and displayed in the GUI.

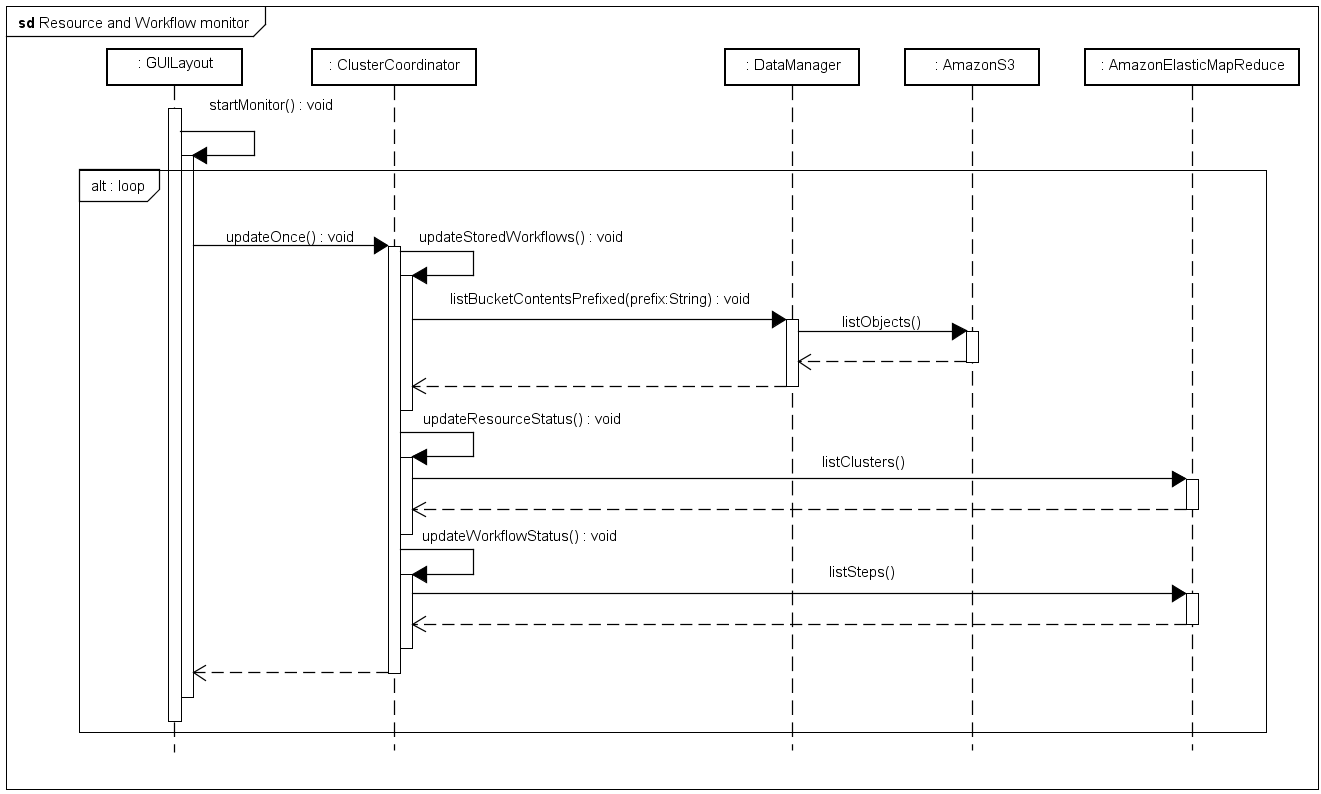


Figure 14 Sequence diagram for monitoring workflows and resources.

### Sequence diagrams for visualising results

Figure 15 illustrates the interactions that occur when the ED clicks the dashboard button on the GUI. The GUI workflow builder class checks the workflow to see if a dashboard exists, if it does, the URL from the analysis parameters is opened in the default browser and loaded into the GUI. If no dashboard exists, the workflow builder class initiates the process of creating one, saving the workflows and creating an instance of the geo visualisation class (within the interaction fragment create dashboard Figure 16).

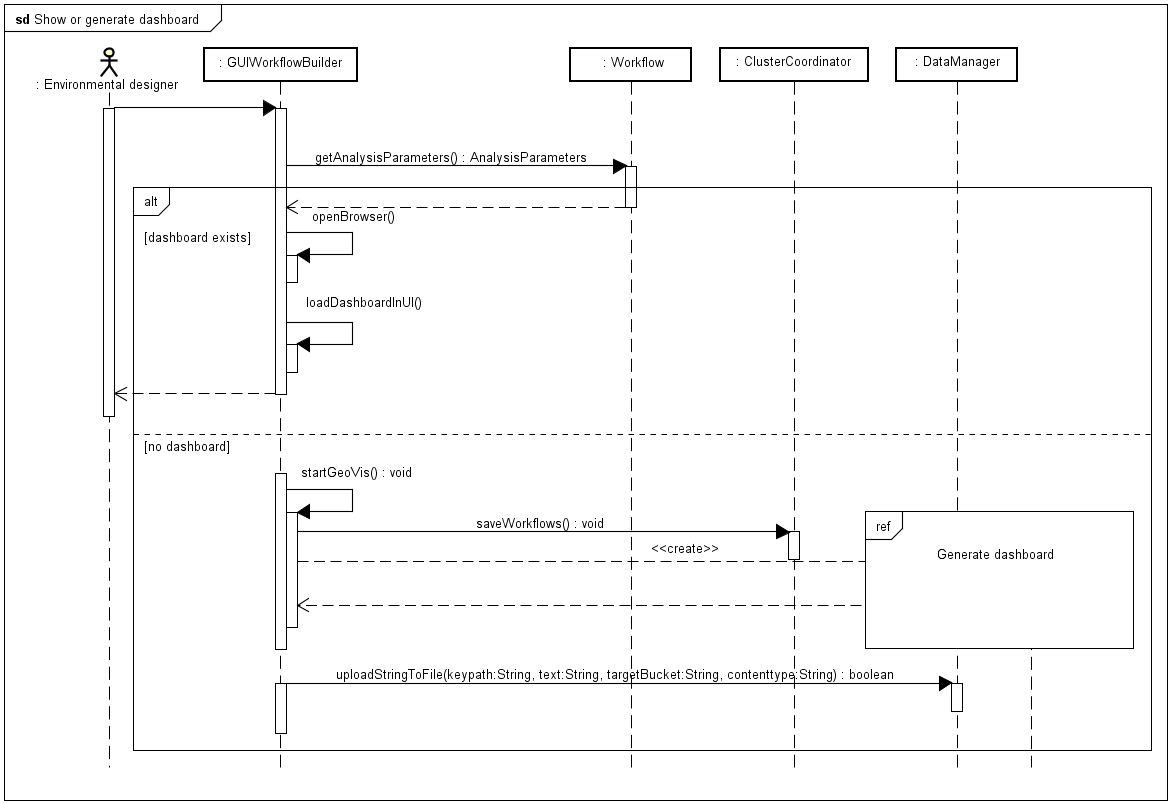


Figure 15 Sequence diagram for showing or generating a dashboard.

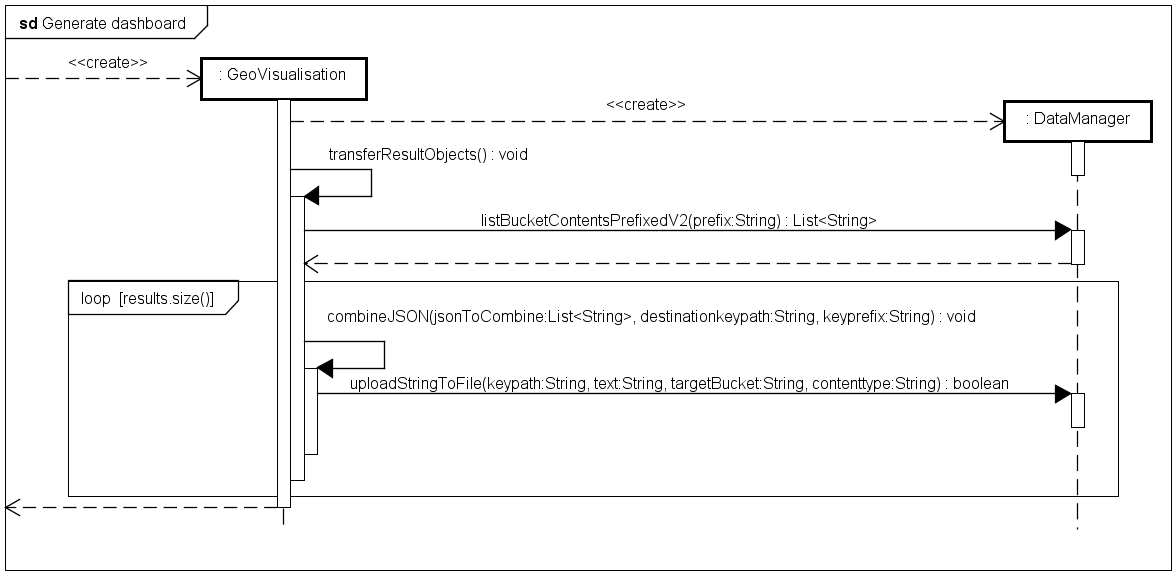


Figure 16 Sequence diagram for generate dashboard also see Figure 15.

This interaction consolidates and transfers output from the analysis and copies the workflow parameters and a landing page for the dashboard to a public folder. When the process completes the workflow’s dashboard URL property is updated with the new URL of the landing page.

## General System Architecture

Figure 17 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 beyond the scope of the project, however local prototype application development requires AWS Credentials to access the AWS API. The development machine stores credentials locally and the IDE automatically instantiates the security settings using the AWS SDK toolkit. Figure 18 decomposes the overall architecture and key elements into lower level subcomponents.

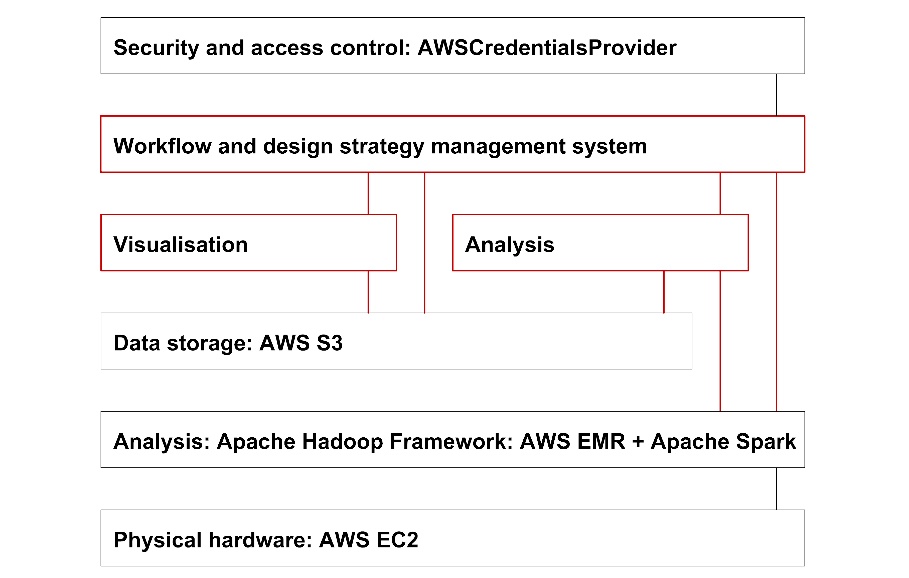


Figure 17 High-level system architecture based around IaaS provided by AWS

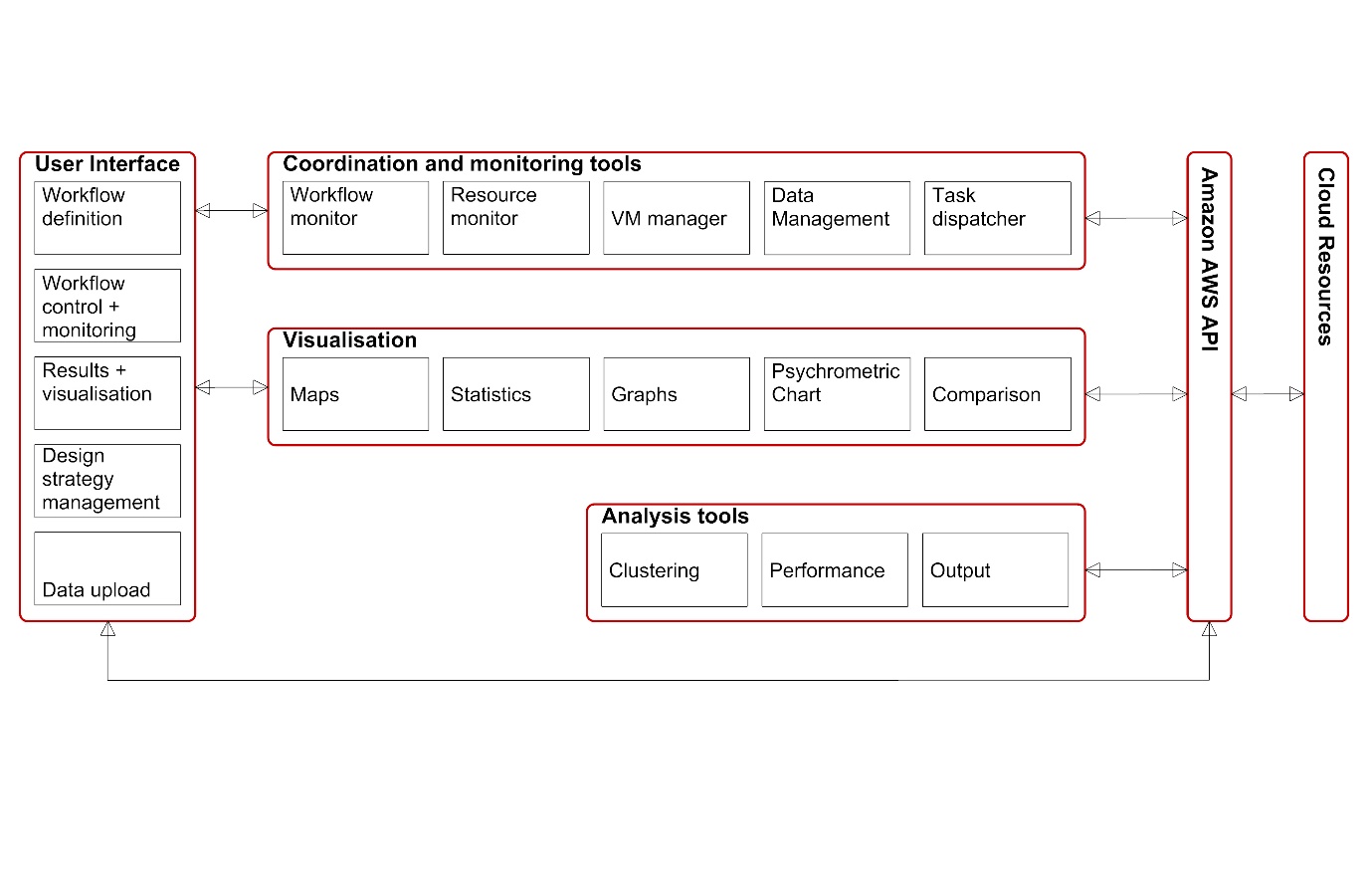


Figure 18 Architecture for the user interface, coordination, visualisation and analysis elements, including subcomponents

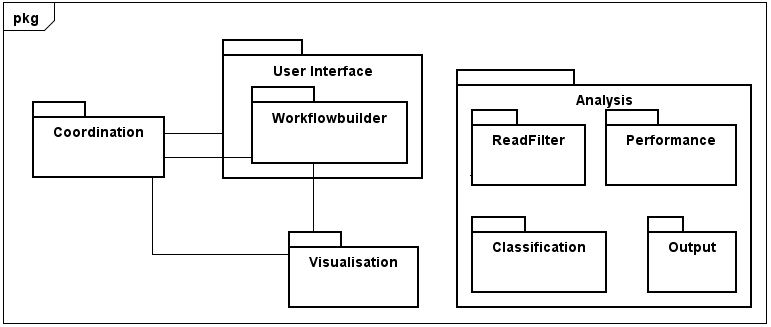


Figure 19 Simplified package diagram.

Key architectural elements define the proposed system packages (Figure 19). The sub-package within the user interface package, Workflowbuilder, contains domain specific components for climate-driven building design and is an interchangeable element for when the system is applied to another domain.

## Design methodology

The design process was broken into a series of major development increments (Figure 20). 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. A series of construction iterations followed, each began with a planning phase. Each iteration prioritised requirements and the highest priority was implemented first. Over several hours, the author produced UML models to explore what should be built for the iteration and to estimate the time required. Just-in-time models, created in less than thirty minutes involving hand-sketched flow diagrams, sequence diagrams and class diagrams were used to address issues identified in the planning models in more detail. Over the following hours or days code was written referencing the modelled details and using a test-first and refactor approach.

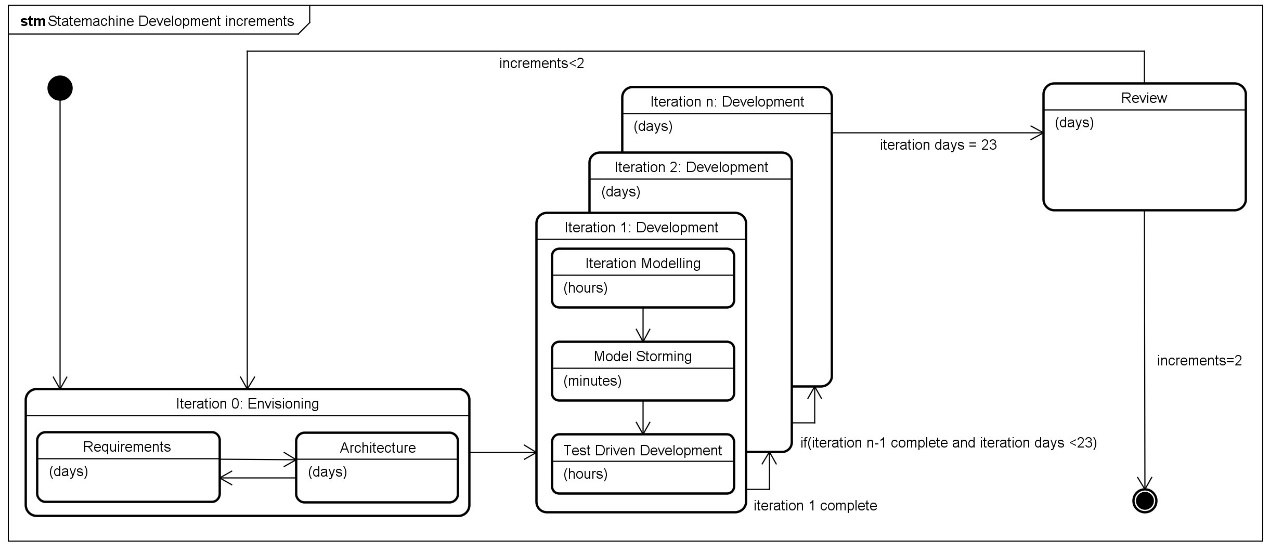


Figure 20 Development increments (extended from Ambler (2004), p119).

## 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. Identified key challenges across all packages 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. Java Swing was used to create a rudimentary user interface. Using the UI, predefined analysis routines and cluster configurations were 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. Restructuring the dashboard framework in the visualisation package 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.

# Implementation

## Environment

The prototype operates locally on a Java Virtual Machine (JVM) and analytics take place within AWS Cloud using the EMR framework running spark-core\_2.30. The user configures the EMR Hardware on 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,” n.d.). Astah Community was used to develop UML models, some additional diagrams were modelled in Rhino3d. JavaScript, html and CSS were used to create the dashboard web framework that 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 for AWS cloud infrastructure

The application uses IaaS on AWS cloud infrastructure (Figure 21). Users access the local application (with AWS login credentials) and can launch resources within Amazon’s VPC. A graphical interface enables users to create workflows (analytic jobs) and configure the hardware for computing them. Amazons’ EMR launches analytic jobs as Hadoop distributed file system (HDFS) clusters, each running Apache Spark. System users can configure and launch multiple clusters. Clusters access input data and store results from the analytic jobs in private buckets within AWS S3. When analytics on a cluster completes, the local application transforms output files and stores them in a publicly accessible bucket. The bucket includes 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.

The system implements four key packages (Figure 22 and Figure 23). The User Interface package provides functionality that presents the status of cloud resources and running analysis jobs. From the UI, users can upload new analysis routines (as .jar packages) and data sets. The UI also provides access to results via an embedded web engine and a mocked-up editor for the design strategies.

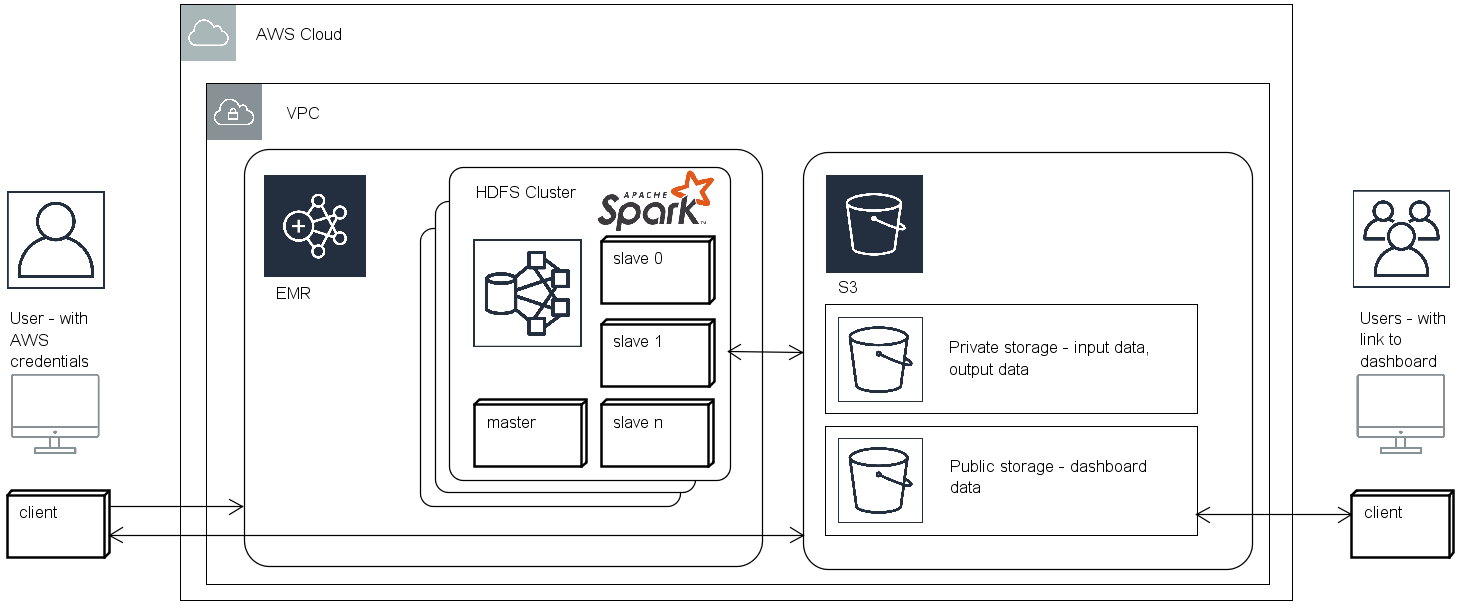


Figure 21 Deployment on AWS

## Packages

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 and allows defining spatial zones for analysis and specifying a series of parameters concerning the data, analysis method and temporal scales (Figure 22). The coordination package (Figure 22) 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.

The Analysis package (Figure 23) 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. The ReadFilter sub-package includes classes to parse and filter the analysis parameters defined via the user interface. The Performance sub-package computes the performance of clustering solutions using a series of standard indices and can define the number of clusters using optimisation of one index. The Classification sub-package implements two clustering methods from Spark’s Machine Learning library MLLib (“MLlib: Main Guide - Spark 2.3.2 Documentation,” n.d.) K-Means clustering and Bisecting K-means clustering and a third method that is a hybrid of the previous two. The ClusterCoordinator class submits data vectors to the Classification sub-package, which constructs clustering models. The models classify each vector and using the ThermalZones class applicable design strategies are assigned. Finally, the Output sub-package organises the results and the meta-data related to the performance of the process and writes these in a logical storage structure to S3.

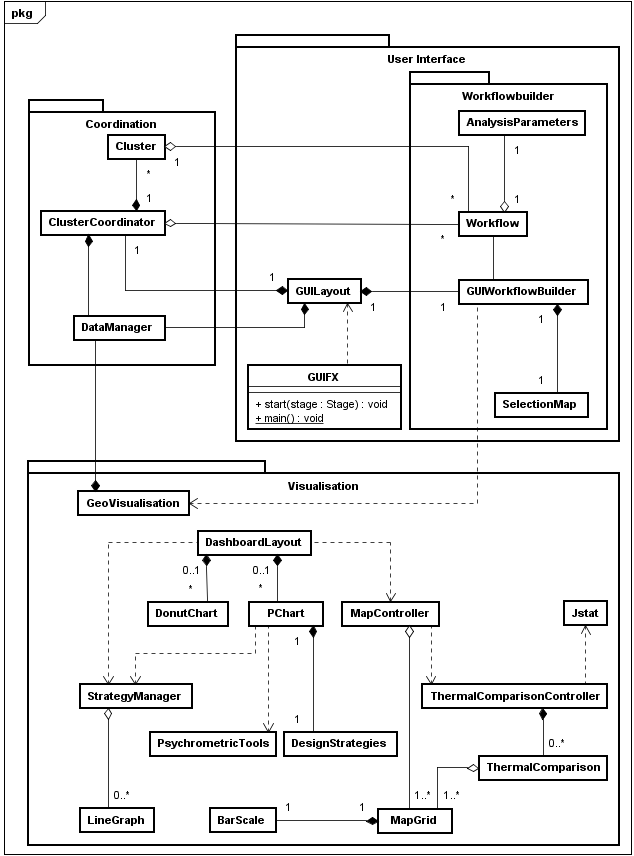


Figure 22 Coordination, User Interface and Visualisation packages overview

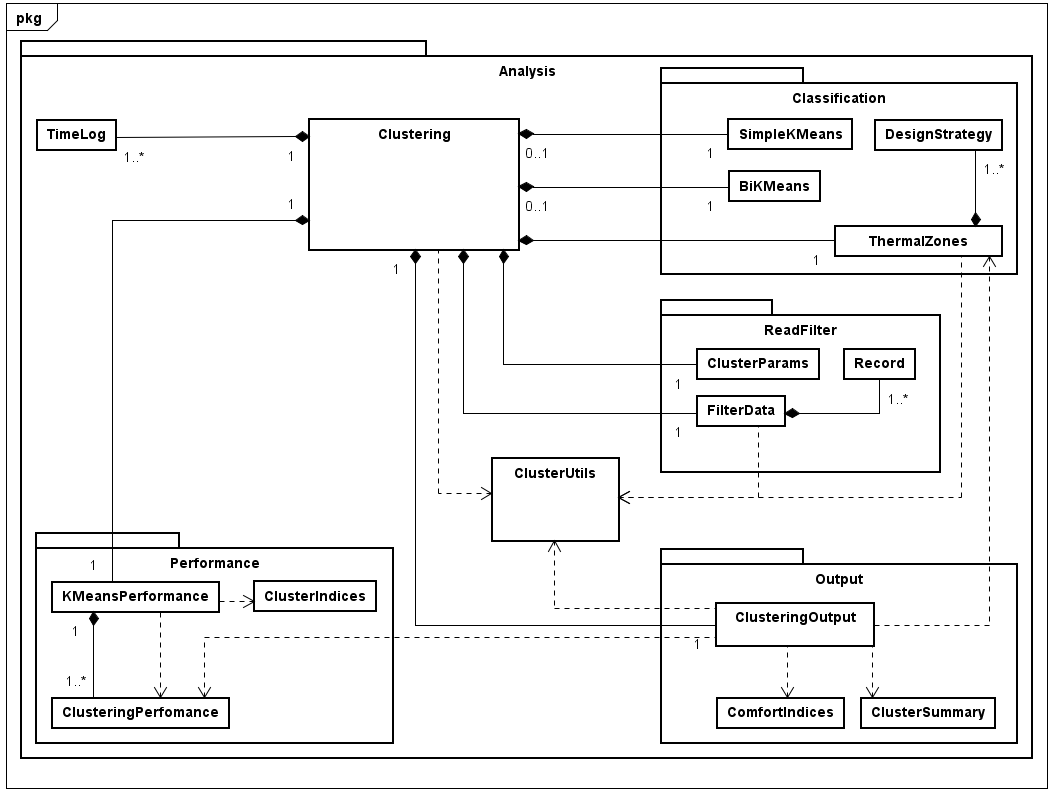


Figure 23 Analysis package overview

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. The GeoVisualisation class creates a webpage that references the dashboard framework components (also hosted within a public area of S3) and provides an interactive, online interface for visualising the results.

## Climate data

The data set comprises 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 & 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 24 illustrates the data preparation steps. A C# dot net program was written that takes a topojson (Bostock, 2017b) format file as input, describing the boundary (or collection of boundaries) that define the zone of interest. The program generates a point grid 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 program extracts climate data from a selected period from the CRU and CCMP datasets using the grid. CRU data has the same 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 Java program converts the netCDF data to the ASCII ESRI raster format. Each ASCII raster file represents a single month of a year and contains data for earth’s surface. The system finds data points in the raster files that coincide with grid vertices and stores them in the georeferenced grid in arrays. A function derives relative humidity using the ratio between vapour pressure and saturation pressure. Finally, the system outputs the prepared climate data as a csv file. Source code for the data preparation is available at: <https://github.com/rolyhudson/griddedClimateDataProcessing.git>

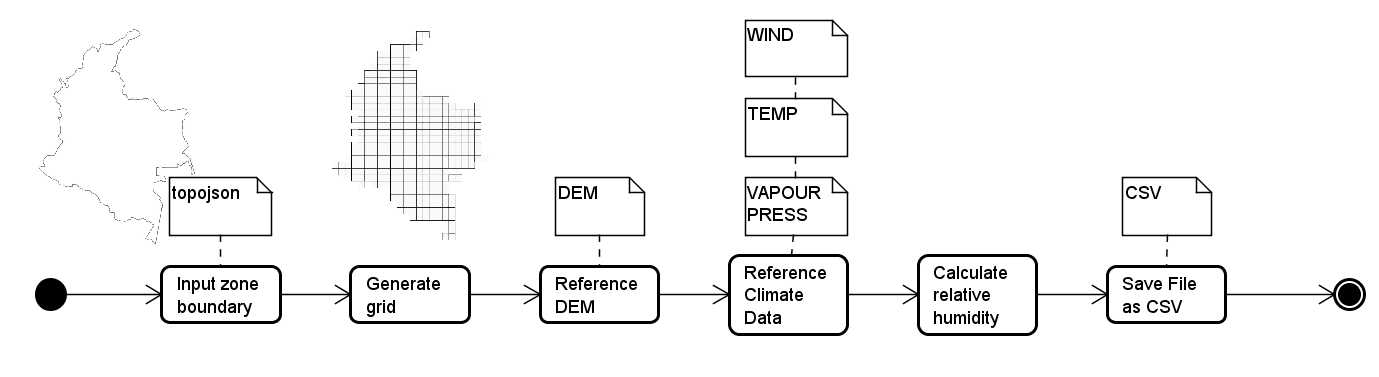


Figure 24 Flow diagram showing the stages of the data preparation process

## Workflow management system

Class diagrams referenced in this section are provide in appendix B.

### Workflowbuilder and Coordination

At the core of the application is the workflow management system consisting of classes defined with the Workflowbuilder package (Figure 58 and Figure 59, Appendix B) which communicate with the classes in the Coordination Package (Figure 60, Appendix B).

### Workflow creation

The Workflow class manages all the parameters for an AWS Step, additionally, the Workflow class includes an attribute, analysisParameters, an instance of the AnalysisParameters. The AnalysisParameters class includes attributes defining the of the domain specific configuration of the workflow. The user can configure this class instance via the GUI (Figure 25) 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, n.d.). The map shows the selected data set, locations as red and users can draw and edit a selection polygon or rectangle to define the region for analysis.

The ClusterCoordinator class maintains a list of workflows that it updates when a user edits a workflow in the UI. Within one session the UI can create and save multiple workflows. On exiting the application session, the system stores all workflows in S3 in a private bucket in JSON format. Each workflow instance has a GUID that determines the S3 object name. The S3 object contains a string in JSON format that is returned by the class’ serialise method. The UI’s instance of the DataManager class, datamanager, pushes all Workflow objects to S3.

### Monitoring resources and workflows

The ClusterCoordinator maintains a list of all EMR clusters created after a datetime parameter – monitorFrom (set by the user). An instance of the Cluster class represents each EMR cluster locally, managing all the parameters required to create a new EMR cluster (Figure 60). The ClusterCoordinator constantly monitors the AWS resources (Figure 14). 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 AWS cloud. The implemented monitoring algorithm (Figure 26) allows users to break between work sessions and shut down the application. The monitor updates the status of any running clusters or workflows when the user returns, and they can continue editing, copying or executing a workflow created in a previous session.

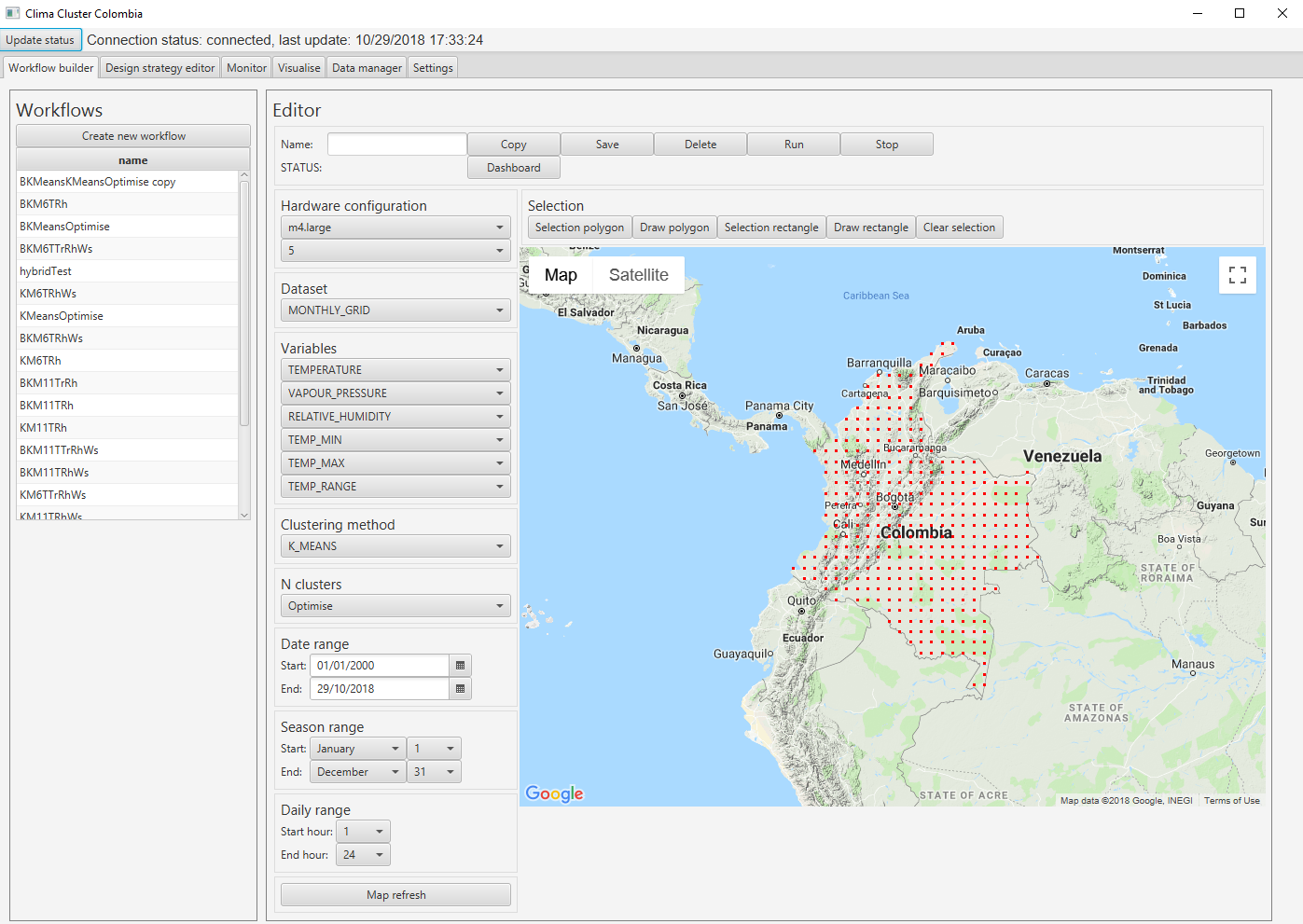


Figure 25 GUI for the Workflowbuilder

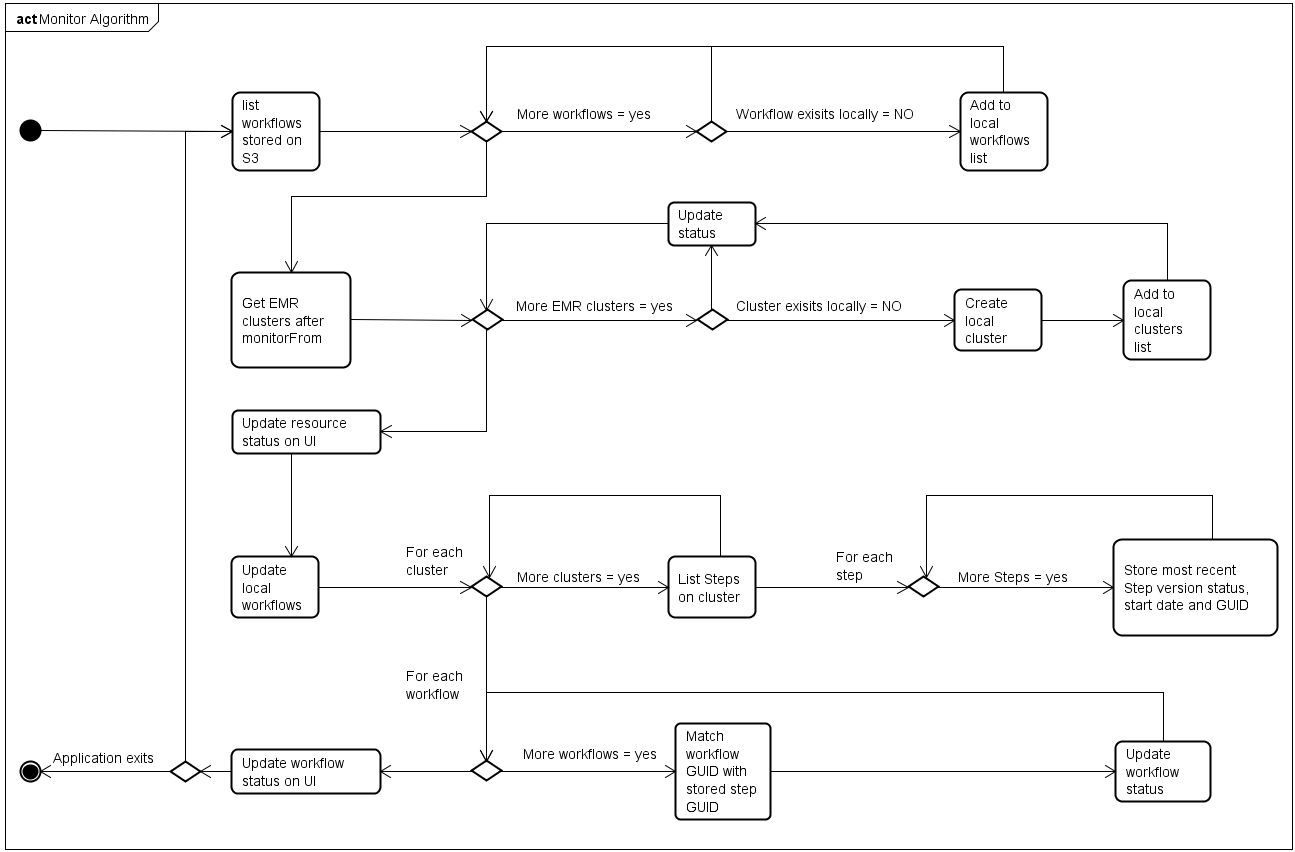


Figure 26 Resource and workflow monitoring algorithm

Resource monitoring takes place on a separate thread to the UI and executes 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 the method updates locally stored lists, it adds them to an ObservableList that populates the UI’s monitor tab (Figure 27). To edit a workflow, the user selects it from the left-hand panel in the Workflow builder tab (Figure 25). The UI shows the status of a workflow, its state (Figure 28) 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 29). Whereas an INITALISED workflow provides access to all workflow configuration controls but not the buttons to stop or generate the dashboard. The activity diagram in Figure 30 shows the different processing paths a user can follow when working with a workflow.

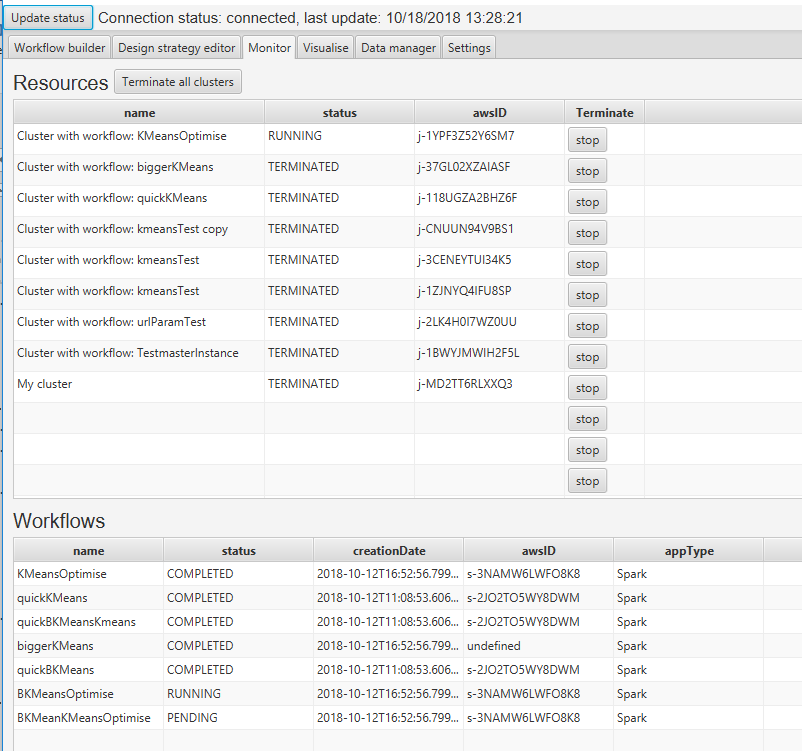


Figure 27 UI for monitoring resources and workflows

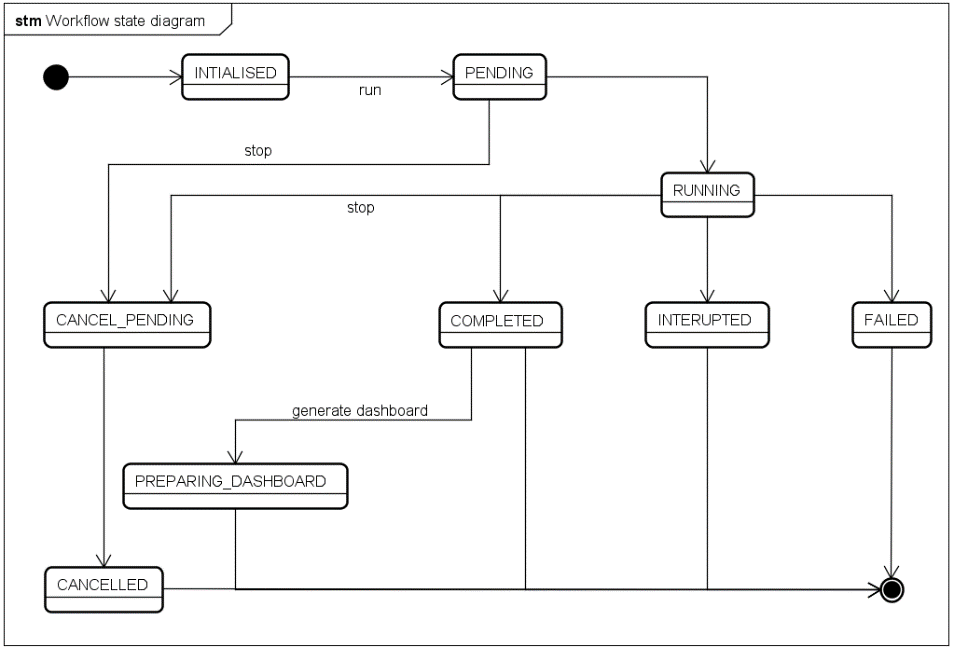


Figure 28 Discrete behavioural states for a workflow

### Running a workflow

Once the user has prepared their workflow, they click the run button on the UI and the ClusterCoordinator sends the workflow to EMR. The workflow is either added to an existing cluster or the system creates a new cluster if none exist, or the user requires a different hardware configuration (Figure 31).

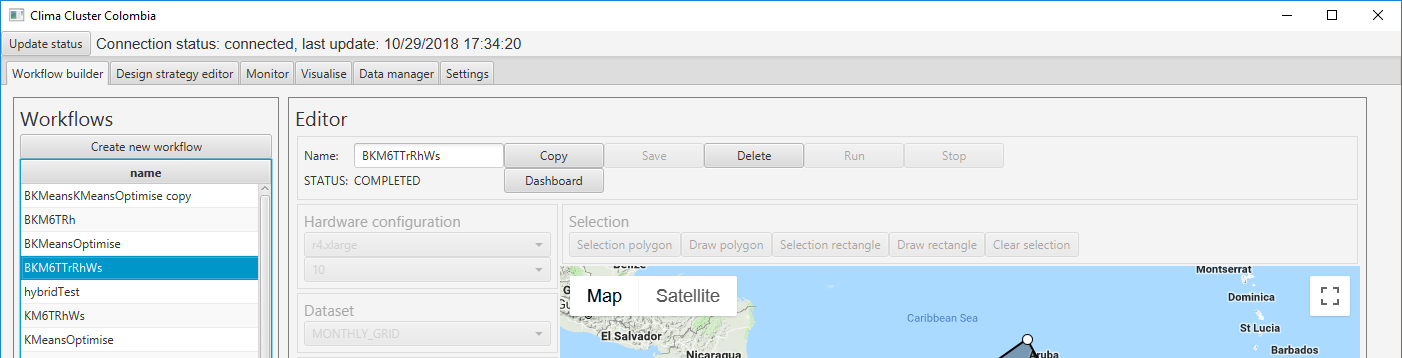


Figure 29 UI showing workflow with status COMPLETED

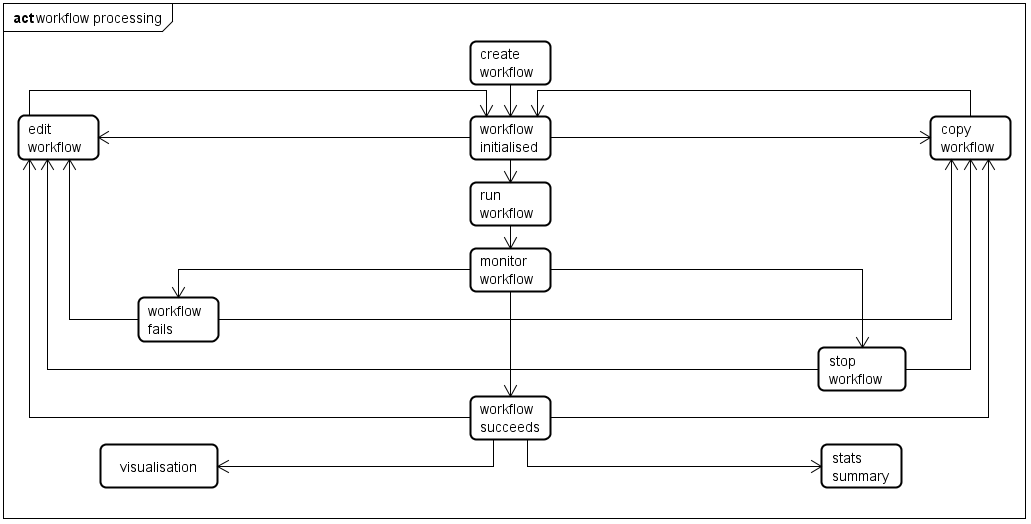


Figure 30 Logical steps in processing workflows

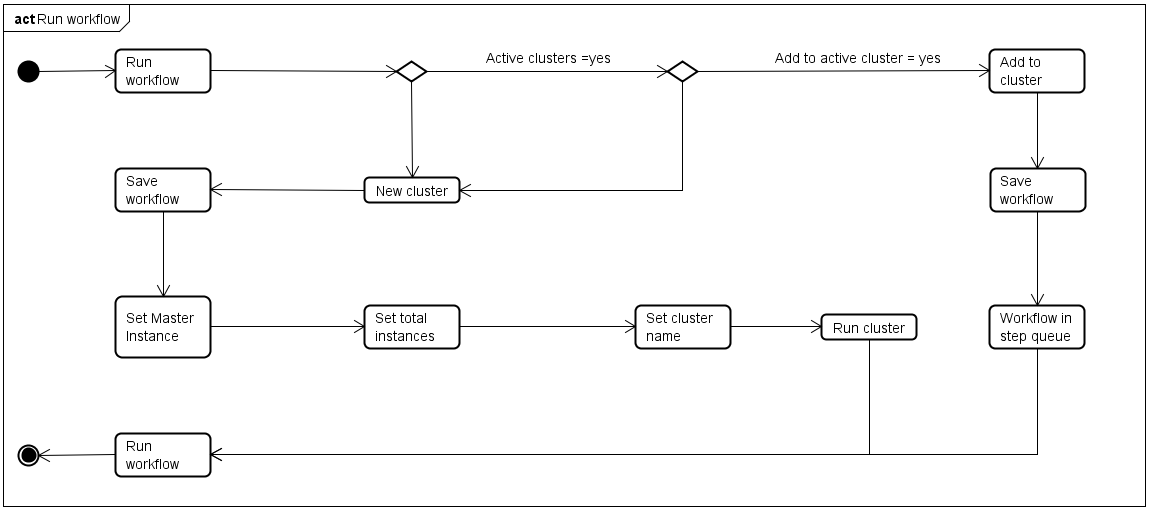


Figure 31 Logical steps in running a workflow

### Management of design strategies

The UI includes functionality for editing design strategies, this is a mock-up of a simple drawing interface that would allow users to draw, edit, import and save strategies over a psychrometric chart (Figure 32). Each strategy is represented as a closed polygon with a name. An array of these polygons can be stored, one per line, in a simple csv 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.

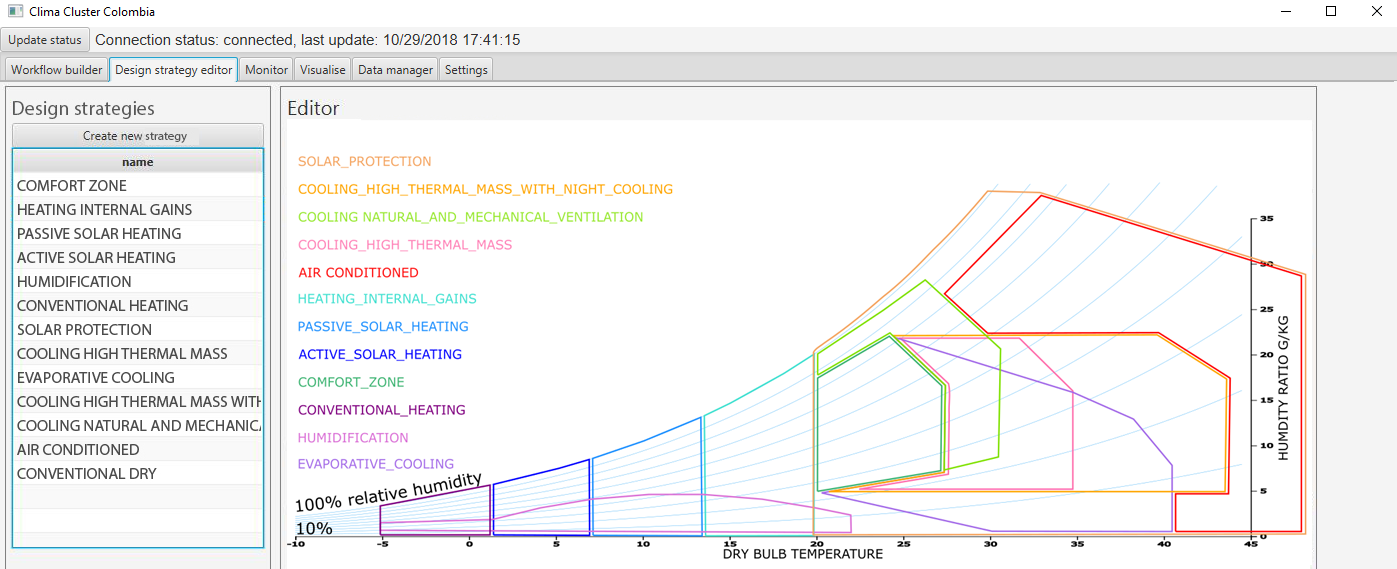


Figure 32 UI showing mock-up for the design strategy editor

## Analytics

### Overview

The analytic pipeline is a Spark application defined as an independent Java project [[1]](#footnote-1) , exported locally as a \*.jar and uploaded to S3 via the UI with the keypath “sparkJAR”. Parameters for the configuration of an EMR Step are defined by the workflow when the user submits it for analysis. 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 a spark-submit script. Arguments for the spark-submit specify:

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

The EMR management console within the cluster details (Figure 33) shows the seven arguments. When the EMR cluster is running, the command-runner passes arguments 4-7 to the class in the application that contains the main method. The Clustering class (Figure 61, Appendix B) includes the main method and acts as a controller for the analytics process, it initialises the SparkSession, controls the parsing of the clustering parameters, reads and filters input data and reads the design strategies. Depending on the selected clustering method, the pipeline evaluates performance before classifying selected data and identifying design strategies appropriate to each cluster and data-point. Finally, the analysis process writes results to S3. Figure 23 shows the class structure of the Analysis package and Figure 34 illustrates the sequence of processes used to implement the analysis.

A screenshot of a social media post

Description generated with very high confidence

Figure 33 AWS interface showing input parameters for a Step on EMR

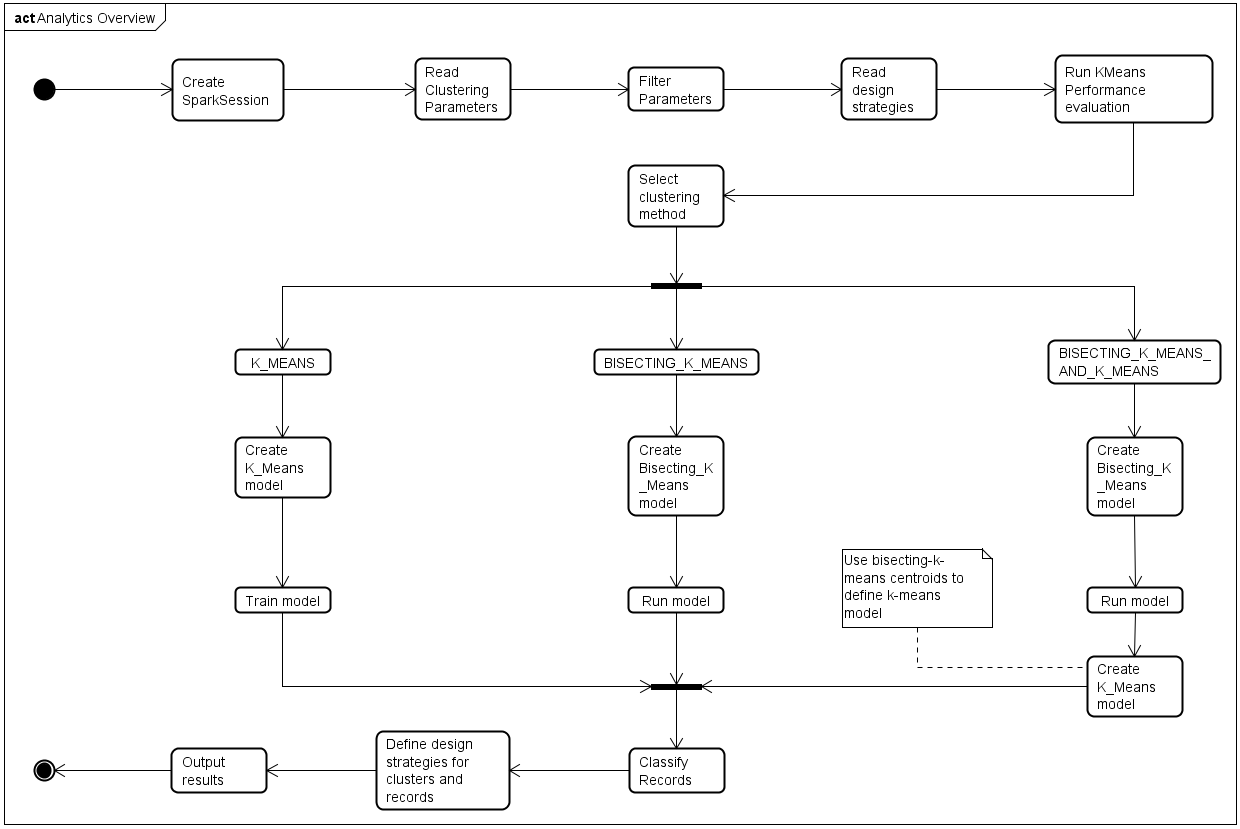


Figure 34 Logical steps in the analytics process

### Parsing parameters and filtering Data

The ClusterParams class (Figure 61, Appendix B) reads and de-serialises a string in JSON format, stored on S3, representing the Workflow class instance created by the GUIWorkflowbuilder. The JSON string is read directly to a Dataset<Row> and from that the analysisParameters property is selected to define the clustering method, number of clusters, temporal and spatial configuration of the workflow.

The FilterData class (Figure 61, Appendix B) 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 the FilterData class defines a JavaRDD of Record objects. This dataset contains only data points within the required spatial and temporal ranges.

The Record class (Figure 61, Appendix B) includes attributes to store all the variables contained on each line of the input data and identifies which features to use for clustering. In addition, the Record object includes a psychrometricPoint, a 2d array defined by temperature and relative humidity, a list of design strategies associated with the Record and the comfort indices (UTCI and IDEAMCI) for the Record. The Record class includes a method that returns a normalised Vector of the variables destined for the clustering processes (features are normalised with Spark’s MLlib L2 norm). The ThermalZones class reads and parses design strategies stored as a text file on S3 (Figure 63, Appendix B) and retains them as a list.

### Performance Indices

The KMeansPerformance (Figure 62, Appendix B) class is responsible for defining performance indices for the selected clustering method. User can configure a workflow with a fixed number of clusters or request it optimise the number of clusters. A list of ClusteringPerformance objects (Figure 62, Appendix B), an attribute of the KMeansPerformance class stores performance metrics for each solution. The list either contains results for a single clustering solution, or if the user selected optimisation it stores clustering indices for each solution from two to sixty clusters. ClusteringPerformance objects store WSSSE, Silhouette and Dunn indices. The ClusterIndices class (Figure 62, Appendix B) calculates these metrics, which include different methods for each index depending on the selected clustering method.

Optimisation is based on the WSSSE for each solution and the optimal number of clusters is determined using the elbow method. The static method, findElbowCluster in the ClusteringPerformance class uses a geometric algorithm that creates a graph for all solutions and finds the point on the curve furthest from a line subtended between the first and last points. Figure 35 shows a scenario where the system found twelve clusters to be the most optimal solution with a WSSSE of 3.7.



Figure 35 Graph showing the elbow method

### Classification and design strategy assignment

Once the system has generated the performance metrics, and the optimal number of clusters found, the analytics proceeds to defining the chosen clustering model. This is based on two classes SimpleKMeans and BiKmeans (Figure 63, Appendix B) which the system uses independently or combines to provide a third, hybrid method. The selected clustering model is used to classify each Record (assigned a cluster number) and the ThermalZones class tests the Record’s psychrometricPoint to find applicable design strategies and assigns these to the Record. The DesignStrategy class (Figure 63, Appendix B) stores the boundary of a strategy as a list of vertex coordinates defined in terms of temperature and relative humidity. To test which design strategies are applicable to a Record its psychrometicPoint (a 2d point defined by temperature and relative humidity) can be containment tested against the strategy’s boundary vertices, if the point is inside the polygon the system adds the DesignStrategy to the Records list.

### Output of results

In anticipation of querying and quickly visualising the results according to various spatiotemporal ranges, the system uses Spark to pre-process, transform and structure the output data. The process stores meta-data related to the clustering performance and configuration of each cluster. The structure of the results anticipates three key spatiotemporal queries. 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 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

The process converts performance metrics (stored as a list of ClusteringPerformance objects in the KMeansPerformance class) to a Dataset<Row> and then written to a file in JSON format. For each cluster, the getComfortIndicesClusters method in the ComfortIndices class (Figure 64, Appendix B) defines average temperature, relative humidity, thermal indices, maximum and minimum values for temperature, temperature range, relative humidity and wind speed. The comfort indices and the centroids of each cluster are arguments for the reportClusterSummary method in the ClusterSummary class (Figure 64, Appendix B). This method generates a list of ClusterSummary objects, one for each cluster in the solution, describing centroids, the number of data points contained and the cluster’s comfort indices. 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. Spark’s createDataFrame method generates a Dataset<Row> from the list of ClusterSummary objects are writes them to a file in JSON format.

#### 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. The system creates a typical period for each month by grouping by location, then finding the most frequent cluster number and assigning appropriate strategies and storing the results.

#### Year + month + cluster

This form of query iterates each year and each month (Figure 36) storing the filtered data and a summary of the frequency of design strategies at each iteration. The system stores one data set for each year and each month of the original data and the frequency of found design strategies for each cluster.

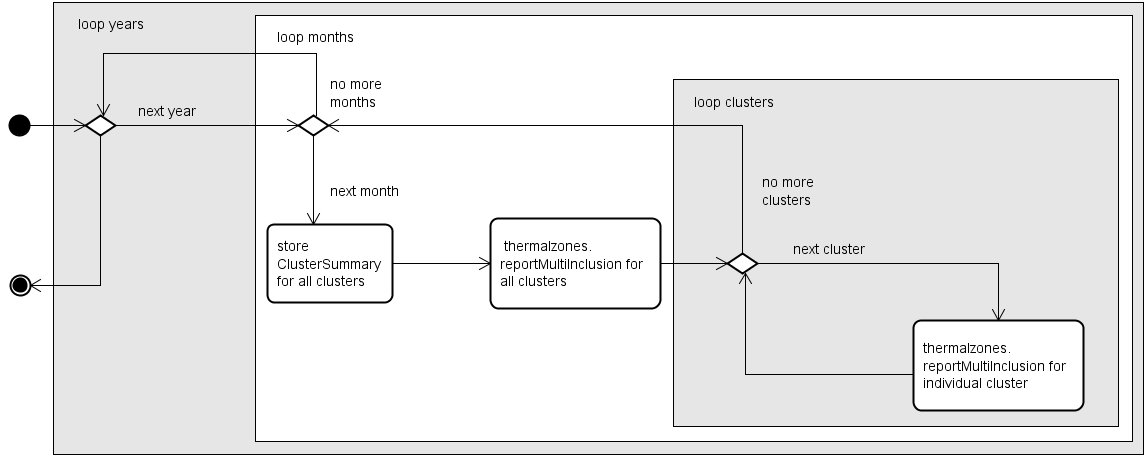


Figure 36 Sequence for extracting data for each cluster, each month and each year

## Dashboard + visualisation

Once the analysis successfully completes, EMR updates the Step’s status to COMPLETED. The UI’s resources monitoring function detects the status and changes the controls available on the interface, enabling the dashboard button (Figure 37) and disabling other controls. Clicking the dashboard button triggers an event in the GUIWorkflowBuilder class that checks if the dashboard exists by checking to see if the workflow has a dashboardURL property. If the URL is present, the UI updates the web engine with the workflow’s dashboard URL. The application also opens the same URL on the default browser.

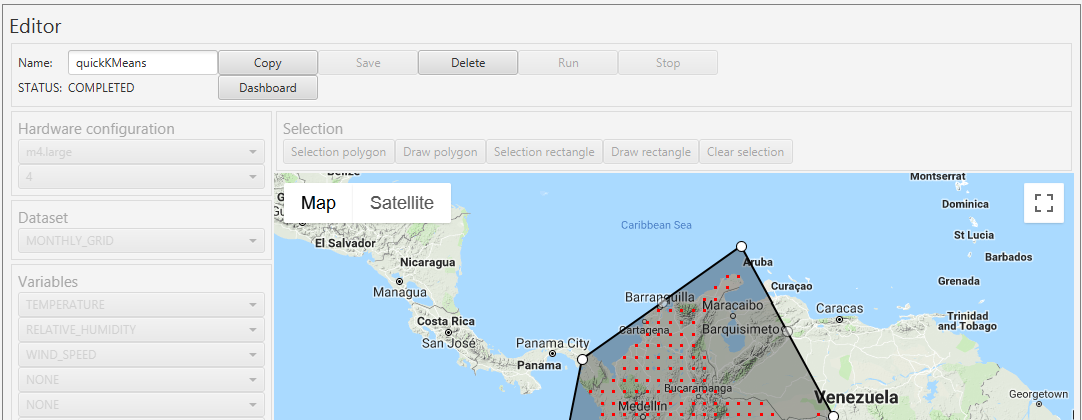


Figure 37 UI showing the dashboard button available

If no dashboard exists a new folder name is created, this will be located within a public bucket on S3 and contain the dashboard data files and an index.html. The GUIWorkflowBuilder creates a new thread that instantiates the GeoVisualisation class (Figure 65, Appendix B). The Spark application is distributed across several machine instances on EMR, output is also distributed so multiple part files are generated. The role of the GeoVisualisation class is to combine these output files from the Spark application and transfer the merged files to a newly created space on a public bucket on S3. The GeoVisualisation class iterates all objects produced by the analysis process and stored under a common keypath[[2]](#footnote-2) prefix on S3. The process combines all files that include the string “part” in their name and share the same sub-folder and transfers them to the public bucket in a location defined by the dashboard folder name and the file’s sub-folder. The text file containing the serialised description of the original workflow is renamed parameters.txt and transferred together with an index.html file. Once the combine-and-transfer process completes the user can click the dashboard button and examine the dashboard either within the visualise tab in the application or on their browser.

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 65, Appendix B) contain functions defining the behaviour of the interface. The classes define charts and maps using the d3.js library (Bostock, 2017a). All the classes are instantiated several times on the dashboard to visualise the results and access data from the analytic process in a standardised file structure. Figure 38 describes the sequences of functions called on opening the page and where objects are instantiated. User interaction with a control on the dashboard triggers a subset of the behaviour (shaded grey). The dashboard contains four sections for exploring the results; overview, cluster summary, cluster explorer and comfort comparison.

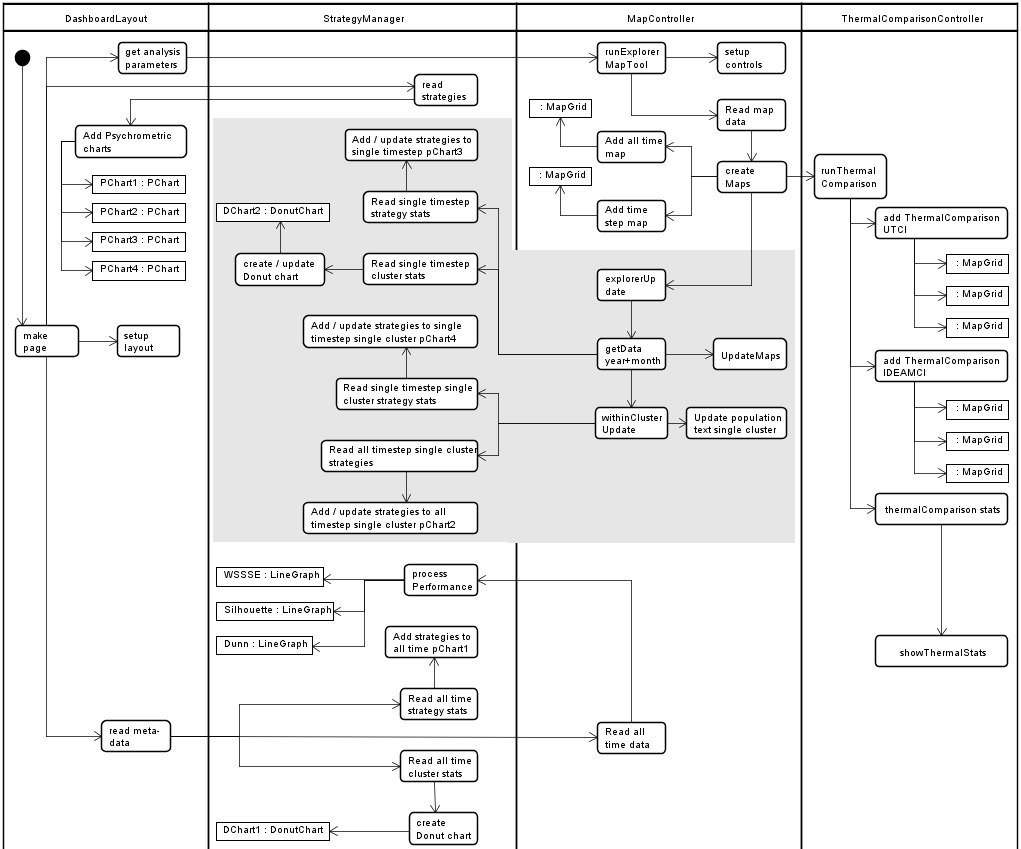


Figure 38 Behaviour model for the JavaScript dashboard system

### Overview

The overview section displays the input parameters and performance indices plus the URL of the page. If the analysis included 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 (Figure 39). Scores for each index are shown on the left-hand panel. If the user specified the number of clusters, the score for each index is shown in each panel (Figure 40).



Figure 39 Section of dashboard UI showing the results of an optimised clustering analysis

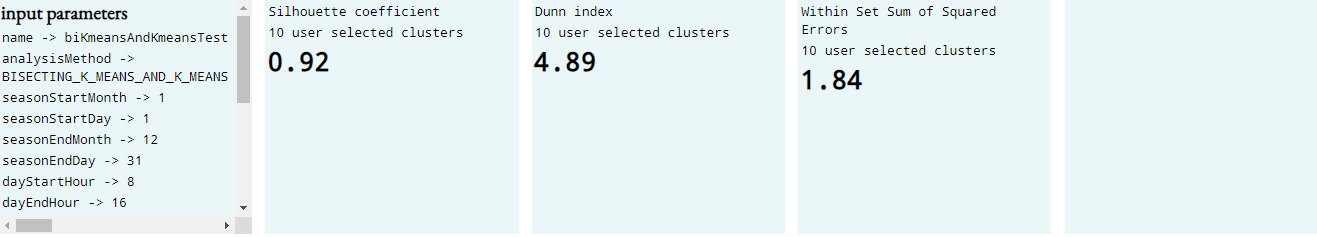


Figure 40 Section of dashboard UI showing the results for a single clustering solution

### Cluster Summary

The summary section contains a table summarising the performance of each cluster defined by the attributes of the ClusterSummary class described in section 4.6.5.1. This includes ranges of the key comfort variables, comfort indices, cluster id, population and associated strategies (Figure 41).

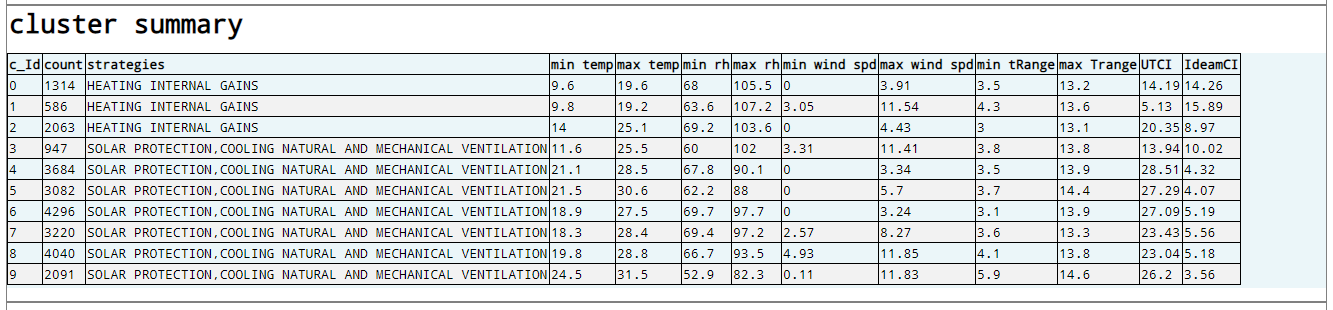


Figure 41 Section of dashboard UI showing summary of each cluster’s population, design strategies, variable ranges and comfort indices

### Cluster explorer

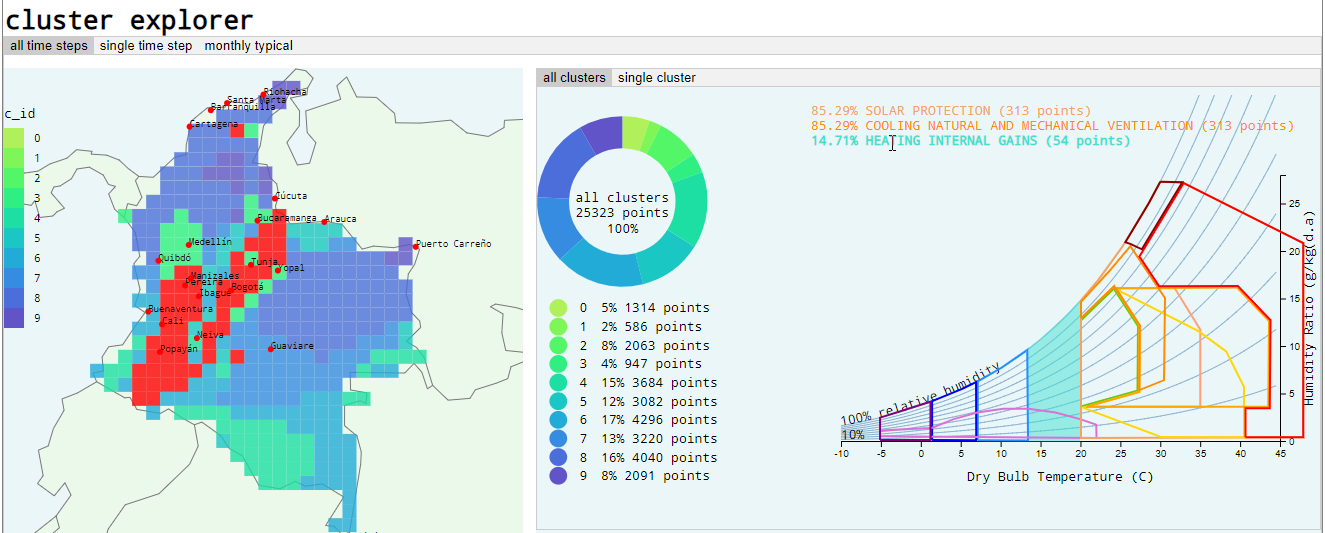


Figure 42 Section of dashboard UI showing cluster explorer with all time-steps and all clusters displayed. The user’s mouse is over the design strategy “heating internal gains” and corresponding map cells are highlighted red

The cluster explorer section provides various ways of looking at the results. These are accessed by clicking on a tab to select a temporal method 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
5. monthly typical / all clusters
6. monthly typical / single cluster

On opening the page all time steps / all clusters are presented (Figure 42) this corresponds with the output described in section 4.6.5.2. Instances of the MapGrid, DonutChart and PChart objects provide the user can interact with these to explore the data. The PChart includes DesignStrategy class boundary zones as shown in the UI (Figure 32), 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 42). The donut chart shows the distribution of points within clusters, 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. On selecting the single cluster tab, an individual cluster can be selected with a range input, its population, associated strategies and their spatial distribution can be examined (Figure 43).



Figure 43 Section of dashboard UI showing cluster explorer with all time steps and single cluster displayed

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 inputs for year and month. This corresponds with the output described in section 4.6.5.4. Selecting the single cluster tab in this mode provides more detailed information of a single cluster at a specific time (Figure 45). Monthly typical / all clusters tab displays the results extracted using the process described in section 4.6.5.3.

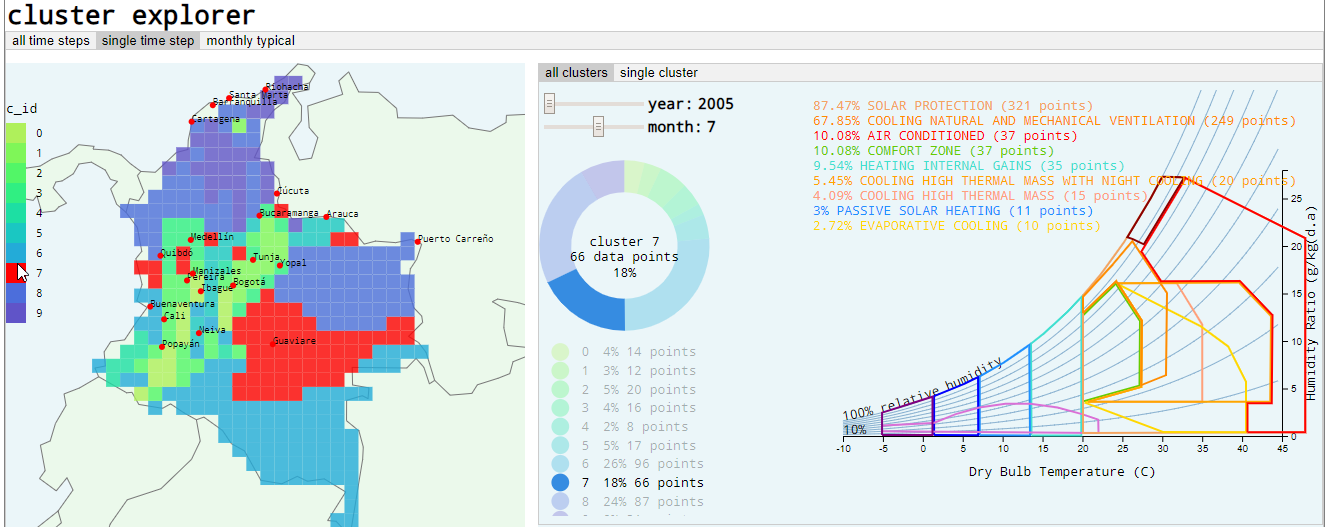


Figure 44 Section of dashboard UI showing cluster explorer with single time step and all clusters displayed

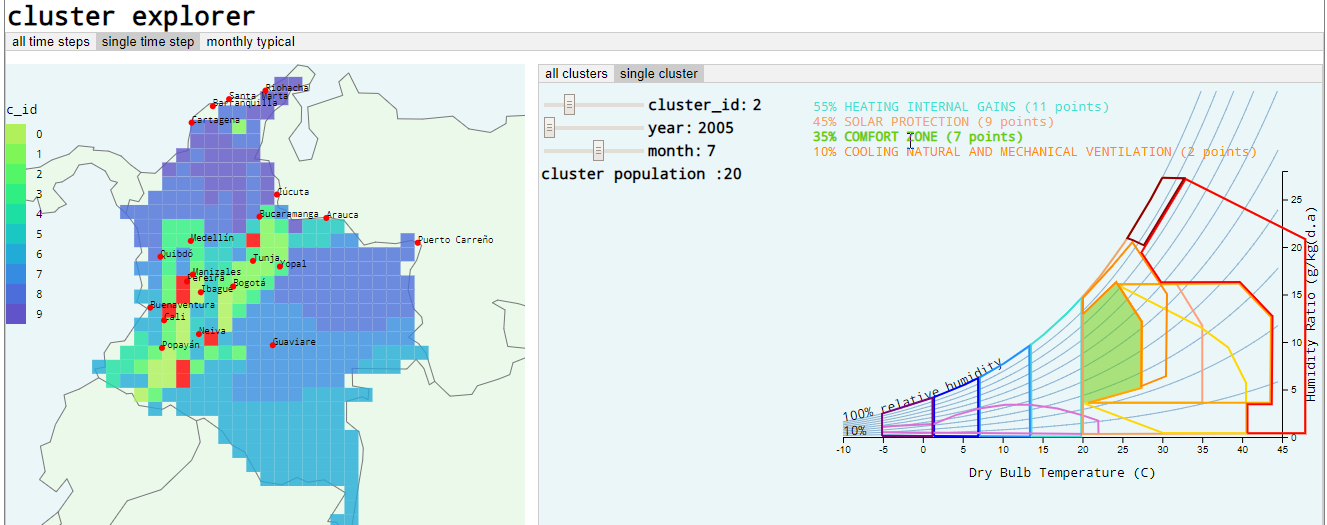


Figure 45 Section of dashboard UI showing cluster explorer with single time step and single cluster displayed

### Comfort comparison

The last section on the dashboard compares comfort indices predicted by the clusters to those calculated by the variables for each grid cell or city (Figure 46). The analysis defined two comfort indices; UTCI and IDEAMCI. The left-hand panel provides a brief description of each index and a link to more detail information. The ThermalComparison object is created for each index, this includes three MapGrid objects showing the predicted index, calculated and difference. On the right-hand panel summary statistics of differences between predicted and calculated values area show as mean, standard deviation and root mean squared error.

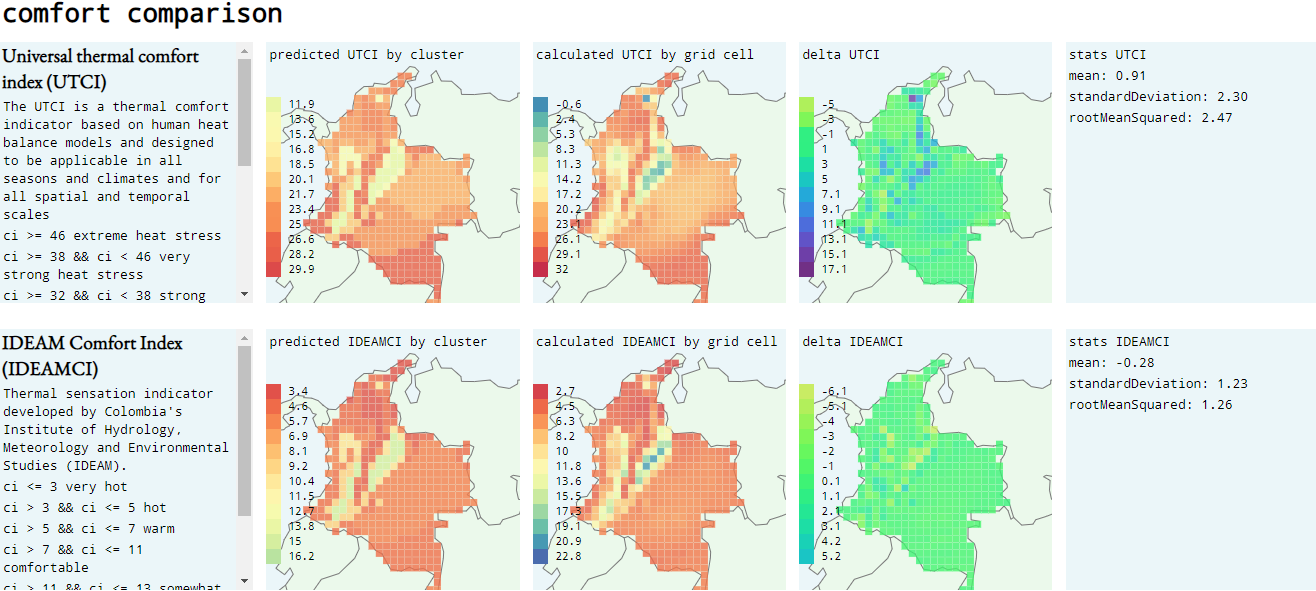


Figure 46 Section of dashboard UI showing thermal index comparison. Top row UTCI comparison, predicted by clusters (left), calculated (centre) and error (right). Bottom row compares IDEAMCI

## Summary

The implemented system combines three key subsystems; workflow management, distributed analytics and a visualisation dashboard. Workflow management is implemented as a desktop application written in Java that enables users to graphically define analysis processes, configure cloud-based hardware, access analysis results from previous jobs, monitor analysis jobs in progress and view a predefined set of low-energy design strategies. Users submit analytic jobs and hardware specifications via the desktop UI to AWS’s EMR. Users can define multiple clusters, each running up to ten nodes and submit multiple jobs to a single cluster. Hadoop clusters running Apache Spark are created in a VPC and once they are running the analytic job is submitted to Spark.

The analytical process is an Apache Spark application written in Java that reads the configuration of the analysis procedure and converts a climate data set stored on AWS’s S3 to a JavaRDD of Record classes. The process filters the RDD according to the user defined parameters and generates one or more clustering solutions using one of three clustering methods. The system provides an option for seeking an optimal clustering according to the clustering cost attribute defined in Spark’s MLlib, alternatively users can define the number of clusters required. Once the system finds the optimal solution, or generates the user defined solution, the system measures the performance of the solution using three clustering indices and two thermal comfort indices. Using the clustering solution, the program classifies all data points and assigns applicable design strategies to each cluster and to each data point. Spark distributes tasks defined by the application across all nodes in the EMR cluster, as each task completes it writes raw results to a private bucket in AWS’s S3. When all tasks are complete the program consolidates and pre-processes individual result files and transfers them to a public bucket on S3. The application generates a URL that the users access via a button on the UI.

The URL provides access to the visualisation dashboard both within the UI and in the default browser. The dashboard is a frontend JavaScript framework based around the d3.js visualisation library that provides different representations for exploring the analysis results. The analysis parameters and clustering performance are summarised at the top of the page. A table shows the population, design strategies, variables ranges and thermal comfort indices for each cluster. The cluster explorer section allows the user to examine the results on a map and at different temporal granularities. The dashboard provides interaction to explore cluster distribution, placing the mouse over one of the map, the scale bar or the donut chart highlights the relevant cluster in the other two. The cluster explorer links the psychrometric chart to the map. Moving the mouse over a graphic representation of a design strategies on the psychrometric chart highlights locations on the map where that strategy is applicable. The last section on the dashboard provides three maps for each of two thermal comfort indices. The first map represents the cluster predicted thermal comfort index and the second the calculated index for each grid cell. The third map shows the error in the prediction of thermal comfort by the clustering solution and provides a domain specific way of visually evaluating the results.

# Results and evaluation

## Evaluation

The system should produce recommendations for environmental design strategies based on the analysis of a set of weather data. This chapter tests the hypothesis that design strategies can be linked with patterns discovered in weather data at various spatiotemporal scales and with different subsets of variables. The dissertation proposed that this can be tested using a big data architecture that enables data analytics over large sets of weather data. The system is decomposed for evaluation; clustering weather patterns and matching of recommended design strategies is assessed quantitively. The functionality of the system and its usefulness to environmental designers is evaluated qualitatively using presentations and software walkthroughs with domain experts.

## Quantitative evaluation

### Metrics used for evaluation

The literature review identified three validaity metrics to express the dissimilarity (Silhouette) combined cohesion and separation (Dunn) and internal cluster cohesion (WSSSE). The system implemented the first two in the analytic process and the latter is part of Spark’s MLlib. Using two comfort indices, the system incorporated domain specific quantitative evaluation. The difference between the comfort index (Δ comfort UTCI and Δ comfort IDEAMCI) predicted by the cluster and the index as determined by each data point indicates how well the clusters represent conditions of thermal comfort.

### Evaluation approach

Modelled on a framework developed for a recent clustering study (Netzel & Stepinski, 2016) a comparison between a series of clustering experiments was undertaken. Discussion with domain specialists (section 5.3) suggested the parameters; temperature, relative humidity and wind speed were the most important in determining human comfort. Results from Netzel and Stepinski's (2016) study indicated that daily temperature range was also significant in creating well defined clustering solutions. Combinations of these four parameters are examined to assess the influence of wind speed and daily temperature range.

* Temperature, relative humidity (TRh).
* Temperature, daily temperature range, relative humidity and wind speed (TrRh).
* Temperature, relative humidity and wind speed (TRhWs).
* Temperature, daily temperature range, relative humidity and wind speed (TTrRhWs).

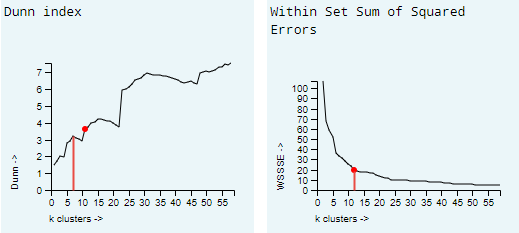


Figure 47 Results from optimised studies. Left: k =6 at first Dunn maximum of 3.3. Right: inflexion is at k = 11 for WSSSE of 20.

The number of clusters to study, *k*, was determined by running three analytics workflows that sought to optimise *k* for each of the three implemented methods using the WSSSE validity metric[[3]](#footnote-3). The hybrid method, combining hierarchical and non-hierarchical clustering produced identical results to the hierarchal method and was discarded. For the remaining two methods, Silhouette and WSSSE indices of the remaining methods indicated that *k=11* performed well (Figure 47 right). The Dunn index of the non-hierarchal indicated that *k=6* (corresponding with the first maximum (Figure 47 left)) was optimal. The hierarchical, bisecting k-means (BKM) and non-hierarchical k-means (KM) methods were combined with the four parameter sets and two values for k giving 16 different analytic workflows (Table 2).

Each workflow was run on an AWS EMR cluster configured with r4.xlarge instances, one master and nine core nodes, this configuration performed well running the optimisation studies and AWS recommends the r4 instances series for in-memory intensive distributed data analysis. Results from each study are summarised in terms of processing time and financial cost, the performance metrics and differences from expected comfort indices. Dissimilarity matrices are used to compare workflows.

Table 2 Evaluation workflows

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| method | KM | KM | KM | KM | KM | KM | KM | KM |
| k | 6 | 6 | 6 | 6 | 11 | 11 | 11 | 11 |
| features | TRh | TrRh | TRhWs | TTrRhWs | TRh | TrRh | TRhWs | TTrRhWs |
|  | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| method | BKM | BKM | BKM | BKM | BKM | BKM | BKM | BKM |
| k | 6 | 6 | 6 | 6 | 11 | 11 | 11 | 11 |
| features | TRh | TrRh | TRhWs | TTrRhWs | TRh | TrRh | TRhWs | TTrRhWs |

### Statistical evaluation of analyses

Table 3 shows the results from the evaluation, ranked according to a combined score based on the ranking for each performance metric and the Δ comfort indices (low scores show better overall performance). No single method tested showed best performance across all indices, both hierarchical BKM and non-hierarchical clustering are present in the first five workflows. Mean values for the Δ comfort indices show that all the workflows systematically failed to capture the colder extremes – estimating warmer conditions in both (lower values are hotter conditions in IDEAMCI). The top five all have k=11 clusters, all include parameter combinations of temperature, relative humidity and wind speed, temperature range is only present at #4 workflow KM11TrRh.

Including temperature range is almost essential in being part of the top group performers in terms of WSSSE. The WSSSE matrix (Figure 48) shows a group of six very similar workflows, four of which include temperature range. Whereas, using wind speed as a parameter seems to result in much lower WSSSE values. The top performing workflow KM11TrRh is interesting as it performs joint 1st in the Silhouette, Dunn and WSSSE indices. Its high Dunn index sets it apart for all other workflows in the dissimilarity matrix (Figure 49), but it performs poorly in the Δ comfort indices. Including temperature range defines a pattern of poor performance in Δ comfort indices. Lighter areas in the dissimilarity matrices in Figure 50 and Figure 51 show how including temperature range as a parameter causes a strong dissimilarity to workflows where it was absent.

Table 3 Ranked results

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  | **Δ utci** | | | **Δ ideamci** | | |  |
| **rank** | **name** | **t secs** | **$** | **sil** | **rank** | **dunn** | **rank** | **wssse** | **rank** | **mean** | **rmse** | **rank** | **mean** | **rmse** | **rank** | **score** |
| **1** | **KM11 TRh** | 2623 | 0.488 | 0.99 | 1 | 5.84 | 2 | 2.51 | 5 | 1.28 | 3.18 | 9 | -0.16 | 1.27 | 1 | **18** |
| **2** | **BKM11 TRh** | 2624 | 0.488 | 0.99 | 1 | 4.14 | 5 | 2.8 | 6 | 1.33 | 3.17 | 8 | -0.2 | 1.3 | 4 | **24** |
| **3** | **KM11 TRhWs** | 4242 | 0.789 | 0.93 | 8 | 4.4 | 4 | 19.01 | 9 | 0.49 | 2.4 | 2 | -0.09 | 1.29 | 3 | **26** |
| **4** | **KM11 TrRh** | 2660 | 0.495 | 0.99 | 1 | 8.46 | 1 | 0.68 | 1 | 1.93 | 5.62 | 15 | -0.65 | 3.23 | 13 | **31** |
| **5** | **BKM11 TRhWs** | 4128 | 0.768 | 0.9 | 10 | 2.93 | 10 | 21.43 | 10 | 0.65 | 2.17 | 1 | -0.18 | 1.27 | 1 | **32** |
| **6** | **KM6 TRh** | 1716 | 0.319 | 0.97 | 5 | 3.53 | 9 | 7.6 | 7 | 1.24 | 3.35 | 10 | -0.14 | 1.36 | 5 | **36** |
| **7** | **BKM6 TRh** | 1648 | 0.307 | 0.97 | 5 | 3.8 | 8 | 7.63 | 8 | 1.41 | 3.35 | 10 | -0.24 | 1.36 | 5 | **36** |
| **8** | **BKM11 TrRh** | 3918 | 0.729 | 0.98 | 4 | 5.04 | 3 | 0.83 | 2 | 1.93 | 5.61 | 13 | -0.66 | 3.24 | 15 | **37** |
| **9** | **KM11 TTrRhWs** | 4061 | 0.756 | 0.88 | 11 | 4.13 | 6 | 34.83 | 12 | 0.6 | 2.77 | 3 | -0.11 | 1.39 | 8 | **40** |
| **10** | **KM6 TrRh** | 1991 | 0.371 | 0.97 | 5 | 4 | 7 | 1.98 | 3 | 1.91 | 5.61 | 13 | -0.64 | 3.23 | 13 | **41** |
| **11** | **KM6 TRhWs** | 2684 | 0.500 | 0.87 | 12 | 2.92 | 11 | 34.45 | 11 | 0.94 | 3 | 7 | -0.26 | 1.47 | 10 | **51** |
| **12** | **BKM6 TRhWs** | 2740 | 0.510 | 0.84 | 13 | 2.78 | 14 | 36.04 | 13 | 0.84 | 2.88 | 5 | -0.26 | 1.37 | 7 | **52** |
| **13** | **BKM11 TTrRhWs** | 2500 | 0.465 | 0.82 | 14 | 2.92 | 11 | 38.07 | 14 | 0.71 | 2.85 | 4 | -0.23 | 1.6 | 11 | **54** |
| **14** | **KM6 TTrRhWs** | 2726 | 0.507 | 0.77 | 15 | 2.85 | 13 | 51.21 | 15 | 0.98 | 2.99 | 6 | -0.24 | 1.46 | 9 | **58** |
| **15** | **BKM6 TrRh** | 2602 | 0.484 | 0.93 | 8 | 2.65 | 15 | 2.39 | 4 | 1.93 | 5.63 | 16 | -0.67 | 3.25 | 16 | **59** |
| **16** | **BKM6 TTrRhWs** | 2719 | 0.506 | 0.77 | 15 | 2.56 | 16 | 54.25 | 16 | 1.06 | 3.35 | 10 | -0.33 | 1.74 | 12 | **69** |

Figure 50 and Figure 51 show a large, distinct set of similarly performing workflows in the top left, dark areas with two to four parameters. Bottom right is a set of poorly performing workflows all with only two parameters and all including temperature range. Within the UTCI matrix, the top five better performing workflows all involve three or four parameters, all include windspeed. This contrasts with the IDEAMCI matrix where including temperature and relative humidity appears more important.



Figure 48 WSSSE dissimilarity matrix

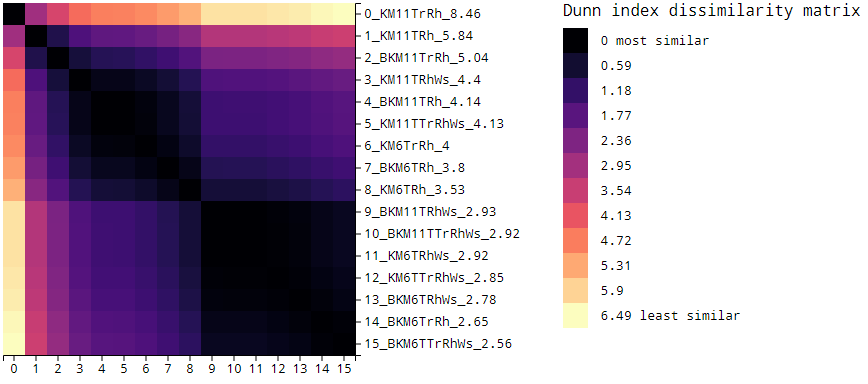


Figure 49 Dunn index dissimilarity matrix

The dissimilarity matrix for the Silhouette index (Figure 52) shows a group of better performing workflows with similar values, ranked in the top four are the *k=11* methods and includes KM and BKM methods with just temperature and relative humidity. These were mid-level performers (#7 and #8) in the Δ UTCI comfort and top performers in the Δ IDEAMCI. Like the Dunn, the Silhouette dissimilarity shows how including wind speed appears to negatively affect performance.

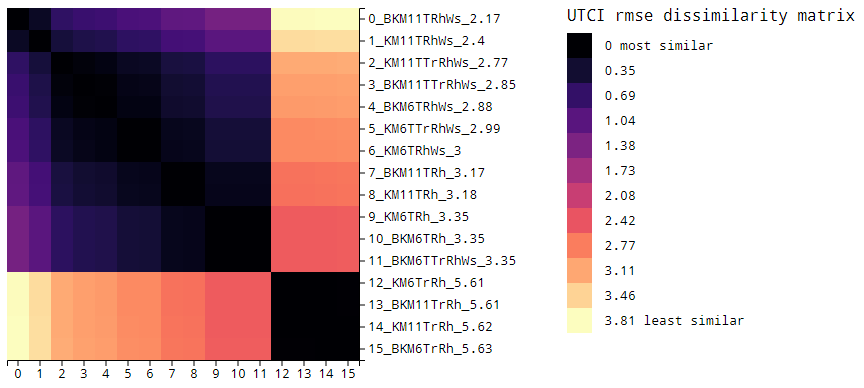


Figure 50 UTCI rmse dissimilarity matrix

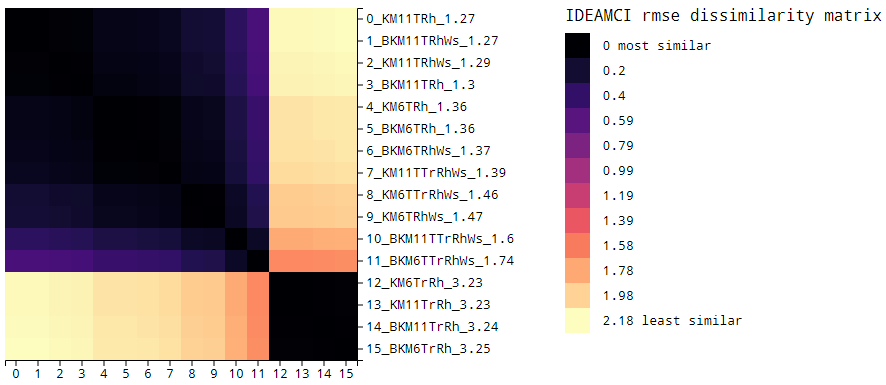


Figure 51 IDEAMCI rmse dissimilarity matrix

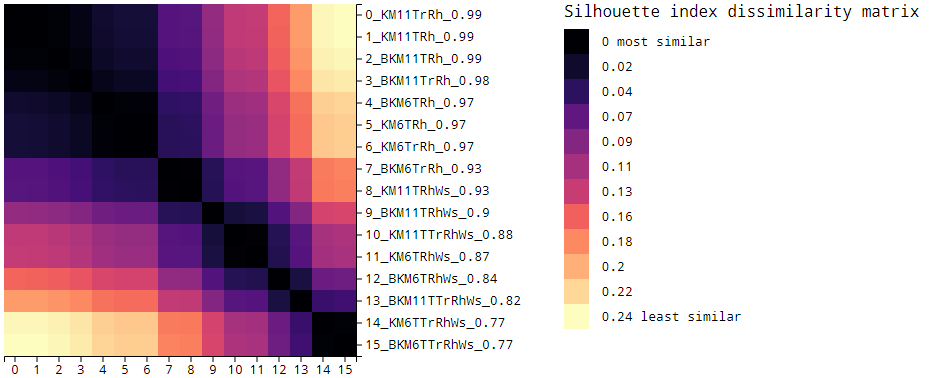


Figure 52 Silhouette index dissimilarity matrix

### Graphical evaluation

The software produced dashboards for each workflow[[4]](#footnote-4) these interactive graphical representations allowed dynamic exploration of spatiotemporal results, which supported the statistical results. Workflow KM11TrRh (ranked 4th) provides a good example of how visual inspection of mapped results is an important part of the evaluation. In Δ comfort UTCI, the cluster-predicted values (Figure 53 left) show almost unchanged comfort indices across the country, whereas the calculated values capture the topography (Figure 53 centre) and the differences expressed in darker colours (Figure 53 right). Figure 53 should be contrasted to the same comparison for the top Δ comfort performing, BKM11TRhWs (Figure 54), where the clusters can be seen to reflect the changes in expected comfort with altitude. This simple visual cross referencing shows ranking in Table 3 alone is not enough to select a workflow.

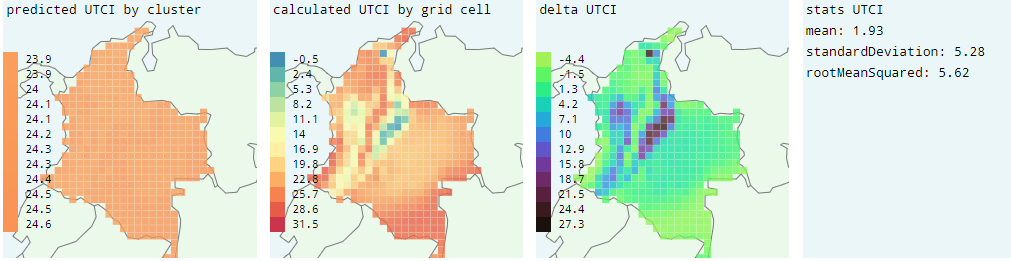


Figure 53 Comfort comparison for KM11TrRh

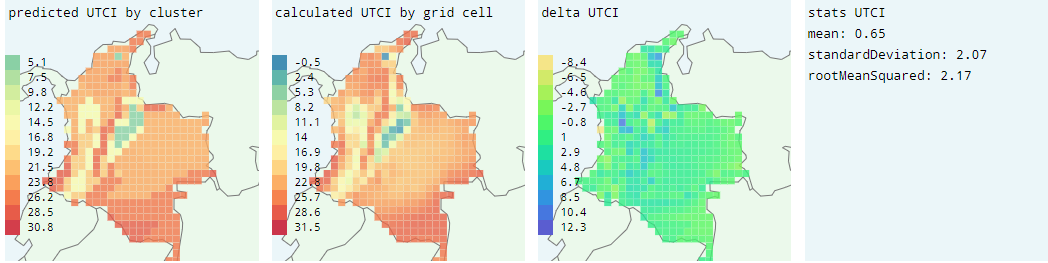


Figure 54 Comfort comparison for BKM11TRhWs

Examining the mapping of the clusters for KM11TrRh and BKM11TRhWs (ranked four and five) provides further confirmation that the first is problematic (Figure 55). In this case some domain knowledge is required to assert that climate varies with altitude and expect the clusters should reflect the split in the Andes Mountains into Colombia’s three smaller Cordillera ranges. KM11TrRh is not capable of this whereas (Figure 55 left), whereas BKM11TRhWs shows in light green two of the three ranges extending from the south-west to the north and north-east (Figure 55 right).

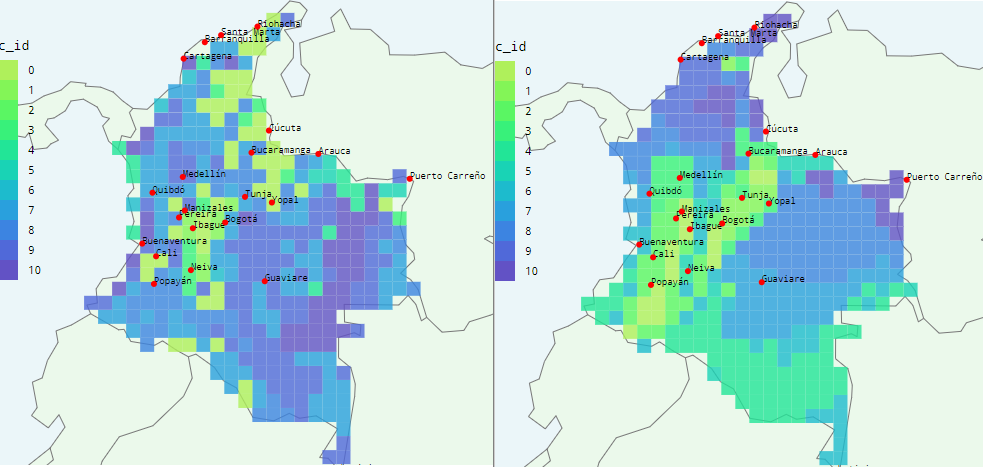


Figure 55 Cluster mapping. Left: KM11TrRh. Right: BKM11TRhWs

The dashboard permits a workflow to be studied monthly, over the course of the year where at each step the topography should be represented. BKM11TRhWs monthly cluster mapping represents the mountain ranges across the year, simultaneously, cluster locations can be seen to subtly shift in an expected cyclic way (Figure 56).

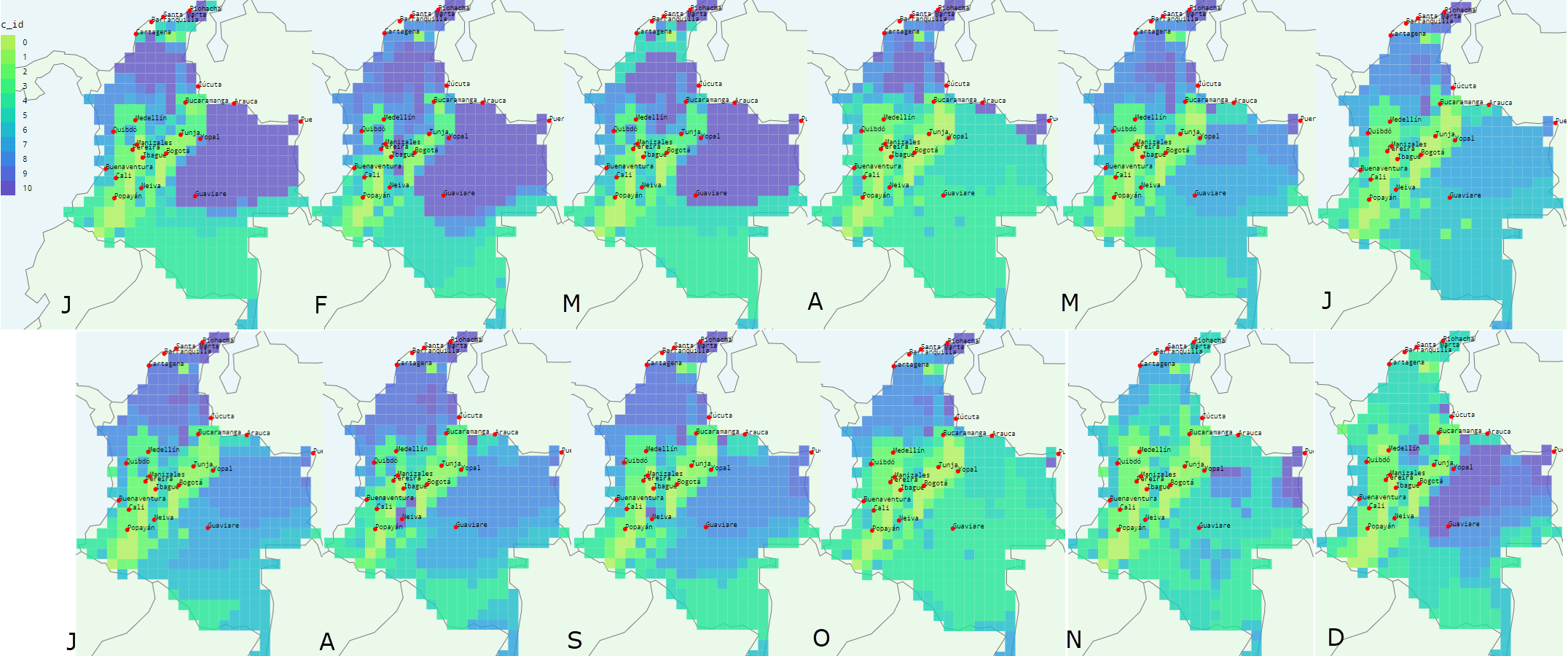


Figure 56 Annual cluster mapping for BKM11TRhWs

Domain knowledge is required to assess if applicable design strategies can be linked with specific locations. The dashboard permits detailed interrogation of strategies, with different levels of granularity across space and time. The least granular flattens all temporal scales into a single period allowing a high-level analysis of the workflow results. For BKM11TRhWs five strategies emerge and these appear logically assigned according to the following observations (based on the authors five years’ experience of Colombian climate as an architect). Only the grid cells at high altitude do not require solar protection (Figure 57 A). Most of the lower elevation zones could be cooled with ventilation (natural and mechanical) (Figure 57 B). Locations in the comfort zone are found at mid-altitude locations (Figure 57 C). Higher altitude regions can benefit from heating using internal gains (energy from occupants and equipment). Lower lying zones with generally dry conditions can be cooled by buildings with high thermal mass that are flushed with cooler air at night (Figure 57 D).

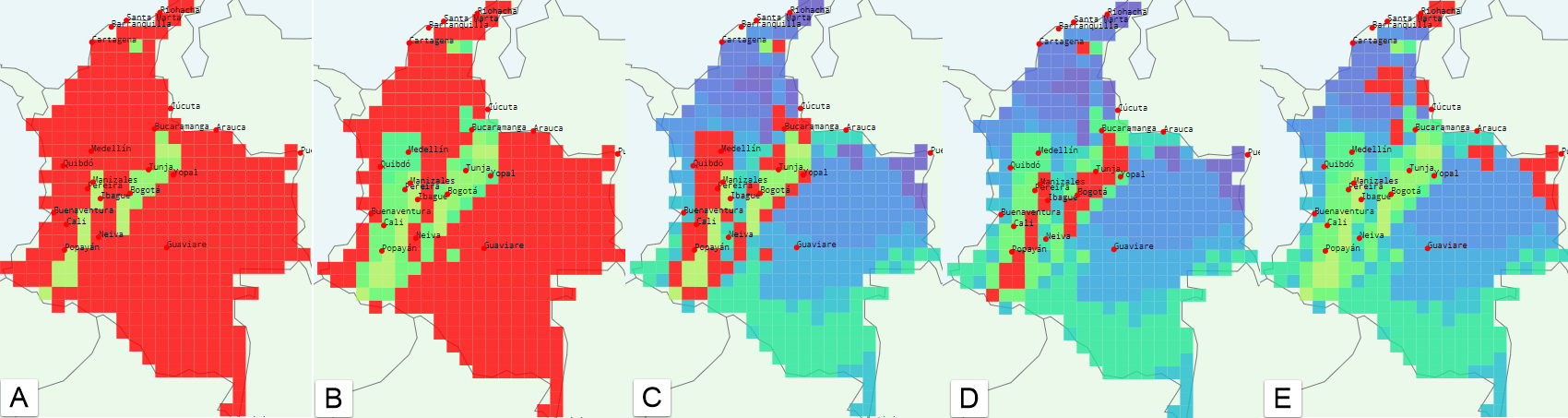


Figure 57 Design strategies and geographic location (red cells) for BKM11TRhWs. A: solar protection. B: cooling by natural and mechanical ventilation. C: comfort zone. D: heating from internal gains. E: cooling by high thermal mass and night cooling.

## Qualitative evaluation

### Opinion by domain experts

The author presented a software walkthrough and a brief project background[[5]](#footnote-5) to a group of trained architects based in Colombia and the UK, which included academic researchers and practicing architects. All participants had either work with or research concepts of thermal comfort and low-energy design and considered to represent domain experts. Following the presentations, the author conducted informal conversations to capture views and opinions. The perspectives to emerge concerned the level of data literacy required of users, the profile of the users, how the software could be customised and the need of finer grained spatiotemporal analysis.

Some of the experts encountered difficulties understanding aspects of the project relating to spatiotemporal data, concepts of multi-dimensionality, clustering, concepts of data vectors with features and the cloud architecture of the project. The notion of data exploration and knowledge-discovery was hard to convey, possibly because time permitted review of only a single set of results. With one participant the potential of a series of clustering studies using a single feature, each then overlaid was conceptually more accessible. Consensus clustering is a term used in the literature (Rhee et al., 2008) to describe this approach and is a potential area for future research.

Certain participants found it hard to understand how or why the proposed user would want such a tool, suggesting that the typical user simply wants to meet all regulatory goals and finish their work as quickly as possible. This contrasted with the authors view of the anticipated user as a diligent and determined designer keen to ensure designs were consuming minimal energy, comfortable for occupants and providing users a unique experience in response to the built spaces, the buildings context and the local climate. Other experts viewed the tool as a central repository of data and analytics which would serve general architectural designers and engineers. These users could be characterised as lacking access to climate data and not specialists in analysing the impact of climate data on their designs.

Experts suggested that the potential user group is not limited to Colombia or the Tropics but could be governmental institutions of any nationality or commercial international businesses. Several participants thought a government body could use the system to develop policies defining design approaches in certain areas and use a version of the software to help individuals adhere to those policies. A suggested alternative user group was large industrial organisations with longer term views and financial planning seeking to save money through fining tuning environmental design strategies in large development and construction projects. The fine tuning could take place based on subtle seasonal variations, at different locations detected by the analysis.

All participants expressed a desire to analyse daily and hourly sets of values to provide increased temporal granularity that would enable search for patterns at the scale of daily periods. Some participants saw this as important in identifying design strategies requiring certain diurnal night-day temperature differentials. Experts also indicate that higher spatial granularity would be useful to detect variations and pattern within individual cities, or across highly local topography such as two sides of a valley.

Participants expressed customisation as an important requirement, both in terms of the ability to weight parameters and to define different comfort zones for different buildings and users. One expert confirmed the use of the temperature, relative humidity and wind speed for the experiments, but he indicated a desire to weight these variables to reflect their relative importance on the perception of comfort. When asked what these weightings should be, the response was vague but involved tuning the correlations. Another area where participants saw customisation as essential is with the design strategies, different participants described the need to capture the way that human perception of comfort varies with demographics, age, sex, socio-economic groups and even user expectations.

## Hypothesis Validation

Quantitative and qualitative evaluation confirmed the dissertation hypothesis; that the system supports design by identifying patterns in the Colombian climate and linking these to appropriate low-energy design strategies. The quantitative evaluation demonstrated that, according to validity indices and comparison to thermal comfort indices, the system generated good clustering solutions. The dashboard and visualisation tools supported this finding, making spatial patterns visible and linking design strategies to instances in space and time when they would be applicable. Qualitative analysis established expert approval, which the author regards as a secondary validation of the hypothesis. Discussions with experts showed a universal desire to use the system, helped evaluate applicability of recommendations made by the system and indicated a consensus view that the artefact offers a marketable service.

The validity indices used in the quantitative evaluation confirmed that the system can generate good clustering solutions, which represented patterns in the climate. Four solutions from the evaluation; KM11TRh, BKM11TRh, KM11TRhWs and BKM11TRhWs, showed good clustering cohesion, separation and dissimilarity indicating the analytic methods, selected number of clusters and chosen parameters were appropriate. Furthermore, all were able to predict UTCI comfort conditions within a root mean squared error of 3.18 (around 1.8 C). Visualisation tools on the dashboard, supported inspection and comparison of geographical mapping of the clusters, which confirmed the usefulness of the solutions based on a domain specific graphical evaluation. Visually inspection of monthly maps across a typical year, enabled the identification of key topographical features. Finally, examining the applicable design strategies and looking at where and when, on the map, they should be applied (within the limits of the authors knowledge of the domain) also confirmed the original hypothesis.

In the qualitative evaluation, all experts asked how they might have access to the tool and suggested additional control requirements needed to adjust the application to meet their own needs. The author regards both responses as an indication that the artefact exhibits fitness for the original purpose. Experts with knowledge of low-energy design in Colombia discussed appropriateness of design-strategies identified by one of the experimental solutions. This input confirmed that KM11TRhWs and BKM11TRhWs identified applicable strategies. All experts agreed that the system, with modification of certain controls and adjustment of the target user group has industrial application. Some of the conversations involved discussing what kind of business model would be needed to offer it as a Climate analysis as a Service (CaaaS).

# Conclusions

## Lessons Learned

### Process and technology

The project, was the authors first application of Agile Model Driven Development (AMDD) for software development. Using AMDD provided a practical understanding of the benefits of a development process that integrates UML and defined a needed structure for the dissertation’s development iterations. Iteration modelling reduced the perceived enormity of each cycle by prioritising requirements, while sequence and class diagrams clarified necessary specifications in response to those requirements. Abstraction of the design through models gave the author a way of managing and integrating the new technology used. The author updated to models and diagrams as each iteration progressed to reflect the implementation and prepare for the next. The UML diagrams included in this document are the result of this process and made summarising the implementation efficient.

AMDD called for a test-driven approach which required the author first learn about unit testing and developing an understanding of the Junit testing framework. Combining unit testing with the AWS SDK proved difficult, required mock testing frameworks which, due to the authors inexperience, slowed the process. Instead, most testing was based on constant integration tests.

During the dissertation the author applied a range of technology that he had not previously experienced, these included Eclipse IDE, Maven, AWS SDK, AWS S3, AWS EMR, Spark, JavaRDD, Dataframes, Jackson JSON library, JavaFx and GMapsFX. The project risk assessment did not anticipate the time implications in learning to use each of these. Developing and debugging JavaRDD and Dataframes in Spark was a challenge requiring the author integrate functional programming with his object orientate background. This new style required deeper understanding of closures (scope and life cycle of variables and methods), the use of chained operations on RDDs and differences in Spark’s behaviour on a local machine and in a distributed cluster.

Unforeseen dependencies in tools added to the underestimations of required time. For example, the author found that the Eclipse IDE with Maven uniform build system better supported the AWS SDK than Java IDEs he was familiar with. In developing the interface, the author found JavaFx was more flexible than JavaSwing, but to integrate mapping tools, JavaFx required GMapsFX. Cloud Computing with AWS required developing knowledge of deploying EMR clusters, which also necessitated learning data manipulation with S3. On reflection, the original project plan did not anticipate the extent of the risks from unknown technology. Future project planning will better account for these risks, allowing more time and better mitigation strategies.

## Industrial Application and Limitations

Key observations that emerged from evaluation with domain experts were; the need for a re-evaluation of the primary users of the system, the importance of allowing users to further customise their analysis jobs and a need for finer grained analysis.

The original design proposed the environmental designer (ED) as the primary user, discussion with experts and their opinions indicated that the typical ED does not possess the data literacy skills required to define the analytic jobs or interpret the results. In fact, the user is probably a data scientist / analyst trained with domain of architectural / environmental design by an architectural or engineering design firm to assist in the process of data driven building design and construction. Alternatively, experts suggested that the user’s employer may be a construction contractor, property developer or a governmental body seeking to develop sustainable design policy. Ultimately all these stakeholders have a common interest, driven by financial motive and ethical outlook to reduce the need for heating and cooling. Experts saw the system could support financial gains by reducing capital required in construction, running costs or by offering higher performing buildings as a product. The ethical position shared by some of these stakeholders is a responsibility to minimise the impacts of climate change by reducing the emissions of the construction industry.

A desire for user customisation indicated that the design strategy management tool and a weighting system for the variables should be prioritised in future development iteration. The design strategy editor exists as a mock-up in the final prototype, with a static representation of a general version of building comfort. Discussions with experts indicated the full potential for the editor in tuning the system, and its design strategies to represent a range of perceived comfort levels. Studies have observed that perception of comfort varies depending on building type, function and demographic. One expert expressed that tools for weighting variables were necessary to allow users to tune the clustering to reflect correlations in the features. Empirical human comfort studies can inform the domain of correlations among variables to determine variable weighting, the system could support this, from a data perspective, by representing correlation between selected variables.

Many participants expressed a need for increasing the granularity of the spatiotemporal data. Including hourly dataset was always a goal for the dissertation. Unfortunately, the hourly data received from the local metrological office for major Colombian cities was inconsistent and required a degree of reformatting, cleaning and generation of missing data beyond the scope of the project. Given the resolution of data available in other regions, similar datasets will be available for Colombia soon. When this data is available the system, in its current state of development, is ready to consume it. Increasing spatiotemporal granularity remains an objective, however, given the challenges of representing the temporal granularity of the monthly results a thorough review of spatiotemporal instance types and best suited datamining methods is essential.

## Knowledge-discovery

The application combines domain expertise, data management and analytics combined in a knowledge-discovery (KD) tool. This provides simple ways to interact, explore and analyse data and through this find new knowledge applicable to a problem. The KD literature emphasises the need for the domain expertise. However, the review with domain experts indicated that domain expertise without working knowledge of the analytic methods used and to a lesser extent data literacy limited KD.

The experimental evaluation showed that KD was possible, but this required a structured approach to define a set of experiments and then compare them to one another. The original requirements for the workflow builder did not include this requirement but, this should be included in the next development iteration. Despite this shortcoming, the system generates results with a structure and format that allowed the author to quickly construct tools to summarise the results and compare them to each other. This exposure of results shows how the system complies with the definition of KD in the literature review.

## Future research

The literature on clustering with climate data is vast, including many analytic methods and tools for determining validity. Most of these were beyond the scope of this dissertation, however, the experimental evaluation shows that the implemented methods can generate good clustering results with practical recommendations for applying low-energy design strategies across different spatiotemporal scales. These positive findings indicate that studying more analytic methods and validity indices would be worthy of further investigation. Future research should consider other independent analytic methods but also more complex workflows that combine multiple machine learning approaches that have proved successful in recent literature. The implemented workflow builder allows selection of analytic methods and configuration by specify parameters. Future research could look at how to combine analytic methods, whereby the output of one becomes the input for the next.

Deeper understanding of the implications of the selected features for clustering is required. This should include examine correlation between them but also the way in which they are normalised. The implemented system left the choice to expert knowledge and all variables were normalised with L2. The results of a principle component analysis may not seem intuitive to a typical user it should be included to support data exploration and choice of variables if required.

The multi-dimensional nature of spatiotemporal (ST) data proved challenging in the analysis and representation of results. The system implemented a single type of ST instance, points, each with a set of features, and sought to find patterns within these. Good clustering solutions were found with point instances but visualising the classified results at different temporal scales was difficult. Visualisations flattened temporal ranges by averaging within clusters and the detail of recommended strategies was lost. A logical next step would be to use time-series instances or raster maps representing local areas, either of which would include a series of features across the period or area of interest and seeking patterns using appropriate analytic methods.

Currently the application focuses on seeking patterns in past climate data. Further research should investigate how the application could integrate data sets from models that attempt to predict future climates, to support building design processes that anticipate climate change.

# 

# REFERENCES CITED

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

Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J. M., & Perona, I. (2013). An extensive comparative study of cluster validity indices. *Pattern Recognition*, *46*(1), 243–256. https://doi.org/10.1016/J.PATCOG.2012.07.021

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

Assunção, M. D., Calheiros, R. N., Bianchi, S., Netto, M. A. S., & Buyya, R. (2015). Big Data computing and clouds: Trends and future directions. *Journal of Parallel and Distributed Computing*, *79*–*80*, 3–15. https://doi.org/10.1016/J.JPDC.2014.08.003

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

Avci Salma, C., Tekinerdogan, B., & Athanasiadis, I. N. (2017a). 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). https://doi.org/10.1016/B978-0-12-805467-3.00004-1

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

AWS Toolkit for Eclipse. (n.d.). Retrieved October 3, 2018, from https://aws.amazon.com/eclipse/

Bahsoon, R., Ali, N., Heisel, M., Maxim, B., & Mistrik, I. (2017). Chapter 1 - Introduction. Software Architecture for Cloud and Big Data: An Open Quest for the Architecturally Significant Requirements. In *Software Architecture for Big Data and the Cloud* (pp. 1–10). https://doi.org/https://doi.org/10.1016/B978-0-12-805467-3.00001-6

Begoli, E., & 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). https://doi.org/10.1109/WICSA-ECSA.212.32

Bostock, M. (2017a). D3.js - Data-Driven Documents. Retrieved June 28, 2018, from https://d3js.org/

Bostock, M. (2017b). TopoJSON. Retrieved May 3, 2017, from https://github.com/topojson/topojson

Bracco, A., Falasca, F., Nenes, A., Fountalis, I., & Dovrolis, C. (2017). Advancing climate science with knowledge-discovery through data mining. *Npj Climate and Atmospheric Science*, *1*(1), 4. https://doi.org/10.1038/s41612-017-0006-4

Buyya, R., Calheiros, R. N., Dastjerdi, A. V., Zhou, A. C., He, B., & Ibrahim, S. (2016). Chapter 18 – eScience and Big Data Workflows in Clouds: A Taxonomy and Survey. In *Big Data* (pp. 431–455). https://doi.org/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. Retrieved November 5, 2017, from http://www.cgiar-csi.org/data/uea-cru-ts-v3-10-01-historic-climate-database

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

Cluster Mode Overview - Spark 2.4.0 Documentation. (n.d.). Retrieved November 21, 2018, from https://spark.apache.org/docs/latest/cluster-overview.html

Dagade, V., Lagali, M., Avadhani, S., & Kalekar, P. (2015). Big Data Weather Analytics Using Hadoop. *International Journal of Emerging Technology in Computer Science & Electronics*, *14*(2), 976–1353. Retrieved from https://pdfs.semanticscholar.org/f2e4/918444be9b30f29132e93ce02d29ccf26eda.pdf

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), 1765–1782. https://doi.org/10.1175/1520-0442(1996)009<1765:DOMCZI>2.0.CO;2

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

El Fazziki, A., Sadiq, A., Ouarzazi, J., & Sadgal, M. (2015). A multi-agent framework for a hadoop based air quality decision support system. In *CEUR Workshop Proceedings* (Vol. 1381, pp. 45–59).

Faghmous, J. H., & Kumar, V. (2014). Spatio-temporal Data Mining for Climate Data: Advances, Challenges, and Opportunities. In W. Chu (Ed.), *Springer Berlin Heidelberg* (Vol. 1, pp. 83–116). https://doi.org/10.1007/978-3-642-40837-3\_3

Fang, W., Sheng, V. S., Wen, X., & Pan, W. (2014). Meteorological data analysis using MapReduce. *The Scientific World Journal*, *2014*, 646497. https://doi.org/10.1155/2014/646497

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

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

Fovell, R. G., & Fovell, M. Y. C. (1993). Climate zones of the conterminous United States defined using cluster analysis. *Journal of Climate*. https://doi.org/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*, 11–23. Retrieved from 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

González, O. (1998). *Metodología para el Calculo del Confort Climático en Colombia: NOTA TECNICA DEL IDEAM.* Bogota. Retrieved from 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*. https://doi.org/10.1007/s004840050118

http-server: a command-line http server. (2018). Retrieved October 3, 2018, from https://www.npmjs.com/package/http-server

Hudson, R. (2018). Big Climate Data Analytics: Effective Knowledge Discovery from Colombia’s Weather Data. Retrieved from http://lacunae.io

Hudson, R. (2018). cluster-colombia Evaluation Summary. Retrieved October 28, 2018, from http://lacunae.io/

Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, *31*(8), 651–666. https://doi.org/10.1016/J.PATREC.2009.09.011

Jain, A. K., & Dubes, R. C. (1988). *Algorithms for clustering data*. Prentice Hall. Retrieved from https://dl.acm.org/citation.cfm?id=46712

Jain, H., & Jain, R. (2017). Big data in weather forecasting: Applications and challenges. In *2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC)* (pp. 138–142). IEEE. https://doi.org/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. Retrieved November 5, 2017, from http://srtm.csi.cgiar.org

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

Jayaratne, M., Alahakoon, D., Silva, D. De, & Yu, X. (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). https://doi.org/10.1109/IECON.2017.8217465

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

Jones, P., & Harris, I. (2008). Climatic Research Unit (CRU) time-series datasets of variations in climate with variations in other phenomena. Retrieved November 5, 2017, from http://catalogue.ceda.ac.uk/uuid/3f8944800cc48e1cbc29a5ee12d8542d

Karpatne, A., Atluri, G., Faghmous, J. H., Steinbach, M., Banerjee, A., Ganguly, A., … Kumar, V. (2017). Theory-Guided Data Science: A New Paradigm for Scientific Discovery from Data. *IEEE Transactions on Knowledge and Data Engineering*, *29*(10), 2318–2331. https://doi.org/10.1109/TKDE.2017.2720168

KMeansModel (Spark 2.3.2 JavaDoc). (n.d.). Retrieved October 18, 2018, from https://spark.apache.org/docs/latest/api/java/org/apache/spark/mllib/clustering/KMeansModel.html

Kodinariya, T. M., & 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*.

Lechner, N. (2014). *Heating, Cooling, Lighting : Sustainable Design Methods for Architects* (4th ed.). Hoboken, New Jersery: Wiley.

Liu, F., Tong, J., Mao, J., Bohn, R., Messina, J., & Leaf, D. (2011). *NIST Cloud Computing Reference Architecture Recommendations of the National Institute of Standards and Technology*. Retrieved from https://ws680.nist.gov/publication/get\_pdf.cfm?pub\_id=909505

Liu, Y., & 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), 1–12. https://doi.org/10.1029/2004JC002786

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

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

Luna-Romera, J. M., Martínez-Ballesteros, M. D. M., García-Gutiérrez, J., & Riquelme-Santos, J. C. (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)*. https://doi.org/10.1007/978-3-319-44636-3\_15

Manzano-Agugliaro, F., Montoya, F. G., Sabio-Ortega, A., & García-Cruz, A. (2015). Review of bioclimatic architecture strategies for achieving thermal comfort. *Renewable and Sustainable Energy Reviews*, *49*, 736–755. https://doi.org/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), 2278–2661. https://doi.org/10.9790/0661-17320612

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

Milne, M., Liggett, R., & Benson, A. (2009). Climate Consultant 4.0 develops design guidelines for each unique climate. *American Solar Energy Society Meeting*. Retrieved from http://www.energy-design-tools.aud.ucla.edu/papers/ases09-milne.pdf

MLlib: Main Guide - Spark 2.3.2 Documentation. (n.d.). Retrieved October 16, 2018, from https://spark.apache.org/docs/latest/ml-guide.html

Netzel, P., & Stepinski, T. (2016). On Using a Clustering Approach for Global Climate Classification. *Journal of Climate*, *29*(9), 3387–3401. https://doi.org/10.1175/JCLI-D-15-0640.1

Nikolaou, T. G., Kolokotsa, D. S., Stavrakakis, G. S., & Skias, I. D. (2012). On the application of clustering techniques for office buildings’ energy and thermal comfort classification. *IEEE Transactions on Smart Grid*. https://doi.org/10.1109/TSG.2012.2215059

Olgyay, V., & 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*, 2265–2300. https://doi.org/10.1016/j.rser.2007.05.001

Piech, C. (2013). K Means Stanford CS221. Retrieved October 22, 2018, from http://stanford.edu/~cpiech/cs221/handouts/kmeans.html

RDD Programming Guide - Spark 2.4.0 Documentation. (n.d.). Retrieved November 21, 2018, from https://spark.apache.org/docs/latest/rdd-programming-guide.html

Rhee, J., Im, J., Carbone, G. J., & Jensen, J. R. (2008). Delineation of climate regions using in-situ and remotely-sensed data for the Carolinas. *Remote Sensing of Environment*, *112*(6), 3099–3111. https://doi.org/10.1016/j.rse.2008.03.001

Rodenburg, B., & Maria Fiore, M. (2017). Detecting Weather Twins using Apache Spark. Retrieved May 29, 2018, from https://event.cwi.nl/lsde/2017/showcase\_n2.shtml

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

Terpilowski, R. (n.d.). GMapsFX. Retrieved October 18, 2018, from https://rterp.github.io/GMapsFX/

Thinsungnoen, T., Kaoungku, N., Durongdumronchai, P., Kerdprasop, K., & Kerdprasop, N. (2015). The Clustering Validity with Silhouette and Sum of Squared Errors. In *The Proceedings of the 2nd International Conference on Industrial Application Engineering 2015*. https://doi.org/10.12792/iciae2015.012

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

Wentz, F., Scott, J., Hoffman, R., Leidner, M., Atlas, R., & Ardizzone, J. (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]. Retrieved November 5, 2017, from www.remss.com/measurements/ccmp

Wiwie, C., Baumbach, J., & Röttger, R. (2015). Comparing the performance of biomedical clustering methods. *Nature Methods*. https://doi.org/10.1038/nmeth.3583

Zaharia, M., Das, T., Li, H., Hunter, T., Shenker, S., & Stoica, I. (2013). Discretized Streams: Fault-Tolerant Streaming Computation at Scale. *Sosp*, (1), 423–438. https://doi.org/10.1145/2517349.2522737

Zaharia, M., Wendell, P., Konwinski, A., & Karau, H. (2015). *Learning Spark*. O’Reilly Media, Inc. Retrieved from https://www.oreilly.com/library/view/learning-spark/9781449359034/

Zscheischler, J., Mahecha, M. D., & Harmeling, S. (2012). Climate classifications: The value of unsupervised clustering. In *Procedia Computer Science* (Vol. 9, pp. 897–906). https://doi.org/10.1016/j.procs.2012.04.096

APPENDICES

###### Approved Proposal

**The Project Aims and Objectives:**

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.

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. The industrial approach is to install heating and cooling which is expensive, costly to run and produces emissions. Low energy construction strategies exist that can minimize or remove the need for heating and cooling through for example; orientation of buildings, sizing and positioning of openings, choice of materials and use of passive heating and ventilation.

These construction strategies are climate responsive. In the Northern and Southern hemispheres, it is relatively easy to identify what strategy to apply where because of the seasonality and lack of regional variation. The variability of weather in Colombia means it is very hard to define an approach for a specific location. Weather records from national, automated stations was recently made available by Colombia’s national weather agency IDEAM.

This proposal aims to design and implement a big data system that enables the analysis and visualization of Colombia’s weather data using clustering, self-organizing maps and delta-maps. The goal is to use these techniques to generate localized approaches to building design and construction can respond to the unique weather conditions in Colombia.

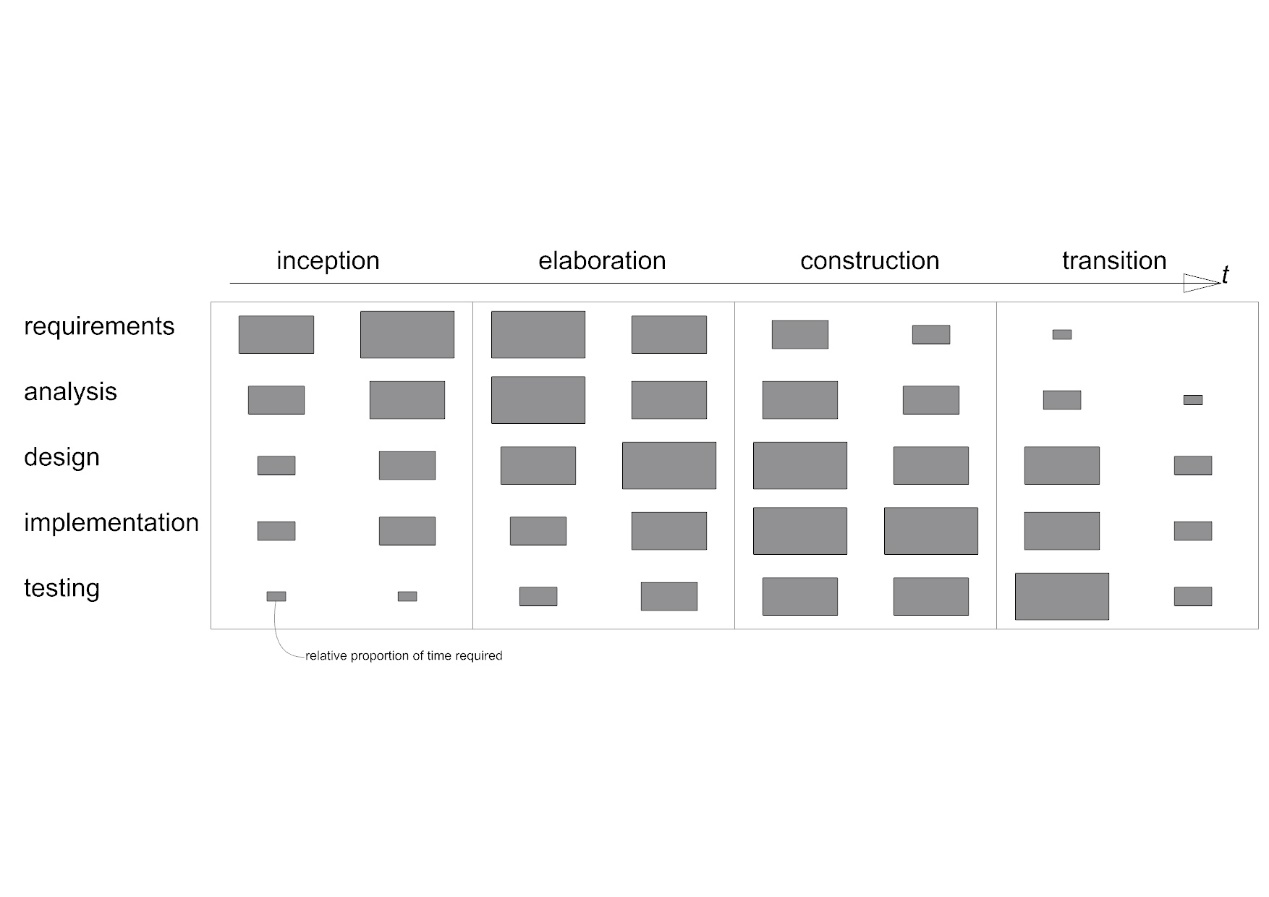
To achieve this goal a big data system is proposed that follows current best practices for the storage, processing, analysis, management and visualization of the data. Specific focus will be on enabling the analytics and visualization that enables KD through Data mining. Knowledge from the data will 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.

|  |  |
| --- | --- |
| **Step** | **Short Description** |
| Hypothesis | KD techniques combined with a big weather data framework can help define localized approaches to building design and construction that improve living conditions and reduce energy consumption in Colombia. |
| Research Methods | **Literature review:** Big Data Architecture, Big Data Analytics methods, KD process models, Spatio-temporal data mining, Data mining techniques applied to meteorology  **Developing the application:** Agile Model Driven Development |
| IT Artefact | An application that facilitates big data analytics for Colombia’s recently released weather data. Through analytics and visualisation, the application should enable data exploration and KD with the goal of providing recommendations for construction strategies dependent on geographical location and related historical weather data.  The application should also provide efficient storage, processing, management and security. |
| Evaluation | Verification, validation and testing of the application using statistical comparisons and review by domain experts  Checking if useful localised construction approaches can be generated.  Validation of application output:  Statistical comparison of different knowledge discovery methods applied  Quality measures for methods applied. (Distance metrics for clustering)  Test cases – identified by domain experts  Interpretation of results by domain experts |

**Project Outline**

**Literature search and review** to identify of KD principles applicable in weather, methods for evaluating KD techniques, existing big data architecture and frameworks in scientific applications and specifically in climatology and meteorology.

**Development phases and workflows time distribution:**

****

**System development process:**

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |  |

**Evaluation:** Results and output of the application evaluated using statistical methods identified for each of the implemented knowledge discovery methods. Application is evaluated through verification, validation and testing – tests identified during the requirements specification and revisited through the prototyping stages. Domain expert(s) presented with a series of studies and results from the application, opinion of experts captured and summarised.

**Literature Survey / Resources’ List:**

Big data architecture and frameworks:

* Software architecture for cloud and big data applications (Bahsoon, Ali, Heisel, Maxim, & Mistrik, 2017)
* Attempts to generalise big data workflows in eScience (Buyya et al., 2016)
* Workflows for scientific big data management applicable weather data (Rodriguez & Buyya, 2017).
* Big data feature model and Domain-Driven Design is discussed (Avci Salma, Tekinerdogan, & Athanasiadis, 2017b)

**Knowledge discovery methodology:**

Knowledge Discovery in Data is broadly defined as (Begoli & Horey, 2012); 1. Collection, storage and organisation of data. 2. Understanding and application of analytic methods. 3. Understanding the problem domain

Three principles for KDD (Begoli & Horey, 2012):1. Support of analysis methods. 2. One size does not fit all. 3. Make data accessible

**Knowledge discovery applicable methods:**

* Clustering methods for climate classification (Forsythe et al., 2015) (Netzel & Stepinski, 2016)
* Self-organising maps (Jayaratne et al., 2017) (Y. Liu et al., 2006).
* Delta-maps (Fountalis et al., 2014).
* Need for better visualisation and user interaction (Assunção, Calheiros, Bianchi, Netto, & Buyya, 2015).

**Scholarly Contributions of the Project**

The project proposes a framework for extracting useful knowledge from weather data in Colombia. Through the application of spatiotemporal knowledge discovery methods the project aims to determine localised patterns that can indicate appropriate climate responsive strategies for the design and construction of buildings. Implementing these strategies can improve building performance in terms of energy consumption and comfort for occupants. The proposed solution will contribute to environmental design (architecture and engineering of buildings) by supporting decision making in the design process with knowledge acquired from weather data.

The test application aims to combine knowledge discovery and big data technologies to identify patterns in complex tropical climates and use these to improve occupant comfort and energy performance of buildings in the region. Colombia is the test case but similar situations in other tropical regions exist, the project seeks to generalise the findings, so the methodologies used may be transferred to other contexts.

**Description of the Deliverables:**

Literature review

UML documentation of the system development process:

* Requirements list
* Use case diagrams
* Package diagrams
* Class, object and component diagrams
* Deployment diagram
* Sequence diagrams
* Test plans
* Test cases

Climate big data analytic system that enables knowledge-discovery.

Example recommendations for construction strategies given a geographical location and the associated weather data.

Visualisation of results / output.

Set of statistical test results of outputs from the application.

Analysis of results and system by domain expert(s).

Generalised, transferable description of system architecture

**Evaluation Criteria:**

The big question to be evaluated is: can the system support environmental design decision making based on weather data?

This will be addressed in the following ways.

* Statistical comparison of the different knowledge discovery methods used
* Statistical comparison between individual knowledge discovery methods
* Opinion by domain experts(s) – presentation of results and analysis

Statistical methods will be identified as part of the literature review of the knowledge discovery techniques. For example, to compare clustering results distance metrics would be used. Secondary evaluation will involve the verification, validation and testing of the IT artefact. Working with a reduced data set, independent component tests will be defined and executed during development. Components will be integrated and the system tested for issues in component interaction and interface problems and if the system meets functional and non-functional requirements. The system will undergo evaluation with the full dataset to check for omission in the requirements definition and for overall performance.

**Resource Plan:**

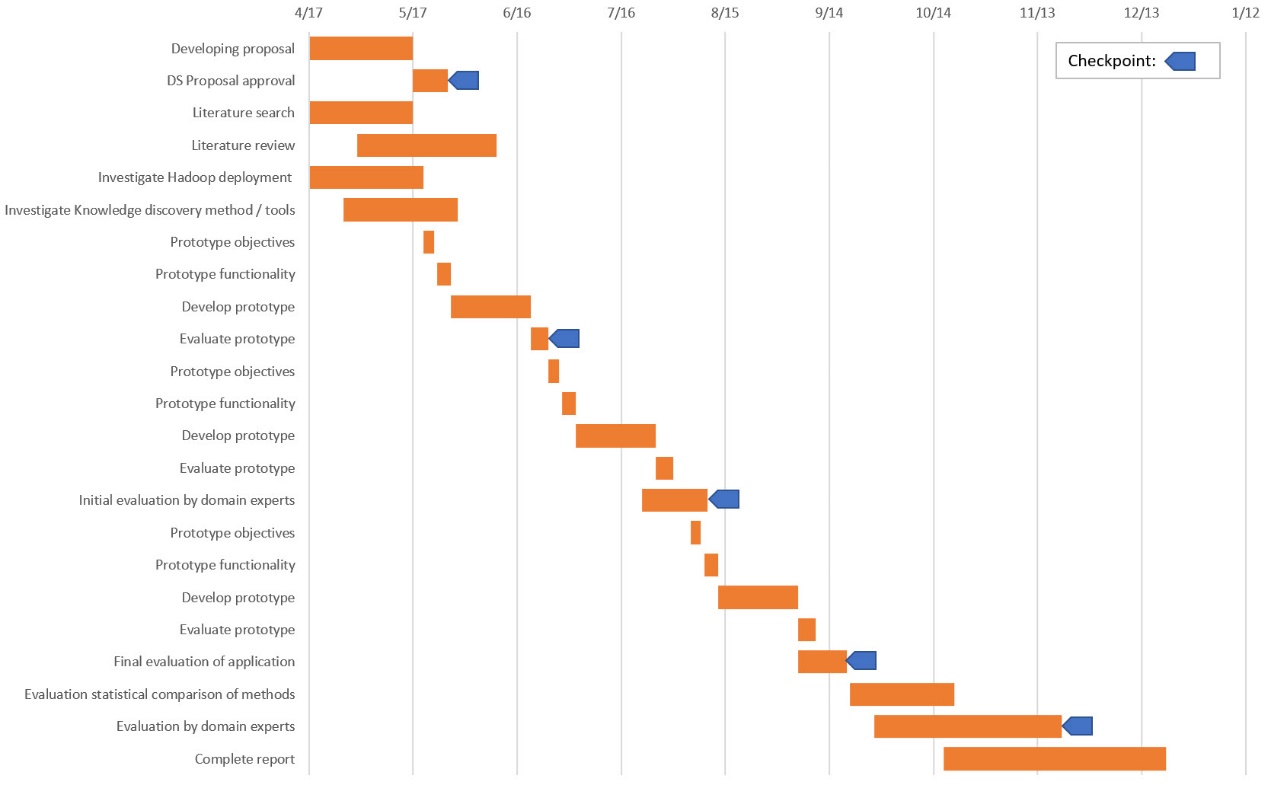
Cloud Hadoop deployment.

Personal computer.

Domain experts to be invited for evaluation.

Colombian Weather Dataset from IDEAM (in hand).

**Project Plan and Timing**

****

**Risk Assessment:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk** | **Indicator** | **Impact 1-15 (probability x effect)** | **Alleviation strategy** |
| Underestimation of development times | Milestone 2 overrun | 12 | Limit scope of requirements, by setting SMART goals |
| Underestimation of the size of the software | Milestone 2/3 overrun, failure to fix defects | 12 | Use of standard tools and components where possible. Investigate best practice with existing tools and frameworks |
| Technical difficulties with Hadoop and components | Problems with data ingestion / basic analysis | 4 | Investigate use of off-the-shelf tools as part of literature search. Start prototyping early and prioritise experiments with Hadoop |
| Technical difficulties with KD methods in Hadoop framework | Problems with first attempts at analytics | 4 | Investigate the use of off-the-shelf tools as part of literature search. Include experiments with KD methods as early as possible in the prototyping process. |
| Onsite hardware failure | Personal laptop failure | 5 | Backups local and offsite, establish access to secondary computer |
| Offsite hardware / platform failure | No access to cloud services | 5 | Ensure chosen platform has deployment on multiple servers and sites and strategy for backups |
| DA leaves Liverpool | notification from the university | 1 | Read Liverpool / Laureate protocols for this circumstance so a replacement can be quickly found. Continue working with initial DA if permitted. |

**Quality Assurance:**

Five milestones are identified in the project plan to coincide with acceptance of the proposal and end of prototype and evaluation phases. Each prototype development phase will be structured around agile model driven development using UML, leveraging continuous documentation to define, record and monitor the process. Tests at the component, integration and system tests will be defined and used to drive application development. Evaluation by domain experts is planned at two milestones. Regular review by DA and at milestones.

**References (see main references section)**

###### Class Diagrams

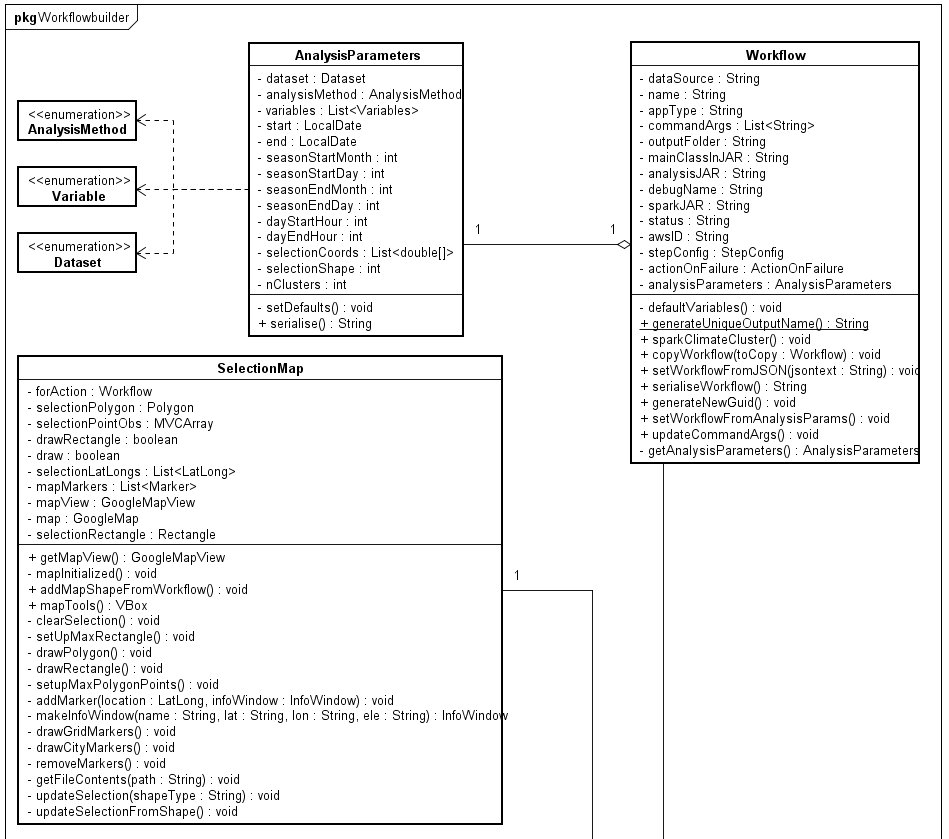


Figure 58 Workflowbuilder package class diagrams for AnalysisParameters, Workflow SelectionMap classes



Figure 59 Workflowbuilder package class diagrams for GUIWorkflowbuilder class

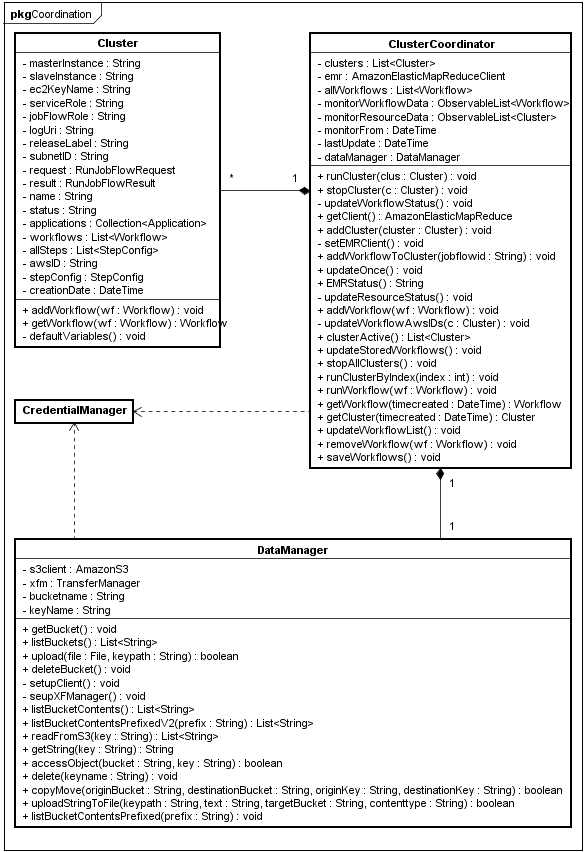


Figure 60 Coordination package class diagrams for Cluster, ClusterCoordinator and DataManager classes

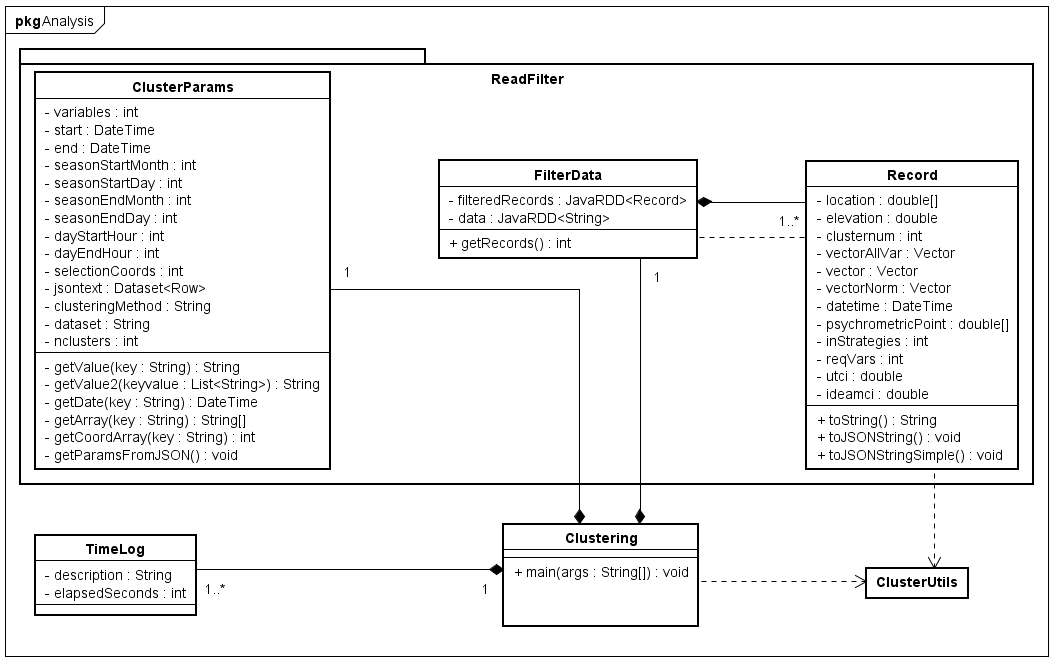


Figure 61 Analysis package and ReadFilter sub-package class diagrams for ClusterParams, FilterData, Record and TimeLog classes

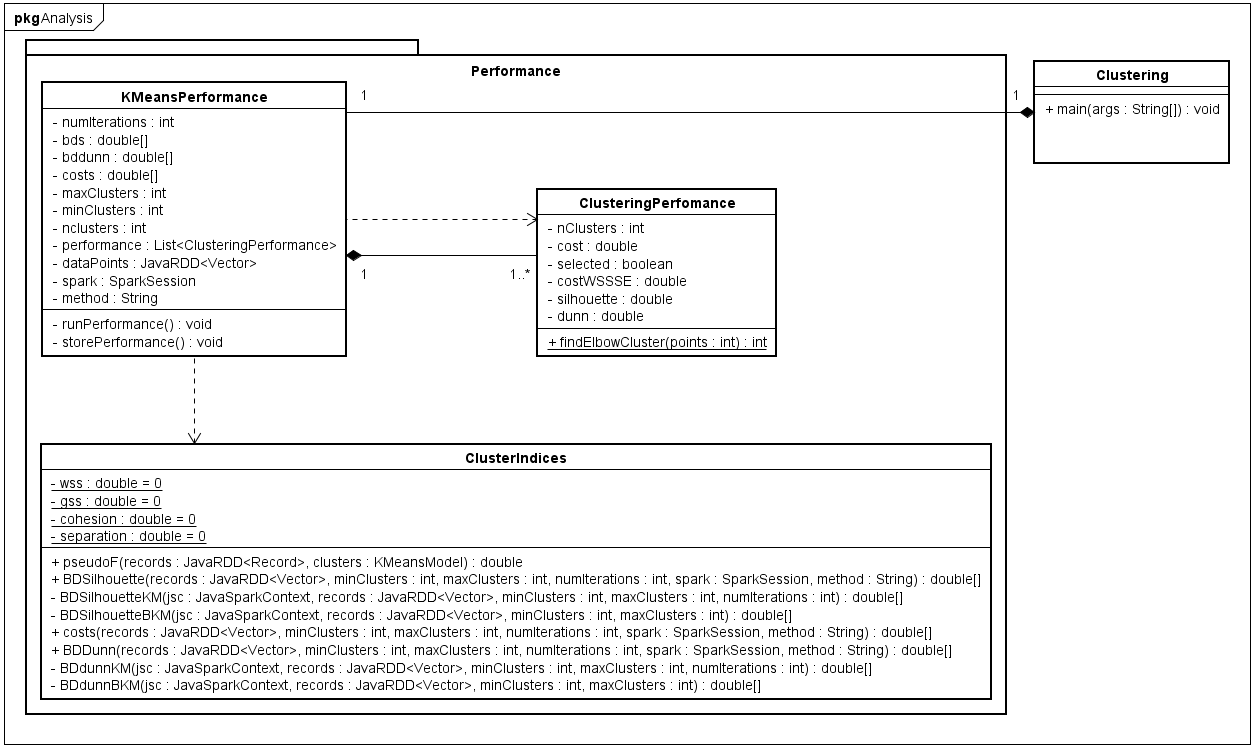


Figure 62 Analysis package and Performance sub-package class diagrams for KMeansPerfomance, ClusteringPerformance, and ClusterIndices classes

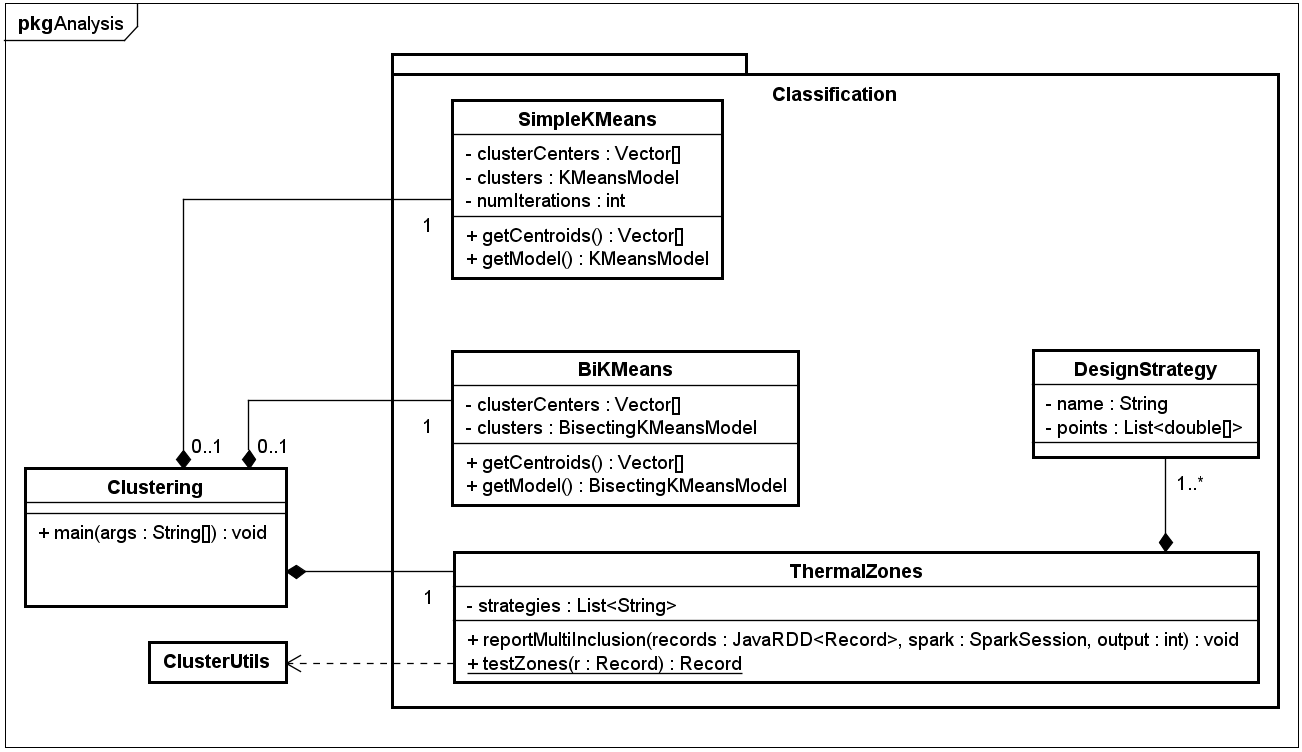


Figure 63 Analysis package and Classification sub-package class diagrams for SimpleKMeans, BiKMeans, DesignStrategy and ThermalZones classes

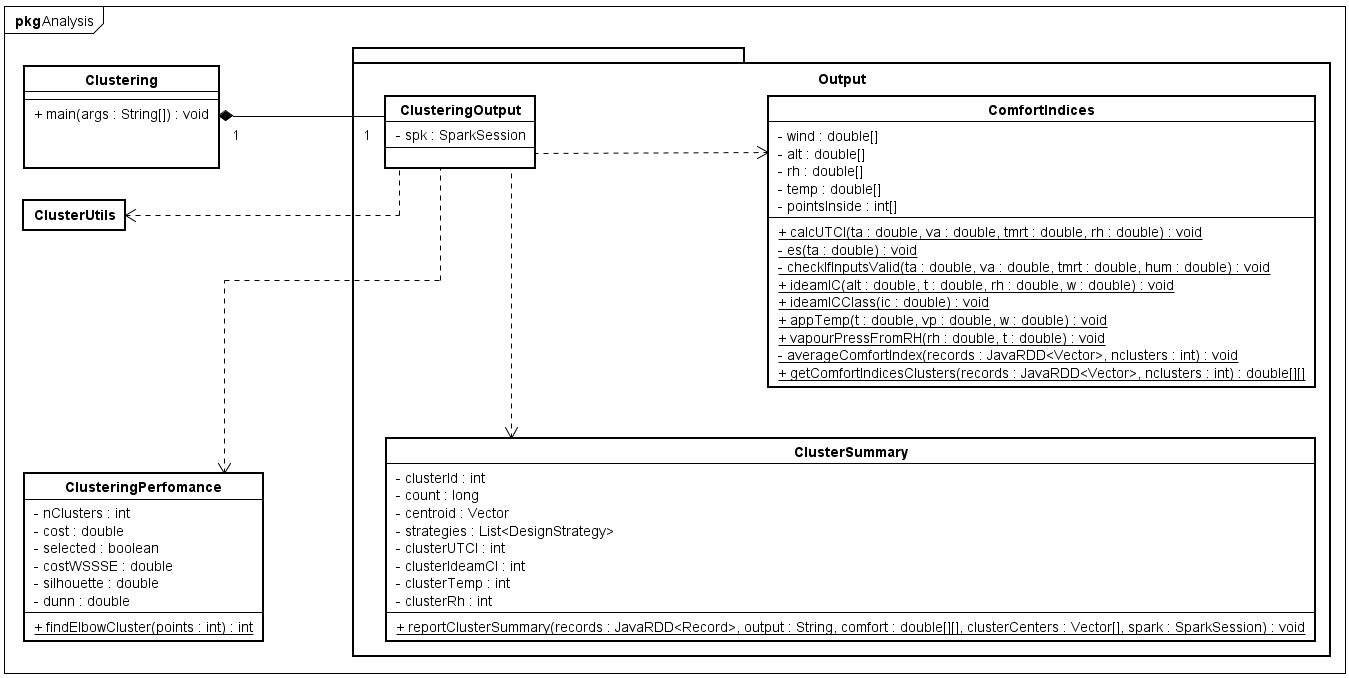


Figure 64 Analysis package and Output sub-package class diagrams for ClusteringOutput, ComfortIndices, ClusteringPerformance and ClusterSummary classes

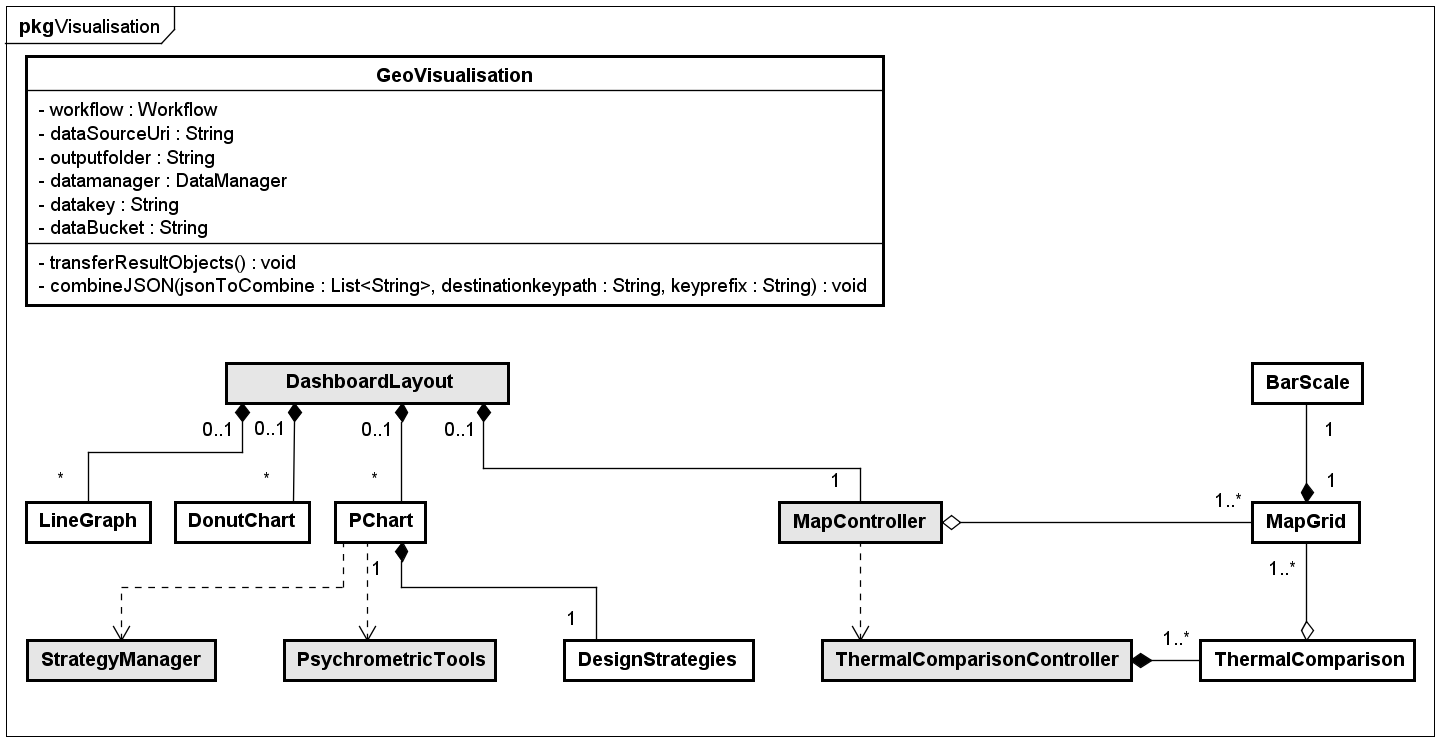


Figure 65 Visualisation package and class diagram for GeoVisualisation class

1. Source code is available: <https://github.com/rolyhudson/climacolombia.git>. [↑](#footnote-ref-1)
2. AWS S3 does not technically use “folders”, instead resources are located using keypaths separated by ‘/’ where the last part refers to a specific object. [↑](#footnote-ref-2)
3. Full, optimized results are available <http://lacunae.io/> (Hudson, 2018) [↑](#footnote-ref-3)
4. Full results are available: [http://lacunae.io/](http://lacunae.io/%20) (Hudson, 2018) [↑](#footnote-ref-4)
5. Presentation is available at [http://lacunae.io/](http://lacunae.io/%20) (Hudson, 2018) [↑](#footnote-ref-5)