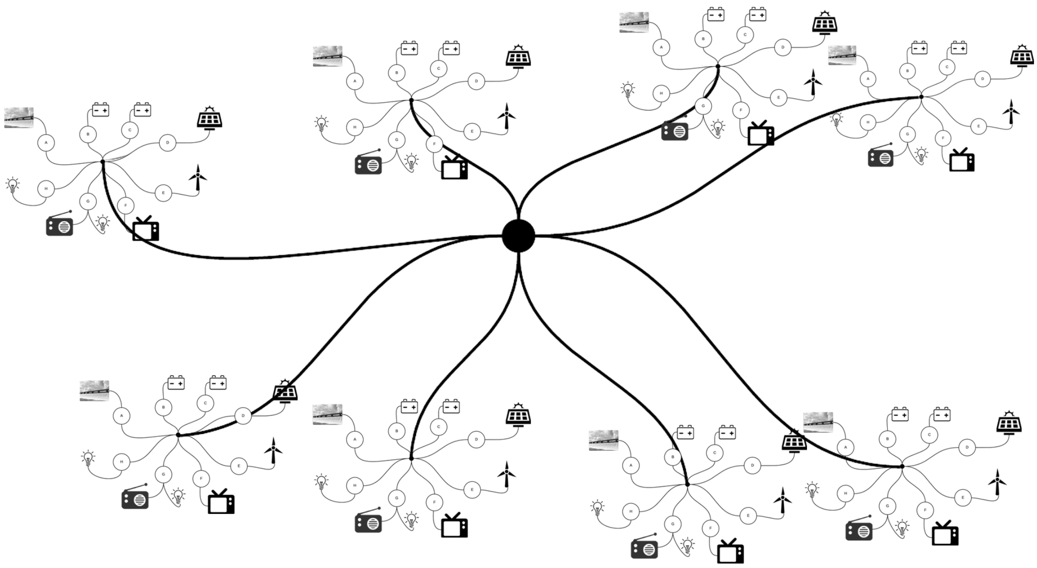
Imperial College London

Department of Electrical and Electronic Engineering

Final Year Project Report 2015



Project Title: **Simulator for Decentralised Energy Systems**

Student: **Yuchen Wang**

CID: **00683700**

Course: **EE4 (EM)**

Project Supervisor: **Prof. Jeremy V. Pitt**

Second Marker: **Dr. Javier A. Barria**

Acknowledgements

Foremost, I would like to thank my supervisor Prof. Jeremy Pitt for taking the time to impart advice, guidance and supervision over the past academic year on this project.

Besides my supervisor, I would like to thank e.quinox, all members past and present for providing me with the opportunity of a lifetime and the information without which, would not make this project possible.

I would also like to take this opportunity to thank my friends, housemates and fellow colleagues for their continuous encouragement, laughter and the fun times we have shared over the last four years.

Last but not the least, I would like to thank my mother for all of her continued support, inspiration and love.

Yuchen Wang
June 2015

Abstract

In many developing countries the electricity grid network is underdeveloped, causing low levels of rural electrification. However, electricity access in these rural communities are not non-existent. There exists a number isolated and independent electricity generators owned by individuals, NGOs and local government institutions. This project aims to accurately simulate a Decentralised Community Energy System which utilises these isolated sources of electricity to better serve the local communities.

This project successfully simulates a Decentralised Community Energy System using a holonic Multi-Agent System, with electricity being contributed to a common pool and allocated fairly to consumers by applying *Rescher's Canons of Distributive Justice.*

Contents

List of Figures

List of Tables

List of Acronyms

Introduction 1

Electricity connections in many developed nations are wide-spread and reliable. Residents in the UK benefits from 100% electricity penetration[2] and a Transmission network which operates above 99% reliability.[3] In the rural areas of many developing countries, electricity is considered a luxury; there are often no wide-spread access to a continuous and reliable electricity supply in the rural areas of many developing countries.[4] However there exists many isolated sources of electricity such as solar panels, wind turbines and standalone diesel generators owned by wealthy households, shops and large organisations.

The aim of this project is to design and implement a self-organising *Decentralised Community Energy Systems* (dCES) which which brings together existing generation infrastructure to provide a reliable electricity supply in rural communities. With the vast majority of people living in communities such as villages, towns and cities, the structure of the decentralised energy system model was designed to be scalable and allows the formation of communities. In a traditional electricity system, electricity largely flows in a uni-directional manner. Electricity is generated away from demand centres, and transported around the country by transmission systems and distributed to consumers by the distribution network. With many of the users of the dCES System capable of both consuming and generating, a scalable micro-grid for this model needs to be designed to be bi-directional.

With the goal of scalability and bi-directional power flow in mind, the model was constructed as a holonic *Multi-Agent System* (MAS). A holonic system by definition is one which is formed of many smaller systems, which are in turn formed of many smaller systems and so on, until reaching the most "elementary" of systems.[5] Within a holonic MAS, individual consumer-provider households and businesses can be modeled as Agents, which can be grouped together to form larger communities such as villages which can be modeled as a single entity. These entities can be further grouped together to form even larger entities such as districts or provinces within the Simulator. Basing the design of the simulator on that of a holonic system simplifies the scalability aspect of having multiple communities in a simulation.

A MAS was selected to simulate the dCES the as Agents within a MAS are required to be:

- able to act independently
- unable to directly manipulate the environment
- unable to control the actions of other Agents.

In the case of this simulator, consumers are independent entities who:

- can't change the environment conditions e.g. have access to Grid Electricity, which satisfies the *unable to directly manipulate the environment* condition

- unable to directly control the consumption and provision of other consumers or providers, satisfying the conditions of *able to act independently* and *unable to control the action of other Agents*

Presage 2 was selected as the MAS platform for this project as support was available from PhD students within the EEE department who currently use the platform. However, there are also some limitations with the platform. The simulation results can occasionally produce inconsistent results due to the parallel execution nature of the simulation platform, and the platform is limited to execute simulations in discrete time steps. Suitable design steps have been taken to mitigate these limitations and have been detailed in the Design section of this report.

To allow the model to be realistic, the actual rural communities in Rwanda with realistic demand and generation profiles were simulated. Due to the lack of available data for some of the Simulation variables, the demand and generation profile has been approximated using data from rural UK demand centres. Results from these simulations indicate the Simulator work as expected.

Background 2

Electricity as a Common Pool Resource

A Common Pool Resource is a depletable resource which can be utilised by a group of people, characterised by a reduction in the availability of this resource as individuals withdraw or utilise this resource.[6] Electricity can be a Common Pool Resource if there exists a finite amount of electricity generation capacity. As users connect demand appliances to the generators, the availability of electricity supply for additional demand diminishes.

In developing communities with significant generation from renewable sources such as wind and solar, the availability of power can be highly variable between two different time periods. This inherent volatility in the amount of available resource means that many communities can not be self-sustainable. By connecting multiple households and communities of consumer-providers (or *Prosumer*) together, electricity generation and consumption can become diversified, increasing the sustainability of electricity access. Electricity in this case becomes a highly volatile Common Pool Resource.

Ostrom showed that Common Property Regimes (arrangements which resource consumers agree to) can be formed to maintain the Common Pool Resources by controlling the access to the resource.[6] For this project, the arrangement will be the participation in a local micro-grid between a number Prosumers. The micro-grid will provide the infrastructure to allow electricity to be pooled as a Common Pool Resource, and will also provide the means to control the access to the resource.

Distributive Justice and Fair Allocation

For the purpose of this project, a micro-grid is assumed to already exist and is being operated automatically by a third party. The micro-grid is also assumed to be able to enforce Prosumer contribution to the Common Pool and restrict its access. With the infrastructure for enforcing the *Common Property Regime* (CPR), it is important to ensure that the allocation would be fair to encourage Prosumers to be part of the system. Being fair forms two of the necessary Ostrom's Principles for Enduring Institutions in a CPR.[6]

Multi-Agent Simulation

The simulation will be designed as a Multi-Agent System (MAS). MASes are particularly suited for this kind of model as Agents in MASes have three very important characteristics:

- Autonomy: Agents act independently

- Local view: no Agent can see or manipulate the environment it is in
- Decentralised: no Agent can control the action of all Agents

In reality, individual households which are represented by Agents in the simulator would all perform actions according to their individual and unique needs, and are not controlled by a third party. This makes Autonomy and Decentralisation a requirement for the Agents in the Simulator. Participants or households connected to the network can't directly control how other participants use or generate electricity for the Common Resource Pool, making it compulsory for the Agent to have a localised view.

Decentralised Community Energy Systems as a Holonic System

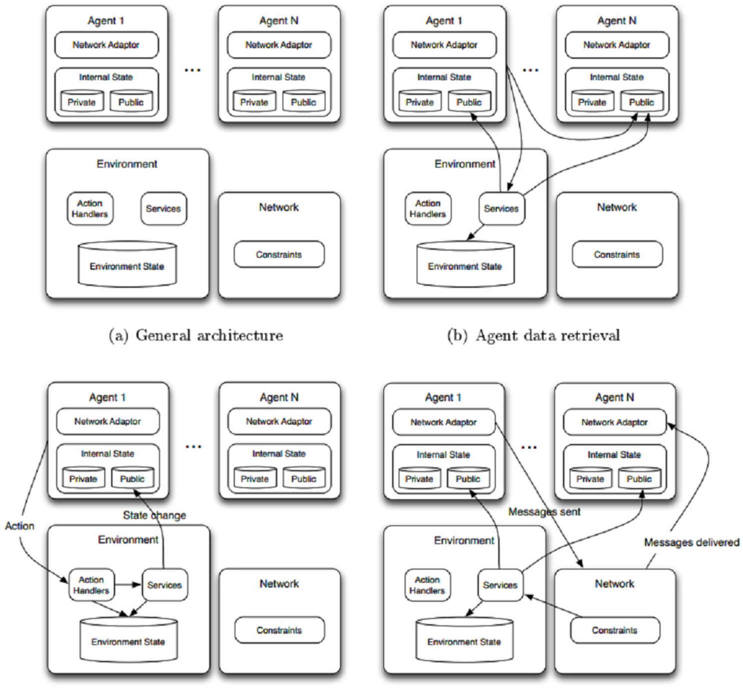
A holonic system (or holarchy) is a system which is composed of interrelated subsystems or institutions, each of which are in turn composed of "sub-subsystems" or institutions and so on, recursively until reaching a lowest level of "elementary" subsystems. Each system, sub-system or institution has a well-defined set of goals or objectives which is achieved through enforcing a set of rules on its members or member-systems.[5] A holonic MAS is what will be utilised to simulate the dCES and the associated CPR to allow the simulator to be scalable with any number of communities and their members.

In the context of a rural dCES, a holonic system in this case would be composed of communities such as Districts, Provinces and Sectors which are composed of "sub-communities" such as Towns and Villages. The sub-communities would be composed of many "elementary" subsystems such as households, businesses and other points of connection for electricity. Each community or institution has the goal of gathering all generation from members, and subsequently fairly allocating electricity to all members. This goal would be achieved with the assumption that they are provided with the necessary infrastructure and powers for enforcing quotas and contribute to a common pool of electricity.

About Presage 2

Presage 2 is a simulation platform for multi-nodal or Agent simulation of societies. The platform was built by Sam Macbeth and is currently maintained by PhD students within Imperial. This platform was chosen for the simulator as it enables the investigation of the impact of agent design (such as household behaviour), network properties (constraints on access) and the physical environment on individual agent behaviour and long-term global network performance.[7] In the context of this Project, each Node or Agent can represent individuals, households, businesses or generators.

In Presage 2, Agents are only allowed to act during increments of time steps, which makes the simulator a discrete time driven one. During each time step, all actions are required to be performed by Agents via *Action Handlers* and *Environment Services*. Figures 2.1 and 2.2 illustrates the general and runtime architecture.



(a) General architecture

(b) Agent data retrieval

(c) Agent acting

(d) Agent messaging

FIGURE 2.1: General Presage 2 architecture[1]

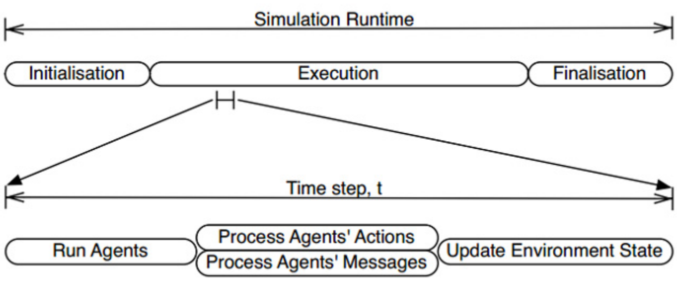


FIGURE 2.2: Presage 2 simulation architecture[1]

Decentralised Generation in Rural Communities of Rwanda

With an estimated 25% rural electrification rate in 2009,[4] vast amounts of rural communities in Africa remain un-electrified. Benefits of a localised local micro-grid would be felt immensely rural Africa, making rural Africa is one obvious candidate for simulation scenarios. For a realistic simulation, knowledge of existing infrastructure in place will need to be obtained. With many possible countries to choose from in Africa, Rwanda in particular has been identified as a good potential simulation scenario to research. Data about electricity consumption is often difficult to source for rural communities in developing countries. Fortunately, some research data are available from the student society e.quinox for Rwanda. e.quinox is a student-led society which aims to find a scalable solution for rural electrification who mainly operate in Rwanda.[8] The subsections below outline some of the ways remote rural communities are able to access electricity in Rwanda.

Electricity Generation

One of the solutions currently being implemented by e.quinox is the "Energy Kisok model".[9] The "Energy Kiosk" model features an Energy Kiosk - a building where the generation, storage and distribution of electricity takes place. In e.quinox operated kiosks, electricity is generated from renewable sources, stored on-site and distributed via leased storage devices. Traditionally, generation have come from roof mounted solar panels. More recently however, hydro-electric generation has been demonstrated to be feasible with the construction of a "Hydro Kiosk" at Rugaragara Falls in Southern Rwanda completed in 2012.[10]

Storage and Distribution

Within each kiosk, electricity that is generated is stored in storage batteries placed within the kiosks. The storage batteries regulate power output and allows access to electricity in the kiosk even during periods of no electricity generation. In the absence of any electricity distribution infrastructure, e.quinox has traditionally leased a number of portable batteries to the local population for the purpose of electricity distribution. An example of the portable batteries leased can be seen in figure 2.3.

Consumers from the local community pay to hire the battery boxes under one of two payment schemes: pay-per-recharge and pay-per-month.[11] The biggest difference between the two schemes are that users can recharge as often as they would like with pay-per-month. Both payment schemes involve the recharge of the boxes at the energy kiosk when they are depleted of energy.

Figure 2.3: First Generation e.quinox Battery Boxes deployed at Rugaragara Falls Kiosk

Standalone Solution

The Standalone Solution is an independent electrification solution which was recently developed by e.quinox for customers who live far from energy kiosks.

The Standalone solution consists of a pay-as-you-go solar electricity generation and storage kit, known as the Izuba.Box.[12] With the Izuba.Box, customers no longer have to travel regularly back to the Energy Kiosk for electricity. Solar panels are installed on the customer's roof, and is connected to a sealed box which contains a large battery box. The attached large battery box allows a regulated power output and access to electricity during dark hours.

Figure 2.4: First Generation Izuba.Box deplyed in Minazi, Northern Rwanda

With the customers not returning to Energy Kiosks, the battery boxes are not hired out like the battery boxes are. The high capital costs of the independent solar system is spread over a typically two year rent-to-own payment plan using a mobile payment system.

It is hoped that the Standalone Solution and additional generation from the Energy Kiosk can be complement the battery boxes in circulation to provide a continuous access to electricity to all households in the village.

Micro-Grid

With the recent completion of a hydro-electric kiosk, e.quinox for the first time has a kiosk with access to an always-on generator. However, with a limited number of battery boxes in circulation, the constant water flow available during the off-peak hours leaves an excess in generation capacity. To improve utilisation of the generator, e.quinox has recently started conducting a feasibility study into constructing a transmission line and a distribution network which will serve communities near the kiosk.

Preliminary surveys conducted in the nearest villages to the kiosk indicates the demand could exceed the amount of excess power generated by the kiosk, making it an ideal scenario for this project. The result of this project can be used in conjunction with other e.quinox feasibility studies to implement a novel type of Micro-Grid with compulsory demand side management.

Rugaragara Falls as a Simulation Scenario

In developing countries such as Rwanda, poor communities with no access to grid electricity are often in isolated locations such as Rugaragara Falls. The local District Sector office estimates that Grid electricity access won't be available in Rugaragara before 2020. In the case of Rugaragara Falls and much of rural Rwanda, the difficult terrain and underdeveloped transport links make fuel expensive to obtain. Therefore instead of non-renewable sources of generation, locals often depend on renewable sources of electricity such as solar and wind. However, renewable generation can be highly variable in the amounts of electricity that is produced due to external factors such as weather, time of day and the season. Without access to redundancies received from the national electricity grid, it can be beneficial for Prosumers within these areas to form a micro-grid to reduce the generation requirement of individual households for continuous electricity access.

Requirements and Design 3

Requirements

To be able to simulate a dCES operating within the Rugaragara Falls community, a model of a rural electricity network that is disconnected from the grid would need to be constructed. With that in mind, the simulator was required to have the following features:

- Multiple Forms of Generation: Renewable and Non-renewable generators which can operate continuously or discontinuously
- Realistic Generator Models: Programmable variable generation power output to simulate wind and solar power
- Multiple demand centres: the simulation will be of one or more communities operating with a number of households/businesses requiring electricity
- Self organising by the system to allocate the available power fairly to all users
- Presage 2: The simulation will be programmed in Java using Presage 2

As this is a simulator implemented in Presage 2, there were no specific requirements which must be adhered to with regards to speed, portability and performance.

Model Assumptions

For the purpose of this project, all participants in the micro-grid are assumed to be prosumers. This means all participants are expected to contribute and consume electricity. Some important assumptions will be made about how the system operates in order to simplify the implementation of both the model and the micro grid:

- No losses would be incurred by the network
- All load on the network will be purely resistive
- All generation will act as negative load
- Only basic appliances such as phones, lights and fridges will be connected to the vast majority of households
- All prosumer households will have a battery
- demand requests are made automatically based on their consumption at the time without user intervention, and therefore there cheating will not take place
- A household will have multiple power outlets. Consumption is measured by the amount of power required to power all of these power outlets with appliances connected at full power

- If allocated power is below the required amount to power all of the outlets, some of the outlets will be automatically turned off by a automatic load shedding mechanism. The order which the outlets are turned off can be set by the user so the user can make sure the most important appliance is connected to the outlet that will be the last to switch off

Model Design

To allow the Community Energy System to work as a holonic system, the micro-grid has been designed akin to the simplified model in Figure 3.1. The Agents in this case are represented by the nodes labeled A-H, with various demand/generation equipment connected to the Agents. The Agents are connected to a Virtual Agent or a Parent Agent represented by the single black dot that all Agents on the periphery are connected to in Figure 3.1. As the simulator is a Multi-Agent System, A Virtual Agent is employed to easily represent group demand and group generation of a small community.

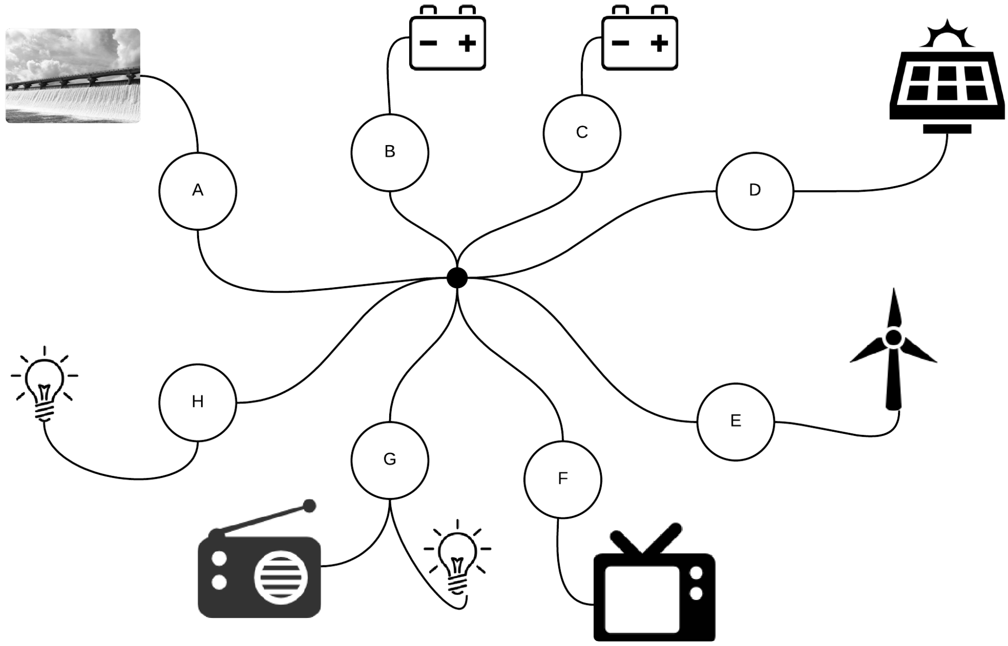


Figure 3.1: A simplified model diagram

Agents that group together are connected via a central Virtual Agent to allow the agents to form a community. These communities can further connected to another Virtual Agent to form even larger communities demonstrated in Figure 3.2.

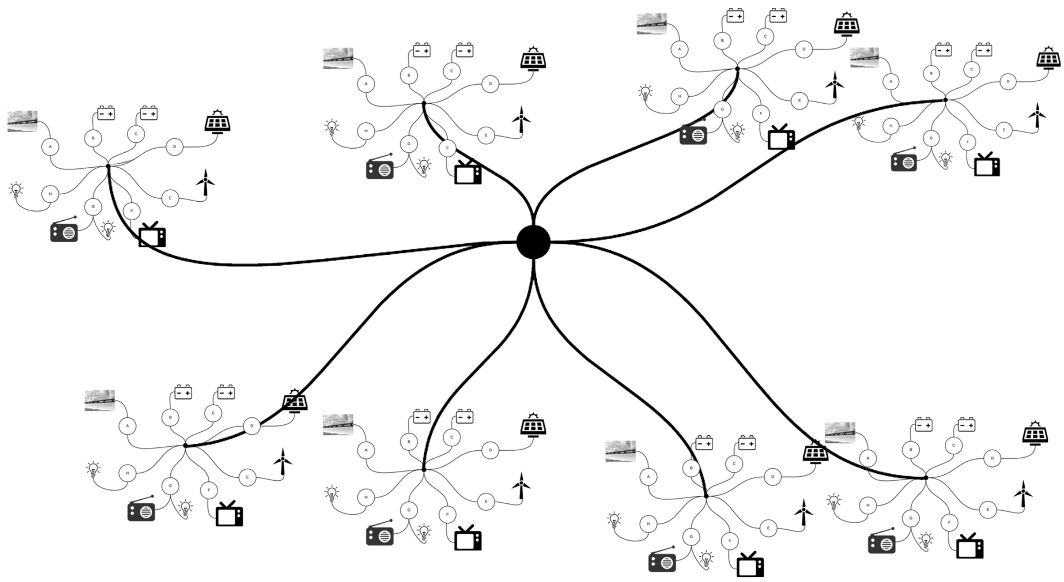


Figure 3.2: A simplified network model diagram

Types of Agents

Agents in a MAS are the actors which can act on the environment. Outlined below are the Agents that will be implemented as part of the simulator.

Virtual Agents represent all connected sub-communities or Agents in a community of Virtual Agents. These Agents will be responsible for enforcing quotas and contribution on connected sub-communities or Agents and collecting generation for the Common Pool.

Supervisor Agents are a special type of Virtual Agent whose responsibility is to enforce contribution and quotas on connected Virtual Agents. There should only be one Supervisor Agent in a simulation.

Prosumer Agents represent the most "elementary" system which includes generators, households, businesses and other demand centres.

Agent Properties

Agent Properties represent the information that is to be relayed to the Community or Institution that the Agent is part of. In the simulator, the communities are represented by Virtual Agents.

Demand - is a property which represents aggregated electricity consumption at a point of connection. Assumed demand curves will be produced from survey data of potential customers in the area for the initial testing. Should the survey data be not available for the area, a reasonable approximation will be made based on the predicted usage habits of the wider local population.

For Virtual Agents, this property represents the aggregated demand of all connected sub-communities or Agents and will not have any associated demand profiles.

Generation - is a property which will allow all agents to generate power. For Virtual Agents, this property represents the aggregated Generation of all connected sub-communities or Agents. Four types of generator properties will be implemented in this simulation model for Prosumer Agents:

- Hydro-electric - a constant source of energy based on a mixture of historical data and projections.
- Solar - a source of energy following the typical output profile of a solar panel connected to households.
- Wind - a source of energy which will be highly variable in output.

With the power output of renewable sources of energy such as wind and solar being non-predictable in nature, one of two approaches will be used to model these sources:

- A probabilistic generating factor is applied to the wind turbines. A constant amount of power each is assumed to be produced for a random amount of time at a random hour of the day that is to be determined. This can also happen for a random amount of times.
- An average generation power output curve is used for solar panels

A variation of both approaches are currently in use by distribution networks in the UK for assessing network congestion.[13]

Storage - Storage devices will be batteries of various types that can be connected to the network. For the purpose of this project, it will be assumed that all Prosumer Agents will have one of these to allow allocation of electricity on a hourly basis.

Simulation with Presage 2

For the initial implementation and testing, the model will only contain two levels of Aggregation designed similar to figure 3.3 because the micro-grid in Rugaragara Falls will be only connecting one or two communities.

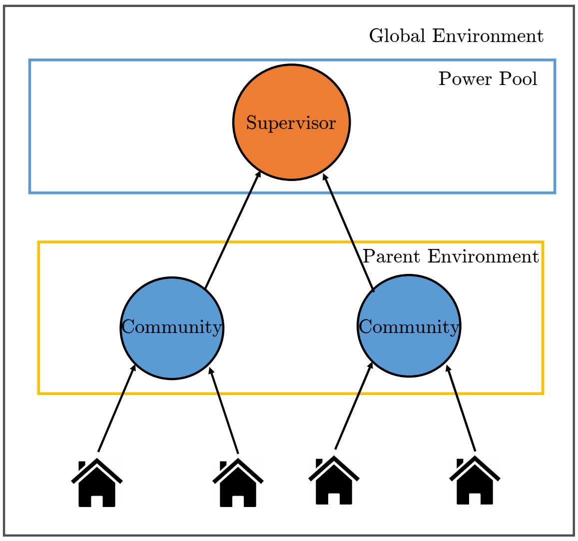


FIGURE 3.3: A simplified overview of the simulation structure

Similar to many other Multi-Agent Simulators, there can be no direct Agent-to-Agent communication. All communication must be done via the environment. As such, the environments have been designed in such a way to enable multiple levels of holonic systems that can be seen in figure 3.4.

Architecturally, Agents in the Simulation hold all of the information that is relevant to them. The Agents at the beginning of each allocation round submits their individual demand and generation request to the Virtual Agent that they

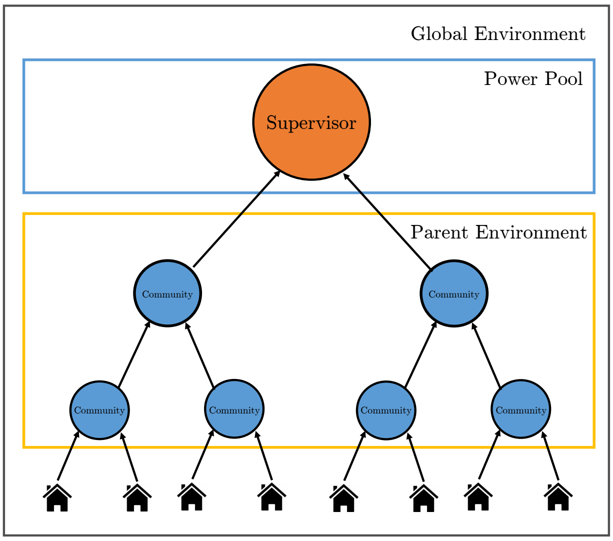


Figure 3.4: A simplified overview of the simulation structure

are connected to. The Virtual Agents can submit their Group demand and Group generation Requests (Aggregated demand and generation Requests) to other "Parent" Virtual Agents that they are connected to through the Parent Environment, and can do so recursively until the group demand and generation have been sent to the Supervisor. By making the first initial allocation, the Supervisor is able to trigger a stream of allocation making actions performed by the Virtual Agents to the Virtual Agents and other Agents that are in their holonic sub-system. During a round, Agents (or the houses in 3.3)are expected to publish their demand/generation request to the Virtual Agent that they are connected to. The Community that these parents are part of are represented by Virtual Agents.

The Virtual Agent aggregates the connected demand/generation requests and publishes this information to the Supervisor. The Supervisor aggregates the total demand/generation and appropriates the electricity fairly, and curtails the generation if there is an excess.

The basic algorithm for the simulation is outlined below in Algorithm 1:

for *each Agent* **do**
  request *Demand* and *Generation* from Virtual Agent;
end
for *each Virtual Agent* **do**
  request *Group Demand* and *Group Generation* from Supervisor Agent;
end
if *Group Generation >= Group Demand* **then**
  curtail Group Generation;
else
  allocate fairly Group Demand among Virtual Agents;
end
for *each Virtual Agent* **do**
  allocate *Group Demand* and *Group Generation* allocated from Supervisor Agent;
end

Algorithm 1: Basic Simulator Algorithm

Time-steps

With the simulation being built on Presage 2, the simulation will be run in discrete time. Currently in the UK, energy is traded in 30 minute blocks on a forward market at least one hour ahead of consumption.[14] In a similar fashion, the micro-grid will have the provision and allocation done on a hourly basis, one hour ahead of consumption. The Demand request will be made automatically based on their consumption at the time of allocation without user intervention. It is assumed that there is a device such a smart meter in place which can intelligently request Demand based on past user demand profiles and current electricity consumption.

Fair Allocation of the Common Pool Resource

With a lot of generation provided by renewable sources, there will be occurrences when the aggregated demand requests will be exceeded by the aggregated generation requests. In circumstances such as these, electricity will need to be allocated to all users fairly. Rescher had observed in 1966 that a fair allocation can be found by ranking participants according to the *Rescher's Canon of Distributive Justice*:

1. Treatment as equals
2. Treatment according to their needs
3. Treatment according to their actual productive contribution
4. Treatment according to their efforts and sacrifices
5. Treatment according to a valuation of their socially-useful services
6. Treatment according to supply and demand

7. Treatment according to their ability, merit or achievements

From Rescher's analysis, using each canon (criteria) alone as a basis for a claim was inadequate as it was deemed unfair. Instead, for a fair system, all canons need to be employed in conjunction with each other. In a democratic and self-organising system, the importance of each canon in the eventual allocation should be decided by the system participants. In this simulation, how the participants are ranked according to the canons are based on a similar allocation system demonstrated in *LPG'*[15] , and are defined as below:

1. ***The canon of equality*** - Agents can be ranked in increasing order of their average allocations.

2. ***The canon of needs*** - Assuming similar agents will have similar demand requests, the Agent most in need is the one that has made the smallest demand request. Therefore Agents can be ranked in decreasing order of their average demand requests.

3. ***The canon of productivity*** - Agents can be ranked in decreasing order of their economic output.

4. ***The canon of effort*** - All Agents have made similar sacrifices and will gain a similar amount by being a participant of the micro-grid. Therefore, this canon is not represented in the simulation.

5. ***The canon of social utility*** - Some Agents will provide services to the community such as providing the venue and equipment to watch a UEFA Champions League football game (something that occurs surprisingly often in rural Rwanda). Agents will be given a score of social utility based on the services they provide, and ranked in decreasing order.

6. ***The canon of supply and demand*** - When allocations are required, the Agents which are most "in demand" are those Agents who contribute the most to the common pool. Therefore to encourage contribution, Agents are ranked in decreasing order of their average provision.

7. ***The canon of merits and achievements*** - The merits and achievements of the Agents can't be accurately depicted in this simulation, and are therefore excluded.

8. ***Weighting of each canon*** - The importance placed in each of the aforementioned canons in deciding the ranking of the legitimate claims made by a participant is decided by a vote. At the end of each allocation round, participants submit a vote to help decide the importance of each of the canons for the next round.

9. ***Voting and multiple claims*** - To allow multiple claims to be allocated fairly, each canon and each Agent is treated as a voter according to the Borda count protocol. During every allocation round, each Agent is given a number of points by each of the canons. Under the Borda count protocol, the number of

points received by each Agent corresponds to *n-r+1* (with n being the number of Agents being ranked, and r where the Agent placed in the ranking). For example, in a ranking of *n=5* Agents, the Agent ranked first receives *n-0*=5 points, and last ranked Agent receiving *n-4*=1 point. Similarly, at the end of each allocation round, Agents rank the canons in a similar fashion to decide the weighting of each of the canons according to the Borda count voting protocol.

Implementation 4

In this section, how the design is implemented will be described in detail.

Discrete Time Simulation Implementation Overview

In the simulation, demand and generation requests are made and satisfied on a hourly basis. Due to the nature of Presage 2, actions and requests happen in discrete time steps. In this simulation, each hour has to be split into 5 discrete time-steps due to the parallel and random execution of actions and the discrete time nature of Presage 2. What actions are performed in each of the time steps are outlined below:

1. During the first time step, Prosumer Agents submit their generation and demand requests to the Virtual Agent (Community) they are part of.
2. During the second time step, Virtual Agents (Communities) aggregate the generation and demand requests received and submit their generation and demand requests to the Supervisor Agent.
3. During the third time step, the Supervisor Agent gathers the total demand and generation requests and subsequently allocates the demand and generation to the Virtual Agents.
4. During the fourth time step, the Virtual Agents (Communities) receive the allocation given by the supervisor, and allocates that appropriately to the Agents.
5. During the 5th time step, the Agents appropriates the demand and generation allocation that has been given to them.

Agent Class Structure and Implementation

Agents in Presage 2 are created by extending the *AbstractParticipant* class as shown in the *Unified Modeling Language* (UML) diagram in figure 4.1.

A Virtual Agent (*ParentAgent* class) represents connected Prosumer Agents. It is required to:

- Keep track of the Supervisor Agent, and communicate with it the group demand and generation requests
- Keep track of the Prosumer Agents that it is connected to and submit demand and generation requests on their behalf
- Submit generation and demand requests to the Supervisor Agent it is connected to
- Enforce demand and generation quotas on connected Prosumer Agents

To perform the duties outlined above, the following methods are implemented in the *ParentAgent* class:

- The *constructor* of this object is responsible for keeping track of which Agent is the Supervisor Agent. The *constructor* is called when this object is created in the initial Simulation set-up as part of the Agent creation and initialisation process.

- *addChild()* method is used for keeping track of the Agents. This is also called during the initial Simulation set-up as part of the Agent creation and initialisation process.

- *step(int)* method is called by Presage during the simulation to allow the Agent to act on the environment. demand and generation requests are sent to the supervisor, and allocation of demand and generation are also made to the Agents when this method is called automatically by Presage during a simulation.

The class *MasterAgent* is how a Supervisor Agent is defined within the Simulator. As a special type of Virtual Agent, it extends from *ParentAgent* class and is required to only do two things:

- Keep track of the Virtual Agents in the community it is responsible for

- Allocating the correct amount of demand and generation to the Virtual Agents

To satisfy those requirements the following methods in the class are implemented:

- *addChild()* method is used for keeping track of the Virtual Agents. This is called during the initial Simulation set-up as part of the Agent creation and initialisation process.

- *step(int)* method is called by Presage during the simulation to allow the Agent to act on the environment. Demand and generation are allocated to the Virtual Agents when this method is called automatically by Presage

Prosumer Agents are Virtual Agents but with no aggregation function, as a Prosumer Agent is the most elementary Agent is the most elementary Agent or sub-community in this holonic system simulation. A Prosumer Agent is required to:

- Keep track of which Agent is the Virtual Agent

- Submit generation and demand requests to the Virtual Agent it is connected to

- Appropriate the allocated amount of generation and demand allocated

The Prosumer Agent is implemented with the following methods to allow it to perform the requirements outlined above:

- The *constructor* of this object is responsible for keeping track of which Agent is its *ParentAgent*. The *constructor* is called when this object is created in the initial Simulation set-up as part of the Agent creation and initialisation process.
- *addChild()* method is used for keeping track of the Agents. This is called during the initial Simulation set-up as part of the Agent creation and initialisation process.
- *step(int)* method is called by Presage during the simulation to allow the Agent to act on the environment. demand and generation requests are sent to the supervisor, and allocation of demand and generation are also made to the Agents when this method is called automatically by Presage.
- *addProductivity(), addSocialUtility(), addProfileHourly()* are the methods that are called during the initial Simulation set-up as part of the Agent creation and initialisation process to define the properties of this Agent.

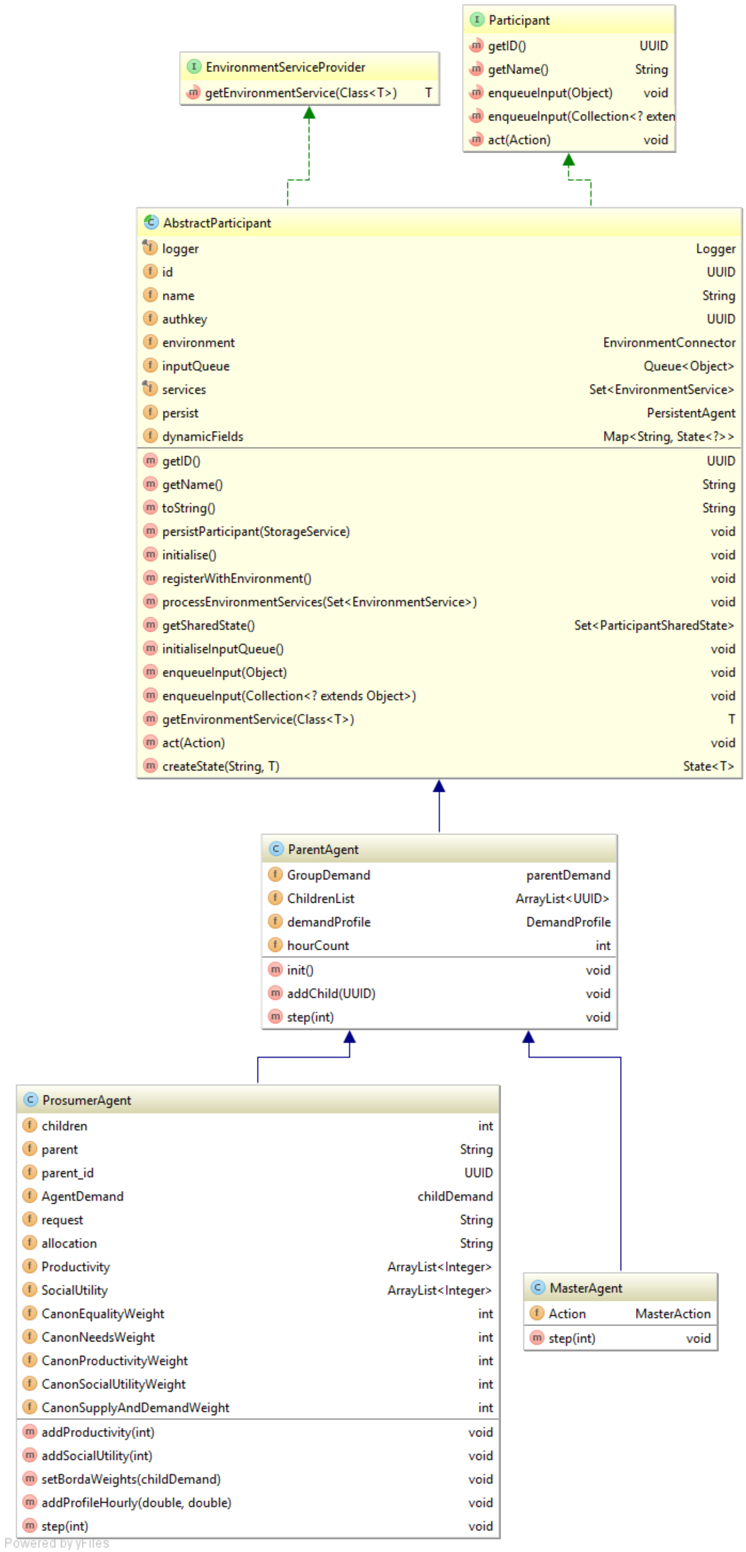


FIGURE 4.1: Agent UML Diagram

Demand Action

Agents act on Environments and "Shared States" by performing *Actions*. *Actions* are implemented by extending the Java interface *Action* from Presage 2. In the context of this simulation, an *Action* would be a demand/generation request or a demand/generation allocation. As generation can be modeled as a negative demand, a single *Action* such as the *Demand Action* can be defined to represent both demand requests/quotas and generation requests/dispatch. *childDemand* and *parentDemand* are special instances of *Demand*, and are responsible for representing demand/generation requests from Prosumer Agents and Virtual Agents respectively. The UML diagram of all of the *Actions* in this simulation is defined in the UML diagram in figure 4.2.

Action Handlers

Demand Handlers

To enable the Environment to be able to process the Action requests, *Action Handlers* need to be created to tell the simulation how to deal with Actions from Agents. In Presage 2, *Action Handlers* are created by extending the implementing the Java interface *ActionHandler*. The superclass *Demand* isn't used by any of the Agents, so there are three *Action Handlers*: *ChildDemandHandler*, *ParentDemandHandler* and *MasterActionHandler*. Depending on which the time step in the simulation hour it is, *ChildDemandHandler* and *ParentDemandHandler Action Handlers* store or retrieve requests and allocations accordingly. The *MasterActionHandler* is responsible for aggregating all demand and generation requests from all Virtual Agents and allocating that in time step 3. How these *Action Handlers* relate to each other is described in the UML diagram in figure 4.3.

Master Action Handler

A special case of the *Action Handler* is the *MasterActionHandler*. Unlike the other action handlers which access and data in one localised Environment Services, and methods in another Environment Service to allocate and st, this *Action Handlers* is a *Action-Environment Service* hybrid, which access and stores data in one localised Environment Service

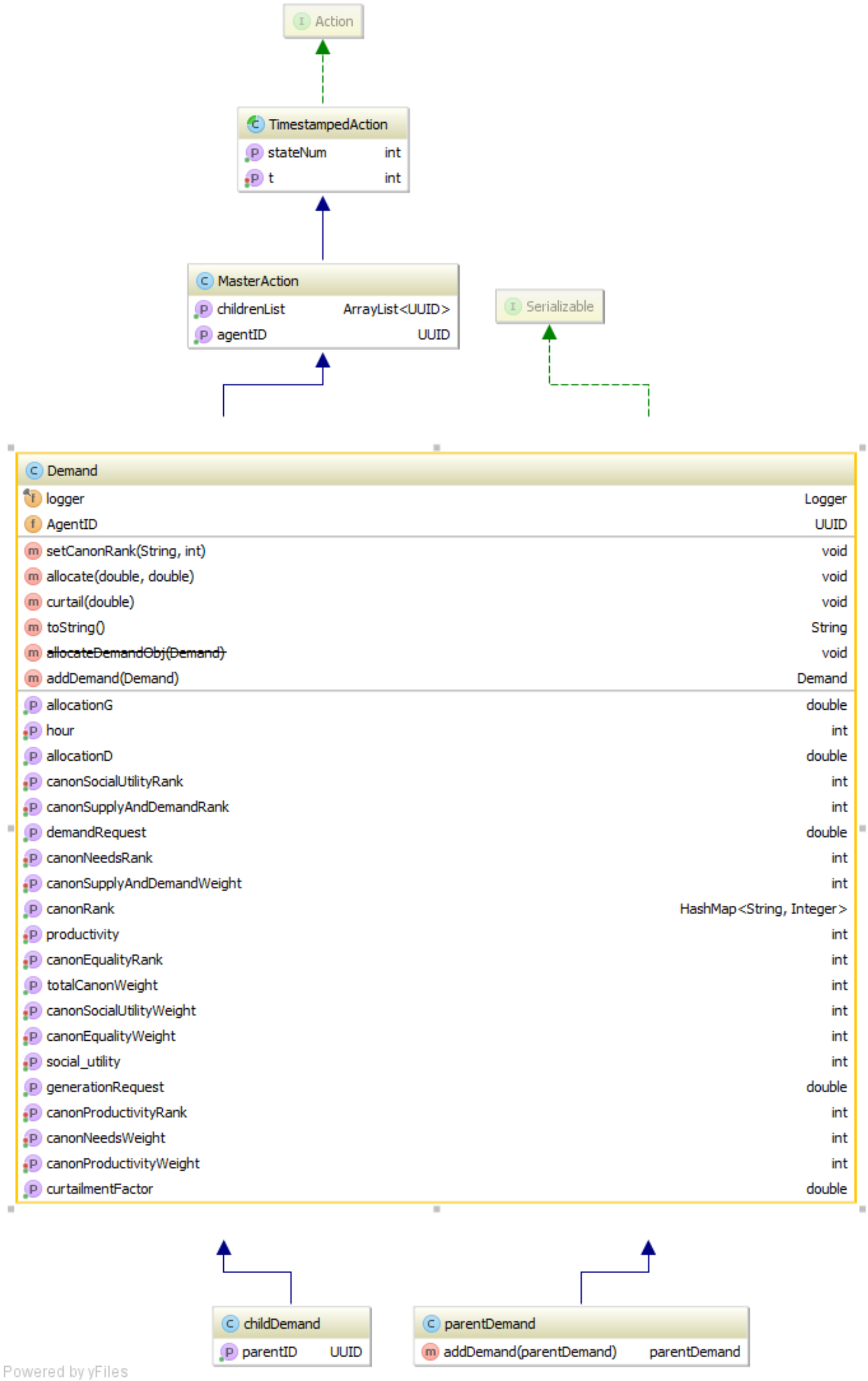


Figure 4.2: Actions UML Diagram

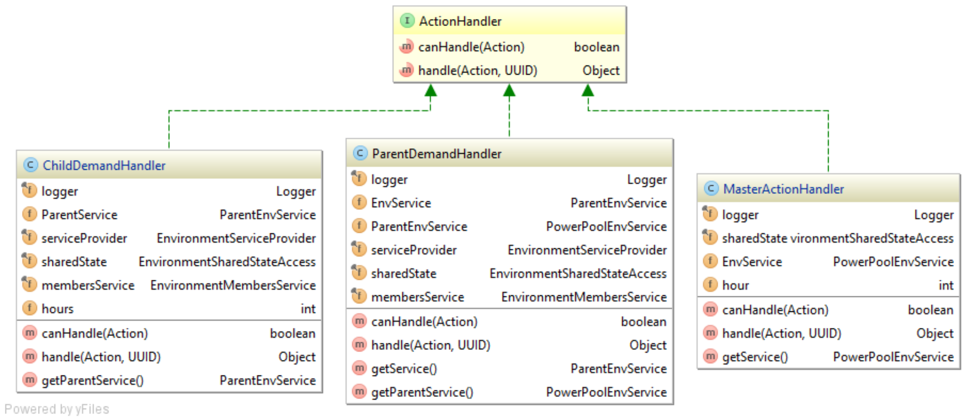


FIGURE 4.3: Action Handler UML Diagram

Environment Services Implementation

All Agents perform actions on Environments, which contains one or more "shared states". In Presage 2, all communication that happens between Agents are done via the environment by storing the data in a "shared state". In the context of this simulation, a "shared state" would be information such as demand and generation requests or the amount available in the Common Resource Pool. A visual diagram on how the Environment classes in this simulation are implemented can be found in figures 4.4 and 4.5.

Global Environment Service

All Agents are registered to the Global Environment Service, defined by the *GlobalEnvService* class. Like all Environment service classes in Presage, this was extended from the *EnvironmentService* class built into Presage. This class contains all of the methods that are called when allocations are being made. During time step 3 of a simulated hour, the *allocate()* method is called from the *MasterActionHandler* to perform allocations on behalf of the Supervisor Agent. The demand and generation requests by all of the Virtual Agents are aggregated and stored in this Service. The *allocate()* method by default satisfies all the requirements of connected Virtual Agents if there is enough generation to support it. If there is excess generation, the excess is curtailed proportionally. If however, there isn't enough generation, method *allocate_fairly()* is called, and the Virtual Agents are ranked according to the five applicable *Rescher's Canons of Distributive Justice*. These rankings are computed by calling the methods such as *canon_of_equality()* and the allocations are subsequently stored in the *PowerPoolEnvService*. At the end of the allocation process, the data about the allocation is stored in the environment ready for time step 3 of the next simulated hour via the method *environmentStore*.

Supervisor Environment Service

The *PowerPoolEnvService* is the Environment Service accessible by both the Supervisor Agent and the Virtual Agents. The *PowerPoolEnvService* extends the *GlobalEnvService* class, and does a lot of the same things on a smaller scale. The *PowerPoolEnvService* is used to store information about the Virtual Agents, such as their aggregated demand and generation requests, and contains the same methods that are called when allocations are being made by the Virtual Agents. During time step 2 of a simulated hour, Virtual Agents sum up their Prosumer Agent demand and generation requests, and store them in *PowerPoolEnvService*. During time step 4 of a simulated hour, the *allocate()* method is called from the *parentDemandHandler*. *allocate()* method by default satisfies all the requirements of Agents if there is enough generation to support it. If there is excess generation, the excess is curtailed proportionally. In a holonic system, Agents are not able to access information concerning other Agents that it is not directly connected to. A separate Environment Service is used to prevent the Supervisor Agent from being able to access and modify "shared states" about Prosumer Agents that are not directly connected to the Supervisor Agent.

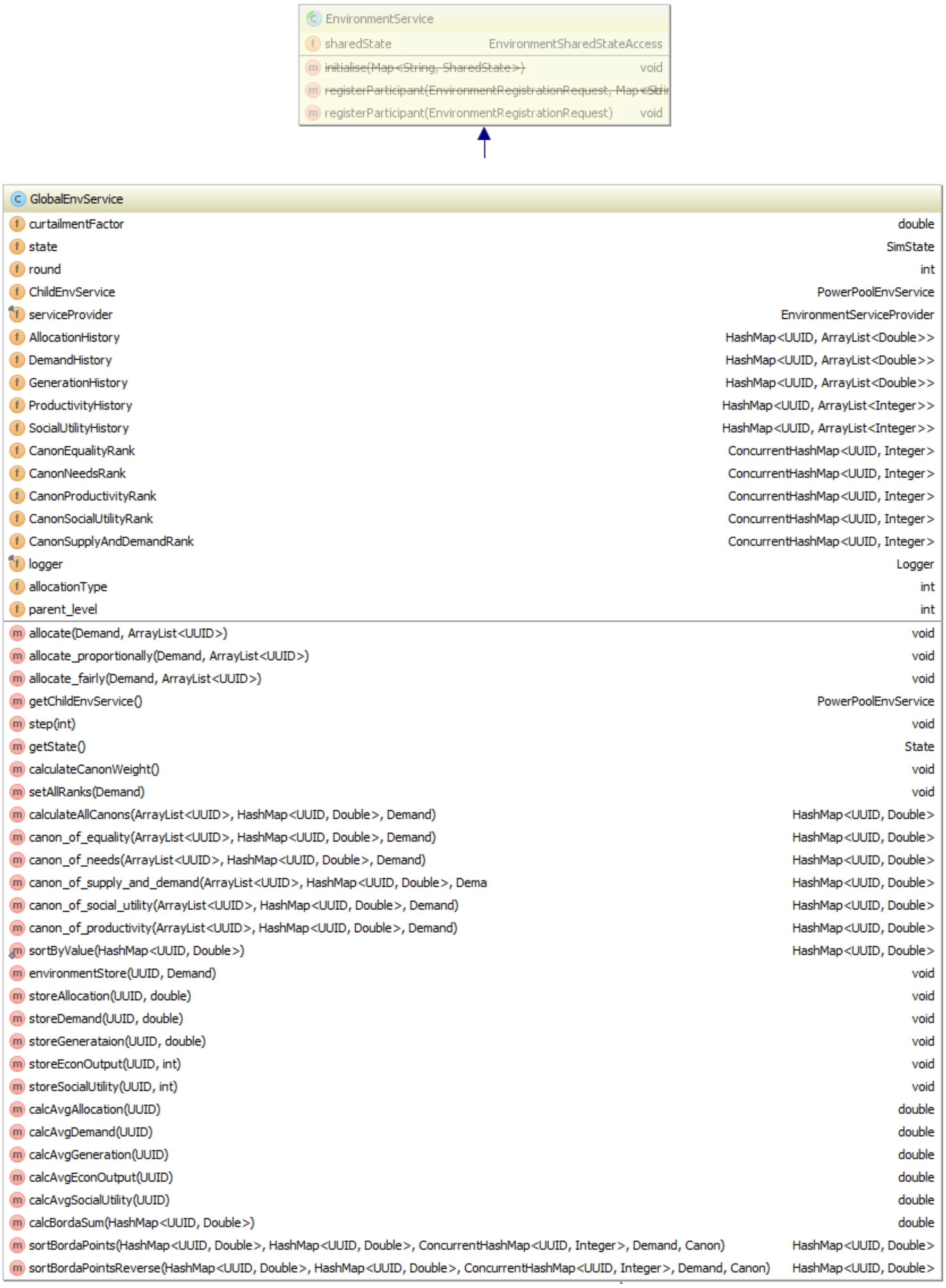


FIGURE 4.4: Environment Services UML Diagram

Similar to the *GlobalEnvService* class, if there isn't enough generation, method *allocate_fairly()* is called, and the Agents are ranked according to the five applicable *Rescher's Canons of Distributive Justice*. These rankings are computed by calling the methods such as *canon_of_equality()* which are inherited from the *GlobalEnvService*. *PowerPoolEnvService* contains "overriding" methods *allocate()* and *allocate_fairly()*

which do not inherit these from the *GlobalEnvService*, as the allocation data is required to be stored in the *ParentEnvService*. At the end of the allocation process, the data about the allocation is stored in the environment ready for time step 4 of the next simulated hour via the method *environmentStore()*.

Virtual Agent Environment Service

The *ParentEnvService* is the Environment Service accessible by both the Virtual Agents and the Prosumer Agents. The *ParentEnvService* inherits from the *PowerPoolEnvService*, and does the same things on a smaller scale. The *ParentEnvService* is used to store information about the Prosumer Agents, such as their individual demand and generation requests, and also contain information about their allocations. During time step 1 of a simulated hour, Agents store their demand and generation requests in this Environment Service as a "shared state". During time step 2 of a simulated hour, Virtual Agents aggregate the stored demand and generation requests and store them in the *PowerPoolEnvService*. During time step 5, Agents retrieve their allocations from the *ParentEnvService*.

Setting up the simulation

To set up the simulation with realistic demand and generation profiles, some measured data over a 24 hour period was used to set up the demand and generation profiles of the Agents. To simulate random and independent behaviour between Agents, each of the measured data points was randomly changed by 20% according to a normal distribution centred around the data points. Figures 4.6, 4.7 and 4.8 show a plot of the wind generation, solar generation and demand data used to set up the simulation.

Issues

Out of order parallel execution meant that Agents need to submit their individual demands to the SharedState, and have that summed at the end of each time step. It is not possible to sum the demands on the fly.

Being new to both Java and Presage presented problems of its own. It was difficult to understand how simulations could be run and therefore create our own. Some aspects of this project is similar to *LPG'*, but due to API changes the code had to be redesigned and rewritten for this project.

One action per time step meant that it takes 5 time steps to simulate one round of request and appropriation of electricity. It would therefore take 24*4 time steps to simulate a full day of requests and appropriation.

Difficulty with initialising Agents with arrays meant that Agents had to be created with no demand or generation Profiles, and the demand and generation Profiles were added one by one by using the *addProfileHourly()* method.

Figure 4.5: Environment Services UML Diagram

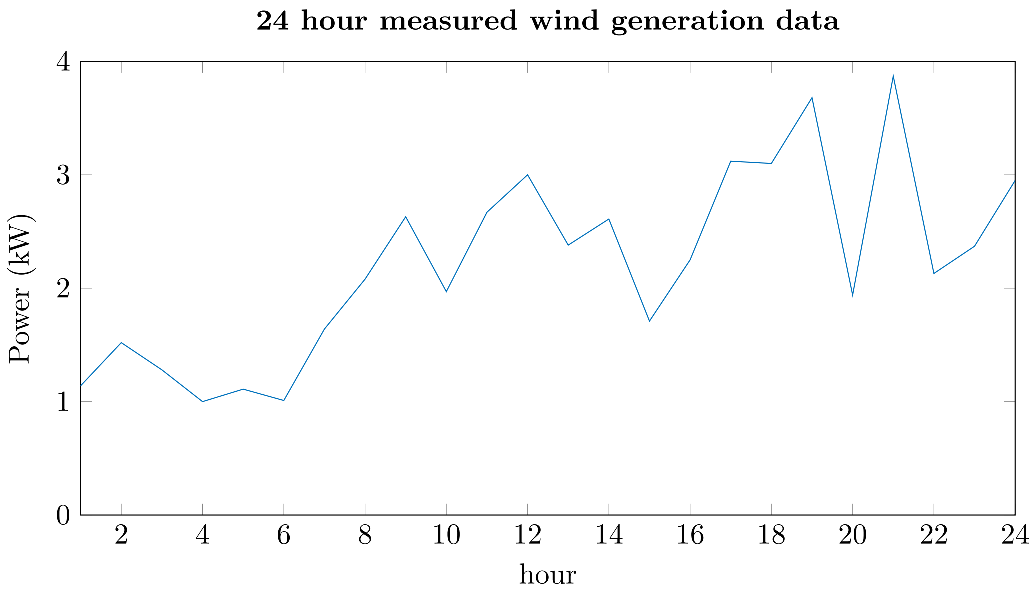


Figure 4.6: Wind Generation Profile

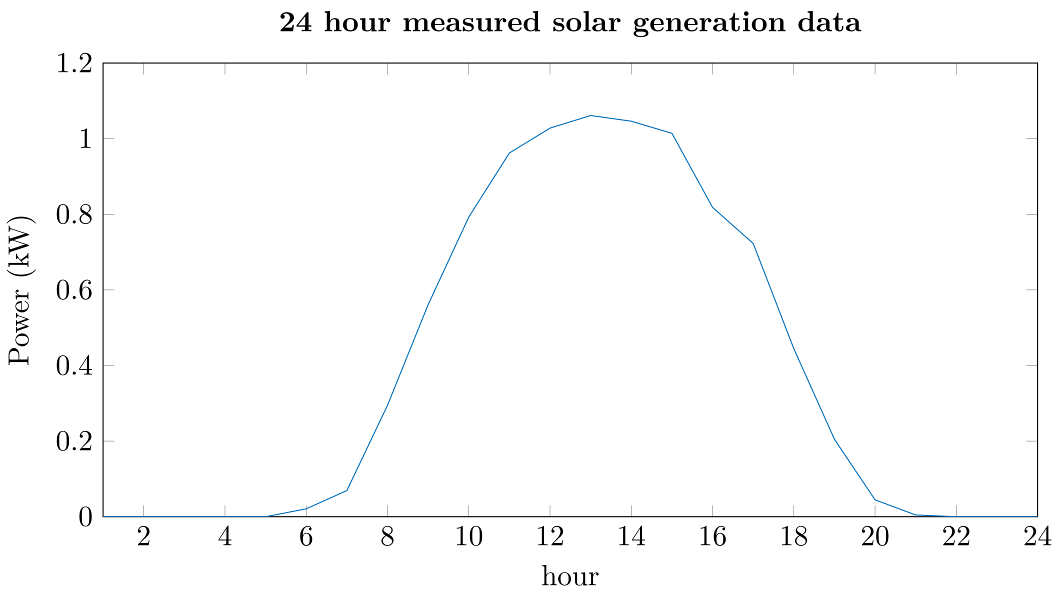


Figure 4.7: Solar Generation Profile

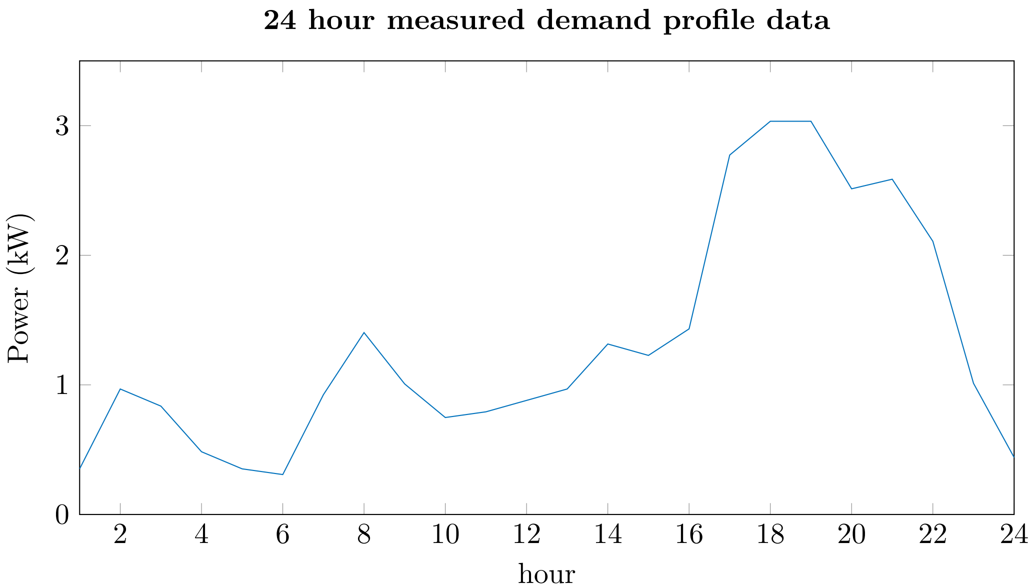


Figure 4.8: Demand Profile

5 Testing

The final product of this project should be a realistic simulator of a Decentralised Community Energy System. To produce realistic simulations, each of the following features will be tested to ensure they work individually and together:

- Agents are able to submit their requests to the Virtual Agent they are connected to.
- Virtual Agents are able to submit their requests to the Supervisor.
- Virtual Agents and their Agents are allocated an amount of generation dispatch and an appropriate amount of electricity to use

Due to time constraints, unit testing was not implemented. A series of tests was done manually by exporting the data using Microsoft Excel to prove that the simulator was working as intended. The testing has been outlined in the sections below. Each of the tests detailed below have been conducted for the following simulation cases:

- 1 Supervisor, 1 Virtual Agent and 1 Agent
- 1 Supervisor, 2 Virtual Agent and 2 Agents (Each Virtual Agent connected to 1 Agent)
- 1 Supervisor, 2 Virtual Agents and 4 Agents (Each Virtual Agent connected to 2 Agents)
- 1 Supervisor, 5 Virtual Agents and 25 Agents (Each Virtual Agent connected to 5 Agents)

Testing Request Submission

To test that Agents are able to submit their demand and generation requests, the individual demand and generation request of each Agent was recorded in a CSV file. The allocations in the CSV file was summed and compared to that of the computed Global demand and generation request.

An example of this test can be seen in figure 5.1, where a test of the contribution of 25 Agents during hour 23 of a simulation was conducted.

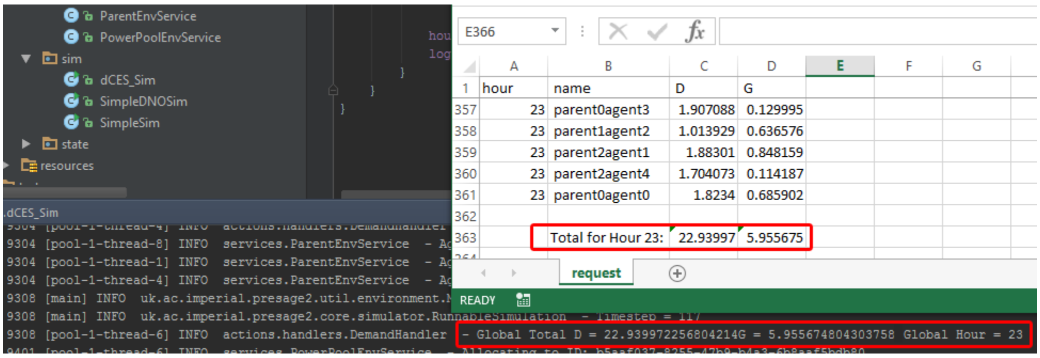


Figure 5.1: Testing the contribution at hour 23 of the simulation for 25 Agents

Testing Allocation Algorithm

When the global aggregated generation requests exceed the global aggregated demand requests, all Agents are expected to receive their requests. Globally, generation is curtailed to not exceed the total demand request by curtailing generation proportionally for everyone. In this case, we should expect all Agents to be allocated less generation than they have offered, and all Agents to be allocated their demand requests. An example of this test being passed can be seen in figure 5.2.

When the global aggregated demand request exceed the global demand request, the "fair allocation method" will need to be used. To ensure this is working as intended, we need to ensure the following tests are passed:

1. Total allocated demand is equal to total allocated Generation
2. Total allocated generation is equal to the total generation request
3. Demand allocation proportion is correct
4. Borda ranking is correct for each of the canons
5. Borda voting by the Agents are correct
6. Borda voting is taken into consideration in the next round

Tests 1 and 2

To check that total allocated demand is equal to total generation dispatch, and the total generation dispatch is equal to the total generation request, the total generation Request, generation dispatch and demand allocation were summed up for each hour of the simulation. Figure 5.3 shows the data for one of such tests. The summation at the bottom shows that the total allocated demand (Allocated D) and generation dispatch (Allocated G) is equal to the total requested generation (Request G).

I13 =IF(F13<D13, "yes", "no")

	A	B	C	D	E	F	G	H	I	J
1	hour	name	Request D	Request G	Allocated D	Allocated G		Request D == Request G?	Is Generation Curtailed?	
2	0	parent0agent0	1.592401359	2.311111995	1.592401359	1.453660808		yes	yes	
3	1	parent0agent0	1.995160167	2.229004158	1.995160167	1.367390779		yes	yes	
4	2	parent0agent0	1.769509059	2.445681676	1.769509059	1.39819329		yes	yes	
5	3	parent0agent0	1.648368572	2.823361438	1.648368572	1.624280561		yes	yes	
6	4	parent0agent0	1.698531447	2.098485929	1.698531447	1.275250851		yes	yes	
7	5	parent0agent0	1.245672833	2.868123024	1.245672833	1.793544947		yes	yes	
8	6	parent0agent0	1.198098745	2.056996387	1.198098745	1.300946125		yes	yes	
9	7	parent0agent0	1.844929509	2.098140893	1.844929509	1.350585918		yes	yes	
10	8	parent0agent0	1.558576104	2.897400312	1.558576104	1.833657969		yes	yes	
11	9	parent0agent0	1.030802763	2.149425156	1.030802763	1.31492169		yes	yes	
12	10	parent0agent0	1.350170927	2.778337346	1.350170927	1.64977437		yes	yes	
13	11	parent0agent0	1.910639238	2.753283402	1.910639238	1.72097032		yes	yes	
14	12	parent0agent0	1.491592782	2.606799055	1.491592782	1.545341026		yes	yes	
15	13	parent0agent0	1.924255335	2.006983643	1.924255335	1.156448361		yes	yes	
16	14	parent0agent0	1.254589004	2.313282333	1.254589004	1.379315387		yes	yes	
17	15	parent0agent0	1.815323723	2.306176827	1.815323723	1.501893006		yes	yes	
18	16	parent0agent0	1.857218884	2.114889386	1.857218884	1.291346395		yes	yes	
19	17	parent0agent0	1.480209778	2.681540784	1.480209778	1.657888309		yes	yes	
20	18	parent0agent0	1.307517911	2.067918309	1.307517911	1.110023994		yes	yes	
21	19	parent0agent0	1.309522885	2.220584418	1.309522885	1.261815735		yes	yes	
22	20	parent0agent0	1.199242606	2.915590237	1.199242606	1.691612538		yes	yes	
23	21	parent0agent0	1.068311668	2.975277496	1.068311668	1.634058805		yes	yes	
24	22	parent0agent0	1.232676004	2.221910139	1.232676004	1.414685194		yes	yes	
25	23	parent0agent0	1.322745932	2.257147764	1.322745932	1.391876975		yes	yes	
26	0	parent0agent1	1.983027954	2.950334956	1.983027954	1.85572413		yes	yes	
27	1	parent0agent1	1.990649072	2.777340226	1.990649072	1.703769552		yes	yes	
28	2	parent0agent1	1.091557528	2.654721342	1.091557528	1.517701017		yes	yes	
29	3	parent0agent1	1.027610958	2.316036187	1.027610958	1.332416214		yes	yes	
30	4	parent0agent1	1.779710396	2.701054557	1.779710396	1.641432079		yes	yes	
31	5	parent0agent1	1.663057514	2.463041132	1.663057514	1.54023204		yes	yes	
32	6	parent0agent1	1.117376148	2.399024928	1.117376148	1.51726187		yes	yes	
33	7	parent0agent1	1.849330187	2.827728518	1.849330187	1.820225863		yes	yes	
34	8	parent0agent1	1.582549611	2.086502422	1.582549611	1.320470553		yes	yes	
35	9	parent0agent1	1.549677387	2.857919857	1.549677387	1.7483469		yes	yes	
36	10	parent0agent1	1.123213878	2.235238446	1.123213878	1.327282702		yes	yes	
37	11	parent0agent1	1.120824313	2.575954255	1.120824313	1.61012877		yes	yes	
38	12	parent0agent1	1.469979531	2.822010251	1.469979531	1.672920744		yes	yes	
39	13	parent0agent1	1.100733124	2.147354561	1.100733124	1.23733179		yes	yes	

FIGURE 5.2: Testing the allocation of 25 Agents when total Generation Request Exceeds total Demand Request

Test 3

To test that the allocation proportion is correct, Borda ranking and point allocation for each Agent under each of the canons were checked to be correct. Figure 5.4 shows the previous requests and allocations made by Agents under Virtual Agent *Parent0* in one of these tests. The data for the requests and allocations were gathered from *"allocation.csv"* and *"request.csv"* , which are generated when the simulation is running. Figure 5.6 shows the ranking that is based on historical data of the Agents, which matches with what the data shows in figure 5.5.

With the Ranking checked to be correct, the Borda Point allocation for each of the Agents were also checked. The Borda point allocation for the example dataset above can be found in figure 5.7. The allocations according to the weighted Borda point allocation can be found in figure 5.8.

	A	B	C	D	E	F	G	H	I
1	hour	name	Request D	Request G	Productivity	Social Utility	Allocated D	Allocated G	Satisfaction
352	14	parent0agent0	1.472868	1.71	1	2	1.062328767	1.71	0.721265427
353	14	parent0agent1	1.472868	1.71	4	4	1.203972603	1.71	0.817434151
354	14	parent0agent2	1.472868	1.014166667	0	4	0.708219178	1.014166667	0.480843618
355	14	parent0agent3	1.472868	1.014166667	0	3	1.203972603	1.014166667	0.817434151
356	14	parent0agent4	1.472868	1.014166667	4	0	0.991506849	1.014166667	0.673181065
357	14	parent1agent0	1.472868	1.71	0	4	1.2925	1.71	0.877539603
358	14	parent1agent1	1.472868	1.71	2	0	1.384821429	1.71	0.940221003
359	14	parent1agent2	1.472868	1.014166667	1	0	1.107857143	1.014166667	0.752176803
360	14	parent1agent3	1.472868	1.014166667	3	4	1.384821429	1.014166667	0.940221003
361	14	parent1agent4	1.472868	1.014166667	0	0	1.2925	1.014166667	0.877539603
362	14	parent2agent0	1.472868	1.71	1	0	1.444201878	1.71	0.980537209
363	14	parent2agent1	1.472868	1.71	2	4	1.547359155	1.71	1.050575581
364	14	parent2agent2	1.472868	1.014166667	3	4	1.444201878	1.014166667	0.980537209
365	14	parent2agent3	1.472868	1.014166667	0	4	1.237887324	1.014166667	0.840460465
366	14	parent2agent4	1.472868	1.014166667	3	4	1.650516432	1.014166667	1.120613953
367	14	parent3agent0	1.472868	1.71	1	2	1.698357488	1.71	1.153095517
368	14	parent3agent1	1.472868	1.71	1	2	1.379915459	1.71	0.936890108
369	14	parent3agent2	1.472868	1.014166667	2	4	1.273768116	1.014166667	0.864821638
370	14	parent3agent3	1.472868	1.014166667	0	0	1.379915459	1.014166667	0.936890108
371	14	parent3agent4	1.472868	1.014166667	3	0	1.592210145	1.014166667	1.081027047
372	14	parent4agent0	1.472868	1.71	2	3	1.07413242	1.71	0.729279488
373	14	parent4agent1	1.472868	1.71	3	4	1.322009133	1.71	0.897574754
374	14	parent4agent2	1.472868	1.014166667	2	0	1.404634703	1.014166667	0.953673176
375	14	parent4agent3	1.472868	1.014166667	3	4	0.991506849	1.014166667	0.673181065
376	14	parent4agent4	1.472868	1.014166667	2	0	1.239383562	1.014166667	0.841476332
602									
603		Sum:	36.8217	32.3125			32.3125	32.3125	

Figure 5.3: Checking that Allocated Demand is less than total Allocated Generation

	A	B	C	D	E	F	G	H
1	hour	name	Request D	Request G	Productivity	Social Utility	Allocated D	Allocated G
2	0	parent0agent0	0.422494228	1.143316614	4	0	0.422494228	1.058670544
3	1	parent0agent0	1.167911784	1.516214737	3	0	0.60758766	1.516214737
4	0	parent0agent1	0.422841507	1.139822124	0	0	0.422841507	1.05543477
5	1	parent0agent1	1.158514651	1.519551714	2	3	0.364552596	1.519551714
6	0	parent0agent2	0.422375141	0	0	4	0.422375141	0
7	1	parent0agent2	1.163226822	0	2	2	0.60758766	0
8	0	parent0agent3	0.42307295	0	2	3	0.42307295	0
9	1	parent0agent3	1.16789018	0	0	1	0.394931979	0
10	0	parent0agent4	0.42266157	0	1	2	0.42266157	0
11	1	parent0agent4	1.159284888	0	1	4	0.455690745	0

Figure 5.4: Historical Parent0 Agent requests and allocations

	Average Allocation	Demand Req	Productivity	Social Utility	Supply and Demand
parent0agent0	0.515040944	0.795203006	7	0	1.329765675
parent0agent1	0.393697052	0.790678079	2	3	1.329686919
parent0agent2	0.514981401	0.792800982	2	6	0
parent0agent3	0.409002465	0.795481565	2	4	0
parent0agent4	0.439176157	0.790973229	2	6	0

Figure 5.5: Historical Parent0 Agent requests and allocations data processed

	hour	name	id	CanonEqualityRar	CanonNeedsRar	CanonProductivityRar	CanonSocialUtilityRar	CanonSupplyAndDemandRar
58	2	parent0agent0	dd98f973-	5	4	1	5	1
65	2	parent0agent1	b72cd33d-	1	1	2	4	2
68	2	parent0agent2	eea2a064-	4	3	2	1	3
71	2	parent0agent3	2836b64b-	2	5	2	3	3
76	2	parent0agent4	f0495904-(	3	2	2	1	3
502								

Figure 5.6: Parent0 Agent ranking

	hour	canon	agent	id	Borda Pts	Borda pts norm	Borda Proportion	Borda Votes for Canon	Total Borda Votes
442	2	equality	parent0agent0	dd98f973-3eaf-45ab-a149-f45c7d162d22	1	0.1875	0.1875	15	80
444	2	equality	parent0agent1	b72cd33d-1e34-441f-af01-d2235c52858e	5	0.9375	0.1875	15	80
446	2	equality	parent0agent2	eea2a064-62c7-4b4c-a07d-58b5a5b1000f	2	0.375	0.1875	15	80
448	2	equality	parent0agent3	2836b64b-1cba-48df-a7e8-93cf1f128248	4	0.75	0.1875	15	80
450	2	equality	parent0agent4	f0495904-0e2b-4fd9-a5e8-8ee92ba0c87c	3	0.5625	0.1875	15	80
453	2	needs	parent0agent0	dd98f973-3eaf-45ab-a149-f45c7d162d22	2	0.375	0.1875	15	80
454	2	needs	parent0agent1	b72cd33d-1e34-441f-af01-d2235c52858e	5	0.9375	0.1875	15	80
457	2	needs	parent0agent2	eea2a064-62c7-4b4c-a07d-58b5a5b1000f	3	0.5625	0.1875	15	80
458	2	needs	parent0agent3	2836b64b-1cba-48df-a7e8-93cf1f128248	1	0.1875	0.1875	15	80
461	2	needs	parent0agent4	f0495904-0e2b-4fd9-a5e8-8ee92ba0c87c	4	0.75	0.1875	15	80
931	2	productivity	parent0agent0	dd98f973-3eaf-45ab-a149-f45c7d162d22	5	1	0.2	16	80
933	2	productivity	parent0agent1	b72cd33d-1e34-441f-af01-d2235c52858e	4	0.8	0.2	16	80
939	2	productivity	parent0agent2	eea2a064-62c7-4b4c-a07d-58b5a5b1000f	4	0.8	0.2	16	80
944	2	productivity	parent0agent3	2836b64b-1cba-48df-a7e8-93cf1f128248	4	0.8	0.2	16	80
949	2	productivity	parent0agent4	f0495904-0e2b-4fd9-a5e8-8ee92ba0c87c	4	0.8	0.2	16	80
953	2	social_utility	parent0agent0	dd98f973-3eaf-45ab-a149-f45c7d162d22	1	0.2	0.2	16	80
957	2	social_utility	parent0agent1	b72cd33d-1e34-441f-af01-d2235c52858e	2	0.4	0.2	16	80
965	2	social_utility	parent0agent2	eea2a064-62c7-4b4c-a07d-58b5a5b1000f	5	1	0.2	16	80
969	2	social_utility	parent0agent3	2836b64b-1cba-48df-a7e8-93cf1f128248	3	0.6	0.2	16	80
976	2	social_utility	parent0agent4	f0495904-0e2b-4fd9-a5e8-8ee92ba0c87c	5	1	0.2	16	80
983	2	supply_and_demand	parent0agent0	dd98f973-3eaf-45ab-a149-f45c7d162d22	5	1.125	0.225	18	80
985	2	supply_and_demand	parent0agent1	b72cd33d-1e34-441f-af01-d2235c52858e	4	0.9	0.225	18	80
994	2	supply_and_demand	parent0agent2	eea2a064-62c7-4b4c-a07d-58b5a5b1000f	3	0.675	0.225	18	80
996	2	supply_and_demand	parent0agent3	2836b64b-1cba-48df-a7e8-93cf1f128248	3	0.675	0.225	18	80
1001	2	supply_and_demand	parent0agent4	f0495904-0e2b-4fd9-a5e8-8ee92ba0c87c	3	0.675	0.225	18	80
2052									

Figure 5.7: Parent0 Hour 2 Borda Point Allocation

		Available Power	2.192999123	
		Weighted Points	%	
Agent0		2.8875	0.169106881	0.37085124
Agent1		3.975	0.232796486	0.51052249
Agent2		3.4125	0.199853587	0.43827874
Agent3		3.0125	0.176427526	0.38690541
Agent4		3.7875	0.22181552	0.48644124
Sum:		17.075		0

Figure 5.8: Parent0 Allocations according to the Borda Points

Tests 4 and 5

To check that the Borda voting mechanism is correct, the votes for each Agent was checked against where they were ranked the previous round. For the dataset used above, this can be found in figure 5.9.

	hour	agent	id	Equalityweight	ProductivityWeight	UtilityWeight	NeedsWeight	SupplyAndDemandWeight
54	2	parent0agent0	dd98f973-3eaf-45ab-a149-f45c7d162d22	1	5	1	2	5
65	2	parent0agent1	b72cd33d-1e34-441f-af01-d2235c52858e	5	4	2	5	4
74	2	parent0agent2	eea2a064-62c7-4b4c-a07d-58b5a5b1000f	2	4	5	3	3
75	2	parent0agent3	2836b64b-1cba-48df-a7e8-93cf1f128248	4	4	3	1	3
76	2	parent0agent4	f0495904-0e2b-4fd9-a5e8-8ee92ba0c87c	3	4	5	4	3
603								
604			Sum	15	21	16	15	18
605			Total:	85				
606								

FIGURE 5.9: Parent0 Agent votes for the weight of the canons in the next round

To check that the voting mechanism was fed back into the system, the aggregate votes for each of the canons for all Agents were checked. For the data set above, this can be shown in figure 5.10. *Borda Proportion* represents the percentage of votes that went into a particular canon under the Borda Voting protocol for all of the Agents.

	hour	canon	agent	id	Borda Pts	Borda pts norm	Borda Proportion	Borda Votes for Canon	Total Borda Votes
495	3	equality	parent0agent0	dd98f973-	2	0.352941176	0.176470588	15	85
765	3	needs	parent0agent0	dd98f973-	2	0.352941176	0.176470588	15	85
1030	3	productivity	parent0agent0	dd98f973-	5	1.235294118	0.247058824	21	85
1043	3	social_utility	parent0agent0	dd98f973-	1	0.188235294	0.188235294	16	85
1091	3	supply_and_demand	parent0agent0	dd98f973-	5	1.058823529	0.211764706	18	85

FIGURE 5.10: Parent0 Agent votes in hour 3

Bugs

Under Presage 2, all communication between Agents are stored in the Environment Services by the sending party and accessed by the receiving party. With all simulations under Presage 2 being multi-threaded, there are some occurrences of Agents being in different time-steps in the simulation. If a Virtual Agent was in a time-step ahead of the Agents connected to it, the Virtual Agent will sometimes access the demand and generation request data before they are ready. When this happens, it usually causes a Java concurrency error warning in the console. Fortunately, this bug rarely occurs, and usually disappears when the simulation is relaunched.

Results 6

Simulating over 24 hours

Five clusters of five Prosumer Agents (25 Prosumer Agents, 5 Virtual Agents and one Supervisor Agent) modeled on the Rugaragara Falls community were simulated over 24 hours. Each cluster had 2 Prosumers generating using wind turbines and 3 Prosumers generating using photo-voltaic cells. A copy of the complete simulation data can be found in the Appendices.

The aim of this was to investigate the effect of individual contribution on individual allocation. The *independent variables* in this simulation were the Agents' contributions to the community which can be ranked against each other by the following *Rescher's Canons of Distributive Justice*:

1. ***The canon of equality*** - Agents were ranked in increasing order of their average allocations.

2. ***The canon of needs*** - Agents were ranked in decreasing order of their average demand requests.

3. ***The canon of productivity*** - Agents were given a score of "productivity" and ranked in decreasing order of their economic output.

4. ***The canon of social utility*** - Agents were given a score of social utility based on the services they provide, and ranked in decreasing order.

5. ***The canon of supply and demand*** - To encourage contribution, Agents were ranked in decreasing order of their average provision

With no data about the economic output or social utility of the individual households in Rugaragara being available, the data had to be generated for the simulation. All Prosumer Agents were initialised with a random *Productivity* rating and a random *Social Utility* representing a household's hourly contribution to the economic or social development of the local community.

The *dependent variables* in this simulation were the allocated demand of each of the Agents. It was expected that those who contribute more would receive more under the Simulator.

Effect on Individual Allocation

As expected, with ample generation, all Prosumer Agents would receive some electricity. At no point in the simulation was there no electricity being received by the Prosumer Agents who were not generating at the time (solar powered Prosumers during the night).

During periods when the demand outstripped generation, the results were also as expected. The general trend was that an Agent which was judged to have contributed more (under the *Rescher's Canons of Distributive Justice*) would receive more allocation, resulting in a higher average satisfaction over 24 hours. As an example, the contribution according to the *Rescher's Canon of Distributive Justice* of three of the Prosumers in *Cluster0* have been included in table 6.1. In this example, it can be observed that the most satisfied *Agent3* is contributing more than the least satisfied *Agent0* across the board while maintaining a slightly smaller amount of requests over the simulated 24 hour period.

Table 6.1: Prosumer Agent contribution data for Agent 0,1,3 in Cluster0

Agent	Average Satisfaction	Average Request	Average Contribution	Average Productivity	Social Utility
0	0.712	1.575	2.214	1.750	1.667
1	0.770	1.575	2.213	2.208	2.167
3	0.676	1.577	0.379	1.958	1.875

Effect on Community (Cluster) Allocations

It is interesting to find that Agents can sometimes receive more than what they have requested, resulting in a satisfaction rating that is above 1 (see table 6.2). Opportunities such as these arise due to the holonic nature of the system, as the Supervisor Agent allocates fairly to the Virtual Agents. If as a cluster, it is contributing more than the other clusters, then at the Supervisor Allocation stage, this particular cluster will be allocated more energy. If an Agent within that Cluster is contributing more, then that Agent will be allocated more of an already big pool, which can lead to allocating more energy than is requested by the Agent.

Take the cases of Clusters 2 and 3, which both have had multiple occurrences of Agents being allocated more than requested; these clusters have had an above average contribution to the common pool which can be seen in table 6.4, and as a result have the two highest average satisfaction ratings. For both clusters, the occurrences happened within the hour before the simulation has finished, and happened at the same time. The average ranking according to the *Rescher's Canons of Distributive Justice* is a good estimate of how much they were contributing prior to hour 22. Figure 6.1 shows the relationship between average ranking according to the *Rescher's Canons of Distributive Justice* and the average Agent satisfaction within a Cluster. The trend is that, with higher ranking, comes higher average Agent satisfaction.

Table 6.2: Occurrences of when Agents are allocated more than requested

Hour	Agent	Requested Demand	Requested Generation	Productivity	Social Utility	Allocated Demand	Allocated Generation	Satisfaction
14	Cluster1Agent4	1.472	1.014	3.000	3.000	1.574	1.014	1.069
22	Cluster2Agent0	1.213	2.367	2.000	4.000	1.282	2.367	1.057
22	Cluster2Agent1	1.213	2.378	4.000	0.000	1.360	2.378	1.121
22	Cluster3Agent3	1.215	0.000	4.000	3.000	1.275	0.000	1.049
22	Cluster3Agent4	1.215	0.000	2.000	1.000	1.229	0.000	1.012
14	Cluster4Agent1	1.473	1.712	4.000	2.000	1.877	1.712	1.275

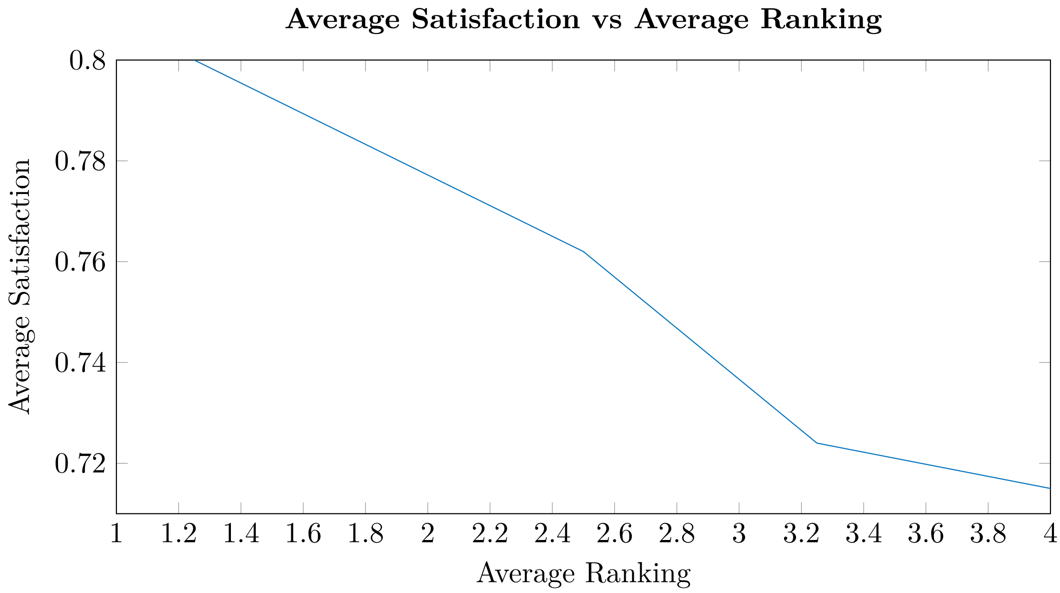


FIGURE 6.1: Variation in satisfaction with varying average contribution ranking

Table 6.3: Cluster contributions

Cluster	Average Request	Average Contribution	Average Productivity	Social Utility	Average Satisfaction
Cluster0	1.576	1.113	1.967	2.075	0.715
Cluster1	1.575	1.113	1.658	2.092	0.724
Cluster2	1.575	1.114	2.058	2.067	0.762
Cluster3	1.575	1.113	2.175	2.125	0.800
Cluster4	1.575	1.113	2.142	2.050	0.715

Implications

With a differing amount of satisfaction between clusters, the results have shown that it benefits to be a contributor in a Cluster in which all Agents contribute more than the average. In the real world, the households modeled in this simulation are likely to be in close geological proximity. Under the same weather conditions, more contribution to the common pool are likely to be as a result of better equipment which can be attributed to wealth. In the same vein, assuming that all households are equal and mechanisms are in place to prevent cheating or other selfish actions, higher productivity and social utility can also be attributed to higher wealth. For example, a communal fridge/shop/TV owned by a household can contribute to social utility, but this would only be the case if not everyone in the community is able to afford it. In the case of productivity, if a household has a spinning wheel to produce more yarn, that household is likely to be wealthier to be able to afford such equipment. With a higher than average allocation, these wealthier households are likely to be even more better off. This could result in either a wealth gap, or

Table 6.4: Ranked cluster contributions using the data from table 6.3

Cluster	Average Request	Average Contribution	Average Productivity	Social Utility	Average Ranking	Average Satisfaction
Cluster0	5	4	4	3	4	0.715
Cluster1	3	3	5	2	3.25	0.724
Cluster2	2	1	3	4	2.5	0.762
Cluster3	1	2	1	1	1.25	0.800
Cluster4	4	5	2	5	4	0.715

alternatively result in more development of the community if the extra allocation the households receive is reinvested in better equipment to generate more productivity, social utility or electricity for the larger community.

7 Evaluation and Further Work

The work that has been done so far has been only be applicable for a small rural community in Rwanda. However, with small modifications it has the scope to be developed further to a larger scale to model future smart grid networks in developed nations. Currently, much of the costs surrounding the network set-up, network maintenance have not been included in the model. In addition, reactive power flows and losses due to equipment efficiencies were also not modeled. Future work can improve the accuracy of the model by taking into consideration the afore-stated additional costs and losses. In its current iteration, excess generation is curtailed. This is the correct behaviour in poor rural communities where energy storage facilities are limited. In more advanced economies, Smart Grids will need to be able to store unused generation,[16] so the Simulator will need to be adapted to carry forward unused generation.

An attempt has been made at simulating a community energy system that has yet to be build in Rwanda, which is capable of fairly allocating energy to end users. However, due to a lack of available data on the economic output, and household information from the Rugaragara area, this data had to be generated for the simulation. More data would be gathered required to accurately implement all 8 *Rescher's Canons of Distributive Justice* and to validate the findings that were stated in the Results section.

In the Simulator's present form, only renewable sources of generation are supported fully, as these were the most prevalent type of generation in the Rugaragara Falls area. However, as renewable generation is not the only form of generation available in many rural areas, it would be more realistic to be able to support Prosumer Agents with diesel generators. Work would need to be done to reflect the cost of providing electricity from non-renewable sources to the Common Pool, and appropriately compensate the Agents who operate those generators with a fair allocation.

The current implementation of the Simulator only takes into account variables at a local level. The Simulator design and implementation can also be further expanded to simulate other variables, bottlenecks and opportunities such as government intervention, grid electricity availability and financial incentives.

Finally, the Simulator assumes no cheating would happen and does not allow cheating in any way. In reality, people will find ways to free-load off the system and bypass the safeguards in place. The design of a future Simulator should be expanded to allow cheating to take place, and implement methods to discourage cheating.

Conclusion 8

Since October, an attempt has been made to understand the concept of a Common Pool Resource, and how electricity can be a form of a Common Pool Resource for remote communities in developing countries. An attempt has been made on building a simulator of a Decentralised Community Energy System in developing countries, which can self-organise and fairly allocate electricity to end-users. The testing and results shows that the attempt has been successful in the following:

- Incorporating multiple forms of generation, by allowing generation profiles to be imported in the form of CSV files
- Incorporating realistic Generator Models by using measured data
- Creating a self-organising holonic system which is also able to allocate available power fairly to all system participants

However, there were also a number of features that have could have been implemented to make the model potentially more realistic:

- Cheating and Selfishness - The model that has been implemented assumes all parties participating in the micro-grid will provide all of their generation and only appropriate their allocated amount from the system. However, in the real world, people will be selfish and attempt to cheat to benefit themselves
- Non-renewable sources of generation - Non-renewable sources of energy have not been included in this model, as they have a fuel cost, which is difficult to model, and also make it difficult to allocate fairly.
- Cost of the system - In reality, a micro-grid costs money to set-up and maintain and no conductor is completely efficient, leading to losses in the cables, and thus additional costs. None of these have been modeled.

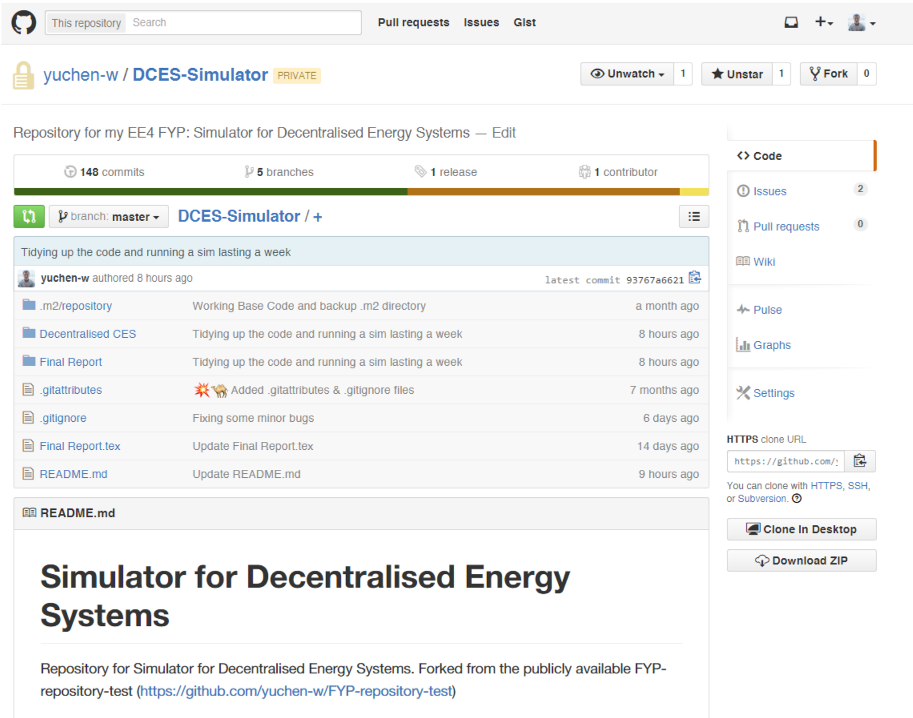
Potential Applications

One obvious use case is the feasibility study of the Rugaragara Falls micro-grid. In the immediate future, it is hoped that this tool could be used to aid the feasibility study of the micro-grid to be implemented in Rugaragara Falls. Aside from modeling small scale community micro-grids, the model can also be scaled up to model country level smart grids once cable losses, reactive power flow have been incorporated into the simulation. This Simulator of a Decentralised Community Energy System can be further developed with more features to explore possible utilisation patterns within rural communities, and also explore controls for regulating the use of electricity in developed countries.

User Guide A

The github source code repository for this project can be found at: https://github.com/yuchen-w/DCES-Simulator

The code was developed in IntelliJ Idea 14.1.3 IDE and should work with other IDEs. More detailed instructions can be found in the README.md of the repository.



Simulation Data B

B.1 24 hour simulation

A table of simulation results can be found on the next page.

hour	name	Request D	Request G	Productivity	Social Utility	Allocated D	Allocated G	Satisfaction	Average
0	Cluster0Agent0	0.422494228	1.143316614	4	0	0.422494228	1.058670544	1	0.711751
1	Cluster0Agent0	1.167911784	1.516214737	3	0	0.60758766	1.516214737	0.52023421	
2	Cluster0Agent0	1.003051052	1.282546899	0	3	0.370851243	1.282546899	0.3697232	
3	Cluster0Agent0	0.579492039	1.000144274	0	1	0.406247776	1.000144274	0.70104117	
4	Cluster0Agent0	0.424466222	1.114012341	0	2	0.424466222	1.059066759	1	
5	Cluster0Agent0	0.368596604	1.009428312	4	1	0.368596604	0.89549106	1	
6	Cluster0Agent0	1.109897279	1.636750863	3	1	0.763146232	1.636750863	0.68758276	
7	Cluster0Agent0	1.684965124	2.087999613	1	3	1.055883057	2.087999613	0.6266498	
8	Cluster0Agent0	1.211384598	2.628800099	1	3	1.211384598	2.291012313	1	
9	Cluster0Agent0	0.895838359	1.97007155	3	2	0.895838359	1.399699786	1	
10	Cluster0Agent0	0.950205614	2.652880114	0	1	0.950205614	1.532715414	1	
11	Cluster0Agent0	1.056126797	2.993145803	4	4	1.056126797	1.739586879	1	
12	Cluster0Agent0	1.159064599	2.380363643	0	4	1.159064599	1.740172864	1	
13	Cluster0Agent0	1.578704865	2.611417993	3	0	1.578704865	2.465460885	1	
14	Cluster0Agent0	1.469755578	1.708179123	0	1	0.99382175	1.708179123	0.67618165	
15	Cluster0Agent0	1.71744677	2.245731005	2	1	1.295096237	2.245731005	0.75408232	
16	Cluster0Agent0	3.321344378	3.107767309	1	3	1.339295121	3.107767309	0.40323886	
17	Cluster0Agent0	3.648540858	3.101054239	1	3	1.187783795	3.101054239	0.32555036	
18	Cluster0Agent0	3.656103312	3.6852754	0	3	1.338193935	3.6852754	0.36601644	
19	Cluster0Agent0	3.012056462	1.940486982	3	0	0.688744465	1.940486982	0.22866253	
20	Cluster0Agent0	3.099359775	3.868182807	3	3	1.350768228	3.868182807	0.43582169	
21	Cluster0Agent0	2.525458335	2.135328158	1	0	0.729849554	2.135328158	0.28899687	
22	Cluster0Agent0	1.213251218	2.365331537	2	0	0.847140384	2.365331537	0.69823988	
23	Cluster0Agent0	0.528153899	2.951840881	3	1	0.528153899	1.320781001	1	
0	Cluster0Agent1	0.422841507	1.139822124	0	0	0.422841507	1.05543477	1	0.770089
1	Cluster0Agent1	1.158514651	1.519551714	2	3	0.364552596	1.519551714	0.31467241	
2	Cluster0Agent1	1.002632619	1.281936693	4	2	0.51052249	1.281936693	0.50918201	
3	Cluster0Agent1	0.582812461	0.999447241	1	0	0.508184487	0.999447241	0.87195199	
4	Cluster0Agent1	0.422655059	1.111625147	2	4	0.422655059	1.056797306	1	
5	Cluster0Agent1	0.369038104	1.009175441	4	4	0.369038104	0.895266731	1	
6	Cluster0Agent1	1.11145139	1.641806811	4	2	0.906236151	1.641806811	0.81536283	
7	Cluster0Agent1	1.680272117	2.072088547	3	3	1.345103199	2.072088547	0.800527	
8	Cluster0Agent1	1.208853881	2.630947833	0	4	1.208853881	2.292884074	1	
9	Cluster0Agent1	0.895513431	1.969756486	4	2	0.895513431	1.399475939	1	
10	Cluster0Agent1	0.949284187	2.666186815	4	3	0.949284187	1.54040343	1	
11	Cluster0Agent1	1.0575794	2.994779569	2	0	1.0575794	1.740536408	1	
12	Cluster0Agent1	1.156947717	2.375643247	0	4	1.156947717	1.736722002	1	
13	Cluster0Agent1	1.575246901	2.609239501	3	2	1.575246901	2.463404153	1	
14	Cluster0Agent1	1.473507926	1.710848103	1	2	1.258840883	1.710848103	0.85431565	
15	Cluster0Agent1	1.717484493	2.253773125	2	1	1.557538646	2.253773125	0.90687203	
16	Cluster0Agent1	3.335695644	3.120817313	0	1	1.817614807	3.120817313	0.54489828	
17	Cluster0Agent1	3.638283572	3.093566171	3	3	1.611992293	3.093566171	0.44306395	
18	Cluster0Agent1	3.647477589	3.667723138	2	2	1.709586965	3.667723138	0.4687039	
19	Cluster0Agent1	3.024700492	1.943240603	4	1	0.838074017	1.943240603	0.2770767	
20	Cluster0Agent1	3.104022813	3.859606401	1	1	1.736702008	3.859606401	0.5595004	
21	Cluster0Agent1	2.533024998	2.127849495	1	2	0.938377998	2.127849495	0.37045746	
22	Cluster0Agent1	1.212010482	2.370484883	3	2	0.903616409	2.370484883	0.74555165	
23	Cluster0Agent1	0.528170402	2.948473644	3	4	0.528170402	1.319274353	1	
0	Cluster0Agent2	0.422375141	0	0	4	0.422375141	0	1	0.714611
1	Cluster0Agent2	1.163226822	0	2	2	0.60758766	0	0.52232948	
2	Cluster0Agent2	1.007238529	0	2	4	0.438278741	0	0.43512905	
3	Cluster0Agent2	0.581081235	0	2	4	0.385260806	0	0.66300679	
4	Cluster0Agent2	0.42210226	0	1	3	0.42210226	0	1	
5	Cluster0Agent2	0.371292098	0.020797033	3	0	0.371292098	0.018449608	1	
6	Cluster0Agent2	1.109505942	0.06906165	2	4	0.524663035	0.06906165	0.47287988	
7	Cluster0Agent2	1.689450914	0.294538054	4	4	0.99620271	0.294538054	0.58966064	
8	Cluster0Agent2	1.207264568	0.559823727	3	2	1.207264568	0.487889152	1	
9	Cluster0Agent2	0.898180007	0.792192345	2	0	0.898180007	0.562838165	1	
10	Cluster0Agent2	0.953949986	0.964842091	2	3	0.953949986	0.557442584	1	
11	Cluster0Agent2	1.057962561	1.027471976	2	1	1.057962561	0.597156599	1	
12	Cluster0Agent2	1.160146665	1.058813724	2	1	1.160146665	0.774049342	1	
13	Cluster0Agent2	1.576781562	1.047346036	1	1	1.576781562	0.988807878	1	
14	Cluster0Agent2	1.469635132	1.015965147	3	4	0.927566966	1.015965147	0.63115459	
15	Cluster0Agent2	1.725138383	0.818292613	1	0	1.392085823	0.818292613	0.80694154	
16	Cluster0Agent2	3.335705769	0.722839259	2	3	1.530622995	0.722839259	0.45886031	
17	Cluster0Agent2	3.634527608	0.445031932	3	1	1.272625494	0.445031932	0.3501488	
18	Cluster0Agent2	3.642063739	0.205050665	2	3	1.34998419	0.205050665	0.37066462	
19	Cluster0Agent2	3.008774428	0.044488815	0	2	0.69788709	0.044488815	0.23195062	
20	Cluster0Agent2	3.099815148	0.00459303	3	4	1.543735118	0.00459303	0.49800877	
21	Cluster0Agent2	2.52442955	0	0	4	0.834113776	0	0.33041674	
22	Cluster0Agent2	1.216060984	0	3	1	0.960092435	0	0.7895101	
23	Cluster0Agent2	0.528981708	0	3	2	0.528981708	0	1	

0	Cluster0Agent3	0.42307295	0	2	3	0.42307295	0	1	0.676349
1	Cluster0Agent3	1.16789018	0	0	1	0.394931979	0	0.33815849	
2	Cluster0Agent3	1.005195014	0	1	0	0.386905409	0	0.38490582	
3	Cluster0Agent3	0.581284174	0	2	0	0.326797104	0	0.56219852	
4	Cluster0Agent3	0.423991634	0	0	0	0.423991634	0	1	
5	Cluster0Agent3	0.370486367	0.020839044	3	3	0.370486367	0.018486878	1	
6	Cluster0Agent3	1.108560175	0.069092878	4	2	0.620056314	0.069092878	0.55933483	
7	Cluster0Agent3	1.685640731	0.294799485	4	2	0.91815918	0.294799485	0.54469447	
8	Cluster0Agent3	1.210898228	0.559808072	0	0	1.210898228	0.487875509	1	
9	Cluster0Agent3	0.897340409	0.791066948	0	2	0.897340409	0.562038591	1	
10	Cluster0Agent3	0.951944565	0.964951454	4	3	0.951944565	0.557505768	1	
11	Cluster0Agent3	1.059214185	1.029781715	4	2	1.059214185	0.598498997	1	
12	Cluster0Agent3	1.161232524	1.065075166	4	2	1.161232524	0.77862679	1	
13	Cluster0Agent3	1.581329257	1.046855198	1	0	1.581329257	0.988344473	1	
14	Cluster0Agent3	1.475835048	1.016068313	0	1	0.927566966	1.016068313	0.62850314	
15	Cluster0Agent3	1.717967679	0.814095989	1	2	1.118232874	0.814095989	0.65090449	
16	Cluster0Agent3	3.336751612	0.723267404	0	1	1.243631184	0.723267404	0.37270715	
17	Cluster0Agent3	3.640861711	0.445175684	2	1	1.102942095	0.445175684	0.30293436	
18	Cluster0Agent3	3.642373313	0.204862258	2	4	1.167235238	0.204862258	0.32046008	
19	Cluster0Agent3	3.012534441	0.04457576	4	4	0.603413292	0.04457576	0.20030088	
20	Cluster0Agent3	3.104204213	0.004584857	2	2	1.254284783	0.004584857	0.40406001	
21	Cluster0Agent3	2.5318998	0	3	4	0.677717443	0	0.26767151	
22	Cluster0Agent3	1.21796378	0	1	2	0.847140384	0	0.69553824	
23	Cluster0Agent3	0.528675203	0	3	4	0.528675203	0	1	
0	Cluster0Agent4	0.42266157	0	1	2	0.42266157	0	1	0.704103
1	Cluster0Agent4	1.159284888	0	1	4	0.455690745	0	0.39307917	
2	Cluster0Agent4	1.00308939	0	0	2	0.48644124	0	0.48494306	
3	Cluster0Agent4	0.58239387	0	1	3	0.365772905	0	0.62805075	
4	Cluster0Agent4	0.421621963	0	4	4	0.421621963	0	1	
5	Cluster0Agent4	0.368956516	0.0208299	1	3	0.368956516	0.018478766	1	
6	Cluster0Agent4	1.114832291	0.069288664	2	4	0.858539511	0.069288664	0.77010643	
7	Cluster0Agent4	1.685839393	0.295392911	0	2	1.060473853	0.295392911	0.62904797	
8	Cluster0Agent4	1.208517443	0.560277396	1	1	1.208517443	0.488284527	1	
9	Cluster0Agent4	0.898406419	0.792940558	4	3	0.898406419	0.563369757	1	
10	Cluster0Agent4	0.948625094	0.960467818	0	2	0.948625094	0.554915324	1	
11	Cluster0Agent4	1.054553274	1.027547039	2	4	1.054553274	0.597200225	1	
12	Cluster0Agent4	1.161780636	1.063643232	0	4	1.161780636	0.77757997	1	
13	Cluster0Agent4	1.579748006	1.048329575	1	1	1.579748006	0.989736446	1	
14	Cluster0Agent4	1.475052631	1.011642526	3	2	0.927566966	1.011642526	0.62883652	
15	Cluster0Agent4	1.723877277	0.815632921	4	4	1.141053953	0.815632921	0.66191136	
16	Cluster0Agent4	3.332203362	0.723565679	4	1	1.243631184	0.723565679	0.37321587	
17	Cluster0Agent4	3.644721656	0.445322867	1	3	1.272625494	0.445322867	0.34916946	
18	Cluster0Agent4	3.644300929	0.204823443	2	0	1.249767023	0.204823443	0.34293738	
19	Cluster0Agent4	3.018707418	0.044429497	4	3	0.646078878	0.044429497	0.21402501	
20	Cluster0Agent4	3.097489826	0.004583794	3	2	1.350768228	0.004583794	0.4360848	
21	Cluster0Agent4	2.524457476	0	3	0	0.729849554	0	0.28911145	
22	Cluster0Agent4	1.213764018	0	3	1	0.847140384	0	0.69794488	
23	Cluster0Agent4	0.527968808	0	1	0	0.527968808	0	1	
0	Cluster1Agent0	0.420253826	1.141437244	1	1	0.420253826	1.056930315	1	0.737088
1	Cluster1Agent0	1.159311216	1.518789742	1	4	0.514662489	1.518789742	0.44393816	
2	Cluster1Agent0	1.005669985	1.281638624	3	4	0.70066032	1.281638624	0.69670998	
3	Cluster1Agent0	0.580672474	1.00081041	0	3	0.458455109	1.00081041	0.78952444	
4	Cluster1Agent0	0.42240699	1.111292426	4	4	0.42240699	1.056480996	1	
5	Cluster1Agent0	0.369532375	1.010765724	1	2	0.369532375	0.896677514	1	
6	Cluster1Agent0	1.109296752	1.639757081	0	0	0.918160311	1.639757081	0.82769584	
7	Cluster1Agent0	1.683009245	2.075504507	4	2	0.940072434	2.075504507	0.55856641	
8	Cluster1Agent0	1.208320589	2.638317142	0	0	1.208320589	2.299306463	1	
9	Cluster1Agent0	0.895739176	1.972553531	3	1	0.895739176	1.401463188	1	
10	Cluster1Agent0	0.94888723	2.674000851	0	4	0.94888723	1.54491803	1	
11	Cluster1Agent0	1.056399353	3.005792279	4	4	1.056399353	1.746936887	1	
12	Cluster1Agent0	1.160612958	2.381714261	1	2	1.160612958	1.741160238	1	
13	Cluster1Agent0	1.580112851	2.607892494	2	3	1.580112851	2.462132432	1	
14	Cluster1Agent0	1.476791301	1.708250105	2	2	1.407914145	1.708250105	0.95336026	
15	Cluster1Agent0	1.720319318	2.255398147	0	1	1.428293519	2.255398147	0.83024907	
16	Cluster1Agent0	3.324662614	3.119805553	0	0	1.698452037	3.119805553	0.51086448	
17	Cluster1Agent0	3.650334773	3.101064681	2	3	1.248214976	3.101064681	0.34194534	
18	Cluster1Agent0	3.643601385	3.674134185	2	0	1.246135808	3.674134185	0.34200662	
19	Cluster1Agent0	3.021626283	1.940772183	1	2	0.679840202	1.940772183	0.22499149	
20	Cluster1Agent0	3.107228304	3.878612926	0	4	1.184775738	3.878612926	0.38129665	
21	Cluster1Agent0	2.53340648	2.124938306	1	4	0.56536467	2.124938306	0.22316382	
22	Cluster1Agent0	1.214220309	2.373813277	1	3	0.686998592	2.373813277	0.56579402	
23	Cluster1Agent0	0.527396818	2.946679188	1	0	0.527396818	1.318471436	1	
0	Cluster1Agent1	0.422915651	1.139478923	0	2	0.422915651	1.055116978	1	0.712319

1	Cluster1Agent1	1.159221642	1.526495345	1	4	0.364552596	1.526495345	0.3144805	
2	Cluster1Agent1	1.003601854	1.274481934	2	0	0.705812234	1.274481934	0.70327913	
3	Cluster1Agent1	0.581569161	0.998040006	3	2	0.336793883	0.998040006	0.57911235	
4	Cluster1Agent1	0.422941115	1.106899237	0	1	0.422941115	1.052304489	1	
5	Cluster1Agent1	0.36965208	1.011022285	2	2	0.36965208	0.896905116	1	
6	Cluster1Agent1	1.110774482	1.642771646	4	0	0.676539176	1.642771646	0.60906979	
7	Cluster1Agent1	1.683527768	2.075997235	3	2	0.947841627	2.075997235	0.5630092	
8	Cluster1Agent1	1.207206266	2.628270532	1	0	1.207206266	2.290550793	1	
9	Cluster1Agent1	0.899811938	1.960908743	1	0	0.899811938	1.393189779	1	
10	Cluster1Agent1	0.947326697	2.669054476	1	1	0.947326697	1.542060236	1	
11	Cluster1Agent1	1.055348033	2.99825459	0	0	1.055348033	1.742556056	1	
12	Cluster1Agent1	1.160365851	2.378442861	0	4	1.160365851	1.73876867	1	
13	Cluster1Agent1	1.58043393	2.609005474	4	4	1.58043393	2.463183206	1	
14	Cluster1Agent1	1.470902126	1.703689203	3	1	0.99382175	1.703689203	0.67565457	
15	Cluster1Agent1	1.720388966	2.250850319	0	2	1.16910512	2.250850319	0.6795586	
16	Cluster1Agent1	3.32368023	3.112062996	2	3	1.461113761	3.112062996	0.4396072	
17	Cluster1Agent1	3.636622512	3.106771895	3	4	1.329494091	3.106771895	0.36558485	
18	Cluster1Agent1	3.642253243	3.673200917	4	2	1.512750911	3.673200917	0.41533381	
19	Cluster1Agent1	3.023291751	1.935246842	2	3	0.882825665	1.935246842	0.29200809	
20	Cluster1Agent1	3.109883663	3.868254767	1	3	1.345863667	3.868254767	0.43276978	
21	Cluster1Agent1	2.528616169	2.128252132	2	2	0.719555035	2.128252132	0.28456475	
22	Cluster1Agent1	1.211363506	2.3672746	4	3	0.898382774	2.3672746	0.74162939	
23	Cluster1Agent1	0.527618327	2.941290633	0	3	0.527618327	1.316060365	1	
0	Cluster1Agent2	0.422389422	0	0	1	0.422389422	0	1	0.68744
1	Cluster1Agent2	1.158449553	0	3	2	0.364552596	0	0.31469009	
2	Cluster1Agent2	1.003983902	0	2	0	0.695508406	0	0.69274856	
3	Cluster1Agent2	0.580338785	0	2	1	0.400591843	0	0.69027239	
4	Cluster1Agent2	0.42351833	0	3	1	0.42351833	0	1	
5	Cluster1Agent2	0.368786092	0.020832571	0	4	0.368786092	0.018481136	1	
6	Cluster1Agent2	1.109478132	0.069115972	2	2	0.77318763	0.069115972	0.69689308	
7	Cluster1Agent2	1.685539845	0.29494799	2	0	0.947841627	0.29494799	0.56233712	
8	Cluster1Agent2	1.212556336	0.562639149	0	1	1.212556336	0.490342807	1	
9	Cluster1Agent2	0.895434532	0.79167016	1	3	0.895434532	0.562467162	1	
10	Cluster1Agent2	0.948799078	0.960147397	4	2	0.948799078	0.554730199	1	
11	Cluster1Agent2	1.056366536	1.028038233	3	0	1.056366536	0.597485702	1	
12	Cluster1Agent2	1.160574881	1.061614108	1	3	1.160574881	0.776096572	1	
13	Cluster1Agent2	1.579387814	1.044992231	3	3	1.579387814	0.986585631	1	
14	Cluster1Agent2	1.476439123	1.011568346	0	1	0.911003271	1.011568346	0.61702732	
15	Cluster1Agent2	1.726247573	0.820100894	2	4	0.931975308	0.820100894	0.53988508	
16	Cluster1Agent2	3.327325919	0.72181213	0	0	1.616867004	0.72181213	0.48593587	
17	Cluster1Agent2	3.648422966	0.444924425	2	0	1.166935862	0.444924425	0.31984665	
18	Cluster1Agent2	3.639758332	0.20483788	1	4	1.072256393	0.20483788	0.29459549	
19	Cluster1Agent2	3.011809727	0.044632527	0	3	0.541294568	0.044632527	0.17972403	
20	Cluster1Agent2	3.09605511	0.004571274	2	2	0.950938421	0.004571274	0.30714519	
21	Cluster1Agent2	2.523962071	0	0	4	0.582496933	0	0.23078672	
22	Cluster1Agent2	1.212334522	0	2	4	0.686998592	0	0.56667411	
23	Cluster1Agent2	0.526552013	0	3	3	0.526552013	0	1	
0	Cluster1Agent3	0.423145268	0	1	1	0.423145268	0	1	0.718433
1	Cluster1Agent3	1.162735985	0	4	1	0.343108326	0	0.29508704	
2	Cluster1Agent3	1.001229772	0	0	1	0.623381608	0	0.62261593	
3	Cluster1Agent3	0.580709734	0	2	1	0.378336741	0	0.65150749	
4	Cluster1Agent3	0.42264761	0	0	4	0.42264761	0	1	
5	Cluster1Agent3	0.369810061	0.020802578	2	0	0.369810061	0.018454528	1	
6	Cluster1Agent3	1.107638904	0.068913738	1	2	0.483242269	0.068913738	0.43628142	
7	Cluster1Agent3	1.685147355	0.293707446	1	0	0.648727671	0.293707446	0.38496792	
8	Cluster1Agent3	1.206582773	0.562555565	4	3	1.206582773	0.490269962	1	
9	Cluster1Agent3	0.893310822	0.791540978	3	4	0.893310822	0.562375381	1	
10	Cluster1Agent3	0.951816031	0.961195777	4	1	0.951816031	0.555335907	1	
11	Cluster1Agent3	1.055634575	1.02840874	1	4	1.055634575	0.597701037	1	
12	Cluster1Agent3	1.160128775	1.062559722	0	2	1.160128775	0.776787866	1	
13	Cluster1Agent3	1.580087288	1.048884907	1	0	1.580087288	0.990260739	1	
14	Cluster1Agent3	1.470503584	1.017297536	1	3	1.407914145	1.017297536	0.95743673	
15	Cluster1Agent3	1.717511216	0.819522177	3	1	1.417264226	0.819522177	0.82518484	
16	Cluster1Agent3	3.32568339	0.722593089	0	4	1.950623955	0.722593089	0.58653327	
17	Cluster1Agent3	3.635422198	0.445994652	1	2	1.521080576	0.445994652	0.41840548	
18	Cluster1Agent3	3.65095658	0.205436856	2	4	1.420015223	0.205436856	0.38894333	
19	Cluster1Agent3	3.010113868	0.044641627	0	2	0.734614057	0.044641627	0.24404859	
20	Cluster1Agent3	3.107314104	0.004574846	1	2	1.340667282	0.004574846	0.43145535	
21	Cluster1Agent3	2.531742998	0	1	0	0.659592116	0	0.26052886	
22	Cluster1Agent3	1.215040213	0	3	0	0.898382774	0	0.73938522	
23	Cluster1Agent3	0.529007581	0	3	1	0.529007581	0	1	
0	Cluster1Agent4	0.423801948	0	1	0	0.423801948	0	1	0.765595
1	Cluster1Agent4	1.160957873	0	4	3	0.235886974	0	0.20318306	

2	Cluster1Agent4	1.00350856	0	4	3	0.793394774	0	0.79062083	
3	Cluster1Agent4	0.579640071	0	2	4	0.405042864	0	0.69878341	
4	Cluster1Agent4	0.421753389	0	4	2	0.421753389	0	1	
5	Cluster1Agent4	0.368914154	0.020816955	0	2	0.368914154	0.018467282	1	
6	Cluster1Agent4	1.108188014	0.069186587	4	4	0.821511857	0.069186587	0.7413109	
7	Cluster1Agent4	1.681462514	0.294510926	1	3	1.184802034	0.294510926	0.7046259	
8	Cluster1Agent4	1.210978271	0.559631604	0	2	1.210978271	0.487721716	1	
9	Cluster1Agent4	0.894780546	0.792160432	1	1	0.894780546	0.562815492	1	
10	Cluster1Agent4	0.948618999	0.96624648	4	4	0.948618999	0.558253977	1	
11	Cluster1Agent4	1.056919527	1.026353378	2	1	1.056919527	0.596506481	1	
12	Cluster1Agent4	1.158566573	1.060824403	0	4	1.158566573	0.775519255	1	
13	Cluster1Agent4	1.576190145	1.046862227	0	2	1.576190145	0.98835111	1	
14	Cluster1Agent4	1.471650909	1.013660016	3	3	1.573551104	1.013660016	1.0692421	
15	Cluster1Agent4	1.712934223	0.818258296	2	3	1.516527868	0.818258296	0.88533923	
16	Cluster1Agent4	3.331069102	0.72333414	1	2	2.076709914	0.72333414	0.62343646	
17	Cluster1Agent4	3.645350226	0.443254507	0	4	1.625582295	0.443254507	0.44593309	
18	Cluster1Agent4	3.637420571	0.204245266	4	1	1.535934833	0.204245266	0.42225935	
19	Cluster1Agent4	3.018922768	0.044646055	3	2	0.886047656	0.044646055	0.29349795	
20	Cluster1Agent4	3.103149047	0.00459078	3	0	1.351060052	0.00459078	0.43538355	
21	Cluster1Agent4	2.539329971	0	2	4	0.81378248	0	0.32047134	
22	Cluster1Agent4	1.213716138	0	0	3	0.898382774	0	0.74019183	
23	Cluster1Agent4	0.528079251	0	0	2	0.528079251	0	1	
0	Cluster2Agent0	0.42204437	1.13955798	0	3	0.42204437	1.055190183	1	0.813578
1	Cluster2Agent0	1.163780777	1.52204724	2	2	0.692188474	1.52204724	0.59477566	
2	Cluster2Agent0	1.006122731	1.282657296	2	2	0.514509133	1.282657296	0.5113781	
3	Cluster2Agent0	0.583078592	1.004025763	4	4	0.437425359	1.004025763	0.75019966	
4	Cluster2Agent0	0.422913635	1.10931602	3	1	0.422913635	1.05460207	1	
5	Cluster2Agent0	0.369619004	1.012629922	3	0	0.369619004	0.898331295	1	
6	Cluster2Agent0	1.108227822	1.641216697	3	3	0.869836084	1.641216697	0.78488923	
7	Cluster2Agent0	1.678319888	2.085872535	1	3	1.291101561	2.085872535	0.76928217	
8	Cluster2Agent0	1.209779704	2.623850474	3	4	1.209779704	2.28669869	1	
9	Cluster2Agent0	0.90010193	1.967092629	4	2	0.90010193	1.397583317	1	
10	Cluster2Agent0	0.948868494	2.664814847	1	2	0.948868494	1.539610768	1	
11	Cluster2Agent0	1.057811243	3.004059386	0	3	1.057811243	1.745929746	1	
12	Cluster2Agent0	1.163208078	2.384589201	0	3	1.163208078	1.743261972	1	
13	Cluster2Agent0	1.573106442	2.604615547	4	4	1.573106442	2.45903864	1	
14	Cluster2Agent0	1.475211111	1.715973225	4	4	1.36374429	1.715973225	0.92444009	
15	Cluster2Agent0	1.717234547	2.24689387	2	4	1.620896697	2.24689387	0.94389942	
16	Cluster2Agent0	3.325643341	3.125850843	3	4	1.83942478	3.125850843	0.55310344	
17	Cluster2Agent0	3.633864746	3.097067073	4	1	1.880883576	3.097067073	0.51759867	
18	Cluster2Agent0	3.63608287	3.682328838	1	1	2.094589661	3.682328838	0.57605663	
19	Cluster2Agent0	3.015568794	1.935650225	0	3	1.139019551	1.935650225	0.377713	
20	Cluster2Agent0	3.096179097	3.871328714	4	3	2.192340988	3.871328714	0.70807951	
21	Cluster2Agent0	2.528575592	2.130668016	2	0	1.157019973	2.130668016	0.45757777	
22	Cluster2Agent0	1.213463234	2.367141617	2	4	1.282484391	2.367141617	1.05687948	
23	Cluster2Agent0	0.527560445	2.956368581	0	3	0.527560445	1.322806889	1	
0	Cluster2Agent1	0.421981998	1.140222583	3	4	0.421981998	1.055805581	1	0.831867
1	Cluster2Agent1	1.162974979	1.522242083	2	4	1.015209762	1.522242083	0.87294205	
2	Cluster2Agent1	1.002357607	1.285243404	3	0	0.627058006	1.285243404	0.62558313	
3	Cluster2Agent1	0.580455575	0.998179458	2	4	0.564606301	0.998179458	0.97269511	
4	Cluster2Agent1	0.422437651	1.110170862	4	4	0.422437651	1.05541475	1	
5	Cluster2Agent1	0.369719269	1.008325875	3	3	0.369719269	0.894513058	1	
6	Cluster2Agent1	1.10850361	1.642528232	3	4	0.869836084	1.642528232	0.78469396	
7	Cluster2Agent1	1.686040517	2.076584345	4	1	1.370860731	2.076584345	0.81306512	
8	Cluster2Agent1	1.207571448	2.627313288	3	2	1.207571448	2.28971655	1	
9	Cluster2Agent1	0.899654557	1.970032277	0	1	0.899654557	1.399671883	1	
10	Cluster2Agent1	0.95023801	2.667777961	3	0	0.95023801	1.541322723	1	
11	Cluster2Agent1	1.05585811	3.001101214	4	1	1.05585811	1.744210486	1	
12	Cluster2Agent1	1.161844201	2.383429069	0	1	1.161844201	1.742413854	1	
13	Cluster2Agent1	1.581136381	2.611312847	3	1	1.581136381	2.465361615	1	
14	Cluster2Agent1	1.472802475	1.711945718	3	3	1.148416244	1.711945718	0.77974899	
15	Cluster2Agent1	1.7096728	2.245059189	1	2	1.28998699	2.245059189	0.75452273	
16	Cluster2Agent1	3.310867335	3.124617372	0	4	1.846138009	3.124617372	0.55759951	
17	Cluster2Agent1	3.642418843	3.095676691	4	2	1.887391824	3.095676691	0.5181699	
18	Cluster2Agent1	3.656184846	3.698216843	3	4	2.094589661	3.698216843	0.57288943	
19	Cluster2Agent1	3.003886442	1.933172047	2	2	1.086564703	1.933172047	0.36171963	
20	Cluster2Agent1	3.090800293	3.875107511	3	3	2.307727356	3.875107511	0.74664396	
21	Cluster2Agent1	2.526308561	2.130770739	2	2	1.221076442	2.130770739	0.48334414	
22	Cluster2Agent1	1.213192607	2.378475018	4	0	1.360210718	2.378475018	1.12118283	
23	Cluster2Agent1	0.526902194	2.947860502	1	4	0.526902194	1.319000007	1	
0	Cluster2Agent2	0.420779726	0	1	0	0.420779726	0	1	0.703429
1	Cluster2Agent2	1.165038033	0	1	4	0.738334372	0	0.63374272	
2	Cluster2Agent2	1.00383661	0	4	3	0.438710913	0	0.43703418	

3	Cluster2Agent2	0.579946088	0	2	4	0.49138091	0	0.84728722	
4	Cluster2Agent2	0.422121946	0	0	0	0.422121946	0	1	
5	Cluster2Agent2	0.369780803	0.02088635	2	0	0.369780803	0.018528844	1	
6	Cluster2Agent2	1.1120799	0.069265268	4	2	0.579890723	0.069265268	0.521447	
7	Cluster2Agent2	1.685035604	0.295019836	0	3	1.061793948	0.295019836	0.63013146	
8	Cluster2Agent2	1.210823259	0.560352566	0	0	1.210823259	0.488350038	1	
9	Cluster2Agent2	0.898885986	0.790443211	1	0	0.898885986	0.561595437	1	
10	Cluster2Agent2	0.950558401	0.958364837	3	0	0.950558401	0.553700316	1	
11	Cluster2Agent2	1.054298712	1.027844208	3	4	1.054298712	0.597372936	1	
12	Cluster2Agent2	1.159738866	1.05977048	3	4	1.159738866	0.774748781	1	
13	Cluster2Agent2	1.576169508	1.047401389	4	0	1.576169508	0.988860137	1	
14	Cluster2Agent2	1.469948957	1.013249677	2	0	0.861312183	1.013249677	0.585947	
15	Cluster2Agent2	1.716359992	0.818868715	0	3	1.189031487	0.818868715	0.69276346	
16	Cluster2Agent2	3.327917094	0.720130072	0	2	1.322506137	0.720130072	0.39739756	
17	Cluster2Agent2	3.631643429	0.446885194	1	2	1.275616543	0.446885194	0.35125049	
18	Cluster2Agent2	3.653221174	0.2048886	2	1	1.433140294	0.2048886	0.39229497	
19	Cluster2Agent2	3.026951554	0.044521417	1	4	0.565763	0.044521417	0.18690851	
20	Cluster2Agent2	3.11355626	0.004587628	2	3	1.153863678	0.004587628	0.37059349	
21	Cluster2Agent2	2.530103351	0	1	0	0.732645865	0	0.28957152	
22	Cluster2Agent2	1.210179868	0	4	2	0.660673777	0	0.54593023	
23	Cluster2Agent2	0.526416086	0	4	4	0.526416086	0	1	
0	Cluster2Agent3	0.423029727	0	4	1	0.423029727	0	1	0.725612
1	Cluster2Agent3	1.161670546	0	2	3	0.646042576	0	0.55613235	
2	Cluster2Agent3	1.002192615	0	4	2	0.650027164	0	0.64860502	
3	Cluster2Agent3	0.581145032	0	1	2	0.522212654	0	0.89859265	
4	Cluster2Agent3	0.422631661	0	2	0	0.422631661	0	1	
5	Cluster2Agent3	0.368898028	0.020904476	3	0	0.368898028	0.018544924	1	
6	Cluster2Agent3	1.106068498	0.069133134	1	0	0.724863403	0.069133134	0.65535128	
7	Cluster2Agent3	1.678218538	0.293856891	4	1	1.051824052	0.293856891	0.62675035	
8	Cluster2Agent3	1.210478424	0.561459484	1	1	1.210478424	0.489314723	1	
9	Cluster2Agent3	0.898647802	0.793294378	3	2	0.898647802	0.563621139	1	
10	Cluster2Agent3	0.94807412	0.963134801	3	3	0.94807412	0.556456188	1	
11	Cluster2Agent3	1.054233119	1.024040384	2	0	1.054233119	0.595162192	1	
12	Cluster2Agent3	1.165001709	1.059282688	0	1	1.165001709	0.774392179	1	
13	Cluster2Agent3	1.578126627	1.046024719	1	0	1.578126627	0.987560412	1	
14	Cluster2Agent3	1.474781585	1.014178938	0	2	1.076640229	1.014178938	0.73003368	
15	Cluster2Agent3	1.718434491	0.81998114	0	1	1.194640126	0.81998114	0.69519096	
16	Cluster2Agent3	3.335669742	0.725623459	1	3	1.429917803	0.725623459	0.42867487	
17	Cluster2Agent3	3.64081759	0.445289908	1	3	0.989253646	0.445289908	0.27171195	
18	Cluster2Agent3	3.639860028	0.205419982	3	2	1.212657172	0.205419982	0.33316039	
19	Cluster2Agent3	3.021232822	0.04457492	0	4	0.685659795	0.04457492	0.22694702	
20	Cluster2Agent3	3.096588824	0.004580558	0	0	1.384636413	0.004580558	0.44714894	
21	Cluster2Agent3	2.536371748	0	2	3	0.732645865	0	0.28885587	
22	Cluster2Agent3	1.215419808	0	4	0	0.738400104	0	0.6075268	
23	Cluster2Agent3	0.528063494	0	2	0	0.528063494	0	1	
0	Cluster2Agent4	0.423359207	0	3	2	0.423359207	0	1	0.73744
1	Cluster2Agent4	1.158518626	0	4	0	0.553750779	0	0.47798177	
2	Cluster2Agent4	1.005791876	0	3	2	0.686777817	0	0.68282299	
3	Cluster2Agent4	0.579235228	0	0	0	0.462476151	0	0.79842546	
4	Cluster2Agent4	0.42184204	0	0	3	0.42184204	0	1	
5	Cluster2Agent4	0.369683526	0.020859998	2	1	0.369683526	0.018505467	1	
6	Cluster2Agent4	1.10942725	0.069152581	3	2	0.62821495	0.069152581	0.56625159	
7	Cluster2Agent4	1.689870071	0.293880348	0	4	0.982034778	0.293880348	0.58113035	
8	Cluster2Agent4	1.211172331	0.560850694	4	1	1.211172331	0.488784159	1	
9	Cluster2Agent4	0.898351158	0.791602082	1	3	0.898351158	0.562418794	1	
10	Cluster2Agent4	0.948127317	0.96519584	3	2	0.948127317	0.557646964	1	
11	Cluster2Agent4	1.05498759	1.0250695	3	0	1.05498759	0.595760304	1	
12	Cluster2Agent4	1.161404993	1.063325123	1	4	1.161404993	0.777347415	1	
13	Cluster2Agent4	1.580878791	1.047967386	3	3	1.580878791	0.9893945	1	
14	Cluster2Agent4	1.469966608	1.013193674	0	0	1.004864214	1.013193674	0.68359663	
15	Cluster2Agent4	1.717613284	0.817651416	0	1	1.189031487	0.817651416	0.69225797	
16	Cluster2Agent4	3.33323168	0.723849338	3	2	1.322506137	0.723849338	0.39676394	
17	Cluster2Agent4	3.637293309	0.445049693	0	4	1.386256754	0.445049693	0.38112317	
18	Cluster2Agent4	3.636778244	0.205286117	1	3	1.543381855	0.205286117	0.42438162	
19	Cluster2Agent4	3.010880216	0.044522994	2	4	0.794316266	0.044522994	0.2638153	
20	Cluster2Agent4	3.103683333	0.004581755	0	3	1.730795517	0.004581755	0.55765854	
21	Cluster2Agent4	2.526560316	0	4	3	0.912804685	0	0.36128355	
22	Cluster2Agent4	1.215826723	0	4	3	1.010442247	0	0.83107422	
23	Cluster2Agent4	0.526502001	0	2	0	0.526502001	0	1	
0	Cluster3Agent0	0.422171446	1.140093776	2	0	0.422171446	1.055686311	1	0.794427
1	Cluster3Agent0	1.15937392	1.517647264	4	0	0.484094234	1.517647264	0.41754798	
2	Cluster3Agent0	0.999677549	1.280200841	0	3	0.609382456	1.280200841	0.60957902	
3	Cluster3Agent0	0.579955227	0.997955812	0	1	0.4266731	0.997955812	0.73570007	

4	Cluster3Agent0	0.420515347	1.11019324	3	1	0.420515347	1.055436024	1	
5	Cluster3Agent0	0.369706422	1.011924556	1	1	0.369706422	0.897705545	1	
6	Cluster3Agent0	1.10347542	1.642745784	1	0	0.697563353	1.642745784	0.63215124	
7	Cluster3Agent0	1.688490995	2.07979879	4	4	0.992722781	2.07979879	0.5879349	
8	Cluster3Agent0	1.210486506	2.62957472	2	2	1.210486506	2.291687399	1	
9	Cluster3Agent0	0.900071906	1.967639849	3	2	0.900071906	1.397972107	1	
10	Cluster3Agent0	0.950968554	2.673293228	4	0	0.950968554	1.544509197	1	
11	Cluster3Agent0	1.055907499	3.007003155	3	0	1.055907499	1.747640636	1	
12	Cluster3Agent0	1.162777596	2.381519448	2	3	1.162777596	1.741017819	1	
13	Cluster3Agent0	1.581711	2.61072295	3	0	1.581711	2.464804688	1	
14	Cluster3Agent0	1.471890375	1.714509685	1	4	1.471890375	1.714509685	1	
15	Cluster3Agent0	1.719063098	2.248951155	3	4	1.719063098	2.248951155	1	
16	Cluster3Agent0	3.323134058	3.114470309	2	2	2.100034067	3.114470309	0.63194383	
17	Cluster3Agent0	3.630603132	3.099097142	4	0	1.89900144	3.099097142	0.52305399	
18	Cluster3Agent0	3.647118133	3.681659563	1	3	2.103237842	3.681659563	0.57668487	
19	Cluster3Agent0	3.021627764	1.939075925	3	3	1.044389327	1.939075925	0.34563798	
20	Cluster3Agent0	3.104232128	3.872487456	0	1	2.196935143	3.872487456	0.70772257	
21	Cluster3Agent0	2.527796772	2.132373082	0	0	1.008226082	2.132373082	0.39885567	
22	Cluster3Agent0	1.214982307	2.373099138	4	0	1.092798042	2.373099138	0.89943535	
23	Cluster3Agent0	0.527207713	2.953548467	2	3	0.527207713	1.321545049	1	
0	Cluster3Agent1	0.423680762	1.137122591	3	1	0.423680762	1.052935099	1	0.80298
1	Cluster3Agent1	1.163539171	1.518040441	3	1	0.414937914	1.518040441	0.35661706	
2	Cluster3Agent1	1.002259729	1.279081234	1	2	0.551963441	1.279081234	0.55071897	
3	Cluster3Agent1	0.582074022	1.004984723	1	4	0.428277134	1.004984723	0.73577778	
4	Cluster3Agent1	0.42151265	1.115144825	1	4	0.42151265	1.060143386	1	
5	Cluster3Agent1	0.369032917	1.010674849	4	3	0.369032917	0.896596896	1	
6	Cluster3Agent1	1.107773945	1.6370938	4	2	0.912198231	1.6370938	0.8234516	
7	Cluster3Agent1	1.68669622	2.076039731	2	0	1.075814115	2.076039731	0.63782328	
8	Cluster3Agent1	1.210396249	2.623371445	4	1	1.210396249	2.286281214	1	
9	Cluster3Agent1	0.897744502	1.971004249	4	2	0.897744502	1.400362452	1	
10	Cluster3Agent1	0.949395496	2.675508304	1	4	0.949395496	1.545788969	1	
11	Cluster3Agent1	1.053113897	3.00494777	4	1	1.053113897	1.746446066	1	
12	Cluster3Agent1	1.159793121	2.383579857	1	4	1.159793121	1.742524088	1	
13	Cluster3Agent1	1.581664172	2.616642462	0	1	1.581664172	2.470393348	1	
14	Cluster3Agent1	1.473936664	1.708521853	3	3	1.473936664	1.708521853	1	
15	Cluster3Agent1	1.720794104	2.247249307	0	3	1.720794104	2.247249307	1	
16	Cluster3Agent1	3.331385936	3.130637646	1	1	1.968781938	3.130637646	0.59097984	
17	Cluster3Agent1	3.649707076	3.100191366	4	1	1.766320117	3.100191366	0.48396216	
18	Cluster3Agent1	3.632951395	3.676239422	0	4	2.383669555	3.676239422	0.65612481	
19	Cluster3Agent1	3.01908605	1.939106439	2	4	0.974763372	1.939106439	0.32286704	
20	Cluster3Agent1	3.099860843	3.8593057	3	0	2.071395992	3.8593057	0.66822225	
21	Cluster3Agent1	2.52230488	2.134010267	1	3	1.152258379	2.134010267	0.45682756	
22	Cluster3Agent1	1.21341759	2.371785373	3	3	1.199042296	2.371785373	0.98815305	
23	Cluster3Agent1	0.528630144	2.946391924	3	4	0.528630144	1.318342902	1	
0	Cluster3Agent2	0.421840016	0	4	3	0.421840016	0	1	0.808829
1	Cluster3Agent2	1.1594602	0	2	0	0.72614135	0	0.62627536	
2	Cluster3Agent2	0.999074817	0	2	4	0.550111214	0	0.55062064	
3	Cluster3Agent2	0.582187266	0	3	0	0.478002194	0	0.82104543	
4	Cluster3Agent2	0.422434297	0	1	2	0.422434297	0	1	
5	Cluster3Agent2	0.370822573	0.020907391	4	2	0.370822573	0.01854751	1	
6	Cluster3Agent2	1.107011214	0.069085143	4	4	0.858539511	0.069085143	0.77554726	
7	Cluster3Agent2	1.688179966	0.294723441	1	3	1.027708606	0.294723441	0.60876721	
8	Cluster3Agent2	1.209765672	0.560363911	0	4	1.209765672	0.488359926	1	
9	Cluster3Agent2	0.900637366	0.795769968	4	0	0.900637366	0.565380001	1	
10	Cluster3Agent2	0.950907514	0.959335771	2	4	0.950907514	0.554261278	1	
11	Cluster3Agent2	1.056309929	1.028717578	3	1	1.056309929	0.597880531	1	
12	Cluster3Agent2	1.16182037	1.056473584	0	4	1.16182037	0.772338574	1	
13	Cluster3Agent2	1.578531601	1.045412652	1	4	1.578531601	0.986982555	1	
14	Cluster3Agent2	1.479450414	1.013192101	4	1	1.479450414	1.013192101	1	
15	Cluster3Agent2	1.715100435	0.820508253	2	2	1.715100435	0.820508253	1	
16	Cluster3Agent2	3.326873245	0.719301824	3	1	1.70627768	0.719301824	0.51287727	
17	Cluster3Agent2	3.658546069	0.44496041	3	3	1.658516541	0.44496041	0.45332668	
18	Cluster3Agent2	3.636055064	0.205646592	1	0	2.383669555	0.205646592	0.65556476	
19	Cluster3Agent2	3.020231941	0.04446563	3	0	0.974763372	0.04446563	0.32274454	
20	Cluster3Agent2	3.100229745	0.004574098	1	2	1.954823923	0.004574098	0.63054163	
21	Cluster3Agent2	2.53067861	0	0	4	1.152258379	0	0.45531597	
22	Cluster3Agent2	1.215121569	0	4	3	1.214220047	0	0.99925808	
23	Cluster3Agent2	0.528544785	0	4	0	0.528544785	0	1	
0	Cluster3Agent3	0.421240744	0	3	0	0.421240744	0	1	0.800847
1	Cluster3Agent3	1.160669383	0	0	0	0.622406872	0	0.5362482	
2	Cluster3Agent3	1.003820366	0	3	1	0.333400736	0	0.33213187	
3	Cluster3Agent3	0.579575668	0	2	3	0.391384347	0	0.67529465	
4	Cluster3Agent3	0.421546123	0	4	0	0.421546123	0	1	

5	Cluster3Agent3	0.369436576	0.020813832	4	3	0.369436576	0.018464511	1	
6	Cluster3Agent3	1.111132924	0.068690856	2	3	0.858539511	0.068690856	0.77267039	
7	Cluster3Agent3	1.682855539	0.294719259	1	2	1.018962149	0.294719259	0.60549591	
8	Cluster3Agent3	1.208160784	0.559469585	4	4	1.208160784	0.487580516	1	
9	Cluster3Agent3	0.896844037	0.791784966	3	2	0.896844037	0.562548729	1	
10	Cluster3Agent3	0.951706056	0.958732479	4	3	0.951706056	0.553912723	1	
11	Cluster3Agent3	1.050916192	1.032423146	1	0	1.050916192	0.60003417	1	
12	Cluster3Agent3	1.16566477	1.062273116	1	0	1.16566477	0.776578342	1	
13	Cluster3Agent3	1.572294257	1.04459036	0	1	1.572294257	0.986206222	1	
14	Cluster3Agent3	1.474795195	1.012938128	2	4	1.474795195	1.012938128	1	
15	Cluster3Agent3	1.712043478	0.819951194	0	4	1.712043478	0.819951194	1	
16	Cluster3Agent3	3.330573049	0.721966742	1	3	2.100034067	0.721966742	0.63053235	
17	Cluster3Agent3	3.628518416	0.444911882	4	3	2.006805015	0.444911882	0.55306458	
18	Cluster3Agent3	3.625877024	0.204444142	4	1	1.963021986	0.204444142	0.54139232	
19	Cluster3Agent3	3.006006064	0.044305006	1	4	1.253267193	0.044305006	0.41692105	
20	Cluster3Agent3	3.096043243	0.004579325	2	1	1.927922676	0.004579325	0.62270535	
21	Cluster3Agent3	2.525971343	0	1	3	1.224274528	0	0.48467475	
22	Cluster3Agent3	1.215145239	0	4	3	1.274931049	0	1.04920055	
23	Cluster3Agent3	0.528757957	0	1	2	0.528757957	0	1	
0	Cluster3Agent4	0.422556995	0	4	3	0.422556995	0	1	0.794613
1	Cluster3Agent4	1.162147892	0	0	4	0.587828712	0	0.50581231	
2	Cluster3Agent4	1.003258116	0	0	4	0.459352125	0	0.45786036	
3	Cluster3Agent4	0.579340008	0	3	0	0.364115765	0	0.62850098	
4	Cluster3Agent4	0.423669171	0	3	4	0.423669171	0	1	
5	Cluster3Agent4	0.370390171	0.020828238	3	4	0.370390171	0.018477291	1	
6	Cluster3Agent4	1.110667628	0.069178317	2	0	0.804880792	0.069178317	0.72468196	
7	Cluster3Agent4	1.679521481	0.294299073	3	4	0.935870815	0.294299073	0.55722468	
8	Cluster3Agent4	1.206579678	0.55944603	1	4	1.206579678	0.487559988	1	
9	Cluster3Agent4	0.897422263	0.793902172	2	0	0.897422263	0.564052966	1	
10	Cluster3Agent4	0.950568897	0.961768704	1	3	0.950568897	0.555666918	1	
11	Cluster3Agent4	1.059012827	1.02593268	0	3	1.059012827	0.596261976	1	
12	Cluster3Agent4	1.16692934	1.061351874	3	4	1.16692934	0.775904864	1	
13	Cluster3Agent4	1.575432914	1.042262295	3	1	1.575432914	0.984008277	1	
14	Cluster3Agent4	1.472442044	1.015943581	1	3	1.472442044	1.015943581	1	
15	Cluster3Agent4	1.716265848	0.817801924	1	3	1.716265848	0.817801924	1	
16	Cluster3Agent4	3.318058978	0.721871675	3	4	2.100034067	0.721871675	0.63291041	
17	Cluster3Agent4	3.631084282	0.443663401	3	3	2.122901173	0.443663401	0.58464663	
18	Cluster3Agent4	3.638630097	0.20450583	0	4	1.682590274	0.20450583	0.46242411	
19	Cluster3Agent4	3.024211172	0.044503457	0	2	1.114015282	0.044503457	0.36836557	
20	Cluster3Agent4	3.105333567	0.004590646	3	1	2.205902225	0.004590646	0.71035919	
21	Cluster3Agent4	2.534459882	0	3	0	1.08024223	0	0.42622187	
22	Cluster3Agent4	1.215182285	0	2	1	1.229397797	0	1.01169826	
23	Cluster3Agent4	0.527695623	0	4	2	0.527695623	0	1	
0	Cluster4Agent0	0.420363767	1.139733155	0	2	0.420363767	1.055352388	1	0.706484
1	Cluster4Agent0	1.160229752	1.517436759	2	0	1.00251964	1.517436759	0.86406993	
2	Cluster4Agent0	1.002365218	1.275062871	1	0	0.347643141	1.275062871	0.34682283	
3	Cluster4Agent0	0.58039007	0.996964583	4	1	0.240585389	0.996964583	0.41452361	
4	Cluster4Agent0	0.423484397	1.111679208	1	3	0.423484397	1.056848701	1	
5	Cluster4Agent0	0.369320914	1.009110382	0	1	0.369320914	0.895209015	1	
6	Cluster4Agent0	1.108231702	1.640585921	1	4	0.447155996	1.640585921	0.40348602	
7	Cluster4Agent0	1.682336587	2.078200417	1	4	0.803629109	2.078200417	0.47768628	
8	Cluster4Agent0	1.210026178	2.624412574	0	0	1.210026178	2.287188563	1	
9	Cluster4Agent0	0.896085489	1.971017259	1	3	0.896085489	1.400371695	1	
10	Cluster4Agent0	0.950233028	2.672972691	0	4	0.950233028	1.544324005	1	
11	Cluster4Agent0	1.058688195	3.000544979	4	4	1.058688195	1.743887207	1	
12	Cluster4Agent0	1.162133419	2.377838971	1	3	1.162133419	1.738327194	1	
13	Cluster4Agent0	1.581330088	2.610951373	4	2	1.581330088	2.465020345	1	
14	Cluster4Agent0	1.475744461	1.706049004	3	0	1.407914145	1.706049004	0.95403654	
15	Cluster4Agent0	1.716595694	2.249984215	4	1	1.12217372	2.249984215	0.65372046	
16	Cluster4Agent0	3.328365787	3.117990304	2	2	1.421376269	3.117990304	0.4270493	
17	Cluster4Agent0	3.633134715	3.101179156	1	4	1.367813129	3.101179156	0.37648291	
18	Cluster4Agent0	3.642181912	3.673948243	4	2	1.63160269	3.673948243	0.44797397	
19	Cluster4Agent0	3.015121728	1.937992857	4	1	0.718705474	1.937992857	0.23836698	
20	Cluster4Agent0	3.114876902	3.859364241	4	0	1.403284715	3.859364241	0.45051049	
21	Cluster4Agent0	2.531624792	2.126129409	3	3	0.686389703	2.126129409	0.27112616	
22	Cluster4Agent0	1.219361635	2.371163039	2	0	0.76791777	2.371163039	0.62977032	
23	Cluster4Agent0	0.527688033	2.951577606	0	3	0.527688033	1.3206632	1	
0	Cluster4Agent1	0.420859191	1.141464946	4	1	0.420859191	1.056955965	1	0.781918
1	Cluster4Agent1	1.162146342	1.511226186	4	2	1.113910711	1.511226186	0.95849436	
2	Cluster4Agent1	1.005873573	1.282439311	3	3	0.369212241	1.282439311	0.36705631	
3	Cluster4Agent1	0.581556505	1.00152137	3	2	0.328381953	1.00152137	0.56466044	
4	Cluster4Agent1	0.420072584	1.108240774	2	1	0.420072584	1.053579858	1	
5	Cluster4Agent1	0.369700338	1.010694133	1	2	0.369700338	0.896614003	1	

6	Cluster4Agent1	1.107146861	1.64301691	3	4	0.566397594	1.64301691	0.51158307	
7	Cluster4Agent1	1.679770794	2.085265689	3	3	1.16960105	2.085265689	0.6962861	
8	Cluster4Agent1	1.211074041	2.632925277	1	2	1.211074041	2.294607426	1	
9	Cluster4Agent1	0.897910539	1.97325989	3	4	0.897910539	1.401965043	1	
10	Cluster4Agent1	0.949573958	2.66520475	4	2	0.949573958	1.539836036	1	
11	Cluster4Agent1	1.05703704	3.006582397	2	0	1.05703704	1.747396096	1	
12	Cluster4Agent1	1.159980546	2.37761312	1	2	1.159980546	1.738162085	1	
13	Cluster4Agent1	1.579217052	2.612007284	1	3	1.579217052	2.466017239	1	
14	Cluster4Agent1	1.472686284	1.712309592	4	2	1.87721886	1.712309592	1.27469026	
15	Cluster4Agent1	1.71222677	2.247038629	4	0	1.64053823	2.247038629	0.9581314	
16	Cluster4Agent1	3.338905756	3.113448268	2	2	2.077396086	3.113448268	0.62217871	
17	Cluster4Agent1	3.641798387	3.093821997	2	0	1.66460277	3.093821997	0.45708263	
18	Cluster4Agent1	3.639784668	3.671817994	4	1	1.657096482	3.671817994	0.45527322	
19	Cluster4Agent1	3.009358358	1.938719645	3	3	0.674134592	1.938719645	0.22401273	
20	Cluster4Agent1	3.101428864	3.855460373	4	4	1.397845627	3.855460373	0.4507102	
21	Cluster4Agent1	2.528484356	2.13092008	4	2	0.931528883	2.13092008	0.36841394	
22	Cluster4Agent1	1.21542901	2.36700395	3	4	1.042174116	2.36700395	0.85745371	
23	Cluster4Agent1	0.52840599	2.954514674	1	4	0.52840599	1.321977371	1	
0	Cluster4Agent2	0.422320873	0	2	4	0.422320873	0	1	0.715603
1	Cluster4Agent2	1.161945491	0	0	0	1.00251964	0	0.86279404	
2	Cluster4Agent2	1.003187497	0	4	3	0.286742153	0	0.28583107	
3	Cluster4Agent2	0.578894273	0	0	2	0.319260232	0	0.55150007	
4	Cluster4Agent2	0.423876066	0	0	3	0.423876066	0	1	
5	Cluster4Agent2	0.36998479	0.020737718	2	2	0.36998479	0.018396989	1	
6	Cluster4Agent2	1.108693999	0.069301589	3	0	0.387535196	0.069301589	0.34954207	
7	Cluster4Agent2	1.684114942	0.294243107	2	2	0.871541428	0.294243107	0.51750709	
8	Cluster4Agent2	1.206305316	0.55956988	2	2	1.206305316	0.487667924	1	
9	Cluster4Agent2	0.897451695	0.796133625	4	2	0.897451695	0.565638372	1	
10	Cluster4Agent2	0.951592979	0.961650462	2	1	0.951592979	0.555598603	1	
11	Cluster4Agent2	1.059762688	1.028349292	4	2	1.059762688	0.597666487	1	
12	Cluster4Agent2	1.162658631	1.060191347	0	4	1.162658631	0.775056458	1	
13	Cluster4Agent2	1.577496428	1.04624127	3	3	1.577496428	0.987764859	1	
14	Cluster4Agent2	1.473564984	1.013905692	3	4	1.126331316	1.013905692	0.76435809	
15	Cluster4Agent2	1.71903756	0.81954262	4	2	1.304455745	0.81954262	0.75882911	
16	Cluster4Agent2	3.32212501	0.722260817	1	4	1.749386177	0.722260817	0.5265865	
17	Cluster4Agent2	3.647827771	0.443290934	0	2	1.677506667	0.443290934	0.45986455	
18	Cluster4Agent2	3.644885599	0.204817278	2	3	1.548747866	0.204817278	0.42490987	
19	Cluster4Agent2	3.013670848	0.04467757	1	4	0.64070643	0.04467757	0.2126	
20	Cluster4Agent2	3.105800809	0.004603052	4	1	1.25099025	0.004603052	0.40279153	
21	Cluster4Agent2	2.526485907	0	2	0	0.735417539	0	0.29108318	
22	Cluster4Agent2	1.216899802	0	0	3	0.932471578	0	0.76626816	
23	Cluster4Agent2	0.52776549	0	4	4	0.52776549	0	1	
0	Cluster4Agent3	0.42308603	0	1	3	0.42308603	0	1	0.668949
1	Cluster4Agent3	1.160760103	0	3	0	0.612650891	0	0.52780147	
2	Cluster4Agent3	1.003002957	0	1	2	0.383168717	0	0.38202152	
3	Cluster4Agent3	0.583894889	0	3	1	0.312418941	0	0.53506024	
4	Cluster4Agent3	0.421474537	0	0	2	0.421474537	0	1	
5	Cluster4Agent3	0.369665383	0.020788511	3	0	0.369665383	0.018442048	1	
6	Cluster4Agent3	1.109420593	0.069085307	2	0	0.417345596	0.069085307	0.37618339	
7	Cluster4Agent3	1.680144039	0.294981783	2	2	0.694214817	0.294981783	0.41318768	
8	Cluster4Agent3	1.210047775	0.559143661	3	2	1.210047775	0.487296471	1	
9	Cluster4Agent3	0.897740542	0.79324732	1	1	0.897740542	0.563587705	1	
10	Cluster4Agent3	0.952321864	0.960628593	1	0	0.952321864	0.555008213	1	
11	Cluster4Agent3	1.056547746	1.029362631	1	1	1.056547746	0.598255429	1	
12	Cluster4Agent3	1.161625302	1.05942403	2	0	1.161625302	0.774495508	1	
13	Cluster4Agent3	1.575396593	1.044317411	0	3	1.575396593	0.985948529	1	
14	Cluster4Agent3	1.476358879	1.014284333	2	1	1.314053202	1.014284333	0.89006353	
15	Cluster4Agent3	1.711931104	0.816790794	1	2	1.116477406	0.816790794	0.65217426	
16	Cluster4Agent3	3.324273181	0.720509747	3	1	1.421376269	0.720509747	0.42757505	
17	Cluster4Agent3	3.657627627	0.444571321	2	2	1.36136118	0.444571321	0.37219786	
18	Cluster4Agent3	3.628088765	0.204299344	0	1	1.064365817	0.204299344	0.29336819	
19	Cluster4Agent3	3.018157455	0.04452451	1	3	0.504208104	0.04452451	0.16705825	
20	Cluster4Agent3	3.105754372	0.004585214	2	2	0.902888615	0.004585214	0.29071475	
21	Cluster4Agent3	2.535440816	0	0	4	0.588334032	0	0.23204408	
22	Cluster4Agent3	1.218108447	0	4	3	0.603363962	0	0.49532861	
23	Cluster4Agent3	0.529619082	0	0	1	0.529619082	0	1	
0	Cluster4Agent4	0.422811772	0	2	1	0.422811772	0	1	0.70234
1	Cluster4Agent4	1.164736476	0	2	1	0.724041962	0	0.62163586	
2	Cluster4Agent4	1.004357023	0	4	2	0.285473382	0	0.28423496	
3	Cluster4Agent4	0.582550144	0	4	3	0.263389691	0	0.45213222	
4	Cluster4Agent4	0.422037665	0	4	1	0.422037665	0	1	
5	Cluster4Agent4	0.369623566	0.020755446	0	3	0.369623566	0.018412715	1	
6	Cluster4Agent4	1.109574487	0.069220478	2	2	0.476966395	0.069220478	0.42986424	

7	Cluster4Agent4	1.684936929	0.29453611	2	3	0.818720735	0.29453611	0.48590586	
8	Cluster4Agent4	1.210835734	0.560909207	2	4	1.210835734	0.488835153	1	
9	Cluster4Agent4	0.897892035	0.792834728	1	3	0.897892035	0.563294566	1	
10	Cluster4Agent4	0.948504554	0.960913729	1	2	0.948504554	0.555172952	1	
11	Cluster4Agent4	1.054902089	1.02792791	1	4	1.054902089	0.597421584	1	
12	Cluster4Agent4	1.161690055	1.058253072	3	4	1.161690055	0.773639475	1	
13	Cluster4Agent4	1.57797059	1.043160989	1	0	1.57797059	0.984856741	1	
14	Cluster4Agent4	1.471950815	1.012885711	4	3	1.407914145	1.012885711	0.95649537	
15	Cluster4Agent4	1.718299957	0.818391787	4	0	1.310152058	0.818391787	0.76246994	
16	Cluster4Agent4	3.335138608	0.723407703	4	3	1.640049541	0.723407703	0.49174854	
17	Cluster4Agent4	3.649531543	0.446407153	2	3	1.380717026	0.446407153	0.37832719	
18	Cluster4Agent4	3.647367497	0.204632028	1	3	1.459519594	0.204632028	0.40015699	
19	Cluster4Agent4	3.013699841	0.044614449	2	1	0.682491632	0.044614449	0.22646304	
20	Cluster4Agent4	3.11112776	0.00457605	2	0	1.245551162	0.00457605	0.40035359	
21	Cluster4Agent4	2.528339134	0	2	2	0.735417539	0	0.29086982	
22	Cluster4Agent4	1.218013835	0	3	3	0.822769039	0	0.67550057	
23	Cluster4Agent4	0.528559527	0	4	1	0.528559527	0	1	

Bibliography

[1] H. e. a. Chou, *Group Project*, 2015 (accessed June 6, 2015). http://github.com/kyotoprotocol/kyoto/raw/development/report/report.pdf.

[2] M. O. Infrastructure, *Energy Sector Strategic Plan*, 2015 (accessed February 2, 2015). http://www.minecofin.gov.rw/fileadmin/templates/documents/sector_strategic_plan/Energy_SSP_June_2013.pdf.

[3] N. Grid, *Customer service and network reliability*, 2015 (accessed February 2, 2015). http://www2.nationalgrid.com/responsibility/how-were-doing/grid-data-centre/Customer-service-and-network-reliability/.

[4] IEA, *IEA - Access to Electricity*, 2015 (accessed February 2, 2015). http://www.worldenergyoutlook.org/resources/energydevelopment/accesstoelectricity/.

[5] J. Pitt, *Holonic Institutions for Multi-Scale Polycentric Self-Governance*, 2015. Draft Unpublished Paper.

[6] Elinor Ostrom, "Governing the Commons: The Evolution of Institutions for Collective Action," 1990.

[7] presage2.info, *Presage2: About*, 2015 (accessed February 2, 2015). http://www.presage2.info/w/Main_Page.

[8] e.quinox, *e.quinox About*, 2015 (accessed January 27, 2015). http://e.quinox.org.

[9] e.quinox, *e.quinox Energy Kiosks*, 2015 (accessed January 27, 2015). http://www.e.quinox.org/index.php/our-solutions/energy-kiosks.

[10] e.quinox, *Hydro Kiosk*, 2015 (accessed January 27, 2015). http://e.quinox.org/index.php/projects/hydro-kiosk.

[11] e.quinox, *Design and Construction of a Pico Hydro Power System in Rural Rwanda*, 2012 (accessed Februar 2, 2015). http://e.quinox.org/reports/equinox_hydro_2012.pdf.

[12] e.quinox, *The Stand Alone Model*, 2012 (accessed Februar 2, 2015). http://e.quinox.org/reports/e.quinox%20Stand%20Alone%20Model.pdf.

[13] TNEI, *Web Based Constraint Analysis - Lincolnshire Low Carbon Hub*, 2015 (accessed February 2, 2015). `http://www.ipsa-power.com/education/2013-user-group-meeting/constraint-analysis`.

[14] D. of Energy and C. Change, *Electricity Market Reform Consultation Document*, 2015 (accessed June 15, 2015). `https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/42636/1041-electricity-market-reform-condoc.pdf`.

[15] J. Pitt, J. Schaumeier, D. Busquets, and S. Macbeth, "Self-organising common-pool resource allocation and canons of distributive justice," in *Self-Adaptive and Self-Organizing Systems (SASO), 2012 IEEE Sixth International Conference on*, pp. 119–128, Sept 2012.

[16] B. Roberts, *IEEE-Substation-Scale and Community Energy Storage*, 2015 (accessed June 15, 2015). `http://smartgrid.ieee.org/january-2013/739-substation-scale-and-community-energy-storage`.

Index