The Intelligent Machine in Urban Open Space: Sensing Urban Data and Performing Architectural Behaviour

Thesis

How to cite:


For guidance on citations see FAQs.

© 2021 Hyun-Jae Nam

https://creativecommons.org/licenses/by-nc-nd/4.0/

Version: Version of Record

Link(s) to article on publisher’s website:
http://dx.doi.org/doi:10.21954/ou.ro.00013990

Copyright and Moral Rights for the articles on this site are retained by the individual authors and/or other copyright owners. For more information on Open Research Online’s data policy on reuse of materials please consult the policies page.

oro.open.ac.uk
The Intelligent Machine in Urban Open Space: Sensing Urban Data and Performing Architectural Behaviour

by Hyun-Jae Nam

A Thesis Submitted in Fulfilment of the Requirements for the Degree of Doctor of Philosophy in Architecture

Graduate School of Architecture at the Architectural Association, Affiliated Research Centre of the Open University

November 2021
Abstract

The ambition of the design is to produce an intelligent machine that can capture real-time urban data pertaining to social events and environmental conditions in an urban open space, use that data to optimise the use of space and to manipulate the environmental phenomena for specific events by controlling kinetic structures. The Literature Review identifies and classifies how intelligent systems have been discussed for architectural spaces, cities and machines, and examines how intelligence has been developed within three domains: architectural intelligence, urban intelligence, and artificial intelligence. It identifies cohesive interpretations, thoughts, and alternatives that are commonly associated with the term ‘intelligent’, focusing on those that have led to the development of computational techniques with consideration given to their actual applications. The primary concepts of the intelligent machine originated with cybernetic applications in the field of architectural design in the 1960s with proposals for socially engaging architectural machines, and the later development of architectural devices that sense and act on changes in users’ activities. This research further develops such experiments in an open source computational platform, and tests how specific context awareness can be coupled to active devices to produce intelligent actions. With the advancement of information technologies, the ways of sharing real-time information have enabled citizens to rapidly come together and create temporary events in an impromptu way. The design of the intelligent machine for urban open space brings the capacity to optimise the organisation for such events in relation to optimal spatial requirements with appropriate intelligent responses to environmental phenomena including light/shade, temperature and acoustics. An algorithm was designed to regulate the physical body of the machine, consisting of foldable structures, by means of logic-based rulesets. Simulations were carried out to monitor and evaluate the machine’s responsiveness to information on real-time circumstances in an urban open space and its consequent behaviours.
Acknowledgements

I am sincerely grateful to my director of studies and first supervisor Dr Michael Weinstock for his supervision, inspiration, support and encouragement during my research. His profound knowledge, precise insight and expertise solidified my thesis. His advice and critical views on my design motivated me to challenge diverse trials and produce fruitful experimental results. I also would like to express my sincere thanks to my second supervisor Dr George Jeronimidis, whose comments and suggestions were extremely helpful to clarify the research. His feedback guided my revision of my thesis, making it more understandable. To conduct my research, meetings with my supervisors were indeed invaluable. It was always enjoyable to have interesting discussions about my thesis with my colleagues and friends. I am thankful to Aiman Tabony, Anna Font Vacas, Elif Erdine, Milad Showkatbakhsh and Mohammed Makki, who shared not only research skills but also computational techniques, collaborating at the EmTech studio. I express my appreciation to Dummoen Techamai, David Hutama Setiadi, Lukas Pauer, Naina Gupta and Shu Hui Andrea Goh for their support during the PhD programme at the AA school. Words are not enough to describe my deep gratitude to my family, my father Hyo-Chae Nam, my mother Seon-Ok Ha, and my brother Kyung-Hun Nam, for their love, help and support.
## Table of Contents

**Chapter 1. Introduction**

1.1. Introduction ........................................ 1  
1.2. Intelligence and Architectural Machine .......... 3  
1.3. Research Context .................................. 4  
1.4. Research Aim ..................................... 6  
1.5. Research Questions ................................ 8  
1.6. Research Methodology .............................. 9  
1.7. Research Contribution ............................. 10  
1.8. Thesis Structure .................................. 11  

**Chapter 2. Literature Review:**

Intelligent Embedded Systems in Architectural Spaces, Cities and Machines

2.1. Introduction ......................................... 19  
2.2. Intelligent Machine in Architectural Space .... 22  
2.2.1. Cybernetic Approach to Architectural Intelligence 22  
2.2.1.1. Cybernetics ................................. 22  
2.2.1.2. The Architectural Vision of Cybernetic Application 23  
2.2.1.3. Cedric Price ................................ 24  
2.2.1.4. Nicholas Negroponte ........................ 28  
2.2.2. Architectural Interactivity and Responsiveness (State of the Art) 31  
2.2.2.1. Interactive Architecture/Responsive Landscape 31  
2.2.2.2. Towards Living Systems .................... 35  
2.2.3. Technological Approach to Architectural Intelligence 39  
2.2.3.1. Intelligent Environments/Ambient Intelligence 39  
2.2.3.2. Intelligent Buildings ....................... 40  
2.2.4. Discussion ...................................... 41  
2.3. Data-Driven Urban Intelligence ..................... 43  
2.3.1. Data Visualisation ................................ 43  
2.3.1.1. Defining Data .............................. 43  
2.3.1.2. A Brief History of Data Visualisation .. 43  
2.3.2. Urban Information in Patterns ................. 48  
2.3.2.1. Kevin Lynch ............................... 48  
2.3.2.2. Richard Saul Wurman ...................... 50  
2.3.3. Visualising Data in Cities ...................... 52  
2.3.3.1. Intelligent Smart City ..................... 55  
2.3.3.2. Sensing Real-Time Urban Data .......... 55  
2.3.3.3. Smart City ................................ 58  
2.3.4. Discussion ...................................... 60
List of Figures

Figure 1.1. Three domains in the literature review and their intersections applied to the design of an intelligent machine in urban open space.

Figure 2.1. Cedric Price’s drawing of the Fun Palace. 1965. Source: Cedric Price Archives, Canadian Centre for Architecture, Montreal.

Figure 2.2. Floor plan of the Fun Palace, drawn by Cedric Price, final version. Source: Cedric Price Archives, Canadian Centre for Architecture, Montreal.

Figure 2.3. John and Julia Frazer’s system that modelled the Generator’s layouts. Source: Canadian Centre for Architecture, Montreal.

Figure 2.4. The ‘SEEK’ project displayed at the Software exhibition at the Jewish Museum. Source: Nicholas Negroponte (1975).

Figure 2.5. Examples of Interactive Architecture. Top: FreshH2O Expo, NOX. 1994; Bottom: Aegis Hypo-Surface, the deCOi office. 2000. Source: Cedric Price Archives, Canadian Centre for Architecture, Montreal.


Figure 2.8. The chart of planetary movements (the tenth century). Source: Funkhouser (1936).

Figure 2.9. Snow’s dot map of a cholera outbreak in the Soho area. Source: Snow (1855).

Figure 2.10. The French publication Album de Statistique Graphique, Left: Movement of passengers and goods in the main railway stations. 1882; Right: Passengers travelling by train in France. 1889. Source: <https://manometries17.rssing.com/chan-9651125/all_p5.html> (accessed 1 December 2020).

Figure 2.11. Kevin Lynch’s cognitive map. Left: Outline map of the Boston peninsula; Right: Boston image as seen in the field. Source: Lynch (1960).

Chapter 1. Introduction
1.1. Introduction

The information technology revolution has led to exponential growth in ways of sharing information. The boundaries of physical space no longer limit the ways in which information can be shared. Using information and communication technologies (ICTs), members of the public can organise events of individual interest. Nowadays, the easy access to spatiotemporal information via ICTs enables the public to interact with physical space more directly. Information on their activities is shared and stored with the technologies.

As a result of advancements in sensor and data-processing technologies, the development of systems in cities has focused on real-time responsiveness. The ‘smart city’ slogan has directed the approach of city planning towards real-time monitoring and controlling technological systems in cities (e.g. transportation, energy and security) (Deakin and Waer, 2012; Picom, 2015). On the other hand, using sensors and control systems, the domain of architectural design has been extended in order to deal with the changeable states of physical spaces (e.g. controllable facades and roofs, interactive media walls, and kinetic structures). Recent movements in interactive and responsive architecture have aimed to challenge intelligent architecture to propose new means of utilising spaces that maximise architectural adaptability (Cantrell and Holzman, 2015; Yiamoudes, 2016).

However, the definition of the term ‘intelligent’ remains still a difficult concept to be described in architectural design. The term ‘intelligent’ or ‘smart’ has been targeted at technological achievement, which is mainly oriented towards developing digital tools or data-processing techniques. There is a gap between the emerging knowledge on intelligent technologies and its applications in architectural design. In this research, it is questioned as to whether an architectural space can act as a machine that is intelligent to control physical forms that adapt to real-time circumstances (e.g. social activities and environmental conditions), with the task of assisting the usage of a public open space in a city.

Public open spaces in cities have become valuable due to the increase or growth of the density and volume of buildings, as well as a large population influx. Urban open spaces are not only places for walking, sitting and resting but also places for citizens to congregate in cities. Recently, the method used to organise public activities in urban open spaces has been directly connected to accessing and sharing information via digital spaces. Along with the increase in the means of collecting data, spatiotemporal data have included information on social activities in cities. Such data can be used to accommodate users’ demands for social space.

This research explores the field of intelligence for designing machines that deal with information. The term ‘intelligent’ is not clearly defined but continues to be widely used for human-made artefacts (machines). The term ‘intelligent machine’ describes a machine that can process information to sense the presence of circumstances and adjust its actions in order to accomplish specific tasks. In order to understand what intelligence is and how intelligence works for machines, the research reviews not only the field of architecture but also relevant studies of intelligence for designing artefacts that are capable of processing information. After investigating the term and its applications, the design experiment tests how intelligent systems can be applied to architectural space. In this research, architectural spaces that can perform actions or behaviours through spatial components were regarded as architectural machines. The design experiment focuses on developing an architectural machine that is capable of acquiring urban data, pertaining to the contexts of social activities and environmental conditions in an urban open space, and, accordingly, controlling kinetic structures which can adapt flexibly to the input data in real-time.

This research aims to contribute to the development of architectural machines. In this research, an architectural machine’s context awareness was investigated, as it was considered to be one of the important abilities to be called ‘intelligent’. The design of the architectural machine focused on not only physical conditions controlling the motions of kinetic structures but also its ability of information processing embedded in the architectural body. Rulesets or logical sequences in algorithms were developed to read input data, select and evaluate the contexts in the data, and decide on the best option for controlling kinetic structures. The task of assisting public events in urban open space was given to the architectural machine, and its behaviours were programmed through the use of algorithms. The design suggests a strategy that encourages public activities, which are organised through digital space, in urban open spaces.

1.2. Intelligence and Architectural Machine

The word ‘intelligence’ is derived from the Latin verb intelligere, which means to understand, comprehend and perceive. The word has been variously defined and informally described in many fields (e.g. philosophy, psychology, biology, and, lately, computer science) (Hernandez, 1998). However, it remains arguable that there is no single generalised definition of intelligence, and is difficult to fully encompass diverse opinions thereupon in order to conclude a standard definition (Legg and Hutter, 2007b). According to Legg and Hutter (2007a), there are multiple definitions of intelligence which share common features in describing its meaning. Based on the
collection of definitions, Legg and Hutter (2007a) suggested the following definition: ‘Intelligence measures an agent’s ability to achieve goals in a wide range of environments.’

In fact, the word has been widely expressed by computer scientists with regard to the evolution of computing machines and artificial systems. Alan Turing (1950) argued that the question of what thinking is or what intelligence is for machines is vague, but the output from a machine’s performance can be judged by a test. In the early stages, computer scientists focused mainly on how machines can emulate the human way of thinking, as intelligence is commonly considered to be a property in a human mind/brain. The term ‘intelligent’ has been expressed to enhance the abilities of machines to simulate the principles of cognitive processes.

However, it is contentious as to whether intelligence is possessed solely by humans and the development of machines needs only to exhibit the human way of thinking. The domain of computational intelligence has produced heuristic searching techniques inspired by a wide range of living beings’ systems. Searle (1980) argued that there is general AI and narrow AI: general AI focuses on a machine’s capability of performing all of the tasks that a human is capable of performing, aiming to mimic a human’s abilities; meanwhile, narrow AI develops a machine to handle one particular task. Computer scientists have researched computer algorithms to produce diverse problem-solving tools.

One of the ways in which to evaluate a machine’s performance or its intelligence is to assess how it solves a given problem or performs a task in a given timeframe. Downing (2015, p.4) described his definition of intelligence as ‘doing the right thing at the right time, as judged by an outside human observer’. He pointed out that intelligence can be evaluated by the result of a process within time measurements, and that defining intelligence is not based on the process itself, but rather on the result of the process. Intelligence for machines is not to be defined by their inner data processing, but rather is the ability to be evaluated by their outcomes. Intelligence can be observed only by means of outcomes that are processed using complex inner processes (Downing, 2015). It is noticeable that intelligence constitutes an entity’s inner ability to adapt to changes of its outer factors, with one of the factors of intelligence being that of time.

In the field of architecture, the term ‘intelligent’ has been variously interpreted in describing computing technologies associated with spatial conditions. However, the term has not been sufficiently argued for architectural design projects. It is an ambiguous concept with respect to its meaning, as it represents a cognitive ability which is highly related to ‘processes’ in between inputs and outputs. It is questionable as to how an architectural machine can be capable of acquiring information, which is rooted in the logic-based processes between inputs and outputs, in order to exhibit reasonable behaviours through the use of spatial components.

Cedric Price and Nicholas Negroponte first examined the design of acting machines that are capable of acquiring information on users and controlling physical spaces, heavily influenced by cybernetics. Price’s Fun Palace and Generator projects suggested socially interactive spaces that can act as machines which adapt spatial conditions to occupants’ changing activities (Mathews, 2006). Meanwhile, Negroponte’s SEEK project tested how a machine can control physical space in order to propose a constantly changing space. Price and Negroponte experimented with programming computer software that controls architectural hardware (Steenson, 2016). While extending the realm of architectural practice, they questioned whether an architectural space could accommodate users’ constantly changing needs.

Architectural machines have been designed by assembling and integrating architectural components with responsive technologies, including sensors, processors and effectors. For the most part, the actions of projects on interactive architecture and responsive landscapes were reactive to the presence of people. The results of those projects showed limitations with regard to being responsive to contexts or the meanings of the states of people. Architectural machines using motion sensors could exhibit reactive actions due to the lack of input information in response to whether or not there are people nearby. An architectural machine capable of being responsive to the contexts of activities carried out by people has rarely been investigated.

To design an intelligent machine for architectural space, its information processing needs to involve the consideration of how it can perform actions with respect to a given task. The difference between architectural interactivity/responsiveness and architectural intelligence can be judged by whether the output results in a simple action reactive to external stimuli or the output is selected as being the best option among multiple choices in order to achieve a given goal. Architectural intelligence requires the ability to acquire the contexts of circumstances and exhibit reasonable behaviours. The challenge of designing an architectural machine which is intelligent lies in how the machine can perform a task to deal with an issue addressed in the real world.

1.3. Research Context

Digitally constructed environments, which have shifted our conventional understanding of the concept of space, have radically influenced new social formations and furthered the transformation of the public sphere (Barlas and Caliskan, 2006). As a consequence of the advancement of ICTs, the logistical demands of forming traditional communities have declined, as organising such communities is increasingly no longer place-based (Evans, 2013). Evans (2013, p.84) pointed...
out that ‘the old spaces of community were either already dead or were dying’ and that the place-based community has disappeared. Moreover, Banerjee (2011, p.17) argued that ‘conventional concepts of public space and place are increasingly becoming outmoded’.

Modemist functionalism in the 1930s indicated the development of cities as focusing on ‘the physical-functional aspects of cities’ (Gehl, 2011, p.43). Outdoor activities were undermined by the logic of the modernist city, which stated that ‘the public space was to become a residue of the buildings’ (Madanipour, 2003, p.203). Barlas and Caliskan (2006) argued that modernist intervention influenced the decline of the public realm in urban fabrics, which was paralleled with the atrophy of public life. In recent years, public realms in cities have been taken over by privatised public spaces (e.g. corporate plazas and shopping malls) and become popular destinations for the public (Banerjee, 2011). According to Gehl (2011, p.47), the desire for leisure, entertainment, and social contact has been fulfilled by privatised public spaces, and those spaces have become ‘the only contact points with the outside world because life between buildings has been phased out’.

Nowadays, digital spaces construct diverse social networks and new social formations in order to interact with urban fabrics. The distinction in public life between digital and physical spaces has been blurred due to the strong connection between each. ICTs have enabled people to acquire information on events and activities through digital space, inviting them to be and act in physical spaces. By sharing common interests, the public can generate various contexts of activities through digital space, and such events take place in temporary places. Associated with the nature of network organisations, which are ephemeral, autonomous and decentral, public life in both digital and physical spaces has been varied, encompassing a wide range of activities.

Although public activities are easily organised by media-based communities, public open spaces in cities have maintained their existing physical conditions. Indeed, there is a contradiction between the emergence of the public sphere in digital spaces and the decline of public life in urban open spaces. Banerjee (2011) stated that it is unlikely that the impact of ICTs will obviate our need for social contact and outdoor life, or our demand for open spaces. Urban open space, a major type of public space, has long served as a physically accessible place, mainly for citizens’ relaxation and recreation. Gehl (2011) identified observable patterns in the activities of people in public open spaces: walking to their destination, optionally sitting, and socially interacting.

Digital space provides easily accessible platforms that collect information on citizens’ activities or interests in cities. Information pertaining to social activities in urban open spaces has been exposed in digital spaces, and those activities are observable through the collection of data. Spatial and temporal data have provided information on where and when people hold events. Batty (2013, p.274) pointed out that the increase in the collection of data, which include people’s activities, has shifted the focus of city development ‘from longer-term strategic planning to short-term thinking about how cities function and can be managed’.

Urban design strategies have dealt with placemaking for the improvement of the ‘soft city’, which focuses on the management of the activities of people, whilst the ‘hard city’ was constructed by buildings and physical spaces (Carmona et al., 2010). Picon (2015) argued that recent ideal scenarios in respect of city development have focused on controlling occurrences, events and situations, rather than physical organisations, since the networked city emerged in the industrial era for constructing systematic flows in infrastructure. Picon (2015, p.51) pointed out that ‘the advent of a new urban intelligence is leading to the transition from the networked city to the event-city’.

In fact, this is an open question for the development of cities with respect to the emergence of the new public sphere in digital space and the decline of the public realm in physical space. It is crucial to consider that the utilisation of ICTs in creating ‘smart’ or ‘intelligent’ systems in cities ultimately stands for improving the quality of citizens’ lives (Deakin and Waer, 2012). To vitalise public life in urban open space, a new design of urban open space needs to reflect the way in which citizens interact with cities today.

1.4. Research Aim

This research examines the design of an architectural machine, comprising a self-control system and kinetic structures, that is capable of adapting to the contexts of temporary events and to environmental conditions in an urban open space. Aiming at architectural intelligence, cybernetics and responsive technologies have been applied in designing active spaces which serve as socially engaged machines (Steenson, 2017; Yiannoudes, 2016). Architectural design projects, integrating embedded control systems and architectural counterparts or components, have shown self-regulating architectural systems that are interactive with people and responsive to the environment (Cantrell and Holzman, 2015; Fox and Kemp, 2009). This research investigates how such machines’ information-processing ability can be developed using urban data extractable from digital spaces, and how such machines can be designed for an architectural space to be installed in a public open area in a city.
The design of the machine is aimed at suggesting a design strategy for urban open space to revitalise public life in a city and reflect the change from place-based to media-based organisation of communal activities. The design focuses on how an architectural space in an urban open area can assist self-organised social events that are generated by members of the public in digital spaces. The development of intelligent technologies for cities is currently in the nascent stage, and the physical conditions of cities have yet to evolve through their applications (Picon, 2015). The design of the machine is aimed at dealing with the physical conditions of urban open space, which can embrace real-time information in a city by means of utilising urban data and applying data-processing techniques.

The design experiment focuses on programming a context-aware system that can produce the best decisions for controlling kinetic structures. It investigates a set of algorithms to be embedded in an architectural machine capable of being responsive to the changeable contexts of social events and to environmental conditions. The machine reads location-based and time-related information in real time and provides focal areas in which people can congregate. The machine needs the ability of finding where and when events occur in an urban open space, acquiring information regarding what the events are, and further exhibiting its reasonable behaviours through architectural elements. For the task of assisting outdoor activities, the ability of the machine to acquire and adjust the environmental conditions is required. The experiment tests and evaluates how the simulation of a machine’s behaviours can be suitable for real-time circumstances in a site.

1.5. Research Questions

1. How can the ability of information processing in an architectural machine that consists of kinetic structures be enhanced to act as an intelligent machine that is capable of not only sensing the presence of people or the changing environment in an urban open space but also identifying contextual changes under the circumstances of both factors and providing optimised spatial conditions for outdoor activities?

2. How can information processing in a set of algorithms mapping the sequence from the input of urban datasets to the output of behaviours of kinetic structures embedded in the machine be designed for its real-time decision making of selecting the best option from the alternatives of spatial qualities?

1.6. Research Methodology

In this research, the literature review identifies how intelligence has been discussed for architectural spaces, cities and machines. Architectural intelligence has not been sufficiently identified, whilst intelligent systems on an urban scale or in the field of AI have been diversely approached in developing machines’ abilities. On an urban scale, through the use of ICTs, diverse information, including the contexts of citizens’ activities, has been collected and mostly used to develop digital tools for the real-time monitoring of information. AI techniques have maximised computers’ data processing in order to perform specific tasks. This research identifies the current knowledge on intelligent systems and applies it to the design of an architectural machine.

For the design experiment, the simulation of an architectural model is tested in a public open space in Manhattan, New York City (NYC). There are various types of public open spaces: parks, squares, spaces in between buildings, and streets. The examination focuses mainly on NYC squares in which self-organised public events take place. The design experiment focuses on programming an architectural machine’s cognitive ability and testing its behaviours in an urban open space in NYC. For the design experiment, there are three considerations: the potential of utilising urban data pertaining to social contexts, the utilisation of ICTs in facilitating the machine’s real-time responsiveness, and the applications of AI techniques in implementing the machine’s data processing. The design experiment has four main processes: analysing urban data, programming an embedded control system, developing an architectural model (i.e. kinetic structures), and simulating a machine’s behaviours in real time. The following is a brief description of the design methodology.

Firstly, to understand social contexts situated in territorial conditions in urban open spaces in NYC, urban data, including geographical and social information, are examined. The distribution of public squares associated with social uses is analysed in order to observe how citizens use urban open spaces. To examine the social dimension of urban open spaces, an investigation is conducted into one of the NYC open datasets, namely NYC Permitted Event Information, which provides data on public events. NYC portals have offered information on ongoing social events which are open to the public and permitted by NYC Government. By visualising and analysing previously collected urban data, the social contexts within urban fabrics are observed. The data are analysed in order to ascertain how social activities have taken place in different open spaces.

Secondly, to programme an embedded system which controls the architectural model, a set of algorithms focus on connecting urban data to an architectural model. The previously collected
data are categorised so as to make rule sets that implement the machine’s behaviours (further
details on how the data were used to control the machine’s behaviours are provided in Chapter
3). Moreover, NYC weather data are used to provide optimised conditions of outdoor social
space. Inputting the social and weather data, the algorithm maps the sequence of urban data to
the architectural model.

Thirdly, to exhibit the behaviours of the architectural model, the design study examines kinetic
structures which are deployable. It focuses on their performances within a timeframe in
order to make adjustable spaces for different types of activities. The design examines architectural
structures that are flexible with respect to exhibiting spatial performances (e.g. open/closed,
folded/unfolded, and expanded/shrunk) in response to different types of social formations (e.g.
arrayed, clustered, divided, and dispersed). The experiment tests how architectural space can
behave or act through variations in the sequence of interchangeable spatial conditions.

Fourthly, the real-time simulation is to evaluate the space’s behaviours and, moreover,
modify the architectural model. Accessing an API (application programming interface), which is
the software documentation that facilitates real-time data streaming, the behaviours of the
architectural model are simulated in real time. Synchronising data updates and architectural
behaviours, the simulation shows the real-time responsiveness of the architectural model.
Through the simulations, problems regarding the algorithm and the limits of architectural
behaviours are evaluated and improved.

This research investigates one of the ways of designing architectural space to promote
citizens’ activities in urban open spaces. Urban data, including social contexts, are investigated
so as to reveal what is happening in those spaces. The design experiment tests how an architectural
machine can perform the given task of providing the interchangeable physical conditions of
outdoor social space by means of acquiring urban data in real time.

1.7. Research Contribution

This research contributes not only to the realm of architectural design but also to the
multidisciplinary approach to designing intelligent machines. Recently, collaboration between
architects, artists, media designers, and urban planners has led to the development of urban
machines that provide hybrid spaces that bridge digital information and physical space (Signore
and Riether, 2018). Through the convergence of multiple technologies, architectural design
projects have attempted to mediate the gap between the non-physical (e.g. information, systems,
and networks) and the physical (e.g. space, tectonics, and materials) (Signore and Riether, 2018).
The distinction between previous design projects and the design in this research is that this
design experiment focuses not only on converging the immateriality of information and the
materiality of physical forms but also on developing an embedded system that enables an
architectural machine to perform its task of assisting social activities by adjusting spatial qualities.

The issue surrounding developing intelligent technologies is addressed in discussing how
intelligent technologies can be useful for the quality of our lives. For the development of cities,
the application of intelligent technologies has rarely focused on improving physical conditions
and the societal liveability of cultural and historical cities (i.e. existing cities), while they are
promoted to be inclusive in digital infrastructure, mainly for new urban towns (Allam and
Dhuny, 2019). With regard to the emergence of digital space and the decline of urban open
space, the design experiment examines how an architectural machine can be designed to provide
a new social space in an existing city.

Aiming towards designing an intelligent machine for architectural space, this research
attempts to integrate applicable intelligent technologies in designing an architectural machine’s
context-aware system and real-time responsiveness in relation to physical conditions. Through
the use of ICTs, happenings or occurrences in urban fabrics have been observable. Social contexts
in cities are no longer hidden, but rather exposed through digitally collected data. Referring to
the applications of current intelligent technologies and ICTs, this research investigates how a
machine can sense social happenings in urban open space by means of urban data, perceive the
changing contexts of these happenings through an embedded intelligent system, and further
manage the updated information generated in digital space and the interchangeable conditions of
physical space in real time.

This research examines a way of designing an intelligent machine for architectural space.
Intelligence is not to be considered solely a human property, but rather has also been applied in
designing machines’ abilities in order to assist society. Architectural spaces are situated between
people and the environment. The environment encompasses both physical and contextual
conditions. By assigning a task to an architectural machine to assist citizens’ activities (i.e. social,
cultural and temporary events in a city), this research examines how it can perform architectural
behaviours (i.e. actions of kinetic structures) suitable for them.

In this research, it was considered that there are two main users: the designers (who continue
to develop architectural machines) and the end users (who can utilise the machines). By using an
algorithm-aided design tool, namely Grasshopper in Rhino 6, the design experiment focused on
forming the algorithm to not only connect the input of data and the output of architectural
behaviours but also involve logic-based sequences. It integrated currently available computational techniques (e.g. algorithms for data visualisation and kinetic structures) and further established rulesets between the input and the output. By employing the easily accessible and widely used software, designers are able to apply, modify and extend the algorithm or the ruleset established in this research so as to test architectural machines’ control systems and reasonable behaviours. In fact, the development of such machines is oriented towards focusing on what they can do or on how people or end users can habit with them. By identifying how an open space in a city was used by the public through extracting digitally collected data, this research investigated the ability of information processing in an architectural machine for their usage of the physical space.

1.8. Thesis Structure

Chapter 1. Introduction

Chapter 1 presents the introduction to the research topics, questions, domains and methodology. This research investigates a method of designing an architectural machine that can process information on real-time conditions in an urban open space. An architectural machine’s context awareness was examined, as it was considered to be one of the approaches to the design of an intelligent machine.

Chapter 2. Literature Review

Chapter 2 classifies how intelligent systems have been developed within three domains: architectural intelligence, urban intelligence and artificial intelligence. It identifies cohesive interpretations, thoughts and opinions on the term ‘intelligent’ which have led the development of computational techniques to realise intelligent systems within the domains. To design an intelligent machine for architectural space, the three domains were examined, with consideration given to their applications (see Figure 1.1).
Chapter 3. Design Methodology
Chapter 3 discusses the research methodology and presents technical descriptions showing the main simplified components in the algorithm. The algorithm consists of the input of text-based data extracted from open data sources, the data processing of sorting, matching and categorising, and the output of an architectural model’s behaviours.

Chapter 4. Design Experiment
Chapter 4 describes the experiments and design outcomes that were developed using the algorithm. Data in Bryant Park were collected, visualised and analysed to be used as the input in the algorithm. This was done so as to gain information on events in the site, which was shared via digital space. Rulesets in the algorithm were established for logic-based sequences between the input and the output.

Chapter 5. Design Development
Chapter 5 demonstrates the advanced algorithm that was applied for the central lawn area in Bryant Park. A problem-solving technique based on evolutionary computation was applied in order to find an optimal form for acoustic and sun/shadow factors. Based on the finally selected form, decision-making processes in the algorithm were updated and the wider range of an architectural model’s behaviours was programmed. The algorithm was developed to provide functionally suitable and formally varied spatial qualities.

Chapter 6. Conclusion
Chapter 6 concludes the thesis by reviewing the reflections, identifying the limitations and suggesting future work. The design method in this research was examined for the development of architectural machines that can assist the activities of people. It stresses the potential of using data to enhance architectural machines’ abilities to process information on people and their environment.
Chapter 2. Literature Review:
Intelligent Embedded Systems in
Architectural Spaces, Cities and Machines
2.1. Introduction

The term ‘intelligent’ or ‘smart’ has been popularly expressed in developing a wide range of advanced information technologies. Although the meaning of intelligence encompasses diverse definitions, aspects and interpretations, the term ‘intelligent’ has been used to establish the goal of improving machines’ cognitive functions. Architectural intelligence, urban intelligence and artificial intelligence have been approached in the development of machines, tools and systems that are capable of acquiring particular information and performing given tasks.

In the field of architectural design, one of the remarkable approaches to architectural intelligence constitutes starting with the design experiments inspired by cybermatics. Opposed to the view of architectural spaces as static entities, the 1960s’ architectural avant-garde attempted architectural experiments that emphasised physically active spaces. Cedric Price and Nicholas Negroponte tested whether architectural spaces could configure and reconstruct spatial conditions in response to occupants’ changing conditions by using computers, programming interfaces and controlling physical spaces (e.g. cubical volumes in spaces). They investigated how technology-enhanced architectural spaces could be suggested by the application of cybermatics.

Architectural design projects have developed self-regulating systems, socially interactive spaces and physically controllable spatial conditions that focus on the aspect of aesthetic values for new types of spatial qualities. Dealing with the subject of architectural interactivity, active spaces have been further practised by assembling architectural elements (e.g. kinetic structure) and responsive technologies (e.g. sensor/actuator). Projects on responsive landscapes have provided architectural spaces that are capable of sensing and detecting surrounding conditions and responding to real-time changes through lighting, sound, media screens, and naturally responsive materials.

Meanwhile, the term ‘intelligent’ has been used more widely in establishing a technological goal of achieving the automated management of electronic and mechanical systems (e.g. lighting, heating, ventilating, and air conditioning). Technological initiatives on Intelligent Buildings (IBs) have developed strategies for buildings’ energy-saving and users’ personalised systems. Projects on Intelligent Environments (IEs) have focused on developing computer interfaces to collect data on house residents’ activities and facilitate context-aware systems. Ambient Intelligence (AmI) projects have built remotely controllable indoor conditions through integrating sensors, electronic devices, and a central computer. Those technological projects have facilitated the efficient control of electronic and mechanical facilities in buildings.

On city scales, the development of machines has been oriented towards designing digital tools that assist citizens’ use of cities. Prior to the advent of digital technologies, thematic mapmaking presented descriptions of empirical observations which were collected by marking the locations of collected data. Since geolocation (i.e. the identification of geographical locations) has enabled digital tools to collect location-based data more easily, urban data have been exponentially collected and gradually useful in observing the status of cities. Urban data visualisations have revealed implicit values in cities, whereby producing spatial knowledge.

Urban data visualisation researches at academic institutions have conducted mapmaking so as to understand cities. Kevin Lynch’s cognitive map presented communicative symbols and notations for depicting how citizens perceive cities. Meanwhile, Richard Saul Wurman’s works on urban data visualisations established effective methods of organising urban information in order to help people’s understanding of cities. Recent urban data visualisations using digital tools have attempted to find the relationship between digital and physical spaces and the social contexts in cities. Moreover, through visualising big data, which are being increasingly collected through people’s usage of electronic devices, citizens’ interactions with cities have been observed.

Since the 1990s, with the advent of ubiquitous networks in ICTs (e.g. the Internet), recent urban developments have focused on real-time monitoring information on cities and digitally manageable city systems (e.g. public transportation and online governmental services). The term ‘smart city’ has been used to promote applications of ICTs. Urban data have increasingly involved diverse contexts that present evidence of how citizens have used cities. Digitally constructed systems have enabled machines to observe where and when citizens use cities in both the long term and the short term. Urban sensing technologies have enabled machines to perceive happenings in cities.

In order to enhance machines’ cognitive capabilities, data-processing algorithms have been investigated intensively in computer science. Since Alan Turing suggested that a machine’s intelligence can be evaluated by its behaviours, machines’ various abilities have been tested. In 1956, the Dartmouth Workshop was organised to initiate sharing independent studies aiming to design thinking machines, and the term ‘Artificial Intelligence’ (AI) was declared as a research discipline. Classic AI systems were commonly examined to be analogous to humans’ logic-based thinking. Meanwhile, bio-inspired AI systems extended AI studies to include a wide range of living beings’ information processing.

Mostly, AI systems solve a narrow or limited range of problems. Rule-based systems have been used to handle expert-level knowledge in an effective manner. Computational models based
on Artificial Neural Networks (ANNs) have enhanced computers’ abilities of image recognition and data classification. Evolutionary Computation (EC) techniques have been used to make problem-solving tools that generate scenarios or probabilistic solutions. Information processing algorithms have facilitated computers’ cognitive abilities and enabled machines to acquire given information to assist users’ decision makings.

Whilst the technological aim of designing intelligent systems has led the development of software, information processing techniques, and digital tools, architectural design projects have produced machines that exhibit the abilities of sensing and behaving, focusing on the physical qualities of architectural spaces. In the field of architectural design, projects that deal with the theme of architectural interactivity and responsiveness have aimed to achieve architectural intelligence. Intelligent technologies are highly related to information processing and mostly developed to manage urban information and to improve computational techniques. Urban data have been increasingly collected and more available to observe the interactions between people and physical spaces in cities, and information processing techniques have been developed to use data easily. For the design of an architectural machine, there is the potential of using urban data to gain information on the activities of people and applying information processing techniques to implement its reasonable physical actions based on the information.

This chapter reviews how different types of intelligent systems have been developed for architectural spaces, cities and machines. The following literature review examines the main arguments, limitations, techniques, and further indications with regard to developing intelligent systems in the three domains. This chapter identifies how architectural intelligence, urban intelligence and artificial intelligence were initially approached and further developed. To understand what they are, it reviews seminal figures and key factors that have led the development of intelligent systems in the three domains. To carry out the design experiment in this research, this chapter seeks the possible links for the means of designing an intelligent machine for architectural space.

2.2. Intelligent Machine in Architectural Space

2.2.1. Cybernetic Approach to Architectural Intelligence

Cybernetics

In the postwar era, with the invention of computing machines, various discourses and ideas were addressed to develop digital technology, such as cybernetics, information theory, general systems theory, and artificial intelligence (Gere, 2008, p.79). In the middle of the 20th century, interdisciplinary approaches to cybernetics began as a science of the brain in order to examine brain-like artefacts (Pickering, 2009). Between 1946 and 1953, the Macy Cybernetics Conferences entitled ‘On the Workings of the Human Mind’ promoted the exchange of different scientific ideas in disciplines including physics, mathematics, electrical engineering, physiology, and neurology. Whereas researchers in artificial intelligence thought of the brain as a ‘representational and cognitive device’ or a ‘thinking machine’, cyberneticians thought of the brain as an ‘acting machine’ that could process performative inputs and outputs (Pickering, 2009, pp.472–473).

In 1948, the term ‘cybernetics’ was coined by Norbert Wiener in his book entitled Cybernetics or Control and Communication in the Animal and the Machine. He explained the term to establish a new science: ‘We have decided to call the entire field of control and communication theory, whether in the machine or in the animal, by the name Cybernetics, which we form from the Greek kubernetes or steersman… governor is derived from a Latin corruption of kubernetes’ (Wiener, 1948, pp.11–12). One of the earliest forms of a ‘control mechanism’ was described in the Clerk Maxwell article entitled ‘On Governors’ published in 1868, which demonstrated the mathematical logic of the steam engine as a device for affecting speed control developed by feedback mechanisms (Wiener, 1948; Ashby, 1960; Pask, 1961). Pask (1961, p.12) remarks: ‘A great deal of cybernetics is concerned with how stability is maintained with ‘control mechanisms’.

In 1948, the term ‘cybernetics’ was coined by Norbert Wiener in his book entitled Cybernetics or Control and Communication in the Animal and the Machine. He explained the term to establish a new science: ‘We have decided to call the entire field of control and communication theory, whether in the machine or in the animal, by the name Cybernetics, which we form from the Greek kubernetes or steersman… governor is derived from a Latin corruption of kubernetes’ (Wiener, 1948, pp.11–12). One of the earliest forms of a ‘control mechanism’ was described in the Clerk Maxwell article entitled ‘On Governors’ published in 1868, which demonstrated the mathematical logic of the steam engine as a device for affecting speed control developed by feedback mechanisms (Wiener, 1948; Ashby, 1960; Pask, 1961). Pask (1961, p.12) remarks: ‘A great deal of cybernetics is concerned with how stability is maintained with ‘control mechanisms’.

In order to examine the systems of control or communication, cyberneticians explored living systems underlying the control processes regulating themselves. Ross Ashby’s notion of living systems within their environment is that adaptive behaviours keep the essential variables within limits in order to survive by means of receiving feedback from variables of change in the environment, and those behaviours affect the environment to maintain a dynamic equilibrium (Ashby, 1960). Ashby uses the term ‘homeostasis’ to describe adaptive systems that maintain...
stability, a steady state and an equilibrium through the interaction between parts and the whole (Ashby, 1960). Inspired by Ashby’s notion of adaptive systems, Stafford Beer discusses the idea of the ‘automatic factory’, which can be operated by individual machines which control themselves and produce materials through automated operational routes without human involvement (Pickering, 2010). Gordon Pask argues that a goal-driven machine should allow the interaction between a human (manager, who trains) and non-human (machine or controller, which learns or answers) because such interaction enables a machine or a controller to produce the same patterns of behaviour while seeking a compromise (Pickering, 2009).

Since scientists collaborated in exploring the intersecting areas of knowledge of ‘systems thinking’, ‘this turn away from the mainstream of science became a turn toward interdisciplinarity’ (Dubberly and Pangaro, 2015, p.130). Their discourses surrounding systems development were not delimited within scientific fields but rather extended to the discussion of management and the realm of art and architecture. Beer addressed the issue of cybernetic approaches to management with the analogy of biological organisms. Meanwhile, Pask engaged in art and architectural experiments focusing on the concepts of performance, interaction, and communication (Pickering, 2010). Dubberly and Pangaro (2015, p.130) remark: ‘A trending topic in the 1960s, cybernetics peaked about 1970 and crashed—its ideas absorbed into many fields, the origins of those ideas largely forgotten or ignored…Nevertheless, other effects of cybernetics live on…about how we interact with computers and how we design for interaction.’

The Architectural Vision of Cybernetic Application

In the 1960s, while cyberneticians explored computational applications through the collaboration of researchers in interrelated scientific fields, architects started to examine the possibility of interactive and responsive architecture referring to their discourses. The architectural avant-garde at that time attempted to design architectural machines based on a cybernetic approach. Their vision of new architecture was to create self-regulating systems which facilitated constant processing of input data on users and output changes of physical spaces. Yiannoudes (2016, p.118) distinguished two groups of the 1960s’ architectural avant-garde who employed a cybernetic approach to architecture: one group (Dutch Structuralists, Japanese Metabolists, and British Archigram) used cybernetics as an indirect and metaphorical concept; the other group (Cedric Price, Nicholas Negroponte, and Yona Friedman) approached it literally and pragmatically, and proposed an architecture that viewed spaces as cybernetic machines.

Given that the 1960s’ architectural thinking engaged with the issues of ‘flexibility, impermanence, prefabrication, computers, robotics’, it was inevitable that their architectural disciplines would embrace cybernetics (Frazer, 2001, p.642). Pask (1969, p.69) points out: ‘The argument rests upon the idea that architects are first and foremost system designers who have been forced, over the last 100 years or so, to take an increasing interest in the organisational (i.e. nontangible) system properties of development, communication and control.’ Moreover, Pask notices that ‘a building cannot be viewed simply in isolation. It is only meaningful as a human environment. It perpetually interacts with its inhabitants, on the one hand serving them and on the other hand controlling their behaviour’ (Pask, 1969, p.70).

The architectural experiments, inspired by cybernetics, extended the limit of architectural thinking by asking the question of whether an architectural space could be implemented as a system to interact and communicate with users more closely. The cybernetic application to architecture was to rethink the boundary of architectural possibilities in order to deal with the advancement of technologies. The cybernetic approach to architecture was the initial attempt at creating control systems and acting machines capable of changing spaces through interaction and communication between the occupant and enclosure.

Cedric Price

Acknowledging the potential of information technology for innovative architecture, Cedric Price, one of the pioneers who investigated new architecture, brought the applications of not only cybernetics but also intelligence to the architectural discipline (Steenson, 2017). Since Price first met Pask in the 1950s, Pask contributed to consulting on Price’s architectural projects that reflected system-oriented thinking involving the ideas of control systems as well as interaction and communication between humans and non-humans (Steenson, 2017). ‘In essence Pask believed that intelligence lies in interaction, not inside a head or a computer’ (Obrist et al., 2003, p.69).

Frazer (2001, p.643) states that ‘cybernetics in architecture is advanced as a new theoretical basis and as a metalanguage for critical discussion. Additionally, cybernetics is advanced for its predictive power…Explanatory power is also claimed for the ability to mimic certain aspects of architectural design by “artificial intelligence computer programs.”’

Collaborating with theatre producer Joan Littlewood and consulted by Gordon Pask, Price proposed the ‘Fun Palace’ project (1961–1974) in order to account for the idea of improvisational architecture and transient space adapting to diverse events. Cedric Price stressed spatial adaptability, flexibility, and temporality instead of pursuing architectural permanence or
monumentality (Mathews, 2005). According to Mathews (2005, p.73), ‘the Fun Palace was…a socially interactive machine, highly adaptable to the shifting cultural and social conditions of its time and place’. Price noticed that the nature of social interaction was indeterminacy. He proposed that architectural designs suitable for social events needed to incorporate systems of spontaneous communication between users and spaces.

The system embedded in the Fun Palace’s spatial configurations was not to constrain users’ choices of movements or activities but to be controlled by them. Pask established the logical flow of the feedback system to programme an interactive machine capable of recognising social patterns and trends of individual users’ changeful preferences through processing collected data and adjusting the parameters of spatial configurations (Mathews, 2005). Based on such a plan of a software program, the constantly changeable hardware of structures was organised by intermediate beams, 270-degree rotatable escalators, and travelling cranes (Price, 1979). As Price (1979, p.327) demonstrated: ‘The reason for the employment of such techniques was to ease up the choice, to free the opportunities of the individual user as to what to do next.’ The ground floor was left clear and open to users’ controls.

Furthermore, developed from the cybernetic approach, Cedric Price and architects/computer consultants John and Julia Frazer collaborated in order to design the Generator project (1976–1979). The project showed how architecture that was connected to central computer devices could be capable of improving spaces through its own configuration of self-generated decisions. Price’s scheme of the project was to suggest interchangeable cubic spaces that could be differently arranged on a grid through the change of users’ activities in order to encourage their use of the spaces. Frazer suggested the Boredom Program, which ran the logic circuit with microprocessors that were connected to each component of the cube so as to implement the performance of spatial configurations. By measuring users’ overuse or underuse of the spaces, the Boredom Program made design decisions in respect of the spatial organisation. Although there is not a single, clear representation that shows how the Generator operates its self-decision processes, the project was intended to challenge architectural intelligence (Steenson, 2017).

The Generator project was one of the first attempts to refer to the term ‘intelligence’ in architecture. Price claimed that ‘his Generator project was the world’s first intelligent building, and perhaps it was – particular because it actually showed how artificial intelligence could work in an architectural setting’ (Steenson, 2017, p.128). Frazer states that ‘the computer program is not merely a passive computer-aided design program nor is it just being used to assist with the organization of the site, but is being used to actively encourage continual change and adaptation.
to changing requirements... In a sense the building can be described as being literally intelligent' (Frazer, 1979, as cited Steenson, 2017, pp.158–159). In broad terms of intelligence, the Generator project was ‘one of the first intelligent buildings’ (Sudjic, 1981, as cited in Yiannoudes, 2016, p.56). Steenson (2017, p.162) observes: ‘Generator marked a point of emergence of important factors for responsive architecture: embedded, distributed, electronic intelligence; active computer-aided design tools; the correspondence of the model to the design tool; and questions of machine intelligence.’

Nicholas Negroponte

Integrating architecture with artificial intelligence, computer science, and electronic engineering, Nicholas Negroponte and the MIT Architecture Machine Group investigated an ‘architecture machine’ that focused on the development of the human-computer interface. Frequently collaborating with the MIT Artificial Intelligence Lab, Negroponte and the Architecture Machine Group (which operated from 1967 to 1985) started to employ ‘technological interfaces – screens, tablets, touch screens, video eyes, camera rooms’ (Steenson, 2017, p. 167). Negroponte stated, ‘A design machine must have an artificial intelligence because any design procedure, set of rules, or truisms is tenous, if not subversive, when used out of context or regardless of context. It follows that a mechanism must recognize and understand the context before carrying out an operation’ (Negroponte, 1970, p. 1). Following his idea, the group first experimented with computer-aided design systems URBAN 2 and URBAN 5, which displayed the dialogue between a user and computer to help architects with the design process.

Negroponte and the group also attempted the convergence of the digital interface and space through the ‘SEEK’ project. This experiment showed how an architectural space could be controllable and therefore capable of providing different spaces through feedback systems in a computer. The cubic environments were controlled by robotic arms and the occupants’ movements were tested using small animals, i.e. gerbils. The control system of the robotic arms was run by the six commands ‘Generate, Degenerate, Fix It, Straighten, Find, and Error Detect’ (Steenson, 2017, p. 185). Its feedback system of configuration and reconfiguration, akin to artificial intelligence, tried to handle unexpected happenings in the physical environment. However, due to the unpredictable nature of the occupants’ movements, its handling mechanism could not promptly respond to the occupants’ movements or provide optimal spatial conditions for the gerbils’ movements (Steenson, 2017). Nevertheless, this project attempted to use the techniques of an operating system that controlled physical components. It suggested the architectural possibility that artificial intelligence could be applied to handle real-time happenings in the real world.

In his books entitled The Architecture Machine (1970) and Soft Architecture Machines (1975), Negroponte made decisive statements on intelligence for machines. Negroponte (1970, p. 1) presented his opinion on intelligence as follows: ‘Intelligence is a behavior. It implies the capacity to add to, delete from, and use stored information. What makes this behavior unique and particularly difficult to emulate in machines is its extreme dependence on context: time, locality, culture, mood, and so forth.’ Negroponte (1975, p. 7) also stressed that ‘intelligence is
a property that is ascribed by an external observer to a conversation between participants if, and only if, their dialogue manifests understanding.' Negroponte (1975) argued that the consequence of the interaction between man and machine is enhancement of both performances. He stressed that the ability to understand context is associated with intelligent behaviours for both man and machine. Negroponte (1970, p. 33) remarked that 'context must be recognized by us in terms of our own behaviors or by a machine in terms of its behavior' and that 'the machine must be constructed in such a manner that its behavior gives us enough confidence to presume that it is acting intelligently and with common sense, that is, in context.'

His notion of the relationship between man and machine has led not only to the development of interfaces for computer-aided design but also to intelligent environments and technologies. Negroponte states ‘my view of the distant future of architecture machines: they won’t help us design; instead, we will live in them’ (Negroponte, 1975, p.5). He remarked that an intelligent environment needs to have the ability to identify an individual user to achieve responsiveness. Since the Architecture Machine Group became the Media Lab, which was established in 1985, its technical experiments have moved beyond architecture (Brand, 1987). The Media Lab has conducted technical research on wearable computing, tangible interfaces, and electronic communication technologies.

Negroponte’s aim of man-machine symbiosis has turned to ‘ultra-personalized intimate technology’ (Brand, 1987, p. 262). In his book entitled *Being Digital* (1995), Negroponte pointed out that the increasing volume of ‘bits’ (i.e. digitalised information) has impacted on our ways of using ‘atoms’ (i.e. tangible objects such as newspapers, magazines and books). Furthermore, he stated that ‘Everything is made to order, and information is extremely personalized’ (p. 164). He conceived that the way of transmitting information would be transformed from mass media oriented towards mass audiences into personalised media directed towards one individual. He noted that intelligence would serve as the transmitter and the receiver, selecting, filtering and delivering information for an individual’s personal interests and that the result of such a movement would lead to four qualities of the digital world: ‘decentralising, globalising, harmonising, and empowering’ (p. 229).
2.2.2. Architectural Interactivity and Responsiveness (State of the Art)

Interactive Architecture/Responsive Landscape

Utilising sensor and actuator technologies, current architectural projects have attempted to realise programmable, interactive, and responsive architectural spaces. Regarding current technologies for the architectural discipline, Bouman (2005, p.16) raised a question: ‘How can architecture adopt a technology which is in itself time-based? not simply to the design of space, but to the experience of space’. The utilisation of responsive technologies in architecture and landscape have facilitated sensing and responding to real-time conditions in the real world, ‘built upon the convergence of embedded computation (intelligence) and physical counterparts (kinetics)’ (Fox and Kemp, 2009, p.12). The realm of architectural practice has been extended to deal with the changeable conditions of physical space through responsive lights and sounds, controllable façades, kinetic structures, and media screens.

Projects on programmable architecture have utilised sensor technologies to provide different spatial conditions for the presence of people. The FreshH2O Expo project by NOX (exhibited at Neeltje Jans in the Netherlands from 1994 to 1997) presented an interactive space that constantly modified the interior qualities (e.g. lights, projections, and sounds) by sensing the presence of visitors. One of the aims of this interactive architecture was to serve as a medium mediating the immateriality of user information and the materiality in physical conditions of architectural elements (Spuybroek and Oosterhuis, 1998). Moreover, the NSA Muscle project by ONL (exhibited at Centre Pompidou in 2003) suggested a type of interactive architecture that consisted of inflatable structures that could change its shapes in response to the proximity of people detected by touch sensors (Oosterhuis, 2005).

Based on responsive technologies, interactive facades have involved applications of kinetics. Moloney (2006) noted that architectural façades that applied kinetic systems have two aspects: the aesthetic performance for socio-cultural engagement and the functional performance for environmental effectiveness. The dECOi’s Aegis Hypo-Surface project (the early prototype was unveiled at the Cebit Trade Fair in 2001) showed how a tessellated wall could physically move in a wave-like manner that generated the complex features of movements or the changes of metallic panels. Its dynamic expression of form responded to the motions of people by means of sensor devices, and it engendered the sense of an interactive environment (Goulthorpe et al., 2001). Moreover, the façade system of the Arab Institute by Jean Nouvel (located in Paris; built in 1988) suggested a type of environmental control systems. The façade control system was
capable of regulating the sunlight penetration by controlling the performance of the opening and closing apertures of metallic panels. Moloney (2011) pointed out most of the kinetic systems in building façades, driven by an environmental performance agenda, were typically designed to be responsive to a sun-tracking motion, and consequently, such an approach was socially unengaged.

There are three main components that implement the control system in interactive architecture: a sensor to detect real-time stimuli (e.g. users’ movements and environmental conditions) (input), a computational processor that transmits data to architectural elements (processing system), and an actuator that processes the received signals and presents the results (output). Moloney (2011) argued that to build an intelligent façade system, two more properties are required for the control system in addition to the three abovementioned systems (i.e. input, processing, output): a response occurring with a consideration of time and a learning ability based on previous data. According to Moloney (2011, p.28), ‘a survey of intelligent façades resulted in the declaration that a truly responsive, adaptive and controllable intelligent facade has yet to be found’.

For public engagements, current responsive landscape projects have provided participatory environments. According to Cantrell and Holzman (2015, p.20): ‘The term “responsive” denotes that an object engages in a process of feedback, a conversation between two actors. In architecture and design, this has typically been approached from a Human-Computer Interface (HCI) perspective’. The Dune 4.2 project by Studio Roosegaarde (located along River Maas in Rotterdam; built in 2010) provided a public landscape that people could experience with the interactive artwork by visiting the place. The field of branches, equipped with presence sensors and microphones, responded to not only the movements of visitors but also the sounds of surrounding environmental conditions, such as the wind sounds. The responsive technologies rendered differentiated landscape conditions by illuminating the LED lights and emitting the amplified sounds. Likewise, the Aviary project by Howeler + Yoon and Parallel Development (exhibited in Dubai; built in 2013) suggested an audio-visual landscape that consisted of interactive tubes. The tubes responded to visitors’ touches and accordingly provided the sounds of birds’ tweets and lighting effects. Such projects showed how sensing technologies detecting the presence of participants could perform the on/off activations of the electronic devices of lights and sounds. The projects encouraged the engagement of people to interact with the architectural components and the enjoyment of being present within the place.
Furthermore, responsive technologies have been utilised to transmit information through architectural elements. The Living Light project by The Living (located in Seoul; built in 2009) presented the pavilion that displayed the environmental information of air quality in Seoul. The LED lights, installed in the panels of the canopy, represented the distribution of the best and the worst air quality in the city districts. Rapidly blinking lights expressed visitors’ interests when they sent the text messages to the Living Light hotline. The Datagrove project by Future Cities Lab (located in California; built in 2012) provided the whispering installation that displayed social media information—Twitter feed. The installation, composed of LEDs, LCD displays, and speakers, exposed the hidden steam data in the social media to nearby visitors. These projects captured the flow of information, by extracting selected data via the Internet, to be exposed and exhibited on tangible and physical elements (e.g. panels, lights, and screens). Such responsive landscapes, located in public accessible places, provided a type of medium not only to convey invisible information to the places so as to arouse visitors’ interests or certain issues (e.g. environmental information and social media information) but also to stress the current blur boundary of the public realm in digital and physical spaces by building the responsive architectural elements accessing both the spaces.

As sensor technologies have enabled architecture to be responsive to the presence of people, various ways of designing interactive architectural elements have been realised for real-world installations. Bullivant (2006, p.10) stated, ‘Evolving effective responsive systems, and creating a credible interface between the work and the user, requires an awareness of many different types of user, contexts, and function, as well as the phenomenological aspects of social and environmental conditions.’ Fox and Kemp (2009, p.12) remarked that ‘If architecture is to continue to respond to the technological innovation that surrounds it as a profession, then we may no longer ask “What is that building?” or “How was it made?” but rather, “What does that building do?”’. Utilising responsive technologies to enhance people’s experience is a challenge for advancing such architecture (Booman, 2005).

Towards Living Systems

Learning from the processes of nature, architects have adopted biological principles in designing architectural projects. Wellesley-Miller (1975, p.126) stated: ‘It would seem that after a century’s preoccupation with the physiology of buildings we are beginning to become involved with their metabolism and are even starting to develop rudimentary nervous systems for them complete with sensors and actuators.’ Fox and Kemp (2009, p.20) noted an interactive architectural movement...
from a mechanical paradigm to a biological paradigm. ‘A biological paradigm of interactive architecture requires not just pragmatic and performance-based technological understandings, but awareness of aesthetic, conceptual, and philosophical issues relating to humans and the global environment.’ According to Brownell and Swackhamer (2015, pp.24–25), there are three types of architectural projects based on the set of design methods relating to biology: behavioural (a direct result of a biological behaviour), genetic (an outcome of a genetically informed system or growth pattern), and epigenetic (a reactive system for real-time changes by external factors in its environment). Epigenetic projects have focused on developing the ability of sensing and behaving in architecture.

Through applying some of the characteristics of living beings’ metabolic processes to such installation, the Hylozoic Ground project by Philip Beesley (exhibited at the Venice Architecture Biennale in 2010) presented a socially interactive work of both art and architecture. The motions of plant-like modular structures could be responsive to visitors’ movements through the interconnected components of sensing and behaving systems. Such motions of components were based on the cyclical processes of pulling air through filtering membranes (breathing), reacting through distributed sensors (caressing), circulating liquid exchanges that affect matrix formations (swallowing), and twitching whiskers. To some extent, the project proposed a type of adaptive system exhibiting self-assembling through integration between the forms that involved decentralised sensors and actuators and the system that animated responsive processes (Beesley and Armstrong, 2011).

In contrast to responsive projects using sensors and actuators, a new mode of climate-responsive architecture, based on the material’s capacity to autonomously change its figures in response to different environmental conditions, was presented through the HygroSkin project by Achim Menges (located in Stuttgart, built in 2013) and the Bloom project by DOSU Studio (located in California, built in 2012). The HygroSkin pavilion showed that portions of the apertures could open and close without any external energy or technical supply, using bendable plywood panels autonomously responsive to different humidity levels. The pavilion utilised a natural material: wood, which involves its own capacity of being responsive to environmental changes. As climate-responsiveness is naturally embedded in nature, the project provided a materially and naturally responsive system in respect of the changes of environmental conditions (Krieg et al., 2017). Similarly, the skin of the Bloom Pavilion used bimetallic strips which were expandable in relation to the level of sun exposure and thermostats. Although such material was nonbiological, the phenomenon of its behaviour resembles circadian response (Brownell and Swackhamer, 2015).

In order to actively and promptly respond to the changes of environmental conditions as well as the presence of people, researches investigating architectural interactivity, responsiveness and adaptivity have focused on sensing and behaving abilities in architectural components through technologies and materials. Such abilities were facilitated by programming embedded systems capable of controlling spatial conditions or architectural elements. Technological researches on such systems have developed computing interfaces to construct interconnected environments mediating between digital and physical spaces. Meanwhile, adopting sensor and actuator technologies, projects of responsive architecture have provided active spatial qualities and extended the possibility of the architectural realm. The advancement of digital technology has enabled architectural spaces to interact with people and the environment.

The development of embedded control systems within the spaces situated between people and their environment needs to intertwine with both internal matters and external changes. According to Weinstock (2005):

The interactivity of the internal phenomenal character to the external environment can be orchestrated in complex patterns that are active, and may enable a new agenda of environmental adaptability aligned to another spatial flexibility, providing spaces that get taller, lighter and more open in the summer, cosier and warmer, or smaller in the winter (p.50).

Architectural design projects dealing with interactive and responsive spaces have engaged with the subjects of social interaction and environmental responsiveness. The challenge of intelligent machines embedded in spaces lies in setting the systems performing cognitive processes such as learning, understanding, and decision making in the flow of the input (i.e. sensing) and the output (i.e. behaving), not only to be reactive but also to be contextually responsive to both terms as to provide optimised conditions of architectural spaces.
2.2.3. Technological Approach to Architectural Intelligence

**Intelligent Environments / Ambient Intelligence**

With the advancement of information technology, computer scientists and software engineers have investigated intelligent systems embedded in spaces capable of recognising occupants' activities and improving their spatial conditions by means of data-processing software and sensor/actuator technologies. They have conducted researches on computationally augmented spaces such as intelligent environments (IEs) and ambient intelligence (AmI) (Yiannoudes, 2016). Mark Weiser coined the term 'ubiquitous computing' to provide the vision of information technology, and initially claimed the technological goal of achieving more connected and more integrated systems, to be embedded not only in electronic devices but also in objects and spaces (Weiser, 1991). Researches on pervasive and ubiquitous computing have focused on the provision of context-aware services for the end user’s everyday life and the interface for human–computer interaction through networking devices (Augusto et al., 2013).

IEs have been intended to achieve technologically enhanced systems for the automation of electronic and mechanical control of lighting/shading, temperature and ventilation. For instance, the intelligent dormitory (iDorm; examined in the Department of Computer Science at the University of Essex) suggested a type of intelligent system capable of adjusting room conditions (e.g. lighting, window opening, and temperature) to maintain an optimal condition in order to support a user’s activities, based on the records of a resident’s habitual behaviour (Callaghan, 2004). Likewise, to develop sensing technologies capable of detecting residents’ common domestic activities, MIT’s PlaceLab and the Georgia Institute of Technology’s AwareHome examined context-detecting algorithms embedded in sensor-driven infrastructure, programmed to implement the processes of tracking the locations and times of residents’ activities, learning patterns of behaviour and measuring common events. Through the collection of users’ data, such researches developed context-aware sensing technologies so as to realise customisable spatial conditions of a house (Kidd et al., 1999; Intille et al., 2006).

Moreover, AmI presents the vision of computing technology that facilitates electronic environments interacting with the presence of people. Researches on AmI have emphasised human-computer interaction technologies and investigated tangible interfaces being operated by the presence of people (ranging from individuals to groups). One such AmI project, MIT’s ambientROOM, showed how a user interface between an occupant and physical objects in space could sense information on users’ behaviours and provide various spatial conditions through visual, auditory and multisensory displays (Ishii et al., 1998).

According to Augusto et al. (2013), IEs embrace AmI with smart environments (i.e. environments with sensing devices capable of collecting and processing data locally) based in pervasive and ubiquitous computing (i.e. distributed systems that are situated in any time and everywhere). Researches on IEs and AmI have mainly investigated middleware (i.e. software applications implementing input and output) capable of exchanging data between a user and a place through a fixed set of sensor/actuator systems; tools and algorithms have allowed deployment in the real world in order to simulate the interaction between digital and physical environments (Roolter et al., 2011).

**Intelligent Buildings**

‘Intelligent buildings’ (IBs) has been variously defined in order to provide a conceptual framework of future buildings (Ghaffarianhoseini et al., 2015). The US-based Intelligent Building Institute (IBI) and the European Intelligent Building Group (EIBG) summarised the most acceptable key features of IBs: ‘one which provides a productive and cost-effective environment through optimization of its four basic elements including structures, systems, services and management and the interrelationships between them’ (Omar, 2018, p.2905). The term was initially described in the United States in the 1980s, and stressed the characteristics of automated technology and cost-effective building management at that time; its definitions have since included other features (Ghaffarianhoseini et al., 2015).

Descriptions of IBs’ key features and components have embraced not only the economic perspective and the technical dimension of indoor conditions of buildings but also the environmental and sociocultural aspects of multifunctional buildings (Ghaffarianhoseini et al., 2015). The development of technologies of IBs started with strategies in respect of enhancing the control systems of single apparatus in a building such as HVAC, electricity, water and access. Since 2000, such systems in IBs have been integrated and remotely monitored through communication network systems (Ghaffarianhoseini et al., 2015). Although there is yet to be a standard definition of IBs, but rather multifaceted, recent objectives of IBs have targeted providing ecologically sustainable buildings with energy-saving strategies as well as personality and socially responsive buildings with the consideration of users’ comfort, convenience, safety and security (Ghaffarianhoseini et al., 2015).

Technological initiatives with respect to developing intelligent systems have led the connectivity of software and hardware to deliver information on interactions between humans and human-made artefacts (ranging from microscales to macroscales) by means of the utilisation of...
advanced sensors, data-processing computing, and telecommunication networks. Furthermore, the widespread availability of electronic devices (i.e. mobile phones, cameras, and sensors), the easy accessibility of wireless networks (i.e. the ubiquitous Internet), and the increase of information resources (i.e. images from cameras, location data from GPS, and social media data) have facilitated intelligent technologies evolving (Augusto, et al., 2013).

2.3.4. Discussion

Architectural design projects have shown that architectural machines can exhibit the ability of sensing and responding to real-time conditions. Cedric Price and Nicholas Negroponte pioneered the investigation of architectural intelligence through design experiments to examine self-regulating, information-feedback-based and physically controllable spaces (Fox and Kemp, 2009; Steenson, 2017; Yiannoudes, 2016). The architectural discipline that deals with the theme of architectural interactivity and responsiveness has aimed to design architectural spaces to be intelligent (Cantrell and Holzman, 2015; Fox and Kemp, 2009; Yiannoudes, 2016).

Architectural machines, assembled with sensors, processors and actuators, have performed two main tasks: interacting with people and responding to environmental conditions. For interactive architecture and responsive landscapes, changes in spatial conditions are commonly indicated by the controlling of light, sound, media screens, and kinetic structures. Recent landscape projects showed pavilions that display real-time information which is transmitted via the Internet (e.g. air quality information and social media data). Besides, bio-inspired projects have examined architectural skin systems and materials that can naturally respond to surrounding environmental conditions (e.g. sunlight and humidity).

However, when architectural machines are assembled with motion-detective sensors, due to the limit or lack of input information, their outputs (e.g. behaviours, actions and responsiveness) are mostly reactive. The majority of interactive architecture and responsive landscapes have focused on being spontaneously and promptly reactive to the motions of people in moments in which they have neared the sensors built into physical objects, as well as to instant changes in environmental conditions. To more closely reach architectural intelligence for design projects, the processor in between inputs and outputs embedded in architectural space requires the abilities of acquiring people's behaviours, characteristics and contexts and learning from previous data to perform tasks which assist their real-time activities (Biullivant, 2006; Fox and Kemp, 2009; Moloney, 2011).

Whilst architectural design projects have used machines to regulate the physical conditions of spatial elements, technological projects have focused on developing machines’ interfaces. Design projects have provided a new mode of spaces that can interact with people or respond to environmental conditions. For technological projects on building scales, the term ‘intelligent’ has been used to develop data collection and context-aware systems of indoor residents’ activities, as well as automated systems that manage electronic devices and apparatus mainly for energy efficiency. While design projects have provided socially engageable spaces, technological projects have examined building management systems.

To develop architectural machines in design projects, it is important to consider what input information can be obtained, what task can be assigned to assist users’ activities, and how their information processing (i.e. embedded systems) can be programmed to make decisions based on previously collected data. Such inputs need to involve the context of users’ activities in order to result in outputs that exhibit reasonable responsiveness. For context-aware systems, the context needs to include information on whom users are (identity), where they are situated (location), when activities happen (time) and what activities users undertake (activity) (Dey and Abowd, 1999). The design focus needs to examine how a machine can exhibit reasonable actions through architectural elements that accommodate users’ activities in their changing contexts.
2.3. Data-Driven Urban Intelligence

2.3.1. Data Visualisation

Defining Data

According to the Oxford English Dictionary, from the Latin, data—the plural of datum—is defined as 'Facts and statistics collected together for reference or analysis'. In 1587, a Latin-English dictionary, Thomas’s Dictionarium Linguae Latinae et Anglicae, first recorded datum as ‘A thing given, a gift delivered or sent’ (Furner, 2016). In the Cambridge Dictionary, data is defined as ‘information, especially facts or numbers, collected to be examined and considered and used to help decision-making, or information in an electronic form that can be stored and used by a computer’. According to those definitions, data is considered to be a set of values or a collection of facts.

There are significant differences between the terms ‘data’, ‘information’ and ‘knowledge’. Zins (2007, p.479) explains the sequential order of these terms in the study of Information Science: ‘Data are the raw material for information, and information is the raw material for knowledge’. Although these terms are interrelated, they are located in a hierarchy that is distinguished by a phase of processing. Data is ‘unprocessed material’ that can eventually become a useful form of information, and knowledge is highly dependent on how a subject or user obtains and evaluates the information. Knowledge is a combination of information that is organised by a user, and information is an interpretation of data for a certain goal or usage.

A purpose of using data is to transform a collection of facts into a form of information in order to convey their meanings (i.e. findings from facts). For the first stage, data needs to be collected. Following this, at the second stage information includes meanings from data. Consequently, this usable information can be communicated by a form of knowledge. In fact, data per se is meaningless. A collection of data is the foundation of organising information, and communication is the purpose of forming information into knowledge (Zins, 2007).

A Brief History of Data Visualisation

Data visualisation has attempted to transmit information in effective ways through visual representations. Friendly (2006) provides a brief history of data visualisation. He points out that the main root of the histories of data visualisation includes thematic cartography, statistical graphics, and medicine, which are intertwined. One of the roots of data visualisation is the history of map-making. The earliest idea of coordinates can be traced back to the Egyptian period. The coordinate system, akin to latitude and longitude, was used to determine the positions of towns until BC 200.

During the seventeenth century, the first attempts were made to depict quantitative information based on empirical observations. The early map-makers developed techniques and instruments to accurately measure geographic positions and physical quantities such as distance, time and space in response to astronomical, navigational, and territorial expansion. The earliest graphical representation of time-matter data is the depiction of planetary movements that was made in the tenth century. This illustration was intended to show the inclinations of the planetary orbits through time series. The zone of the zodiac was represented on the horizontal line, which lists Venus, Mercury, Saturn, Sun, Mars, Jupiter, Moon (from top to bottom), and it is divided by vertical lines, which serves as a time or longitudinal axis. The significance of this illustration is the use of a grid to describe the changing value of time (Funkhouser, 1936).

Figure 2.8. The chart of planetary movements (the tenth century). Source: Funkhouser (1936).
In 1644, the first visual representation of statistical data was attempted by Michael Florent van Langren (1598-1675), who was an astronomer and cartographer. He collected the different estimates of distance in longitude from Toledo to Rome, as measured by 12 astronomers, and he tried to show the wide range of variations in a graphic illustration. He noted the problem of unreliable data in the measurements and he attempted to improve the accuracy of the mapping technique to determine longitude through that data visualisation. In the 1660s, along with the beginnings of probability theory and statistics, the collection and study of social data in European countries began with the purpose of informing matters of wealth, population, taxes and land. Eventually, these data collections began to be displayed visually (Friendly, 2006).

The foundation of the modern form of data display was formed by William Playfair (1759–1823). He was aware of the need for new and effective ways to communicate data, and consequently invented remarkable methods to build statistical graphs, such as line, bar and pie charts, and circle graphs. He contributed not only to the method of visualising commercial and political data through graphs but he also added to the design technique of cartography. John Ainslie (1745–1828) and Samuel John Neele (1758–1824)—cartographers who were Playfair’s engravers—were influenced by Playfair’s effective methods of visual display such as titling, grid lines, labels, legends, and the use of colours (Spence, 2006).

In 1854, John Snow (1813–1858), an English physician, used thematic mapping to find the cause for the outbreak and spread of cholera epidemics. Snow collected data about each individual case during a cholera epidemic in Soho, London, and he drew each case as a dot on a map. This enabled him to demonstrate that the cases were clustered around a single contaminated pump that was located on Broad Street in Soho. For the first time, Snow was able to prove that cholera was a contagion that was transmitted via the water source (Snow, 1854). Snow hypothesised that the water supply could spread cholera and he verified his hypothesis by mapping a local epidemic in Soho. Snow’s data visualisation technique of overlaying the quantitative data on the map not only proved his hypothesis but it subsequently helped to prevent future epidemics of cholera.

Between 1850 and 1900, which Friendly (2006) calls ‘the Golden Age of statistical graphics’, numerical information about industrialization, commerce, and social planning began to be used for political purposes. European governments started to officially manage and nationally publish information in statistical albums and atlases. For example, from 1853 to 1876, nine international statistical congresses, which were held in European cities, began to develop graphic standards to ensure uniformity in the methods that were used to produce national statistics. The French publication Album de Statistique Graphique, which was sponsored by the French Ministry of Public Works, is a significant sample of the graphical methods that were used to depict economic and financial data. In another example, the US Census Bureau published the Statistical Atlas of the Ninth Census, which included novel graphic forms and covered a wide range of topics, such as geology, weather, population, and religious affiliation (Friendly, 2006). According to Friendly (2006), after the ‘Golden Age’, there were few graphical innovations and the early 1900s could be called a ‘Modern Dark Ages’ of statistical representation.
Prior to the 1950s, map-makers produced thematic cartography in order to present specific information in relation to geographical positions, while methods of statistical graphing were developed for economic and political reasons. Since the 1950s, alongside the development of computing technologies, geovisualisation techniques have allowed individuals or researchers to interact with the content in maps so as to produce spatial knowledge. MacEachren (2004, p.4) pointed out that: ‘Due to the communication paradigm, the purview of cartography expanded to encompass more than map-making. It was approached as a process of communicating spatial information that had inputs, transmission, and reception of information, and that therefore could be analysed as a system.’

According to Andrienko et. al. (2008), there are three approaches to visual analysis: direct depiction, summarization and pattern extraction. The spatial data-mining approach provides a more effective means of analysing spatiotemporal data through methods that computationally extract aggregations, generalizations and patterns. These methods can visually reveal the hidden spatiotemporal patterns in maps to gain information, new ideas, and knowledge (Andrienko et. al., 2008).

Kevin Lynch

Kevin Lynch, who was considered one of the remarkable urban planners in the United States, attempted to depict mental maps of three cities — Boston, Jersey City and Los Angeles — in his book entitled The Image of the City, which was published in 1960. By collecting subjective observations of the cities, he mapped the memorable information of the city in order to understand how people perceive the city. Through the use of communicative measures and graphic notations, the depiction of the mentally or psychologically described structure of the city accounted for the relationship between physical elements and symbolic and cultural dimensions (Iaconesi and Persico, 2017). Lynch (1960) stressed that the research was not only to illustrate the collection of observed images of the city but also to suggest applicable principles of city planning so as to establish an imageability or strong visuality of the city.

Lynch pointed out that the ‘apparent clarity’ or ‘legibility’ of the cityscape is crucial for inhabitants so as to indicate citizens to easily identify city elements, as well as to vividly recognise an overall pattern. According to Lynch, the types of city elements, which form some properties or characteristics in mental maps, can be categorised into path, landmark, edge, node and district. In order to analyse an environmental image, he proposed that three components can be used: identity (the identification of an object so as to differentiate others), structure (the spatial or pattern relation of the object) and meaning (the practical or emotional reflection of the object). When an observer remembers an image of a city, the aforementioned components always appear together. Whereas the five formal types consist of a city image, the three components can evoke a strong mental image through the mental process of sorting, comparing and highlighting. As a result of his research, examples of cognitive maps revealed the visual quality and the hierarchy of the importance of the city, which were distinguished by its citizens (Amoroso, 2010).

As stated by Lynch (1960), the process of interactions between the observer and the environment strengthens the observer’s understanding of city images. ‘The environment suggests distinctions and relations, and the observer – with great adaptability and in the light of his own purposes – selects, organizes, and endows with meaning what he sees’ (Lynch, 1960, p.6). He remarked: ‘Our perception of the city is not sustained, but rather partial, fragmentary, mixed with other concerns.’ (Lynch, 1960, p.2). He argued that the process of generating city images is based on the interconnection between the physical forms or structures of cities and observers’ continuous pattern findings.
According to Halpern (2015), Lynch’s said perspective was influenced by artist György Kepes’ idea of perception and design, which dealt with the relationship between subjective perception and the materialising process of figures. In his book entitled *Language of Vision*, which was published in 1944, Kepes stated: ‘To perceive an image is to participate in a forming process; it is a creative act’ (p.15). In other words, perceiving an image itself is the action of creating an image. Lynch provided a communicative method of measuring city elements in order to produce a visualisation of the inner-mind process of mapping the outer environment. Halpern (2015, p.115) noticed that ‘Lynch’s concepts of ‘imageability’ and cognition were tightly derived from cybernetics, the rising cognitive sciences, and computing’. Lynch presented how communicative symbols or notations can assist in illustrating public images of the built environment. Halpern (2015, p.116) pointed out that ‘Lynch wants to push for an anticipatory design that gathers data first in order to then retroactively unearth the characteristics that might be desired… For Lynch the best social science focuses on post-data-gathering analytics and the production of instrumentation rather than the generation of structural ideals or types for urban forms.’ Cities encompass various values (e.g. historical continuity, economic productivity, political management, and social life), which are embedded within their forms, and the assessment of such values is crucial in order to lead decisions to make a good city (Lynch, 1981). Lynch (1981, p.1) remarks: ‘Without some sense of the better, any action is perverse.’ Taking into consideration the importance of the relationship between physical forms and implicit values, Lynch suggested an analytical approach to understanding citizens’ perceptions of cities for the inquiry of ‘what makes a good city?’, as well as a communicative method of expressing city images which reveals the embedded social values.

Richard Saul Wurman

In order to make urban issues easily readable, clearly understandable and more useful for the public, Richard Saul Wurman, an American architect and graphic designer, has developed the mechanisms of organising diverse information situated in cities. His series of city Access Guides, which received the MIT Kevin Lynch Award, presented a collection of city information in the formation of maps, graphics and texts, adopting the methods of categorising, serialising, spatialising and typologising information (Steenson, 2017). Wurman (1971, p.6) states: ‘Information about our urban environment should be made understandable… Making the city observable implies allowing the city to become an environment for learning.’ Starting with the theme ‘The Invisible City’ in 1972, through organising architecture conferences, he has discussed communicative methods in respect of how to make information more available through technologies and visual means (Amoroso, 2010). Later, in 1984, he founded the TED (Technology, Entertainment, Design) conference in order to find and exchange unknown information.

Moreover, he clarified the concept of information architecture (Steenson, 2017). In 1970, architecture of information was first coined by Peter McColough, the president of Xerox, in developing the company’s management systems of services. Commonly referring to computer-based environments or digital spaces, the term ‘information architecture’ has been used to describe the design of systems that include the structure and organisation of shared information (Dillon and Turnbull, 2005). After Wurman introduced the concept ‘information architecture’ in 1975, he popularised the name ‘information architects’, which refers to the group of designers who present the thoughtful structures of graphic illustrations and the systematically organised displays of computer interfaces for easily accessible and easily understandable information (Wurman, 1996; Wurman, 2001).

In order to make urban issues acknowledgeable, Wurman attempted to organise and inform the collection of urban data. His 1999 publication entitled *Understanding USA* was intended to guide the reader in comprehending urban information, ranging from population density to health funding (Amoroso, 2010). Cooperating with a geographic information system (GIS) company, i.e. Esri, Wurman participated in designing the website ‘Urban Observatory’ for the visualisation of urban data in different cities. Via accessing www.urbanobservatory.org, any individual could compare and contrast different cities’ thematic maps, exhibiting senior and youth populations, land use, public space, traffic, temperatures, and flood zones.

Through map-making, Wurman has intended to help people’s understanding of information in cities. Wurman states (2001, p.155): ‘Most things can be found in context with a map.'
A map provides people with the means to share in the perceptions of others. It is a pattern made understandable. Through making the urban invisible visible, he provided information of cities for people’s perceptions of cities. Wurman (2001, p.187) remarks: ‘Information is not the final product. Instead, it’s usually communication.’ Reforming invisible data into visible patterns is to build the means by which to communicate with people.

Visualising Data in Cities

By means of digital technologies, data visualisations, which deal with geolocation (e.g. latitude and longitude), have facilitated the tracking of urban activities, and depict the flow of spatiotemporal patterns in cities. The open data movement has allowed the public to deal with urban-related data resources in order to navigate and monitor urban issues such as population, income, crime, and land use through digital map-making. OpenStreetMap (OSM) has been voluntarily produced through the collaboration of individuals in order to construct geographic information in cities. Furthermore, the visualisation of social media data has exposed the patterns of social interactions which are digitally communicated within cities. Picon (2015, p.14) points out: ‘Although the form of cities has not yet evolved, urban mapping has undergone a series of rapid and spectacular shifts… the map [digital maps produced by collectives] appears as one of the preferred ways to express the city’s nascent intelligence.”

Urban researchers (utilising data visualisation tools and techniques) have shown various levels of contexts in cities. Visualisation of urban data can reveal many of the significant dimensions of a city and analysis of telecommunication data helps us to understand dynamic interactions in cities (Ratti and Claudel, 2016). The MIT Senseable City Lab, which was established in 2004, investigated how digital technologies intervened in urban lives in physical space. They tried to monitor the hybridity of digital-physical space through several research studies and data visualisation projects. Projects such as mapping bicycle movements in Copenhagen (2009) and tracking the waste removal chain in the United States (2009) have shown how data collected by GPS tracking sensor devices can be summarised by visual representation with mapping techniques through different timescales. By tracking the changed positions of items, such projects have revealed how such flows are interrelated to hidden urban infrastructures (Ratti and Belleri, 2018).


Figure 2.13. Trash Track. Data visualisation of tracking the waste removal chain in the United States. Source: Offenhuber and Ratti (2014).
Location-based services in social media allowed users to virtually explore and navigate places in cities, and the visualisation of these records in the socio-spatial data could discover where citizens tended to visit and how they collectively thought of locations in cities. By using geotagging and check-ins on social media such as Twitter, Flickr, and Foursquare, individuals have started to share their interests in location-based information in urban space. Consequently, citizens using electronic devices have eventually become sensors for both collecting and sharing information of happenings in cities. The project “Here Now: Social Media and The Psychological City”, which was produced by the Spatial Information Design Lab at Columbia University, attempted to show an economic and emotional layer of a city through the scope of collective information of users’ check-ins on Foursquare and Facebook. The record of check-ins on social media enabled this group to trace those locations where users expressed their experiences. This visual representation of “a cognitive map of the city showing the collective psychology of social media users’ followed the idea of the Situationist movement, which considered that cities should be understood by citizens’ experiences (Williams, 2012, p.88).

In addition, the visual depiction of network analysis of built environment attempted to reveal the relationship between social interaction and its environment. Sevtsuk (2014) points out that to understand the complexity of the urban environment that has been shaped by social uses, the analysis of cities should focus on not only formal patterns but also spatial function and connectivity of urban elements. Jacobs (1961, p.439) criticized that city planners “cling to the unexamined assumptions that they are dealing with a problem in the physical sciences, city planning cannot possibly progress.” Following her argument, the City Form Lab at Harvard University has investigated the analytical methods to achieve evidence-based urban planning focusing on spatial and social interactions.

Using the geographic information and the data of land use, they developed the tool, which called UNA (Urban Network Analysis), and measurement techniques of urban networks (Sevtsuk and Mekonnen, 2012). Through the following methods: reach; gravity; betweenness; closeness; and straightness, the UNA tool enables to measure the relationships between forms of streets and functions of buildings. For example, betweenness analysis shows most expected pathways from a subway station to specific building properties such as retail shops. Reach analysis can count how many commercial buildings are reachable from each other in a certain radius. This technique can model a network of neighbourhoods and show how collectively people employ urban areas. In fact, the aim of data visualization is not only to see or understand various layers of cities through the lens of different urban data but also to lead decision-making for city planning.
Urban data visualisation techniques have been employed to monitor digitally collected urban data, as well as to provide the resources that present the different types of contexts situated in cities. Ratti and Belleri (2018) point out that ‘data can become either an instrument exploited for private, adversarial interests or a tool to constitute a new positive “commons.”’ As stated by Ratti and Belleri (2018), there are three types of data for visualisation: ‘purposely sensed data’ (utilising sensors such as GPS tracking devices), ‘user-generated data’ (through social networking services such as Twitter and Flickr) and ‘opportunistic data’ (collected and provided by some corporations such as mobile phone and credit card companies); each type of data needs its own set of challenges in order to identify their patterns. According to Halpern (2014), the rise of models for visualising data allowed new types of knowledge to emerge, and a set of visualisation techniques, which calculate, manage and act upon information, drove computational approaches to intelligence.

2.3.3. Intelligent/Smart City

Sensing Real-Time Urban Data

Urban sensing is prior to processing urban computing for the analysis of occurrences in cities, and to providing collected evidence that leads decision making with respect to actions. Picon (2015, p.15) states: ‘Through digital technology, urban space seems to be becoming receptive to the events and situations that affect it, in a similar way to how organisms become sensitised.’ Weinstock (2013b, p.92) identifies ‘sentience’, or the ability to be aware, as the primary driver of innovation in the urban environment; as the extended “nervous system” of a city develops the potential to sense changes in the city’s flows, and in its internal and external environments”. Spatiotemporal data have been collected by sensors and shared through ubiquitous networks.

With data being increasingly collected, which have recently been called ‘big data’, real-time data streaming has enabled citizens to promptly respond to urban changes and act upon information. Batty (2013, p.277) states: ‘Big data is certainly enriching our experiences of how cities function, and it is offering many new opportunities for social interaction and more informed decision-making with respect to our knowledge of how best to interact in cities.’ Furthermore, Batty (2016, p.324) stated that big data is ‘generated by real-time streaming from fixed or mobile sensors which implies that temporality’. According to Offenhuber and Ratti (2014, p.7): ‘The term big data refers to the availability of massive amounts of machine-readable information. This information is generated by the sociotechnical systems in which humans are increasingly entangled.’

Through the scope of visualisation of big data, human interactions have been observed in real time. The ‘Real Time Rome’ project, which exhibited in the 2006 Venice Biennale, displayed real-time flows of public transit and citizens’ phone usage. Juxtaposing depictions of the real-time data on the Rome city map, the visualisations exposed the patterns of citizens’ fluid situations within the fixed condition of urban pedestrian (Rojas et al., 2008). Moreover, based on a specific date that held an event (the 2006 World Cup final), another real-time Rome map showed the dense locations of mobile phone usage in different timeframes. The data visualisation observed different moments of citizens reacting to the event in the city. Ratti and Claudel (2018, p.211) state: ‘Big data is a way of seeing the true, real-time patterns and modes of habitation of people living in a city – it can reveal what we call “the signature of humanity.”’
Moreover, by distilling and delivering open data resources, real-time data in cities has been provided to the public so as to acquire the ongoing information of cities. For example, launched by the Centre for Advanced Spatial Analysis (CASA) at University College London in 2012, the London Dashboard website presented live observations of the city such as scenes of traffic through cameras, public transportation services, updated news, and the weather in London. Besides, Dublin Dashboard provided interactive maps and analytical graphs in order to monitor urban issues, such as housing prices and vacancy, employment rates, and social welfare, as well as real-time environmental and traffic information. Arrangements of different types of location-specific live data on online platforms, the so-called ‘city dashboards’, serve as users’ cognitive tools with which to comprehend the states of cities (Kitchin et al., 2015).

Through the utilisation of sensors, data visualisations, and pervasive networks, citizens’ perceptions of cities became faster and wider by means of digital tools. Means of perceiving the built environment have been assisted by extracting patterns from data fields through analytical processes (Halpern, 2014). Urban sensing technologies impact on how citizens react to instant urban changes informed by data processing, and have influenced urban development strategies. According to Weinstock (2013a):

> Although sensors are now ubiquitous in urban environments, there is as yet relatively little research within the sciences of the city on the development of systems that integrate the data from sensors into a sentient system, and even less on how separate sentient systems can be integrated with each other and contribute to the cognitive complexity of intelligence (p.57).

The challenge in respect of urban development lies in coupling the software of embedded urban sensing systems with the hardware of networked urban fabrics (Batty et al., 2012). Batty et al. (2012, p.482) remark that ICTs are central to developing ‘urban intelligence functions’ that are able to integrate urban data.

**Smart City**

Since the advancement of ICTs in the 1990s, an urban development strategy called ‘smart city’ has extensively utilised information technologies associated with internet networks. Prior to 1991, the ARPAnet — the foundation of the Internet — was run by the US government. In the early 1990s, the ARPAnet was handed over to commercial service providers (Keen, 2016). As a result of the nearly ubiquitous access to the Internet, smart city initiatives have focused on the integration of digital and physical infrastructure in order to improve urban accessibility, citizens’ traceability of urban information, and inhabitability with respect to the quality of citizens’ lives (Mora et al., 2017). According to Hollands (2008, as cited in Deakin & Waer, 2012, p.9), there are four main factors that feature smart cities:

- the application of a wide range of electronic and digital technologies to communities and cities;
- the use of information technologies to transform life and work within a region;
- the embedding of such ICTs in the city;
- the territorialisation of such practices in a way that brings ICTs and people together, so as to enhance the innovation, learning, knowledge, and problem-solving that they offer.
The term ‘smart city’ has not been cohesively identified, but widely used nonetheless as a slogan to promote city development (Hollands, 2008). According to Deakin and Waer (2012, p.2), city projects that expressed ‘smart’ were directed through different approaches to addressing the ‘digitally inclusive’ vision of cities: in Amsterdam, smart city projects encouraged collaboration between businesses, government and citizens so as to deal with sustainability through online platforms; SmartCity Malta promoted a mixed-use district that manages energy saving and security through ICTs; Southampton City Council provided a smart card to integrate multiple city services (e.g. online payment for bus passes, e-services for libraries and leisure).

To develop smart cities, Paskaleva (2012) questioned how this new technology could impact social and cultural factors in cities. According to Hollands (2008, p.315), ‘smart cities must seriously start with people…rather than believing that IT itself can automatically transform and improve cities’. He pointed out that smart cities should consider cultural development; the term smart cities is merely a label for intelligent, progressive, or entrepreneurial objectives of city development via high tech. According to the concept of Peripheria’s Future Internet (FI) headed by the European Commission, smart cities in the EU will be based on three themes of information technologies: Internet of things (IoT), Internet of services (IoS), and Internet of people (IoP). Moreover, Paskaleva (2012) stressed that one of the city’s assets is people and citizen empowerment of their own wisdom. Recognising people as an asset and building social networks is the practical agenda for system change in cities.

Picon (2015) pointed out two visions of smart cities. The first vision is the neo-cybernetic ambition based on the control of information. Such control leads to scenarios regarding how to develop cities based on entrepreneurship, including the participation of companies such as IBM, Cisco, and Siemens. The second vision is the bottom-up approach: the notion that individuals can invent new modes of cooperation which can lead urban development through networking. The vision of the collaborative city is that individuals’ participation can build a path towards better cities (Picon, 2015). Many smart city initiatives have sought technological solutions, envisaging urban development mostly for governance or economic growth; meanwhile, the utilisation of ICTs has enabled the effective management of civic services and urban utilities (Kitchin, 2014). Townsend (2013, p.291) states: "A web of smart urban things and services will reinforce the sociability that makes cities thrive. Instead of being centralized, many vital services could be left to the social networks of small communities." While such smart city approaches have constructed diverse networks in digital infrastructure, urban contexts in data have been increasingly collected, measured and analysed in order to understand cities and their users through computing processes.

Allam and Dhunny (2019) stressed that the development of smart technologies for cities, which incorporate big data and AI techniques, needs to focus on improving the societal liveability of urban fabrics. Smart city initiatives have indicated cities to be ‘inclusive, safe, resilient, and sustainable’, but smart city application has been rarely utilised for existing urban fabrics (Allam and Newman, 2018, p.4). The term ‘smart city’ has been used as a brand to promote technology-including new urban towns, and smart city schemes have often overlooked a citizen-centred approach (Allam and Newman, 2018). Instead, cities are habitats for people and the focus of smart cities needs to be upon how to encourage urban liveability, rather than targeting technology-based solutions. Big data has presented the source with which to understand cities comprehensively through various dimensions, and AI techniques have eventually emulated thinking patterns to assist in reading such data. The use of big data and AI techniques is a key factor that constitutes the technological foundation of smart cities, and the challenge lies in how such technologies can be applied to the liveability of existing cities or cultural and historical urban fabrics (Allam and Dhunny, 2019).

2.3.4. Discussion

Urban data visualisation is one of the means by which to observe cities. Prior to the advent of digital technologies, urban data visualisation techniques were established for accurate and uniformed methods with which to describe location-based empirical observations. Thematic mapmaking has extended the range of scopes to understand cities. For urban studies, Kevin Lynch provided communicative techniques with which to represent citizens’ perceptions of cities, and Richard Saul Wurman investigated effective techniques for organising information that reveals implicit values in cities. Urban data visualisation techniques have been developed to monitor urban occurrences, measure patterns in those occurrences, discover the overall tendencies of urban changes and produce spatial knowledge.

Urban sensing technologies have been used to collect urban data, including with regard to the potential to develop intelligent systems in cities. Urban data visualisations and urban sensing technologies with ICTs have been employed in the early stages of developing urban intelligence (Halpern, 2014; Picon, 2015). On a city scale, the term ‘intelligent’ or ‘smart’ has been expressed in providing the vision of a digitally controllable urban environment through the utilisation of ICTs which facilitate real-time monitoring of urban information through ubiquitous networks. Technological developments on an urban scale have aimed to achieve embedding digital technologies into physical infrastructure.
While focusing on technological goals, a recent question has arisen regarding how such advancement of technologies can impact on people, and what smart cities stand for (Hollands, 2008; Paskaleva, 2012). The issue of urban development has been addressed in discussing how the integration of digital and physical systems can improve the quality of citizens’ lives. Citizens have used digital maps interlocked with urban data as a tool with which to observe the status of cities. Digital maps, which present location-based data that represent the happenings in cities, have enabled citizens to more immediately interact with short-term occurrences in cities. Nevertheless, physical environments have continued to be built to maintain long-term uses for their activities. Intelligent technologies have not yet applied to improve the urban fabrics or physical conditions of cities (Allam and Newman, 2018; Picon, 2015).

Urban data present the contexts of events, occurrences and situations in cities. The open data movement has allowed individuals to use urban data, and more data have been available and shared by city governments. With the exponential growth in the ways of collecting data and sharing information, social interactions in cities became observable through the scope of data visualisation. Urban data have provided information on social happenings in cities, and the abstraction of cultural contexts in cities is no longer hidden, but rather exposed through urban data. Geolocation has provided a way in which to collect location-based data, and ICTs have facilitated sharing and integrating urban data. Data processing techniques have enabled machines to sort and evaluate massive amounts of data.

Urban data visualisation and analysis have presented ways of understanding how citizens have used cities. What is more, urban data have incorporated the contexts of events in cities, and current ubiquitous networks in ICTs have facilitated sharing information in real time from anywhere. Research on urban data, which observes citizens’ behaviours, can provide crucial facts and evidence for the design process of physical environments that focus on improving social places in cities.

### 2.4. Artificial Intelligence Computing

#### 2.4.1. Thinking Machine

**Turing Test**

In 1950, Alan Turing, a British mathematician and logician, proposed the Turing Test to provide an operational definition of intelligence in his paper entitled ‘Computing Machinery and Intelligence,’ which was published in the philosophical journal Mind (Russell and Norvig, 2016). Instead of asking the ambiguous question ‘Can machines think?’, Turing replaced the question with a test called the ‘Imitation Game’ (i.e., the Turing Test), which evaluates a machine’s behavioural intelligence to be judged by a human. This hypothetical game dictates that a machine can pass the test if an interrogator cannot distinguish which responses are from a human or a machine after written conversations between the interrogator and the machine. The test was suggested to measure whether a machine is capable of exercising human-level verbal behaviours, and to evaluate whether it can act as human ways of logical processes (Shieber, 2004).

Hodges (2007, p.4) remarks that Turing turned ‘the logical into the physical’ and, furthermore, discussed ‘the practicality (or impracticality) of the universal machine.’ In his 1936 paper entitled ‘On Computable Numbers,’ Turing proposed an automatic machine (often referred to as the Turing Machine) that can mechanically perform input (read) and output (write or edit) operations for any simulation of given algorithms. By moving a pointing scanner on a single symbol (i.e., 0 or 1) written in units on a strip of tape, this theoretical computing device was to be capable of manipulating sequences of binary digits and acting on rule sets for outputting mathematical answers. Turing characterised computable operations as physically embodied computations within machines (Hodges, 2007).

Although the Turing Test suggested a method with which to measure machine intelligence, the definition of intelligence for machines remains controversial. An American philosopher, John Searle (1980), argued that a computer capable of communicating with humans that is operated by an input/output system is programmable only to manipulate symbols without the ability of understanding meanings or semantics (Cole, 2014). Searle’s Chinese Room Argument provided an example of a simulation in which a programmed computer (assuming that it is an English speaker) could be instructed to answer any given Chinese question by merely reading figures or
characters of words. This argument points out that a computer may possibly emulate the human way of conversation or communication, but its verbal capabilities are not truly based on the human ways of thinking.

In fact, when Turing proposed the Imitation Game, he stated that the discussion of whether machines can think or not is less crucial than the design of machines’ implementation of actions in order to settle ‘common usage’ (Chomsky, 2004, p.318). He clarified his proposition of thinking machines with the consideration of various aspects of objections such as ‘machine’s lack of a soul, error, novelty, learning, continuity, flexibility and extra-sensory perception’ (Shieber, 2004, p.63). Turing (1950, p.442) stated, ‘The original question, “Can machines think?” I believe to be too meaningless to deserve discussion.’ He urged the pursuit of active investigations of constructing and improving machines that can reach human-level intellectual capacities (Chomsky, 2004).

Turing gave a philosophical framework to consider the relationship between thinking (intelligence) and machine (artefact) not to be conceptually argued but to be practically tested. Since his notion of thinking machines established the foundational vision of computer science, research on programming approaches to machine intelligence has developed computational models using logic, probabilities, learning, and background knowledge (Muggleton, 2014). According to Russell and Norvig (2016, p.2-3), to pass the Turing Test or to achieve machine intelligence, a computer needs to be programmed for the abilities of ‘natural language processing’ (communicating), ‘knowledge representation’ (memorising), ‘automated reasoning’ (information processing), and ‘machine learning’ (adapting); furthermore, for physical simulations, it needs ‘computer vision’ (perceiving) and ‘robotics’ (moving objects).

The Birth of Artificial Intelligence

In 1956, the field of artificial intelligence (AI) was introduced at a Dartmouth workshop organised by John McCarthy, often known as the father of AI (Dartmouth College), and the proposal was submitted with Marvin Minsky (Harvard University), Nathaniel Rochester (IBM Corporation), and Claude Shannon (Bell Telephone Laboratories). McCarthy coined the term AI as a research discipline to clarify the ideas about thinking machines. In the 1955 proposal for the workshop, the ambitious statement solidifying the research orientation was as follows:

The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. (p.1)

The proposal included different research projects on AI: Shannon’s application of information theory to computing machines and brain models; Minsky’s machine training or learning process based on sensory and motor abstractions that exhibit goal-seeking behaviours; Rochester’s use of randomness for an automatic calculator; McCarthy’s study of the relation of language to machine intelligence for solving problems that require self-reference. The field was initiated to share and examine the cohesive vision of research approaches to machine intelligence associated with the themes of ‘learning, search, networks, robotics, vision, reasoning, language, cognition, and game playing’ (Moore, 2006, p.88).

The classic AI systems, often called GOFAI (which stands for Good Old-Fashioned AI) as coined by John Haugeland in 1985, focused on the development of the manipulation of symbols and logic-based reasoning systems for computer programmes. Allen Newell and Herbert A. Simon designed the Logic Theorist (1955) and the General Problem Solver (1959) computer programmes for cognitive simulations underlying heuristic rules by using searching models which could generate every potential solution (Flasiński, 2016). Newell and Simon (1975, p. 113) pointed out that in psychology, human cognition could be described as a symbol system; they developed the concept of the computer as ‘a system of manipulating symbolic structures and not just as a processor of numerical data’. Newell and Simon (1975, p. 116) hypothesised that ‘A physical symbol system has the necessary and sufficient means for general intelligent action.’ In other words, ‘The hypothesis implies that computers, when we provide them with the appropriate symbol-processing programs, will be capable of intelligent action’ (Nilsson, 2007, p.9).

Meanwhile, McCarthy’s logic-based approach to AI formalised models of logical reasoning and designs of logical programming (Flasiński, 2016). In his 1959 paper ‘Programs with Common Sense’, McCarthy proposed a hypothetical computer programme called the ‘Advice Taker’ which actions were instructed by formal languages and improvable by making statements in its symbolic environment. He observed that ‘a human is instructed mainly in declarative sentences describing the situation in which action is required together with few imperatives that say what is wanted;’ and he stated, ‘In order for a program to be capable of learning something it must first be capable of being told it’ (McCarthy, 1959, p.4). Based on the idea of the Advice Taker that is
compared to artificial systems, which use these systems to produce 'thinking machines'.

Evolutionary computation is the name of the field that unifies the studies of computational algorithms (De Jong, 2002). Since the late 1960s, the term 'evolutionary computation' has been used as the name of the field. For example, in 1966, the development of evolutionary programming by Fogel et al. (University of California, Los Angeles) was introduced. In 1967, the development of evolution strategies by Rechenberg and Schwefel (Technical University in Berlin) was introduced. These developments of computer models capable of performing high-level information processes have been used to describe human-like intelligence. AGI is often referred to as strong AI that can fully implement human cognitive abilities. It has also been distinguished from narrow AI. Research on narrow AI (often called weak AI) has examined goal-oriented or problem-solving models in limited domains. John Searle (1980) coined the term strong AI to identify the machine intelligence akin to the human brain/mind, which can do any tasks on the basis of understanding. The current AI is closely related to the initial goal and the original meaning of AI (Wang and Goertzel, 2007).

Artificial General Intelligence

For the development of human-level AI, the term artificial general intelligence (AGI) has been used to describe human-like intelligence. AGI is often referred to as strong AI that can fully implement human cognitive abilities. It has also been distinguished from narrow AI. Research on narrow AI (often called weak AI) has examined goal-oriented or problem-solving models in limited domains. John Searle (1980) coined the term strong AI to identify the machine intelligence akin to the human brain/mind, which can do any tasks on the basis of understanding. The current AI research on special-purpose systems is focused on the early stage of the progress of AGI (Wang and Goertzel, 2007). Dreyfus criticised that the computer programmes designed on the basis of formalised rules and logic, commonly called classic AI or symbolic AI, were flawed to simulate the models analogous to human cognitive processes because human brain/mind could function not formally but intuitively (Kenway, 2008). Dreyfus (1992) argued that the programmes with specific objectives can only work with fixed rules and fixed data. Although the problem-solving programmes can find solutions by organising data, the results from the programmes can be the most probable within the basis of the predicted alternatives. In contrast to the human way of acquiring data in certain environments, the current computers can passively receive data under the given conditions. The meaning of AI has been argued to distinguish between human-level or real intelligence and the technical simulation of the information process.

Despite this philosophical argument addressing the discussion of what AGI or strong AI can be, most AI communities have dealt with ‘practical’ and ‘manageable’ problems for domain-specific knowledge (Wang and Goertzel, 2007, p.3). They have produced highly specialised systems (e.g. medical diagnosis, automobile navigation systems, and chess-playing programmes) and AI techniques following a symbolic, connectionist, evolutionary, robotic, mathematical, or integrative approach (Wang and Goertzel, 2007). As the aim of thinking machines is to create a human-level machine, the research on computer programmes has produced several cognitive simulations by using algorithms. The widely used term AI has encompassed the varied technical developments of computer models capable of performing high-level information processes.
2.4.2. Symbolic Computing

Knowledge Representation

The study of knowledge representation (KR) investigates how knowledge can be represented symbolically and manipulated by automated reasoning programs (Brachman and Levesque, 2004). KR systems have examined the methods of encoding and sorting information using formal symbols. In order to design intelligent systems, initially, the systems need to know situations or facts that are addressed in the world. Bench-Capon (1990, p.11) stated that in a computer program, ‘representation’ can be considered ‘a set of syntactic and semantic conventions that makes it possible to describe things’. By syntax, they can be expressed in the form of symbols, and by semantics, valid expressions can be interpreted (Bench-Capon, 1990). Through reasoning strategies, the systems can be used as tools to solve complex problems.

Davis et al. (1993, p.17) remarked that KR is essentially ‘a surrogate, a substitute for the thing itself’ which can be used to facilitate an independent entity’s reasoning about the world. Moreover, it is ‘a set of ontological commitments’ which can describe certain objects through a language. Minsky (1974) observed a human way of thinking is constantly constructed by the framework that stores memories and modifies them for acquiring changes. Davis et al. (1993) remarked that KR systems can imperfectly capture things in the world, the representations offer diverse aspects of the world that can be relevant (Davis et al., 1993).

As human minds tend to categorise things to identify their similarities, relationships, and types, the act of categorisation is a remarkable process in KR systems to describe or characterise things in the external world, and such categorisation is commonly performed to find their hierarchical relationships (Bergman, 2018). Regarding KR languages, Doyle (1977, p.3) remarked that ‘Hiearchy is an important concept. It allows economy of description, economy of storage and manipulation of descriptions, economy of recognition, efficient planning strategies, and modularity in design’. In order to form knowledge, KR techniques use many types of structures that feature a hierarchical representation of data such as lists, taxonomies, typologies, and sequences (Bergman, 2018).

The widely used KR schemes are logical representation, production rule representation, frame (structure-based representation), and semantic network (network-based representation) (Patel and Jain, 2018). Production rule representation and logical representation schemes use a form of logic. These KR schemes are mainly built upon rule-based and truth-value statements such as if/then pairs or rules of propositional logic (Poonam et al., 2010). In contrast, using semantic networks and frame schemes, knowledge can be stored in the form of graphs that involve nodes that represent objects and links denoting relations (Poonam et al., 2010). The KR schemes based on networks or structures can characterise each class of objects and the relations of its properties.

Rule-Based System

Rule-based systems (also called production systems or production rule systems) have been used to handle knowledge to interpret information in an effective manner. To understand the nature of intelligence, they were prominently featured in the study of cognitive modelling, learning systems, and problem-solving systems (Gupta et al., 1989). They have also been used to develop expert systems which were built by knowledge from human experts in various disciplines (e.g. medicine, military, chemistry, engineering, and management) (Tan et al., 2016). On the basis of reasoning processes, the if/then rules were applied to solve complex problems for computer configuration tasks.

For instance, the expert system MYCIN has been used as a consultation tool that can interact with users in the medical domain. It is often referred to as an evidence-gathering programme that provides diagnostic and therapeutic advice on the basis of medical data about patients and infectious diseases (Buchanan and Shortliffe, 1984). The expert-level medical knowledge is represented as rules in the system (approximately 450 rules and 1,000 facts about medicine), and it assists in solving difficult situations of a patient and adding new knowledge to a database (Buchanan and Duda, 1982).

A typical rule-based system is made of the working memory, the rule base (often referred to as the knowledge base), and the inference engine (Sasikumar et al., 2007). There is a separation between the data and the control in the system framework. The working memory contains the set of facts (i.e. the collected data) that represents domain-specific knowledge. The rule base is formed by the structure of if (condition) and then (action or conclusion). When the condition in a rule is satisfied with the contents of the working memory, it executes actions through symbol manipulations while seeking solutions for the given problems. To evaluate the input of new knowledge, the inference engine makes logical deductions by matching conditions and executing actions.
Buchanan and Shortliffe (1984, p.4) provided a conceptual statement to explain the basic principle of the if/then rule component as follows: IF: There is evidence that A and B are true, THEN: Conclude there is evidence that C is true. This form can be abbreviated as follows: If A and B, then C (A & B > C). In a string of the rule-based language, commonly, the if/then rule consists of the antecedent or the condition part (the left-hand side) and the consequent or the action part (the right-hand side).

Built upon symbolic knowledge in a set of rules, there are two modes of logical procedures for obtaining results: forward and backward chaining. Forward chaining is a data-driven or data-directed process to obtain a conclusion, for example (Buchanan and Shortliffe, 1984, p.4):

If A, then B (Rule 1)
If B, then C (Rule 2)
A (Data) > C (Conclusion)

Backward chaining is the process of inferring unknown conditions. It uses a goal-directed reasoning system that can find the data of conditions based on a given conclusion, for example (Buchanan and Shortliffe, 1984, p.5):

Find out about C (Goal)
If B, then C (Rule 1)
If A, then B (Rule 2)
> If A, then C (Implicit rule)

Forward chaining uses the data of if/then rules to conclude a solution. Backward chaining gathers data when it needs relative rules (Merritt, 1989). Based on these two chaining strategies, expert systems consistently check, match, and implement problem-solving processes to find the consequents.

Furthermore, expert systems are mainly designed by a domain expert, a knowledge engineer, or a system engineer (Merritt, 1989). The domain expert is the individual or individuals who can provide current expertise to collect domain-specific knowledge. The knowledge engineer encodes the knowledge in a declarative form in the system. The system engineer builds applications and user interfaces on the basis of the format of the knowledge base. In some cases, the knowledge engineer and the system engineer can work on both the tasks (Merritt, 1989). Finally, for consulting the knowledge, users can interact with the system to get advice.

In expert systems, the data represent expert-level knowledge that is structured by a set of rules. The system implements effective ways of finding evidence to reach a conclusion and inferring the required knowledge. Because of the simple logical structure, the uniform syntax, and the readable meaning of rules, the system can manage the given expert’s knowledge and practically assist users (Sasikumar et al., 2007). However, the independent rules are not interactive among themselves, and the processes take a considerable amount of time to match each rule (Sasikumar et al., 2007). By setting the computational processes of the if/then rules, the systems serve as tools in specific domains and simulate human ways of knowledge-based reasoning or logical thinking.

Figure 2.18. Interaction of a knowledge engineer and a domain expert with software tools that aid in building an expert system. Source: Buchanan and Shortliffe (1984).
2.4.3. Bio-Inspired Computing

Artificial Neural Networks

Artificial neural networks (ANNs) are computing models inspired by biological neural networks. The technique of ANNs has been popularly used for developing machines’ capabilities of text recognition, speech recognition, and image classification. It has been also applied to a wide range of information-processing systems for data-pattern classification, data mining, and robot control. Brain-inspired systems have been used to programme computing systems that can perform tasks that the human brain can.

The human brain involves approximately 100 billion neurons, and neurons communicate via electrochemical signals. A biological neuron consists of dendrites, a cell body, an axon, and synapses. Dendrites receive signals from other neurons, whilst the axon carries the signals to other neurons. The cell body (called soma) generates the activating and inhibiting signals and accumulates them. For the transmissions of the signals, a synapse connects each neuron for the inputs and the outputs. When the signals exceed a certain threshold, synapses trigger the activation of delivering the signals. McCulloch and Pitts (1943, p.115) noted ‘At any instant, a neuron has some threshold, which excitation must exceed to initiate an impulse… From the point of excitation, the impulse is propagated to all parts of the neuron’.

The design of an artificial neuron is inspired by the abovementioned components and structure of a biological neuron together with its information processing principle. An artificial neuron is made up of multiple inputs (referred to as dendrites) and one output (referred to as an axon) (Shiruru, 2016). The mathematical model of an artificial neuron has three basic sets of rules: multiplication, summation, and activation (Krenker et al., 2011). Before the input values are entered into a unit (an artificial neuron), each input (modelled as a synapse) is individually multiplied by a weight. The unit sums the weighted values with an additional parameter (bias). The unit processes the sum of the weighted input and the bias values through the activation function (also called the transfer function). In the transfer function, the sum of values is compared with a threshold. After the assessment of whether the sum of the values exceeds the threshold or not, the output produces values that are normally in a range from 0 to 1 (Gurney, 2004).

ANNs perform the computing process through the network-based structure of units (artificial neurons) in multiple layers (ANNs consist of input, hidden, and output layers). Each artificial neuron is connected via a weight from one layer to the next layer. To configure the input data through the artificial neurons in multiple layers, two network topologies are used: feed-forward networks and recurrent networks. In feed-forward networks, the data flow in one direction from the input to the output without feedback connections or back loops. Meanwhile, recurrent networks contain feedback connections (Krose and Smagt, 1996). Feedback connections in recurrent networks recall the previously processed results in the hidden layer and update the final outputs in the output layer. Recurrent networks allow keeping track of previously processed data and using an internal state, which is referred to as a memory. Recurrent networks are used as a key tool to deal with time series data or sequential data. Both feed-forward and recurrent networks are widely used for machine learning.
Machine learning is a prominent feature in ANNs and enables machines to learn data automatically, improve their performance, and predict the outputs. There are three major classes of machine learning: supervised learning, unsupervised learning, and reinforcement learning. Supervised learning algorithms are trained by labelled or categorised input and output pairs. With the given training data (training examples), a training set in supervised learning compares the differences between the desired outputs and the actual outputs, and adjusts the weight values to minimise errors; this learning process is called back-propagation (Engelbrecht, 2007). By means of clustering or associating data, unsupervised learning algorithms discover hidden patterns or features in the input data (Engelbrecht, 2007). Supervised learning algorithms are mainly used for classification (e.g. document classification) and regression (e.g. stock values prediction) models, and unsupervised learning algorithms are mainly used for clustering (e.g. social network data analysis) and dimensionality reduction (e.g. digital image filter) models (Mohri et al., 2012). Reinforcement learning algorithms focus on developing a decision-making agent’s interactions with its environment. Agents in reinforcement learning algorithms seek action choices, receive feedbacks as reward signals, and improve their actions over time. Reinforcement learning algorithms are used for goal-directed learning and planning (e.g. mobile robot movement control) (Sutton and Barto, 2018).

Information processing systems in ANNs are built upon non-linear, parallel, and distributed processors that consist of simple units interconnected in networks (Haykin, 2008). The information in a human brain can be stored by the synaptic connections and strengths in a nervous system, while the information in ANNs can be acquired by modifying the weight values via a network system. Fischler and Firschein (1987, p.64) defined knowledge as ‘the stored information of the models used by a person or machine to interpret, predict, and appropriately respond to the outside world’. ANN learning models are designed to store the acquired knowledge and make it useful (Haykin, 2008).

Evolutionary Computation

The study of evolutionary computation (EC) has extended the way of understanding intelligence. Fogel (2006) argued that intelligence is not the property solely owned by humans, and human intelligence is one of the examples to define intelligence. Fogel (2006) discussed that intelligence can be founded in the study of living beings in nature that are capable of adapting to their environments and making decisions to achieve their goals. Fogel (2006, preface p.15) remarked that

The majority of research in artificial intelligence has simulated symptoms of intelligent behavior as observed in humans. In contrast, I propose a definition of intelligence that does not rely only in its comparisons to human behavior. Intelligence is defined as the capability of a system to adapt its behavior to meet its goals in a range of environments.

With this aspect of intelligence, the theory of natural evolution is applied to the development of EC algorithms so as to generate artificial intelligence.

Inspired by the Darwinian principle of natural selection and the microscopic view of natural evolution featured in molecular genetics, the study of EC has led to the development of computational tools for solving complex problems (Eiben and Smith, 2015). EC techniques are built upon the following idea of natural evolution: ‘given a population of individuals, the environmental pressure causes natural selection (survival of the fittest) and hereby the fitness of the population is growing’ (Eiben and Schoenauer, 2002, p.1). To describe the main components of EC, biological expressions are made compatible with computational terms: an environment is considered a given problem that limits resources; individuals represent candidate solutions; fitness values denote the measurements of quality that are used to seek better candidate solutions; an evolutionary process produces a population of individuals that are increasingly fit or gradually better adapted to the environment, and an EC process can be understood as a heuristic process toward maximising objective functions within a search space (Eiben and Smith, 2015).

Evolutionary algorithms (EAs) use the aforementioned components such as a population of individuals (candidate solutions), fitness values (the measurements of quality), and an environment (a search space), with the schemes of population-based, stochastic, and direct search (Hartel-Sheltenstein, 2014). EAs are known as ‘generate-and-test’ or ‘trial-and-error’ algorithms (Eiben and Schoenauer, 2002, p.2). Through constant searching processes, the reproductions of new solution candidates (offsprings) generated from two selected candidate solutions (parents) result in better solutions.
To implement the searching process, EAs are driven by two main operators, namely variation and selection. Variation operators create the diversity in a population, whereas selection operators serve as a force to improve quality (Eiben and Simith, 2015). Variation operators commonly include the methods of recombination and mutation. Recombination exchanges or rearranges the existing information in the parents, and mutation explores new information (Bartz-Beielstein, 2014). In the form of pseudo-code, initially, the process starts with an arbitrarily selected population. From the randomly selected individuals, the population evolves toward better solutions through the processes of selection that favour individuals of higher fitness, recombination that mixes information from parents and passes it to the offspring, and mutation that brings novelty in the population (Bäck and Schwefel, 1993).

In EAs, candidate solutions (parents/offspring) that form possible solutions are called phenotypes, and the underlying structures (the genetic coding) that represent phenotypes are called genotypes (Bartz-Beielstein, 2014). The form of genotype and phenotype mapping can be various depending on the given problem (for instance, bit-strings, integers, vectors, and trees) (Eiben and Schoenauer, 2002). Selection operators act on phenotypes to encode information, and variation operators act on genotypes to decode information through recombination and mutation (Eiben and Simith, 2015). The process of EAs is described by the following two terms: exploration and exploitation (Eiben and Schoenauer, 2002). Exploration is commonly performed by search operators such as recombination and mutation, and exploration is handled by selection (Crepinsek et al., 2013). Exploration discovers new regions of a search space, while exploitation optimises solutions through these regions of a search space, which were previously discovered (Crepinsek et al., 2013).

Through a population-based framework of searching processes, EAs provide probabilistic solutions. In terms of optimisation problems, EAs search for the most feasible scenarios for solutions. EAs have been used for problem solving in various domains such as planning and scheduling for business, structural optimisation for engineering, and data retrieval and mining for public services (Blum et al., 2012). Blum et al. (2012, p.2) provided the examples of problems that EAs can be applied to: "how can a ship be designed for highest safety and maximum cargo capacity?" and "how should the production in a factory be scheduled in order to satisfy all customer requests as soon and timely as possible?" To solve such problems, the most possible solutions in different scenarios can be obtained through EAs. Instead of seeking a deterministic solution, EAs implement searching processes to provide the most probabilistically or statistically better scenarios.

2.4.4. Discussion

The aspiration of realising machine intelligence in computer science has led to the progressive development of software which results in making machines’ information processing systems practically useful and more available in the real world. The human way of logical thinking was prominently referred in creating thinking machines, as human intelligence is generally considered an exemplary model that has the ability of thinking. Meanwhile, biologically inspired approaches to machine intelligence have developed tools capable of problem solving. By designing computer algorithms, varied ways of setting rules have attempted to facilitate simulating machines' behaviours as if they can solve problems.
Alan Turing pointed out that terms such as ‘intelligent’, ‘thinking’ and ‘understanding’ are vague and not worth arguing in developing machines; rather, such terms should be replaced by the question ‘Can a machine behave like a thinking person?’ (Levesque, 2017, p.10). Since this question was framed, many computer science researchers have struggled with emulating the human property of thinking and finding clues in a human’s black box in order to achieve machine intelligence (Downing, 2015). Although no machine has yet to sufficiently perform human-level intelligence (i.e. machines have not yet passed the Turing Test), the term ‘AI’ has been used to describe ongoing researches and achieve the target of accomplishing the design of machines that can behave or act intelligently to be judged by external observers (Russell and Norvig, 2016).

The term ‘AI’ has been widely expressed to encompass diverse approaches aiming to realise intelligent machines. The development of KR and rule-based systems has focused on how human knowledge can be manipulated using symbols in computers that refer to human logical thinking. Expert systems have been designed to make reasoning systems based on if-then rules that store the facts or interpretations of domain-specific knowledge. Inspired by biological neurons, one of the major AI techniques, i.e. ANNs, has enabled machines to perform a wide range of tasks, such as the recognition of images/voices, through networks of single computing units. EC has provided an effective way in which to review different types of scenarios that can lead a user’s decision making. Not only can a human being achieve goals or solve problems, a wide range of living systems are also capable of making decisions in order to adapt to their environment (Fogel, 2006).

Whilst AI systems have attempted to emulate a human or living being’s abilities so as to enable machines to perform a wide range of tasks, an architectural machine can perform a narrow range of tasks which accommodate the activities of occupants. The task of architectural machines in design projects has been oriented towards accomplishing human interaction and environmental responsiveness. As architectural spaces are situated between people and their environment, constant changes of these two factors have been subject to architectural machines. The behaviours of architectural machines can be judged by whether they sufficiently assist the activities of users or occupants, which is a given task.

2.5. Conclusion

With the advancement of information technologies, the challenge of giving cognitive abilities to artefacts has been attempted for architectural spaces, cities, and machines. The aim of architectural intelligence has focused on how spatial conditions can be controlled using real-time information on changes in occupants’ activities and in nearby environmental conditions. Urban intelligence has been approached by developing digital tools with which to share location-based and time-related information pertaining to the diverse contexts of happenings, occurrences and situations in cities. Researches on artificial intelligence have encompassed diverse ways of setting computer algorithms that facilitate acquiring given information and performing specific tasks. In fact, intelligence is the ability of information processing, and an entity’s intelligence can be measured by its outputs.

For architectural design projects, the first architectural machines (e.g. the Generator and the SEEK project) attempted to apply cybernetics and utilise computer technologies to control spatial configurations that can adapt to constant changes in users’ needs. However, only drawings or pictures remained in showing the idea of architectural machines; thus, it is difficult to verify how they were operated through timeframes. Projects on interactive architecture and responsive landscapes have shown that architectural spaces can sense the presence of people and physically exhibit actions. The current state of architectural machines’ abilities of interactivity and responsiveness is such that they can exhibit reactive motions which are activated or deactivated by instant changes in the surrounding conditions. As responsive technologies can detect simple information (e.g. the presence of nearby people), their actions are limited. Nevertheless, the application of responsive technologies has enabled architectural machines to sense and act. For further development of such machines, the challenge of achieving architectural intelligence (that lies between sensing and acting) has been issued.

On the other hand, for projects on a building scale, the achievement of architectural intelligence has also been targeted by computer scientists and engineers in a different way. To maximise the comfort of occupants’ activities as well as the efficiency of energy usage, projects on IBs, IEs and AmI have developed computer interfaces that control buildings’ apparatus mainly based on information on lighting, electricity usage, temperature, and air quality. For technological projects, intelligent systems in buildings have been examined in order to test whether a central computer could achieve the goal of energy-saving, cost-efficient, automated management of electronic and mechanical facilities.
Intelligence is the ability of processing information situated between inputs and outputs. To enable machines to acquire detailed information on occupants' activities or on the surrounding environment, the input data need to involve the contexts of them. To enable an architectural machine to be intelligent, its ability of information processing needs to be capable of evaluating the contexts of surrounding circumstances, selecting an action from a number of options, and providing a suitable space for the activities of occupants. To evaluate an architectural machine's intelligence, its outputs or behaviours can be judged by whether its spatial configurations of physical conditions are reasonable for occupants' activities.

Urban data have included the social contexts of happenings in cities. Through the use of location-based data, thematic maps have been made to observe when, where and how citizens have used cities. Information on the social uses of cities, which were commonly considered to be one of the abstract and hidden contexts in cities, has been revealed by depicting the collection of data. Urban data provide previously collected records of citizens' interactions in their physical organisation of cities. They have been mostly used to illustrate happenings in cities, but rarely applied in order to improve physical conditions in cities. However, urban data can be useful in discovering regional characteristics, addressing issues or problems based on facts and evidence in respect of how citizens have used cities, and consequently leading the modification of physical conditions.

Recently, ubiquitous networks in ICTs have allowed sharing urban information in real time. As interconnected computer networks (e.g. the Internet) have facilitated collecting, exchanging and extracting urban information based on geolocation, the accessibility of urban information is no longer dependent on users' locations. As a result of the extension of ways of sharing information, big data have been produced and used to observe the trends and patterns in citizens' activities. Since urban information is more available and easily accessible, smart city projects have promoted a wide range of applications of ICTs in developing urban systems. The early stages of smart city initiatives were oriented towards proposing new urban town design and planning that integrate digital infrastructure that maximises the networks of devices, facilities and transportation in cities. Recent discussions have explained how the concept of smart cities can focus on the management of urban happenings, situations and events in order to improve the liveability of existing cities.

Recent architectural design projects have demonstrated pavilions that can transmit real-time information (e.g. social media data and air quality data) that is shared via the Internet. By selecting specific information from digital platforms (e.g. websites), the pavilions displayed urban information on architectural components. Although such projects showed that urban information was merely displayed on architectural installations, they presented the possible applications and utilisation of current information technologies in architectural design. To achieve architectural intelligence, information processing techniques need to be embedded in architectural machines that are oriented towards exhibiting reasonable behaviours.

To enable machines to acquire given information, AI techniques have developed rule sets in computer algorithms. In KR systems, the act of categorisation is a prominent feature to describe or characterise things in the external world by using formal symbols. In expert systems, the if-then rule, which includes interpretations of data, has been used to store domain-specific knowledge. ANNs have shown that when the sum of values is approximately close to a given threshold, a system triggers actions. The input values of ANNs are simplified (commonly in the range between 0 and 1) and evaluated by a threshold. EC has presented a searching technique based on trial and error and oriented towards better solutions. Through selection and variation, the algorithm searches for possible solutions.

Although AI techniques are not directly applicable in designing architectural machines' behaviours, the basic principles of the techniques can be useful in facilitating an architectural machine’s information processing and consequent actions. For instance, previously collected data that include occupants’ activities as well as environmental conditions can be categorised and organised in if-then rules in order to lead actions. As for an architectural machine’s decision-making process, input values can be manipulated or simplified to be evaluated by a given threshold. Searching techniques, built on trial and error, can be applied in producing possible design options of architectural machine’s spatial conditions.

For architectural design projects, due to the limit of information when utilising motion sensors, architectural machines’ behaviours have been mostly reactive. Regarding city scales, through visualising the collection of urban data, machines became capable of sensing what is happening in cities, but urban data have rarely been applied in designing physical spaces. Urban sensing has enabled machines to perceive the diverse contexts of citizens’ lives. Moreover, information processing techniques have enabled machines to manipulate data that are easily readable and more manageable. Aiming to realise AI, rule sets in algorithms have facilitated the implementation of machines’ logical consequents of inputs.

There is the potential of utilising current information technologies to enhance an architectural machine’s ability of perceiving information on its surrounding circumstances. Through geolocation, location-based and time-related data have been increasingly collected. ICTs have allowed the real-time accessibility of urban data. Information processing techniques have been advanced to make data practically useful. In this research, the design experiment focuses on implementing the design of an architectural machine that is capable of acquiring the contexts of citizens’ activities as well as environmental conditions by utilising urban data, providing an interchangeable condition of physical space through deployable kinetic structures, and setting an algorithm to be embedded in the machine based on information processing techniques in order to lead its intelligent actions.
Chapter 3. Design Methodology
3.1. Introduction

To design artefacts that can perceive information, there have been diverse attempts and tests aiming to achieve intelligent systems. The meaning of the term ‘intelligent’ has been constantly discussed to be defined for machines. Meanwhile, the term has been widely expressed in asking how machines can be developed through the application of information technologies. It has been used to establish a goal of enhancing machines’ ability of information processing. Various machines have been designed to perform a specific task with particular information. The advancement of technologies has facilitated effectively storing, transmitting and manipulating data. And, it has impacted on increasing ways of exchanging data and forming information. With the development of information technologies, data will likely be more collected and more available. Nowadays, various types of data are directly obtainable through online networks.

In this research, an architectural machine’s ability of information processing was developed with online data including the contexts of activities of people and the environmental conditions in a city. The research sought available data to gain information on those two factors situated in open spaces in a city. Through collecting data, happenings in the spaces could be observed. Collected urban data were visualised, analysed and categorised so as to form readable information. Furthermore, the research carried out programming an algorithm involving rulesets to read specific data and command an architectural model’s reasonable behaviours. Through several simulations, the algorithm could be modified to improve a self-control system and develop the design of architectural behaviours. The design experiment tested how the rulesets in the algorithm could facilitate an architectural model’s intelligent actions. Summing the abovementioned procedure, there are seven steps in the research process for the design of an architectural machine, which are illustrated as follows:

1. Collecting data related to open spaces in a city.
2. Observing the contexts situated in open spaces in a city.
3. Selecting an open space in which to implement an architectural model’s behaviours.
4. Using location-based and time-related data allocated in an open space to programme an algorithm.
5. Modifying the algorithm through real-time simulations.
6. Evaluating the architectural model’s behaviours.
7. Improving the algorithm and developing the design of the architectural model to exhibit intelligent actions.

To monitor and evaluate how the real-time behaviours of an architectural model perform through rulesets in an algorithm, Grasshopper in the Rhino 3D computer-aided design application was mainly used in the design experiment. The software is useful for directly visualising 3D physical forms from a set of algorithms, and has been widely used in geometrical studies, generative designs, and parametric approaches to examining architectural spaces. A community of designers have produced and shared sets of different types of algorithms formatted in simple plug-ins in Grasshopper. Several plug-ins, which were shared by the community, were applied in order to set an algorithm for this design experiment and develop an architectural space. It was expected that the algorithm established in this research can be applied in continuing to develop architectural machines through the widely used design tool.

To implement an active space, the design experiment examined the techniques of extracting urban data, programming rulesets, and modelling architectural behaviours with kinetic structures through algorithms. Data visualisation techniques have been utilised in transforming urban data into readable information, mostly to discover the overall happenings in a city. Previous architectural machines assembled with sensors/actuators have facilitated the data flow from detecting the presence of people to responding through architectural counterparts. Kinetic structures have been practised in enabling architectural structures to behave flexibly. The design experiment in this research attempted to integrate those techniques.

This research investigated a method with which to develop an architectural machine’s embedded system, which can perceive real-time information by receiving data via online networks and, consequently, can control the physical operations of spatial components through an algorithm. The technique or algorithm, which this research investigated was designed to suggest one of the ways in which to enhance an architectural machines’ ability of information processing. This chapter describes the main data-processing components in the algorithm.
3.2. Data Resource

Urban data visualisation techniques have been employed in dealing with many types of spatiotemporal information situated in cities. Urban data have been used to form geographical information and generate spatial knowledge. Location-based and time-related data describing urban happenings are mostly stored in the form of tables arranged using columns and rows (i.e. the majority of the urban data are collected through tabular formats). A data visualisation technique could be similarly reused to deal with different types of tabular-formatted urban data.

Since urban sensing techniques were practised, it was further applied in order to enable an architectural model to read urban data.

For the design experiment, two specific datasets, including social event and weather data, were selected and scrutinised. The data were collected through the following websites:

- NYC permitted event dataset: https://data.cityofnewyork.us/City-Government/NYC-Permitted-Event-Information/tvpp-9vvx

- NYC weather dataset: https://www.worldweatheronline.com/developer

The two open data portals provide the records of previously collected data. Using historical data, it was possible to observe the overall changes or patterns of social happenings and outdoor conditions in a city (the result of the analysis is described in chapter 4). Moreover, the two portals allow extracting data via an API (application programming interface). The most recent or current data can be extracted through the interface by calling and requesting specific data via a URL (see Figure 3.1). The NYC open data portal offers different types of data formats (e.g. CSV, JSON, XML and RDF). In the weather portal, two formats (i.e. JSON and XML) are extractable. Those data have different file types but are all convertible into tabular formats (e.g. openable in Excel software). Through an API, the portals allow directly receiving current data within software (instead of downloading them).

To observe the information on social contexts and environmental changes in a city, this research focused on using these open data. The technique of using an API is mostly applicable to design software applications (e.g. websites). Popularly used social media (e.g. Facebook, Twitter and Instagram) allowed accessing their APIs, including users’ information with geotagging. Several kinds of research and software application developments could use a vast amount of data in social media through APIs. Recently, however, access to such data has been limited due to the issue surrounding privacy. In this research, using open data that are available without restrictions, a data-processing technique was developed.

3.3. Extracting and Sorting Process of Input Data

Receiving data from an API facilitates real-time streaming processing. The technique of extracting and sorting social event and weather data from the APIs enabled providing information on real-time changing conditions in an open space in a city to an architectural model. An algorithm in the software Grasshopper could directly receive current data through the interfaces on the two websites. To select specific data (e.g. social event data held in Bryant Park only) from the vast amount of data (e.g. all social event data in NYC), the URL addresses were specified in the API sections on the websites. The following URLs, which include API keys (i.e. the underlined parts) for accessing the data directly, were typed into the algorithm in order to obtain requested data from the interfaces:

- Social event dataset (Bryant Park event data only): https://data.cityofnewyork.us/api/views/se2t-6t4j/rows.csv?accessType=DOWNLOAD&bom=true&format=true&delimiter=;  

- Weather dataset (current weather conditions in Midtown data only): http://api.worldweatheronline.com/premium/v1/weather.ashx?key=3a7d057684c048d5b6c195142201406&q=10018&format=json&num_of_days=1&extra=localObsTime&date=today&fx=no&cc=yes&mca=no&fx24=no&includelocation=no&show_comments=no&showlocaltime=yes

Figure 3.1. Raw material of data (in JSON format) extracted by URLs and displayed on webpages.
The scripts in a C# component in Grasshopper could convey the texts in the algorithm, and the sorting process in the algorithm indicated the display of selected texts and numbers (see Figure 3.2). The input of current time was used to find the list of social events held on that day. A timer component was set to update weather data every 5 minutes through the C# component in order to gain the current weather conditions. Through a sorting process in the algorithm, requested texts could be selected and displayed. They constitute the raw material of data to be processed to form information. In other words, the sorted texts, obtained through the URLs, were not yet meaningful enough to give any information or meaning in the algorithm, but they presented the records or lists.

```csharp
if (url == null) return;
System.Net.ServicePointManager.Expect100Continue = true;

var request = System.Net.WebRequest.Create(url);
var response = (System.Net.HttpWebResponse) request.GetResponse();
Console.WriteLine(response.StatusDescription);
Stream dataStream = response.GetResponseStream();
StreamReader reader = new StreamReader(dataStream);
A = reader.ReadToEnd();
reader.Close();
dataStream.Close();
response.Close();
```

3.4. Matching Process of Input Data and Categorised Previous Data

The analysis of previous data was crucial in providing the meanings of the input data. The one-year historical data were analysed so as to categorise the contexts in the data. The analysis resulted that there have been repeatedly used texts in both social event and weather data (i.e. the majority of the social events have been repeated for different times, and weather descriptions have used the same expressions). It identified and classified repeated event names and weather descriptions. Repeated texts were listed along with the categorisation of contexts and saved in files in order to match the input of current data. It was used to indicate the act of categorisation and interpret the upcoming data. The current time measurement in the algorithm distinguished whether or not an event was currently happening and whether it was daytime or night-time. The input data providing starting and ending times of events as well as sunrise and sunset times were used to recognise the current conditions in an open space (see Figure 3.3).

There are two different types of saved files in the algorithm: one file is used to distinguish the contexts of texts; meanwhile, the other is used to give numbers so as to alter the behaviours of an architectural model that consists of foldable frame structures and openable skin panels. The inputs, which could be updated by timers, provided real-time data, and the rulesets that read the contexts in the input data and transform text-based data into the given numbers indicated rotating an architectural model’s structures and skin panels. The algorithm was programmed to find the
input of texts (i.e. current event names and weather descriptions) from the categorised texts (i.e. the contexts are given in the list of previous data). After matching the texts, the algorithm could recognise the type of current event and distinguish the weather conditions. After classifying the contexts of input data, python components (built on if–then rules) bring the numbers from the saved files in order to rotate an architectural model’s structures and panels.

3.5. Real-Time Simulation of Architectural Model’s Behaviour

The design of an architectural model focused on simulating deployable and foldable structures to be placed in an open space. After simulating architectural behaviours (without connecting to the datasets), the numbers (angles of rotation) could be determined and saved in the files. The algorithm instructed the architectural model to activate folding and unfolding structures and adjust the opening of skin panels. Movement of the frame structures was programmed to be responsive to currently happening events, whereas the opening of skin panels was programmed to be adjustable, depending on the weather conditions.

Data were used to enable an architectural model to perceive real-time happenings in an open space in a city. The digital simulation of architectural behaviours was carried out to evaluate its real-time data processing. The real-time simulation showed that the algorithm facilitated the automation of an architectural model’s behaviours to be responsive to updated data, as well as facilitating the conversion of data into usable information which instructed its reasonable actions. As this established algorithm is an embedded system, only the physical forms of behaviours could be visible (i.e. how the model recognises that the current happenings in a location were invisible). The real-time simulation of the digital model was observed with a display of texts showing the input data and the output of numbers (angles of rotation), and the algorithm was modified through evaluations.

A prototypical physical model, which shows one part of the frame structures including the skin panels, was tested by using the Firefly plug-in component that bridges the algorithm with an Arduino microcontroller (see Figure 3.4). After loading the Firefly_Firemata script in Arduino software, an Arduino board was able to receive the numbers resulting from the process of the algorithm (i.e. the numbers of angles to rotate the structures and panels) through the Firefly component (see Figure 3.5). The board could transform the numbers into the signal in order to operate the actuators (i.e. servo motors) so as to show the behaviours of the physical model in the real world. The picture of the physical model is shown in Figures 4.28-4.29 in Chapter 4.

Figure 3.3.
Simplifed version of the algorithm between sorted data to the numbers that alter an architectural model’s behaviours.
3.6. Conclusion

This research investigated the development of an algorithm mapping the sequence from extracting urban data to exhibiting architectural behaviours. The algorithm focused on constructing a logic-based sequence to examine the design of an architectural machine that can act intelligently with particular information (i.e. data allocated from urban open space). The distinction between the architectural model in this research and previous architectural machines is that the embedded system was designed to receive real-time data, select an action from alternative options, and perform a self-regulating behaviour that exhibits the motions of kinetic structures through given rulesets. The architectural body was linked to the information processing of automatically updated input data, matching with categorised data saved in files, and indicating architectural behaviours. The saved files, based on the analysis of previous data, served as memories with which to read upcoming data. The established algorithm demonstrated that the collection of data in digital spaces can be used to offer detailed information on users’ location, time and activity as well as the environmental contexts, to exhibit logic-based behaviours in an architectural model.

As stated in the literature review, there is a gap between developing physical and digital spaces to deal with intelligent systems. Based on sensors and Arduino, architectural machines have been designed to interact with people and respond to the environment. However, sensors installed in their bodies cannot deliver enough information on the activities of people or the environmental conditions involving the contexts. The recent technique of extracting and selecting data from a digital space has been employed to display real-time information on architectural components to be read by visitors. On urban scales, data has been manipulated as the primary material to visualise readable information and to understand the changing circumstances in cities as well as to design software applications (e.g. websites). In this research, the algorithm was constructed by integrating different techniques that have been developed separately for urban or architectural spaces.

The algorithm in this research showed that text-based data can be transformed into numerical values that indicate architectural behaviours, and such transformation with the act of categorisation facilitated a context-aware system for an architectural model. Architectural machines have been designed to provide active spaces built upon the process of the input of sensing and the output of responding. The input of sufficient information, including contexts to their embedded systems, is crucial to indicate their specific performances. Data in digital spaces have been collected and exchanged mainly to produce digital tools that provide information to
be read by people. In this research, urban data were directly incorporated into an architectural model and inserted into the algorithm to enable the architectural model to perceive real-time situations in a location in a city. The architectural model’s behaviours were programmed to be responsive to the texts (i.e. the name of events and weather descriptions) in the data as if it can read texts (within repeatedly used fixed data). This technique provides information, based on text-based data, to an architectural machine.

The basic principle of data processing in this algorithm can be applied to different datasets and locations. Most of the urban data have been collected with tabular formats. The sorting process of real-time data (i.e. the selection process of data associated with a location) extracted from APIs provides current conditions that describe the users’ activities and the environmental contexts. The matching process with the defined data enabled the architectural model to categorise the texts in the data. This algorithm illustrated that enough information from users and the environment can be obtained digitally and the rulesets can then transform text-based data to information that indicates architectural behaviours. An architectural model can be programmed to exhibit a self-decision-making process for its behaviours built upon the data processing algorithm.

This design methodology presented how the text-based data can be processed so as to control architectural behaviours. In fact, a text is a kind of symbol that can be presented, converted, or transformed into a meaning after information processing. In this research, the design procedure included categorising previously used texts in social and environmental data. By inserting the simplified group of the data, the algorithm could match with real-time data. The simulation of an architectural machine’s real-time responsiveness and self-control could be monitored via a 3D display using the design tool Rhino 6. In order to evaluate its reasonable behaviours in response to specific contexts in the data, the input of texts and the output of forms could be observed directly, visualising both of them via the tool. To improve the ability of information processing in an architectural machine, rulesets in an algorithm between the input and the output were mainly investigated in this research. The logic-based sequence of the algorithm, the result of the design experiment and development as well as the real-time simulations are illustrated in chapter 4 and 5.
Chapter 4. Design Experiment:
Sensing Urban Data and
Performing Architectural Behaviour
4.1. Introduction

One of the megacities, New York City (NYC), was selected in examining the design of an architectural machine. The growth of population in the city has caused an increase in the volume of buildings. Meanwhile, open spaces in the city have become one of the important places for citizens’ social interactions. As a result of managing and promoting their activities, diverse events have been organised in open spaces in the city. Information on their uses of open spaces has been shared via open data. With regard to the increasing value of urban open spaces and the recent availability of open data with which to observe citizens’ public events, the design experiment investigated one of the ways of designing an architectural machine to be installed in urban open space. The design experiment tested whether the machine could acquire the contexts of circumstances in an open space and provide suitable physical conditions that assist the activities of people.

The open data movement has allowed the public to access, reuse and republish data that are offered by governments. The movement has been extended globally to share government-collected data. Since the US open data portals were launched in 2009, nearly 200,000 datasets have been made available and easily accessible via data.gov (Kim, 2019). The NYC open data portal has provided a vast amount of data along with categories such as business, health, education and environment. The publicly available open data have included urban-related data resources such as population, income, and land use. Some of the city data contain information with geolocation (e.g. longitude and latitude, zip code, and address). Using such geographically related data, thematic mapping techniques have depicted different conditions in the city. Individuals, institutions and business sectors could develop their own digital tools by freely using and manipulating open data without restrictions (Ayre and Craner, 2017). Urban data have recently been made available to the public, and the contexts of the data have varied.

In this research, data visualisation and analysis were carried out in order to understand the overall social contexts related to open spaces in the city. Although urban data were not sufficient to trace individuals’ uses of open spaces, they were useful in illustrating the regional characteristics of open spaces by means of information on land use, population, and types of open spaces. With one of the types of NYC open data, i.e. NYC Permitted Event Information, it was possible to observe the contexts of public events held in open spaces and that took place at different times. To design an outdoor space, NYC weather data were also used. Previously stored event and weather data (which were collected in past years) were analysed in order to observe how citizens have used open spaces.

In the following description of the design experiment, the term ‘architectural model’ was used to denote not only an architectural machine’s three-dimensional representation of physical forms but also a control system capable of information processing that results in its consequent actions. The design experiment focused on implementing rulesets in algorithms mapping the sequence from data containing social and environmental contexts to architectural behaviours (i.e. motions of kinetic structures). Design experiment 1 tested whether the actions of deployable structures could respond to the schedule of events (using NYC Permitted Event Information) and the texts of weather descriptions in an open space. It was programmed to verify whether an algorithm could process data in real time and transform text-based data into architectural behaviours. In design experiment 2, developed from the sequence, the algorithm was further extended to perform the act of categorisation. It was programmed to read the contexts of social and weather data. The technique of data manipulation was applied in order to enable the architectural model to sense real-time happenings, and rulesets in algorithms were programmed to regulate architectural behaviours.
4.2. Data Visualisation and Analysis of Open Spaces in New York City

4.2.1. Spatial Distribution of Open Spaces in New York City

To depict social contexts situated in NYC, the research explored available data and used urban analytical tools. Through the use of the ArcGIS and UNA (Urban Network Analysis) tools, an analysis of the contextual relationship between open spaces and the surrounding building properties in NYC was conducted. Using one type of geographical data, i.e. MapPLUTO – Manhattan, the ArcGIS tool classified NYC data so as to show the land use of buildings and the locations of public parks/squares, and the UNA tool was used to count the number of buildings within 1km of each park/square. The data visualisation and analysis illustrated where open spaces are located and distributed in different districts in the city. As it is likely that people may visit nearby open spaces located within walkable distance, 1km was measured for the analysis of the relationship between open spaces and the surrounding buildings.

Before the analysis, a direct depiction of the building height and land use data was carried out in order to understand the overall contextual organisation of buildings in the city (see Figure 4.1). The data visualisation presented that the majority of tall buildings (more than 29 storeys) were concentrated in districts 1 (Lower Manhattan) and 5 (Midtown Manhattan) (see Figure 4.1, top left). The land use data showed that tall buildings are mostly associated with commercial and office uses (see Figure 1, top right). Except for districts 1 and 5, the majority of buildings are associated with residential or mixed uses. The locations of public facilities and institutions (e.g. community centres, museums, and libraries) are dispersed and widely ranged in all districts.

Furthermore, the data visualisation of 2015 NYC census data presented the Manhattan population by age (see Figure 4.2). The population below the age of 20 was relatively higher in Upper Manhattan (districts 9 to 12) than in other areas. Meanwhile, the population between the ages of 20 and 40 was high in districts 6 and 8. In districts 7 (Uptown West) and 8 (Uptown East), which are adjacent to Central Park, the population above the age of 40 was high. There were few residential properties in districts 1 and 5. Through the data visualisation of population, it was observed that in district 5 the majority of tall buildings are associated with commercial and office uses, whilst there are few residential buildings in district 1. The data presented that the parks/squares in district 5 are likely to be used more frequently by office workers or visitors than by residents of Manhattan.
Using the geographical data, an analysis of the relationships between parks and the surrounding buildings was carried out. The UNA tool was used to measure the number of buildings within a distance of 1km from open spaces. According to Azmi et al. (2012), 1km (0.62 miles) normally takes approximately 12 minutes to walk, and 5 to 10 minutes can be considered the preferred walking time before choosing to take transportation. Measuring and counting the surrounding buildings located within a 1km radius from the entrances of each park, the tool presented a comparison of the counted numbers within a range of colours (high in red and low in blue) (see Figure 4.3). It showed that there were more accessible public parks in Lower Manhattan (districts 1 and 3) and very few open spaces in Midtown and Upper East (districts 5 and 8). Moreover, the distributions of open spaces of different types were analysed as follows: garden-type parks were densely located in district 6; community parks and playgrounds were widely distributed and partially clustered in district 3; and triangular plazas were concentrated in district 2 (see Figure 4.4).

The analysis further investigated the categorisation of NYC parks, which are distinguished by the following two land uses: residential and commercial (Figures 4.5 and 4.6). The tool counted the number of residential and commercial buildings that were located within a distance of 1km from the entrances of each park. A list of parks associated with those two categories was provided. As a result of the analysis, on Manhattan Island the following parks were highly associated with commercial buildings: Central Park, Bryant Park, Madison Square Park, Union Square Park, Herald Square, and Greeley Square Park. More than 150 commercial buildings were located within a distance of 1km from these parks. It was noticed that there were relatively fewer parks/squares in Midtown (district 5), which is more of a commercial and office area than are the other districts.

The results of the data visualisation and analysis present that the parks/squares located in Midtown (district 5) have some characteristics that are distinguished from the open spaces in other districts. In district 5, the open spaces are surrounded by tall and high-rise buildings which are mostly associated with commercial or office uses (almost none are residential properties). There are very few open spaces in comparison to other districts. Whereas other open spaces in Manhattan could be frequently used by residents, the open spaces located along Broadway (e.g., Bryant Park, Madison Square Park, and Union Square Park) are likely to be used by floating visitors.
Figure 4.5: Analysis that differentiates public parks/squares in residential areas (top) and commercial areas (bottom).

Public open spaces connected to residential properties in 1km
- More than 100 connections
- More than 50 and less than 100 connections

Public open spaces connected to commercial properties in 1km
- More than 50 connections
- More than 30 and less than 50 connections

Figure 4.6: List of public parks/squares associated with residential properties (top) and commercial properties (bottom).
4.2.2. Spatiotemporal Event Data in New York City

With one of the types of NYC open data, i.e. NYC Permitted Event Information, social activities held in open spaces in Manhattan were visualised. The data include detailed information on social events in NYC open spaces (e.g. parks, squares, small plazas, and streets) within the categories of event names, locations, and time schedules. The data were downloaded via the NYC open data portal and formatted in a comma-separated values (CSV) file (i.e. a tabular format that can be opened in Microsoft Excel). The section that includes the addresses and locations in the CSV file was converted into geolocation information (i.e. longitude and latitude). By finding geolocation, the text-based data were superimposed with geographical data MapPLUTO – Manhattan (Shapefile). With the collection of 1 month’s data (from August 2017 to September 2017), data visualisation illustrated the locations of NYC permitted events, which were classified by community districts on the map of the city (see Figure 4.7).

The comparison of the total number of events in Manhattan per month showed that the occurrences of the events peaked in July and August (see Figure 4.8). For 1 year, the highest number of events was 3,041 in August, whilst the lowest number was 733 in February. In the case of special events, as shown in Figure 4.8, approximately twice as many events occurred in the months from May to October as in the months from November to April. The lowest number of special events occurred in January and February. According to the analysis of event types, events were mostly categorised into the special event type, which took place in parks and small plazas. The number of special events constituted 69.4% of the total number of events. The number of occurrences of construction accounted for 21.7%, which constituted the second-largest frequency. The percentage of farmers’ markets and sidewalk sales was 5.1%. The remaining miscellaneous events, such as street events, parades, and athletic races, which temporarily occupied streets, accounted for 3.8%.

Moreover, based on the collection of 1 year’s data (from November 2017 to October 2018), the data visualisation attempted to depict the duration and frequency of social events in Manhattan. The data visualisation of the overall frequency of events per month showed that from January 2018 to March 2018, fewer social activities occurred in NYC open spaces in comparison to the other months, and from May 2018 to October 2018 there was a high frequency of social activities (see Figure 4.9). Because the events were organised in outdoor spaces, the frequency of events highly depended on weather conditions. Measuring the total number of events per location, based on the 1-year data, the public open space most frequently used for organising social activities was Bryant Park, located in Midtown Manhattan (see Figure 4.10). Compared to other public spaces, many more events were arranged in Bryant Park: 850 in Bryant Park, 450 in Randall’s Island Park, 400 in Madison Square Park, and 100 to 400 in multiple locations in Central Park (these numbers exempted the list of construction).
Figure 4.9.
Locations and duration of NYC permitted events in Manhattan.
Figure 4.10.
Frequency of NYC permit-ted events (from November 2017 to October 2018).
Information on social contexts in open spaces in NYC was observed by extracting, manipulating and visualising urban open data. Data visualisation and analysis presented that Bryant Park holds the highest frequency of social events among other open spaces and is one of the parks in the commercial and office area. Bryant Park is located in Midtown (district 5), which contains relatively fewer parks/squares than in other districts. In Midtown, along with Broadway, there are few open spaces such as Bryant Park, Madison Square and Union Square. The data analysis showed that those parks/squares hold a relatively high frequency of events. As they are located in the central area of Manhattan Island and the commercial and office area, the easy accessibility likely allows more uses of those parks/squares. Through the scope of data visualisation and analysis, it was observed that Bryant Park is one of the open spaces most used for organising public events, and the high sociocultural value is implicit in the park.

4.3. Design Experiment 1

4.3.1. Social Event and Weather Data in Bryant Park

It was verified that the data visualisation technique can be used as a way of observing social contexts situated in urban open spaces through location-based and time-related data. Identification of the locations (i.e. geolocation) was the key to sorting the vast amount of data, and the schedule was used to distinguish the time intervals of occurrences. Moreover, it was noticed that the frequency of outdoor events was inevitably influenced by weather conditions. To design an architectural model to be installed in an open space, weather data were also investigated. Applying data visualisation techniques, the design experiment further examined how an architectural model can sense the contexts of social and weather data in Bryant Park.

According to the list of Bryant Park events extracted from NYC Permitted Event Information, the majority of the events reoccurred in different locations in the park (see Figure 4.11, top, with the left side of the column along with event names counting the number of repeated events). The most frequently repeated event was Juggling. An event called Winter Village, which provides an ice-skating rink (replacing the lawn with the rink) and shops/bars, was the second most frequent event. In the Upper Terrace and Fountain Terrace areas, recreational types of activities were mostly arranged. In East 42nd Street Allee, the public are able to join book clubs and workshops.

Figure 4.11. Social events in Bryant Park (from November 2017 to October 2018) and weather descriptions in Midtown Manhattan (in 2016 and 2017).
Moreover, the information on weather conditions in Bryant Park was observed. The 2-year weather data (2016 and 2017) in Bryant Park were sorted by specifying the zip code of 10018 (Midtown) and extracted via a weather website (worldweatheronline.com/developer). The most repeated weather descriptions were Sunny, Clear, Overcast, and Partly cloudy (see Figure 4.11, bottom). Descriptions in relation to rain were varied, such as Light rain possible, Light rain shower, Light rain, and Moderate rain. The weather conditions in Midtown were described mainly within 35 types of text.

4.3.2. The Input of Urban Data and the Output of Architectural Behaviour

The design experiment developed an algorithm mapping the sequence from the input of text-based data to the output of the responsiveness of architectural skin (see Figure 4.12). The two tabular datasets — social event and weather data formatted in CSV files — were inserted into an algorithm (using Grasshopper software), and the texts were sorted by a set of rules. In the case of social event data, the sorting process classified the texts in the three categories of location, event name, and schedule. In the matching process, the specifically selected date and time distinguished whether an event occurred between the starting and ending times. At this stage of the design experiment, the past date and time were selected, and the input of social event and weather conditions was based on past data (i.e. the simulation was tested on 20 August 2018 using 1 August 2018 data). The two datasets were used as the input to command the performance of architectural behaviours: the social event data activated folding the skin frames and the weather data controlled the angles of rotation of the skin panels. The following describes the process of the algorithm:

1. Insert the two datasets (social event and weather data).
2. Sort the event data and by category.
3. Match with the selected date and time.
4. If matched, activate folding the skin frames. If not, do not process.
5. If activated, match with the weather data and find the given angles to rotate the skin panels.

This algorithm was further stimulated with the datasets allocated in the location of East 42nd Street Allee in Bryant Park and with the date of 1 August 2018 (see Figure 4.13). Firstly, the algorithm reads the schedule of the event in order to activate and deactivate folding the skin frames. Secondly, using weather data, the text of weather descriptions is matched with predetermined numbers that rotate the angles of the skin panels. The text describing rainy conditions implemented closing the skin panels completely. For a sunny or clear sky, the openings of the panels were maximised. The roof and the wall panels were separately tilted depending on the different weather conditions. This test model demonstrated that an algorithm could turn on and off the activation of the deployable skin frames as well as the opening of the skin panels in response to the two datasets using the CSV files. For this test model, the weather data were based on the CSV file downloaded from the following website: openweathermap.org.
Figure 4.13.
Simulation of responsive skin connected to social and weather data.
4.3.3. Real-Time Simulation of Architectural Model

For the real-time simulation, the algorithm was modified in order to connect an API (application programming interface) and read the current time (see Figure 4.14). That is to say, the algorithm directly connected the online data (instead of downloading files), and the input of the current time enabled the model to act in real time. The algorithm could automate the performance of architectural behaviours folding structures and opening skin panels with the input of online data in real time. With the digital model installing the skin systems in four different locations (East 42nd Street Allee, Fountain Terrace, Upper Terrace, Le Carrousel) on the map of Bryant Park, the real-time simulation was tested on 26 September 2018 (see Figure 4.15). The skin systems were activated by the different event schedules arranged in the four locations. Moreover, because the real-time simulation was tested in London, the time was converted to New York’s time zone, i.e. Eastern Standard Time.

To extract the real-time data — the event and weather data — two specified URLs (uniform resource locators) were typed to request Bryant Park’s online data from the servers through APIs, and the received data were sorted by the algorithm. A script in the C# component in Grasshopper extracted the weather data in a JSON format, and its input was updated every 5 minutes. Compared to the previous test model using the downloaded CSV files, which required inserting the files manually, extracting the online data through APIs facilitated the real-time simulation for the automation of the control system by means of a set of rules.

The two datasets were used in the algorithm to gain information on the currently happening social events in different locations in Bryant Park as well as on the weather conditions. The analysis of previously collected data identified the repeated descriptions of social activities and weather conditions in Bryant Park. The simulation examined whether the text-based data could promptly transform into the performances of architectural behaviours in real time. Its behaviours were mainly oriented towards controlling architectural structures and skin panels by detecting whether there are events or not and whether it is raining or not. At this stage, the design experiment mainly focused on automation of the system for the real-time performances of the architectural model. The experiment of this architectural model (Design Experiment 1) was described in the CADDRIA 2019 conference paper entitled ‘Programming Intelligent Architecture to be Responsive to Real-Time Data’.

---

**Figure 4.14.** Flowchart of the algorithm for the real-time simulation.

**Figure 4.15.** Real-time simulation (26 September 2018).
4.4. Design Experiment 2

4.4.1. Categorising the Contexts in Bryant Park Data

This research further investigated categorising the contexts in Bryant Park data for the design experiment. The classification of social contexts was based on the one-year data (from November 2017 to October 2018) and the events that were arranged in the four locations in Bryant Park (East 42nd Street Allee, Fountain Terrace, Upper Terrace, and Le Carrousel). To observe the most repeated events in the four locations, a tag cloud was generated (see Figure 4.16). The data were grouped into four types of activities: workshop/class (red), performance/show (yellow), fitness/exercise (green), and party/dance (blue). As shown in the tag cloud, the event names Piano, Celebration, and Tai Chi represent the events that occurred most repeatedly (bigger words represent events that occurred more frequently).

Figure 4.16. Tag cloud of the social events occurring in the four locations in Bryant Park (East 42nd Street Allee, Fountain Terrace, Upper Terrace, and Le Carrousel).

Figure 4.17. Data visualisation of social events in Bryant Park (depicting the frequency of events).
Figure 4.18. Data visualisation of social events in Bryant Park (depicting the frequency and duration of events).
A visualisation of the data that marked the list of events on the map of Bryant Park depicted easily readable information on the arrangement of event locations and types (see Figure 4.17 and 4.18). Arrangement of the workshop/class event type was concentrated in the location of East 42nd Street Allee. In Fountain Terrace, the fitness/exercise event type was mostly organised. The event Piano, which represented a relatively high number of occurrences, took place in Upper Terrace. In Le Carrousel, very few events, which were mainly of the performance/show type, were arranged. Other events (except for the four location events) were arranged within areas close to the 42nd street side. Events of the workshop/class type were mainly organised within the areas of Upper Terrace, East 42nd Street Allee, Upper Terrace, and Gravel Walk. Meanwhile, events of the performance/show type took place in the areas of Upper Terrace, Lawn, Stage Performance (located at the edge of the lawn area), and Le Carrousel. The fitness/exercise and party/dance event types were organised in diverse areas. This list of data (including the categorisation of event types) was reorganised in order to classify which types were mostly organised in the four locations (see Figure 4.19, top). The list presented that in the four locations there were relatively more workshop/class and performance/show events, and there were less fitness/exercise and party/dance events. Later, this list was referred to in designing the output of architectural behaviours.

Moreover, the numerical values in the four weather parameters (i.e. temperature, humidity, wind speed, and cloud cover) were divided into different levels (see Figure 4.19, bottom). According to Sasaki et al. (2000), the air temperature level can be divided into three ranges: $t < 12°C$, $12°C < t < 22°C$, and $t > 22°C$; these ranges are referred to as low-temperature, middle-temperature (comfort level) and high-temperature levels. In the case of humidity levels, an upper relative humidity level is about 80% and a low dry level is about 20% for indoor environments (as stated by the ASHRAE 55-2004), and the comfort level of humidity can vary depending on the levels of air temperature (Djamila et al., 2014). According to Baughman and Arens (1996), the optimum humidity range for indoor space is between 40% and 60%. According to the Beaufort wind force scale, wind forces of less than 11kmph are considered a light air and light breeze; meanwhile, wind forces of 12–28kmph are gentle and moderate breezes, and wind forces greater than 29kmph are fresh and strong breezes (Cullen, 2002). As per the Beaufort code, three levels of cloud cover were provided in order to describe the state of the sky: $b (0–2$ oktas) < 25%, 25% < bc (3–5 oktas) < 75%, and 75% < c (6–8 oktas); an okta is a unit used to describe the amount of cloud cover (Wolstanton Marsh Weather Station, n.d.). Referring to the aforementioned descriptions, the levels of weather parameters were divided and the records that show the weather conditions at 9 a.m. every day in Midtown in 2017 were depicted in order to observe the ranges of the weather parameters in Bryant Park.
4.4.2. Implementing the Behaviour of Kinetic Structures and Skin Panels

In this design experiment, the architectural model exhibits its behaviours with foldable kinetic frames and adjustable skin panels. The behaviours of the vertically movable structures provide different partition types that can enclose private or public social events, and the openings of the skin components adjust their angles in order to control the sunlight/shadow. The vertical folding movements were instructed to match the four types of categorised contexts of events, and the skin panels were adjusted by the levels of weather parameters (see Figure 4.20 and 4.21).

In the case of the workshop/class context, which is considered a private type, the frame structures close the inside space. Its action can limit visitors’ access to the inside of the space and block their view from the outside. Meanwhile, in the case of the party/dance context, the majority of the structures are folded so as to open the space and allow more access. As for the other two types of events, a partially closed partition offered a space in which visitors could look at a focal point (e.g. an instructor and a performer). It was considered that the contexts of events can indicate the different ways of forming the arrangement of participants.

To allow only a partial view from the inside and outside, movement of the skin panels, placed on the frame structures, was set to be less openable than the horizontal part of the panels. As for the vertical part of the skin panels, their opening adjustments are approximately half of the originally assigned values (except for wind parameters). The digital simulation showed how the vertical and horizontal parts of the skin panels were adjusted differently by weather parameters. The text of weather descriptions was used to initially distinguish whether skin panels need to be completely closed for the rain contexts. Each parameter was used to control the skin panels by summing the required amounts of openings. The different levels indicate whether or not more openings are required.

With the categorised contexts in social event and weather data, a ruleset in an algorithm was programmed to exhibit an architectural model’s reasonable behaviours. For instance, if an event is of a private type, then the kinetic structures provide more closed partitions; if an event is more of a public type, then the kinetic structures allow more access of visitors; if the weather is hot, then the skin panels are more closed in order to offer more shadow inside; if the weather is cold, then the skin panels are open in order to gain more sunlight; if the weather is windy, then the skin panels, placed on vertical frames, are more closed. Using common sense, the behaviours of structures and panels were programmed.
4.4.3. Programming Decision-Making Processes

Decision making is the process of choosing an action among possible alternative options. This experiment focused on programming an algorithm that can involve a decision-making process, setting rules in the sequence from the input of information to the output of action. There are two main flows of data processing in the algorithm: the input of social event data indicates the movements of folding structures and the input of weather data is processed in order to adjust the openings of skin panels.

The ruleset in the algorithm, which controls the frame structures, was programmed with the social event data as the following example:

1. If the current time (e.g. 13:01) is between the starting time (e.g. 13:00) and the ending time (e.g. 16:00) of an event, the algorithm starts the input of the event name.
2. If the input data of the event name state ‘BookClub’, the event belongs to the category of ‘Workshop/Class’.
3. If the category of the event is ‘Workshop/Class’, it executes the replacement of the text data to the numbers saved in the Type A file. (‘Workshop/Class’ is assigned to be Type A, and the Type A file includes the given numbers (ranging from 0 to 90) that control the rotation of the frame structures.)
4. If the data are of Type A, the numbers control the rotation of the frame structures that form an architectural model.
5. If the current time (e.g. 16:01) of the event is after the ending time (e.g. 16:00) of the event, the numbers saved in the Type A file are changed to the numbers saved in the Type E file. (When there is no event, the numbers remain in the Type E file.)

While the numbers saved in a file (e.g. Type A file) move the structures, the algorithm deals with the input of the weather data, as illustrated in the following example:

1. If the input of a text-based description of weather data states ‘Light Rain’, it belongs in the context of rain.
2. If the category of the weather description is the context of rain, the skin components, placed on the frame structures, close completely (rotation angle = 0°).
3. If the weather description (e.g. ‘Cloudy’) is not in the context of rain, the algorithm starts further calculations on the basis of the assigned rulesets with the weather parameters.
4. If the current time (e.g. 18:01) is after the sunset time (e.g. 18:00), the algorithm stops the calculation for the skin movements.

5. If the current time (e.g. 07:01) is after the sunrise time (e.g. 07:00), the calculation for the skin movements restarts.

To adjust the angle of the skin panels, the value starts with the assigned numbers according to the text of the weather description (except for raining contexts), as illustrated in the following examples: sunny (115), overcast or partly cloudy (110), and fog or mist (90). After the setting of the initial value, the algorithm transforms the numerical values in the weather parameter to the assigned numbers (see Figure 4.22, top). After finding and calculating the assigned numbers, it results in the final value (which adjusts the opening or closing of the skin components). The assigned plus and minus numbers determine the angle of rotation (ranging from 0° to 90°) of the skin panels. If the calculation result exceeds the range of 0° to 90°, it is revised to either 0° or 90°; for example, if the result of the calculation is 100, the number is revised to be 90 and it rotates the angle 90°.

The weather parameters were calculated as the following example:

1. If the description of the weather is 'Partly Cloudy', the calculation starts with the value of 110°.
2. If the temperature parameter is 10°C (i.e. cold), the abovementioned value is decreased by 21.9°.
3. If the cloudy parameter is 62% (i.e. partly cloudy), the abovementioned value is increased by 13°.
4. If the humidity parameter is 19% (i.e. dry), the abovementioned value is decreased by 10°.
5. If the wind parameter is 30kmph (i.e. windy), the abovementioned value is decreased by 18.4°.
6. The final angle of rotation is 72.7° (i.e. in the case of a cold, partly cloudy, dry, and windy day, the skin components nearly open to a rotation angle of 72.7°).

The categorised weather parameters were used to set the assigned numbers considering whether the skin components need to move towards opening or closing (see Figure 4.22, bottom). The most effective number with which to rotate the angle is the temperature parameter (ranging from -90 to 0). The ranges of other assigned numbers are illustrated as follows: humidity (from

---

Figure 4.22.
Assigned numbers for the calculation of weather parameters, described in tables (top) and graphs (bottom).
-30 to +30), cloud cover (from 0 to +20), and wind (-20 to 0). The initial and assigned numbers were set with the following considerations:

- For a sunny day (the initial number is 115) during summertime (e.g. 33°C, which indicates -69.2), the cloud cover is likely to be less than 25% due to the weather description (e.g. 1%, which indicates 0). The value with these assigned numbers is 45.8 (i.e. 115 - 69.2). For a humid condition (e.g. 85%, which indicates +22.2), it opens more in order to gain more sunlight and the output of the number for the angle is 68°. For a dry condition (e.g. 15%, which indicates -21.6), it closes more so as to provide more shadow, and the angle is 24.2°. Furthermore, if occasionally it is windy (e.g. 30kmph, which indicates -18.6), the final output is 49.4° for the humid condition and 5.6° for the dry condition.

- For a misty or foggy day (the initial number is 90) during summertime (33°C is -69.2), the cloud cover is likely to be more than 75% because of the context of the weather description (e.g. 90%, which indicates +19.6). The value with these assigned numbers is 40.4 (i.e. 90 - 69.2 + 19.6). A misty or foggy day is unlikely to be dry, but it is humid (85% is +22.2). The final angle is 62.6°. If it is windy, it closes more.

- For a sunny day (the initial number is 115) during wintertime (e.g. 1°C, which indicates 0), the cloud cover is likely to be less than 25% (e.g. 1% is 0). The value with these assigned numbers is 115. For both a humid condition (e.g. 85% is +22.2) and a dry condition (e.g. 15% is -21.6), the result of calculated numbers from the value of 115 is above 90 (137 and 93.4). The number above 90 is revised to be 90 (the final angle is 90°). For a dry condition, if it is windy, it can close more.

- For a misty or foggy day (the initial number is 90) during wintertime (1°C is 0), the cloud cover is likely to be more than 75% (e.g. 1% is 0). The value with these assigned numbers is 109.6 and it is humid (85% is +22.2). The result is 131.8. The final angle is revised to be 90°.

- While the temperature is below 10°C, the skin panel mostly maximises openings in order to allow sunlight. From the temperature above 10°C, it starts to control closing. For a sunny day (115) with a temperature of 10°C (-21.9) and cloud cover (0), the middle value is 93.1. This value can be decreased by dry or windy conditions (which indicates minus numbers). For a misty or foggy day (90) with a temperature of 10°C (-21.9) and cloud cover (+22.2), the middle value is 90.3, which can be decreased by windy conditions.

- From the temperature above 25°C, the assigned numbers indicate gradually decreasing numbers in order to block sunlight more. During a normal weather condition (e.g. normal humidity, moderate cloud cover, and calm wind), the opening ratio of skin panels can be mainly influenced by the level of temperature. The assigned numbers are set to exponentially increase or decrease the initial numbers when the weather parameters reach extreme levels.

The algorithm, embedded in the architectural model, consists of three main components (see Figure 4.23):

- The input: the current time and the updated social and weather data extracted via APIs.
- The data processing: the categorised contexts based on the previously collected social and weather data, the if–then rules that sort the updated data, and the calculation of the assigned numbers matching in weather parameters.
- The output: one of the four types of folding structures and the adjusted skin panels resulting from the calculation.

The output of the numerical values, which provide the angles of rotation, was used to render different spatial conditions in the digital model. With several simulations, the algorithm was adjusted and modified. The confirmed algorithm, which involves the abovementioned rulesets and assigned numbers, was simulated in order to observe its real-time processing.
Figure 4.23. Flowchart of the sequence of input of event and weather data, decision-making process, and output of architectural behaviours.
4.4.4. Real-Time Simulation of Digital and Physical Models

In order to verify whether the described algorithm could perform a self-regulating control system for the behaviour of the architectural model (i.e. changing the states of folding structures and adjusting the skin panels) over time, a real-time simulation was carried out and recorded. The data allocated in the location of East 42nd Street Allee in Bryant Park were inserted into the algorithm and the behaviours of an architectural model were monitored. By updating the input data, the architectural model could change its state according to the ongoing events and weather conditions in Bryant Park. In other words, when the current data were renewed, the behaviour could be altered.

The real-time simulation was carried out to observe whether a physical model could be controlled by the algorithm spontaneously through using the digital model. One of the vertically standing foldable structures (the first front one in the array in the digital model) was selected for the real-time simulation of the physical model. The results of numerical values that provide specific angles for controlling the structures and skins were sent to the actuator to move a prototypical physical model. Using a microcontroller (Arduino) and actuators (servo motors), motions of the physical model were synchronised with the simulation of the digital model (see Figure 4.24).

Whilst Grasshopper software rendered the motions of the digital model, Arduino software operated the control of the actuator to regulate the physical model. A Firefly plug-in component in Grasshopper could send the numbers (i.e. calculated values in the algorithm) to Arduino. The numbers received by Arduino were sent to the Arduino board, which was connected to a computer via a USB cable. The Arduino board transmitted the numbers to the servo motors. The servo motors received the numbers as signals for rotating the physical model’s structures and the skin panels. The servo motors were sealed within the frame in the physical model. Two mini servo motors were placed underneath the centre of the diagonal frame so as to open four skin panels.

The real-time simulation of the digital and physical models was recorded during April 2019 and from 30 June to 1 July 2019 (see Figures 4.25–4.29). Monitoring the real-time simulations eventually verified that the architectural model’s behaviours could be responsive to the input of weather and social data. On 5 and 6 April 2019 (shown in Figures 4.25 and 4.26), the architectural model could close the skins (responding to the raining context in the data) and maintain a night mode after sunset. On 30 April 2019 (shown in Figure 4.27), the model’s responsiveness to a social event was observed. On 29 June and 1 July 2019 (shown in Figures 4.28 and 4.29), the simulation showed that the digital and physical models were responsive to both weather and social data.

The real-time simulations were to demonstrate that the established algorithm could perform a self-regulating system with which to control both the digital and physical models. The algorithm could control the motions of the digital and physical models, responding to text-based data presenting the current conditions in an open space. The output of behaviours during the real-time simulation was reviewed with the input of data. In Figures 4.25–4.29, the input data were monitored while the model exhibited its behaviours. A weather application, i.e. ‘The Weather Network’, was displayed on the monitoring screen to compare with the data inserted into Grasshopper. With the texts, the application showed the image of weather conditions in the background in real time.

For different weather conditions during the recorded days, it was observed that the skin openings were adjusted. As the weather conditions in April were mostly moderate (temperature from 4°C to 19°C), the skins tended to open (shown by the digital model). For the simulation in late June and early July, due to the hot temperature (from 21°C to 32°C), the skins were mostly closed (shown by both the digital and physical models). Figures 4.25–4.29 show some moments captured during the recorded days. The recorded videos of the real-time simulations showing the behaviours of the digital and physical models during the days are included in Appendix D.
The architectural model maximised the openings of the skin panels on the roof (rotated 90°) for an overcast condition comprising a cold temperature (5°C), cloudy sky (100%), normal humidity (55%) and light wind (7kmph), and completely closed the panels for a raining context.

Figure 4.25.
Real-time simulation of the digital model (5 April 2019).

The model tilted the panels on the roof (rotated 67.2°) for a sunny condition comprising a moderate temperature (19°C), clear sky (0%), normal humidity (34%) and gentle wind (15kmph), and during the night-time it remained with a given condition, which provides partially open panels.

Figure 4.26.
Real-time simulation of the digital model (6 April 2019).
While a workshop/class type of event was taking place, the model unfolded the structures. After the sunset time, the skin panels maintained a night mode.

- For a party/dance type of event, the model folded the structures. The physical model was synchronised with the movement of the first front frame structure shown in the digital model.
While holding a workshop/class type of event, the digital and physical models completely closed the skin panels because of rain. When the rain stopped, the models reopened the skin panels.

During the design experiment, there were input texts in social and weather data that had newly appeared. The established algorithm could recognise the updated data on the basis of the previously described texts through a matching process. The NYC event dataset offered the schedule of events one month in advance (e.g. on 1 July, the open data portal provided data on the social events scheduled from 1 July to 31 July). New event names had to be manually updated to be within the four types. Moreover, in the weather data, there were very few new texts that were not included in the collection of previous data (in Figure 4.11, bottom). For the most common cases (e.g. sunny or clear sky condition), a detailed description is not required. Such new texts could be included in the list of weather descriptions and updated after the simulation. For the real-time simulation, the model was programmed to convert the input of such few unknown texts to the initial number of 60, although the model could take actions in the case of misinterpreting an unusual weather condition; otherwise the model misses the initial number required for starting the calculation. It was observed that there were few limitations in facilitating an independent operating system.

Using currently available data, the experiment investigated a possible way of implementing real-time behaviours of architectural elements. This design experiment presented that there is the potential of using data to link a digital platform and a physical space. If a digital space (e.g. website) is designed to collect and offer detailed user information such as the number of occupants or user demands, an architectural model can evolve and exhibit more specific behaviours. In addition, the aforementioned algorithm can be applied if an architectural machine is planned to be placed in a different location (e.g. Bedford Square in London — weather data can be collected and extracted via the postcode WC1B 3RA) with a website which can organise social events and provide data.

This experiment examined an architectural model’s ability of context awareness and the linkage between digital and physical models. The data directly extractable via APIs enabled the architectural model to receive updated data and exhibit its behaviours in real time. Using the previously collected data, the context in the data was categorised and the algorithm was programmed to recognise the context of the event currently happening through the use of a given ruleset. The set of if-then rules could indicate data processing by sorting the data and transforming text-based data into numbers that alter the spatial conditions rendered in the digital model. The real-time simulation showed how the architectural model exhibits its behaviours according to contexts in the data. Moreover, the architectural model could exhibit its reasonable behaviours by selecting one of the given options for a social event and calculating the numerical values in weather parameters for a weighting process.
4.5. Conclusion

The design experiment tested the simulation of both digital and physical models that perform the behaviour of architectural elements using currently available urban data. Urban data was used not only for a way in which to monitor citizens' interactions in urban open spaces, but also to programme the motions of a physical space that assists their activities. The analysis of the previously collected data allocated in urban open spaces provided information on the activities of people and the environmental conditions in a city. By identifying repeated contexts in the previous data, it was possible to anticipate which kinds of input would be inserted into an architectural model. Dividing the data into types and levels was carried out to enable the architectural model to act in relation to the real-time streaming data.

The result from urban data visualisation and analysis illustrated that in Midtown Manhattan (i.e. district 5), there are relatively fewer open spaces than other districts. Nevertheless, the analysis of event data showed that there are high frequencies of events in those spaces which are surrounded by tall buildings that are mostly commercial and office properties. Those open spaces (e.g. Union Square Park, Madison Square Park, and Bryant Park) have been used not only by nearby residents but also served as recreational areas to organise public activities. Although the data used in this design experiment were insufficient to trace how individuals visit urban open areas, they were useful in obtaining information on real events in a city. With the information that includes the contexts of events causing people to congregate in specific locations at different times, this research undertook to implement an architectural machine’s actions.

There is the potential for applying this technique in developing digital and physical spaces spontaneously. If digital spaces (or websites) are designed more to share information on location-based activities, this technique can be applied in designing varied architectural forms that exhibit real-time behaviours using geolocation. Furthermore, if digital spaces collect data with categories, it is not necessary to analyse historical data, but the data can be easily and directly used for this technique. This method suggests that digitally collected information is applicable in designing not only the software for digital spaces but also the hardware for physical spaces assisting real-time uses of a city.

Investigating a possible way of designing an intelligent system for architectural spaces, this design experiment created and developed an algorithm with regard to the terms associated with intelligence, such as self-regulating (or automation), context-awareness, and decision-making. Although the algorithm needs few assists to update new texts for social event data, it could perform tasks with most of the repeated input data. Data per se are merely symbols before transforming into information. To make data useful, it needs to be processed to give meanings. In this research, text-based data were categorised and converted into numbers to rotate an architectural model’s structures and panels. The if/then rulesets indicated the true and false statements to activate and deactivate the data processing. Through several real-time simulations of the digital model, the architectural models’ behaviours or its spatial qualities were evaluated, and consequently, the algorithm was improved. Nowadays, information is easily obtainable through online networks. Using urban data that presents real happenings in a location of a city, the design experiment tested how an architectural machine can perform its given tasks by itself to assist users of open spaces with their changing events.
Chapter 5. Design Development and Simulation: Design of Intelligent Machine
5.1. Introduction

To advance the algorithm and test its application in a different location, the central lawn area in Bryant Park was designated for design development. The data of the area, including the context of the social events in the lawn and its surrounding physical environment, were collected and observed. The information, which presented more detailed time/location-related data of the site, indicated the design direction. It identified which specific tasks can be assigned to an architectural machine that assists visitors’ uses of the area. The design development focused on dealing with two main factors: the acoustic factor for the performance events held on the lawn; and the sun/shadow factor. The previous design experiments achieved establishing an algorithm that enabled an architectural model to exhibit its automated behaviours through decision-making processing. Using a problem-solving technique based on evolutionary computation, form searching processes were carried out to find or generate a geometry that satisfies maximising sound transmission reaching far areas in the lawn area, maximising shadow covering for the summertime events, and maximising sun exposure during winter events. After the searching processes, an optimal geometry for the two factors was selected. The final selected geometry was used to simulate an architectural model’s behaviour and was programmed to be responsive to the input of social event and environmental data in real-time.

There is a difference between problem-solving and decision-making. Problem-solving is a process of creating options by identifying problems and resolving them whereas decision-making is an act of making the best choice from the known options, but they are interrelated (Hicks, 2004). Simon (1960) pointed out there are three principal phases in decision making: searching the conditions or the occasions calling for decision making; inventing, developing, and analysing possible alternative courses of action; and choosing a particular course of action from available courses. Moreover, Simon (1960, p.26) stated problem-solving techniques involves the processes of ‘noticing, searching, modifying the search direction on the basis of clues’. The surveying processes, which establishes a criteria or a goal, indicate the next stage of designing options and, subsequently, the final action can be made by a choice (Simon, 1960). In order to lead decision-making, problem-solving processes need to precede this final stage. The search for available information is essential to affect decisions. The searching processes of information are as important as the design activity as it produces alternative options. Decision making is the selection activity that indicates which one of the options is the finest among limited options. Better options provide better decision makings.

As described in this chapter, the computational techniques of problem-solving and decision making were applied to design an architectural machine. There are three main parts that describe a method of designing an intelligent machine in this chapter: the collection of data, the form-searching processes based on the use of evolutionary computation, and the establishment of an algorithm that facilitates the real-time simulation of an architectural model’s behaviours. Within Grasshopper in Rhino software, among several plug-ins, the following main tools were utilised for this design method: Snail for sound ray tracing; Ladybug for shadow casting; Wallacei for evolutionary computation. These plug-ins aided the form-searching processes through simulations. To enable the architectural model to exhibit a wider range of behaviours, the algorithm for the real-time simulation was modified. The problem-solving technique used evolutionary computation to provide computationally generated geometrical options through simulating sound and shadow factors. The algorithm that controls the actions of decision-making was updated after observing possible geometrical options that can provide suitable conditions for different situations in the site. The behaviours exhibited by the architectural model were monitored by means of real-time simulations from 27 August to 3 September 2020, from 24 October to 27 October 2020, and from 18 December to 20 December 2020. These simulations showed the daily and seasonally adapting behaviours with which to assist citizens’ activities in the lawn area.
5.2. Data Collection

The central lawn area in Bryant Park has been used not only for freely sitting/resting but also for actively organising city events in different seasons. Performance events have taken place during the summertime, and the grassy area has been replaced by an ice rink during the wintertime. Whereas small group events have been arranged in four locations (i.e. East 42nd Street Allee, Fountain Terrace, Upper Terrace, Le Carrousel), the lawn has accommodated a large number of people to hold main events in the park such as movie screenings, musical performances, sports activities, and games. Surrounded by tall buildings and exposed to the sky, the open area has been one of the outdoor recreational places in the city inviting the public.

According to the collection of NYC Permitted Event Information (in 2018 and 2019), summertime events in the lawn area were organised from the middle of June until the end of September, and wintertime events such as the so-called ‘Winter Village’ started in November and ended in April. Compared to the events in the four locations, only three types of events (except for the workshop/class type) have been organised in the lawn area during the summertime. The majority of the events were of a performance/show type, and the most frequently organised event names in 2019 were Movie Nights, Bryant Park Moves, Thursday Yoga, and Celebration (see Figure 5.1). Moreover, events in the lawn area were mostly scheduled for late afternoon and night-time (from 5–6 p.m.). The average high temperatures from June to September in NYC (from 2010 to 2019) are higher than 25°C: June (27°C), July (30°C), August (29°C), September (25°C) (New York City Temperatures: Averages by Month, n.d.). Due to the high temperature and the open condition that is exposed to sunlight during the daytime, such events were likely preferred to be scheduled for the late afternoon; film screenings and stage performances were arranged during the night-time for lighting effects.

Using geolocation information on NYC latitude and longitude (40.78° N, 73.97° W), this research simulated the shadow casting on Bryant Park with its surrounding buildings (see Figure 5.2). In 2019, the hottest day was 20 July and the coldest day was 21 January. To observe shadow casting in the extreme cases of temperatures during the summertime and wintertime, the aforementioned days were selected for the simulation. Although in the real world such shadows can be observable only with a clear sky (as sunlight is usually dispersed by cloud conditions), it shows the directions of sunlight in different seasons. The simulation for the summertime shows that during morning time the park is mostly exposed to sunlight and during afternoon time the area is partly covered by shadows as a result of the tall buildings located on the west side of the park. During the wintertime, due to the low angle of sunlight and the surrounding buildings, the park can be fully covered by shadows during both morning and afternoon time.

Through the collection of the social event data and the environmental data, the time-related condition of the lawn area was identified: the large open area has been used to hold performance types of events mostly scheduled for the late afternoon during the summertime, and the exposed area can be mostly covered by shadows due to the surrounding buildings during the wintertime. The simulation of shadow casting on 20 July and 21 January (the extreme cases of temperatures) presented that the thermal comfort within the lawn area can be highly influenced by exposing sunlight during the summertime and covering shadows during the wintertime.

The observation of the data indicated the design direction of architectural space for the lawn area. The majority of the lawn events belong in the category of a performance/show type. A temporary stage has been installed at the end of the lawn for summertime events such as musical performances, accordion festivals, and movie screenings. To design an architectural model that assists such events, the research focused on dealing with the acoustic factor. For the summertime events, which have been mostly arranged during late afternoon or night-time, the control of sun/shadow is a secondary factor. Nevertheless, for the wintertime events, when the park is actively used for an ice rink and covered by shadows, maximising sun exposure was a critical factor. With regard to the contexts of events and the site condition, a volume study of architectural space on the central lawn was carried out.
Figure 5.1. Tag cloud that shows the most frequently organised social events (top) and the schedules of repeated events (bottom) in the lawn area in 2019.

Figure 5.2. Shadow casting on Bryant Park with NYC geolocation (latitude 40.78° N and longitude 73.97° W) on 21 January and 20 July (from 9 a.m. to 4 p.m.).
5.3. Acoustic Simulation

The acoustic study examined geometrical volumes that can equally distribute sound transmissions to audiences for stage performances within the lawn area. With the dimensions of the area (about 54m x 85m), it simulated sound ray tracing radiated from a sound source placed at the end of the area (where the stage has been installed). The simulation focused on widely spreading sound reflections through an architectural shell. The sound model was first tested with manually deformed, enclosed spaces. Thereafter, using evolutionary computation, a base form was manipulated by an automated searching process, generating possible outcomes with which to manage sound ray tracing.

In free-field conditions (i.e. outdoor spaces in which there are no sound reflections), sound waves from a point source can expand outwards and their virtual form can be illustrated as a series of spherical spreading (Egan, 2007). The decibel (dB), named after Alexander Graham Bell, has been used as a logarithmic unit (which is based on a ratio measurement of energies or powers) to express the level of sound (Barron, 1993). Commonly, sound above 80dB is considered to be at a very high level; meanwhile, sound in the range of 40dB to 60dB is considered to be at a moderate level (e.g. the sound of office activities is in the region of 50dB) (Egan, 2007). Musical instruments within orchestral performances can produce high decibel levels ranging from 80dB to 110dB (Phillips and Mace, 2008). According to the inverse-square law (which applies to the principle of sound spreading), every doubling of the distance from a sound source decreases sound intensity by 6dB. The formula for the law is illustrated as follows: $\text{dB(D2)} = \text{dB(D1)} - 20 \cdot \log_{10}(D2/D1)$, where $\text{dB(D1)} = \text{the decibel at the initial location}$, $\text{dB(D2)} = \text{the decibel at the new location}$, $D1 = \text{the distance from the sound source to the initial location}$, and $D2 = \text{the distance from the sound source to the new location}$.

The calculation applying the inverse-square law within the dimensions of the long side of the lawn area and the sound level of musical instruments resulted in the following: if the sound intensity located within 1m of a sound source is 100dB, 61.41dB can be measured at a distance of 85m; at a distance of 42.5m (the middle of the lawn), 67.43dB can be measured; at a distance of 20m, the sound level is 73.98dB; at a distance of 10m, the sound level is 80dB. It represented that at a distance of 20m to 85m, radiated from a point source with musical instruments, the range of the sound level is that of 61.41dB to 73.98dB, whereas distances within 10m can maintain 80dB to 100dB, which is considered to be a high sound level. However, it is ideal only with an anechoic space and an open condition free from any obstructions. In outdoor conditions in the real world, noises can be intervened by surrounding conditions.

Reverberation — i.e. the persistence of sound after producing a sound mostly caused by reflections — is one of the most important factors for acoustic spaces. Reverberation is generally considered disadvantageous for the clarity of speech-based sounds, albeit advantageous for the quality of musical sounds (Kinoshita, 2014). Within enclosed spaces, a sound can be reflected until it loses its energy absorbed by surface materials (Barron, 2010). Following the law of reflection, the angle of reflection of a sound wave equals the angle of incidence, as if it produces a mirror image of the incidence originating from the opposite side of the reflective surface (see Figure 5.3, left). While a sound wave continues to bound around in a room, a receiver can obtain the sound through direct sound, early reflections and late reflections (see Figure 5.3, right).

In large halls, sound reflections (from side walls, the ceiling, etc.) arrive at a receiver after direct sound, and sound transmissions after a series of early reflections (about 100ms) are measured as reverent sound, whose duration is described by the reverberation time (Barron, 2010) (see Figure 5.4, left). During sound reflections providing the resonance of sound, the sound level (dB) decreases within reverberation time (see Figure 5.4, right). According to Barron (2010), if reverberation time is too short, sound can be too stark, and sound with reverberation time that is too long can be undetectable or masked by earlier sound. Approximately 1 second for speech and 2 seconds for symphonic music are considered the optimum reverberation times, and for the design of multipurpose halls, managing the reverberation times for both types of sounds is a dilemma (Barron, 2010). For music sounds, the range of optimum reverberation time is that of 1.2 to 2.6 seconds (Newman, 1974). The recommended reverberation times for music sounds are described as follows: above 2.5 seconds for organ music, 1.8–2.2 seconds for Romantic/Classical music, and 1.3–1.8 seconds for opera (Barron, 2010).
To measure the approximate reverberation time related to the volume of a room and the total amount of sound absorption, the Sabine formula has been used, which is described as follows: $RT_e = \frac{(0.16s/m) V}{S_e}$ where $RT_e$ is the corresponding reverberation times (seconds), $V$ is the volume of acoustic space (length x width x height) (m$^3$), and $S_e$ is the effective absorbing area (m$^2$). The surface area ($S_e$) is calculated as follows: $S_e = \sum (\alpha i x si)$, where $\alpha i$ constitutes the absorbing areas and $si$ constitutes the sound absorption coefficients of those surface areas. While sound travels in a room, its energy is absorbed by surface materials. Soft, porous or pliable materials (such as cloth) can reduce sound via the vibration of the air contained in the pores of the materials, whilst hard, dense or impenetrable materials (such as metal) can less absorb and more reflect sound. Moreover, the audience in a room is one of the major factors that can absorb sound.

Absorption coefficient values are highly related to material properties and specific frequencies (Hz), and range from 0 to 1, where a coefficient of 0 represents that none of the sound is absorbed, while a coefficient of 1 indicates that all of the sound is absorbed. Within 500Hz (usually 500Hz or 1,000Hz is measured with the Sabine formula), 1.5 persons/m$^2$ (for the audience) indicates a coefficient of 0.71, while normal, flat materials (such as thin plywood panelling or plasterboard ceiling) have a coefficient value of 0.1 (Vorlander, 2008, pp. 307–309). To decrease reverberation time in a room, materials with a high coefficient value (close to 1) can be used. In other words, in the Sabine formula, the higher the coefficient value, the shorter the reverberation time. Normally, materials with a high absorption coefficient are useful in reducing sound reflections (or echoes) in a room.

To determine a volume relative to the dimensions of the lawn area (85m length and 54m width) that is suitable for the recommended reverberation time for music sound, a calculation of the Sabine formula was carried out. The calculation was based on the heights of 10m, 15m and 20m and in relation to the usage of a hard surface material (such as metal), which has a low coefficient value, giving an absorption coefficient of 0.1 and a ground absorption coefficient of 0.71 (1.5 persons/m$^2$) within 500Hz. The following describes the calculated values for the height of 10m: the volume with 85m x 54m x 10m = 45,900m$^3$, with the ground coefficient value of 0.71 and the ceiling and four walls coefficient value of 0.1, $Se = (0.71 x 4,590m^2) + (0.1 x 2,780m^2) = 3,258.9 + 278 = 3,536.9$ sabins; $RT_{e,10m} = 0.16 (s/m) x 45,900m^3 / 3,536.9$ sabins = 1.83 seconds. For the height of 15m, $RT_{e,15m}$ is 2.66 seconds, and for the height of 20m, it is 3.4 seconds. Within the aforementioned condition, a height in the range of 10m to 15m is appropriate, which provides a reverberation time in the range of 1.83 to 2.66 seconds.

However, the results (i.e. $RT_{e,10m}$) can be varied by different coefficient values and the range of Hz. For instance, if surface materials that have a higher coefficient (above 0.1 and close to 1), such as thin or perforated panels, are used, the reverberation time can be shorter; meanwhile, if the audience constitutes less than 1.5 persons/m$^2$ (not fully sitting within the lawn area), the reverberation time can be longer. Although the aforementioned calculation with the Sabine formula could predict only an approximate reverberation time, it confirmed that in a volume with a height of 10m or 15m for the large lawn area, sound reflections can be maintained during the recommended reverberation time for music sound through the use of a flat or plane surface material.

The simulation of manually deformed, enclosed spaces (whose height mostly ranges from 10m to 15m) with sound ray tracing was tested in order to observe the effect of different ceiling and wall conditions upon sound reflections (see Figure 5.5). The simulation of ray tracing projected sound paths from a point source showing maximum 4$^{th}$ bounces (sound reflections) that reached the ground. The colours of the ground presented the high number of reflected sounds that reached the ground as red, the middle number as yellow, and the low number as blue. Simulating random sound paths, the test model showed how different shapes and volumes affect sound spreading to the ground.

The test models presented that both ceiling and wall manipulations are necessary in order to manage sound rays, although they could show only a few results of sound reflection simulations with different geometries. The first row in Figure 5.5 (showing simple geometries with a height of 10m to 20m) presented that flat ceilings are ineffective in spreading sound to distances that are far from a sound source. The following test (described in the second row) examined how fluctuated shapes of ceilings distribute sound reflections; through ceiling manipulations, sound could be spread to side areas but not areas far from the sound source. The test continued to control side (or wall) surfaces. It resulted that the geometry, narrowing the front part of the wall and widening the second front part of the ceiling, can cause relatively wide-spreading sound reflections.
5.4. Form Searching through Evolutionary Computation

5.4.1. Genetic Algorithms in Architecture

The volume study for the lawn area continued with the use of a problem-solving technique based on evolutionary computation. In many fields, genetic algorithms (GAs) have been variously employed as a search and optimisation technique. Meanwhile, for architectural designs, GAs have been adopted in computer-aided tools that find possible geometric forms simulated with given spatial conditions. The utilisation of such tools has been advantageous in generating alternative options beyond what designers can manually produce within a short period. GAs in architecture have provided a design method in which to explore computationally created, novel forms and analytical outcomes for functional factors (e.g. for structural, lighting, thermal and acoustic conditions), and the involvement of cultural and social notions in implementing encoding rulesets is a challenge with regard to using such techniques or tools (Fasoulaki, 2007). With consideration given to which spatial types are suitable for citizens’ changing uses of the lawn area, the technique was utilised to find possible spatial qualities and select an optimal option.

In GA-based computing environments, based on the biologically inspired operators such as selection, crossover, and mutation, the algorithm produces and reproduces offspring through generations while searching for the fittest individuals. In biology, a chromosome — a DNA molecule which contains genes — is considered as the fundamental element that influences individuals’ innate or inherited traits. In the techniques based on GAs, the chromosome (i.e. the basic element) is typically the form of bit strings (commonly binary numbers: 0 and 1), and the gene (i.e. the numbers) that characterises the outcomes can be manipulated by those operators to generate a population (Mitchell, 1998). Starting with randomly generated chromosomes, the selection operator chooses two chromosomes for latter bleeding; the crossover operator picks a part of the bit string and exchanges the subsequence (e.g. 10000100 and 11111111 are replaced with 10011111 and 11100010, where the underlined part is crossed over). The mutation operator changes some of the numbers (e.g. 00000000 is changed into 01000000, where the underlined number is flipped). During the process, the algorithm evaluates the objective function value (also called fitness function score and fitness value) associated with each chromosome and selects the fitter chromosomes for the reproduction of the next generation. The fitter chromosomes (the chromosomes that indicate fitter outcomes) are selected through the evaluation of the calculated fitness values (Mitchell, 1998).
The phrase ‘survival of the fittest’ or ‘survival of the form that will leave the most copies of itself in successive generations’ is one of the key principles to understand the processes of searching for better solutions in GA-based computing environments (Herbert Spencer first used the phrases to describe the Darwinian evolutionary theory). In GAs, the genetic package (the combination of chromosomes) is called a genotype, and it can be decoded to output readable outcomes such as a parameter set or a solution alternative, which is called a phenotype (Goldberg, 1989). Through generations (one iteration of the process driven by the operators is a generation), the operators can hand over the numbers in fitter chromosomes. Consequently, the highest fitness value, that is, the number that most closely satisfies the objective, can be ranked to identify the fittest individual (i.e. the phenotype that is the most optimised).

In the architectural tools using GAs, the number that determines the positions of points in a 3D space or an XYZ coordinate system can be referred to as a genotype, whereas a geometry composed by those points is a phenotype. For the implementation of form-searching, design problems (with measurable factors) can be defined in between genotype-phenotype mappings. For example, to search geometries or design a building that is optimised for thermal performance in a hot climate zone, minimising sun penetration into the interior space can be defined as the objective. The measurement of the area exposed to the sun or covered by shadows can be used to evaluate which geometries are fitter to achieve the objective. With such measured numerical values, the algorithm with evolutionary computation can generate geometrical variations (built upon altered point positions) towards better solutions and can result in fitter geometries.

5.4.2. Optimisation of Acoustic Space

For the design development in this research, form-searching through evolutionary computation was carried out with the acoustic factor. The early searching process was simulated with one objective: maximising widely spreading sound reflections through an architectural shell. To count the number of virtually projected random sound reflections that reached the ground, the lawn was divided into three areas as followings: Area 1: one-quarter of the lawn, which is the farthest area from a sound source; Area 2: half of the lawn that includes Area 1; Area 3: three-quarters of the lawn that includes Area 1 and 2. In the algorithm, with these three divided areas, three fitness criteria (i.e. maximising the number for each area) were given to the searching process. The algorithm was set to generate possible geometries that maximise sound reflections reaching areas far from a sound source.

Before running the algorithm using evolutionary computation, the primitive form was made based on a vaulted volume. The height of 14m was determined to build the base form (referred to in the previously described acoustic study). The range of point position was set as follows (see Figure 5.6): the corner points on the ground (A1 & A5) were set to move 6m inward on the X and Y axes, and the points between the corners on the ground were 3m movable in the X direction and 3m inward on the Y axes; the points on the second row from the ground (from B1 to B5) were 3m moveable on the X and Y axes and 3m for upwards on the Z axis; the points arranged in the centre line (from E1 to E5) were set to move 3m on only the X and Z axes; the other points from C1 to C5 and D1 to D5 were 3m moveable on all XYZ axes. By moving the points arranged on one side of the volume and mirroring it to the opposite side, the base form could be deferred. There were two reasons to generate symmetric forms: one was to generate geometries fit to the symmetric layout of Bryant Park; the second was that the computationally generated geometries could be unnecessarily affected by arbitrarily radiated sound ray tracings which were virtually simulated but projected unequally for two sides. In Figure 5.6., the top left image shows the simulation of sound ray tracing with the vaulted volume (i.e. the base form).
From the base form, the algorithm generated alternative solutions of geometries for the acoustic factor. In the automated searching process, there are three main sets of algorithms:

- The first set designs a geometry based on points that build lines and planes.
- The second set simulates sound ray tracing.
- The evolutional engine (i.e. the algorithm built upon evolutionary computation) that moves the point positions and evaluates the result of the simulation, searches and generates geometries through generations.

The whole set of the searching algorithm executes five main procedures illustrated as follows (see Figure 5.7):

1. Arranging the point positions that manipulate the surfaces of the volume.
2. Simulating ray tracing and counting the number of reflected ray tracings in area 1, 2 and 3.
3. Evaluating the counted numbers (used as a fitness value) and selecting better point positions through the genetic operators.
4. Producing individuals (geometries) through a generation.
5. Redoing the first step and reproducing the next generation.

After running the evolutional engine, all searched geometries, which were saved during the iterations, could be collected, observed, and compared with their fitness values (see Figure 5.8). 25 individuals and 50 generations (total 1,250 individuals) were set to be searched by the evolutionary computation, and 1000 ray tracings with one sound source were set to simulate. As the fitter individuals were more frequently selected through generations, there were more geometries in the latter generations that were similar or the same. Meanwhile, in the earlier generations, more varied geometries were produced. The numbers of the reflected ray tracing that were counted in the divided areas ranged as follows: Area 1: 9 - 119; Area 2: 45 – 268; Area 3: 171 – 591. In Figure 5.9, the parallel coordinate plot graphically illustrated all searched individual’s fitness values (the fluctuating lines present the searched geometries’ fitness values) and the more frequently selected fitness values (the more frequently selected individual’s fitness values to present the longer horizontal lines).
Figure 5.8. Outcomes of form-searching through evolutionary computation, processed with the fitness objective of maximising sound reflection (generation 1, 20, and 49 were rendered).
Through ranking the outcomes based on ordering the fitness values, the fittest geometry that satisfies the objective could be selected (see Figure 5.10). It was Generation 49, Individual 5. Its fitness values were the highest for Area 1 (119) and 2 (268), and relatively high for Area 3 (579). In Figure 5.10, the graph of standard deviation and mean value trendlines showed how the variation and the average of 25 individuals’ fitness values were changed towards optimisation through generations (the X-axis is the number of generation and the Y-axis is the fitness values). The standard deviation trendline illustrated that after about ten generations the variation of fitness values was reduced. The mean value trendline showed the progress of increasing fitness through generations. Moreover, the simulation of sound ray tracing using the fittest geometry was visualised to verify that the volume could adequately distribute the reflections (see Figure 5.10, top right).

For the further searching process, including multiple objectives, a base form was modified. To design folding structures, a stipe-based geometry was used as the base form. Twenty frames were arrayed on the previously generated volume (i.e. the fittest geometry resulting from the abovementioned searching process). From this base form, 25 individuals in 50 generations were generated through evolutionary computation (see Figure 5.11). It was considered that sound amplifiers (speakers) would be placed on two sides at the stage. For the simulation of sound
Figure 5.11.
Outcomes of form-searching through evolutionary computation, processed with strip-based form and the fitness objective of maximising sound reflection using three sound sources (generation 1, 20 and 49 were rendered).
ray tracing, two additional sound sources were included in the searching algorithm. With three sound sources (projecting 500 ray tracings for each) and a stripe-based volume, the second round of the searching process was carried out. It resulted in the outcomes with the fittest values in the range of 11 – 52 in Area 1, 63 – 159 in Area 2 and 297 – 439 in Area 3 (see Figure 5.12).

The geometry that presents the fitness value 50 for Area 1, 154 for Area 2 and 439 for Area 3 was selected as the fittest outcome (see Figure 5.13). To compare the difference between the outcomes resulting in one sound source and three sources, an additional searching process using one sound source was carried out (see Figure 5.14). The two selected fitness geometries were similar. The frames located on the front and back sides were slightly altered. This comparison demonstrated that the number of sound sources is barely influential to change the outcome of geometry and its fitness values. For the next searching process, which includes additional fitness objectives, the geometry resulted by using three sound sources was used as a base form.
5.4.3. Multi-objective Optimisation

For the last round of the searching process, three more fitness criteria (minimising shadows during the wintertime, maximising shadows during the summertime, and minimising the length of structures) were added in the algorithm. Figure 5.15 shows the simulations that were involved in the algorithm to carry out six fitness criteria. For the evaluation process of shadow casting simulation, 10,000 points were arranged in the lawn area, and the areas covered by shadows were counted with those points. For the wintertime simulation, folded structures were simulated with shadow casting for 10 a.m., 12 p.m., 2 p.m. and 4 p.m. on 21 January. For the summertime simulation, shadows were cast by unfolded structures for 10 a.m., 12 p.m., 2 p.m. and 4 p.m. on 20 July. For the last fitness value, the total length of the 20 frames was measured. While the evolitional engine searched the fitter volumes through changing point potions, the two sets of algorithms involving folded and unfolded structures were used to simulate shadow casting for different seasons (see Figure 5.16).

6 Fitness Criteria

1. Maxmise sound refection (for Area 1)
2. Maxmise sound refection (for Area 2)
3. Maxmise sound refection (for Area 3)
4. Minimise shadow with folded structures (for wintertime)
5. Maxmise shadow with unfolded structures (for summertime)
6. Minimise the length of folding structures

As a result of producing 25 individuals in 50 generations with the six fitness criteria, six options for architectural volumes were selected based on the highest rank of each fitness criterion (see Figure 5.17). The diamond chart in Figure 5.17 showed the selected geometries’ rankings for each fitness criterion (the higher rank is closer to the centre point in the diamond chart). Among the selected six outcomes, the geometry, that was identified as generation 19 and individual 6, which was selected by the highest rank for maximising sound reflection for Area 3, satisfied most of the six criteria. Thus, this geometry was selected as the fittest.
Figure 5.16. Scheme of form-searching through evolutionary computation for six fitness criteria.
Figure 5.17. Selected outcomes based on the highest rank for each criterion.

Figure 5.18. Selection of generation 19 and individual 6, described in the fitness value graphs.
The finally selected geometry (i.e. the outcome of generation 19 and individual 6) was the result of compromising different fitness criteria. As illustrated in the parallel coordinate plot in Figure 5.18, the selected one’s fitness values were mostly close to the highest number although the fitness objective of minimising the total length of structures was relatively less satisfied. In the previous searching processes, three criteria were given by designating three areas, but essentially the previous algorithms searched geometries for the single objective of maximising sound reflections. In this searching process, the mean value trendline showed more fluctuated or unstable graphs compared to the previous searching processes. The trendline of the progress of fitness criteria 1 (top left in the mean value trendline graph section) showed that the early generations presented high fitness values (observed by the mean value), but the values in later generations were less satisfactory for fitness criteria 1. The algorithm modified a base volume that was the result of the objective of maximising sound reflection, and generated geometries to satisfy the additional objectives with shadow casting simulations. Among other individuals in generation 19, individual 6 presented the highest fitness values for the criteria 1, 2 and 3 (marked in the bottom right in Figure 5.18). The form searching processes through evolutionary computation provided alternative options of geometries to deal with specific conditions (i.e. acoustic and shadow factors) in the site, and the best option that satisfied the majority of the six fitness criteria was selected.

5.5. Behaviour Setting

5.5.1. Categorisation of Behaviour

Based on the finally selected geometry, an architectural model’s possible behaviours were further examined. Through visualising the simulations that were used for the abovementioned searching process, the main behaviours (i.e. folding and unfolding structures and opening skins) of the selected geometry were evaluated (see Figure 5.19). The folded structures could adequately allow sunlight to the lawn area through the low sun angle for the wintertime (simulated with the date of 21 January); the unfolded structures could provide nearly full shadow for the summertime (simulated with the date of 20 July); by closing the skins with unfolded structures, the volume could widely spread the ray tracing. After examining various types of folding structures with their shadow casting for summertime, the architectural model’s behaviours were classified with the categorised social events (see Figure 5.20 and Figure 5.21). The observation of 2019 data demonstrated that there were three sorts of events held in the lawn area: performance/show, fitness/exercise, and party/dance. These categorised social event types were matched with varied spatial qualities.
Figure 5.20. Behaviour setting (A to C).

Type A & B
A. Closed/Intro
B. Musical
BB. Musical
BB. Movie

Type C
C. Fitness

Type D
D. Celebration

Type D
D. Dance

Type D
D. Dance

Type E
E. None

Figure 5.21. Behaviour setting (D to E).
In the algorithm that controls the architectural model’s behaviours acting with input data, the types from A to D were featured as follows:

- **Type A**: The algorithm completely closes both skins and structures when the input of environmental data presents raining contexts.
- **Type B**: This type indicates the same action as type A when the input of social event data belongs to performance/show type. It switches the input of the event name related to an acoustic factor to one of the following symbols: B_Music, BB_Musical and BBB_Movie. With these symbols, it executes the action.
- **Type C**: This type mostly closes the vertical structures (the first row of the arrayed frames from the ground) to limit visitors’ access when the input is fitness/exercise type and it opens/closes the rest of structures through identifying specific event types, illustrated as follows. When the algorithm changes the event names to C_Fitness, it opens the horizontally arranged structures to expose the sky and allow sun exposure (the environmental data is not effective while folding the structures). In the case of CC_Game, it closes most of the structures, so the area can be less affected by changes in the sky’s condition. For CCC_Exercise, it closes most of the top structures and opens the side structures to limit sun exposure.
- **Type D**: This type partially or completely opens the vertical structures to allow visitors’ access and it folds and unfolds the rest of the structures to provide partially distributed shadows. In the cases of D_Celebration and DD_Dance, for randomly located visitors or their preferred areas (to be under shadows or not), it controls the structures to provide partly or half-covered shadows. For DDD_Picnic, it indicates all folded vertical structures and mostly unfolded horizontal structures (zigzag folded on the edges).
- **Type E**: This type stays with all folded vertical structures and all unfolded horizontal structures when none of the social events is happening. This behaviour maintains a moderate and high temperature; the horizontal structures can be folded for low temperature (the detail is described in the next section).

The state of the skins, under the condition of the folded structures, is fixed to maintain closing during the events, while the skins that are arranged along the unfolded structures are constantly movable with the input of the environmental data. While folding the structures, the algorithm controls the motion of skin openings to decrease their angles to avoid conflicting with each other. Types B, C and D, with fixed folded structures, were designed for summertime events (according to the 2019 data, the lawn events were held from 16 May to 28 September). From October to April, the architectural model stays with type E that maintains the folded state of vertical structures and allows the folding and unfolding of the rest of the structures.

### 5.5.2. Threshold Rule

Compared to the previous model that was developed in the design experiments (described in Chapter 4), the range of the architectural model’s behaviour was extended to control the horizontally arranged folding structures. The previous model could not sufficiently expose sunlight only through skin openings during the wintertime. After the evaluation, this model was designed to gradually fold structures based on a threshold rule (see Figure 5.22). The rule is that the algorithm indicates specific angles such as 20°, 40°, 60° and 80° that fold the structures when the result of the calculation of the pre-given increasing or decreasing numbers allocated along with weather parameters reaches numbers such as 90, 120, 140 and 160. In the case of the previous model, the skins could move with the angle within the range from 0° (indicating completely closing the skins) to 90° (for completely opening), and when the result of the calculation was above 90, the skins stayed with 90°. For this model, when the result of the calculation that controls skins is above 90, the algorithm starts to fold the structures.

![Figure 5.22.](image_url)

The state of folding structures controllable by a threshold rule.
Figure 5.23. The updated version of assigned numerical values for the calculation of weather parameters (the temperature part was modified).

Figure 5.24. The threshold rule.

The architectural model’s behaviour was programmed to act with the result of calculation of the assigned numbers allocated with weather parameters (shown in Figure 5.23) and the threshold rule (shown in Figure 5.24), illustrated as follows: where $x$ is the result of the calculation and $y$ is the angle of folding structures; if $90 < x < 121$, $y = 20^\circ$; if $120 < x < 141$, $y = 40^\circ$; if $140 < x < 161$, $y = 60^\circ$; if $160 < x$, $y = 80^\circ$. In the algorithm, the threshold rule was applied in the following examples:
Example 1: The input of the environmental data is inserted as follows: the weather description ‘Sunny’, temperature 5, cloud cover 0, humidity 50, wind speed 0, from the initial number 115 (allocated by the text ‘Sunny’), the calculation of 115 + 72 + 0 + 0 + 0 results in 187. When the result is 187 (above 160), the angle of folding structures is 80° (fully folded state).

Example 2: With the weather description ‘Sunny’, temperature 10, cloud cover 0, humidity 50, wind speed 0, from the initial number 115, the calculation of 115 + 0 + 0 + 0 + 0 results in 115. When the result is 115 (which is above 90 and below 121), the angle of folding structures is 20°.

Example 3: With the weather description ‘Sunny’, temperature 22, cloud cover 0, humidity 50, wind speed 0, from the initial number 115, the calculation 115 - 26 + 0 + 0 + 0 results in 89. When the result is 89 (which is below 90), the angle of folding structures is 0° and the skin angle is 89°.

Example 4: With the weather description ‘Sunny’, temperature 30, cloud cover 0, humidity 50, wind speed 0, from the initial number 115, the calculation 115 - 85 + 0 + 0 + 0 results in 30. When the result is 30 (which is below 90), the angle of folding structures is 0° and the skin angle is 30°.

Example 5: With the weather description ‘Fog’, temperature 30, cloud cover 85, humidity 90, wind speed 0, from the initial number 90, the calculation 90 - 85 + 19 + 24 + 0 results in 48. When the result is 48 (which is below 90), the angle of folding structures is 0° and the skin angle is 48°.

The temperature parameter is the main factor to control folding structures through seasons. For temperatures above 22° C (under the condition of mild weather), the skins that are arranged on unfolded structures can be moveable through the other parameters (cloud cover, humidity, and wind speed) in a day, whilst for temperatures below 5° C (during the wintertime), the model maintains the fixed state of folded structures.

### 5.6. Algorithm Development

The updated algorithm was established to manage the intertwined paths in the sequence from the two datasets to the architectural behaviours. In the previous algorithm (used in the design experiments), there were two separated paths: from the event data to the folding structures and from environmental data to skin openings. In this algorithm, the motions of folding structures and opening skins were controllable with both datasets. To avoid the conflict of data processing from two datasets, additional if/then rules were programmed in the updated algorithm. There are three main questions to recognise the current conditions in the lawn area, and the algorithm executes the following procedure:

1. Is the social event currently happening? If yes, choose a specific type from alternative options in types B, C, and D. If not, stay with type E.
2. Is it currently raining? If yes, choose type A. If not, stay with one of the above-mentioned types.
3. Is it currently daytime? If yes, stay with the type. If not, change to the night mode.
4. If type A is chosen, replace type B, C, and D to type A.
5. If it is night-time, change type E to night mode, but do not change type B, C and D to night mode.
6. If it is raining during the night-time, change the night mode to type A.
7. If the result of the calculation for adjusting skins is above 90, fold the structures according to the threshold rule.
8. If it is night-time, unfold the structures to be in night mode.

In addition, since July 2020, real-time data has not been offered via the portal (as all of the events in NYC were cancelled due to the COVID-19 pandemic). Virtual social event data (based on 2019 data) was made and used to carry out the real-time simulation. Instead of using the URL address to download the list of events via an API, in the updated algorithm, a virtual list, which copied the 2019 data CSV format, was inserted as the input (see Figure 5.25). For the test of the real-time simulation in 2020, the design of the algorithm was modified and updated (see Figure 5.26). When the event is reorganised in the lawn area after 2020, the input can be replaced to the URL address (described in Chapter 3) to read the CSV format.

Figure 5.25. Virtual data of social events in 2020 used for real-time simulation.
Figure 5.26. Flowchart of the updated algorithm (used for the real-time simulation in 2020).
5.7. Real-Time Simulation

5.7.1. From 27 August to 3 September 2020

To test the algorithm and monitor the behaviours of the architectural model, a real-time simulation was carried out from 27 August to 3 September 2020 (recorded videos of the simulation are included in Appendix D). The noticeable moments with respect to the behaviours of the simulation were captured and shown in Figures 5.27–5.38. The aforementioned procedure, including the if-then rules, was added when misleading behaviours were observed. The simulation was to verify whether the algorithm can facilitate an architectural model’s self-regulating system, indicating proper behaviours for the right moments. The schedules in the social event data enabled the architectural model to turn on/off the activation of selecting a behaviour among alternative options, and the environmental data indicated adjusting its behaviours that control shadow casting.

In the virtual social event data, the schedule of time remained as the 2019 data, but the date was modified (e.g. the schedule of the Thursday Yoga event from 18:00-19:00 on 29 August described in the 2019 data was replaced to the schedule of the Thursday Yoga event from 18:00-19:00 on 27 August in 2020). The modified schedule of the social event in the data was listed as follows:

1. 27 August (Thurs.): Thursday Yoga (18:00-19:00)
2. 28 August (Fri.): Celebration (17:00-22:00)
3. 29 August (Sat.): Bryant Park Moves (10:00-11:00), Shakespeare - Othello (19:00-21:00)
4. 30 August (Sun.): Wellness Festival (9:00-16:00)
5. 31 August (Mon.): Movie Nights (17:00-22:00)
6. 1 September (Tues.): Picnic (14:00-16:30)
7. 2 September (Wed.): Kubb Tournaments (18:00-21:00)
8. 3 September (Thurs.): Broadway in Bryant Park (12:30-13:30), Thursday Yoga (18:00-19:00)

For the duration of eight days, the temperature ranged from 17°C to 34°C during the daytime, according to data from worldweatheronline.com. The weather descriptions and parameters from two data sources, as shown on the upper and lower left side of the image of the simulation, were occasionally different, although the same zip code (i.e. 10018) was designated for the location of both sources. The model was programmed to be responsive to the data shown on the top note (the data were extracted from worldweatheronline.com); the bottom data (presented by an application of the Weather Network) were displayed to compare with the text-based data. For the late summertime, the architectural model tended to close the skins with high temperatures. During the simulation days, it often rained; therefore, the range of temperatures was relatively wide. With the frequently occurring rainy conditions, the model often closed the structures. The following describes the observation of the architectural model’s behaviours performed during the real-time simulation.
• After raining that caused the high humidity (98%), the architectural model folded the horizontal structures with a mild temperature (19°C).

• With a relatively high temperature (29°C) and normal humidity (63%), the angle of the skin was 34.9°, which was a closed state to provide more shadows to the lawn area.

• During the event, the model featured the C_Fitness type. With the high temperature (34°C), the skins arranged on the unfolded structures were closed.

• After sunset, the model remained in night mode.
• With the slightly high humidity (74%) and the mild temperature (23°C), the model folded the structures to be at a 20° angle and the skins maximised the openings.

• With the high temperature (30°C), partly cloudy conditions (50%), and gentle wind (17kmph), the skins were mostly closed at a 14.1° angle.

• During the event, the model featured the D_Celebration type. With the high temperature (28°C), the skins were mostly closed.

• During the event, due to rain, the model completely unfolded the structures and closed the skins, as indicated by the A_Close type.
• With full cloud cover (100%) and high humidity (87%), the model folded the structures, despite the mild temperature (23°C).

• During the event, the model stayed with the DD_Dance type. For the full cloud cover (100%) and slightly high humidity (76%), the skins were mostly opened to a 79° angle, despite the gentle wind (19kmph).

• With the sky mostly covered by cloud (75%), and the high humidity (83%), the structures were folded to a 20° angle while maximising the skin openings.

• For the musical event, the model maintained the B_Musical type, which completely closed the structures and skins even after sunset.
• For a mild temperature (20°C), the skins were mostly opened to an 83.4° angle.

• During the event, the model stayed with the CCC_Exercise type with fully opened skins during full cloud cover (100%).

• While staying with the CCC_Exercise type, the angle of the skins was 58.3°, when the sky was mostly covered by cloud (75%), and with moderate wind (24kmph).

• For a slightly higher temperature (27°C) and fresh wind (30kmph), the skins were mostly closed to a 33.5° angle.
• With a mild temperature (17°C) and high humidity (90%), the model folded the horizontal structures.

• With a mild temperature (23°C) and the mostly covering cloud (75%), the angle of the skins was mostly opened to 87.5°.

For the event indicating BBB_Movie, the model completely closed the structures and skins until the event ended.
• On 1 September, the model closed the structures and skins during rain. With full cloud cover (100%) and high humidity (90%), the model folded the structures. During the event, the model featured the DDD_Picnic type, and the skins were adjusted.

• On 2 September, the model closed the structures and skins during rain. During the event, it maintained the CC_Game type, and the night mode of the skins was applied after sunset.

5.7.2. From 24 October to 26 October 2020

To monitor the behaviours of the architectural model after summertime, the real-time simulation continued from 24 to 27 October 2020 (see Figures 39–42). While remaining with Type E, the behaviour was controlled by the threshold rule. During the recorded days, the highest temperature was 21°C and the lowest was 7°C. The temperatures shown during the three days ranged mostly within the thermal comfort level. With a moderate temperature, the behaviour of the folding structures was mainly affected by the weather parameters of cloud cover, humidity, and wind speed. The real-time simulation from 27 August to 3 September 2020 showed that the skins were constantly adjusted by updated weather data. Furthermore, the structures were occasionally folded when the temperature was at around 20°C. During the real-time simulation from 24 to 27 October 2020, the states of the folded structures were maintained mostly more than 1–2 hours.

• On 3 September, for two events in a day, the model featured the BB_Musical type, which completely closed the structures and skins; later on, it featured the C_Fitness type. Between the events, the skins were mostly closed to a 14.2° angle for the high temperature (31°C) and the mostly covering cloud (75%).
With the moderate temperature (17°C), full cloud cover (100%), and high humidity (87%), the model folded the structures to be at a 60° angle.

With the moderate temperature (21°C), cloud cover (50%), normal humidity (57%), and wind speed (11kmph), the model unfolded the structures to be at a 0° angle and adjusted the skin angle to be at 86°.

With the cold temperature (7°C), cloud cover (0%), normal humidity (60%), and wind speed (0kmph), the model folded the structures to be at an 80° angle (the maximum foldable angle).

With the slightly cold temperature (11°C), cloud cover (0%), normal humidity (52%), and wind speed (13kmph), the model folded the structures to be at a 20° angle.
• With the moderate temperature (13°C), full cloud cover (100%), and high humidity (93%), the model folded the structures to be at a 60° angle.

• During raining conditions, the model completely unfolded the structures and closed the skins.

With the moderate temperature (12°C), full cloud cover (100%), and high humidity (82%), the model folded the structures to be at a 60° angle.

With the moderate temperature (14°C), full cloud cover (100%), slightly high humidity (67%), and wind speed (9kmph), the model folded the structures to be at a 40° angle.
5.7.3. From 18 December to 20 December 2020

During wintertime, the architectural model maintained the state of folded structures to maximise sun exposure in the lawn area (see Figures 43–44). From 18 to 20 December 2020 (the recorded days), the temperature ranged from -7°C to 4°C. It was monitored that with the temperature below 5°C the fully folded state was fixed. In other words, it was verified that for cold temperature the behaviour of the model was not affected by changes of other parameters such as humidity, cloud cover, and wind speed. Over the three days, only three behaviours were observed: fully folded state for cold temperature, fully unfolded state for light rain, and night mode.

- On 18 December, for temperature ranging from -3°C to 0°C, the model maintained the fully folded state, despite wind speeds of 11kmph at 10 A.M. and 15kmph at 2 P.M., full cloud cover (100%), and normal humidity.
- On 19 December, for temperature ranging from -4°C to -1°C, the model maintained the fully folded state, despite wind speeds of 7kmph at 2 P.M., cloud cover (0%), and normal humidity.
- On 20 December, for temperature ranging from 2°C to 4°C, the model maintained the fully folded state. It changed to the unfolded state for a moment of light rain.

Figure 5.43.
Real-time simulation (18 December 2020).

Figure 5.44.
Real-time simulation (19 and 20 December 2020).
5.8. Evaluation

5.8.1. Assessment of Real-Time Simulation Outputs

The outputs of the behaviours during the real-time simulation were evaluated via thermal comfort analysis (see Figures 45–48). One of the main tasks assigned to the model was the management of temperature on the site through the control of folding and unfolding structures and skins to provide outdoor thermal comfort. Using the Ladybug plugin that assesses thermal comfort levels based on the Universal Thermal Climate Index (UTCI), the effect of the behaviours for the adjustment of temperature in the lawn area was observed. For the analysis, some moments during the recorded real-time simulation were selected. The weather parameters and the behaviours that were exhibited during the moments were used to measure the distribution of thermal comfort levels in the lawn area, wherein a square grid (3m x 3m) was arranged.

The UTCI is a thermo-psychological measurement of human reactions to weather conditions. With the input of meteorological data such as dry temperature, solar radiation, relative humidity, and wind speed, the standard of the UTCI presents a method that assesses thermal sensation (i.e. the ‘feel like’ temperature, which is different from the actual temperature). The following is the simplified equation of the UTCI, wherein $T$ is air temperature (°C), $MRT$ is mean radiant temperature (°C), $V$ is wind speed (m/s), and $RH$ is relative humidity (%) (Błażejczyk, 2011):

$$UTCI = 3.21 + (0.872 \times T) + (0.2459 \times MRT) - (2.5078 \times V) - (0.0176 \times RH)$$

As stated in the equation, the higher the values of wind speed and humidity, the lower the value of the results of a UTCI calculation (under the condition of applying the same value of mean radiant temperature). That is to say, when the values of wind speed and humidity are high, people may sense that the ‘feel like’ temperature is lower than the actual temperature. Furthermore, a higher value of solar radiation can cause a higher result of a UTCI calculation.

In terms of thermal stress, the range of UTCI equivalent temperature is categorised as follows (Błażejczyk et al., 2013; Romanzko et al., 2019): a UTCI (°C) between +9 and +26 presents no thermal stress or thermal neutral condition; with a UTCI between +26 and +32, moderate heat stress can be felt; a UTCI between +32 and +35 can cause slight and moderate cold stress; a UTCI below +13 or above +32 indicates a strong cold or hot stress. As shown in Figures 45–48, the resulting values of the UTCI calculated using the weather parameters (displayed at selected moments) were visualised on the grid in the lawn area with the following colour groups: blue represents a UTCI below +10; yellow and orange represent a UTCI ranging between +10 and +30; red represents a UTCI above +30. For the simulation of the analysis, a decimal value of 0.25 ground reflectivity was applied, which presents outdoor grass (the value of the fraction of solar radiation reflected from the ground ranged from 0 to 1).

The analysis presented the approximate distribution of outdoor thermal comfort levels within the lawn area, which was influenced by specific behaviours of the architectural model. The analysis was conducted to observe whether the architectural model could perform the right behaviour for the right moment. The ranges of UTCI values distributed in the lawn area (RT is reduced temperature) are described as follows:

1. 28 Aug. at 3 P.M.: 24.96°C - 20.05°C (RT = 4.91°C)
2. 29 Aug. at 11 A.M.: 24.92°C - 16.85°C (RT = 8.07°C)
3. 30 Aug. at 4 P.M.: 21.22°C - 14.08°C (RT = 7.14°C)
4. 31 Aug. at 2 P.M.: 25.24°C - 12.29°C (RT = 12.95°C)
5. 1 Sept. at 4 P.M.: 21.06°C - 15.81°C (RT = 5.25°C)
6. 3 Sept. at 1 P.M.: 29.01°C - 18.43°C (RT = 10.58°C)
8. 26 Oct. at 12 P.M.: 15.55°C - 9.98°C (RT = 5.57°C)
9. 27 Oct. at 2 P.M.: 14.90°C - 6.17°C (RT = 8.73°C)
10. 18 Dec. at 10 A.M.: (-10.76°C) - (-11.54°C) (RT = 0.88°C)
11. 18 Dec. at 12 P.M.: (-0.22°C) - (-1.10°C) (RT = 0.88°C)
12. 18 Dec. at 2 P.M.: (-11.27°C) - (-12.88°C) (RT = 1.61°C)

From 28 August to 3 September, the model unfolded the structures and opened the skins so as to reduce the temperature in the lawn area through providing mostly covering shadows. From 25 to 27 October, it occasionally folded the structures in order to expose the sky with moderate temperature. From 18 to 20 December, it maintained the fixed folded state to maximise sun exposure.
Figure 5.45. Outdoor thermal analysis (simulated with selected moments from 28–30 August 2020).

Figure 5.46. Outdoor thermal analysis (simulated with selected moments from 31 August to 5 September 2020).
Figure 5.47. Outdoor thermal analysis (simulated with selected moments from 25–27 October 2020).

Figure 5.48. Outdoor thermal analysis (simulated with selected moments from 18 December 2020).
The model is designed to consist of a computational process that can capture specific kinds of data and make decisions from that to be responsive to the spatial requirements for temporary events in Bryant Park. It is developed to provide an active control system of kinetic structures, which purpose is to modify the light/shade, temperature, acoustic condition and access of the public space that is underneath the structures. Through the real-time simulation, it was observed that the model regulated spatial qualities while performing the task of maintaining the thermal comfort level within the lawn area, occasionally providing an optimised acoustic volume through completely closing skins and structures and controlling visitors’ access through folding structures.

Figures 5.49 - 5.53 present a comprehensive comparison of how the model performed its task within a range of specific circumstances on five days (28 August, 1 September, 3 September, 26 October and 18 December). The following describes the noticeable moments resulting from the model’s reasonable behaviours as programmed for the temperature parameters and the social events:

- On 28 August, the model tended to close the skins, due to the high temperature (ranged from 26 to 29°C) with the normal humidity (ranged from 46 to 60%) at the daytime of 10 A.M., 12 P.M. and 2 P.M. (55.2° angle for 26°C at 10 A.M.; 43.7° angle for 27°C at 12 P.M.; 32.6° angle for 29°C at 2 P.M.). The amount of cover cloud ranged from 25 to 50% and the wind speed ranged from 0 to 20 kmph. At 4 P.M., due to the high temperature (31°C), the computing process for folding the skins resulted in 5.1° to provide more shade underneath the structures. The social event of Celebration was arranged from 5 P.M. (nearly night-time). The opening of the structures was aimed to expose the sky. For the relatively private type of this social event, the model limited the access to be partially open/closed.

- On 1 September, for the moderate but slightly hot temperature (ranged from 22 to 24°C) and the high humidity (ranged from 69 to 94%) at the daytime of 10 A.M., 12 P.M. and 2 P.M., the angle of skins resulted in being above 90°, which folded the structures to 20°. The 100% of cover cloud was maintained and the wind speed ranged from 15 to 20 kmph. While the model partially opened the top structures for the social event of Picnic, the skins resulted in being 90° at 2 P.M. At 4 P.M., the skins resulted in being 52.7° for the slightly hot temperature (27°C). If the temperature was higher during the event, the model would close the skins to provide more shade. The open-access for this event was maintained.

- On 3 September, for the hot temperature (25 and 27°C), the 75% of cover cloud, and the normal humidity (45 and 50%), the angles of skins were 73° at 10 A.M. and 55.2 at 12 P.M. At 1 P.M. for the event of a musical type, the model completely closed the structures and the skins to provide an acoustic space for widely spreading sound reflections. During the event, the access was closed on two sides. At 6 P.M. (near night-time) when the event of a fitness type (a relatively private type) was held, the model partially open/closed the access. During the night event, the model completely opened the top structures as the light/shade was not effective.

- On 26 October and 18 December, the model performed its folding structures according to the threshold rule. For the moderate temperature (from 12 to 13°C), the model folded the structures to be at 40° and 60° during the daytime. For the cold temperature (from -4 to 0°C), it maintained the structures to be at 80°.

The adjustment of the light/shade enabled by the model in the lawn area was simulated with the angle of sun rotation. Although this simulation could project only the direction of sunlight (excluding the sun reflections on surrounding buildings or cloud amounts, which disperse sunlight intensities), it presents the approximated measurement of possible shadow casting on the lawn area. According to the simulation of shadow casting at 12 P.M. in five days, the area covered by shadow increased during the summertime and reduced during the wintertime despite the different sun angle. The measured shadow casting and the input data are illustrated as following:

- On 1 September, S1: 20°; A: 63.9% (for T: 23; C:100; H:94; W:15)
- On 28 August, S1: 0°; S2: 43.7°; A: 72.8% (for T: 27; C: 50; H:56; W:20)
- On 3 September, S1: 0°; S2: 55.2°; A: 68.0% (for T: 27; C:75; H:45; W:15)
- On 26 October, S1:60°; A: 51.3% (for T: 13; C:100; H:93; W:0)
- On 18 December, S1: 80°; A: 41.8% (for T: -5; C:100; H:63; W:0)

The social data indicated that the events in the lawn area were mostly scheduled at night-time. When the light/shade was less effective at nearly night-time, the model could expose the sky. For the daytime social events, the designs of partially opening the top structures were to provide users’ choice of their occupation underneath exposed or shadow covered areas. The control of open/closed access was to offer enclosed spatial qualities for certain groups’ organisations. For the events that require acoustics, the model was designed to provide an optimised volume, which can improve wide-spreading sound reflections radiated from a sound source.
28 August

12:00

Temperature (°C): 31
Cloud Cover (%): 50
Humidity (%): 40
Wind Speed (kmph): 13

Hot
Partly Cloudy
Normal
Windy

Access: Open

18:00

Temperature (°C): 31
Cloud Cover (%): 75
Humidity (%): 39
Wind Speed (kmph): 15

Hot
Partly Cloudy
Normal
Windy

Access: Partially Open/Closed

Figure 5.49. Architectural behaviors controlling the light/shade, temperature, and access in the lawn area (28 August).
1 September

12:00

- Temperature (°C): 23
- Cloud Cover (%): 100
- Humidity (%): 94
- Wind Speed (km/h): 20

Architectural Behaviours controlling the light/shade, temperature in the lawn area (1 September).

14:00

- Social Event: Picnic (14:00-16:30)
- Temperature (°C): 34
- Cloud Cover (%): 100
- Humidity (%): 69
- Wind Speed (km/h): 10

Access: Open

UTC Range: 13.67 - 28.16°C

Light/Shade: 14:00
3 September

13:00

Social Event: Broadway in Bryant Park (12:30-13:30)
- Temperature (°C): 28
- Cloud Cover (%): 75
- Humidity (%): 37
- Wind Speed (kmph): 9

Temperature (°C): 28
Cloud Cover (%): 75
Humidity (%): 37
Wind Speed (kmph): 9

Access: Closed

18:00

Social Event: Thursday Yoga (18:00-19:00)
- Temperature (°C): 37
- Cloud Cover (%): 75
- Humidity (%): 39
- Wind Speed (kmph): 15

Temperature (°C): 37
Cloud Cover (%): 75
Humidity (%): 39
Wind Speed (kmph): 15

Access: Partially Open/Closed

Figure 5.51. Architectural behaviors controlling the light/shade, temperature, access and acoustic condition in the lawn area (3 September)
**Figure 5.52.** Architectural behaviors controlling the light/shade and temperature in the lawn area (26 October).
18 December

12:00

Temperature (°C): -3
Cloud Cover (%): 100
Humidity (%): 83
Wind Speed (kmph): 0

18:00

Temperature (°C): -1
Cloud Cover (%): 0
Humidity (%): 55
Wind Speed (kmph): 9

Architectural behaviours controlling the light/shade and temperature in the lawn area (18 December).
5.8.2. Comparison between Dynamic Behaviours and Static Structures

The further evaluation focused on the comparison of the outcomes simulated with the model’s dynamic behaviours of kinetic structures, which can be altered by input data, and the static states of structures (see Figures 5.54-5.62). It assessed how the programmed model could regulate its actions in response to required spatial qualities. The data collected for 2019 (the past year’s data) were used for the input data of weather parameters and social events. The same input data (e.g. the same sun position and weather parameters) directed by selected moments in 2019 were used in the simulation of the changing outputs of the model’s kinetic structures and the fixed structures for the comparison of different results. The evaluation demonstrated how the model’s behaviours satisfied its criteria for specific circumstances and compared them with the static structures, which may include a single advantage. This comparison was conducted to verify how suitable the dynamic behaviours were, to different seasons and social events.

The analysis measured the UTCI ranges and corresponding thermal stress levels caused by the changing behaviours and the fixed structures. The main purpose of the programmed behaviours was the maintenance of outdoor thermal comfort through modifying sunlight exposure and shadow areas. The UTCI value of the exposed condition of the site (i.e. open condition) was compared with the UTCI values simulated with the structures (in Figures 5.54-5.62). The UTCI value of the open condition of the site was labelled S1. There were three kinds of static structures for the comparison. These were the completely closed structures (labelled S2), the structure whose skins were 45° tilted (labelled S3), and the structure whose frames were 60° folded (labelled S4). S2 is the form of folded frames that can provide distinct sunlight/shadow areas. S4 is the form that features partially open/closed skins that can provide dispersed portions of sunlight/shadow areas. S4 is the form of folded frames that can provide distinct sunlight/shadow areas. Although one of the model’s effective behaviours was that the model could distinguish the rain contexts of weather and completely close its structures and skins, such moments were excluded in this comparison. (For moments involving rain contexts, the closing behaviour is appropriate regardless of other factors, and the task of maintaining thermal comfort level becomes a secondary priority at such moments).

In Figures 5.54-5.59, the UTCI values caused by the model’s seasonally responsive behaviours, the abovementioned three static structures, and open conditions were compared. The selection of the moments for the simulation was based on days with particular weather conditions, such as the hottest day in summer, the coldest day in winter, and cloudy, dry, and windy days in all four seasons (see Figure 5.54). The moments at noon on different days were simulated for this comparison. The thermal comfort level can be affected not only by temperature but also by humidity and wind conditions. This UTCI analysis was simulated with the selected moments and carried out to evaluate whether the model could provide appropriate actions for thermal comfort level when the secondary weather parameters (i.e. the parameters except for temperature) are inclined or biased. According to the UTCI analysis, the extent of shade area does not affect the range of UTCI. In other words, the wider shadow area is not meant to result in a lower UTCI value, and underneath any structures, the highest and lowest UTCI values are similar (for the highest and lowest UTCI values, a variation of less than 1 in UTCI values would result).

The result of the UTCI analysis showed that the model regulated its actions to be suitable to thermal stress levels through four seasons (see Figures 5.55-5.58). In spring, for the three days observed (15 and 24 April; 9 May) the model folded the structures to be at 20° when the UTCI values of open condition (S1) belonged to the no heat stress level. In spring, when the UTCI value simulated by S1 was below 9 (3.72, which is a slight cold stress level) on 26 March, the model folded the structures to be at 40° (see Figure 5.55). In summer, on 30 June and 20 July, the S1 UTCI values were above no heat stress levels. When the value belonged to a strong heat stress level (34.75), the model’s skins were tilted at 0° to provide a full shadow area underneath the 0° folded structures. At a moderate heat stress level (29.31), its skins were tilted at 26.9° (see Figure 5.56). In autumn, except for 21 October, the model folded the structures to expose the underneath area. On 21 October, the S1 UTCI value was close to no heat stress level (8.81) and the model rotated the skins to be at 87.1°. When it was -14.06 (below -13 and a strong cold stress level) on 16 November, the model folded the structures to be at 80° (see Figure 5.57). In winter, the model maintained the folded state at 80° while the values were below the no heat stress level (see Figure 5.58). Referring to the UTCI value of the open condition (S1), the responsiveness of the dynamic behaviours for thermal comfort level through four seasons was observed. With no heat stress level in spring, the model showed the 20° folded structures that exposed the sky. In summer, it tended to close the skins to provide more shade areas. In autumn and winter, for cold stress levels, it folded the frames to adjust sun exposure.

The high and low UTCI values mainly caused by exposed and shade areas simulated with the two kinds of structures were compared using bar graphs. As shown in Figures 5.55-5.58, the varied UTCI values that were differentiated by the structures were visualised on the site. The effects of the behaviours and the fixed structures were compared by simplifying the values into four levels (quartering the values). In the charts, the darker grey coloured bar presented the
Figure 5.54. Selection of input data (2019 weather data) and the model’s dynamic behaviours.
Figure 5.55. Comparison between the dynamic behaviours (seasonally responsive behaviours in spring) and the static structures.
Figure 5.56. Comparison between the dynamic behaviours (seasonally responsive behaviours in summer) and the static structures.
Comparison between the dynamic behaviours (seasonally responsive behaviours in autumn) and the static structures.
Figure 5.58. Comparison between the dynamic behaviours (seasonally responsive behaviours in winter) and the static structures.
Figure 5.59. Comparison of the UTCI values between the dynamic behaviours (seasonally responsive behaviours) and the static structures.

colder area. The collection of all bar charts showing the simplified percentages of the measured UTCI values is described in Figure 5.59. As illustrated, in spring the low UTCI values (the darkest bar presenting the shaded area) caused by the dynamic behaviours (D1-D4) were less than the values resulting from the static structures (S2-S4). This shows that in moderate spring weather in the behaviours could allow more sun exposure than the static structures. In summer, the model provided more shade area by closing its skins. Except for 12 June (when the SI UTCI value belonged to the no heat stress level), the behaviours provided more shade areas than S3 and S4. In autumn and winter, through the folded states of structures, the model minimised shaded areas, and in comparison, they were less than the static structures. The model provided more low UTCI values (shaded areas) in summer (except for S2) and fewer values in other seasons than the static structures.

In terms of the model’s responsiveness to thermal stress levels, there were noticeable moments of its corresponding behaviours within seasons, such as D3, D5, D8, D11 and D14 (see Figure 5.59). When the S1 value was a slight cold level in spring, the model folded the structures more than at other moments (shown by D3). Nevertheless, the outcomes of the UTCI values between D3 (40° folded structures) and S3 (45° tilted skins) provided relatively slight variations. For other critical moments, the behaviours provided the best outcomes comparing to the static structures. In summer, D5 maximised the shaded area through 0° tilted skins for a strong heat stress level and it provided more low UTCI values than S2. Although D8 and S3 were similar forms (providing a difference of 18.1° skin angle), the variations of the outcomes were easily readable in the bar graph (D8 was beneficial for more shade areas). In autumn, when the S1 value was a strong cold stress level, the model provided the most exposed area compared to the other structure (shown by D11). In winter, the model maintained 80° folded frames (shown by D14), even while the S1 UTCI value belonged to a slight cold level.

As the model was programmed to be responsive to text-based data including weather descriptions and parameters, the model could provide different behaviours for similar S1 UTCI values. For instance, on 12 and 15 June, the S1 UTCI values were 24.76 and 25.03 (belonging to the no heat stress level). For the cloudy moment (on 12 June), the model provided 20° folded structures (shown by D6), whereas it provided 44.5° tilted skins (shown by D7) for the sunny and dry moment (on 15 June). On 22 and 23 October, with the S1 UTCI values of 8.81 and 7.5 (belonging to a slight cold stress level but close to 9, which is a no heat stress level), it provided 87.1° tilted skins for a sunny day with a moderate temperature (shown by D9), and it provided 20° folded structures for the cloudy and humid moment (shown by D10). This shows that when the UTCI or the feel-like temperature is not critical to cause thermal stress, the range of behaviours
can be varied in response to the text of weather descriptions and secondary weather parameters. As mentioned above (describing D3, D5, D8, D11 and D14), there were important moments to evaluate the outputs of behaviours when the feel-like temperatures could cause thermal stress. The comparison of the S1 UTCI values and the behaviours shows that the behaviours caused by the secondary weather parameters (i.e. cloud cover, humidity, and wind speed) could be altered when the UTCI or feel-like temperature was not crucial.

In addition, the directly projected shadow area through the structures was measured and described by the percentages in Figures 5.55-5.58. Although it is generally considered that folded frames would minimise shadow area, this measurement was conducted to verify whether the length and the angle of the folded state are properly positioned to allow sun exposure through wintertime sunlight directions. Whereas the UTCI analysis simulated the reflections of sunlight, the measurement of shadow areas was generated and directly projected by sunlight directions. In spring, the percentages of shadow areas cast by the dynamic behaviours (D1-D4, which showed 20° and 40° folded structures) were between S3 (the skins were 45° tilted) and S4 (the 60° folded structures). In summer, except for D6 (20° folded structures), the shadow areas from the dynamic behaviours (D5, D7 and D8) were between S2 and S3 (D6 was between S3 and S4). In autumn, except for D11 (80° folded structures), the shadow areas of D9, D10 and D12 were between S3 and S4 (D11 was below S4). In winter, D13-D16 were below S4. The measurement showed that the behaviours provided more shadow areas than S2 and S3 in summer and fewer areas than S2 and S3 in winter.

By measuring the UTCI value of open conditions and the percentage of shade areas, the comparison shows that the model’s actions provided the best outcomes among other states of structures. As described in the UTCI formula, thermal comfort levels are related not only to temperatures but also to the variables of solar radiation, relative humidity, and wind speed. The model was programmed to fold its frames more if the input of temperature parameter was below 10°C and to close its skins more if the input was above 20°C. With moderate temperatures (i.e. temperature from 10°C to 20°C), the range of the behaviours could be varied to be responsive to the text of weather description and the secondary weather parameters. For the critical moments causing thermal stresses, the model could maximise and minimise its shade areas. Compared to static structures, the model could provide the folded state of structures to obtain the most exposed areas for the wintertime event called Winter Village at Bryant Park, and its closing skins for summertime could provide the most shadow-cast areas. In other seasons it could adjust its skin openings to provide partially dispersed shade areas.

The model’s social responsiveness for summertime events was compared with the static structures (see Figure 5.60-5.62). As shown in the list in Figure 5.60-5.61, the schedule of social events started on 16 May and ended on 29 September 2019. Figure 5.60 presents a comparison between the model’s dynamic behaviours for daytime events (DS1 for 10 A.M. on 1 June; DS2 for 1 P.M. on 25 June; and DS3 for 12 P.M. on 15 September) and the static structures (the same S1, S2, S3 and S4 conditions that were shown in Figure 5.54-5.59). Figure 5.61 included the behaviours for nearly night-time events (DS4 for 5 P.M. on 24 May; DS5 for 5 P.M. on 15 July; DS6 for 6 P.M. on 29 August). The behaviours responding to three types of social events were simulated: DS1 and DS4 were the behaviours responding to party/dance type events. DS2 and DS5 were the completely enclosed space for performance/show type events. DS3 and DS6 were the behaviours for fitness/exercise.

For the responsiveness to social events, the model’s criteria were extended from the adjustment of shade areas to the control of acoustic space, access, and private/public zones. The model was programmed to maintain the assigned features of partially folded structures during the occurrence of events (as shown in Figure 5.20-5.21). The purpose of forming the features was to provide more exposed and distinct shade areas for small-group gatherings and their preference for the two areas. For summertime, the enclosed acoustic space was appropriate to provide the fully shade-covered area for a large group audience. While the model maintained the partially folded features of frames for social events, it regulated the skin angles according to the environmental conditions. For instance, as shown in Figure 5.60, the skin angle of DS1 was 73.9° whose shade areas were partially exposed when the S1 value was 27.42, which was close to a no heat stress level. The skin angle of DS3 was 21.3° when the S1 value was 33.87, which belonged to a strong heat stress level. The model’s skin angles could be adjusted by the environmental conditions during the occurrence of social events.

For the temporary events which normally last about one or two hours, the model was programmed to provide more exposed areas to allow more sunlight or open the sky. For the socially responsive behaviours, the prioritised tasks were the control of access and the offer of distinct exposed and shaded areas. Regarding only the percentage of shade area, the dynamic behaviours were not always the best options when compared to the static structures, except for the enclosed acoustic space. As shown in Figure 5.60, the percentages of shade areas of DS1 and DS3 were between S3 and S4. That is to say, during the daytime, the model provided shade areas more than the 60° folded structures and the unfolded structures with partially opened skins. For nearly night-time events, the shade areas were less covered by the steep sunlight directions, and they were not critical. Following such given rulesets and responding to both social and environmental data, the model provided formally diverse features and spatial qualities.
Figure 5.60. Comparison between the dynamic behaviours (socially responsive behaviours in the daytime) and the static structures.
Comparison between the dynamic behaviours (socially responsive behaviours in nearly night-time) and the static structures.

Figure 5.61.
The dynamic behaviours provided a wide range of dissimilar variations of UTCI values, and the static structures provided slightly different UTCI values (see Figure 5.62). The varied range was caused by the completely enclosing behaviour shown as DS2 and DS5. Such behaviour was multifunctional, as that was advantageous for both acoustics and shade areas in the summertime. Moreover, the percentages of shade area resulting from DS3 and S3 presented similar outcomes of 69.6% and 69.7%. However, according to the result of UTCI values, DS provided more low UTCI values than S3. DS3 provided distinct shade areas while S3 provided widely dispersed shade areas. When the lawn is open to random visitors or individuals while social events are not being held, the model provided wider shade areas. For certain group activities, the model provided varied spatial qualities. When the model responded to social events, the shade areas cast by the static structures could be more than the dynamic behaviours. However, during the events, the model’s criteria were to provide distinct shade areas and control the access.

As described in the comparison, for the extreme or critical cases of weather conditions such as the hottest or the coldest days, the model’s behaviours could provide more thermal comfort areas through its changing states of structures than the static structures. For the level of no heat stress, the range of the model’s behaviours was wider, and the behaviours were more responsive to the text of weather descriptions and the secondary parameters. In summertime and wintertime, when the thermal level of the open condition reached the critical moment causing thermal heat stress, the model’s behaviours provided better options (i.e. maximised or minimised shade areas) compared to the static structures. For the social events, the model offered distinct exposed and shade areas and its skin angles were adjusted in response to environmental conditions. In very few cases, the model provided relatively less effective outcomes of shade areas. For instance, as mentioned above, the result of UTCI values of D3 (40° folded structures) and S3 (45° tilted skins) were slightly different and the percentages of shade area between DS3 and S3 were similar. In that case, DS resulted in more low UTCI values (i.e. more shade areas) through its nearly closed skins, which was advantageous for a strong heat level.

Using one-year data, the model’s seasonally and socially responsive behaviours were simulated, and their outcomes were compared with the static structures. Although the outcomes of shade areas cast by folding states could be predicted, this measurement showed its possible results from the two kinds of structures at selected moments. As the main task for the model’s action was the maintenance of outdoor thermal comfort, the UTCI values and shade areas were measured, and they evidenced the model’s appropriate behaviours for weather conditions. The seasonally responsive behaviours adjusted the angle of the folding structures, positioned on the top for thermal comfort, and kept the fully folded state of the side structures to allow full access. Meanwhile, the socially responsive behaviours offered users’ choice of exposed or shaded areas through the assigned features of structures and controlled visitors’ access by unfolding the side structures. Whereas the static structures compared maintained certain conditions of exposed and shade areas (i.e. fully covered, dispersed, or distinct shade areas) that were advantageous mainly for summertime, the dynamic behaviours could adjust the areas through selecting the best option among those kinds of conditions across all seasons. The socially responsive behaviours provided more varied conditions for small groups, while the shade areas underneath the folding structures were adjusted by the environmental conditions.
5.8.3. Instances of Unexpected and Contradictory Outcomes

The design process involved identifying the errors or defects and fixing the actual or potential failures in inputting data, information processing, and outputting behaviours through simulations. Through monitoring the relationship between the model’s input of text-based data and output of behaviours, the ruleset in the algorithm was adjusted and instructed. The design phases (i.e. Design Experiments 1 and 2; Design Development) examined similar output systems of folding structures. When the limitations of the model’s behaviours were noticed in each phase, the input and the output systems in the algorithm were modified. The description of real-time simulations demonstrated how the model provided adequate behaviours suitable for the input data. There were failure moments that were unexpected and contradictory, and the failure modes were resolved. This section illustrates how the occurrences of unexpected and contradictory outcomes led to the improvement of the model.

In Design Experiment 1, the test focused on simulating the system that connects the input of real-time data and the output of behaviours. One of the critical problems of the model in Design Experiment 1 was that the ruleset involved a direct process between the input and the output. While the model unfolded the structures for the occupation during a social event, the model titled its skins based on the instruction from text only (without parameters) to assigned numbers (without calculations). For instance, when the input of text was ‘sunny’, the model tilted the angle to be at 30°; when the input was ‘cloudy’, it tilted the angle to be at 80°. Its behaviours could result in conflicting or contradictory outcomes with weather parameters such as temperature: for a sunny and hot day the 30° skin angle is fairly appropriate as it provides more shaded areas, but for a sunny day with a slightly cold temperature, a 30° tilted skin angle could be inappropriate; with a slightly cooler temperature and a cloudy day, the 80° tilted skin angle could be appropriate to expose the site, but for a cloudy day with a hot temperature, it could be inappropriate. The test in the first phase succeeded the model’s real-time actions but missed context-responsiveness for both environmental and social circumstances. As it could be seen that there was the potential for failure of the model to perform regarding the environmental or seasonal responsiveness as it could only be active during summertime. As the model could provide the action of the skins only within the range of 90°, the skins open at 90° for temperatures below 10°C, and occasionally closed the skins for temperatures around 20°C. As the model could provide the action of the skins only within the range of 90°, the environmental responsiveness could be limited. Although the model could rapidly adjust the opening of the skins during summertime, it was noticed that for the other seasons, its actions would maintain a 90° open position. In Design Experiment 2, the model involved limitations around 20°C. As being responsive to weather parameters after sunset was noticed was deemed to be an unnecessary behaviour for environmental circumstances (the behaviour performing after sunset was not effective with thermal comfort), the night mode was added to the algorithm.

The model in Design Experiment 2 was programmed to be responsive to the context of weather parameters and social events through the assigned numbers. The basic principle was that the skins were responsive to environmental data and the side folding structures were altered by the social events. While recording the real-time simulation, the model mostly maintained the skins open at 90° for temperatures below 10°C, and occasionally closed the skins for temperatures around 20°C. As the model could provide the action of the skins only within the range of 90°, the environmental responsiveness could be limited. Although the model could rapidly adjust the opening of the skins during summertime, it was noticed that for the other seasons, its actions would maintain a 90° open position. In Design Experiment 2, the model involved limitations regarding the environmental or seasonal responsiveness as it could only be active during summertime. As it could be seen that there was the potential for failure of the model to perform its actions in all seasons, a wider range of the model’s behaviours was programmed in the phase of Design Development.

Design Development focused on forming the range of the model’s behaviours and on programming the prioritised rulesets. In the algorithm in the Design Development phase, the pathways of information processed from the environmental and social data to the folding observed, and its ruleset was specified. The recorded real-time simulation showed the model’s expected behaviours without errors. Before the recording, there were occasionally surprising or unexpected outputs due to the misleading ruleset. For instance, one of the errors occurred, when the input data directing to a night mode and a rain mode were inserted into the algorithm and they conflicted. In the early simulation, the model was programmed not to be responsive to input data while maintaining a night mode. Later, the algorithm was modified to be responsive to the input data involving rain contexts at any time, as the social events could be held during night-time. Figure 5.63. shows that the model was not responsive to night-time after sunset (left), but after including the input of sunrise/sunset times and changing the ruleset, the model could provide a night mode after sunset (right). As being responsive to weather parameters after sunset was noticed was deemed to be an unnecessary behaviour for environmental circumstances (the behaviour performing after sunset was not effective with thermal comfort), the night mode was added to the algorithm.

The recorded real-time simulation showed the model’s expected behaviours without errors. Before the recording, there were occasionally surprising or unexpected outputs due to the misleading ruleset. For instance, one of the errors occurred, when the input data directing to a night mode and a rain mode were inserted into the algorithm and they conflicted. In the early simulation, the model was programmed not to be responsive to input data while maintaining a night mode. Later, the algorithm was modified to be responsive to the input data involving rain contexts at any time, as the social events could be held during night-time. Figure 5.63. shows that the model was not responsive to night-time after sunset (left), but after including the input of sunrise/sunset times and changing the ruleset, the model could provide a night mode after sunset (right). As being responsive to weather parameters after sunset was noticed was deemed to be an unnecessary behaviour for environmental circumstances (the behaviour performing after sunset was not effective with thermal comfort), the night mode was added to the algorithm.

The recorded real-time simulation showed the model’s expected behaviours without errors. Before the recording, there were occasionally surprising or unexpected outputs due to the misleading ruleset. For instance, one of the errors occurred, when the input data directing to a night mode and a rain mode were inserted into the algorithm and they conflicted. In the early simulation, the model was programmed not to be responsive to input data while maintaining a night mode. Later, the algorithm was modified to be responsive to the input data involving rain contexts at any time, as the social events could be held during night-time. Figure 5.63. shows that the model was not responsive to night-time after sunset (left), but after including the input of sunrise/sunset times and changing the ruleset, the model could provide a night mode after sunset (right). As being responsive to weather parameters after sunset was noticed was deemed to be an unnecessary behaviour for environmental circumstances (the behaviour performing after sunset was not effective with thermal comfort), the night mode was added to the algorithm.

The recorded real-time simulation showed the model’s expected behaviours without errors. Before the recording, there were occasionally surprising or unexpected outputs due to the misleading ruleset. For instance, one of the errors occurred, when the input data directing to a night mode and a rain mode were inserted into the algorithm and they conflicted. In the early simulation, the model was programmed not to be responsive to input data while maintaining a night mode. Later, the algorithm was modified to be responsive to the input data involving rain contexts at any time, as the social events could be held during night-time. Figure 5.63. shows that the model was not responsive to night-time after sunset (left), but after including the input of sunrise/sunset times and changing the ruleset, the model could provide a night mode after sunset (right). As being responsive to weather parameters after sunset was noticed was deemed to be an unnecessary behaviour for environmental circumstances (the behaviour performing after sunset was not effective with thermal comfort), the night mode was added to the algorithm.
structures were intertwined. As shown in Figure 5.64, one of the errors was that the model missed its pathway for the responsiveness for social data. The left image in Figure 5.64 shows that the misleading ruleset resulted in an unexpected folding state: a social event was scheduled from 19:00 to 21:00 and the model was not responsive at 19:30 during the social event. Its skins were changed to a night mode while its structures were 20° folded in response to the environmental data. The right image in Figure 5.64 also shows that the model only responded to the environmental data at 14:30 during the occurrence of a social event (scheduled from 14:00 to 16:30). To control both the structures and the skins and to avoid errors, a prioritising procedure was included in the ruleset. As the raining contexts in the data were the most critical case for the activities underneath the structure, the model was programmed to unfold the structures and the skins to cover or close the area at any time in such contexts. During holding a social event, the model was programmed to maintain an assigned feature of structures and to control its skins by the weather parameters. After sunset, the structures could continue to be responsive to social events while the skins maintained a night mode. Through observing unexpected outcomes, such a prioritising procedure in the ruleset could be programmed.

To demonstrate that the process of calculating the finalised assigned numbers enabled the model to be suitable for weather conditions and to show a case of how to fail the test caused by a misleading ruleset, supplementary simulations applying unadjusted assigned numbers were carried out (see Figure 5.65). Before recording the real-time simulation, the design process included adjusting the numbers to get expected outputs through inputting a different set of weather parameters. Using the input of 2019 data and selecting the critical moments that caused thermal stress (shown in Figure 5.54), the four cases were simulated. Instead of inputting the exponentially increasing or decreasing numbers, which were the finalised adjusted assigned numbers (shown in Figure 5.23), the linearly or proportionally increasing and decreasing numbers were inserted into the algorithm (resulting in D3_R1, D8_R1, D11_R1 and D18_R1). Also, the wider range of assigned numbers (the number of 20 were added to the ranges) involving the exponentially increasing and decreasing numbers, which were similar to the finalised assigned numbers, were inserted into the algorithm (resulting in D3_R2, D8_R2, D11_R2 and D18_R2).

In Figure 5.65, the graphs present the unadjusted assigned numbers (top), and the outputs from those numbers compared to the original outputs (D3, D8, D11, and D16).
The simulations in Figure 5.65 show that for the critical weather conditions belonging to thermal stress levels, the model could provide less effective behaviours by applying a different set of inclinations and ranges for the assigned numbers. D3 presented a more folded state (40° folded state) than other outputs (20° folded state shown by D1, D2 and D4) in spring, for a slightly cold thermal stress level (UTCI value: 3.72), D3_R1 provided the 20° folded state and D3_R2 provided the 60° folded state. D3_R1 provided the same angle with other outputs in spring, and D3_R2 provided the overly open state for a slightly cold level. For a slightly hot stress level in summer (UTCI value: 29.31), D8 provided 26.9° titled skins whereas D8_R1 provided 45.6° titled skins and D8_R2 provided 65.6° titled skins. For a hot condition, the model needed to close the skins to provide more shaded areas, but D8_R1 and D8_R2 provided more open skins for the condition. For a strong cold stress level (-14.06) and a slightly cold stress level (-11.3) in autumn and winter, D11 and D16 provided the maximum folded state (80°) and D11_R1 and D16_R1 provided the 60° folded state, with D11_R2 and D16_R2 providing the maximum folded state. Because of the change of the inclinations and ranges in the graphs presenting the assigned numbers and the misleading weighting processes, the model provided unsuitable or inadequate behaviours for the weather conditions.

Through observation, evaluation, and instruction, the model’s behaviours were improved. The model required its prompt behaviours within the time interval for updating real-time data. At the moments of occurring errors, the information processing needed to cease and be fixed. The model’s self-learning or self-improvement processes are a way to enhance the model’s ability to process information (without interrupting its continuous information processes). However, such processes need to allow the model to output its behaviours within the time of updating data. In other words, such processes are required not to delay the model’s prompt responsiveness.

For real-time simulation, the model’s responsiveness within a given time is one of the ways to evaluate or measure its ability. Identifying and reducing the frequent occurrences of failure modes is an important factor for evaluation, and the possible critical failure modes were fixed through the simulations.

5.9. Conclusion

The advanced algorithm was established for programming an architectural model’s behaviours that provide functionally suitable and formally varied spatial qualities. The algorithm developed in the phase of the Design Experiment was further modified, tested, and applied for a different location and scale. In the phase of Design Development, the design examination focused on not only developing the control system or the ruleset in the algorithm for the real-time behaviours but also utilising the algorithm-aided design technique for the physical condition of the model, which can be optimal for different conditions at the site. In architectural design, the term ‘intelligent’ encompasses a wide range of computational techniques. There are different approaches between ‘intelligent modelling’ and ‘intelligent machines’ in the field of architecture. Intelligent modelling is a computer-aided method applying data-processing techniques that gather information related to spatial or environmental conditions, and interprets the information to examine geometries. The intelligent machine is an architectural or spatial entity that is capable of acting, behaving, or performing certain tasks through an embedded system. For the design development of this architectural model, both computational techniques were applied.

The collection of information was essential to design the architectural model. The observation of data on the lawn area identified specific tasks to be assigned to the model. The acoustic and sun/shadow factors were important to be managed by developing an architectural space for the performance event and the site condition surrounded by tall buildings. More detailed information, which included such factors, was investigated. Through the utilisation of a problem-solving technique based on evolutionary computation simulation using the gathered information, computationally generated forms were examined. In the searching processes, three different cases of conditions in the site were simulated (i.e. the searching processes were driven by optimising forms for the moment when the area holds the performance event and situates in the hottest and the coldest days). A geometry, which provided the best compromise between the three cases, was selected. For the moments in between the three cases, the algorithm was developed with the behaviour setting, the threshold rule and the additional if/then rules. The model’s behaviours were assessed by monitoring the real-time simulation with the input of text-based data displayed on the note.

The real-time simulation demonstrated the architectural model’s behaviours that were responsive to the changing conditions on the site. The development of various methods of computational designs for architecture has been accelerated with the advancement of digital tools.
The development of architectural machines has been directed towards examining architectural intelligence. For the aim, this research mainly focused on how a system for processing information between the input of data extracted from digital spaces and the output of physical forms can be designed. The limitation of the model remained in terms of its self-improving system for information processing. In this research, through observing the model’s input and output, the actual and potential problems in the information processing resulting from misleading results were identified and resolved. Although such an adjustment process was not a self-improving system, the process of instructing and restructuring the rules led to reducing its unsatisfactory or unadjusted outputs. The real-time simulation and the evaluation verified the model could provide its adequate actions by selecting the best option. However, possibly some unknown or unpredicted errors could still occur, or occasionally its outputs could be unsatisfactory to the end-users. To facilitate a self-improving system, the consideration of what factors can be evaluated and instructed is necessary. The algorithm can be extended to include additional data to assess and improve its behaviours by itself to enhance its ability to process information.

The information given about the site, such as the type of events, weather conditions and sunlight direction, was used to design the model’s body and dynamic behaviours. For the critical moments of thermal comfort, the model provided maximised or minimised shaded areas. Within its range of the angle of folding structures and skins, various options of spatial qualities were given to the model to match with the input of social data. The design experiment phase verified the technique of the algorithm design, facilitating the model’s responsiveness for the real-time circumstances on the site. Confronting the problems or the limitations in the early tested model’s range of behaviours further indicated the modification of the model’s body and behaviours. The extended range of its behaviours was created during design development. By reducing the occasional errors or failure moments of the model’s behaviours, the model’s self-controlling or automated system could continuously provide appropriate behaviours for the input data. The weighting and prioritising processes enabled the model to provide optimised spatial conditions. The evaluation presented evidence that the model’s selections of spatial qualities could provide the best options in different environmental circumstances and social factors.
Chapter 6. Conclusion
6.1. Introduction

Intelligence and information are inextricably linked. With the advancement of technologies, machines have become increasingly capable of dealing with information. Their outputs as a result of information processing have been continuously evaluated to be called intelligent. The application of cybernetics to architectural experiments initiated the approach to achieving architectural intelligence. Active spaces assembled with sensor/actuator technologies are classified as architectural machines. Although architectural machines have provided novel spatial qualities that can interact with people or respond to environmental changes, their actions have mostly been designed to be simply reactive to either of them. In this research, it was considered that one of the primary abilities with which to control a machine’s consequent reasonable behaviours is based on information processing that can identify contextual changes in location-based and time-related information and select the best options from alternatives. Design methods facilitating such information processing were examined and tested in an urban open space. Applying computational techniques and using urban data, an algorithm was established. Simulations verified that this algorithm was able to develop an architectural model for the real-time conditions on a site. This research stressed that utilisation of data is one of the key means by which to implement an intelligent machine for architectural space.

Based on the foundational question ‘How can an intelligent machine on an architectural scale be designed?’, the following research questions were addressed (as stated in section 1.5):

1. How can the ability of information processing in an architectural machine that consists of kinetic structures be enhanced to act as an intelligent machine that is capable of not only sensing the presence of people or the changing environment in an urban open space but also identifying contextual changes under the circumstances of both factors and providing optimised spatial conditions for outdoor activities?

2. How can information processing in a set of algorithms mapping the sequence from the input of urban datasets to the output of behaviours of kinetic structures embedded in the machine be designed for its real-time decision making of selecting the best option from the alternatives of spatial qualities?

It was noticed that current architectural machines with the aim of architectural intelligence have missed the development of a system situated between input and output. To design an intelligent machine that is capable of optimising spatial conditions through its self-control system in response to the contexts or the meanings in both environmental and social factors in a specific open space in a city, this research examined an architectural model’s decision-making processes involving several rule sets between the input system obtaining real-time data and the output system providing consequent reasonable actions.

Environmental and social factors are the main factors that are to be the input in architectural machines. The assessment of the input of those factors and the output of the required spatial conditions that are suitable to those factors indicates the measurement of an architectural machine’s information-processing ability. Diverse sorts of architectural machines can be designed to consist of various forms and provide different motions, but those two factors are central to the input for architectural machines’ information processing. Utilising the collection of urban data, the challenge regarding developing an intelligent machine in this research was the real-time management of both the external changes (i.e. the environmental conditions) and the internal events (i.e. the activities of people). This research investigated the design of an architectural machine whose spatial condition is not deterministic, but rather flexibly adaptive to changing circumstances. By monitoring the real-time simulation, the model’s inputs and its consequent outputs were evaluated. The model possessed the ability to capture or extract information in digital spaces and perform architectural behaviours through kinetic structures that can adjust the light/shade, temperature, acoustic condition, and access underneath the structures. The evaluation demonstrated that the model could provide the required actions and the best options for the different circumstances on the site.

In this research, the development of the model was based on instructions and reinstructions through noticing and fixing problems in the algorithm. The design of the model could be progressed via observation, modification and improvement, but the limitation of the model is the missing part of its self-improvement ability. Commonly, a machine’s self-improvement system is considered one of the main features of intelligent systems. In terms of this aspect, the model could provide only rule-based actions. However, as described in the evaluation of the model’s outputs, the model provided sufficiently appropriate actions in response to its given information within the time interval of updating real-time data. With additional input data, its advanced information-processing ability can be programmed. Information is essential in order to programme a machine’s body and its range of behaviours. The established algorithm can be extended with more information. This research provided a way of establishing an algorithm utilising urban data and online networks. Based on the method, more information-processing abilities can be tested with respect to the aim of architectural intelligence for future works.
6.2. Reflection

This research contributes to the research domain that has investigated the design of active spaces operable by sensing/controlling/actuating systems which serve as one of the machines that can be distributed in the built environment. Most architectural machines that use detective motion sensors/actuators have provided limited reactive actions due to the lack of information presenting whether there were nearby changes or not. The input of enough information into an architectural machine is essential for it to process particular behaviours. In this research, it was considered that one of the information-processing abilities underlying an entity’s intelligence is the process of decision making, enabling it to select the best options from alternatives, and the actions resulting from the process can be assessed in order to measure its intelligence. Its embedded information-processing ability can be evaluated by observing whether it can constantly select the best options leading to the most advantageous actions that are relevant to given inputs that describe certain circumstances or conditions.

The term ‘intelligent’ has been widely used and informally defined in establishing the goal of the improvement of information technologies. In the literature review, the inquiry as to “What is intelligence for the design of machines?” directed the exploration of the discussions. This research examined domain-specific intelligence: architectural intelligence, urban intelligence, and artificial intelligence. For architectural intelligence, designers have examined machines for the physicality of active spaces, while engineers have dealt with automated control systems of facilities in buildings. Efficient management of data flows between physical and digital spaces has been promoted for urban intelligence. Artificial intelligence has led to the development of diverse, high-level data-processing techniques. With the rise of advanced ICTs along with widespread online networks, the term ‘intelligent’ has been popularly used to encourage investigation into the application of information technologies in designing artefacts that possess diverse information-processing abilities in those domains.

There have been critical points of view on intelligent technologies which have been intensively investigated for technical solutions, but these have often omitted consideration of how people can use these technologies. Sociocultural factors constitute the prime consideration in designing spaces. Although such factors are rarely measurable or information on them is hardly collectable and narrowly identifiable, the data which include them have been collected, since the exponential ways of sharing information have been developed. In this research, the potential of the utilisation of such data was acknowledged for the design of an intelligent machine on an architectural scale. Using such data, this research examined a method of designing an architectural machine that can provide a new type of spatial quality in a public open space in a city. The algorithm was developed to design an architectural space that can assist the changing outdoor activities of people via real-time information processing that can identify environmental and social factors and provide flexibly changing spatial qualities that are suitable to them.

In the field of architectural design, further development of projects on interactive architecture and responsive landscapes has aimed towards achieving architectural intelligence. For the design experiments in this research, the ability of context-awareness to be responsive to social and environmental factors was examined as one of the ways of approaching the development of architectural intelligence. In other words, the ability of acquiring and responding to the meaning of the circumstances surrounding the two factors was considered to be one of the principal abilities that needs to be investigated for the enhancement of architectural machines that can be called ‘intelligent’. With the aim of developing ‘intelligent systems’, urban visualisation techniques have provided a way in which to observe, discover and expose social contexts in a city, and data-processing techniques have been established for machines’ specific tasks. To enhance an architectural model’s information processing, urban data were utilised to construct an algorithm, and data-processing techniques were referred to and applied to the rulesets to establish the algorithm.

As examined in this research, the model could process real-time information to simultaneously manage spatial qualities for differing spatial requirements responding to the occupations of diverse activities and the environmental changes. The experiments verified that the rulesets in the established algorithm could modify various formal and functional configurations for the ongoing social and environmental phenomena in a public open space — tested in Bryant Park. The purpose of the presented model is to provide optimal patterns of structures that can manipulate the light/shade, temperature, acoustic condition, and access for the constantly changing types of occupations in the outdoor space. To maintain the level of thermal comfort, the model could adjust the open/closed portions of the structures in real time. Text in the data could present the meaning of phenomena. By using the text and the numerical values in the data (describing the past and present conditions), the model could be designed to perform its task of maintaining the temperature by controlling the light/shade, providing an acoustic space via an optimised volume, and controlling the access for a range of private/public types of social events.

The data-gathering process in this research was carried out in order to find facts presenting the contexts of social events and environmental conditions in open spaces in NYC. Data visualisation techniques were used to classify where, when and how citizens have used open
spaces in NYC. The general circumstance of temporary events is that they are occasionally created, changed, and, in some cases, disorganised. Open spaces in NYC have allowed organising and reorganising various contexts of events for citizens' social and cultural lives. Analysis of the data identified that a high frequency of such events have been organised in Midtown, which is one of the commercial and office areas in NYC in which few open spaces are located amongst tall buildings (described in sections 4.2.1 and 4.2.2). The results from the data analysis presented that Bryant Park has held social events most frequently (shown in Figure 4.10). The frequency of organising outdoor events in NYC is highly influenced by the weather conditions (shown in Figure 4.8). For the design of outdoor spaces, environmental data in NYC were also collected and analysed. It was identified that the majority of descriptions of social events and weather conditions in text within the two datasets were repeated over past years.

Based on the collection of past data, this research further classified the data into the types of social events and the levels of weather conditions. Categorisation of the two datasets was essential in programming the architectural model's behaviours and enabling the architectural model to acquire real-time conditions in an urban open space. The early experiment (described in section 4.3) focused on mapping the sequence between the datasets and the model's behaviours. Although it verified that the model could activate/deactivate its performance of enclosing a selected area in Bryant Park for the occasional occurrence or occupation of temporary events in real time through reading text-based data, the responsiveness of the contexts in the data was insufficient. In the further experiment (described in section 4.4), the information processing of context-awareness and decision-making processes was developed on the basis of the categorisation of the two datasets. Its behaviours were programmed to be responsive to the levels or circumstances of weather conditions (e.g. hot/cold, cloudy/sunny, dry/humid, calm/windy, and daytime/night-time) as well as to the four grouped social events. In the algorithm, the text-based data were transformed into information through the process of categorisation built upon assigned numbers that indicated the control of angles of the architectural model's structures and skins. The question arose as to whether this algorithm could be applied to a different location and scale and improved so as to perform specific tasks. The evaluation of the model's behaviours indicated further development focusing on both formal and functional aspects.

The algorithm was modified and advanced using the data of the central lawn area in Bryant Park. A wider range of behaviours than observed in the previous model were programmed in the updated algorithm. Data collection (described in section 5.2) presented three main social and environmental conditions in the lawn area: one of the main social events in the area was acoustics-related; the majority of the lawn events were scheduled for nearly night-time; and in summertime the area was exposed to sunlight, while in wintertime it was covered by shadows projected by the surrounding tall buildings. Based on the observation of the data, the model’s specific behaviours were implemented to be suitable for the changing conditions on the site. A problem-solving technique based on evolutionary computation was utilised to search for optimal forms that could be suitable for acoustic and sun/shadow factors. The objective of maximising sound reflections for performance events, maximising shadows for summertime events and minimising shadows for wintertime events was given in the form-searching processes. As a result, a specific geometry compromising the best option for the three conditions was selected. The geometry's folding and unfolding structures provided alternatives that could widely spread sound reflections via an optimised volume, increase the shaded area underneath the structures during summertime and expose sunlight during wintertime (shown in Figure 5.19). The previously established algorithm verified its real-time simulation of an architectural model's behaviours, and the algorithm was modified so as to exhibit more varied spatial qualities and a wider range of behaviours. The better alternatives or behaviours could be programmed with more collected information which was used to simulate different conditions on the site.

Collection of the data presented how the social events and the weather conditions were repeated in the past and further used to implement real-time responsiveness to them. Representation of the data provided the scope of observing happenings, conditions or phenomena in the city. On the basis of facts, the changing conditions in Bryant Park could be identified. The model was designed to follow common sense that need reasonable actions (e.g. if the condition is hot, then provide shaded areas; if the condition is cold, then provide exposed areas; if the condition requires an acoustic space, then enclose the space). Through monitoring its behaviours via real-time simulations, the combination of such if/then rules was detailed in order to avoid conflicts (e.g. if the social event is happening during night-time, then do not change to the night mode; if it is raining during the social event, then enclose the area) (described in section 5.6). The evaluation of the recorded real-time simulation of the model was carried out so as to validate whether it performed the right behaviour at the right moment for the observed phenomena on the site (described in section 5.8.1). It measured the output of the model's behaviours that could adjust the spatial qualities for the thermal comfort level and the context-responsiveness of social events within the lawn area in real time.

To support evidence that the model could provide the best options of spatial qualities in response to contextual changes in input data, the evaluation using the 2019 data was carried out (described in section 5.8.2). The comparison between the static structures and the model's dynamic behaviours of the kinetic structures showed that the model controlled its actions in
order to modify the exposed/shaded areas according to the levels of thermal stress measured by the UTCI analysis. In general, in winter the model could provide the most exposed areas; in summer it could provide the most shade-covered areas (except for the fully shade-covered areas simulated by S2); and in spring and autumn its actions provided more exposed areas in comparison to the areas that the static structures could provide. In spring and autumn, when the UTCI values belonged to levels of thermal cold stress, the model provided more folded states (shown in D3, D11 and D12). On occasions of social events, it maintained partially exposed/shaded areas (where small groups can choose to congregate) while controlling side accesses, and in the case of performance/show types of events, it provided the optimised volume for acoustics while closing side accesses (where a large group can occupy). By identifying text-based data and following the ruleset, the model could select the most suitable actions among the given alternatives for different seasons and different types of social events.

For the design process of producing an intelligent machine, reducing failure moments showing the model’s errors was as important as programming logic-based rulesets showing the model’s expected outputs (described in section 5.8.3). Misleading pathways from input data could cause unexpected and contradictory outputs. The prioritising procedure in the algorithm enabled the model to sort more significant factors in the input data. For instance, the factors of raining context, sunrise/sunset time, and type and schedule of social event in the input data were designated so as to affect the model’s distinguishable actions from its continuous actions responsive to weather parameters. By instructing the model to act extraordinarily with regard to such factors, it could provide its reasonable actions contextually relevant to the input data. In other words, programming the algorithm to identify more important factors among the simultaneously given multiple texts in the input data enabled the model to avoid contradictory actions for prevailing circumstances through its appropriate interpretation of the input data. Moreover, to validate whether or not the model shown in the real-time simulation was able to provide the right actions, the simulation of the model’s behaviours, resulting from the calculation of unadjusted weighting numbers, showed that for the critical moments causing thermal stresses it may mostly provide less efficient behaviours for exposed/shaded areas than the original model’s behaviours simulated using the finally established algorithm.

The essence of using urban data was that they presented facts with which to identify situations in a particular territory or district. Based on the facts described in texts, the design of the machine could focus on establishing the flow of information. The design process in this research followed the procedure of collecting data in a designated area, identifying the usage or conditions of the area, classifying the data, and implementing a design of architectural space based on the observed data. The architectural model’s behaviours were designed to exhibit consequent reasonable outputs. With the advancement of information technologies, it is likely that data in digital spaces can be further exchanged and increased, involving various contexts. The utilisation of such data could be helpful in finding more information on how we inhabit physical spaces and develop them. Urban data are a substantial resource with which to establish the scope to find characteristics in happenings within a certain region. Architectural machines could enhance its sensing ability not only to detect happenings but also to process obtained information so as to provide contextually responsive architectural behaviours. However, some unpredicted circumstances could occur while relying on data presenting local conditions. It is possible that there could be occasional differences between the simulated and actual conditions. Locally sensible data may still be necessary in order to sense the actual conditions. Numerous data can be easily collectable via digital spaces. More locally collected data are advantageous in obtaining accurate information through supplementary scopes for the machine’s sensing ability.

An intelligent machine on an architectural scale needs the ability of providing the best or the optimised spatial qualities for both environmental and social factors as not to merely exhibit reactive motions. In order to possess such ability, the information processing of acquiring contexts in input data from those two factors is crucial. Texts can present descriptions of circumstances with regard to the factors. In order to implement what the model could do or control, it was important to identify which detailed information it could obtain. By classifying repeatedly used texts and numbers in the available data presenting detailed information on the two factors (e.g. weather descriptions, weather parameters, names and schedules of social events), the model’s expected spatial qualities were designed for the alternatives. This research investigated the method of mapping the sequence from text-based data to architectural behaviours in the algorithm. The architectural behaviours were programmed to control the amount of sunlight, the thermal comfort area, private/public zones, and accesses. The manipulation of data associated with the ruleset constituted the core method of designing an intelligent machine that controls such spatial qualities and selects the best option from the given alternatives. Its outputs were evaluated by monitoring how its responsiveness was relevant to its input data. The real-time simulation and the evaluation demonstrated that the model could provide the most suitable actions for the surrounding real-time circumstances for both environmental and social factors in an urban open space.

This research examined a way of obtaining detailed information built upon text-based data that illustrate the environmental conditions and social happenings on a site. It focused on investigating a technically verifiable algorithm that can deliver real-time information from
digital spaces to physical spaces and can connect those spaces. It presented a possible method of programming an architectural model’s behaviours, employing rulesets in the algorithm for the act of categorisation, context-awareness, and decision-making processes, to be embedded in its physical body. Those three terms were considered to be requisite information-processing abilities with which to design an intelligent machine. This research examined the design of not only an architectural body, object or physicality but also its control system. To improve the architectural machine’s information processing, computational techniques developed for urban and machine scales were referred to, applied and tested. The principle of establishing information flow from texts in data to architectural behaviours was examined. The utilisation of text-based data was fundamental in establishing information flow for the architectural machine’s responsive behaviours.

6.3. Limitations

In this research, one of the main focuses was the development of an architectural machine’s ability to identify the activities of people in a specific location. However, the input of social data needs to gather more detailed information on users. The way of collecting data requires certain standards in order to measure the quantities (e.g. numbers) or qualities (e.g. characteristics) of values or facts. Social contexts are hardly measurable in producing social data, in contrast to some urban data that present physical conditions (e.g. lengths of streets, land uses of buildings, and weather parameters within a location). It is an important matter as to how more social data can be collected to read various and specific information on activities. Social data have been increasingly stored, mostly via social media in digital space. Though it was noticed that social media data are available for exposing socially related occurrences within cities, in this research the utilisation of such data was excluded, not only because of the issue surrounding users’ privacy but also due to the lack of utilisable information with which to implement an architectural machine in a public open space. A possible way of collecting and providing social data could be further investigated and accomplished by designing a digital space that is directly connected to a location.

Although the algorithm established in this research provided a method of mapping the sequence between the input of data and the output of architectural behaviours for the self-control system, it has limitations in terms of self-improvement, including its self-learning processes. To enable a machine to learn, it first needs to be told, taught or instructed whether its outcome is satisfactory with respect to a given goal. Feedback-based systems are required in order to implement a machine’s ability to learn. The core theme in the realm of cybernetics is feedback and it is thought that the communication between man and machine can improve both of their performances. Meanwhile, the study of symbol manipulation is focused upon in the field of AI. In this research, to test a real-time self-control system, the design experiment examined the application of a symbolic approach on the basis of if/then rules. Using currently accessible data, the rulesets in the algorithm facilitated the transformation from text-based data into numbers with which to control architectural behaviours. However, for the ability of learning, the input needs additional information on how end users consider the changing conditions of architectural behaviours.

A machine’s information-processing ability can be measured by its outputs and the relevance of its inputs. Reducing the moment of the machine’s unexpected outputs or maintaining its consistent adequate outputs is important for the evaluation of real-time simulation. For the input system, more information such as end users’ opinions is supportive in implementing the machine’s self-learning system. In this research, improvement of the algorithm was based on instructing and reconstructing the ruleset through monitoring the simulation of the model’s inputs and outputs. To fix the ruleset, the model’s information processing needed to be paused. Advancement of the ability of self-learning is that for the moment of fixing its behaviours its information processing can be maintained. For the evaluation of the environmentally responsive behaviours, the UTCI analysis measured the effects of the outputs. For public use of the machine, information on their preferences is still requested. For socially responsive behaviours, collectively given feedback information via online networks can be used to enhance its information-processing ability and real-time responsiveness.

This research focused on the development of the system between the input and the output in an architectural machine, as it has been considered that the information processing between them needs to be enhanced for the aim of architectural intelligence. Collaboration with specialists who can deal with constructing actual digital and physical spaces remains necessary for further improvement of the system. However, this research tested the system based on currently available and applicable technologies for digital and physical spaces. For instance, data have become more open and more available for developers of digital space, which are shared via APIs on websites (mostly for the extension of digital tools) with the purpose of efficient ways of exchanging information; kinetic structures foldable within a 90° angle have been constructed mostly for facade systems in order to provide flexibly adjustable physical forms. Architectural machines’ abilities of sensing and behaving have been realised through the application of responsive technologies. This research investigated the ability of information processing between those abilities.
6.4. Future Work

Computer capabilities have been diversely investigated for architectural designs. In the field of architecture, the examination of algorithm-aided design has focused on computationally generative forms such as highly sophisticated patterns of structures, complex geometries and spatial organisations. This research identified that one of the pathways aiming to achieve architectural intelligence originated from the design of architectural machines. It investigated a design method and examined technical applications with which to build an algorithm for the enhancement of an architectural machine’s embedded control system. For future work towards architectural intelligence, the following suggests the further applicable design of three main systems:

- The input system: the design of digital space directly associated with users in a location.
- The control system: the design of users’ feedback-based systems.
- The output system: the design of physical space in a different location and with a different scale and purpose.

To achieve the goal of assisting the activities of users, an architectural machine essentially needs to possess the ability to read the information on them. The first consideration should be given to how enough information can be obtained to implement an architectural machine’s consequent behaviours. Measurable data are the foundation upon which to process and form information. Future research could investigate different methods through which to collect socially related information. If information, including users’ feedback, is collectable through digital space and input into the machine, the process of evaluating its behaviours is implementable for the learning process to provide more adequate conditions based on users’ opinions. In other words, if there is a website available on which users give their opinions on the physical conditions of an architectural machine, its collected data can be used to evaluate whether or not its behaviours have been suitable for their uses. Concerning the ability of self-improvement, a design of digital space that provides more user information is recommended.

The nature of a computer algorithm is such that it can be manipulated, advanced or extended using additional rule sets. Verification is needed as to whether a certain algorithm can process without errors. Through simulations, this research demonstrated how an algorithm can be mapped between the input of text-based data and the output of architectural behaviours. The established algorithm can be used as a base or an underpinning for a logic-based sequence and applied with additional input data (e.g. users’ feedback). For instance, via a specified URL address (designated from the designed digital space that presents the result of users’ collective opinions), the input can be added to the established algorithm and used to evaluate the output via a ranking system to maintain or modify the status of the output. The input of information on the number of individuals or their preferences can be supplementary for the advancement of the algorithm. The input system is emphasised because the given inputs configure the range of outputs. To implement the outputs or behaviours of an architectural machine that can read contexts, its responsiveness needs to be oriented towards association with specified categories.

This research provided a theoretical model for the design of an architectural machine’s information processing. The three main systems (i.e. input, data processing and output) should be inseparably implemented. By identifying contexts situated in a location, the presented framework is applicable. In this research, an urban open space was the site on which to test an architectural machine’s behaviours. The design focused on maintaining the open condition of the public park and on controlling visitors’ access less. The application of this system is such that the behaviours of an architectural model can be modified for different purposes, scales and sites. For example, in indoor spaces, the control system presented in this research can be modified to divide or distinguish spatial conditions through folding structures. Meanwhile, in private outdoor spaces, behaviours can be oriented towards providing more enclosed spaces, and the control system can be applied to a kinetic building facade by specifying geolocation information. One of the main factors altering social and environmental contexts is location. After designating a location, the information on users can be identified and can further implement an architectural machine’s suitable behaviours.

In the field of architecture, the term ‘intelligent’ has raised the question of how information technologies could be applied in the design of architectural machines. In this research, the information processing ability underlying intelligence is associated with context awareness, the act of categorisation and decision-making processes. The term has been interpreted in diverse ways, which has increased the challenges surrounding enhancing machines’ diverse abilities of information processing. In other words, it has encouraged various approaches to the aim of developing machines that can deal with information. This research provided a way of using the information in digital space, where it was easily sharable, collectable and readable. Information collected in digital space can be used not only to identify phenomena in cities but also to design new types of active spaces.
Bibliography


Barlas, M. A., Çaliskan, O. (2006) Virtual Space as a Public Sphere: rethinking the political and profession. METU Journal of the Faculty of Architecture, JFA.23(2) 1-20.


Shieber, S.M. (2004) The Turing test. verbal behavior as the hallmark of intelligence, MIT.
Shieber, S.M. (2004) The Turing test. verbal behavior as the hallmark of intelligence, MIT.
Appendix
List of Appendices

Appendix A: Raw Data

Appendix B: Data Processing
B.1. Input and Output (Monitored during the Real-Time Simulation from 27 Aug. to 3 Sept., from 24 to 26 Oct. and from 18 to 20 Dec. 2020)

Appendix C: Algorithms
C.1. Algorithm for Real-Time Simulation (Established for Design Experiment 2)
C.2. Algorithm for Real-Time Simulation (Established for Design Development)

Appendix D: Animations
D.1.1. Data Visualisation of NYC Permitted Event in Manhattan (from Nov. 2017 to Sept. 2018)
D.1.2. Data Visualisation of Bryant Park Event (from 1 to 15 Aug. 2018)
D.2.1. Behaviour of Kinetic Structures and Skins (for Design Experiment 1)
D.2.2. Simulation with Event & Weather Data (from 1 to 5 Aug. 2018)
D.3.1. Behaviour of Kinetic Structures and Skins (for Design Experiment 2)
D.3.3. Behaviour of Kinetic Structures and Skins (for Physical Model Test)
D.3.4. Real-Time Simulation of Physical Model (from 29 June to 3 July 2019)
D.4.5. Real-Time Simulation (from 18 to 20 Dec. 2020) _9 Seconds Version
D.4.6. Real-Time Simulation (from 18 to 20 Dec. 2020) _1 Minute and 47 Seconds Version
Appendix A: Raw Data

Appendix A presents the raw data of the Bryant Park event, which are sorted from the data of NYC’s “NYC Permitted Event Information” (from 1 Nov. 2017 to 31 Oct. 2018). Except for the list shown in Appendix A.1. (i.e. Event ID; Event Name; Start Date/Time; End Date/Time; Event Location), in the original data the same text was typed as follows: Event Agency: Parks Department; Event Type: Special Event; Event Borough: Manhattan; Event Street Side: blank; Street Closure Type: N/A; Community Board: 5; Police Precinct: 14. Described in sections 4.3. Design Experiment 1 and 4.4. Design Experiment 2, the data were analyzed to identify the repeatedly arranged public activities in Bryant Park.
## Raw Data of Bryant Park Event (from 1 Nov. 2017 to 31 Oct. 2018)

<table>
<thead>
<tr>
<th>Event ID</th>
<th>Event Name</th>
<th>Start Date</th>
<th>End Date</th>
<th>Event Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>201701</td>
<td>Event 1</td>
<td>2017-11-01</td>
<td>2017-11-30</td>
<td>Bryant Park</td>
</tr>
<tr>
<td>201802</td>
<td>Event 2</td>
<td>2018-02-01</td>
<td>2018-02-28</td>
<td>Bryant Park</td>
</tr>
<tr>
<td>201803</td>
<td>Event 3</td>
<td>2018-03-01</td>
<td>2018-03-31</td>
<td>Bryant Park</td>
</tr>
<tr>
<td>201904</td>
<td>Event 4</td>
<td>2019-04-01</td>
<td>2019-04-30</td>
<td>Bryant Park</td>
</tr>
<tr>
<td>201905</td>
<td>Event 5</td>
<td>2019-05-01</td>
<td>2019-05-31</td>
<td>Bryant Park</td>
</tr>
<tr>
<td>202006</td>
<td>Event 6</td>
<td>2020-06-01</td>
<td>2020-06-30</td>
<td>Bryant Park</td>
</tr>
<tr>
<td>202007</td>
<td>Event 7</td>
<td>2020-07-01</td>
<td>2020-07-31</td>
<td>Bryant Park</td>
</tr>
<tr>
<td>202108</td>
<td>Event 8</td>
<td>2021-08-01</td>
<td>2021-08-31</td>
<td>Bryant Park</td>
</tr>
<tr>
<td>202109</td>
<td>Event 9</td>
<td>2021-09-01</td>
<td>2021-09-30</td>
<td>Bryant Park</td>
</tr>
<tr>
<td>202210</td>
<td>Event 10</td>
<td>2022-10-01</td>
<td>2022-10-31</td>
<td>Bryant Park</td>
</tr>
</tbody>
</table>

A.I.
Raw Data of Bryant Park Event (from 1 Nov. 2017 to 31 Oct. 2018)
### Appendix B: Data Processing

In section 5.7, Real-Time Simulation, the architectural model’s input and output monitored at selected moments were illustrated. Appendix B.1 lists the model’s input and output data processing that were monitored during the recorded real-time simulation (from 27 Aug. to 3 Sept., from 24 to 26 Oct. and from 18 to 20 Dec. 2020) at every hour from 10 A.M. to 8 P.M. The table in this Appendix B shows the list of the input and output observed at every hour, and it presents the model’s overall responsiveness.

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Weather Description</th>
<th>Temperature (°C)</th>
<th>Cloud Cover (%)</th>
<th>Humidity (%)</th>
<th>Wind Speed (knot)</th>
<th>Social Event</th>
<th>Shocklist (T)</th>
<th>Spike (T)</th>
</tr>
</thead>
<tbody>
<tr>
<td>27/08/2020</td>
<td>15:00</td>
<td>Showers</td>
<td>23</td>
<td>75</td>
<td>60</td>
<td>9</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
</tr>
<tr>
<td>17:00</td>
<td>Showers</td>
<td>20</td>
<td>0</td>
<td>90</td>
<td>17</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
</tr>
<tr>
<td>20/08/2020</td>
<td>12:00</td>
<td>Partly Cloudy</td>
<td>27</td>
<td>58</td>
<td>51</td>
<td>13</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
</tr>
<tr>
<td>15:00</td>
<td>Partly Cloudy</td>
<td>23</td>
<td>21</td>
<td>70</td>
<td>40</td>
<td>14</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
</tr>
<tr>
<td>25/08/2020</td>
<td>18:00</td>
<td>Partly Cloudy</td>
<td>31</td>
<td>59</td>
<td>53</td>
<td>25</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
</tr>
<tr>
<td>01/09/2020</td>
<td>15:00</td>
<td>Showers</td>
<td>23</td>
<td>75</td>
<td>44</td>
<td>21</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
</tr>
<tr>
<td>17:00</td>
<td>Partly Cloudy</td>
<td>32</td>
<td>69</td>
<td>52</td>
<td>20</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
</tr>
<tr>
<td>10/09/2020</td>
<td>19:00</td>
<td>Partly Cloudy</td>
<td>33</td>
<td>78</td>
<td>60</td>
<td>26</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
</tr>
<tr>
<td>20/09/2020</td>
<td>16:00</td>
<td>Partly Cloudy</td>
<td>34</td>
<td>76</td>
<td>40</td>
<td>26</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
</tr>
<tr>
<td>30/09/2020</td>
<td>10:00</td>
<td>Showers</td>
<td>26</td>
<td>55</td>
<td>54</td>
<td>15</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
</tr>
<tr>
<td>11/10/2020</td>
<td>11:00</td>
<td>Partly Cloudy</td>
<td>29</td>
<td>58</td>
<td>50</td>
<td>15</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
</tr>
<tr>
<td>22/10/2020</td>
<td>12:00</td>
<td>Partly Cloudy</td>
<td>29</td>
<td>58</td>
<td>50</td>
<td>15</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
</tr>
<tr>
<td>01/11/2020</td>
<td>13:00</td>
<td>Partly Cloudy</td>
<td>29</td>
<td>58</td>
<td>50</td>
<td>15</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
</tr>
<tr>
<td>14/11/2020</td>
<td>14:00</td>
<td>Partly Cloudy</td>
<td>29</td>
<td>58</td>
<td>50</td>
<td>15</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
</tr>
<tr>
<td>17/11/2020</td>
<td>15:00</td>
<td>Partly Cloudy</td>
<td>29</td>
<td>58</td>
<td>50</td>
<td>15</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
</tr>
<tr>
<td>20/11/2020</td>
<td>16:00</td>
<td>Partly Cloudy</td>
<td>29</td>
<td>58</td>
<td>50</td>
<td>15</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
</tr>
<tr>
<td>23/11/2020</td>
<td>17:00</td>
<td>Partly Cloudy</td>
<td>29</td>
<td>58</td>
<td>50</td>
<td>15</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
</tr>
<tr>
<td>26/11/2020</td>
<td>18:00</td>
<td>Partly Cloudy</td>
<td>29</td>
<td>58</td>
<td>50</td>
<td>15</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
</tr>
<tr>
<td>02/12/2020</td>
<td>19:00</td>
<td>Partly Cloudy</td>
<td>29</td>
<td>58</td>
<td>50</td>
<td>15</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
</tr>
<tr>
<td>05/12/2020</td>
<td>20:00</td>
<td>Partly Cloudy</td>
<td>29</td>
<td>58</td>
<td>50</td>
<td>15</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
<td>Type: C (D)</td>
</tr>
</tbody>
</table>

**B.1. Input and Output (Monitored during the Real-Time Simulation from 27 Aug. to 3 Sept., from 24 to 26 Oct. and from 18 to 20 Dec. 2020)**
Appendix C: Algorithms

Appendix C includes the algorithms (established for Design Experiment 2 and Design Development), which are a non-book component. The files can be opened using Rhino 6 and Grasshopper. In the algorithms, weather API ID needs to be updated (obtained from the following website: https://www.worldweatheronline.com/developer). In the Grasshopper files, the texts and numbers in the CSV files, established for the categorisation, are saved in the panels and connected to the architectural model's data processing. The Grasshopper files can be opened and run independently without requesting any file paths associated with a laptop. The non-book component of the algorithms presents the actual systems (that can be run by the software) established for the real-time simulations described in sections 4.4. Design Experiment 2 and 5.7. Real-Time Simulation.

Appendix D: Animations

Appendix D includes the animations of data visualisation, architectural models' behaviours, and real-time simulations. The non-book component of the animations can be seen in a PDF file or via MP4 files. Via the animations of D.1.1. and D.1.2., the data of NYC Permitted Event Information depicted on the map of Manhattan and Bryant Park can be seen through timeframes. Those two animations were conducted to confirm whether the sensing algorithms could read the data within territorial conditions through the use of timeframes. The animations of D.2.1., D.2.2. and D.2.3. show the early model of kinetic structures and skins, as described in section 4.3. Design Experiment 1. The tests (shown in those three animations) focused on enabling folding structures that could enclose an area within the park, connecting with the data (tested using the past data) and simulating in real time. The animations of D.3.1., D.3.2., D.3.3. and D.3.4. show the second model of kinetic structures and skins that was tested using categorised data, as described in section 4.4. Design Experiment 2. The tests (shown in those four animations) verified that the algorithm could regulate both digital and physical models' behaviours in real time. The following animations (i.e. D.4.1., D.4.2., D.4.3., D.4.4., D.4.5. and D.4.6.) show the recorded real-time simulation activated by the algorithm established for Design Development. Selected moments from the animations are described in section 5.7. Real-Time Simulation.