Architectural Design Exploration of Low-Exergy (LowEx) Buildings in the Tropics

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZURICH
(Dr. sc. ETH Zurich)

presented by

KIAN WEE CHEN

BA (Arch) Hons, M (Arch), National University of Singapore

born on 02.10.1986

citizen of Malaysia

accepted on the recommendation of

Prof. Dr. Arno Schlueter
Chair of Architecture and Building Systems,
Institute of Technology in Architecture (ITA),
Faculty of Architecture ETH Zurich

Asst. Prof. Dr. Patrick Janssen
Department of Architecture,
National University of Singapore

2015
# Contents

PUBLICATIONS AND CONFERENCES DURING PHD THESIS WORK .............................. V

ACKNOWLEDGEMENT .......................................................................................... VI

ABSTRACT ........................................................................................................ VII

KURZFASSUNG ................................................................................................. X

1 INTRODUCTION ............................................................................................... 13

1.1 Motivation and background ......................................................................... 13

   1.1.1 LowEx space conditioning ................................................................. 14

   1.1.2 Building Performance Simulation (BPS) ............................................ 15

1.2 Problem statement ....................................................................................... 16

1.3 Aim and research questions ......................................................................... 17

1.4 Method ......................................................................................................... 17

1.5 Contribution ................................................................................................ 18

1.6 Organisation of thesis .................................................................................. 18

2 LOW-EXERGY (LOWEX) COOLING SYSTEMS FOR SEPARATION OF SENSIBLE AND LATENT COOLING ......................................................................................... 21

2.1 Space conditioning ..................................................................................... 21

2.2 Application of exergy in space conditioning ............................................. 22

   2.2.1 Vapour-compression refrigeration cycle ......................................... 22

   2.2.2 LowEx approach in moderate climates ........................................... 24

   2.2.3 LowEx approach in hot and humid climates .................................... 26

   2.2.4 Conventional cooling system in Singapore ...................................... 27

2.3 SSLC system in research .......................................................................... 28

   2.3.1 Radiant cooling panels ................................................................. 28

   2.3.2 Decentralised Ventilation Unit (DVU) ........................................... 31

   2.3.3 SSLC with radiant cooling panels and DVU ............................... 33
3 MODELLING RADIANT COOLING PANELS AND DECENTRALISED VENTILATION UNIT (DVU)

3.1 Modelling approach

3.2 Modelling and evaluating radiant cooling panels and DVU

3.3 Calculations

3.3.1 Demand

3.3.2 Supply

3.3.3 Application of calculations in building design

4 DESIGN METHOD FOR BALANCING THE ARCHITECTURAL DESIGN AND LOW-EXERGY (LOWEX) COOLING SYSTEM

4.1 Balancing architectural design and LowEx cooling system using optimisation algorithms

4.2 Review of optimisation algorithms in building design

4.2.1 Design generation

4.2.2 Design evaluation

4.2.3 Feedback

4.2.4 Data analysis

4.3 Proposed multi-objective balancing process

4.3.1 Design generation

4.3.2 Design evaluation

4.3.3 Feedback

4.3.4 Data analysis

4.4 Design method

5 PROTOTYPE TOOL FOR IMPLEMENTATION OF DESIGN METHOD

5.1 Concept for the prototype tool

5.1.1 Workflow

5.1.2 Scientific workflow management tool
5.1.3 Graph-based interface for composition of workflows ........................................ 67
5.1.4 Interoperability and collaboration between designers from different domains .......... 68
5.2 Prototypical tool using Kepler workflows .......................................................... 69
  5.2.1 The Extensible Markup Language (XML) schema for the transfer of data between applications ......................................................................................... 70
  5.2.2 Integration of applications for execution of workflow templates ......................... 71
5.3 Workflow templates .............................................................................................. 73
  5.3.1 Encoding workflow template ............................................................................ 73
  5.3.2 Optimisation workflow template ...................................................................... 76
  5.3.3 Analysis workflow template ........................................................................... 80
  5.3.4 Visualisation workflow template ..................................................................... 83
6 CASE STUDY ............................................................................................................. 85
  6.1 Encoding stage ..................................................................................................... 86
  6.2 Optimisation stage .............................................................................................. 88
  6.3 Data analysis stage ............................................................................................. 91
  6.4 Results ................................................................................................................ 92
    6.4.1 Design clusters with balanced performances .................................................. 92
    6.4.2 Design clusters with best cooling electricity consumption performance .......... 94
    6.4.3 Design cluster with best cost performance ..................................................... 96
    6.4.4 Design cluster with the best daylight performance ........................................ 97
    6.4.5 Design clusters that are unable to employ LowEx cooling systems ................ 98
  6.5 How the design method influences the early design stages .................................. 100
7 SUMMARY AND DISCUSSION .................................................................................. 103
  7.1 Summary ............................................................................................................. 103
  7.2 Improvement to the analysis stage ..................................................................... 104
  7.3 Scalability of the design tool .............................................................................. 105
    7.3.1 Design generation ......................................................................................... 105
    7.3.2 Design evaluation ......................................................................................... 106
APPENDIX

7.4 Appendix A – $T_{Deq}, \Delta T$, and $SF$ for ETTV formula according to different scenarios

7.5 Appendix B – Workflow components developed for the workflow templates

7.5.1 Appendix B1 – Workflow components for manipulating the XML schema

7.5.2 Appendix B2 – Integrating Houdini3D with Kepler

7.5.3 Appendix B3 – Integrating Radiance with Houdini3D

7.5.4 Appendix B4 – Integrating the evaluation method for radiant cooling panels and DVU

7.5.5 Appendix B5 – Integrating envelope material cost calculation with Houdini3D

7.5.6 Appendix B6 – Integrating the design evaluation procedures in Kepler

7.5.7 Appendix B7 – Integrating scikit-learn K-means in Kepler

7.5.8 Appendix B8 – Integrating PyMF Archetypal Analysis in Kepler

7.5.9 Appendix B9 – Integrating matplotlib PCP in Kepler

7.5.10 Appendix B10 – Integrating NSGA2 in Kepler

7.6 Appendix C – Kepler workflow template settings for the case study

FIGURES

TABLES

BIBLIOGRAPHY
Publications and Conferences during PhD thesis work


Acknowledgement

This research is only possible with the help of many people I would like to thank.

- First of all, Prof. Arno Schlueter and Dr. Patrick Janssen for the guidance, support and help through this research.

- My colleagues in the Low Exergy Module, Future Cities Laboratory: Dr. Forrest Meggers, Dr. Matthias Mast, Dr. Marcel Bruelisauer, Dr. Adam Rysanek and Portia Murray for giving me advise on exergy related topics. Extra thanks to Matthias and Adam for reading my chapters on exergy to ensure the information is correct.

- All my colleagues who have worked on the BubbleZERO project: Li Cheng, Dr. Rupesh Iyengar, Esmail Saber and Clayton Miller.

- My fellow PhDs who I frequently discussed my PhD topic with: Yazid Ninsalam, Ervine Lim and Poh Jia Hou.

- Lastly, all my family and friends who have supported me along the way through this 4 PhD years.
Abstract

Buildings are major contributors to global energy consumption and Green House Gas (GHG) emissions. The design and construction of energy-efficient buildings is one of the key mitigation strategies in the fight against climate change. In the hot and humid climate of Singapore, 40–50% of the energy consumption of a building is used for space conditioning. Efficient space conditioning can greatly reduce the energy consumption of a building. Sensible and latent cooling are the two main cooling processes in hot and humid climate. Sensible cooling requires temperatures 14-18°C to maintain the interior temperature at 23-25°C, while latent cooling requires 8°C to dehumidify the interior air to maintain thermal comfort for the occupants. In conventional air-based space conditioning, these two processes are not separated. As a result, 8°C is used for both sensible and latent cooling.

The concept of exergy assesses the quality of the energy flow. The low-exergy (LowEx) approach aims to design cooling systems that reduce the use of high quality energy and increase the use of low quality energy, such as the use of higher temperatures that are closer to the room temperature, to provide cooling to the space. The Separation of Sensible and Latent Cooling (SSLC) splits the load by using two chillers, one chiller to provide higher temperature for sensible cooling and one for latent cooling. The consumption of electricity, which is a high quality energy, is reduced as the sensible chiller produces higher temperatures, and the latent chiller only works to dehumidify the required volume of air for ventilation and maintaining the humidity in the space.

However, there are architectural implications when these LowEx cooling systems are employed. For example, one of the major limitations of employing these systems is that the sensible load of the space needs to be low to be within the cooling capacity. LowEx cooling systems that use high temperature cooling components have a lower cooling capacity than conventional cooling systems because of the use of higher temperatures. The sensible cooling load is dependent on the solar heat gain from the envelope and on the internal load, which includes occupancy, lighting, and equipment gains. Architects are required to factor these constraints into their architectural design from the early design stages, or else it might not be feasible to employ LowEx cooling systems. The
early design stages inevitably become more demanding and complicated when architects have to factor LowEx cooling systems into their design. The objective of this research is therefore to facilitate the integration of LowEx cooling systems into the early architectural design stages.

Assuming an integrated design process during which architects work in close collaboration with the Heating, Ventilation, and Air-Conditioning (HVAC) designers, a design method that consists of three main stages is proposed to achieve the objective. The first stage is the encoding stage, which in turn consists of three steps. The first step requires the HVAC designers and architects to select the LowEx cooling systems according to the HVAC system requirements and properties. The second step is to decide how to model and evaluate the systems with the architectural design. An architectural design change will affect not only the performance of the LowEx cooling systems, but also other performances, such as daylight. Thus, in the third step the design concept is formulated as a Multi-Objective Optimisation Problem (MOOP). It is then encoded into a design generation procedure, and a series of design evaluation procedures. Architects will have to balance the architectural design and the LowEx cooling systems to satisfy the requirements of multiple performances.

The encoded design schema is then used as input in the optimisation stage where an optimisation algorithm is employed. The design generation procedure is used for the generation of a population of design variants. The evaluation procedures are used to measure the performance of each design variant in the population. The optimisation process then uses a feedback procedure consisting of various mechanisms that select the best design variants from the population and automatically generate new design variants. The analysis stage is where architects explore the design variants evolved in the optimisation stage. K-means and Archetypal Analysis is employed to facilitate the data analysis process. Parallel Coordinate Plots (PCP) and 3D models are used to visualise the results. The analysis process aids architects in understanding the relationship between the architectural design and LowEx cooling systems.
A prototype tool is developed to support the execution of the design method. The prototype tool is implemented in a scientific workflow management tool. Domain-specific applications are integrated in the platform, and its graph-based interface provided flexibility for one to customise one’s workflow for different projects. Three workflow templates are composed for the design method. The optimisation workflow template is composed for the encoding and optimisation stage, and the analysis workflow template for the analysis stage. The visualisation workflow template is also created to visualise the results from the analysis stage. Minimal alterations are required to customise the templates for a new design project. Finally, the design method and prototype tool are demonstrated on a case study.

The United World College South East Asia (UWCSEA) building is re-designed with the LowEx approach. The case study explores the courtyard typology. The size, dimensions, and position of the courtyard are varied for design exploration. The design concept is then encoded with the prototype tool. The optimisation algorithm generated 5000 design variants. The proposed data analysis procedure is then used to cluster the generated design variants into respective design clusters. Exemplar designs are extracted to offer a visual representation of the typical designs from each design cluster. The results show potential of the design method, where the generated design variants are separated and visualised into manageable design clusters. Architects are able to explore a large number of design variants, and be aware of the potential and limitations of the design concept. This would in turn enable the architects to balance the architectural design and LowEx cooling system to satisfy the multiple performances.
Kurzfassung


Das Konzept der Exergie beschreibt die Qualität eines Energieflusses. Der Niedrig-Exergie Ansatz (LowEx) zielt auf eine Auslegung des Kühlsystems ab, die die geringstmögliche Energiequalität für die Kühlung des Raums verwendet. Dies kann z.B. durch die Nutzung höherer Temperaturen für die sensible Raumkühlung erreicht werden. Durch die Trennung von sensibler und latenter Kühlung (SSLC) kann die Gesamtkühllast auf zwei verschiedene Kältemaschinen verteilt werden. Eine der Maschinen erzeugt die höhere Temperatur für die sensible Kühlung und die andere die tiefen Temperaturen für die latente Kühlung. Hierdurch kann der Konsum der hochwertigen Energieform Elektrizität reduziert werden.


Für die Erprobung der Methode wurde ein prototypisches Werkzeug entwickelt und in einer Umgebung für wissenschaftliche Workflows realisiert. Hierfür wurden domainspezifische

1 Introduction

This chapter gives an overview of the thesis. Low-Exergy (LowEx) space conditioning is introduced, and the key requirements for it to be applied to a building design in a hot and humid climate are identified. The problem statement and the research aim are stated. The research aim is broken down into smaller research questions, which are addressed in later chapters. Finally, the outline of the thesis is presented.

1.1 Motivation and background

Buildings account for about 40% of global energy consumption and one-third of Green House Gas (GHG) emissions (Lucon et al., 2014). As buildings have a long lifespan, energy efficiency strategies currently adopted will have a medium-term effect on their emissions. Effective building design therefore has the potential to significantly reduce global energy consumption and GHG emissions (UNEP SBCI, 2009).

In Singapore, 40–50% of the energy used by a building is used for space conditioning (NEA, 2010). An efficient space conditioning system has the potential to significantly reduce the energy consumption of a building. Singapore has a hot and humid climate, characterised by a uniform air temperature of around 28°C and a relative humidity of 84%. Figure 1-1 below is a psychrometric chart depicting 8760 hours of Singapore’s weather data. According to the Singapore’s code of practice, the air temperature of the interior space needs to be maintained at about 23–25°C, and the relative humidity should not exceed 65% for occupant comfort (SS553, 2009). None of the hours on the chart fall within the comfort range, and thus interiors need to be cooled and dehumidified for all 8760 hours.
1.1.1 LowEx space conditioning

The two main space cooling processes in a hot and humid climate are sensible and latent cooling. Sensible cooling lowers the interior air temperature, and latent cooling dehumidifies the interior air to maintain comfort for the occupants. Although it’s typically necessary to only sensibly cool interior spaces to temperatures of 23-25 °C, conventional air-conditioning systems are supplied with much colder fluid for the cooling process – usually water in the range of 6-8 °C. Such cold water is required in order to facilitate the latent cooling process, which involves the condensation of water directly out of the supplied air stream. As sensible and latent cooling are not separable in conventional air-based cooling systems, the result of using about 8°C fluid to both dehumidify and cool down the interior is an energy-intensive process, for which latent cooling is the driver.

To substantiate this claim, the concept of exergy can provide a more precise assessment of energy flow in an air-conditioning system. Exergy is widely used in the analysis of thermal systems in the field of mechanical engineering (Bejan, 2006; Dincer and Rosen, 2007). Energy provides an assessment of the potential of an engineered system to perform a physical process, while exergy offers a more precise view on the quality of the energy source, and assesses if it is being utilised appropriately in the system. A thermodynamic system is most efficient when the temperature
differences between the states at which energy is processed and later used to perform the physical process is as low as possible. Systems that aim to reduce this temperature difference is referred as Low Exergy (LowEx) systems. In recent years, these LowEx principles has been adapted and used in the design of building systems (Ala-Juusela, 2004; Torio and Schmidt, 2011).

The LowEx approach aims to design space conditioning systems that use low quality energy to maintain thermal comfort for the occupants, such as the use of higher temperatures that are closer to the room temperature to provide sensible cooling for the occupants. This can be achieved by the Separation of Sensible and Latent Cooling (SSLC), it splits the cooling load by using LowEx cooling systems that employ high temperature cooling components, such as radiant cooling panels or chilled beams (Mumma, 2002; Rumsey and Weale, 2006) to provide sensible cooling with temperatures at 14–18°C. Two chillers are used in this configuration, one for the sensible and one for latent cooling. The consumption of high quality energy such as electricity is reduced when the sensible chiller produces higher supply temperatures, and the latent chiller only works to dehumidify the required volume of air for ventilation and maintaining the humidity in the space.

However, the cooling capacity of high temperature cooling components is limited due to the higher temperatures. In order to use these components, the sensible cooling load of the building needs to be within its cooling capacity. The sensible cooling load of a building is dependent on the solar heat gain from the envelope and internal loads such as occupancy, equipment, and lighting loads. Architects and engineers therefore need to address the entire building as a system, and need to be aware of the interrelationships and constraints when designing the building, otherwise the performance of the cooling systems will be compromised, or it might not be possible to employ these systems at all. As a result, it is important to integrate the LowEx cooling systems in the early design stages, when there is still flexibility to alter the design.

1.1.2 Building Performance Simulation (BPS)

BPS is a multi-disciplinary field which draws on disciplines such as physics, mathematics, material science, human behavioural studies, and environmental and computational science (Spitler, 2006). It is usually based on numerical methods which use mathematical models to simulate the behaviour of a system. In order to run a BPS, a model is required to represent the key characteristics of the
building to be simulated. The model is used by the BPS to simulate the system’s operation over a period of time. In the building design field, the most common BPSs are Building Energy Simulation (BES), lighting simulation, airflow simulation, and structural simulation.

The field of BPS can facilitate the integration of LowEx cooling systems in the early design stages by informing architects of how their design changes will affect the cooling systems. Currently, the use of BPS in the design process is still mainly limited to the final design stages for validation purposes (Flager and Haymaker, 2007). Attia et al. identified six main barriers to using BPS in the early design stages: differences in geometry representation between design applications and BPS, filling in of high resolution input parameters that are not available in the early design stages, no informative support during decision making, inability to compare a range of design alternatives, difficulty in interpreting results, and, finally, a lack of feedback to the architects for iterative design exploration (Attia et al., 2012). Various research efforts have been made to overcome these barriers and better integrate BPS into the early design stages. These include integrating BPS into 3D modelling applications, parametric 3D modelling for rapid generation of designs, and integration of optimisation algorithms to automate the search for optimal design variants.

1.2 Problem statement

In the early design stages, architects play a key role in determining the form, envelope, construction, and interior layout of the building (ASHRAE, 2006). The building envelope will eventually house the technical building systems. If the building is designed without consideration of the LowEx cooling systems from the beginning, it might not be able to accommodate them in the later design stages. For example, in a room with high equipment and occupancy load, if the envelope is not well designed to reduce the solar heat gain, it might not be possible to use higher temperature to sensibly cool down the space. Even if the room is able to accommodate high temperature cooling components, evidence has shown that buildings whose innovative heating/cooling systems have only been defined in a later design stage, cannot reap the most benefit from them (Hirn, 2009; Mahler and Himmler, 2008).
The early design stages inevitably become more demanding than usual when architects have to consider incorporating LowEx cooling systems into their design. It is a difficult task to foresee how design changes will affect the cooling systems and their performances. This impedes the design process as architects are unable to receive feedback about their design changes. The missing link lies in the capability to identify interactions and simulate consequences of design changes on cooling systems.

1.3 Aim and research questions

This research envisions an integrated design process with both architects and Heating, Ventilation, and Air-Conditioning (HVAC) designers in close collaboration, jointly defining the requirements and properties of a building’s HVAC systems. **The aim is to facilitate the integration of LowEx cooling systems into the early stages of architectural design.** The research aim is met through a series of research questions:

1. What are the LowEx cooling system configurations and components suitable for hot and humid climates?
2. What are the interactions of LowEx cooling system components with architectural design?
3. How and at which granularity can LowEx cooling systems be modelled, evaluated, and thus integrated into early architectural design stages?

1.4 Method

Research questions (1) and (2) are addressed using case studies of existing buildings. The systems identified are examined in terms of their interactions with the architectural design. Parameters relevant for early architectural design stages are then extracted to be modelled and evaluated. These LowEx cooling systems serve as examples for architects and HVAC designers in the selection of possible LowEx cooling systems for building design. Research question (3) is addressed by modelling the architectural design with the relevant parameters of LowEx cooling systems. The proposed design evaluation method needs to support design decisions in the data-poor situation of the early design stages. It serves as an example for architects and HVAC designers in setting up their design evaluations for their project.
The implications of integrating LowEx cooling systems into architectural design are studied, sorted and represented as a design problem. Lastly, a design method is proposed, which is a consolidation of the solutions to the three research questions. The design method is implemented on a prototype platform and applied to a case study.

1.5 Contribution

This research is targeted at, but not limited to, an architecture audience. It serves to explore and exemplify the integration of LowEx cooling systems in the early design stages from the perspective of an architect. The main contribution of the research lies in the development of a design method for the integration of cooling systems in early architectural design stages. The method assumes a collaborative, integrated design process, as described above. It aims to enable architects to have a better understanding of LowEx cooling systems and their interrelationships with architectural design, and thus have more in-depth discussions with HVAC designers about their application.

1.6 Organisation of thesis

The thesis is organised into seven chapters. Chapter 1 gives an overview of the research. Issues and research questions relevant to the research are identified.

Chapter 2 addresses research questions (1) and (2). Concepts of LowEx cooling in a hot and humid climate are elaborated on, and SSLC as a strategy to increase the efficiency of the chiller for air-conditioned buildings is introduced. Components of the LowEx cooling system which include radiant cooling panels, and Decentralised Ventilation Unit (DVU) are also introduced. This chapter provides the background of problems encountered when integrating LowEx cooling systems in the early architectural design stages. It serves as an example for the selection of LowEx cooling systems for architects and HVAC designers. Chapter 3 addresses research question (3). Utilising information that is available in the early design stages, a description of the modelling approach of cooling systems in the early design stages is provided.

Chapter 4 addresses the research aim with the proposal of a design method. In the integration of LowEx cooling systems into architectural design, a design change affects not only the cooling performances, but multiple performances. This is known as a Multi-Objective Optimisation
Problem (MOOP). A multi-objective optimisation approach is proposed, as a solution to facilitate architects in balancing the architectural design and LowEx cooling systems to satisfy the multiple performances. A design method is proposed by consolidating the solutions from Chapter 2, 3 and 4.

Chapter 5 describes the development of a prototype tool to support the design method. A scientific workflow management tool is used for the development of the tool. It enables the integration of different applications into one common platform, which solves the issue of interoperability between applications. Workflow templates are created for each stage of the design method to facilitate its execution. The graph-based user interface provides the flexibility and modularity for the architects to customise the templates according to each design project.

Chapter 6 describes the application of the method and tool in an industry-university building project case study, a new building block in the United World College South East Asia (UWCSEA) Dover. Finally, Chapter 7 provides a summary of the research, with a discussion of its contributions and possible future research.
2 Low-Exergy (LowEx) Cooling Systems for Separation of Sensible and Latent Cooling

In this chapter, a brief introduction of space conditioning is provided, followed by a description of LowEx space conditioning. LowEx space conditioning is illustrated with a heating case that was successfully implemented in Zurich, Switzerland. The approach and LowEx systems from this implementation are then theoretically adapted to the hot and humid climate of Singapore. These LowEx cooling systems are reviewed to offer insights into the implications of their usage in building design. This chapter serves as an example for architects and HVAC designers in their selection of LowEx cooling systems.

2.1 Space conditioning

The main aim of space conditioning is to provide thermal comfort for the occupants of a building. This includes controlling the air temperature, humidity, and supply of fresh air for ventilation, the removal of airborne particles, and the control of air movement in the space. Space conditioning consists of seven main processes (McDowall, 2007a):

- Heating – the process of supplying heat to the interior to maintain or increase the temperature.
- Cooling – the process of removing heat from the interior to maintain or lower the temperature.
- Humidifying – the process of adding moisture to the air to maintain or raise the humidity of the air.
- Dehumidifying – the process of removing moisture from the air to maintain or lower the humidity.
- Cleaning – the process of removing particulates and biological contaminants from the air to maintain or improve the air quality.
- Ventilating – the process of exchanging air between the outdoors and the conditioned space to remove gaseous particulates and maintain or improve the air quality and freshness.
- Air movement – the process of circulating the air in the conditioned space to achieve proper ventilation and thermal energy transfer.
Different space conditioning processes are required for different climates. For example, in the moderate climate of Zurich, heating is not required in summer and cooling is not required during winter. Although it can get too dry in winter when the relative humidity drops below 50%, space conditioning does not perform much humidification as the lack of moisture does not cause discomfort as much as high relative humidity. On the other hand, for the hot and humid climate of Singapore, where there are no seasonal variations, neither heating nor humidification is necessary all year round.

2.2 Application of exergy in space conditioning

The use of exergy in this research is only limited to its application to the design of space conditioning systems. Exergy enables the assessment of the quality of energy flow in a system. In recent years, it has been adopted for the analysis and design of space conditioning systems in buildings (Shukuya, 2012; Torio and Schmidt, 2011). The LowEx approach aims to design space conditioning systems that utilise low quality energy to maintain thermal comfort for the occupants. This is achieved by matching the supply and demand temperatures in the system.

In the case of hot and humid climates, instead of using only one temperature for both sensible and latent cooling, it may be exergetically more efficient to use separate chilled water networks operating at different supply temperatures in order to match the independent thermal requirements of sensible and latent cooling processes. Doing so can lead to the reduction of exergy (high-quality energy) consumption such as electricity, lesser electricity is consumed to produce the higher chilled water temperatures required by sensible-only cooling system (Juusela and Rautakivi, 2003).

2.2.1 Vapour-compression refrigeration cycle

The vapour compression refrigeration cycle, which is the most common method used for space conditioning in Singapore (Yap et al., 2011), is still compatible with a LowEx space cooling system. For a better understanding of the LowEx system setup, this section provides a brief description of the cycle.
The two main principles behind a refrigeration cycle are that liquid absorbs heat when it changes state from liquid to gas, and gas releases heat when it changes state from gas to liquid. The refrigerant is the substance circulating in the refrigeration thermodynamic cycle. The desired thermodynamic properties of a refrigerant are a low evaporation temperature for heat absorption and a high condensation temperature for heat rejection. These temperatures need to be viewed in relation to the reference environment. Figure 2-1 below shows a vapour-compression refrigeration cycle with its four main components: evaporator, compressor, condenser, and expansion device. A vapour compression refrigeration machine usually comprises of these four components. The refrigeration cycle works as follows:

1. The low-pressure refrigerant moves to the evaporator, absorbs heat, and changes from liquid to gas. The heat is usually transported from the interior space to the evaporator through water or air. Warm water/air goes through the evaporator, the refrigerant absorbs the heat, and chilled water/air exits from the evaporator.

2. The low-pressure refrigerant in its gaseous state moves to the compressor, is compressed, and moves out of the compressor as a high-pressure hot gas.

3. The high-pressure hot gas flows to the condenser, releases heat, and changes from gas to liquid. The heat is usually transported from the condenser to the ambient environment through water or air. Cooling water/air goes through the condenser, the refrigerant releases heat to the cooling water/air, and the warmer cooling water/air exits from the condenser.

4. The high-pressure refrigerant in its liquid state then moves to the expansion device, the device restricts the flow of the refrigerant, and lowers its pressure as it leaves the expansion device. The cycle is then repeated from step 1.

5. The refrigeration cycle can be reversed to provide heating to a space.
In the refrigeration cycle, electricity is needed at the compressor to pressurise the refrigerant. The amount of work needed in the form of electricity is dependent on the desired temperature difference between the evaporator and condenser. The greater the difference in temperature, the greater the amount of work needed at the compressor.

2.2.2 LowEx approach in moderate climates

Two notable building projects in Zurich, Switzerland serve as successful examples of the LowEx approach applied in the moderate climate. One is a partial demonstration of the radiant heating panels and the Decentralised Ventilation Unit (DVU) in the refurbishment of an office building HPZ in ETH Zurich (Mast et al., 2010), and the other is a full demonstration in a residential building B35 (Meggers et al., 2012).

The research into LowEx space conditioning in a moderate climate focused on low-temperature heating systems using a low-temperature lift heat pump (Meggers et al., 2010). The low-temperature lift refers to the design of the heat pump made to handle the low-temperature
difference of 10–20K between the evaporator and condenser, and thus achieve high performance. Conventional heat pumps are built for a temperature lift of 30–60K (Wyssen et al., 2011).

Heat is extracted from a low-temperature heat source using a heat pump, and the temperature is raised using electricity and pumped to the space for heating (Meggers and Leibundgut, 2012). The electricity input is dependent on the Coefficient of Performance (COP) of the heat pump: the higher the COP, the less electricity input is required. The COP is dependent on the temperature difference between the condenser, $T_c$, and evaporator, $T_e$, as shown in equation (2-1) below. The lower the temperature difference, the higher the COP.

$$COP_{heating} = \frac{Q_h}{W} = g \left( \frac{T_c}{T_c - T_e} \right) \quad (2-1)$$

$Q_h =$ heat supplied to the building (W)

$W =$ work consumed by the heat pump (W)

$g =$ exergetic efficiency of the heat pump

$T_c =$ temperature of the evaporator (K)

$T_e =$ temperature of the condenser (K)

The COP is express as a fraction of the ideal Carnot efficiency, where the exergetic efficiency represents the efficiency of the vapour compression machine. A vapour compression machine, including both heat pump and chiller, will have a typical exergetic efficiency of 0.4~0.5 (Bruelisauer et al., 2014). The research in the heating case focused on increasing $T_e$ and lowering $T_c$. The evaporator temperature, $T_e$, was increased using geothermal heat as an environmental heat source. The condensation temperature, $T_c$, was lowered by the use of low-temperature heating systems such as radiant heating panels. DVUs were used for ventilation and to maintain the indoor air quality. As the units were placed in close proximity to the space, the conditioned air did not gain temperature, and less electricity was used to pump air through long-distance ducts. After their successful application, the LowEx systems were transported and adapted to the hot and humid climate of Singapore.
2.2.3 LowEx approach in hot and humid climates

In hot climates, heat is removed from the building with a chiller and rejected into the environment. Similarly, a low-temperature lift chiller is required for a LowEx cooling system (Wyssen et al., 2010).

The equation for calculating the COP of the chiller is similar to equation (2-1) above:

\[ \text{COP}_{\text{cooling}} = \frac{Q_c}{W} = g\left(\frac{T_e}{(T_c-T_e)}\right) \quad (2-2) \]

\( Q_c \) = heat removed from the building (W)

As there is no readily available environmental resource in the hot and humid environment of Singapore for lowering the condensation temperature, \( T_c \) (Bruelisauer et al., 2013b), the research for the cooling case focused on increasing \( T_e \). The outdoor (28°C) and indoor (23–25°C) air temperature difference is much smaller than that of the heating case in the moderate climate. A high COP can be achieved by using a temperature close to the indoor air temperature to cool down the space, thus increasing \( T_e \). This use of higher supply temperature can in turn be achieved by Separation of Sensible and Latent Cooling (SSLC) (Hwang et al., 2010; Ling et al., 2014, 2008), in which two chillers are used, one for sensible cooling and the other for dehumidification. The chiller for sensible cooling achieves a higher COP and thus reduces electricity consumption by a significant amount.

Figure 2-2 (a) below shows the sensible cooling process of lowering the dry-bulb temperature from around 28°C to 23–25°C; a supply temperature of about 14–18°C is required to maintain the indoor temperature. Figure 2-2 (b) shows the latent cooling process in which the air has to be first cooled to its dew-point temperature, then further cooled to remove moisture. About 8°C is required to mechanically dehumidify the air from about 20g/kg to 12g/kg. Latent cooling is an exergy intensive process compared with sensible cooling, due to the requirement for lower temperature.
2.2.4 Conventional cooling system in Singapore

In conventional cooling systems, sensible and latent cooling are often not separated and are thus carried out using a low temperature of 8°C. The most common cooling system used in Singapore’s large commercial buildings is a central all-air system (Yap et al., 2011). In a conventional all-air system (Figure 2-3), the return air is mixed with fresh air intake. A central Air Handling Unit (AHU) is a device for regulating and circulating air in the conditioned space. The chiller produces 8°C chilled
water and delivers it to the AHU to dehumidify and cool the air. The cooled and dehumidified air is then delivered to the conditioned space through a network of ducts. The condensation temperature is usually about 28°C with a wet cooling tower, and the temperature lift in the chiller is then about 20°C.

2.3 SSLC system in research

The LowEx system described in this research is based on the system developed by the Building Systems Group at ETH Zurich for moderate climates, and later adapted and used by the Future Cities Laboratory, LowEx Module for hot and humid climates. A LowEx system consisting of radiant cooling panels and DVU is used for SSLC. In this section, components are first introduced, followed by a description of the system setup. This is not the only possible system setup for achieving SSLC; there are other possibilities such as an all-air system (Ling et al., 2008), a Dedicated Outdoor Air System (DOAS) (Conroy and Mumma, 2001; Kosonen and Tan, 2005; Mumma, 2002), and a desiccant system (Niu et al., 2002).

2.3.1 Radiant cooling panels

Radiant cooling panels are metal ceiling panels with chilled water pipes installed at the back of the panels to control their surface temperature. They are hydronic systems, in which water is used as the medium for the removal of heat from the conditioned space. They are also categorised as
Radiant cooling systems. Radiant cooling systems are defined as systems that provide more than 50% of their cooling through radiant exchange using a cooled surface (ASHRAE, 2008). The two main cooling mechanisms of radiant cooling panels, radiant heat transfer and convection, are illustrated in equation (2-3) and (2-4) below, respectively (ASHRAE, 2008).

\[
q_r = 5 \times 10^{-8} [t_p^4 - \text{AUST}^4] \quad (2-3)
\]

\[
q_c = 1.78 |t_p - t_a|^{0.32} (t_p - t_a) \quad (2-4)
\]

- \(q_r\) = heat removal rate due to radiation (W/m²)
- \(q_c\) = heat removal rate due to convection (W/m²)
- \(t_p\) = mean temperature of panel surface (K)
- \(t_a\) = indoor dry-bulb air temperature (K)
- \(\text{AUST}\) = area-weighted temperature of all indoor surfaces (K)

One of the key disadvantages of radiant cooling panels is their limited cooling capacity. The chilled water cooling the surface needs to be 1.5K above the dew-point temperature of the space to avoid condensation (McDowall, 2007b). A space within the comfort range of 50% relative humidity and 25°C dry-bulb temperature has a dew temperature of about 15°C, assuming that \(t_a\) and \(\text{AUST}\) are at 25°C. Due to the inefficiency of the thermal conductance between the chilled water pipes and the panel surface, the supply temperature can be assumed to be 3°C lower than the effective panel temperature (Conroy and Mumma, 2001). 16.5°C would give a mean panel temperature of 19.5°C, with the cooling capacity of the radiant cooling panels at 45 W/m². The low cooling capacity requires a large surface area as illustrated in equation (2-5).

\[
A_c = \frac{Q_c}{q} \quad (2-5)
\]

- \(A_c\) = Cooling surface area (m²)
- \(Q_c\) = Cooling capacity of radiant cooling panels (W)
- \(q\) = Heat removal rate (W/m²)

The sensible cooling load of the space needs to be within the limited cooling capacity of the radiant cooling panels (Mumma, 2002; Niu et al., 2014; Rumsey and Weale, 2006). The sensible
cooling load is the rate of heat removal for maintaining the temperature of the space. It is dependent on the solar irradiation through transparent surfaces, heat conduction through external and interior walls, heat generated by occupants, and equipment. The cooling capacity is the ability of the radiant cooling panels to remove heat, and is mainly dependent on the ceiling surface available for cooling, chilled water supply temperature, and chilled water flow rate.

There are various advantages of using radiant cooling panels, the first being their compact design (Mumma, 2002): radiant cooling panels have a slim profile of about 150 mm. A greater floor-to-ceiling height can be achieved without the need for deep plenum space. In addition, the panels can be integrated with other appliances such as lighting, as shown in Figure 2-4 below. Finally, there is high flexibility in their installation, as they can be readily mounted onto the ceiling. It has also been shown that radiant cooling panels are able to achieve better thermal comfort than conventional all-air systems (Imanari et al., 1999; Kosonen et al., 2014).

![Figure 2-4 Multi-functional radiant cooling panel](image)

However, there are architectural implications for the use of radiant cooling panels. First, architects need to design a well-sited and orientated building form with a good quality envelope to keep the sensible load low, and the interior layout needs to have sufficient ceiling surface area for the radiant cooling panels. These parameters need to be balanced to ensure that the sensible cooling load of the space is within the cooling capacity of the radiant cooling panels. The other contributors to the sensible load, such as the internal gains, can be kept low with the use of more energy-efficient equipment and lighting.
Second, the conditioned space cannot be exposed directly to outdoor conditions because of the risk of condensation on the cooled surface of the radiant cooling panels. The most suitable application for radiant cooling panels is therefore in an office building, in which most spaces are conditioned, and which has a consistent usage profile during weekdays.

Third, in Singapore the volume of air required to satisfy the ventilation requirement in spaces of high occupancy density, such as auditoriums, already provides adequate sensible cooling to the space, and so the use of radiant cooling panels is unnecessary in such environments. In regions where reheating is allowed, the air is reheated to comfortable temperatures of around 16–18˚C before being delivered to the space. However, as reheating is not allowed in Singapore (SS553, 2009), the temperature of the dehumidified air released to the interior space is about 12–14˚C. As a result, in addition to ventilating the space, the sensible load is also satisfied by the chilled air provided.

### 2.3.2 Decentralised Ventilation Unit (DVU)

A Decentralised Ventilation System (DVS) uses multiple DVUs for dehumidification and ventilation. In a central AHU, dehumidified air travels long distances before reaching the interior space. More powerful fans are required to move the air, and temperature gains occur through the long distances. The DVS reduces the distance travelled by the dehumidified air to reach the interior space, by locating the DVS in close proximity to the space it serves. DVUs are installed at the perimeter of the space, and outdoor air is drawn through the façade. The air is then dehumidified and delivered to the interior space. As a result, less fan power is required and temperature gains are reduced (Baldini et al., 2014).

The DVU’s smaller physical size also means a lower cooling capacity, and thus there is difficulty achieving dehumidification (Mahler and Himmler, 2008). A DVU specially designed for dehumidification in the hot and humid climate of Singapore has been developed (Baldini et al., 2014). The dehumidified air is delivered to the interior through an integrated network of air ducts (Baldini and Meggers, 2008). The DVU has to supply a certain volume of dehumidified outdoor air to maintain the humidity and air quality of the space. The required volume of fresh air is dependent on the occupancy and space usage. As the DVU is much smaller in size and cooling capacity than a
central AHU, a few units are required to provide the required latent cooling to a space. A central AHU will usually have the capacity to serve multiple spaces. Figure 2-5 below shows a DVU.

There are architectural implications for the use of DVUs. DVUs have to interface with the façade for the supply and exhaust of air from the interior space. In a central system, air is supplied and exhausted through a centralised ducting network. In a decentralised system, the air is supplied and exhausted through the façade directly in contact with the space. The positioning of the inlets and outlets needs to be integrated functionally and aesthetically into the façade design. Figure 2-5b shows an example of how the DVU interfaces with the façade to draw outdoor air and dehumidify it.

As the DVUs are located in close proximity to the conditioned space, the interior layout needs to accommodate the placement of DVUs. The decentralisation and miniaturisation allows the DVUs to be integrated either into the floor construction (Figure 2-5a) or the façade. As the DVUs only deliver the required volume of air for ventilation and latent load removal, the duct size is compact, and so the ducts can be embedded into the floor slab (Figure 2-6) or a raised floor for underfloor ventilation. The advantage of embedding the duct network in the slab is that the ducts need not be insulated. In addition, the thermal mass of the floor slab would provide a limited amount of cooling to the space. Because of this, the floor-to-floor height can be reduced with no large mechanical head space, and the integration will save floor area as a central AHU room can be eliminated.

![Figure 2-5 (a) DVU integrated into the concrete slab (b) DVU integrated into the raised floor](image)

*Figure 2-5 (a) DVU integrated into the concrete slab (b) DVU integrated into the raised floor*
2.3.3 SSLC with radiant cooling panels and DVU

A LowEx cooling system consisting of radiant cooling panels and DVUs has been set up in the BubbleZERO laboratory in Singapore (Bruelisauer et al., 2013a). Figure 2-7 below illustrates the schematics of the LowEx cooling system. The system totally removes the need for a central AHU and large air ducts by using water as a medium for the removal of heat from the conditioned space. Water can carry 4000 times more heat than the same volume of air. Large air ducts have been replaced by compact water pipes. As a result, there has been a reduction in mechanical head space.

The decentralisation also enables flexibility in operations, making it possible to adjust the cooling of each DVU according to the occupancy. Dynamic control strategies are required for the operation of the DVUs. Li et al. have shown how the DVUs can be controlled by the use of wireless sensors (Li et al., 2014). Results from the BubbleZERO have also shown that the DVUs are able to achieve the required dehumidification in Singapore’s hot and humid climate (Iyengar et al., 2013; Meggers et al., 2013).

The use of a LowEx cooling system requires architects to think holistically, as each component is highly integrated with the architecture design. Architects need to be aware of the scale, capacity, and demand of each component in the early design stages to be able to accommodate them in the design.
Figure 2-7 SSLC implementation with radiant cooling panels and DVU (adapted and redrawn from Yap et al., 2011)
3 Modelling Radiant Cooling Panels and Decentralised Ventilation Unit (DVU)

In this chapter, the aim of modelling and evaluating the LowEx cooling system is identified, and the amount of information available in the early design stages is taken into consideration. The modelling approach is introduced and demonstrated. This chapter may therefore serve as a guide for architects and HVAC designers to model and evaluate the selected LowEx cooling systems.

3.1 Modelling approach

In the early design stages, architects explore the project brief and the design simultaneously (Harfield, 2007). This is a cyclical process during which the brief and design feed into each other to define the boundaries of the project (Lawson, 2004). The purpose of running a Building Energy Simulation (BES) at this stage is not to obtain an absolutely accurate result, but to discover the performance tendencies of the different designs.

Conducting a detailed Building Performance Simulation (BPS) regardless of scale and resolution might often result in over-engineering and time-consuming BPS runs (Augenbroe, 2011). The evaluation of the LowEx cooling systems for the early design stages should require only coarse inputs, while still factoring in the relationship between the architectural design and the cooling system. Architectural design includes the building’s form, envelope, construction, and interior layout (ASHRAE, 2006). A BES is usually used to predict the cooling load of a space and subsequently the performance of the cooling system. BES such as EnergyPlus (Crawley et al., 2001) and DOE2 (DOE2, 2014) require detailed inputs for their prediction. Most of these inputs do not exist in the early design stages. Filling these inputs is identified as one of the main obstacles to using BPS in the early design stages (Attia et al., 2012).
Instead of a detailed BES, simplified calculations are used to model the relationship between the architectural design and the cooling systems. The results of the simplified calculations are merely performance indicators, and not an absolute measure of a performance (Augenbroe and Park, 2005). In Singapore, the Building Construction Authority’s (BCA) Envelope Thermal Transfer Value (ETTV) (BCA, 2013) is an example of such a calculation. This approach gives an indication of how well a building is performing, and is ideal for comparative studies.

3.2 Modelling and evaluating radiant cooling panels and DVU

This section is an extension of the conference paper (Chen et al., 2013b). This modelling approach is illustrated with the use of radiant cooling panels and DVU for the implementation of Separation of Sensible and Latent Cooling (SSLC). The building form and envelope play a key role in shaping the building and eventually house all the other building systems (Turrin et al., 2011). This is the first step in defining the building, and serves as the basis for a discussion of possible design strategies applicable to the project. At the same time, solar heat gain through the building envelope contributes significantly to the sensible load. Thus any design changes to the building form or envelope will affect the feasibility of using radiant cooling panels.

The modelling and evaluation provide feedback about whether the sensible load matches the cooling capacity of the radiant cooling panels, and about the performance of the cooling system. This allows architects to design the optimal building form and envelope that will accommodate the use of radiant cooling panels. There are also secondary design aspects, such as the ceiling area required for the radiant cooling panels, and the number of DVUs required for handling the latent load. All this feedback facilitates the integration of the cooling system into the design of the interior layout.
3.3 Calculations

The building form, envelope, and ceiling area, and the LowEx cooling system, was modelled and evaluated. The calculation is separated into two parts, the demand and supply. The demand calculations used are appropriate for all office and residential buildings in Singapore, while the supply calculations are only applicable to radiant cooling panels and DVUs. The evaluation can be adapted for other cooling systems by changing the supply calculations.

3.3.1 Demand

The demand calculations provide the total sensible load, total latent load, and outdoor air supply required for the space. The sensible load calculation is shown in equation (3-1) below. The sensible load of a building consists of the solar heat load from envelope, occupancy, equipment, and lighting load. The main concern at this design stage would be the solar heat gain from the envelope. The solar heat gain is calculated from the Envelope Thermal Transfer Value (ETTV) in equation (3-2) below.

The ETTV is used to calculate the solar heat gain through the glazing and opaque wall of an office building (Chua and Chou, 2010a). The Residential Envelope Transmittance Value (RETV) is used to calculate the solar heat gain through the glazing and opaque wall of a residential building (Chua and Chou, 2010b). The Roof Thermal Transfer Value (RTTV) is used to calculate the solar heat gain through the roof (BCA, 2013). The three calculations are based on the same equation: equation (3-3) below. The $TD_{eq}$, $\Delta T$, and $SF$ are constants used for calculating the solar heat load from the building envelope according to the different scenarios in Singapore (BCA, 2013), refer to Appendix A – $TD_{eq}$, $\Delta T$, and $SF$ for ETTV formula according to different scenarios, for the values.

$$\dot{Q}_s = \dot{Q}_{env} + \dot{Q}_o + \dot{Q}_l + \dot{Q}_e \quad (3-1)$$

$\dot{Q}_s$ = Total sensible load (W)

$\dot{Q}_{env}$ = Solar heat load from envelope (W)

$\dot{Q}_o$ = Occupancy load (W)

$\dot{Q}_l$ = Lighting load (W)

$\dot{Q}_e$ = Equipment load (W)
\[
\dot{Q}_{\text{env}} = A_{\text{env}} \cdot ETTV \quad (3-2)
\]

\(A_{\text{env}}\) = Total area of building envelope (\(m^2\))

\(ETTV\) = Envelope thermal transfer value (\(W/m^2\))

\[
ETTV = TD_{eq}(1 - \omega) \cdot U_w + \Delta T \cdot \omega \cdot U_f + SF \cdot \omega \cdot cf \cdot sc \quad (3-3)
\]

\(TD_{eq}\) = Equivalent temperature difference (°C)

\(\Delta T\) = Temperature difference (°C)

\(SF\) = Solar factor (\(W/m^2\))

\(\omega\) = Window wall ratio

\(U_w\) = Thermal transmittance of the opaque wall (\(W/m^2K\))

\(U_f\) = Thermal transmittance of the fenestration (\(W/m^2K\))

\(cf\) = Correction factor for solar heat gain through the fenestration

\(sc\) = Shading coefficient of the fenestration

The calculation of the ETTV accounts for orientation, construction, and simple shadings of the envelope. In a hot and humid climate, a well-conceived shading design has significant impact on the reduction of solar heat gain through the envelope. As a result, the ETTV calculation has been improved to account for complex shading and self-shading geometry (Fong et al., 2009). In the proposed approach, the ETTV equation is furthered enhanced by using Radiance (Ward and Shakespeare, 1998) to calculate the shading coefficient of the fenestration. This ensures that the calculation is more sensitive to changes in the shading design. The glazing manufacturer provides the shading coefficient of the glazing, \(sc_1\), in equation (3-4) below. To obtain \(sc_2\), an annual solar irradiation simulation is conducted using Radiance to obtain \(I_s\). The Radiance simulation is conducted again without the shadings to obtain \(I_o\) in equation (3-5) below.

The occupancy load is calculated in equation (3-6) below. The \(O_a\) is dependent on the building type. The \(q_o\) is dependent on the activity of the occupants. In a typical office in Singapore, \(O_a\) is about 10\(m^2\)/person and \(q_o\) is about 75W. The lighting load is calculated in equation (3-7) below. The \(q_l\) is dependent on the lighting fixtures. In a typical office in Singapore, \(q_l\) is about
The equipment load is calculated in equation (3-8) below. The $q_e$ is dependent on the equipment. In a typical office in Singapore, $q_e$ is about 25W/m$^2$.

The outdoor air supply required for removing the latent load is specified by the building codes. In Singapore, the minimum outdoor air supply required for an office building type is 0.6 l/s/m$^2$ (SS553, 2009). The outdoor air supply is calculated in equation (3-9) below.

$$Q_o = A_f \cdot q_o$$ (3-6)

$$OA = A_f \cdot OA_{rate}$$ (3-9)

$OA = $ Outdoor air supply (l/s)

$OA_{rate} =$ Outdoor air supply requirement (l/s/m$^2$)
3.3.2 Supply

The assessment of the cooling system is based on the ASHRAE guide (ASHRAE, 2008) and the Carnot refrigeration cycle. The supply calculations will provide the total electricity consumption of the chiller. They will also provide the Coefficient of Performance (COP) of the chiller, cooling capacity, supply temperature, and ceiling cooling surface area required for the radiant cooling panels. Finally, they will also provide the number of DVUs required for latent load removal.

The electricity consumption is calculated as a ratio of the total cooling load to the COP of the chiller in equation (3-10) below. The COP of the chiller is calculated in equation (3-11) below. The Exergetic efficiency, $g$, is usually around 0.4 to 0.5. $T_s$ is dependent on the cooling system. For example, radiant cooling panels will require a supply temperature of about 14–18°C (289.15–293.15K), while an air-based system will require a supply temperature of about 8°C (281.15K). $T_r$ in Singapore can be assumed to be about 28°C (301.15K).

$$E_c = \frac{Q}{COP} \quad (3-10)$$

$E_c$ = Cooling electricity consumption of the chiller (Wh)

$Q$ = Total cooling load (Wh)

$COP$ = Coefficient of performance of chiller

$$COP = \frac{g \cdot T_s}{T_r - T_s} \quad (3-11)$$

$g$ = Exergetic efficiency which represents the fraction of the ideal performance of the chiller

$T_s$ = Supply temperature (K)

$T_r$ = Heat rejection temperature (K)

The DVUs will provide a certain amount of sensible cooling. The amount of sensible cooling is calculated in equation (3-12) below. $\rho$ is assumed to be 1.225 kg/m$^3$. $c$ is assumed to be 1.005 kJ/kg.K. $\Delta T$ is about 11 K, considering the interior is about 25°C and the supply air temperature is about 14°C. The cooling capacity of the radiant cooling panels is calculated in equation (3-13) below. $q_c$ is dependent on the supply temperature of the radiant cooling panels. The supply temperature is assumed to be 3°C lower than the effective panel temperature (Conroy and Mumma, 2001). Based on equations (2-3) and (2-4) in section 2.3.1, supply temperatures of 18°C, 17°C, 16°C, 15°C, and 14°C
would give heat removal rates of 32W/m$^2$, 41W/m$^2$, 50W/m$^2$, 59W/m$^2$, and 68W/m$^2$ respectively. The number of DVUs required for latent load and ventilation is calculated in equation (3-14) below.

The AF of the DVU is about 22 l/s or 80m$^3$/h.

\[
\dot{Q}_v = \left(\frac{QA}{1000}\right) \cdot \rho \cdot c \cdot \Delta T \quad (3-12)
\]

\(\dot{Q}_v\) = Sensible cooling provided by outdoor air supply (W)
\(\rho\) = Density of air (kg/m$^3$)
\(c\) = Specific heat capacity of air (kJ/kg.K)
\(\Delta T\) = Temperature difference between interior space and supply air temperature (K)

\[
\dot{Q}_c = A_c \cdot q \quad (3-13)
\]

\(\dot{Q}_c\) = Cooling capacity of radiant cooling panels (W)
\(A_c\) = Cooling surface area (m$^2$)
\(q\) = Heat removal rate (W/m$^2$)

\[
n_{dvu} = \frac{QA}{AF} \quad (3-14)
\]

\(n_{dvu}\) = number of DVUs
\(AF\) = Air flow rate of DVU (l/s)

### 3.3.3 Application of calculations in building design

The calculations only require a minimal amount of information to describe the thermal properties of the building envelope. The required information consists of only the \(u\)-value of the opaque wall and the shading coefficient and \(u\)-value of the glazing. The values used to calculate the lighting, occupancy, and equipment loads can be easily obtained from reference guides (ASHRAE, 2009, 2007; SS530, 2006; SS553, 2009). The geometrical information of the building form can be readily extracted from a 3D modelling application. The calculations are relatively simpler than those of a detailed BES. This indicates that it would be easier for an architect to decipher the workings of the calculations and obtain a better understanding of the relationship between the various building systems. Figure 3-1 below illustrates how the equations correspond to the building envelope, cooling system, and interior layout, and the relationships between these three elements:
Equation (3-2) factors in essential envelope and building form parameters and calculates the solar heat load. When architects are exploring building form and envelope design in the early design stages, it is the main contributor to the total sensible load.

The outdoor air supply will provide a certain amount of sensible load (equation (3-12)). The remaining sensible load that needs to be removed by the radiant cooling panels is calculated in Figure 3-1(b).

The cooling capacity of the panel is calculated in equation (3-13) with a range of feasible heat removal rates (32–68 W/m²). The method assumes a constant available cooling surface area, which is the ceiling area. If the sensible load exceeds the cooling capacity, the SSLC strategy is not feasible, in which the only option is to use an air-based cooling system which provides higher cooling capacity.

The heat removal rate \( q \) is derived from the supply temperature \( T_s \).
e) The COP is calculated using equation (3-11). The COP of the sensible cooling is calculated with the translated supply temperature ($T_s$), and the COP for the latent cooling is calculated with a temperature of 8°C. If an air-based cooling system is used, 8°C will be used for both the cooling processes.

f) The outdoor air supply is calculated in equation (3-9). It is the minimum volume of air required to remove the latent load from the space. The number of DVUs required to provide that volume of outdoor air is then calculated according to equation (3-14).

g) The cooling electricity consumption is calculated with equation (3-10)
4 Design Method for Balancing the Architectural Design and Low-Exergy (LowEx) Cooling System

Chapter 3 has shown how architectural design elements such as building form, envelope, construction, and interior layout are modelled to evaluate the performance of the LowEx cooling systems. Other than the performances of the cooling system, these architectural design elements also affect other building performances. This is identified as a Multi-Objective Optimisation Problem (MOOP). In this chapter, a literature review on the use of optimisation algorithm in architectural design is presented. From the review, a multi-objective optimisation process is proposed for balancing architectural design and LowEx cooling systems. Finally, the modelling approach from chapter 3 and the balancing process are consolidated into a design method, for the integration of LowEx cooling systems in the early architectural design stages.

4.1 Balancing architectural design and LowEx cooling system using optimisation algorithms

The building form and envelope design play a major role in determining the daylight performance (ASHRAE, 2006). The maximisation of daylight will increase the sensible cooling load as solar heat inevitably enters the interior with daylight, and conversely a reduction of the sensible cooling load will decrease the daylight performance. Architects therefore have to balance the two contradicting performances through a well-designed building form and envelope. Moreover, they need to achieve it within certain cost constraints. Other than the environmental and economic performances, the building form and envelope also have to satisfy the aesthetic intentions of the architects. The aesthetic is difficult to quantify as it is highly subjective and differs significantly between architects. In summary, the aim is to minimise the electricity consumption of the cooling systems, maximise the daylight performance, minimise the cost of the envelope, and at the same time maintain the aesthetic of the building envelope.
4.2 Review of optimisation algorithms in building design

The problem described above is a Multi-Objective Optimisation Problem (MOOP) (Coello Coello, 2001). With such a problem, when a single design change affects multiple performances, it is not possible to optimise the performances independent of each other; they have to be optimised simultaneously. The aim is to find a series of solutions that balance the multiple performance objectives. These solutions are then presented to the users for decision making. A Multi-Objective Optimisation (MOO) algorithm can be used to automatically search for a series of balanced solutions that satisfy the various performance objectives. In building design, the solutions are the design variants.

Optimisation algorithms have been applied throughout the different stages of the design process, from optimising building layout to finding the best control strategy for air-conditioning operations. Two review papers have extensively documented the use of optimisation algorithms in building design (Attia et al., 2013; Evins, 2013). For the literature review, eight research projects (Table 4-1) that use optimisation algorithms for design exploration in the early design stages were chosen and reviewed as follows:

Gerber and Lin (2013) developed a Building Information Modelling (BIM)-based system for evolutionary design called Beagle for energy optimisation. The system was validated with a demonstration showing how design variants can be improved in various different design scenarios (Lin and Gerber, 2014).

Granadeiro et al. (2013b) developed an optimisation system focused on two main aspects: first, the use of parametric modelling coupled with shape grammar in an optimisation process, and second the assessment of envelope-related energy performance in the early design stages. The demonstration showed that an optimisation algorithm was able to generate design variants of topological variations by coupling parametric modelling with shape grammars, and also showed the effectiveness of the envelope-related energy evaluations in the early design stages (Granadeiro et al., 2013a).
Yi and Malkawi (2009) developed a novel agent-based, form-making modelling technique for the generation of complex building forms; each point in the 3D geometry was treated as an agent and the 3D geometry changed according to the movement of each agent. This modelling technique was then used with Building Performance Simulation (BPS) and optimisation algorithms to generate building forms based on energy and airflow performances (Yi and Malkawi, 2012).

Janssen (2009) developed a platform for running optimisation algorithms on distributed computational infrastructures, called DEXEN. For running evolutionary optimisation algorithms, a separate plugin was developed called EDDEX (Janssen, 2015). The plugin used existing parametric modelling and BPS software to generate and evaluate design variants, thereby allowing architects with no programming knowledge to use evolutionary algorithms for design exploration (Janssen et al., 2011).

Von Buelow (2012) developed a system called ParaGen, a tool that uses parametric modelling, BPS, and optimisation algorithms to support the exploration of designs. The novelty of ParaGen is its interactivity with the architects during the optimisation process. Architects are able to intervene during the optimisation process to determine the building form based on their preferences (Turrin et al., 2011).

Flager et al. (2009) developed a system that integrated structural, energy, and Computer-Aided Design (CAD) applications with an optimisation algorithm for building design optimisation. The results of the optimisation were then visualised in various 2D and 3D plots for the designers.

Geyer (2008) developed an evolutionary design workflow for building optimisation. Instead of using existing BPS applications, he proposed the use of utility functions based on the preferences of the architects to measure the performances of the building. He also suggested decomposing the various building components according to the Industry Foundation Class (IFC) schema. The setup was demonstrated with the use of a Multi-Objective Genetic Algorithm (MOGA) (Geyer, 2009).
Caldas (2001) developed a generative system for sustainable building design called GENE_ARCH. It uses DOE2 with optimisation algorithms to optimise the energy performance of a building design. The system was used to optimise 3D building forms (Caldas, 2002) and re-designed the façade of Alvaro Siza’s School of Architecture at Oporto (Caldas, 2006). Recently, it was combined with shape grammar for the design of the patio house in Marrakesh Medina (Caldas and Santos, 2012).

Table 4.1 Summary of the reviewed research that used optimisation for the early design stages

<table>
<thead>
<tr>
<th>Research</th>
<th>Design generation</th>
<th>Design evaluation</th>
<th>Feedback</th>
<th>Data analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Gerber and Lin, 2013)</td>
<td>Parametric modelling</td>
<td>Energy &amp; cost</td>
<td>Multi-objective genetic algorithm</td>
<td>Pareto ranking</td>
</tr>
<tr>
<td>(Granadeiro et al., 2013b)</td>
<td>Parametric modelling and shape grammars</td>
<td>Energy</td>
<td>Genetic algorithm</td>
<td>Sorting and filtering</td>
</tr>
<tr>
<td>(Yi and Malkawi, 2009)</td>
<td>Agent-based form finding</td>
<td>Energy &amp; airflow</td>
<td>Genetic algorithm</td>
<td>Sorting and filtering</td>
</tr>
<tr>
<td>(Janssen, 2009)</td>
<td>Parametric modelling</td>
<td>Energy &amp; daylighting</td>
<td>Multi-objective evolutionary algorithm</td>
<td>Pareto ranking</td>
</tr>
<tr>
<td>(Flager et al., 2009)</td>
<td>Parametric modelling</td>
<td>Energy &amp; structural</td>
<td>Multi-objective genetic algorithm</td>
<td>Pareto ranking</td>
</tr>
<tr>
<td>(Caldas, 2001)</td>
<td>Parametric modelling &amp; shape grammar</td>
<td>Energy, lighting, cost</td>
<td>Multi-objective genetic algorithm</td>
<td>Pareto ranking</td>
</tr>
</tbody>
</table>
Four common components in the optimisation process are identified: design generation, design evaluation, feedback, and data analysis. Design generation consists of the methods used to generate design variants. Design evaluation consists of the BPS or evaluation methods used to evaluate the design variants. Feedback consists of the search mechanisms used to find the optimal design variants. Data analysis consists of the analysis methods used for post-processing of the results. Table 4-1 above shows a list of the eight research projects and their characteristics, classified according to the four components.

4.2.1 Design generation

The design schema (Janssen, 2004) captures the design intention of a building design. It is conceived by the architects to describe their design concept and the design evaluations made to assess the design concept, and is unique to each building project. The design schema is encoded computationally and used to generate design variants. One method of doing so is by parameterising the design schema. Most research projects use a form of parametric modelling for their design generation. The construction of the parametric model and the choice of BPSs for the design evaluations are based on the design schema.

Parametric modelling allows architects to explore multiple design variants with one design schema. The number and topology variation of design variants that can be generated is dependent on how a design schema is parameterised into a model. Most of the time, it is not possible to exhaustively explore all the possible design variants. This is the main reason parametric modelling is used with optimisation algorithms for a more directed and fruitful search through the design space. The design space represents all the possible design variants for a building project (Woodbury and Burrow, 2006).

Granadeiro et al. (2013b) and Caldas and Santos (2012) coupled parametric modelling with shape grammars to achieve topological variations in the generated design variants. Shape grammar is a design system in which rules of transformation are applied recursively to an initial form to generate new forms (Stiny and Gips, 1972). The coupling of the two modelling techniques generates design variants of topological variations. However, the complexity of parameterising the design schema is increased with the need to define the shape grammar system.
Lin and Gerber (2014) achieved topological variations in the design variants by constructing and running multiple parametric models instead. Each parametric model is a topological variation, and multiple parametric models need to be constructed and optimised. This manual approach requires more time and effort from the architects. The advantage is that the complexity of parameterising the design schema is reduced, compared with coupling the parametric model with shape grammar.

Yi and Malkawi (2009) is the only research that did not use parametric modelling for the design generation procedure. They introduced a novel approach called agent-based form finding. For this modelling technique, each point in the geometry is treated as an agent. The relationships between the agents are defined by the architects. The movement of one agent will affect its neighbouring agents. The building form is generated from the interactions and movement of the agents based on environmental performance. This technique is able to generate building forms of topological variations.

### 4.2.2 Design evaluation

The design variants are evaluated based on performance objectives. Energy, lighting, structure, and cost are the most common objectives. Different BPSs and calculations are used depending on the context and aim of the optimisation. Research that examines multiple performance criteria usually also covers conflicting performances, as the optimisation algorithm searches for optimal design variants that balance the various conflicting performances. For example, daylighting and energy are conflicting performances, and the improvement of one will lead to the degradation of the other.

In most research, a detailed Building Energy Simulation (BES) is used to calculate the energy performance of a building design. The detailed BES is mainly used to calculate the effect of a change in the building form or envelope on the thermal loads. Its effect on the performance of the cooling/heating systems is not a major concern. Finite Element Method (FEM) is used for the structural analysis of the building design. Computational Fluid Dynamic (CFD) is used for airflow analysis. Backward ray-tracing simulation is used for daylight analysis.
Most research integrates BPS into the early design stages by automating the exchange of information between the modelling and BPS applications. However, there is a lack of information about the inputs. As a result, an extra knowledge database is required to fill in the inputs of the unknown parameters. As these detailed BPSs are validated, most research assumes they can be applied throughout the design process.

Other research have adopted different approaches. Geyer (2009) used a series of matrices for the calculation of performances. The coefficients used in the calculation were derived from empirical data. Granadeiro et al. (2013a) developed a simplified energy calculation for residential buildings called the Envelope-Related Energy Demand (ERED). It is beneficial to opt for a simplified calculation over a full-fledged dynamic BPS in the early design stages, as the main aim is to find the correct performance tendencies of a design rather than obtain an absolutely accurate result.

4.2.3 Feedback

Most researchers used a variation of an Evolutionary Algorithm (EA) (Deb, 2001; Kicinger et al., 2005) for the feedback procedure. EAs have been widely used in different disciplines, such as engineering, scientific, and industrial applications (Coello Coello and Lamont, 2004). EA is a class of optimisation algorithm known for its flexibility and effectiveness. EAs are the most widely used optimisation algorithms in building design (Attia et al., 2013; Evins, 2013). Their effectiveness in building design has been shown by Wetter and Wright (2004), who compared a range of optimisation algorithms and found that the Genetic Algorithm (GA) was one of the most effective for BES-based optimisation problems. Caldas (2001) also showed its superior performance, compared with Simulated Annealing (SA), when solving a building design problem by aiming to minimise electricity consumption.
Instead of using optimisation algorithms, one can also use other algorithms to search through the design space. Basbagill et al. (2014) used sampling algorithms such as Orthogonal Array and Latin Hypercube to obtain a set of design variants that were uniformly distributed in the design space. Welle et al. (2011) used Design of Experiment (DoE) to determine the impact of building form and envelope construction on the thermal performance of the building design. However, depending on the design schema, it might be computationally expensive to generate a uniform population across the design space.

4.2.4 Data analysis

Most researchers (e.g. (Gerber and Lin, 2013), (Janssen, 2009), (Flager et al., 2009) and (Caldas, 2001)) used Pareto ranking to search for the optimal design variants after the optimisation. Research focusing on the use of single objective optimisation algorithm (e.g. (Granadeiro et al., 2013b), (Yi and Malkawi, 2009), and (Von Buelow, 2012)) used sorting and filtering for knowledge extraction after the optimisation.

Other data analysis techniques were used apart from Pareto ranking. Geyer (2009) manually clustered the 150 optimal designs on the Pareto front. He was able to identify four clusters, and in each cluster he identified a typical design. This was only possible due to the small population size on the Pareto front. He was then able to visually relate the forms and performances of the four types of design generated.

Welle et al. (2011) used sensitivity analysis to identify the most significant parameters affecting the performance of the design variants. Their data analysis method provided feedback of finer resolution to the architects. Through sensitivity analysis, architects were able to make adjustments to the significant parameters of their design schema. Basbagill et al. (2014) used a probability distribution function to analyse the generated design variants. With each design decision made, the function calculates the probability that the remaining design variants will be able to meet the performance target set for the project.
After the data analysis, the research results were visualised in various forms. The design variants were visualised in 3D geometry, and plots were used to examine the characteristics of these optimal designs. X–Y plots and bar graphs were usually used to visualise two-dimensional data, while surface plots were used to visualise three-dimensional data. Flager et al. (2009) used a Parallel Coordinate Plot (PCP) to visualise multi-dimensional data. PCP helps architects visualise multi-dimensional data on a 2D plane, and is useful for identifying trends.

4.3 Proposed multi-objective balancing process

A process for balancing the architectural design and LowEx cooling system to satisfy the multiple performances is proposed in this research. The LowEx cooling systems are modelled and included in the optimisation process. The balancing process uses an optimisation algorithm to generate a large number of design variants. A novel data analysis method is introduced to cluster the generated design variants, so that architects can compare the different clusters to see how different design affect the LowEx cooling systems and building performances.

4.3.1 Design generation

The proposed multi-objective balancing process uses parametric modelling for its design generation. Due to the additional complexity of designing shape grammars systems, parametric modelling is not coupled with shape grammars. The agent-based form-finding method from Yi and Malkawi is able to generate building forms of topological variations. However, the method is catered to form finding, and envelope-related parameters such as shadings and glazing are not considered in the method.

Parametric modelling is an abstract representation of a system. Some elements of the systems have attributes that are fixed, and some have attributes that can be varied. The fixed attributes are called explicit attributes, while the varying attributes are called parameters or variables. A parameter is constrained to a range of values. The parameter can be independent or dependent on another value from the parametric model (Barrios, 2005).

Parametric modelling in the context of building design refers to the representation of a design with parameters. It is able to represent the various building systems and their relationships. The attributes (explicit attributes and parameters), relationships, and dependencies are established
hierarchically in the parameterisation stage. The independent parameters act as inputs. An instance of the model is computed and constructed based on the values of the input parameters. Input parameters are manipulated to generate different instances of the parametric model (design variants). Once a design variant is generated, information can be calculated and extracted from the model.

Janssen and Stouffs (2015) separate parametric modelling into four main categories: object modelling, associative modelling, dataflow modelling and procedural modelling. The categories are defined by the ability to support iterations for constructing a parametric model. Object modelling applications such as Autodesk Revit (Autodesk, 2014) do not support iteration. Associative modelling applications such as Solid Works (“Solidworks,” 2015) supports single-operation iteration. Dataflow modelling applications such as Rhinoceros 3D Grasshopper (“Grasshopper,” 2015) support implicit multi-operation iteration. Lastly, procedural modelling such as Autodesk Dynamo (“Dynamo,” 2015) and Sidefx Houdini (“SideFx,” 2015) supports explicit multi-operation iteration. Procedural modelling provides the most flexibility and capability in constructing design variants of topological variations. Thus, parametric procedural modelling is adopted for the multi-objective balancing process.

The act of parameterising a design schema requires architects to explicitly externalise the relevant concept and design strategies. This is contrary to the conventional design process, in which ambiguity is the norm (Aish and Woodbury, 2005), and so facilitates the communication of design rationale in a multidisciplinary team for building design (Gane and Haymaker, 2012).

As it is a difficult task to quantify the aesthetic of a building, it will not be encoded as a design evaluation method. Instead, the aesthetic of the building form and envelope would be factor into the multi-objective balancing process through the design generation procedure. Architects control the aesthetic of their design through how they set up their parametric model. They then balance the aesthetic of the design with the environmental and economic aspects of the design by visualising the results from the optimisation algorithm.
4.3.2 Design evaluation

Detailed BPSs such as BES, CFD, and FEM require inputs of high resolution and are computationally costly. These BPSs are therefore not suitable for the early design stages due to the lack of information for the high-resolution inputs, and because the long computational time will not provide timely support for design decisions. As a result, the proposed multi-objective balancing process uses simplified calculations to model the design evaluations. The modelling approach for evaluating the energy consumption of the cooling systems has been described and illustrated in Chapter 3.

This approach can be extended to all design evaluations. For example, when evaluating daylight performance, instead of conducting a full year’s analysis of the daylight performance, one can run a single daylight simulation for the worst case scenario, to indicatively judge the performance of a building design. Radiance (Ward and Shakespeare, 1998) can be used to evaluate the daylight performance. The resolution of the information required for a daylighting simulation is lower than that required for a detailed BES. The main inputs required for the daylighting simulation are the geometry and the surface finishes of the building design, which are available in the early design stages. Compared to BES it is usually easier to automate the exchange of information between the 3D modelling applications and lighting simulation. There are various successful examples that have integrated lighting simulation with commonly used 3D modelling applications. Two such examples are DIVA for Rhino (DIVA, 2014) and Vasari for Revit Architecture.

4.3.3 Feedback

The multi-objective balancing process uses a Multi-Objective Evolutionary Algorithm (MOEA) for the feedback procedure. As compared to the sampling algorithms and DOE used by Basbagill et al. (2014) and Welle et al. (2011) respectively, it is more efficient to use an MOEA to search for optimal design variants that satisfy the multiple performance objectives. In a generic MOEA, designs are represented as genotypes and phenotypes. Procedures are defined to map the genotypes to form a design variant. A population of design variants are evolved using biologically inspired mechanisms such as crossover, mutation, and selection to produce better design variants. The quality of the design variants is measured by the performance objectives.
In building design, the genotypes are the parameters of a parametric model. A population is
generated. Each design variant is evaluated, and the performance objectives are then calculated
based on the evaluation results to determine the fitness of the variant. The selection mechanism
chooses the design variants for reproduction. The fitter design variants will have a better chance of
being chosen. Crossover and mutation are applied to pairs of design variants depending on the
crossover rate and mutation rate. The population then goes through a selection process to decide
which design variants will survive for the next generation. The cycle is then repeated in the next
generation with the surviving design variants. Architects can specify the population size, total
number of generations, crossover rate, and mutation rate.

In the proposed method, a specific variation of MOEA called the Non-dominated Sorting
Genetic Algorithm 2 (NSGA2) ( Deb et al., 2000), is used for the feedback. It is regarded as one of the
standard MOEA commonly used for optimisation. The pseudo code is presented as follows:

**NSGA 2:**
Randomly generate the initial population \( n_u \).
Evaluate the design variants according to the performance objectives.

**For Each** generation:

**Fast-non-dominated-sort ()** the design variants.

**Crowding-distance-assignment ()** and sort the Pareto fronts of the design variants.

**Crowded-comparison-operator ()** to select the design variants for reproduction:

- One Point Crossover according to crossover rate, \( k \).
- Single Point Mutation according to mutation rate, \( m_u \).

Create a child population of size \( n_u \).

Evaluate the offspring according to the performance objectives.

Kill off the parent population.

**Until** maximum number of generations \( g \).
Fast-non-dominated-sort (design variants):

Rank = 0

While (design variants are not Non-dominated ranked):

- Extract the Pareto Front of the design variants.
- Remove Pareto Front from design variants.
- Assign Non-dominated rank (Rank) to all design variants on Pareto Front.
  - Rank += 1

Until all design variants are Non-dominated ranked

Crowding-distance-assignment (design variants):

For each objective:

- Sort the design variants according to the objective
- Assign crowding distance of infinity to the first and last design variant

For each design variant:

- Crowding distance = crowding distance + ((objective of former design variant – objective of latter design variants)/(objective of last design variant - objective of first design variant))

Until all design variants are assigned a crowding distance

Crowded-comparison-operator (design variant1, design variant2):

If design variant1.non domination rank < design variant2.non domination rank:
  Choose design variant1

If design variant1.non domination rank > design variant2.non domination rank:
  Choose design variant2

If design variant1.crowding distance > design variant2.crowding distance:
  Choose design variant1

If design variant1.crowding distance < design variant2.crowding distance:
  Choose design variant2

4.3.4 Data analysis

This section is an extension of the conference paper (Chen et al., 2015). The design variants generated by the optimisation process are then compared. Through the comparison, architects are able to better understand how the design changes are affecting performances. This requires that the generated design variants be processed and visualised for the architects. Sensitivity analysis, as used by Welle et al. (2011), can help the architects determine the important parameters affecting the
performances. However, sensitivity analysis requires a uniformly spread population of design variants, which is not the case when an MOEA is used. The multi-objective-balancing process adopts Geyer’s (2009) approach of clustering the design variants. Geyer was able to manually cluster the design variants in his research due to the small Pareto front population.

As opposed to using only the Pareto front, this research proposes to cluster all the generated design variants, so that architects can have a better understanding of the design across all performances. This means that an automated clustering method is required for executing the data analysis. The data analysis procedure then uses PCP in the visualisation of multi-dimensional data. The large number of generated design variants makes it difficult for architects to browse through the design variants even after they are clustered. As a result, design variants that best represent each cluster is extracted. These design variants give an idea of the typical building design of a cluster. Two techniques are introduced to perform the task, K-means clustering analysis and Archetypal Analysis.

**Partitioning cluster analysis: K-means**

Cluster Analysis (Everitt and Hothorn, 2011; Han et al., 2012; Hartigan, 1975) partitions a set of data into clusters such that the individual units of data in each cluster are similar to one another, while different from those in the other clusters. This is a form of unsupervised learning that is able to automatically detect clusters within a data set without any user specification. Cluster Analysis is used to gain insights into the distribution of a set of data, observe characteristics unique to each cluster, and help identify clusters of interest for further analysis. It is a useful technique for exploring hidden patterns in a large data set. However, the quality of the clusters is still dependent on what data attributes are fed into the analysis.

Design variants can be clustered based on the performance scores, derived parameters, and input parameters. For example, a design variant (a data point) has performance score attributes such as cooling electricity consumption, envelope material cost, and daylighting. It will also have derived parameter attributes that are derived from information extracted from the 3d model. Derived parameters includes shape factor and Window Wall Ratio (WWR). Input parameter attributes are the inputs required to develop the 3d model, such as the shading depth or window
height of a building. If only the performance scores were used, the Cluster Analysis would be able to cluster design variants with similar performance scores together.

One of the central mechanisms for identifying clusters is the measurement of similarity and dissimilarity. Clusters are formed by grouping similar data points while ensuring that clusters are as dissimilar to each other as possible. The Euclidean distance is used in most basic cluster algorithms to measure the similarities between each data point, while other advanced cluster algorithms use measures such as density, network connectivity, or probability function (Everitt et al., 2011).

The partitioning method is the simplest and most fundamental type of Cluster Analysis that uses Euclidean distant as the measure of similarities. The partition method is sufficient for separating the design variants into respective performance clusters. It organises the data into exclusive clusters with the resultant number of clusters being specified in advanced by the user. One of the most commonly used partitioning methods is K-means. Figure 4-1 below shows how a partitioning method starts with a random initial clustering using random selected centroids (crosses in Figure 4-1a). Centroids are the centres of the cluster. The analysis iterates through the data set searching for the best clusters. New centroids are calculated at each iteration (Figure 4-1b), the quality of the cluster is measured by the within-cluster-variance. The smaller the variance, the more compact a cluster. The process stops when there is no change in the within-cluster-variance for a few iterations (Figure 4-1c).

![Figure 4-1 K-means process](redrawn from (Han et al., 2012, p. 453))

In a partitioning cluster algorithm, it is critical for users to specify the right number of clusters. The elbow method is based on the observation that by increasing the number of clusters, users reduce the within-cluster sum of the square of each cluster. This is because by splitting the data into finer clusters, each cluster becomes more coherent. However, splitting clusters that are
already coherent into finer clusters will result in only a marginal reduction of the within-cluster sum of the square. The turning point of the sum of the within-cluster curve is a heuristic to help specify the number of clusters for K-means (Figure 4-2) (Everitt and Hothorn, 2011).

**Figure 4-2 Elbow method: the turning point of the curve marked in red (redrawn from (Everitt and Hothorn, 2011, p. 181))**

*K-means and Archetypal Analysis of generated design variants*

Archetypal Analysis (Cutler and Breiman, 1994; Eugster and Leisch, 2009) identifies the archetypes of a data set. Figure 4-3 below shows the prototypes picked up by Archetypal Analysis and K-means, respectively. The Archetypal Analysis identifies data points on the convex hull (Figure 4-3, top) while the K-means centroids are amongst the data (Figure 4-3, bottom). The archetypes are the extremes, while the centroids are the averages of the design. By combining the two, a good sampling of the data set can be obtained.

The archetypes and centroids are new design variants generated with inference from the original design variants in the cluster, thus needs to be validated to ensure their performance and derived parameters are within the range of the cluster. For example, for a cluster with a daylight performance range of 400-800 lux and a shape factor of 0.2-0.4, the centroids and archetypes have to be within this range to be validated.
A hierarchical and sequential approach is needed to process the design variants; each step will prepare the design variants for the next step of clustering. The appropriate data are required for the clustering and Archetypal Analysis. Each design variant carries three main categories of data: performance scores, derived parameters, and input parameters. The proposed method consists of three main steps (Figure 4-4 and Figure 4-5):

Step 1, clusters the design based on their performances. In Step 2, each performance cluster is again clustered based on their derived parameters that describe the building form and envelope of the design variants. Step 2 also uses k-means clustering. The result shows design clusters of different building forms and envelopes that have similar performances. In step 3, the exemplars are extracted for each cluster, Archetypal Analysis and K-means are used on each derived parameter cluster. The design variants’ input parameters are used for the analysis. Archetypes and K-means centroids are extracted from each cluster. The archetypes and centroids are newly constructed design variants based on the original data, and have to be validated to prove they are within the performance range of their clusters. Archetypes and centroids that fail the validation are discarded from the exemplars.
4.4 Design method

This section is an extension of the conference paper (Chen et al., 2013a). The design method is a consolidation of the solutions from previous chapters, to facilitate the integration of LowEx cooling systems into architectural design. Three main stages are proposed: encoding, optimisation, and analysis. As each stage is conceived to be iterative. They are illustrated as three sub-loops connected with one main loop in Figure 4-6 below.

The encoding stage is interactive and consists of LowEx cooling systems selection (Figure 4-6(1a)), LowEx cooling systems evaluation (Figure 4-6(1b)), and design-schema formulation (Figure 4-6(1c)).
First, the architects and HVAC designers will decide on the LowEx cooling systems for the design project; Chapter 2 may serve as an example for their selection. They will then model and evaluate the LowEx cooling systems; Chapter 3 demonstrates the modelling approach suitable for the early design stages. Chapter 4 describes the integration of the LowEx cooling systems as a MOOP, where architects need to balance the architectural design and the LowEx cooling systems to satisfy multiple performance objectives. Architects will formulate the design schema for the optimisation stage and encode it as a parametric model and a series of design evaluations. The encoded parametric model and design evaluations will be iteratively tested to assess if the design problem is well represented.

The optimisation (Figure 4-6(2)) and analysis stages (Figure 4-6(3)) are based on the proposed multi-objective balancing process. The optimisation stage is fully automated and consists of design generation, design evaluation, and feedback. The architects will use the encoded design schema, which is a MOOP, as inputs for the optimisation stage. Design variants will be generated (Figure 4-6(2a)), evaluated (Figure 4-6(2b)), and optimised in this stage. This stage is also where the feedback (Figure 4-6(2c)) occurs. It is fully automated and the only interaction with the architects is through the input (encoded design schema) and output (generated design variants).
Finally, the analysis stage is interactive and consists of data analysis (Figure 4-6(3a)) and visualisation (Figure 4-6(3b)). This is an iterative process during which architects process the results from the optimisation stage. The data analysis proposed in 4.3.4 is used by architects to cluster, conduct Archetypal Analysis, and visualise the design variants. It will take a few cycles of analysis and visualisation before the architects can obtain a good understanding of the generated design variants.

The design method allows the architects to escape the limits of the search space imposed by a specific design schema. Information generated from the optimisation stage feedbacks into the encoding stage through the analysis stage. The design schema is iteratively adjusted and modified after each full cycle. This structure will also encourage architects to formulate multiple design schema to explore the design space.
5 Prototype Tool for Implementation of Design Method

A prototype tool based on the workflow approach is developed to support the design method by facilitating the encoding, optimisation, and analysis of the design schema. Applications are integrated into the prototype tool as workflow components. These workflow components are used for the composition of the workflow templates to support the design method in the early design stages. As the tool is a prototype, the focus is on getting the workflows to function. A certain level of experience with computational design such as parametric modelling and Building Performance Simulation (BPS) is required for executing the workflows. This chapter is an extension of the conference paper (Chen et al., 2012).

5.1 Concept for the prototype tool

For the selection and evaluation of LowEx cooling systems, architects have to work closely with the HVAC designers to decide on the cooling systems, and how to evaluate them accordingly. This research has provided an example for the selection of the LowEx cooling systems in Chapter 2, and demonstrated the modelling approach with a specific implementation of Separation of Sensible and Latent Cooling (SSLC) in Chapter 3. The aim of the prototype tool is mainly to facilitate the encoding of the design schema and balancing the architectural design and LowEx cooling systems, as described in Chapter 4.

5.1.1 Workflow

The workflow approach is adopted for the development of the prototype tool. The concept of workflow originated from the field of Business Process Management (BPM). A workflow automates a business process, in whole or part, during which tasks are passed from one participant to another for execution according to a set of rules (“Workflow Management Coalition,” 2014). An example would be the steps required to gain the approval of a loan application.

The workflow concept has been borrowed by the scientific community for application in the scientific discovery process (Taylor et al., 2007). Scientific workflows describe experiments. A
scientific workflow is defined as a flow of tasks – mostly computational tasks that are part of a scientific experiment. Scientific workflows are usually executed on distributed systems because of their huge demand for computational power (Qin and Fahringer, 2012a).

Business workflows are usually less dynamic and executed in a routine fashion, while scientific workflows are more exploratory. In a scientific workflow, processes are altered and changed constantly throughout the experimental period. From an end-user point of view, business workflows can be constructed by professional software developers, but scientific workflows need to be constructed by scientists themselves for exploration of experiments (Barga and Gannon, 2007).

The proposed design method for the integration of LowEx cooling systems resembles a scientific workflow. The design evaluation can be described as running “virtual experiments” on a building design. As each design project will have its own unique requirements and constraints, like scientific workflows, architects need to customise the workflow for each design project in order to effectively execute the design method.

5.1.2 Scientific workflow management tool

The scientific workflow management tool supports the composing of scientific workflows. Research efforts have been made to adapt the use of the workflow management tool to the building design process by a number of affiliated researchers.

Toth et al. proposed a design framework based on the concept of scientific workflows, to link design and analysis applications for the building design process. The framework adopted a process-oriented approach for the development of a flexible, visual, collaborative, and scalable open system (Toth et al., 2012).

Thomas and Schlüeter integrated the Design Performance Toolkit with the workflow management tool, Kepler, to facilitate the use of dynamic Building Energy Simulation (BES) in the building design process. Inputs for the BES were automatically extracted from the Building Information Model (BIM). The workflow management tool was used as a GUI for the customisation of the workflow between the BIM and the BES. The workflow enabled a smooth transition of data
between the BIM and the BES. This facilitated the adoption of a performance-driven approach for the exploration of the design space (Thomas and Schlueter, 2012).

Janssen et al. proposed a data-mapping approach to solve the issue of interoperability between 3D modelling and BES applications. The workflow management tool was used as a GUI for architects with minimal programming skills to set up mapping workflows. The mapping workflows could then be reused and shared between architects. This allowed fluid and interactive exploration of the design space (Janssen et al., 2014).

These research projects reveal three main advantages of using the workflow management tool in the building design process. First, the workflow management tool allows interoperability between applications from different domains. Second, the flexibility of its GUI allows architects to customise their workflows. Finally, the ability to share workflows enables collaboration between designers from different domains. However, these are still theoretical attempts, in practice the varied applications still require significant effort for them to interoperate, and it is still difficult for architects who are programming novices to operate these workflows. It is believed with better design GUI it is possible for architects to eventually use these workflows for design task.

5.1.3 Graph-based interface for composition of workflows

Most workflow management tools have a graph-based GUI. A graph-based GUI consists of nodes and edges. The nodes represent computational tasks while data are passed from node to node by edge connections. A workflow component is a node in a graph. A new workflow can be constructed by assembling the workflow components required and connecting them accordingly.

The graph-based GUI offers flexibility in the customising of the workflow for each design project. The graph-based GUI is a form of Visual Programming Language (VPL). VPL is useful for novices in computer coding, like architects, to learn the basics of computing (Celani and Vaz, 2012; Leitão et al., 2012). Visual Dataflow Modelling (VDM) applications are parametric modelling application with a VPL interface. The increase in the number of VDM applications in architectural design field is an indicator of its popularity as an approach to engage architects in computational design.
The graphical language expresses the logic behind the workflow and allows architects the freedom to compose their own workflow with the workflow components available. However, a graph-based GUI is unable to fully express all programming activities. Thus it needs to be accompanied by the ability to build new components to adapt to new contexts and scenarios (Gannon, 2007).

### 5.1.4 Interoperability and collaboration between designers from different domains

The workflow management tool allows users to develop new workflow components, either by the use of a scripting language or by defining a part or the whole of a workflow and collapsing it into a workflow component. The ability to build a new workflow component resolves the issue of interoperability between applications from different domains. It also offers a way to enable collaborative effort between users from different domains (Qin and Fahringer, 2012b).

The issue of interoperability arises when applications from different domains require different type of data. Architects are able to sort and filter the data in-between the applications through the use of a workflow management tool, with different applications linked together in a common environment. The process of linking the applications is only required once. Once the linkage is set up, it can be saved as a workflow component. These workflow components are deposited in a library and can be shared and reused for different projects. The library of workflow components increases as more applications are linked to each other through the workflow management tool. The library builds up over time and incrementally resolves the interoperability issue.

Collaborative effort is encouraged when workflow components can be shared among consultants. For example, in a modular workflow, which each workflow component represents a sub-workflow. The necessary data are passed between the nodes through the connecting edges for the execution of each design evaluation. Such an arrangement will enable designers from different domains to construct and define their own workflow, collapse it into a workflow component, and share it with the design team. The workflow management tool therefore becomes a common platform to analyse and visualise the performance of a building design.
In an ideal scenario, by predefining the necessary interfacing data between design generation and design evaluation, an HVAC designer can share his sub-workflow with an architect. Both of them can work on a common platform and see each other’s design process. For example, in the LowEx Cooling Systems Evaluation (Figure 4-6(1b) in chapter 4), HVAC designers and architects will figure out how to model and evaluate the LowEx cooling systems, and develop the design evaluation into a workflow component. The architects will then use this component with other design evaluations to balance the architectural design and LowEx cooling systems to satisfy the multiple performances. This facilitates the collaborative effort in a design process.

5.2 Prototypical tool using Kepler workflows

The prototype tool was developed in a workflow management tool. There are various open-source and commercial workflow management tools available. For the commercial applications there are the Phoenix Integration Model Centre ("Phoenix Integration," 2014) and Simulink ("MathWorks Simulink," 2014). For the open-source applications there are Kepler ("The Kepler Project," 2014), Vistrails ("Vistrails," 2014), Triana ("Triana - Open Source Problem Solving Software,” 2014), and Taverna ("Taverna,” 2014). All the workflow management tools offer the essential functions required to execute the design method. Open-source tools are preferred over the commercial tools, due to their low deployment cost. Kepler offers much more flexibility regarding the flow of data in the workflow than the other open-source tools (Deelmana et al., 2008). As a result, Kepler was chosen for the development of the prototype. The flexibility of Kepler has been demonstrated in (Pennington et al., 2007), where the tool was able to handle a different array of computational tasks.

In this chapter, the linking of the applications that are required for executing the design method is demonstrated. These applications include parametric modelling for design generation, design evaluations for assessing the cooling systems, optimisation algorithm for feedback, and analytical processes and visualisation for data analysis. In this research, the prototype tool is foreseen to serve as a template for architects and HVAC designers to build upon by creating new workflow components for their projects. In order for others to build upon the tool, the thesis documents how the tool has been developed and can be used by architects in their projects. The
tool will be made available online for any interested party to download and use (https://chenkianwee.wordpress.com/low-exergy-design-method-environment/).

5.2.1 The Extensible Markup Language (XML) schema for the transfer of data between applications

XML is a mark-up language used for the description of data. It defines a set of rules for encoding data in a format that is both human and machine-readable. Architects are able to define their own data schema to describe the data they want to document. XML is a well-established format. There are existing coding libraries and Kepler workflow components for editing and retrieving data from XML. Due to its well-structured format and flexibility, XML is used as a format for exchange of data within the Kepler workflow management environment.

In order to keep the XML file light and manageable, the XML schema is designed to carry only essential information necessary for the design method. Each design variant carries three main categories of data: performance scores, derived parameters and input parameters. The XML schema is illustrated in Figure 5-1 below. Each category of information is referred to as a node in an XML schema. The name of each node is referred to as a tag. The data node is a container for all related data of the design method. It only contains the population node. For the extendibility of the schema, the data level was created to ensure that any additional data can be easily appended in future. The population node is a container for all the design variants. Within each design variant, it carries essential information, as shown in Table 5-1 below. Essential information, except the geometries, is stored in the XML to facilitate the exchange of information between different applications. The geometries of the design variants are stored as 3D models. The file format of the 3D models is dependent on the 3D modelling application.
### Attributes of a design variant

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identity</td>
<td>A unique number for identifying each design variant.</td>
</tr>
<tr>
<td>Generation</td>
<td>Generation the design variant was reproduced in.</td>
</tr>
<tr>
<td>Status</td>
<td>Indication whether a design variant is alive or dead.</td>
</tr>
<tr>
<td>Inputparams</td>
<td>A container for the inputparam.</td>
</tr>
<tr>
<td>Inputparam</td>
<td>Input parameter for the generation of design variant.</td>
</tr>
<tr>
<td>Derivedparams</td>
<td>A container for the derivedparam.</td>
</tr>
<tr>
<td>Derivedparam</td>
<td>Derived parameter retrieved from the 3D model.</td>
</tr>
<tr>
<td>Scores</td>
<td>A container for the score.</td>
</tr>
<tr>
<td>Score</td>
<td>Performance score retrieved from a design evaluation.</td>
</tr>
</tbody>
</table>

#### 5.2.2 Integration of applications for execution of workflow templates

A series of workflow components are developed for the execution of the design method. In Kepler, there are two ways to build new workflow components with the Python programming language. The first way is through the use of Kepler’s native component “Python Actor”, which allows for the creation of custom workflow components through Python scripting. Inputs are processed by a Python script written within the component, and the results are the outputs. As Kepler is a Java-based application, an incompatibility issue arises with the use of the external Python libraries in the “Python Actor” component. This option is therefore only appropriate for workflow components that...
do not need the external Python libraries for their processes. Workflow components developed with this option include the components for the manipulation of the XML schema: “XmlRead”, “XmlExtract”, “XmlEditNode”, “XmlCreateNode”, and “XmlConsolidate”.

The second option resolves this limitation by building workflow components through Kepler’s native workflow component “external execution”. It allows for the execution of an external Python script. Inputs are passed to the external Python script and the results are returned to Kepler through the component. With this option, external Python libraries can be used in the scripts for the execution of specialised tasks.

Workflow components developed with this option include “HouDev”, which integrates Houdini3D (“SideFx,” 2015) through its Python Application Programming Interface (API) for design generation. From a previous study comparing three main VDM applications: Grasshopper, Generative Components, and Houdini3D (Janssen and Chen, 2011). Iterative mechanisms such as a “For” loop are the main mechanisms used to set up a parametric model. It is concluded that as Houdini3D explicitly expresses its iterative mechanism, it is cognitively less demanding for architects to set up a parametric model using Houdini3D.

All the design evaluation are integrated in Houdini3D to facilitate the design schema formulation in the encoding stage, to enable architects to get direct feedback within Houdini3D when setting up the parametric model and design evaluations. The method for assessing radiant cooling panels and DVU is integrated into Houdini3D. According to the evaluation method, Radiance simulation is used to obtain the solar irradiation for calculating the shading coefficient of the glazing. Radiance is also used for the evaluation of the daylight performance. A simple cost calculator is also integrated to calculate the cost of the building envelope. “HouEval” then integrates the Houdini3D evaluations into Kepler.
“Kmeans”, “Archetype Analysis”, and “PCPtxtfile” workflow components integrate scikit-learn (Pedregosa et al., 2011), PyMF (Thurau et al., 2012), and matplotlib (Hunter, 2007) for execution of K-means, Archetypal Analysis, and generate Parallel Coordinate Plots (PCPs), respectively. A detailed account of all workflow components is provided in Chapter 7.5 Appendix B – Workflow components developed for the workflow templates.

5.3 Workflow templates

Once the necessary applications are integrated into Kepler, workflow templates are composed from these workflow components to facilitate the execution of the design method. Workflow templates are composed for the encoding, optimisation, and analysis stage of the design method.

5.3.1 Encoding workflow template

The encoding workflow template is developed to facilitate the encoding of the design schema. Once the LowEx cooling systems have been selected and modelled, the design schema is developed, factoring in the considerations of these systems, then encoded into a parametric model and series of design evaluation procedures. The encoding of the design schema into a parametric model takes place in the parametric modelling application. Architects will iterate through a few models and find the most appropriate way to represent the design schema.

Figure 5-2a below shows an example of a design schema translated into a parametric model in Houdini3D. The resultant geometry is saved as a Houdini3D geometry file. The geometry has to have all the required data for the specific evaluation, it is then passed to the design evaluation node for conducting the evaluation. Figure 5-2b shows an example of a design evaluation in Houdini3D. The evaluation result is processed and interpreted into a single number that can best represent the performance of the design variant. That number is retrieved for the feedback procedure. The derived parameters are then decided and extracted to facilitate the data analysis procedure.
The optimisation workflow template in Kepler (Figure 5-3) is used to test the encoded design schema. The template is used both for the encoding and optimisation stage. A number of design variants is generated to ensure that the links between the parametric model and design evaluations are robust and free of error. The alterations required to adapt the workflow template to different design schemas are elaborated in the optimisation workflow template section. To test the schema, the “generation” parameter is set to 1 and the “init_population” parameter to the desired number of design variants. The workflow template facilitates the rapid generation of design variants to test the encoded schema.

The score and input parameter settings (Figure 5-3b), and the “DesignGenerationEvaluation” sub-workflow will have to be altered according to each test attempt (Figure 5-3c). Finally, the “live_file” and “dead_file” is specified for documenting the generated design variants. If the design generation and evaluations are robust and error-free, the workflow will automatically generate the specified number of design variants. Architects will iteratively test out different design schema during the encoding stage.
Figure 5-3 (a) Encoding the design schema (b) The optimisation workflow template configured for the encoding stage (c) The sub-workflow in the encoding workflow template that requires customisation for each test attempt.
5.3.2 Optimisation workflow template

Once architects are confident of their encoded design schema, the parametric model and evaluations are used as the design generation and design evaluation in the optimisation stage. Figure 5-4 below shows the optimisation workflow template in Kepler. The workflow is mainly made up of two components: the “NSGA2” (Figure 5-4b) and “DesignGenerationEvaluation” (Figure 5-4c) workflow components. The workflow template has to be altered according to the different design schema. Figure 5-4a shows all the necessary inputs for executing the workflow template. The optimisation workflow template requires 11 inputs, as summarised in Table 5-2 below. The 11 inputs are for the NSGA2 workflow component described in Appendix 7.5.10.

![Workflow Diagram](image)

Figure 5-4 The optimisation workflow template in Kepler (a) The inputs required for the execution of the workflow (b) The NSGA2 workflow component (c) The “DesignGenerationEvaluation” sub-workflow (d) Writing the results from the evaluations to the XML file

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>num_type_of_gene</td>
<td>Array of an array of an array of integers</td>
<td>Number and type of the input parameters.</td>
</tr>
<tr>
<td>range_of_gene</td>
<td>Range or a list of choices</td>
<td>Boundary of the input parameter.</td>
</tr>
<tr>
<td>num_of_set</td>
<td>Array of integer</td>
<td>Number of sets to generate.</td>
</tr>
<tr>
<td>score_names</td>
<td>List of string</td>
<td>List labelling the performance scores.</td>
</tr>
<tr>
<td>scores_min_max</td>
<td>List of string</td>
<td>A list indicating minimise or maximise the performance scores.</td>
</tr>
<tr>
<td>generation</td>
<td>Integer</td>
<td>Number of generations to be run for the NSGA2 algorithm.</td>
</tr>
<tr>
<td>init_population</td>
<td>Integer</td>
<td>Number of design variants to be generated for the initial population.</td>
</tr>
</tbody>
</table>
mutation_rate | Float | Rate of mutation during the feedback procedure.
crossover_rate | Float | Rate of crossover during the feedback procedure.
live_file | String | Path of the XML file that documents the design variants that are still alive.
dead_file | String | Path of the XML file that documents the design variants that are dead.

**Inputs defining the input parameters of the design-generation procedure**

The first three inputs are for defining the input parameters for the design variants. For the workflow to randomly generate the initial population, the input parameters need to be defined. For the first input, “num_type_of_gene” (Figure 5-5a), there are four types of input parameters: range of integers, a choice of integers, range of floats, and choice of floats. These four types of input parameters are labelled 1, 2, 3, and 4 respectively. The input parameters can be categorised into sets. For each input parameter, the number of input parameters of a certain type needs to be specified, then, according to the type, specify either a range or a list of the possible choices in the second input, the “range_of_gene” parameter (Figure 5-5b). If the type of the input parameter is a float or integer range, the range is defined by specifying its start, end, and step. If the type of the input parameter is a float or integer choice, the choice is defined by specifying a list of floats or integers. Multiple sets for each input parameter set is generated by specifying the input “num_of_set” (Figure 5-5c). Table 5-3 below is an example of the input parameters in the workflow.

![Figure 5-5 (a) Settings for “num_type_of_gene” (b) Settings for “range_of_gene” (c) Settings for “num_of_set”](image)
Table 5-3 Example of the settings and results of specifying the input parameters in the Kepler workflow

<table>
<thead>
<tr>
<th>Num_type_of_gene</th>
<th>{{{1,1}, {1,2}},{{4,3}, {1,4}}}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range_of_gene</td>
<td>{{{12,82,5}, {0,6,23,37,41}}, {{1,5,2,0.3}, {1.2,1.4,2.8,3.5,4.9}}以外}</td>
</tr>
<tr>
<td>Num_of_sets</td>
<td>{2,1}</td>
</tr>
<tr>
<td>Result</td>
<td>{12, 23, 82, 6, 1, 1.3, 1.6, 1.9, 1.4}</td>
</tr>
</tbody>
</table>

The “num_type_of_gene” is first specified. For example, {1, 1} means one input parameter of type integer range. {4, 3} means four input parameters of type float range. Then specify the “range_of_gene”. The array positions of the range need to correspond to the array positions in “num_type_gene”. For example, {1, 1} is in the first set first position of the “num_type_of_gene”, so its corresponding range needs to be in the first set first position of the “range_of_gene”. {1, 1} is an integer range, so a range is defined by specifying {12, 82, 5}: 12 is the start, 82 is the end, and it is in steps of 5. So when randomly generating a design variant, the script will generate a number within this boundary. Similarly, for {1, 4} in the second set second position, the corresponding range is a list of float choices {1.2, 1.4, 2.8, 3.5, 4.9}. When randomly generating a design variant, the script will choose one of the choices in this list. Finally, the number of sets is specified. In this example, {2,1} is specified, and thus the workflow will generate two sets for the first set and one set for the second set of input parameters.

The input parameters for a design variant are randomly generated for the example. The first four input parameters are randomly generated from the boundary of the first set of input parameters specified. As two sets are specified, four input parameters are generated. The fifth to eighth input parameters are generated based on the second set first position {4, 3} with range {1, 5.2, 0.3}. All the input parameters are in steps of 0.3, from 1 to 5.2. The last input parameters are chosen from the list of choices as specified. Thus by specifying the necessary information about the input parameters, the workflow will be able to randomly generate design variants.

*Inputs for the NSGA2 algorithm*

The fourth and fifth inputs for the workflow template are for defining the performance scores. Table 5-4 below is an example of how the performance scores is specified. The two lists need to correspond to each other; for example, “electricity” needs to be “min” (minimised), so both
“electricity” and “min” are in the first positions of the list. This continues with “cost” corresponding to “min” in the second position, and “daylighting” to “max” (maximise) in the third position. The seventh to ninth input are populated with default values, only advance users who are interested in adjusting the NSGA2 algorithm will alter these inputs. The default values for “init_population”, “mutation_rate” and “crossover_rate” are 100, 0.01 and 0.9 respectively.

Table 5-4 Example settings of the performance scores

<table>
<thead>
<tr>
<th>score_names</th>
<th>{“electricity”, “cost”, “daylighting”}</th>
</tr>
</thead>
<tbody>
<tr>
<td>scores_min_max</td>
<td>{“min”, “min”, “max”}</td>
</tr>
</tbody>
</table>

Workflow description

In the first cycle, “NSGA2” workflow component randomly generates an initial population of design variants. The design variants are evaluated in the “DesignGenerationEvaluation” and documented in the XML file as specified in “live_file”. In the next cycle, the “NSGA2” workflow component reads the “live_file” and performs feedback, better design variants are generated while the worst performing ones are “killed”. At the end of every turn, the dead design variants are appended to the “dead_file”. The “dead_file” provides a history of all the design variants generated.

The “DesignGenerationEvaluation” workflow component is a native “IterateOverArray” workflow component in Kepler (Figure 5-4c). It is equivalent to a “foreach” loop in textual coding. The workflow component takes in an array, loops through each element in the array, and performs the task defined in the workflow component. An array of input parameters are passed into the “DesignGenerationEvaluation”. The design generation and evaluation procedure are performed in the component. The output of the workflow components is the evaluated design variants.

Figure 5-6 below shows the sub-workflow in the “DesignGenerationEvaluation” workflow component. It shows the computational tasks that act on each element of the array; in this case, each element is an array of input parameters. The input parameters are passed as input into the “HouDev” component to generate the parametric model (Figure 5-6b). The component outputs the file path of the resultant geometry. The file path is passed to the “HouEval” components. The “HouEval” then outputs the result and derived parameters (Figure 5-6c). The number of “HouEval”
components differs according to the different design schemas. In order to adapt the optimisation workflow template to different design schemas, the number of “HouEval” components needs to be adjusted and the inputs fill in accordingly. Figure 5-6d below shows the sorting of the results and derived parameters into XML format.

In Figure 5-4d above, all the “live” design variants are consolidated, sorted into the XML format, and written to file. At the next cycle, the NSGA2 workflow component reads the “live” file and retrieves the design variants for the evolutionary process. The cycle continues until as specified by the architects in the “generation” parameter.

![Figure 5-6 Sub-workflow in the “DesignGenerationEvaluation” workflow component (a) The input parameters for developing a design variant (b) The “HouDev” workflow component (c) The “HouEval” workflow components (d) The results and the derived parameters from the design evaluation procedures sorted into the XML format and passed as output](image)

**5.3.3 Analysis workflow template**

A large number of design variants are generated in the optimisation stage. An analysis workflow template is developed based on the proposed data analysis method described in Chapter 4.3.4. Figure 5-7 below shows the analysis workflow template for post-processing the generated design variants. Figure 5-7a shows all the necessary inputs for executing the workflow template. The inputs under the heading “Kmeans Scores” are for the “Kmeans” component in Figure 5-7b. The inputs under the heading “Kmeans Derived Parameters” are for the “Kmeans” component in Figure 5-7c. The inputs under the heading “Exemplars” are for the “Archetypal” and “Kmeans” component in Figure 5-7d. A summary of the inputs are provided in Table 5-5 below.
Figure 5-7 Data analysis template workflow for post-processing optimisation result (a) The inputs required for the execution of the workflow (b) K-means of the performance scores of the design variants (c) K-means of the derived parameters of the design variants (d) Identification and validation of the exemplars of the design clusters

Table 5-5 Summary of the inputs for the analysis workflow template

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>overall_file</td>
<td>String</td>
<td>Path of the XML file to be analysed</td>
</tr>
<tr>
<td>result_dir1,</td>
<td>String</td>
<td>Path of the directory in which to store the result of the k-means clustering XML files.</td>
</tr>
<tr>
<td>result_dir2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>filename1,</td>
<td>String</td>
<td>Name of the resultant k-means XML file.</td>
</tr>
<tr>
<td>filename2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>max_cluster1,</td>
<td>Integer</td>
<td>Maximum number of clusters permissible for k-means clustering.</td>
</tr>
<tr>
<td>max_cluster2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>field1, field2</td>
<td>String</td>
<td>Field that is used for k-means clustering. It can either be “inputparam”, “score”, or “derivedparam”</td>
</tr>
<tr>
<td>field_range1,</td>
<td>Array of</td>
<td>Within the field which elements will be used for the k-means clustering, specifying {0, 1, 2} indicates only the first three elements is used for the clustering.</td>
</tr>
<tr>
<td>field_range2</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>result_dir3</td>
<td>String</td>
<td>Path of the directory in which to store the resultant XML files</td>
</tr>
<tr>
<td>filename3</td>
<td>String</td>
<td>Name of the resultant k-means and Archetypal Analysis XML file.</td>
</tr>
<tr>
<td>max_cluster3</td>
<td>Integer</td>
<td>Maximum number of clusters permissible for k-means clustering.</td>
</tr>
<tr>
<td>num_archetypes</td>
<td>Integer</td>
<td>Desired number of archetypes.</td>
</tr>
<tr>
<td>field3</td>
<td>String</td>
<td>Field that is used for k-means clustering and Archetypal Analysis. It can either be “inputparam”, “score”, or “derivedparam”</td>
</tr>
<tr>
<td>field_range3</td>
<td>Array of</td>
<td>Within the field which elements will be used for the k-means and Archetypal Analysis, specifying {0, 1, 2} indicates only the first three elements is used for the clustering.</td>
</tr>
<tr>
<td>integer</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
In Figure 5-7b above, the design variants are K-means clustered according to their scores. The resultant design clusters are passed to Figure 5-7c. In Figure 5-7c, the design clusters are further K-means clustered according to their derived parameters (Figure 5-8). Finally, the design clusters from Figure 5-7c are passed to Figure 5-7d. In Figure 5-7d, the design clusters are K-means clustered and subjected to Archetypal Analysis to identify the exemplars for each design cluster.

Figure 5-9 below shows the sub-workflow contained in Figure 5-7d. In Figure 5-9a, Archetypal Analysis and K-means are performed on the input parameters of the design clusters to obtain the archetypes and centroid of each design cluster. The archetypes and centroids are new design variants derived from the original design variants. They represent the typical designs in the design clusters. Thus they need to be evaluated again to validate that their performances are within the range of the original design variants (Figure 5-9c). Architects will have to alter the workflow template by copying the design generation and evaluations used in the optimisation workflow into Figure 5-9b. The validated design variants will then be recognised as the exemplars of the design clusters.
5.3.4 Visualisation workflow template

The performance scores and derived parameters of the design clusters are then visualised as Parallel Coordinate Plots (PCP). PCPs are used to visualise data of high dimensionality on a 2D plane. The use of PCPs facilitates the identification of trends of high-dimensional data. For example, by comparing the two PCPs in Figure 5-10 below, one may observe the trend of design variants that achieve low cooling electricity consumption resulting in higher costs, compared with design variants with higher cooling electricity consumption.

![Figure 5-10 Comparison of two PCP graphs](image)

Figure 5-11 below shows the visualisation workflow template for the generation of PCP. A summary of the inputs are provided in Table 5-6 below. Architects will use this visualisation workflow to go through the XML files. For example, to visualise an XML file for the first time, architects can set the “minmax” parameter to [“a”, “a”] so that the “GeneratePCP” component will automatically set the ranges on the PCP axis. Through this, architects can gradually explore and visualise the XML file.

- `result_directory`: C:\
- `result_filename`: score
- `minmax`: [“a”, “a”]
- `score_labels`: [“cool_elec”, “cost”, “daylight”]
- `xmlfile`: C:\xmlfile.xml
- `field`: score
- `field_range`: [0, 1, 2]

![Figure 5-11 Visualisation workflow template for generating PCPs](image)
<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_directory</td>
<td>String</td>
<td>Directory to save to for the generated PCP graphic</td>
</tr>
<tr>
<td>result_filename</td>
<td>String</td>
<td>File name of the generated PCP graphic</td>
</tr>
<tr>
<td>minmax</td>
<td>Array of an array of floats</td>
<td>Minimum and maximum points of each axis, {{“a”}, {{“a”}} automatically set the minimum and maximum points of each axis according to the input data.</td>
</tr>
<tr>
<td>score_labels</td>
<td>Array of strings</td>
<td>Labels for each axis</td>
</tr>
<tr>
<td>xmlfile</td>
<td>String</td>
<td>XML file to be visualised</td>
</tr>
<tr>
<td>field</td>
<td>String</td>
<td>Field that is visualised. It can either be “inputparam”, “score”, or “derivedparam”</td>
</tr>
<tr>
<td>field_range</td>
<td>Array of integer</td>
<td>Within the field which elements will be used for visualisation, specifying {0, 1, 2} indicates only the first three elements is used.</td>
</tr>
</tbody>
</table>
6 Case Study

The design method and the prototype tool are demonstrated using a case study based on the United World College South East Asia (UWCSEA) Dover design project. The case study uses the workflow templates and shows the results from the proposed design method.

Future Cities Laboratory (FCL) Low Exergy Module was offered a 600m² office space by UWCSEA for the implementation of our research in Low-Exergy (LowEx) cooling in the hot and humid climate of Singapore. The design team had already decided on the building form and interior layout. It is a rectangular building with a courtyard in the middle (Figure 6-1b). Amendments to the 600m² office space are still possible for the implementation of the LowEx cooling systems. Figure 6-1 below shows a rendering of the new building block and the plan of the office space.

For this case study, the building is re-designed with the LowEx approach to demonstrate how the proposed design method is used in a design scenario. The case study is based on UWCSEA’s location, floor area, and architectural intention.

Figure 6-1 UWCSEA new building block (a) Rendering of the north façade (b) The 600m² office space located on the 3rd storey
6.1 Encoding stage

The architects work closely with the HVAC designers for the selection and evaluation of the cooling systems. Radiant cooling panels and Decentralised Ventilation Unit (DVU) are used as described in chapter 2. Radiant cooling panels provide opportunities for integration with other ceiling systems such as lighting and sprinklers, while maintaining a slim profile of around 150 mm. The DVUs and the air ducts are small enough to be integrated into the floor slab. The evaluation method for the radiant cooling panels and DVU is described in chapter 3.

The design schema is formulated, once the LowEx cooling systems are selected. The design schema describes how the design concept is parameterised into a model, the design evaluations required to assess the design concept, and the derived parameters to extract to better understand the design problem.

The case study explores a courtyard typology. The size, dimensions, and position of the courtyard are varied for design exploration. The design is an interplay of built and void space. The external walls are angled at different orientations to reduce solar irradiation, maintain acceptable daylight, and create interesting building forms (Figure 6-2). A range of building envelope materials suitable for the design project are used for the envelope.

As mentioned in Chapter 4.1, when using radiant cooling panels and DVU, architects have to balance the building form, envelope, construction, and cooling systems to satisfy the multiple performance objectives of cooling electricity consumption, daylight, and cost. The ceiling area (interior layout) mainly acts as a constraint for the calculation of the cooling capacity of the radiant cooling panels.
Encoding the design schema is an iterative process. The design schema is represented as a parametric model. The parametric model needs to be well formed, and architects need to observe how the design evaluations react to changes in the input parameters. The design generation procedure will be altered, if the changes in input parameters cause little variation in the evaluation results. The iterations between the design generation and evaluations refine the two procedures. For example, in Figure 6-3d below, the addition of shadings was determined in the encoding stage by running a solar irradiation simulation on the models.

For example, in Figure 6-3d below, the addition of shadings was determined in the encoding stage by running a solar irradiation simulation on the models.

![Figure 6-2 Exploration of courtyard (white) and built area (black) (a–c) Exploration of the 2D configuration of courtyards and built-up area (d) The massing of the building when extruded into 3D form](image)

Architects decide on the derived parameters to be extracted from the evaluated design variants. These derived parameters will be used in the analysis stage to facilitate the understanding of the design problem. Table 6-1 below shows the five derived parameters that is extracted from the evaluated design variants. ETTV and shape factor describe the envelope quality and building form of a design variant. The sensible load gives an indication of the cooling demand. Finally, the panel surface area and panel temperature describe the radiant cooling panels. It is useful to be aware of the panel surface area, as it illustrates how cooling system directly affects the interior design and vice versa. The architects can quickly prototype a few dozen design variants using the encoding workflow template to test the robustness of the design generation and evaluations.
Table 6-1 Derived parameters extracted for data analysis

<table>
<thead>
<tr>
<th>Derived parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Envelope Thermal Transfer Value (ETTV)</td>
<td>An indicator of the quality of the building envelope in the tropical climate.</td>
</tr>
<tr>
<td>Shape factor (CEN, 2007)</td>
<td>A description of the geometry of the design. The shape factor is defined as the ratio of the external envelope to the conditioned floor area.</td>
</tr>
<tr>
<td>Sensible load</td>
<td>The sensible heat load to be removed by the cooling system.</td>
</tr>
<tr>
<td>Panel surface area</td>
<td>The surface area required for the radiant panels.</td>
</tr>
<tr>
<td>Panel temperature</td>
<td>The supply temperature required for the radiant panels.</td>
</tr>
</tbody>
</table>

6.2 Optimisation stage

The design generation procedure was finalised after several iterations, it is described in Figure 6-3 below. In summary, there are nine input parameters for the procedure (Table 6-2).

Figure 6-3 Design generation procedure
a) The building footprint is made up of a four-sided polygon. A point is chosen from the 24 points at each corner. By connecting dots at different positions at each corner, the external walls can be angled at different orientations.

b) A rectangular void is generated according to two parameters: void area and void length. The width of the void is derived by the division of void area by void length.

c) In order to test out the different ways of positioning a courtyard, the void is placed onto the building footprint in ten possible placements.
   - c(0) – c(4) keep the void as one and place it in a different position on the building footprint. c(0) copies the void to the mid-point of the footprint, while c(1) – c(4) copy the void to the four different sides of the footprint.
   - c(5) – c(7) divide the void into halves and arrange them accordingly. c(5) overlaps the halves creating a cross-like geometry. c(6) arranges the halves side by side and c(7) arranges them in a top-down arrangement. They are mapped onto the mid-point of the building footprint.
   - c(8) divides the void into thirds and arranges them in a triangular formation. This is then copied onto the mid-point of the building footprint.
   - c(9) divides the void into quarters and arranges them in a formation of rows and columns. This is then copied onto the mid-point of the building footprint.
   - Due to the variations of the dimension of the void, it is possible for the divided voids in c(5) – c(9) to overlap.

d) The number of storeys of the building is determined by fulfilling the gross floor area of 20,000 m². Strip windows of 2m in height are added to all the external façade. Windows receiving more than 300kW/m²/yr are shaded with 1-m horizontal shades. 300kW/m²/yr is the average solar irradiation received by the north and south façade, according to a base case simulation.

e) Different materials are assigned to the wall and windows.
The design variants are evaluated in terms of cooling electricity consumption, envelope material cost, and daylight. First, the cooling electricity consumption needs to be minimised. It is calculated according to the evaluation method in chapter 3.3. If the cooling load is within the radiant panels’ cooling capacity, the radiant panels are used for the design evaluation. Otherwise, a conventional air-based cooling system with 8°C supply temperature is used for the evaluation. The design evaluation procedure involved calculating the cooling electricity consumption of the building with the chosen system, and normalising it according to the floor area.

Second, the envelope material cost needs to be minimised using a simplified cost indicator for building materials. The cost is calculated according to the surface area of the different façade materials, with materials of higher insulation quality costing more. The cost of the envelope materials are listed in Table 6-3 below. For each envelope material, the surface area is multiplied by its cost per square metre. The sum of the cost of all the materials is the cost of the envelope. Finally, the daylight needs to be maximised. The daylight is measured as the ratio of the floor area receiving
more than 300 lux to the total floor area. The natural daylight is calculated using Radiance lighting simulation (Ward and Shakespeare, 1998). The simulation is conducted using an overcast sky model on the equinox (March 31st), which is the worst-case scenario. The design method uses NSGA2 for the feedback procedure. The details of the NSGA2 are provided in Chapter 4.3.3. 5000 design variants were generated.

Table 6-3 Cost of each envelope material

<table>
<thead>
<tr>
<th>Material name</th>
<th>Cost (dollars/m²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concrete wall, uvalue = 2.3</td>
<td>1</td>
</tr>
<tr>
<td>Brick wall, uvalue = 1.9</td>
<td>1.5</td>
</tr>
<tr>
<td>Insulated concrete wall, uvalue = 0.5</td>
<td>5</td>
</tr>
<tr>
<td>Single glazed, uvalue = 5.2, sc = 0.72</td>
<td>1</td>
</tr>
<tr>
<td>Uncoated double glazed, uvalue = 2.82, sc = 0.81</td>
<td>5</td>
</tr>
<tr>
<td>Double glazed hard lowe, uvalue = 1.86, sc = 0.71</td>
<td>10</td>
</tr>
<tr>
<td>Double glazed reflective lowe, uvalue = 1.6, sc = 0.39</td>
<td>12</td>
</tr>
</tbody>
</table>

6.3 Data analysis stage

The proposed analysis method was applied to the case study. The design variants were clustered based on their performance scores, three design clusters were generated from this analysis. Each design cluster was then clustered based on the Envelope Thermal Transfer Value (ETTV) and shape factor. Nine clusters were generated. Exemplars were extracted from each of these nine clusters. Finally, the design clusters were visualised as Parallel Coordinate Plots (PCPs) and 3D geometry.

In the performance score clustering step, the K-means identified three main types of performance score clusters: LowEx, LowEx-Low Cost, and Non-LowEx. In the derived parameters clustering step, K-means clustered nine derived parameter clusters from the three main performance score clusters: LowEx 1-3, LowEx-Low Cost 1-3, Non-LowEx 1-3.
For each design cluster, two Parallel Coordinate Plots (PCPs) were created using the visualisation workflow template to visualise the scores and the derived parameters. The 3D geometry of the exemplars was generated when they were evaluated and validated in the data analysis workflow. The exemplars are arranged in three rows, the centroids are located in the middle row while the archetypes are at the first and last row (Figure 6-4c-Figure 6-12c). The description on the top of each exemplar indicates the wall and window material. Figure 6-3e above shows the legend of the window and wall material.

6.4 Results

Figure 6-4 to Figure 6-12 show the final result of the data analysis procedure, the exemplars, the PCPs, and a table summary of the scores and derived parameters of each design cluster. They summarise the 5000 design variants generated from the optimisation process.

6.4.1 Design clusters with balanced performances

In comparison to the other eight design clusters, LowEx 1 (Figure 6-4) is the most balanced design cluster. It has a cooling electricity consumption of 35.70–41.36 kWh/m²/yr while maintaining a total envelope material cost of around 25.62–82.78k and daylight of 56.77–87.11% (Figure 6-4a). The exemplars show a variety of building forms (Figure 6-4c). LowEx 1 shows design variants with the highest potential to balance the architectural design and cooling systems, so as to satisfy the multiple performances while allowing interesting building forms.
LowEx-LowCost 3 (Figure 6-5) achieves a similar cooling electricity performance with a lower cost of 6.82-49.34k (Figure 6-5a). However, compared with LowEx 1, it compromises on the daylight performance with 58.25-77.15% (Figure 6-5a), while having little variability in building forms (Figure 6-5c). In order to achieve similar cooling electricity consumption to that of LowEx 1 with a lower cost, design variants in the cluster have building forms with a lower shape factor of 0.32-0.54 (Figure 6-5b), compared with 0.33-0.65 for LowEx 1 (Figure 6-4b). The low shape factor indicates less external surface, which results in a lower daylight performance. For both the design clusters, the sensible load is kept low for the design variants to use a chilled water supply of 14-15 °C (Figure 6-4b & Figure 6-5b) for the radiant cooling panels.
6.4.2 Design clusters with best cooling electricity consumption performance

LowEx-Low Cost 2 (Figure 6-6) has a cost of 25.17–57.69k, which is higher than that of LowEx-Low Cost 3, but lower than that of LowEx 1. It is able to achieve the best cooling electricity consumption performance, 33.36–36.23 kWh/m² (Figure 6-6a), of the nine design clusters. However, the design cluster sacrifices daylight performance, being the design cluster with the worst daylight performance of 36.80–62.54% (Figure 6-6a). It also sacrifices variability in building form, as the exemplars are mainly in cuboid form without any courtyard (Figure 6-6c). A cuboid has the lowest amount of external surface, making it the ideal form to reduce solar heat gain into the interior. The lack of variability in building forms is indicated by the low shape factor of 0.25–0.41 (Figure 6-6b).
LowEx 3 (Figure 6-7) achieves similar cooling electricity performance, but attains better daylight performance of 51.73–72.03% (Figure 6-7a) by having a higher envelope cost, and thus better envelope quality, as indicated by a lower ETTV than that of LowEx-Low Cost 2. The shape factor is similar to that of LowEx-Low Cost 3. Both the design clusters are able to keep the sensible load low by having either a low shape factor or low ETTV, so that a supply temperature of 15°C can be used for the radiant cooling panels (Figure 6-6b & Figure 6-7b).
6.4.3 Design cluster with best cost performance

LowEx-Low Cost 1 (Figure 6-8) has the best cost performance of 5.06–43.38k (Figure 6-8a). It is similar in terms of cooling electricity consumption and daylight performance to LowEx-Low Cost 3. However, as with LowEx-Low Cost 2, it sacrifices building form variability, as most of the building forms are mainly cuboids without any courtyards (Figure 6-8c). LowEx-Low Cost 1 is able to achieve the best cost performance by having a low shape factor while sacrificing the quality of the envelope, as indicated by its much higher ETTV of 45.97–67.23 W/m² (Figure 6-8b), compared with LowEx-Low Cost 2 and 3. LowEx-Low Cost 1 is able to achieve better daylight performance than LowEx-Low Cost 2 because it uses single glazing windows that have poor thermal performance but better daylight performance.
6.4.4 Design cluster with the best daylight performance

LowEx 2 (Figure 6-9) has the best daylight performance of 58.75–85.15% (Figure 6-9a). The design cluster is able to achieve this because of its high shape factor of 0.50–0.74 (Figure 6-9b). A large external surface area increases the chances of the interior receiving daylight. At the same time, there is also more surface area for solar heat to enter the interior. In order to keep the sensible load within the capacity of the radiant cooling panels, a high-quality envelope is required. As a result, this is the most costly design cluster at 58.75–85.15k (Figure 6-9a). A specific building form, namely a building form with four courtyards, is ideal for achieving good daylight performance, which is apparent in the exemplars of the clusters. With more courtyards, there is a higher chance that daylight will penetrate deep into the building.
6.4.5 Design clusters that are unable to employ LowEx cooling systems

Non-LowEx 1, 2, and 3 (Figure 6-10–Figure 6-12) present a series of design variants that are unable to employ LowEx cooling systems but have good daylight and cost performance. They are characterised by having high shape factors and the use of low-quality window material. The high external surface area and bad thermal quality of the envelope leads to the cooling load exceeding the cooling capacity of the radiant panels, resulting in high cooling electricity consumption.
Figure 6-10 (a) Performance score ranges of cluster Non-LowEx 1 (b) Derived parameter ranges of cluster Non-LowEx 1 (c) Exemplars of cluster Non-LowEx 1 (d) Table summary of cluster Non-LowEx 1

Figure 6-11 (a) Performance score ranges of cluster Non-LowEx 2 (b) Derived parameter ranges of cluster Non-LowEx 2 (c) Exemplars of cluster Non-LowEx 2 (d) Table summary of cluster Non-LowEx 2
6.5 How the design method influences the early design stages

By comparing the nine design clusters, architects are able to obtain a bigger picture of the design problem. The visualisation of the performances, derived parameters, and exemplars of each design cluster aids architects in understanding the potential of the design schema. They are able to look at the design variants across all the different performances and be aware of the trade-offs between performances. The exemplars provide a strong visual for architects to see how different architectural designs affect the performances and cooling systems. Qualitatively, architects are able to quickly judge which cluster fits their design intention.

The comparison of design clusters reveals the relationship between building form, cooling systems, and performance. The daylight performance correlates to the shape factor of the design: the higher the shape factor, the better the daylight performance. This is because the larger external surface and amount of glazing leads to better daylight performance. However, the shape factor is limited by the envelope’s ETTV, as an envelope with high ETTV requires a low shape factor to maintain a low sensible load. This will in turn affect the supply temperature of the radiant panels.
and the cooling electricity consumption of the building. The ETTV can be improved to allow for a higher shape factor, but this will inevitably increase the envelope material cost. This relationship can be illustrated by a comparison of LowEx 3 and LowEx-Low Cost 2. Being informed of these relationships facilitates the architects’ understanding of how design changes affect the building systems and performances.

The design clusters also reveal the conflicting nature of the performance objectives. For example, with daylight and cooling electricity consumption, the improvement of one leads to the degradation of the other. In addition, they are both also affected by the envelope material cost. This is illustrated by LowEx 1 and LowEx-Low Cost 3. The higher envelope cost in LowEx 1 enables the design variants to achieve similar cooling electricity consumption while achieving better daylight performance by having a higher envelope cost. This is valuable information when architects are considering trade-offs between performances.

Architect are able to interrogate the performance scores and derived parameter ranges of their preferred design clusters. These scores and parameter ranges can then be used as a basis for developing more detailed designs. For example, the ranges in LowEx 1 and LowEx-Low Cost 3 can be used as references for the detailed design stages.

The design clusters also give an indication of the spatial requirements of the implementation of the radiant cooling panels. In the design of the ceiling space, the required panel surface area needs to be integrated with various other essential components such as lightings and sprinklers. This required area provides a clear target for architects tasked with integrating ceiling systems. The number of DVUs is also provided by the evaluation method: about 600 units are required to ventilate the whole building. As the latent load is kept constant in the design schema, the number of DVUs will also be the same for all the design variants. The required number of units will provide a reference for architects to accommodate for in their design.
7 Summary and Discussion

This chapter provides a summary of the contributions made by this research and discussion on future studies.

7.1 Summary

The research aims to facilitate the integration of LowEx cooling systems into the early stages of architectural design. In order to meet this aim, the research first identified Separation of Sensible and Latent Cooling (SSLC) as an effective cooling strategy in the tropics, then focused on its implementation with the use of radiant cooling panels and Decentralised Ventilation Unit (DVU). As, radiant cooling panels are only feasible when the cooling load of a space is low, and since the cooling load is highly dependent on the construction of the envelope of a building, the envelope plays a crucial role in its application.

Next, considering the data poor situation in the early design stages, simplified calculations were used to model the cooling systems and the architectural design. Coarse inputs are used for the calculations, and results such as, required cooling surface area, supply temperature and cooling load, are produced to directly support design decisions. As the building envelope affects multiple building performances, the design problem is represented as a Multi-Objective Optimisation Problem (MOOP). A multi-objective balancing process is proposed for balancing the cooling systems and the architectural design, while satisfying the multiple performances. Lastly, the simplified modelling approach and the balancing process is consolidated into a three-stage design method. The three stages are encoding, optimisation and analysis as described below:

1. Encoding – Architects and HVAC designers first select the LowEx cooling systems and decide how to model and evaluate them. The architects then formulate and encode the design schema, factoring in the LowEx cooling systems in the encoding stage.

2. Optimisation – The architects input the encoded design schema into the optimisation stage to search for optimal design variants that can balance the architectural design and LowEx cooling system to satisfy the multiple performances.
3. Analysis – The generated design variants are analysed and visualised in the analysis stage with cluster and Archetypal Analysis. This summarise and facilitate the understanding of the generated design variants.

A prototype tool was developed on a workflow management tool to support the design method. Applications were integrated into the workflow management tool as workflow components. Three workflow templates, i.e. optimisation, analysis, and visualisation templates, were composed to aid the execution of the design method. The workflow templates can be easily customised for different design projects.

The method and tool were demonstrated on a case study. By using the tool, architects are able to explore a large number of design variants, and be aware of the potential and limitations of the design schema. Preferred design clusters are interrogated and their performance and derived parameter ranges can be use as references for later design stages. Finally, the results give an indication of the spatial demand of the LowEx cooling systems, and can therefore help architects gain a better understanding of how design changes are affecting the building systems and performances. This would in turn enable the architects to balance the architectural design and LowEx cooling system to satisfy the multiple performances.

7.2 Improvement to the analysis stage

The data analysis procedure still requires architects to manually interpret the results, and the interpretation process might not be as straightforward. It is not an easy task for architects to derive knowledge about the relationship between architectural design, LowEx systems and performances by comparing the design clusters. This requires careful examination and certain amount of prior knowledge of the LowEx systems and performances. Further investigations are required to improve the analysis stage, for it to be more intuitive for the architects to browse through the generated design variants.

It is possible to automate the interpretation process to enable the analysis to output the relationship between the performance scores and the derived and input parameters. Innovization (Deb et al., 2014) is such a technique applied in the engineering field. However, when applied to the
The visualisation uses PCP to provide an overview of the multi-dimensional results of each design cluster. These images are static, while interactive plots would be a more effective method for architects to explore each cluster. This is a feature that needs to be improved for future implementation.

### 7.3 Scalability of the design tool

The prototype tool in its current state is only suitable for the early design stages. The tool needs to cater to the LowEx approach throughout the design process. The building design at this stage is still malleable; there is maximum flexibility in making changes to the design. In the later design stages, this flexibility is significantly reduced. The late design stages correspond to stages 3 (developed design) and 4 (technical design) of the RIBA plan of work 2013 (Sinclair, 2013). The final project brief is defined and the concept is developed into detailed design. The aim of the tool in the late design stages is to ensure that the building design satisfies the performance requirements decided in the early stages.

#### 7.3.1 Design generation

For the early design stages, the design generation procedure is mainly driven by the building form. At the later stages of design, architects will settle on a building form, with design changes only on the component level. The design schema in the later design stages will only require variations on the component level, as opposed to the flexibility required in varying the building form in the early design stages. The resolution of the building information will become finer, as the design process progresses from the conceptual to the development, and finally the construction stage.

Visual Dataflow Modelling (VDM) applications are not suitable for the design generation procedure in the late design stages. In order to model such high-resolution building information, the graph-based interface will become very complicated and unmanageable. Building Information Modelling (BIM) applications with parametric capabilities would be more suitable for the purpose.
BIM will also be able to handle the high-resolution building information efficiently. Therefore, a BIM modelling application such as Digital Project or Revit has to be integrated into the workflow management tool.

7.3.2 Design evaluation

The aim of the design evaluation procedure at this stage is to experiment and receive feedback about the interactions between components of the systems. The evaluation method needs to be detailed and dynamic in order to accurately simulate the behaviours of the building systems. Higher accuracy and resolution are required from the simulation results, in order for architects to make design decisions. Detailed high-resolution simulations such as Building Energy Simulation (BES) and Computational Fluid Dynamics (CFD) are required for the evaluations at this stage.

7.3.3 Feedback

At the later design stages, the design schema tends to generate a smaller number of design variants. There are limited variations as the design is constrained by previous design decisions. Because of this, it would be more appropriate to use an optimisation algorithm that uses a small population for optimisation. One such algorithm is micro Genetic Algorithm (GA) (Coello Coello and Toscano Pulido, 2001). This is in view of the fact that the exploration is more directed, and of the higher computational time needed for the detailed simulations. Optimisation algorithms that use a small population will reduce the total computation time required for the detailed simulation.

7.3.4 Data analysis

A different set of data analytic techniques is required for the late design stages. The available information that can be used for data analysis will differ significantly from the early design stages. In the early design stages, the design method aids architects in balancing the architectural design to accommodate LowEx systems in the design. In the later design stages, architects will examine the design variants to ensure that the performances and interactions of the building systems are as expected in the early stages. From the examination of the results, architects will identify opportunities to create synergies between different components of different building systems.
Appendix

7.4 Appendix A – $T_D^{eq}$, $\Delta T$, and $SF$ for ETTV formula according to different scenarios

Table 0-1 Constants of ETTV formula according to different scenarios

<table>
<thead>
<tr>
<th>Scenarios</th>
<th>$T_D^{eq}$</th>
<th>$\Delta T$</th>
<th>$SF$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wall and fenestration of an office building</td>
<td>12</td>
<td>3.4</td>
<td>211</td>
</tr>
<tr>
<td>Wall and fenestration of a residential building</td>
<td>3.4</td>
<td>1.3</td>
<td>58.6</td>
</tr>
<tr>
<td>Roof of office and residential building</td>
<td>12.5</td>
<td>4.8</td>
<td>485</td>
</tr>
</tbody>
</table>

7.5 Appendix B – Workflow components developed for the workflow templates

7.5.1 Appendix B1 – Workflow components for manipulating the XML schema

A series of workflow components are developed for manipulating the XML schema (Figure 0-1). They form the basis of all workflow components for workflow templates. A summary of the inputs for the nodes is provided in Table 0-2 below.

![Series of workflow components for manipulation of XML](image)

Table 0-2 Summary of inputs for the XML nodes

<table>
<thead>
<tr>
<th>XmlRead</th>
<th>Parameter name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>xml_filepath</td>
<td>String</td>
<td>Path of the XML file to be read</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>XmlExtract</th>
<th>Parameter name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>xml_string</td>
<td>String</td>
<td>String in XML format describing a design variant</td>
</tr>
<tr>
<td></td>
<td>tag_name</td>
<td>String</td>
<td>Name of node which values will be extracted</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>XmlEditNode</th>
<th>Parameter name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>xml_string</td>
<td>String</td>
<td>String in XML format describing a design variant</td>
</tr>
<tr>
<td></td>
<td>edit_node</td>
<td>String</td>
<td>Node tag to be modified</td>
</tr>
<tr>
<td></td>
<td>array_string</td>
<td>Array of float</td>
<td>Values to replace the original values of the node</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>XmlCreateNode</th>
<th>Parameter name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Name</td>
<td>Type</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------</td>
<td>--------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>xml_string</td>
<td>String</td>
<td>String in XML format describing a design variant</td>
<td></td>
</tr>
<tr>
<td>create_node</td>
<td>String</td>
<td>Node to be created</td>
<td></td>
</tr>
<tr>
<td>parent_node</td>
<td>String</td>
<td>Parent node of the created node</td>
<td></td>
</tr>
<tr>
<td>array_string</td>
<td>Array of float</td>
<td>Values for the new created node</td>
<td></td>
</tr>
</tbody>
</table>

**XmlConsolidate**

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xml_array</td>
<td>Array of string</td>
<td>Array of design variant XML strings to be consolidated</td>
</tr>
</tbody>
</table>

“XmlRead” (Figure 0-1a) reads and outputs an array of design variants in XML string format from the XML file. For example, an XML file is plugged in, and the workflow component will output an array of all the design variants in the XML file.

“XmlExtract” (Figure 0-1b) is used to retrieve information from an XML string. The component will output the values of the specified node. For example, a design variant is plugged into the “xml_string” and the “tag_name” is specified as “score”, the workflow component will output an array of score values.

“XmlEditNode” (Figure 0-1c) is used to edit an XML string. For example, a design variant is plugged into the “xml_string” and the “edit_node” is specified as “score”. The “array_string” will then replace the original values. The number of values specified has to match the number of original values.

“XmlCreateNode” (Figure 0-1d) creates a new node for an XML string. There are four inputs. For example, a design variant is plugged into the “xml_string” and the “create_node” is specified as “score” and the “parent_node” as “scores”. The node “score” will then be created under the node “scores”, and populated with the values specified in the “array_string”.

“XmlConsolidate” (Figure 0-1e) consolidates multiple design variant XML strings into a single XML string to be written into an XML file. For example, an array of design variant XML strings is plugged into “xml_array”, the component then outputs a single XML string to be written to a file.

**7.5.2 Appendix B2 – Integrating Houdini3D with Kepler**

A workflow component is developed to integrate Houdini3D with Kepler. Houdini3D is remotely controlled by Kepler through its Python API. The workflow component, “HouDev” (Figure 0-2a), is used for the design generation procedure. Summary of the inputs is provided in Table 0-3 below.
### Table 0-3 Summary of inputs for “HouDev”

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xml_string</td>
<td>String</td>
<td>String in XML format describing a design variant, only requires information about the input parameters, identity, generation, and status</td>
</tr>
<tr>
<td>HoudiniFile</td>
<td>String</td>
<td>Path of the Houdini file that is used for the design generation procedure</td>
</tr>
</tbody>
</table>

Inside the “HouDev” workflow component is a sub-workflow for remotely controlling Houdini3D (Figure 0-2b). The native Kepler workflow component “external execution” is used to execute the external Python script. The Python script (Figure 0-2c) controls Houdini3D through its Application Program Interface (API); it executes the development procedure in Houdini3D and extracts the resultant 3D geometry. The “external execution” workflow component then returns the location of the 3D geometry file for the design evaluation procedure.

Architects have to follow certain guidelines when parameterising the design schema in Houdini3D. In the Houdini3D network that defines the parametric model, two components named “parameters” and “phenotype” are required. The “HouDev” component will activate the script in Figure 0-2c to look for the two nodes. The “parameters” component is constructed from a native Houdini3D “null” component. Architects are to configure the parameter interface of the Houdini3D component to house the input parameters. The Houdini3D parameters have to be named “inputparm{x}”, where x is the count of the input parameter (Figure 0-2d). The input parameters are passed to the “parameters” component in Houdini3D to trigger and execute the parametric model. The 3D geometry is then retrieved from the “phenotype” component. The “phenotype” component is constructed by renaming the native Houdini3D “file” component. Figure 0-2 below illustrates the overall process of how Kepler is linked to Houdini3D. The Houdini3D workflow component can be reused and shared by the architects.
A component is developed to integrate Radiance with Houdini3D. The Radiance component (Figure 0-3a) requires eight inputs (Figure 0-3b). Summary of the inputs is provided in Table 0-4 below.

**Table 0-4 Summary of inputs for Radiance component in Houdini3D**

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sensor grid</td>
<td>Set of points</td>
<td>Each point measures the amount of solar irradiation or illuminance falling on its position.</td>
</tr>
<tr>
<td>geometries</td>
<td>Geometries</td>
<td>Geometries of the design variant with their name and material information</td>
</tr>
<tr>
<td>month</td>
<td>Integer</td>
<td>The month the simulation is conducted.</td>
</tr>
<tr>
<td>day</td>
<td>Integer</td>
<td>The day the simulation is conducted.</td>
</tr>
<tr>
<td>hour</td>
<td>Integer</td>
<td>The time the simulation is conducted.</td>
</tr>
<tr>
<td>latitude</td>
<td>Float</td>
<td>Latitude of the location.</td>
</tr>
<tr>
<td>longitude</td>
<td>Float</td>
<td>Longitude of the location.</td>
</tr>
<tr>
<td>meridian</td>
<td>Float</td>
<td>Meridian of the location.</td>
</tr>
</tbody>
</table>

Inside the Radiance component (Figure 0-3c), a Python script is used to remotely control Radiance. It runs a Radiance simulation, returns the result, and stores it in the geometry of the design variant. The result is then visualised in a false colour diagram, as shown in Figure 0-3d below.
7.5.4 Appendix B4 – Integrating the evaluation method for radiant cooling panels and DVU with Houdini3D

The evaluation method for radiant cooling panels and DVU is integrated into Houdini3D as a series of components. For this research, it was decided to separate them into multiple components, instead of having one component. Figure 0-4 below illustrates the workflow of the simulation in Houdini3D. Figure 0-4a shows a network for constructing a parametric model and preparing it for the evaluation method. When transiting from Figure 0-4a to Figure 0-4b, the geometries need to be properly grouped into their building elements such as wall, floor, roof, and windows.

In Figure 0-4b, the component “ettv_geom_prep” will derive the orientation of the surfaces base on the surface normal. Architects assign materials to the external walls using “set_uvalue” and to glazing using the “set_uvalue_win” component. The geometries with the data are passed to Figure 0-4c, where Radiance is used to calculate the Shading Coefficient of the windows as according to chapter 3 evaluation method, in the “ettv_external_shading” component. With all the necessary data on the design variant, the “ettv” component calculates the ETTTV of the design variant. The “load_calc” component then calculates the cooling load, and “exergy_calc” calculates the system performances (Figure 0-4e). The transparency of the calculation method facilitates the architects’ understanding of the limitations and boundaries of the method. This is an advantage offered by a workflow approach, compared with a black box.
7.5.5 Appendix B5 – Integrating envelope material cost calculation with Houdini3D

A component is developed for envelope material cost calculation. The envelope material cost component (Figure 0-5a) requires only one input (Table 0-5). In addition, architects have to fill in the cost of each material in dollars per square metre. For example, for wall material 0, the cost is 1 dollar per square metre (Figure 0-5b). Inside the envelope material cost component (Figure 0-5d), a sub-workflow calculates the envelope material cost according to the surface area of the envelope and the cost of each material. For each envelope material, the surface area is multiplied by its cost per square metre. The sum of the cost of all the materials is the cost of the envelope.
### 7.5.6 Appendix B6 – Integrating the design evaluation procedures in Kepler

A workflow component was developed to integrate the design evaluations into Kepler. Kepler executes the design evaluations through Houdini3D. The integration process is similar to that described in Chapter 7.5.2. The inputs are summarised in Table 0-6 below.

#### Table 0-6 Summary of inputs for “HouEval”

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>phenotype_filepath</td>
<td>String</td>
<td>Path of the Houdini geometry file (bgeo). The bgeo file has to carry the geometry and necessary data as defined in Figure 0-3, Figure 0-4, and Figure 0-5</td>
</tr>
<tr>
<td>HoudiniFile</td>
<td>String</td>
<td>Path of the Houdini file that is used for the design evaluation procedure</td>
</tr>
</tbody>
</table>

Inside the “HouEval” workflow component is a sub-workflow for remotely controlling Houdini3D (Figure 0-6b). The native Kepler workflow component “external execution” is used for executing the external Python script that controls Houdini3D. The Python script (Figure 0-6c) controls Houdini3D through its API and executes the design evaluation. The “external execution” workflow component then returns the score and derived parameters from the design evaluation procedure.
Architects have to follow certain guidelines when setting up the design evaluation. In the Houdini3D network, there need to be two components named “phenotype” and “result”. The “HouEval” component will activate the script in Figure 0-6c to look for the two components. Architects need to rename a native Houdini “File” component into “phenotype”. The path of the design variant geometry is passed to “phenotype” to trigger and execute the design evaluation. The score is then retrieved from the “result” component. The “result” component is constructed by renaming a native Houdini3D “null” component. Figure 0-6 below illustrates the overall process. The workflow component “HouEval” will then output the score (result) from the design evaluation procedure. The score is not necessary the result of the design evaluation. For example, instead of using the total cooling electricity consumption, architects will normalise the electricity consumption by the building floor area.

A design variant carries three main categories of data: performance scores, derived parameters, and input parameters. Derived parameters are usually extracted during the design evaluation procedure, and thus form one of the outputs in the design evaluation workflow component.

Figure 0-6 Linking Houdini3D design evaluations to Kepler (a) The design evaluation workflow component in Kepler (b) The design evaluation sub-workflow contained within the workflow component (c) The design evaluation sub-workflow calling the Python API of Houdini 3D to remotely control it from Kepler (d) The Houdini3D application that is being remotely controlled by Kepler
7.5.7 Appendix B7 – Integrating scikit-learn K-means in Kepler

The K-means workflow component was used for the data analysis procedure (Figure 0-7a). The workflow component “Kmeans” requires five inputs. The inputs are summarised in Table 0-7 below.

Table 0-7 Summary of inputs for “Kmeans”

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xml_filepath</td>
<td>String</td>
<td>Path of the XML file to be clustered.</td>
</tr>
<tr>
<td>n_clusters</td>
<td>Integer</td>
<td>Maximum number of clusters permissible for the operation.</td>
</tr>
<tr>
<td>result_dir</td>
<td>String</td>
<td>Path of the directory in which to store the resultant XML files.</td>
</tr>
<tr>
<td>field</td>
<td>String</td>
<td>Field used for the clustering. It can be “inputparam”, “score”, or “derivedparam”</td>
</tr>
<tr>
<td>field_range</td>
<td>Array of integers</td>
<td>Within the field which elements will be used for the clustering, specifying {0, 1, 2} uses the first three elements for clustering.</td>
</tr>
</tbody>
</table>

Inside the “Kmeans” workflow component is a sub-workflow for executing K-means with the scikit-learn Python library (Figure 0-7b). The native Kepler workflow component “external execution” is used for executing the external Python script that calls the scikit-learn Python library. The Python script (Figure 0-7c) executes K-means on the design variants. The “external execution” workflow component then returns the file path of the XML files that document the centroid and the clusters. The first output is an array of XML files in which each XML file is one cluster from the clustering result, and the second output is an XML file of all the centroids from the clusters.

7.5.8 Appendix B8 – Integrating PyMF Archetypal Analysis in Kepler

The Archetypal Analysis workflow component was used for the data analysis procedure (Figure 0-8a). The workflow component, “Archetypal”, requires five inputs. The inputs are summarised in Table 0-8 below.
Table 0-8 Summary of inputs for “Archetypal”

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xml_filepath</td>
<td>String</td>
<td>Path of the XML file to be analysed.</td>
</tr>
<tr>
<td>n_archetypes</td>
<td>Integer</td>
<td>Number of archetypes</td>
</tr>
<tr>
<td>result_dir</td>
<td>String</td>
<td>Path of the directory in which to store the resultant XML files.</td>
</tr>
<tr>
<td>field</td>
<td>String</td>
<td>Field used for the clustering. It can be “inputparam”, “score”, or “derivedparam”</td>
</tr>
<tr>
<td>field_range</td>
<td>Array of integers</td>
<td>Within the field which elements will be used for the clustering, specifying {0, 1, 2} uses the first three elements for clustering.</td>
</tr>
</tbody>
</table>

Inside the “Archetypal” workflow component is a sub-workflow for executing the Archetypal Analysis with the PyMF Python library (Figure 0-8b). The native Kepler workflow component “external execution” is used for executing the external Python script that calls the PyMF Python library. The Python script (Figure 0-8c) executes Archetypal Analysis on the design variants of the XML file. The “external execution” workflow component then returns the file path of the XML file that documents the archetypes extracted from the design variants.

Figure 0-8 Summary of the Archetypal Analysis workflow component (a) The Archetypal Analysis workflow component in Kepler (b) The sub-workflow contained within the workflow component (c) The sub-workflow calling the PyMF Python library to execute the analysis

7.5.9 Appendix B9 – Integrating matplotlib PCP in Kepler

The PCP workflow component is used for the visualisation of the results from the data analysis procedure (Figure 0-9a). The workflow component, “PCPtxtfile”, requires four inputs. The inputs are summarised in Table 0-9 below.

Table 0-9 Summary of inputs for “PCPtxtfile”

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>plot_txtfile</td>
<td>String</td>
<td>Path of the text file to be visualised, the data in the text file needs to be sorted into arrays of arrays ({{}, {}, ..., {}}), each array within an array being a polyline that runs across each axis in the PCP</td>
</tr>
<tr>
<td>score_labels</td>
<td>Array of strings</td>
<td>Labels for each axis</td>
</tr>
<tr>
<td>minmax</td>
<td>Array of array of values</td>
<td>Minimum and maximum points of each axis; by specifying {&quot;a&quot;}, {&quot;a&quot;}) the workflow component will automatically set the maximum and minimum points of each axis according to the input data.</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>result_filepath</td>
<td>String</td>
<td>File path to save to for the generated PCP graphic</td>
</tr>
</tbody>
</table>

Inside the “PCPtxtfile” workflow component is a sub-workflow for executing the visualisation with the matplotlib Python library (Figure 0-9b). The native Kepler workflow component “external execution” is used for executing the external Python script that calls the matplotlib Python library. The Python script (Figure 0-9c) visualises the data from the text file. The “external execution” workflow component then returns the file path of the visualisation.

![Figure 0-9 Summary of the PCPtxtfile workflow component](image)

(a) The workflow component in Kepler (b) The sub-workflow contained within the workflow component (c) The sub-workflow calling the matplotlib Python library to generate the PCP.

### 7.5.10 Appendix B10 – Integrating NSGA2 in Kepler

Inside the “NSGA2” workflow component is a sub-workflow for executing the evolutionary algorithm (Figure 0-10b). A Python script is written within the “Python Actor” component. It executes NSGA2 by taking in the inputs. The inputs are explained in Chapter 5.3.2: Optimisation workflow template.

In the first cycle, the NSGA2 script takes in the relevant parameters and randomly generates an initial population of design variants. In the subsequent cycle, the script retrieves the live design variants from the “live_file” for the optimisation process and appends the dead design variants to the “dead_file” at the end of each cycle.
7.6 Appendix C – Kepler workflow template settings for the case study

The optimisation workflow template is used to perform the optimisation process. Table 0-10 below shows the inputs for the optimisation workflow template. The inputs are configured as described in Chapter 5.3.2. The first four input parameters are the four point positions of the building footprint. For each parameter there are 24 points to choose from, and thus the integer range is from 0 to 24 with a step of 1. The void length parameter has a range of 12 m to 85 m with a step of 5 m. The void area parameter has a range of 500 m$^2$ to 800 m$^2$ with a step of 50 m$^2$. There are 10 ways to position the void, and thus the void placement parameter has a range of 0 to 10 with a step of 1. There are four different window materials, and thus the window material parameter has a range of 0 to 4 with a step of 1. There are three different wall materials, and thus the wall material parameter has a range of 0 to 3 with a step of 1.

Figure 0-11 below shows the optimisation workflow template configured for the case study.

Figure 0-12 below shows the sub-workflow in the “DesignGenerationEvaluation” component configured for the case study. There are three “HouEval” components as three design evaluations are defined. The necessary input parameters are filled in as described in Chapters 7.5.2 and 7.5.6.
NSGA2 Settings
- generations: 50
- init_population: 100
- mutation_rate: 0.01
- crossover_rate: 0.9

Input Parameters Settings
- num_type_of_gene: \{\{4,1\},\{1,1\},\{1,1\},\{1,1\}\}
- range_of_gene: \{\{0,24,1\},\{12,83,5\},\{500,801,50\},\{0,10,1\},\{0,4,1\},\{0,3,1\}\}
- num_of_sets: \{1\}

Performance Scores Settings
- score_names: \{cool_elec","cost","daylight"\}
- scores_min_max: \{min',min',max\}

File Settings
- live_file: C:\live.xml
- dead_file: C:\dead.xml

---

Figure 0-11 Optimisation workflow template configured for the case study

Figure 0-12 Sub-workflow inside the "DesignGenerationEvaluation" workflow component
<table>
<thead>
<tr>
<th>Input Parameters</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>num_type_of_gene</td>
<td>{{1, 1}, {1, 1}, {4, 1}, {1, 1}, {1, 1}, {1, 1}}</td>
</tr>
<tr>
<td>range_of_gene</td>
<td>{{12, 83, 5}, {0, 10, 1}, {0, 24, 1}, {500, 801, 50}, {0, 4, 1}, {0, 3, 1}}</td>
</tr>
<tr>
<td>no_of_gene_sets</td>
<td>{1}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Performance Objectives</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>scores_names</td>
<td>{“exergy”, “cost”, “daylight”}</td>
</tr>
<tr>
<td>min_max_list</td>
<td>{“min”, “min”, “max”}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NSGA2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>generations</td>
<td>50</td>
</tr>
<tr>
<td>init_population</td>
<td>100</td>
</tr>
<tr>
<td>mutation_rate</td>
<td>0.01</td>
</tr>
<tr>
<td>Crossover_rate</td>
<td>0.9</td>
</tr>
</tbody>
</table>

The analysis workflow template is used to execute the data analysis method. Table 0-11 below shows the inputs for the analysis workflow template. The workflow produces two XML files for each design cluster – one documents the design variants in the design cluster, while the other documents the exemplars of the design cluster. The settings for “max_clusters1”, “max_clusters2”, and “max_clusters3” are all set to 5 because a maximum of 5 design clusters is more manageable. The “KmeansScores” sub-workflow executes the performance scores clustering step, and all three of the performance scores are used for the clustering; thus “field1” is set to “score” and “field_range1” is set to {0, 1, 2}. The “KmeansDerivedParm” sub-workflow executes the derived parameters clustering step, and only the first two derived parameters, ETTV and shape factor, are used for the clustering. Thus “field2” is set to “derivedparam” and “field_range2” is set to {0, 1}.

The “Exemplars” workflow executes the exemplars extracting step, and all nine input parameters are used for the Archetypal Analysis and K-means. Thus “field3” is set to “inputparam” and “field_range3” is set to {0, 1, 2, 3, 4, 5, 6, 7, 8}. “num_archetypes” is set to 10 because 10 archetypes are sufficient to sample a design cluster. Nine design clusters are generated from the analysis.
workflow template. Figure 0-13 below shows the workflow template configured for the case study.

For the validation of the exemplars in the “Exemplar” sub-workflow (Figure 0-14), architects have to replace the “DesignGenerationEvaluation” with the component used in the optimisation workflow.

![Figure 0-13 Data analysis workflow configured for the case study](image)

![Figure 0-14 Exemplar sub-workflow: the “DesignGenerationEvaluation” (indicated by the dotted box) has to be customised for each design schema](image)

Table 0-11 Settings for data analysis workflow

<table>
<thead>
<tr>
<th>Kmeans Scores</th>
<th>Kmeans Derived Parameters</th>
<th>Exemplars</th>
</tr>
</thead>
<tbody>
<tr>
<td>max_cluster1</td>
<td>5</td>
<td>max_cluster3</td>
</tr>
<tr>
<td>field1</td>
<td>score</td>
<td>field2</td>
</tr>
<tr>
<td>field_range1</td>
<td>{0,1,2}</td>
<td>field_range2</td>
</tr>
<tr>
<td>field_range3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 0-15 below shows an example of generating a score PCP of the design variants. The inputs are explained in Chapter 5.3.4. In this example, the template is visualising the first three scores of the design variants. The cooling electricity consumption, cost, and daylight have maximums of 72, 100, and 88, and minimums of 33, 5, and 36 for the axis, respectively. Figure 0-16 below shows an example of generating a derived-parameter PCP of the design variants. In this example, the template is visualising the first five derived parameters of the design variants. The ETTV, shape
factor, sensible load, panel surface area, and panel temperature have maximums of 68, 0.8, 1854, 22, and 15, and minimums of 20, 0.25, 1062, 0, and 8 for the axis, respectively.

- result_directory: C:\
- result_filename: score
- minmax: {{72,100,88},{33,5,36}}
- score_labels: {"elec_consumption","cost","daylight"}
- xmlfile: C:\xmlfile1.xml
- field: score
- field_range: {0,1,2}

![Figure 0-15 Visualisation workflow score example](image1)

- result_directory: C:\
- result_filename: derivedparam
- minmax: {{68,0.8,1854,22,15},{20,0.25,1062,0,8}}
- score_labels: {"ettv","shape_factor","sen_load","panel_srf","panel_temp"}
- xmlfile: C:\overall.xml
- field: derivedparam
- field_range: {0,1,2,3,4}

![Figure 0-16 Visualisation workflow derived parameters example](image2)
Figures

Figure 1-1 Psychrometric chart showing 8760 hours of Singapore’s weather, with the comfort range delineated by the blue box (image generated from (“Climate Consultant,” 2015) with weather file from (“EnergyPlus Energy Simulation Software,” 2014)) ........................................................................... 14

Figure 2-1 The basic refrigeration cycle (redrawn and adapted from (McDowall, 2006, p. 79)) ........... 24

Figure 2-2 (a) Sensible cooling (b) Latent cooling (redrawn from image generated from (“Climate Consultant,” 2015) with weather file from (“EnergyPlus Energy Simulation Software,” 2014)) ........... 27

Figure 2-3 Schematics of an all-air system (redrawn from (Yap et al., 2011)) .................................. 28

Figure 2-4 Multi-functional radiant cooling panel ............................................................................. 30

Figure 2-5 (a) DVU integrated into the concrete slab (b) DVU integrated into the raised floor .......... 32

Figure 2-6 Air ducts integrated with the floor slabs before the concrete pouring ............................. 33

Figure 2-7 SSLC implementation with radiant cooling panels and DVU (adapted and redrawn from (Yap et al., 2011)) .................................................................................................................. 34

Figure 3-1 Illustration of the evaluation method ..................................................................................... 42

Figure 4-1 K-means process (a) initial clusters (b) iterations (c) final clusters (redrawn from (Han et al., 2012, p. 453)) .................................................................................................................. 59

Figure 4-2 Elbow method: the turning point of the curve marked in red (redrawn from (Everitt and Hothorn, 2011, p. 181)) ......................................................................................................... 60

Figure 4-3 Archetypes (top red) and Centroids (bottom blue) (redrawn from (Hastie et al., 2009, p. 557)) .................................................................................................................................................. 61

Figure 4-4 Procedure of proposed analysis method .................................................................................. 62

Figure 4-5 Illustration of the procedure of the proposed analysis method ............................................. 62

Figure 4-6 Three-stage structure of the proposed design method .......................................................... 63

Figure 5-1 XML schema for design method ............................................................................................ 71

Figure 5-2 (a) Encoding of design schema into parametric model in Houdini3D (b) Testing the parametric model with the design evaluation .................................................................................. 74
Figure 5-3 (a) Encoding the design schema (b) The optimisation workflow template configured for the encoding stage (c) The sub-workflow in the encoding workflow template that requires customisation for each test attempt ................................................................. 75

Figure 5-4 The optimisation workflow template in Kepler (a) The inputs required for the execution of the workflow (b) The NSGA2 workflow component (c) The “DesignGenerationEvaluation” sub-workflow (d) Writing the results from the evaluations to the XML file ................................................................. 76

Figure 5-5 (a) Settings for “num_type_of_gene” (b) Settings for “range_of_gene” (c) Settings for “num_of_set” ........................................................................................................................................ 77

Figure 5-6 Sub-workflow in the “DesignGenerationEvaluation” workflow component (a) The input parameters for developing a design variant (b) The “HouDev” workflow component (c) The “HouEval” workflow components (d) The results and the derived parameters from the design evaluation procedures sorted into the XML format and passed as output ........................................................................ 80

Figure 5-7 Data analysis template workflow for post-processing optimisation result (a) The inputs required for the execution of the workflow (b) K-means of the performance scores of the design variants (c) K-means of the derived parameters of the design variants (d) Identification and validation of the exemplars of the design clusters .................................................................................................. 81

Figure 5-8 The sub-workflow contained within Figure 5-7c .................................................................................. 82

Figure 5-9 The sub-workflow contained within Figure 5-7d (a) Performing the Archetypal Analysis and K-means to identify the exemplars of the cluster (b) Copying the design generation and evaluations used in the optimisation workflow for validating the exemplars (c) Validating the exemplars generated by ensuring they are within the range of the cluster ........................................................................ 82

Figure 5-10 Comparison of two PCP graphs ........................................................................................................ 83

Figure 5-11 Visualisation workflow template for generating PCPs ........................................................................ 83

Figure 6-1 UWCSEA new building block (a) Rendering of the north façade (b) The 600m² office space located on the 3rd storey ........................................................................................................ 85

Figure 6-2 Exploration of courtyard (white) and built area (black) (a–c) Exploration of the 2D configuration of courtyards and built-up area (d) The massing of the building when extruded into 3D form ........................................................................................................ 87

Figure 6-3 Design generation procedure ........................................................................................................ 88
Figure 6-4 (a) Performance score ranges of cluster LowEx 1 (b) Derived parameter ranges of cluster LowEx 1 (c) Exemplars of cluster LowEx 1 (d) Table summary of cluster LowEx 1 ........................................... 93

Figure 6-5 (a) Performance score ranges of cluster LowEx-Low Cost 3 (b) Derived parameter ranges of cluster LowEx-Low Cost 3 (c) Exemplars of cluster LowEx-Low Cost 3 (d) Table summary of cluster LowEx-Low Cost 3 .............................................................................................................. 94

Figure 6-6 (a) Performance score ranges of cluster LowEx-Low Cost 2 (b) Derived parameter ranges of cluster LowEx-Low Cost 2 (c) Exemplars of cluster LowEx-Low Cost 2 (d) Table summary of cluster LowEx-Low Cost 2 .............................................................................................................. 95

Figure 6-7 (a) Performance score ranges of cluster LowEx 3 (b) Derived parameter ranges of cluster LowEx 3 (c) Exemplars of cluster LowEx 3 (d) Table summary of cluster LowEx 3 ........................................... 96

Figure 6-8 (a) Performance score ranges of cluster LowEx-Low Cost 1 (b) Derived parameter ranges of cluster LowEx-Low Cost 1 (c) Exemplars of cluster LowEx-Low Cost 1 (d) Table summary of cluster LowEx-Low Cost 1 .............................................................................................................. 97

Figure 6-9 (a) Performance score ranges of cluster LowEx 2 (b) Derived parameter ranges of cluster LowEx 2 (c) Exemplars of cluster LowEx 2 (d) Table summary of cluster LowEx 2 ......................... 98

Figure 6-10 (a) Performance score ranges of cluster Non-LowEx 1 (b) Derived parameter ranges of cluster Non-LowEx 1 (c) Exemplars of cluster Non-LowEx 1 (d) Table summary of cluster Non-LowEx 1 .............................................................................................................. 99

Figure 6-11 (a) Performance score ranges of cluster Non-LowEx 2 (b) Derived parameter ranges of cluster Non-LowEx 2 (c) Exemplars of cluster Non-LowEx 2 (d) Table summary of cluster Non-LowEx 2 .............................................................................................................. 99

Figure 6-12 (a) Performance score ranges of cluster Non-LowEx 3 (b) Derived parameter ranges of cluster Non-LowEx 3 (c) Exemplars of cluster Non-LowEx 3 (d) Table summary of cluster Non-LowEx 3 .............................................................................................................. 100

Figure 0-1 Series of workflow components for manipulation of XML ........................................... 107

Figure 0-2 Linking Houdini3D to Kepler (a) The Houdini3D workflow component in Kepler (b) The sub-workflow contained within the workflow component (c) The sub-workflow calls the Python API of Houdini 3D to remotely control it from Kepler (d) The Houdini3D application that is being remotely controlled by Kepler .............................................................. 110
Figure 0-3 Radiance integrated into Houdini3D (a) The radiance component in Houdini3D (b) The inputs for the radiance component (c) The Python script that extracts the geometry from Houdini3D and executes either the solar irradiation or daylight simulation (d) The results of the simulation are visualised as a falsecolour diagram in Houdini3D

Figure 0-4 Workflow of evaluation method in Houdini3D (a) The geometry of the design variant (b) Assigning building orientation and materials to the geometry of the design variant (c) Calculating the Shading Coefficient (SC) using Radiance (d) The ETTV calculation (e) Calculating the exergy consumption from the ETTV result

Figure 0-5 Envelope material cost calculation integrated into Houdini3D (a) The envelope material cost component in Houdini3D (b) The inputs for the envelope material cost component (c) The geometry with information about the material for each surface (d) The sub-workflow of the envelope material cost component

Figure 0-6 Linking Houdini3D design evaluations to Kepler (a) The design evaluation workflow component in Kepler (b) The design evaluation sub-workflow contained within the workflow component (c) The design evaluation sub-workflow calling the Python API of Houdini 3D to remotely control it from Kepler (d) The Houdini3D application that is being remotely controlled by Kepler

Figure 0-7 Summary of the K-means workflow component (a) The K-means workflow component in Kepler (b) The sub-workflow contained within the workflow component (c) The sub-workflow calling the scikit-learn Python library to execute the K-means

Figure 0-8 Summary of the Archetypal Analysis workflow component (a) The Archetypal Analysis workflow component in Kepler (b) The sub-workflow contained within the workflow component (c) The sub-workflow calling the PyMF Python library to execute the analysis

Figure 0-9 Summary of the PCPtxtfile workflow component (a) The workflow component in Kepler (b) The sub-workflow contained within the workflow component (c) The sub-workflow calling the matplotlib Python library to generate the PCP

Figure 0-10 Summary of the NSGA2 workflow component in Kepler (a) The NSGA2 workflow component in Kepler (b) The sub-workflow contained within the NSGA2 workflow component and the Python script executing the algorithm

Figure 0-11 Optimisation workflow template configured for the case study
Figure 0-12 Sub-workflow inside the “DesignGenerationEvaluation” workflow component ..........119
Figure 0-13 Data analysis workflow configured for the case study ........................................121
Figure 0-14 Exemplar sub-workflow: the "DesignGenerationEvaluation" (indicated by the dotted box) has to be customised for each design schema ..............................................................................................121
Figure 0-15 Visualisation workflow score example ........................................................................122
Figure 0-16 Visualisation workflow derived parameters example ..................................................122
Tables

Table 4-1 Summary of the reviewed research that used optimisation for the early design stages.....48
Table 5-1 Attributes of a design variant ..................................................................................71
Table 5-2 Summary of the inputs for the optimisation workflow template............................76
Table 5-3 Example of the settings and results of specifying the input parameters in the Kepler workflow .................................................................................................................78
Table 5-4 Example settings of the performance scores................................................................79
Table 5-5 Summary of the inputs for the analysis workflow template ......................................81
Table 5-6 Summary of the inputs for the visualisation workflow template ...............................84
Table 6-1 Derived parameters extracted for data analysis........................................................88
Table 6-2 Input parameters .......................................................................................................90
Table 6-3 Cost of each envelope material ................................................................................91
Table 0-1 Constants of ETTV formula according to different scenarios .................................107
Table 0-2 Summary of inputs for the XML nodes ..................................................................107
Table 0-3 Summary of inputs for “HouDev” ...........................................................................109
Table 0-4 Summary of inputs for Radiance component in Houdini3D ....................................110
Table 0-5 Summary of inputs for envelope material cost component ....................................113
Table 0-6 Summary of inputs for “HouEval” .........................................................................113
Table 0-7 Summary of inputs for “Kmeans” .........................................................................115
Table 0-8 Summary of inputs for “Archetypal” .....................................................................116
Table 0-9 Summary of inputs for “PCPtxtfilel”......................................................................116
Table 0-10 Settings for optimisation workflow .....................................................................120
Table 0-11 Settings for data analysis workflow .....................................................................121
Bibliography


Presented at the 29th Advanced Building Ventilation and Environmental Technology Conference (AIVC Conference 2008), Kyoto, Japan.


Woodbury, R., Burrow, A., 2006. “Whither design space?” *AI EDAM* 20, 63–82. doi:10.1017/S0890060406060057


