UNIVERSITY OF ZAGREB

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

MASTER THESIS Nr. 610

INTELLIGENT TRADING AGENT FOR POWER TRADING BASED ON THE REPAST TOOLKIT

Ivana Pranjić

Zagreb, June 2013

Contents

List of abbreviations

List of figures

Introduction

Most of the existing systems in the world, both natural and artificially created are non-linear and show complex behavior which cannot be predicted by simply decomposing the system into components and analyzing those components individually. These systems represent interesting subjects to researchers from various academic fields who use different methods for their analysis and simulation. They can be defined as complex systems consisting of many components able to adapt to their environment and whose interactions may yield emergent outcomes. A common method for modeling such systems is agent-based modeling, which is able to simulate interactions of those components, i.e. agents, and evaluate their effect on the whole system.

The electricity market is considered to be one of the most complex systems in the world due to the fact that energy supply and demand must be balanced in real-time. Recently, electricity markets around the world have been undergoing modernization processes which resulted in introduction of modern power grids called smart grids. The purpose of the smart grid is to provide a competitive environment for all market participants and enable two-way communication between consumers and other market entities. Power Trading Agent Competition simulates a modern electricity market in which agents in roles of brokers compete with each other in order to maximise their profit. The purpose of this thesis is to implement a trading agent that acts as a broker in the PowerTAC market using the Repast toolkit.

First chapter defines complex adaptive systems and their characteristics. Agent-based modeling approach and its key features, as well as the most popular modeling toolkits are described in the second chapter. The third chapter gives an overview of modern electricity markets and smart grids, whereas agent-based models of electricity markets are described in the fourth chapter. A brief description of PowerTAC competition is provided in the fifth chapter. Finally, the sixth chapter describes experiences while implementing the PowerTAC broker in the Repast toolkit.

1. Background and motivation

For years, the scientists considered the world to be a linear system in which simple cause-effect rules can be applied. They thought that if they broke down a system into parts and understood each of those parts individually, they would understand the system as a whole and, therefore, be able to predict and control its behavior. However, most of the existing systems, whether they are natural or artificially created, show signs of complex behaviors which arise from various non-linear interactions among their components [1]. There are numerous examples of these systems, for instance ecological, economic, human society, biological systems, electrical grid, telecommunications network, etc. For years their characteristics and behavior have been of interest to researchers in various scientific fields such as artificial intelligence, computer science, economics, mathematics, complexity theory, numerous areas of engineering etc. Hence, they have begun to combine methods of theoretical, applied and experimental research in order to address the analysis of such complex systems, which have become known as Complex Adaptive Systems (CAS) [2].

1.1. Complex Adaptive Systems

The term „complex adaptive systems“ was created during 1980s at the Santa Fe Institute[1] by Murray Gell-Mann, the Nobel laureate in physics, and John Holland, the founder of the domain of genetic algorithms. In general, it refers to the same systems that can be described and simulated by Multi-Agent Systems (MAS). Natural sciences studying these systems, such as physics, biology and chemistry offer the theoretical background, whereas a combination of theoretical and experimental methods, such as mathematics and computer simulation, is used for their analysis [2].

It is difficult to give a unique definition of complex systems and complexity in general, because there is no simple way for describing those systems. Therefore

[1] http://www.santafe.edu/

various definitions of complex systems have been used. A complex system can be defined as [4]:

- A highly structured system showing variations in its structure that are hard to predict;
- A system whose evolution depends greatly on initial conditions or small changes;
- A system with a large number of independent components between which multiple interactions occur;
- A system that is by design or function difficult to understand and verify;
- A system that is constantly evolving over time.

Complex systems are not to be confused with complicated systems. A complicated system usually has a top-down design and only one major property corresponding to its function, whereas a complicated system consists of many different components which are connected in such a way that all of them have to work harmoniously to accomplish the system's function. Furthermore, complex systems are more fault-tolerant because of their flexibility and ability to adapt, which means that a defect of one element does not affect the function of the whole system, like it would in a complicated system [4].

Complex adaptive systems are complex systems made up of numerous interacting components and have the capacity to adapt themselves according to the changes that occur in the environment [2]. There is no unique definition of CAS, instead different researchers provide various definitions:

- John H. Holland*: "*A complex adaptive system is a dynamic network of many agents which may be represented by cells, species, individuals, firms or nations acting in parallel, constantly acting and reacting to what the other agents are doing. Any coherent behavior that occurs in the system arises from competition and cooperation among the agents themselves. Every moment many agents make individual decisions that impact the system's overall behavior. Complex adaptive systems can continue to function only if they continuously adapt to the constantly changing environment."[3]
- Murray Gell-Mann: "A complex adaptive system acquires information about its environment and its own interaction with that environment, identifying

regularities in that information, condensing those regularities into a kind of schema or model, and acting in the real world on the basis of that schema."[5]

A simplified representation of complex adaptive systems is shown in Figure 1.

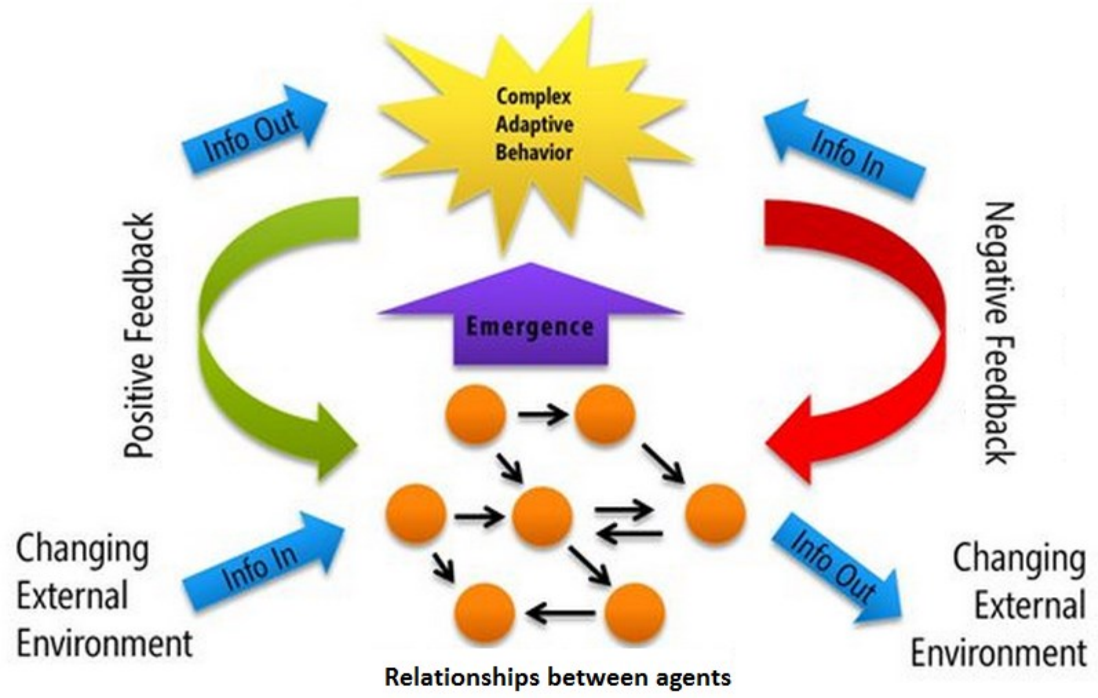

Figure 1: Simplified representation of CAS[2]

As noted earlier, complex adaptive systems are dynamic systems with the ability to adapt and evolve with a changing environment. They can also be described as collections of individual interacting components that interact with one another and react to the environment they populate [6]. The following chapter describes some of their characteristics.

1.2. Characteristics of Complex Adaptive Systems

The field of studying complex adaptive systems is a growing academic discipline often referred to as complexity science. Rather than being a single theory, complexity science is interdisciplinary and it includes multiple theoretical

---

[2] Adapted from: http://blogs.msdn.com/b/zen/archive/2010/08/10/the-new-world-of-emergent-architecture-and-complex-adaptive-systems.aspx, accessed June 2013

approaches while trying to address numerous questions related to adaptable systems [4]. Despite their differences, research has shown that many complex adaptive systems share a number of characteristics, such as:

- **Self-organization and adaptation** – in order to increase their efficiency and effectiveness, systems respond to the feedback from the environment by changing in a way that would improve their performance [3]. Adaptation is the ability to learn from experience, it describes changes in a system that help adjust the system to the conditions in its environment.
- **Emergence** - the overall behavior cannot be explained as the sum of individual parts of the system; there is no centralized control mechanism that determines system behavior. The evolution of complex systems can yield results that cannot be predicted on the basis of initial conditions.
- **Non-linearity** – Changes in the input characteristics are not linearly correlated with the outcome, which means that small changes can lead to unpredictable consequences on overall behavior, or vice-versa [2].
- **Co-evolution** – It describes the interdependent evolution of two or more interacting entities in which the changes of each entity influence the evolution of the others. The same applies to complex adaptive systems in a way that a system exists within its own environment and adapts to it as the environment changes, causing the environment to change as well [2],[3].

2. Agent-based modeling

Due to the previously described characteristics of complex adaptive systems, which is a result of interaction between individual components and their environment, an interdisciplinary approach is required for addressing the issues related to these systems [6]. The most common method for modeling complex adaptive systems is agent-based modeling (ABM) [7],[8]. ABM has gained increasing attention over the past 15 years and has been used successfully in many areas including supply chains, market analysis, economics, ecology, etc. ABM is related to a variety of other simulation techniques, including discrete event simulation and artificial intelligence [7].

2.1. Key features of agent-based models

Agent-based models use autonomous entities capable of making decisions called agents. It can be said that an agent is a software representation of a decision-making unit [8]. Agents make decisions based on a set of rules, they are able to learn and adapt to current conditions in their environment.

Agent-based modeling is a methodology which enables creation and analysis of various artificial environments filled with agents whose interactions are not trivial [9]. It is a technique that serves as a platform for imitating the non-linear characteristics of real-world complex systems and it is most suitable when the system of interest shows complex and adaptive behavior [2].

Agent-based models belong to the class of computational models. They put a special emphasis on the interactions of agents to evaluate their effects on the system as a whole. The key characteristic is that simple behavioral rules on a micro-level result in complex outcomes on a macro-level [4]. Constant competitive interactions between the agents are considered the main feature of ABM [8]. General characterization of agents is given by the following properties [9]:

- **Autonomous** – They are able to make decisions and take actions without the need for human intervention, therefore every agent has control over its state;

- **Adaptable** – Agents can be intelligent and make decisions based on their current state and the conditions in their environment, they can adapt their behavior by learning from previous actions and consequences.
- **Interactive** – Agents interact with other entities in the environment, such as other agents or even humans, through some kind of agent communication language;
- **Reactive** – They are aware of their environment and they observe the situation in order to react to inevitable changes in that environment on time;
- **Goal-oriented** – Agents are able to make conclusions and take the initiative in accordance with their goals, instead of just acting in response to the changes in the environment

Agent-based modeling is useful when agents show complex behavior such as learning and adaption and when the interaction between the components of the observed system is complex. There are several reasons why agent-based models are more appropriate than other methodologies when modeling complex adaptive systems [12]. They are able to:

- Grasp **emergent phenomena**, which are results of the interaction among many individual entities. Emergence is, as described earlier, a feature of complex systems and cannot be simply explained by understanding their parts individually. In other words, the whole is more than the sum of its parts. Emergent phenomena can have properties which are independent of the properties of single components that make up the system.
- Describe the system in a **natural way**, especially if that system is composed of real-world entities whose individual behaviour is complex. In that case the agent-based models are more adequate and are considered a more realistic approach.
- They are **flexible** in many different ways. For instance, an agent-based model can be applied to almost any complex system. Agents can be mobile, concurrent, interactive, goal-oriented, the level of their complexity, intelligence and the ability to evolve can be tuned. Finally, adding more agents to an existing model is simple, they can be aggregated or divided into groups, which makes agent-based models appropriate for experimenting with different complexity levels [8].

2.2. Agent-based modeling toolkits

There are numerous agent-based modeling toolkits available for use, which enable individuals to develop their own agent-based models and applications. An overview of the most popular ones is given in the following chapter. Most of them are open-source and free for non-commercial use.

2.2.1. StarLogo and NetLogo

StarLogo is an extension of the Logo programming language designed for educational purposes and it uses Java interpreter and interface. The main principle of StarLogo is modeling behavior of decentralized complex systems [17].

NetLogo is a cross-platform programmable agent modeling environment that uses a modified version of Logo. It was designed for educational purposes i.e. teaching concepts of complex systems using turtles, patches and observers [18]. However, it can also be used for development of complex models that simulate social and natural phenomena. It was designed to be simple enough to make programming more accessible and allow beginners to quickly learn the principles of agent-based modeling, but without limiting what expert users can do [7].

2.2.2. Multi-Agent Simulation of Neighbourhoods (MASON)

MASON is a modular multi-agent simulation library written in Java that supports custom discrete-event simulations containing large number of agents [20]. Apart from the library itself, it also includes a collection of 2D and 3D visualization tools capable of generating charts, graphs, screenshot and short movies. Visualization is completely optional and it can be added to or removed from any model. Models created using MASON are platform independent. As a toolkit, MASON contains all of the functionalities needed for simple simulations [19].

2.2.3. Swarm

Swarm is an object-oriented multi-agent software platform originally developed for the creation of simulation models in the field of artificial life [7]. The basic unit of simulation is the swarm which is made up of multiple agents that execute scheduled actions. Since Swarm supports hierarchal modeling, an agent can include swarms of other agents [21]. Due to the object-orientation and nesting support, models can be combined and nested within other related models. Swarm influenced the development of other simulation toolkits such as Repast 3 and Repast Simphony described below [7].

2.2.4. Recursive Porous Agent Simulation Toolkit (Repast) 3

Repast 3 is a suite of three libraries written in several languages - Repast J (Java), Repast .NET (C#) and Repast Py (Python), which differs it from Swarm, from which it has borrowed many concepts. For beginners it is recommended to use Repast Py because of the visual interface it provides, whereas advanced developers are encouraged to use Repast J and Repast.NET [24]. During the implementation of the three libraries it was concluded that in order to allow the flexibility of the library and models, the design must be object oriented and modular. Furthermore, Repast 3 enables modeling in different programming languages which simplies entry for new users and makes systems approachable to a wider audience in comparison with single library approach [10],[22].

2.2.5. Repast Simphony

Repast Simphony[3] is a toolkit for agent-based modeling of complex adaptive systems written in Java. The most recent version was released on 5th March 2012. Repast Simphony was created on the basis of the previously described Repast 3 but offers new approaches of model development and simulation execution [23].

The graphical user interface which allows the user to control the simulation by starting, pausing and resetting different models using, for example, different initial parameters is one of the main features of Simphony. It allows users to analyse the behavior and simulation results of the same model, but with various starting conditions.

Simphony provides visualization of custom data sets through graphs and charts during runtime, which can be useful for simulation monitoring and result analysis. It also supports external statistical programs, like R, Weka, Pajek, Matlab etc. [38].

---

[3] http://repast.sourceforge.net/repast_simphony.html

3. Electricity Market as a Complex Adaptive System

Electricity is a commodity that can be traded like any other, but what is specific about it is that it cannot easily be stored, so supply and demand must be balanced in real time, which makes the electricity market one of the most complex markets in the world [11]. In the past almost all of the electricity markets were vertically integrated monopolies owned by private companies or the state itself. All of the components, including electricity generation, transmission and distribution were managed by a single electric utility. Over the last few years most electricity markets have been undergoing a process of restructuring and modernization with the aim of ensuring greater competitiveness and lowering prices. These processes include moving away from those vertical monopolies to a more liberalized market [29].

Nowadays most electricity markets are based on centralized systems, meaning that electricity is mainly generated in large power plants and delivered to end consumers through the transmission grid. There are numerous drawbacks of centralized systems. For instance, while transmitting electricity over long distances, due to its nature energy losses are inevitable, causing higher prices to end customers. Another problem is reliability, i.e. in case of a failure somewhere in the grid, large number of customers could end up deprived of electricity. For this and various other reasons, such as economic and environmental, distributed systems have been given more and more attention over the past years [16].

3.1. Distributed electricity generation

There are several reasons why distributed generation has recently been gaining on popularity: the liberalisation of the electricity markets, environmental reasons i.e. reducing greenhouse gas emissions, consumers' requirement for reliable electricity supply etc. [13]. Nowadays the term „distributed generation“ involves various micro- or small-scale generating units which are located close to the areas where the power will be consumed, which are described in the following chapter.

3.1.1. Distributed generation technologies

Some of the most common technologies used for distributed electricity generation are:

- **Microgrids** – A relatively new technology often described as local small-scale version of the centralized grid. It consists of one or more generation units that can operate autonomously but also as a part of a bigger grid. Units that a microgrid can contain are various, for instance wind turbines, solar panels and fuel cells and, since it can function on its own, it provides a reliable power source to the customers. Another advantage is the fact that the waste heat can be used for heating space and water of nearby households and offices [14].
- **Reciprocating engines** – They are running on natural gas, bio-gas or fuel and use compression to convert mechanical energy to electricity. They are common and widely used due to their high efficiency and low investment costs [15].
- **Fuel cells** – Natural gas fuel cells can be used for combined heat and power (CHP) generation. They can be as small as a refridgerator and have the capacity to satisfy the demand for electricity and heating of households and smaller buildings. The power is generated by an electrochemical process instead of combustion and their efficiency can be as high as 90% [14].
- **Renewable sources** – Wind turbines, solar panels, photovoltaic cells, geothermal sources etc. have been gaining increasing interest in distributed electricity generation [15]. Their biggest advantage is their sustainability, i.e. the fact that they will never run out and that they are more environment friendly since they do not emit gases that cause pollution and global warming.

3.1.2. Benefits of distributed power generation

Restructuring and deregulation of electricity markets has enabled distributed generators to enter the market and exploit flaws of traditional generation by reducing entry barriers [15]. In comparison with traditional centralized power generation, distributed generation has many advantages. For instance, considering the fact that generation units are located close to their end consumers, the heat can be used productively for heating water and space, which is generally not the case with large power plants. Furthermore, local power generation also reduces demand and possible congestion in the rest of the grid during peak times, while increasing reliability at the same time.

Distributed generation also helps with improving power quality, since there is no long distance transmission and line losses [14]. As stated in the previous chapter, distributed generation units are independent from the rest of the grid, so they can serve as a backup to consumers who require reliable and uninterrupted supply of energy, such as hospitals, airports, industrial facilities etc. One of the biggest advantages of distributed generation is reducing gas emissions through the use of renewable sources and cogeneration, i.e. CHP [15].

In order to meet the growing demand for electricity and integrate various new elecricity generation units, it is important to establish an intelligent grid able to provide secure, sustainable and stabile supply of energy – the smart grid, described in the following chapter.

3.2. Smart grid

The smart grid involves active infrastructure capable of satisfying increasing demand for electricity production, transmission and distribution using technologies that enable management, supervision and control of the entire system. It supports a two-way flow of information and electricity between the consumers and the grid, meaning that they have insight into their consumption and can give the excess of power back to the grid during increased demand [26].

Some of the principles of the smart grid are [25]:

- Integration of numerous electricity producers of all capacities and generation technologies in a sustainable and reliable system;
- Allowing consumers to participate in optimising the system;
- Stimulating market competition and liberalization;
- Encouraging distributed generation and use of renewable sources;

- Providing consumers with relevant information about their consumption and offering them to choose from various energy suppliers;
- Ensuring the best possible utilization of existing power infrastructure;
- Balancing electricity supply and demand in real time.

3.2.1. Smart Grid Components

Although there is no unique architecture of the smart grid, it can be divided into these components: demand management, grid management and distributed generation, which was described in Chapter 3.1. A brief overview of the remaining two is given in the following section.

Demand management

The key concept when talking about demand management is advanced metering infrastructure (AMI). It consists of a smart meter, which is located on the consumer side, and the infrastructure needed for two-way flow of information between the consumer and the utility [27]. The smart meter collects the data from consumer's appliances and sends it to the utility and it is also capable of controlling appliances, i.e. turn them off and back on if needed. AMI gives consumers an insight into their energy consumption and greater control over the amount they use and when they use it, whereas utility companies use AMI for monitoring and billing purposes and sending alerts in case of increased energy demand asking consumers to reduce their consumption [28]. Furthermore, utility companies can offer consumers variable-rate prices through online energy dashboards, so that consumers can monitor their consumption, shift it to lower-price period and reduce their electricity bills [26],[28].

Grid management

Real-time monitoring of electricity production and consumption within the smart grid requires appropriate IT solutions. They allow utility companies to monitor the grid's performance, detect potential failures and take appropriate actions to prevent them. Software solutions give utility companies full insight into electricity production and consumption, therefore they can also be used for grid optimization, improving its efficiency and reliability [27]. Many of the grid components are vulnerable to cyber and terrorist attacks or natural disasters, therefore utility companies must be able to actively monitor their security via live video or sensors [28].

A general architecture of a smart grid is depicted in Figure 2.

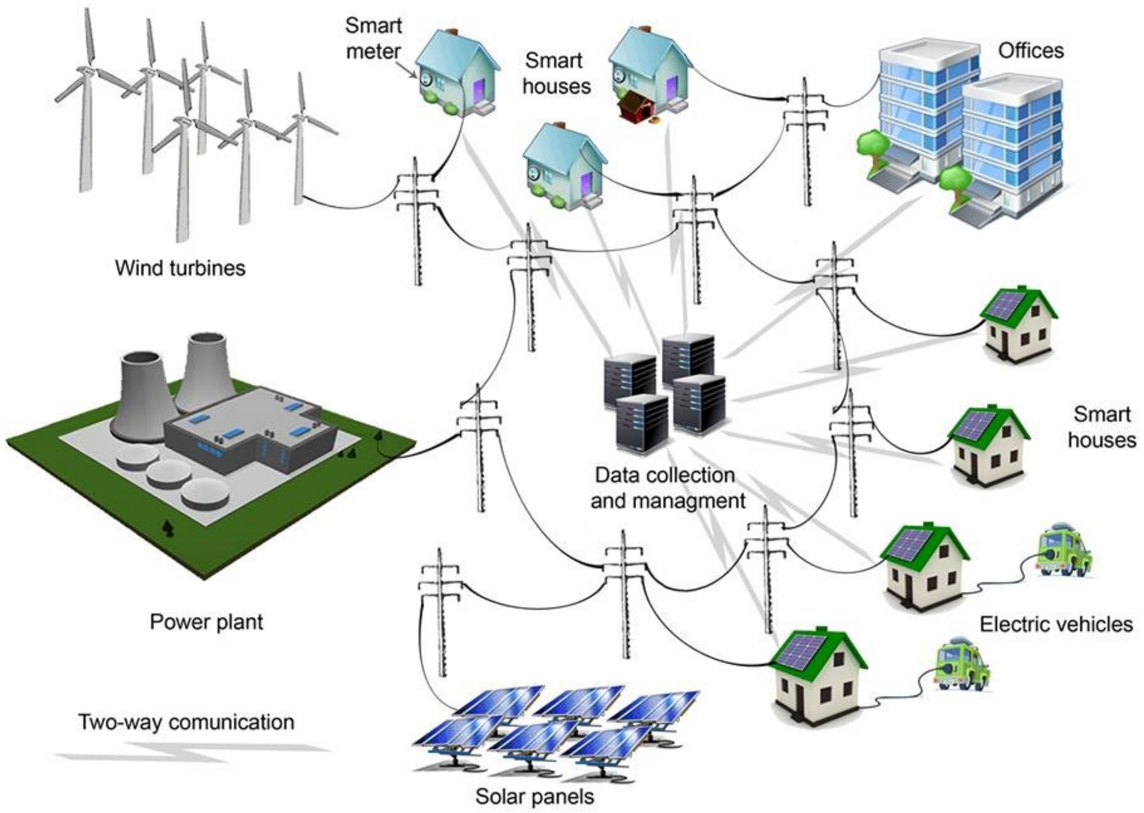

Figure 2: Smart grid architecture

3.2.2. Smart Grid Benefits

It is expected that the smart grid, once fully implemented, will have benefits in the following areas [25]:

- **Electricity reliability** – A grid is considered reliable if it provides consumers with electricity of adequate quality when they need it. Reliability can be measured by the number and duration of power failures. To some consumers even short power outages can cause severe problems and damage. Various technologies that are a part of smart grid concept can be used to improve reliability. For instance, AMI can quickly identify power failure, demand response can control consumers' consumption by turning of non-essential appliances during peak demand, distributed generation units can serve as backup and ICT can enable utility companies to use the information they collect from customers for improving grid reliability.

- **Environmental benefits** – Environmental improvements include reducing greenhouse gas emissions and pollution by using renewable sources, cogeneration and electric vehicles. The smart grid gives consumers the option to choose their electricity supplier and encourages renewable sources and CHPs.
- **Economics** – Improvements in economics involve introducing new services and products such as storage devices and smart appliances, lowering electricity prices for consumers and giving them the option to sell excess energy back to the grid, offering new price rates depending on the time of day which stimulates them to reduce consumption during peak demand. AMI and ICT allow collecting consumption and billing information which are sent to consumers so they have an insight into the amount of energy they are consuming and its price.
- **Efficiency** – Efficiency can be improved by reducing peak demand by giving the consumers information about their consumption and pricing and using distributed generators and local storage. Smart grid also reduces production and transmission costs, transmission and distribution line losses as well as overall customer energy consumption by motivating them to consume power more efficiently. More and more consumers with high consumption levels accepted the demand response concept which helps with reducing peak load. Cogeneration, i.e. CHP generation has gained increased interest due to its ability to use the heat from generation productively instead of wasting it, which results in increased efficiency.
- **Security and safety** – Smart grid is constantly monitored in order to detect potential hazardous events and avoid them. It is also able to detect attempts of cyber and terrorist attacks. AMI gives operators the ability to disconnect customers' loads to protect them in case of any danger. Large number of distributed generation units also help with security and robustness, giving consumers an alternative power supply if the grid is out of service for some reason, such as attacks or natural disasters.

4. Agent-Based Modeling of Electricity Markets

The evolution of electricity markets and their recent restructuring, i.e. separating processes of electricity generation, transmission and distribution has given many new participants an opportunity to enter the market while encouraging competition among them. As a result, a growing need arose for finding appropriate modeling approaches that would be able to simulate complex behaviors of all market participants and explore their impact on the environment in which they operate [29]. There are several types of models commonly used in the study of electricity markets: optimization models, equilibrium models and simulation models [33].

Optimization models focus on the profit maximization problem of a single firm competing in the market, and are the least used of the three types. Furthermore, equilibrium models represent the overall market behavior and attempt to solve mathematical equations that represent decisions of all market participants. However they are unable to reproduce complex behaviors because they only represent the economic side of the system and therefore cannot cover all important factors that determine efficient market operations, which leads to an incomplete representation of the system [33].

On the other hand, simulation or agent-based modeling, also referred to as agent computational economics (ACE), represents a more appropriate approach when the problem under consideration is too complex (nonlinear or dynamic) to be addressed by traditional methods. Agent-based modeling has the ability to overcome many of the computational limitations of equilibrium models, which were unable to model more than simplified transmission systems and a few active decision-makers [31]. Reasons for recent increasing attention in agent-based modeling are its numerous advantages in modeling complex systems, which will be described in the following sections.

4.1. Advantages of agent-based modeling of electricity markets

Many of the tools developed for electricity market modeling over the past 20 years were based on the assumption that decisions are made in a centralized manner. Although they may continue to provide useful insights into the way electricity systems, they cannot adequately analyze the complex interactions among all the market participants [29],[30].

ABM has the capability of modeling complex behaviors of various participants in large-scale markets. The fundamental approach of agent-based models is simulating real world systems with a collection of interacting agents and their emergent behaviour [12]. As noted earlier, agent-based models provide the possibility to analyze the dependencies of the market players, i.e. participants (micro level) and the overall structure of the market (macro level). This approach proved to be appropriate for assessing the process of electricity market evolution. Power generating units can be modelled as agents, they are autonomous, have a social ability in a sense that they interact on the market with each other, they are reactive because they collect information about their state and the environment to decide on future actions [8],[12].

In order to explore the influence of interdependencies and interaction between market participants on the evolution of electricity markets, a flexible framework offered by agent-based simulation is required. Agents are able to learn from past experiences so they make their decisions based on their past information. This important feature is neglected by traditional models, such as equilibrium or game theory [31]. In other words, by making conclusions from past experiences, agents are able to improve their decision making and adapt to changes in the environment. Based on its own perception of the environment and the actions of other participants, an agent makes decisions on its further actions. These features make ABM an interesting and powerful tool for simulation and testing of various market designs [32].

Furthermore, interactions between agents, each having their own goals, characteristics and ability to make decisions, can result in various complex outcomes. Because of that, there are no assumptions or implications made in advance so that unexpected behavior as well as new cognitions and conclusions can arise from simulations.

The electricity market outcome is not determined just by the economic aspect, there are also environmental and social factors originating from many market participants as well as technical constraints whih cannot be ignored. Therefore modeling electricity markets presents a challenge which can be met by agent-base modeling [32].

Clearly, the success of new restructured electricity markets depends greatly on market design, risk management, balancing supply and demand in real time and reliability control along the power supply chain. That being said, detailed and thorough testing of system designs in a simulated environment is crucial before implementing the designs in real life [34]. The following chapter describes some of the existing agent-based models of electricity markets.

4.2. Existing agent-based models of electricity markets

The OPTIMATE Model

OPTIMATE is a open platform used for validating rules of various existing market designs as well as testing new ones including massive intermittent energy producers dispersed in multiple western European electricity systems [32]. Its goal is to help developing solutions compatible with a virtual unique European grid and regional electricity network management processes by estimating expected outputs of new market designs.

Intermittent generation technologies, especially renewable sources have characteristics that do not fit easily in the current electricity market. For example, day-ahead wind or solar production forecasts are not reliable with wind energy characterised by a 20% forecast error [35]. Making intermittent electricity producers profitable is possible if using flow-based market coupling[4] [32]. Some European markets, such as Germany, Belgium and France have already implemented a flow-based market coupling mechanism, but to confirm that intermittent generation in Europe is profitable, a formal study must be carried out. In order to fully understand the effect of large-scale intermittent energy generation in European electricity markets, Optimate uses agent-based modeling to analyze behaviors of all market participants assuming different electricity market designs [35].

[4] Market coupling uses implicit auctions in which players do not actually receive allocations of cross-border capacity themselves but bid for energy on their exchange.

The AMES model

The Agent-based Modeling of Electricity Systems (AMES) Wholesale Power Market Test bed is an open source agent-based model designed to investigate the effects that the recent restructuring process has had on wholesale market efficiency and realiability. Special emphasis was put on electricity markets that operate over transmission grids prone to congestion [36]. The secondary goal of further development of the AMES model was to unify market participants, regulators and researchers in order to develop the model appropriate for research and teaching. AMES test the performance of market design recommended by the Federal Energy Regulatory Commission (FERC) [32].

Market participants include bulk-power buyers, sellers, producers (GenCos), and an independent system operator (ISO). Bulk-power buyers aim at securing electricity for their retail customers, whereas bulk-power sellers, i.e. generation companies try to maximize their daily earnings by choosing the right supply offer to bid on the day-ahead market. The ISO guards the efficiency of the wholesale electricity market by operating a day-ahead market and conducting optimal power flow analysis in order to anticipate grid congestion. However this model does not take system disturbances, such as the volatile electricity production or sudden outages, into consideration. As a result there is no modeling of real-time market trading [32].

The EMCAS model

The Electricity Market Complex Adaptive System (EMCAS) model was developed at the Argonne National Laboratory in Chicago. It is an electronic laboratory that simulates strategic behavior of all market participants in order to explore possible effects of market rules and explore how the market deregulation and restructuring processes effect electricity reliability and prices [29].

The model includes numerous different agents such as generation companies (GenCo), independent system operators (ISO) or regional transmission organizations (RTO), transmission companies (TransCo) and consumers. All EMCAS agents learn from past information and adapt their behavior based on the outcomes of their previous actions. GenCos are modelled as agents who evaluate their past strategies, analyze the competition, predict future conditions and explore new strategies in order to utilize the limitations of the power grid and the market [29]. GenCos can also change their bidding strategies, subject to factors such as capacity reserve margins, forecasted demand, and generator risk characteristics [32]. The ISO agent operates and manages the transmission grid while maintaining reliability, publishes demand, supply and weather forecast to all participants, gathers bids and offers submitted by the generatiing and demand companies and calculates market clearing price. Furthermore ISO enables communication between generation and demand companies through various markets, such as forward, bilateral contract, pool and ancillary services markets. Remaining market participants, such as TransCos, regulators, demand and distribution companies do not exhibit strategic behavior. Transmission grid configurations are obtained by the North American Electric Reliability Corporation (NERC) in order to apply the model on the electricity market in the mid-western USA [32].

Power Trading Agent Competition

Power Trading Agent Competition (PowerTAC)[5] is a competitive electricity market simulation in which agents in the role of brokers buy and sell energy in the retail and wholesale market to meet their customers' needs. They act as mediators between electricity producers and consumers with the intent of earning a profit. During the competition agents seek to maximize their profit by offering various types of tariffs to retail customers and concluding individual contracts with large customers, while trying to balance energy supply and demand in real time. The winner is the agent which has the biggest profit at the end of the competition [37]. The competition is described in more detail in the following sections.

[5] www.powertac.org/

5. PowerTAC

Every team participating in PowerTAC creates its own intelligent agent, i.e. broker that aggregates energy supply and demand from its clients and trades on wholesale and retail markets. On the retail market, which refers to households with “smart” appliances, smaller enterprises, plug-in electric vehicles, smaller producers and households with solar panels, brokers try to attract as many customers as possible by offering tariff contracts, whereas on the wholesale market they negotiate with other participants such as generation companies and each other. Apart from these participants, there is also a distribution utility that owns the distribution grid and manages the distribution process. Main entities of the PowerTAC competition are shown in Figure 3.

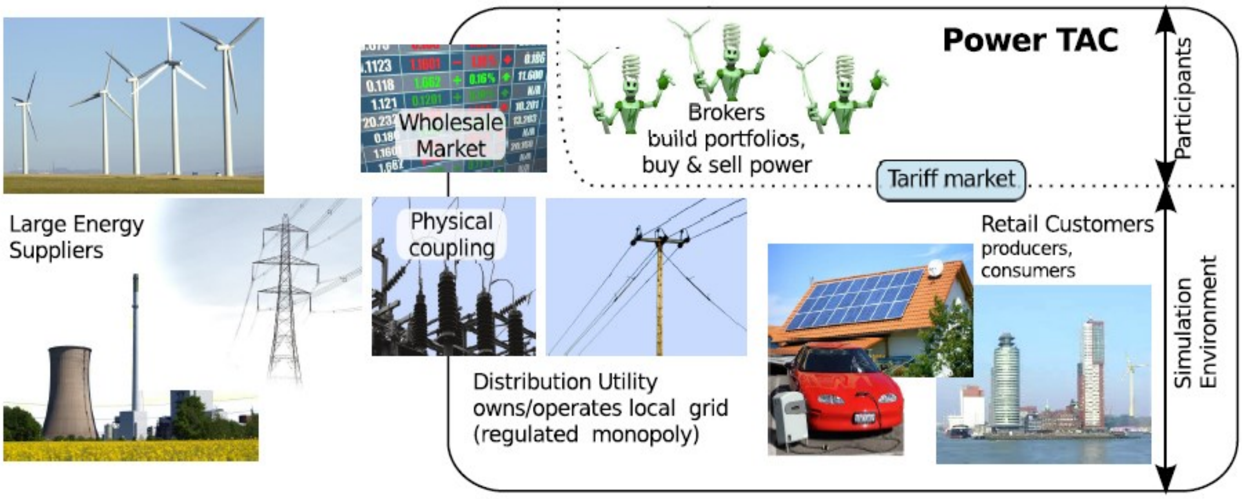

Figure 3: Main entities of the PowerTAC competition [37]

The time in the competition proceeds in discrete time blocks called timeslots or TAC-hours that represent 1 hour in the simulation or 5 seconds in real time. The competition runs for approximately 60 simulated TAC-days, which take up to more than 2 hours in real time. At the beginning of the simulation all agents receive bootstrap data which contains information about customer records, market transactions and weather reports in the previous 2 weeks.

5.1. Customer market

In the customer market, brokers can buy and sell energy through offering tariffs and by negotiating individual contracts with large consumers and producers[6]. Most of the customers interact with brokers by subscribing to tariffs the brokers offer and producing and consuming energy. Customers can be energy producers and consumers, and some of them may even change their role depending on the current market situation. For instance, plug-in electric vehicles act as consumers most of the time, but they can also switch to being producers by feeding the energy back to the grid in case of energy shortage. Therefore they are called *prosumers.*

5.1.1. Customer types

There are several types of customers in PowerTAC:

- Households – mostly in the role of consumers with the possibility of producing energy using solar panels;
- Enterprises and offices – special type of consumers consuming electric energy evenly throughout working hours
- Industrial facilities - similar to office buildings, but with larger variations and deviations;
- Plug-in electric vehicles – consume large amounts of energy, but they may also serve as producers in case of energy shortage in the grid
- Various institutions in the role of consumers – such as hospitals, universities, etc.

All customers can be divided by their power type, i.e. their role in the market:

- Consumption – Electricity flows from the grid to the customer;
- Interruptible consumption – Electricity flow towards the customer can be interrupted at certain times, for example when demand is bigger than the amount produced (certain appliances that support interruption can be shut down);
- Production – Produced electricity flows from the customer to the grid;

[6] However, this feature hasn't been implemented yet, but it is planned for PowerTAC 2014

- Storage capacity – used to store excess power or as a source during times of shortage, limited by the customer's capacity.

5.1.2. Tariff market

During the competition brokers try to attract as many customers in the retail market by offering a variety of new tariffs and modifying existing ones. Some of the possible tariff features are tiered rates[7], weekday/weekend rates, two-part tariffs[8], signup fees and bonuses, withdrawal penalties, interruptible rates, etc.

After creating a new tariff, the agent submits it to the market and waits for customers to subscribe to it. From the moment a customer subscribes to a tariff, that tariff becomes active. If a broker wants to modify or update its tariff, it has to supersed it with a new one and force customers to unsubscribe from the old one by revoking it. In case the superseding tariff does not exist, the customers are automatically subscribed to the default tariff offered by the distribution utility.

5.2. Wholesale market

PowerTAC wholesale market operates as a periodic double auction, meaning that buyers (for instance brokers) place their bids and sellers (i.e. generation companies) simultaneously place their asks, after which the clearing price is determined by the auctioneer. In this market brokers can, depending on their customers' forecasted supply and demand, buy and sell amounts of energy for future delivery (between 1 and 24 hours), hence the name "day-ahead market".

In the wholesale market agents negotiate with large generation companies (GenCos). Every bid (or ask) consists of the following: the broker's identification, the future timeslot for which the broker wishes to bid, the desired amount of energy in megawatt-hours and, optionally, the limit price per MWh[9]. At the beginning of every timeslot brokers receive information about their customers'

[7] The customer pays one rate for a predefined amount of energy and a different rate for remainder.

[8] The customer pays a daily rate and usage fee, depending on the amount of energy consumed that day.

[9] Limit price is the maximum price a broker is willing to pay for 1 MWh.

consumption, weather forecast and current financial situation. If the broker anticipates excess of energy, it can place a selling offer (i.e. an ask) on the market in order to sell a certain quantity of energy. The offer containing a positive amount of energy and negative price, it is considered a buying offer, otherwise it is a selling offer. At the end of every timeslot, after collecting all of the bids and asks, supply and demand curves are created in order to determine the clearing price and quantity. The clearing price is the one where these two curves cross, or in case they do not, it is set at the mean value of the highest buying and lowest selling offer. All of the higher buying and lower selling offers are accepted and cleared, whilst others are discarded.

5.3. Distribution utility

The distribution utility (DU) is the entity that owns the distribution network modelled as a regulated monopoly. Brokers pay the fee to the distribution utility for using the grid to transport energy to their customers. The fee is proportional to the amount of energy flowing from a broker to its customers.

Another important role of the DU is balancing the electricity supply and demand in real time. In cooperation with the ISO it maintains balance while keeping various factors, such as frequency, voltage and power, within acceptable limits. Brokers are encouraged to keep their consumers' demand and producers' supply in balance, otherwise they are charged balancing fees for causing imbalance in the grid.

At the beginning of the competition all customers are automatically subscribed to the default consumption and production tariffs that the DU offers. That way the customers always have the option to choose a default tariff if those offered by brokers are unacceptable. Competing brokers are supposed to offer such tariffs which would cause customers to unsubscribe from the default one and subscribe to theirs.

5.4. Broker

The goals of every competing broker are creating a good quality portfolio by offering various types of attractive tariffs to its customers and minimising the imbalance between energy supply and demand. The balance can be achieved and maintained by selling excess energy or buying predicted shortage of energy on the wholesale market. In every timeslot a broker can:

- Create a new tariff and submit it to the tariff market;
- Modify an existing tariff by superseding it with a new one;
- Place an offer (bid or ask) on the wholesale market;
- Create a balancing offer to provide controllable capacities to the distribution utility;
- Negotiate contracts with large customers (not yet implemented).

Figure 4 shows actions each broker performs and information it receives within a timeslot.

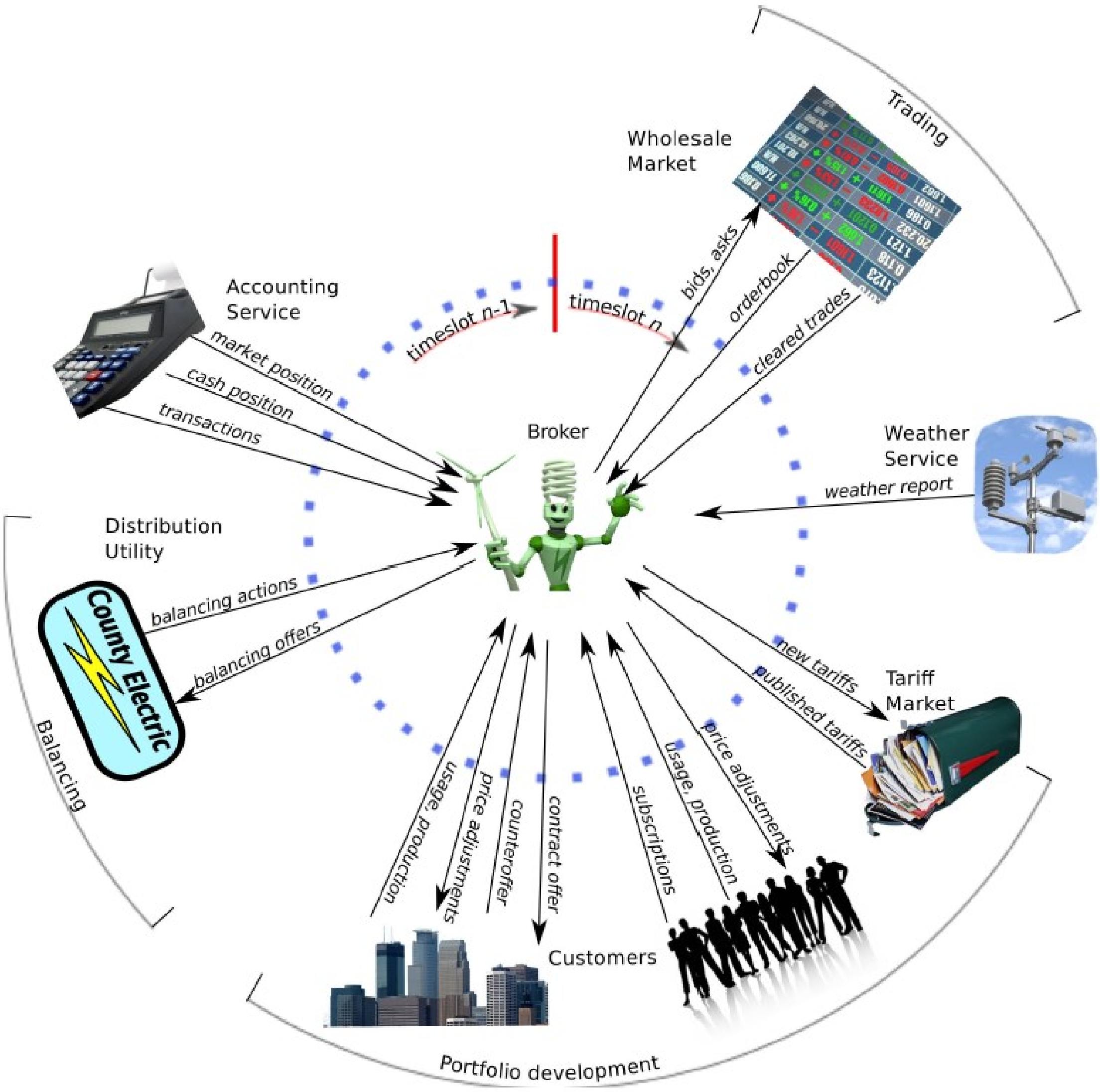

Figure 4: An overview of broker's activities in one timeslot [37]

Brokers manage their portfolios by creating tariffs and collecting information about their customers’ consumption and production capacities. In addition to that information, brokers also receive weather forecast, which can be very important in predicting supply and demand, as some producers such as wind farms and solar panels depend greatly on weather conditions. Furthermore, every timeslot the brokers receive information about their current financial situation, tariff transactions in the retail and clearing transactions in the wholesale market, distribution and balancing transactions, as well as market and cash positions.

6. Implementing PowerTAC broker in Repast

PowerTAC developers provide potential participants with a template of a broker called „*sample broker*". It is a fully working broker written in Java that serves as a foundation for further development. It offers a set of basic tariffs, one for each type of customer in the retail market and sends bids to the wholesale market. The source code can be acquired from the PowerTAC Github repository[10].

The sample broker implements core functionalities, such as communication with the server and time synchronization, which simplifies the entrance for new participants willing to develop their own competing broker. Some of the main functionalities and corresponding services are:

- **`PowerTacBroker`** – Core service of the sample broker, initializes other services, maintains the broker's clock and terminates the broker at the end of simulation. The communication with the simulation server, i.e. the simulator is achieved through sending and receiving XML messages using Java Message Service (JMS) technology. `PowerTacBroker` handles messages that control the simulation execution, such as *SimStart*, *SimEnd*, *TimeslotComplete* and *TimeslotUpdate*. Messages are received by `BrokerMessageReceiver` and dispatched further by the **`MessageDispatcher`** service. The `MessageDispatcher` service is also used for sending outgoing messages to the simulator. The JMS configuration parameters, such as the JMS URL, the server queue name and JMS connection setup are managed in the **`JmsManagementService`**.
- **`BrokerTournamentService`** is responsible for creating the connection with the tournament scheduler and logging into the tournament.
- **`ContextManagerService`** – A service which handles context and bank related messages such as *Competition*, which contains information about customer records, *BankTransaction* and *CashPosition* used to update the broker's bank balance, and *DistributionReport*, containing records about total production and consumption.

- **`PortfolioManagerService`** – A service responsible for the management of broker's portfolio, keeping track of customes' energy usage, specifying and managing tariffs.
- **`MarketManagerService`** – A service that is in charge of broker's interaction on the market, It handles all of the market related messages, such as *ClearedTrade*, *DistributionTransaction*, *MarketTransaction*, *MarketPosition* etc. It also collects customers' energy demand from the `PortfolioManagerService` and submits wholesale orders at every timeslot.

The idea of implementing the sample broker in Repast is to enable other potential participants, which are familiar with modeling agents in Repast, to integrate their models easily into the broker by creating a multi-agent system. A more detailed overview of the Repast Simphony toolkit is given in the following section.

6.1. Repast Simphony

Repast Simphony is a „richly interactive and easy to learn Java-based modeling system designed for use on workstations and small computing clusters“ [24]. Repast was originally developed in 2000 by North, Collier, Sallach, Howe and others at the University of Chicago in collaboration with Argonne National Laboratory[11]. Nowadays, Repast is being developed and managed by Repast Organization for Architecture and Design (ROAD), a non-profit volunteer group led by a board of directors whose members come from a wide range of academic, industrial and government organizations [22].

Repast Simphony was built on the lessons of Repast 3, but it was designed completely from the ground up, using a new code base [7]. The installation file can be downloaded from the Repast Simphony download page[12], along with

[10] https://github.com/powertac/sample-broker

[11] http://www.anl.gov/

[12] http://repast.sourceforge.net/download.html

installation instructions and tutorials explaining basic examples. It uses Eclipse[13] as its development environment.

Simphony provides a collection of classes for building and running agent-based simulations, collecting and displaying simulation parameters and other data through different types of charts and graphs and also enables integrating geographical information science (GIS) data directly into simulations [23].

Repast offers a variety of features including the following [38]:

- Ability to model agents using different programming languages, such as Java, ReLogo[14], Groovy[15] or flowcharts;
- Sample models, agents and their environment;
- Graphical user interface which allows the user to control the simulation during by pausing, resetting and restarting models, modify configurable parameters and use external tools, such as Pajek, Weka or R for processing simulation results;
- Point-and-click model development environment that generates Java classes;
- Concurrent discrete event scheduler which allows certain actions to be carried out at a specific time or in a specific order;
- Managing run parameters using the GUI;
- Data collection and charting;
- Batch and parallel runs;
- Simulation checkpointing and restoration using data storage;
- Various libraries for neural networks, regression, genetic algorithms and specialized mathematics;

[13] www.eclipse.org

[14] ReLogo is a special dialect of Logo programming language evolved from Repast and NetLogo.

[15] Groovy is a Java-like programming language that compiles to Java bytecode.

- Ability to model and visualize 2- and 3-dimensional environments and geographical spaces, including GIS support.

An example of the Repast graphical user interface during a simulation is shown in Figure 5.

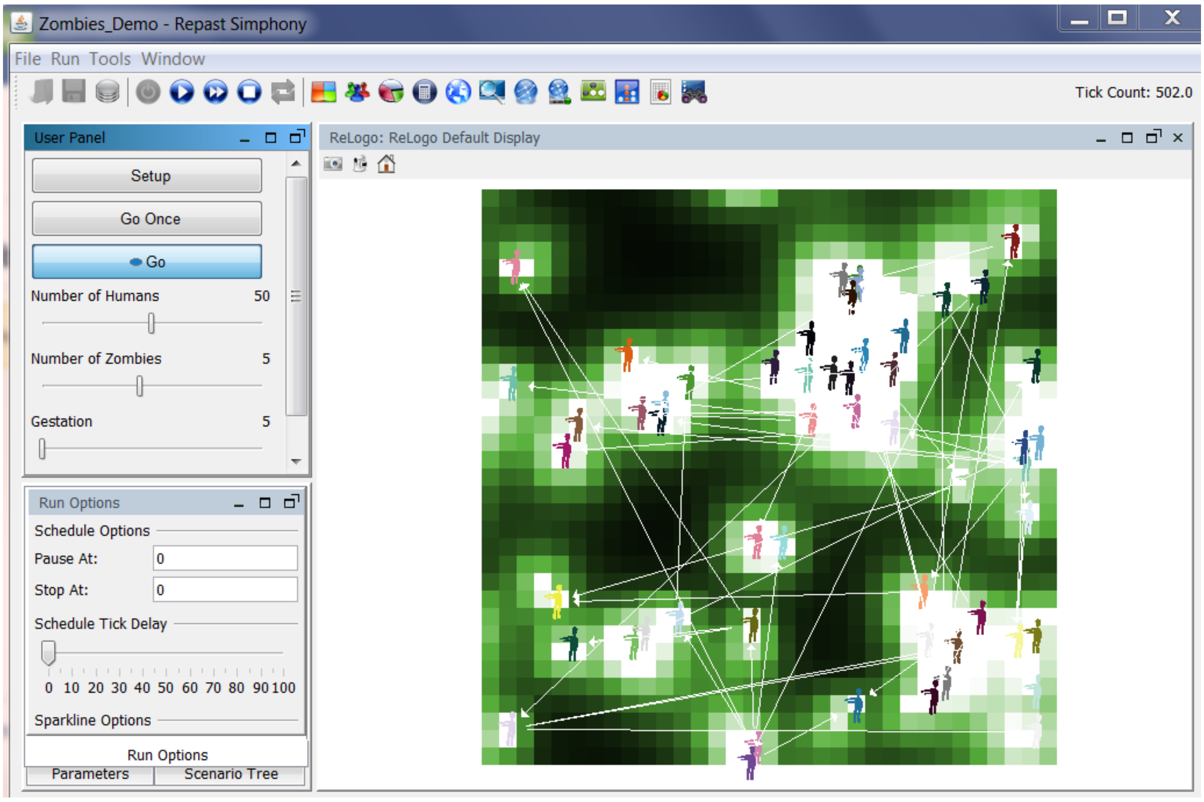

Figure 5: Repast Simphony GUI during simulation

6.2. Experiences implementing PowerTAC broker in Repast

Currently the PowerTAC sample broker is available only in Java, so in order to enable developers which are not skilled Java programmers to also take part in PowerTAC by creating their own agents using ReLogo or Groovy, it is necessary to implement the sample broker in Repast.

The core data structure in Repast is the **context**, a container that holds model components, which can be agents or other contexts who make up the population of the model. A context provides the basic infrastructure to define a population and the interactions of that population without actually providing the implementations [7]. It can be said that a context represents an abstract environment in which the agents exist at a given point in the simulation. Contexts also support **projections**, sets of relationships defined over the members of a context. Projections can be network projections, defining relationships between the members of its context, and also spatial, such as grid projection that stores a set of coordinates for each member of the context. A projection is added to a particular context and applies to all of the agents in that context, so an agent must be added to the context before it could be used in a projection [24]. Context and projections are defined in a configuration file which is called *context.xml* in every Repast project.

As mentioned earlier, Repast offers a discrete event scheduler that determines which actions will be executed at what point in the simulation. The quantum unit of time in a Repast simulation is known as a tick [22]. Ticks are used to order the execution of actions relative to each other. By clicking the *Run* button in the Repast GUI, the simulation starts and ticks are executed automatically. The user can also manually iterate through the simulation by clicking the *Step* button. However, Repast also supports more sophisticated scheduling mechanisms. There are several ways to schedule an action:

- **Directly** by calling the `schedule()` method and passing the parameters, such as the agent whose method is supposed to execute, the name of the method and, optionally, parameters that the method requires. For instance, `schedule(params,myAgent,"step")` will cause myAgent's `step()` method to execute;
- Using **annotations** – By adding `@ScheduleMethod` annotation to a method, it can be scheduled to execute at certain points during simulation. The annotation supports various parameters such as *start* which tells when the action should execute for the first time, then *interval* which says how often the action will execute, etc.
- Using **watchers** for dynamic scheduling – instead of using static scheduling times, the user can specify the trigger condition that must be met for the action to execute, for instance when the value of a certain variable crosses a defined threshold.

6.2.1. Configuring simulation parameters through Repast GUI

Apart from defining relationships between agents and their spatial arrangement, Repast also offers custom configuration of parameters used in the model during the simulation. Those parameters, their names, default values and types are defined in the *parameters.xml* file. They can be easily accessed in the code by referencing them by their name.

The idea was to enable the user to define and configure parameters specific for the PowerTAC broker before starting the simulation. Currently such parameters are located in the *broker.properties* file inside the PowerTAC sample broker project, so the user has to open and edit it manually before starting the broker. By configuring those parameters at model initialization in the Repast GUI, it would be easier for the user to add new parameters and compare the impact of different parameter values on the broker's result at the end of the competition, such as tariff rates, signup payments, bidding prices and other parameters the broker uses while running. However, various problems have occured while trying to start the broker through the Repast GUI.

PowerTAC sample broker is a Maven[16] project, meaning that it has dependencies on other external libraries defined in the *pom.xml* file. By converting the Repast project to a Maven project, adding the *pom.xml* file and updating dependencies, all of the external libraries i.e. *.jar* files are downloaded and added to the project's build and classpath. However, every Repast project has its specific Repast development libraries which cannot be modified. By adding PowerTAC-specific code to the Repast project and running the model, `ClassNotFound` and `NoClassDefFound` errors occur, as shown in Figure 6. The meaning of these errors is that Repast class loader is unable to find the classes at runtime. The classes for which the errors occur are the ones located in common.jar, a library which contains domain classes common to the PowerTAC projects, such as *Competition*, *TariffTransaction*, *BankTransaction* and so on. It seems that the

[16] Apache Maven is an open source build management tool for Java projects. (http://maven.apache.org)

Repast class loader cannot find the class files located in libraries that are a part of Maven dependencies.

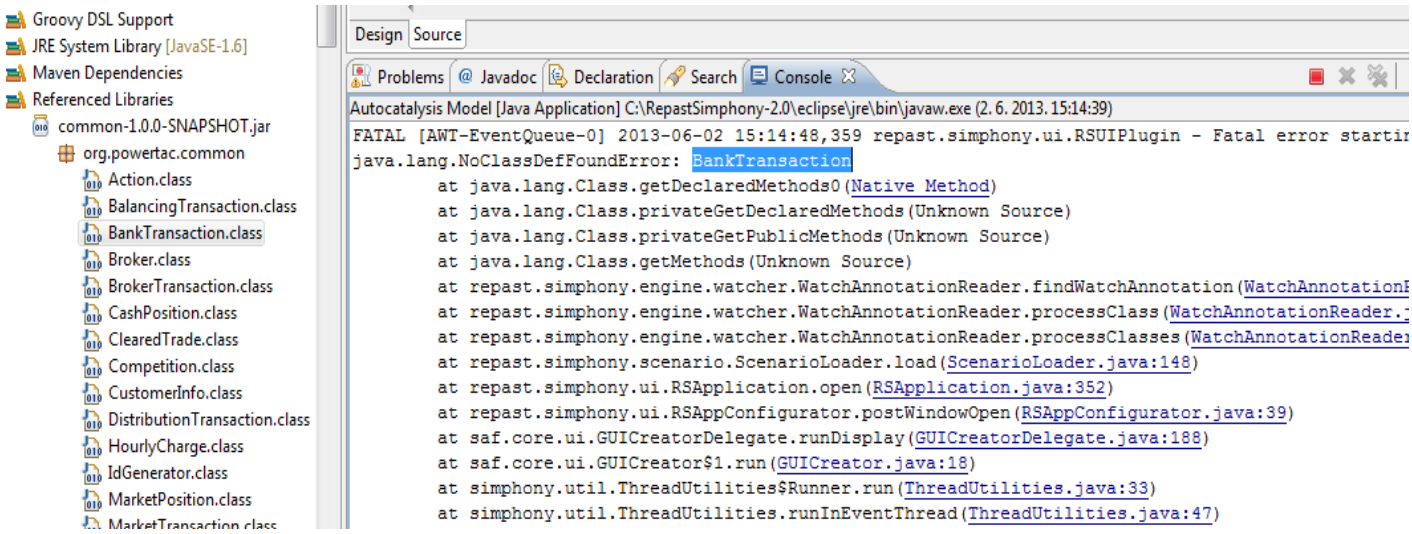

Figure 6: *NoClassDefFoundError* while running the simulation

One way to add an external library to a Repast project is by placing it inside the *lib* folder in the project and adding it to the build path. Additionaly, an external library can be added to the classpath by adding an entry containing the path to that library to the *user_path.xml* file in the project. Unfortunately, both of these approaches resulted in the same `NoClassDefFoundError`, meaning that the the class loader was unable to find those classes at runtime.

6.2.2. Controlling the simulation execution manually

Another idea was to create a model which would give users the opportunity to manually create tariffs and wholesale offers and submit them to the market by clicking a certain button in the GUI. That way users could explore and analyze the impact of different bidding strategies and tariff designs on the broker's results. However, the PowerTAC simulation is not managed by the broker but by another PowerTAC component - **the simulator**. The simulator sends the *SimStart* message at the beginning of the game, and *TimeslotUpdate* message at the beginning of every timeslot. That way the broker knows when the simulation clock is updated and synchronizes its internal clock with the competition.

At the beginning of every timeslot all of the services implementing the `Activatable` interface are being activated. For instance, the

`MarketManagerService` submits a wholesale order, while the `PortfolioManagerService` specifies and submits new tariffs or improves existing active tariffs.

In order to give the user the ability to send messages to the simulator manually, it would be necessary to change the way the simulation is managed, meaning that the broker would control the execution of the simulation instead of the simulator. However, this is not possible because the purpose of PowerTAC is that more brokers compete with each other, therefore it is clear that the simulation cannot be managed by all of them, instead it must be managed by the simulator.

There is also a big difference in how simulations in Repast work in comparison with PowerTAC. The Repast scheduler expects the scheduled method to execute and return, and repeat that at every step of the simulation, while PowerTAC broker runs continuously, receiving *TimeslotComplete* and *TimeslotUpdate* messages which signalise that the simulation clock is updated, until it receives the *SimEnd* message from the server, meaning that the simulation has ended.

6.2.3. Starting the simulation without using the GUI

Typically a Repast project is run by calling the `main()` method in the `repast.simphony.runtime.RepastMain` class with the argument that contains the path to the folder where all of the configuration files are located *(scenario.xml, context.xml, parameters.xml* etc.). After that the GUI loads and the user is able to see and configure parameters defined in the model. However, it is also possible to run the model from another Java class provided that it extends `repast.simphony.environment.AbstractRunner` class. The runner class can be used to initialize, step, run, pause, stop and reset the model, just like clicking the corresponding buttons in the GUI.

This approach basically means that the simulation is being run purely programatically and the user is not able to use the GUI to initialize the model, modify defined parameters and start the simulation. The idea was to extend the existing sample broker code with a class that would be responsible for iterating through the simulation, i.e. controlling when scheduled actions will be executed. For instance, when the broker receives a *TimeslotUpdate* message, meaning that a new timeslot has started, it could call the scheduler's `execute()` method, which would then trigger all agents' methods that were scheduled for that timeslot.

By running the broker using the PowerTAC run configuration instead of Repast default configuration, the `main` method inside `BrokerMain` class is called. However, during runtime an `AbstractMethodError` ocurrs. Apparently there is a conflict between the version of SLF4J[17] library used in Repast and the one defined in PowerTAC broker's dependencies. The error occurs in the `TcpTransport` class inside the *ActiveMq* library, which PowerTAC uses for JMS communication with the simulator. The `AbstractMethodError` can be resolved by using a newer version of the library[18]. However, the version which Repast uses cannot be updated or modified because it is defined in the Repast development libraries, which are not configurable, as shown in Figure 7. The reason this error happens is that there are two versions of the same library on the project's classpath, and Repast uses the one defined in its development libraries, instead of the newer one in Maven dependencies.

Figure 7: Build path properties

---

[17] The Simple Logging Facade for Java (SLF4J) is an abstraction for various logging frameworks (e.g. java.util.logging, logback, log4j)

[18] http://forum.springsource.org/showthread.php?74925-AbstractMethodError-NOPLogger-trace-when-starting-server

Conclusion

Agent-based modeling is a method used for simulating behavior of complex adaptive systems. Agents have the ability to make decisions, interact with each other and adapt their actions to the environment they populate.

One of the most complex systems in the world is the electricity market. Due to the ongoing processes of modernization and deregulation, various new entities have entered the market, making it more complex and competitive. In order to satisfy increasing energy demand, integrate all entities into a reliable system and provide sustainable and secure energy supply, current power grids are switching to smart grids. Once fully implemented, smart grids are expected to have a positive influence on economics, environment, as well as electricity reliability, security and efficiency.

Prior to implementing new market designs in real world, it is important to test them in a risk-free environment. There are numerous agent-based models of electricity markets. Power Trading Agent Competition is a simulation of an advanced electricity market in which agents, which act as self-interested brokers, compete with each other by attracting customers and trading on the retail and wholesale markets with the intent of earning a profit.

The goal of this thesis was to implement a PowerTAC broker using the Repast toolkit. However, due to limitations of the toolkit and incompatibility with technologies used in PowerTAC, as described in the thesis, this goal has not been achieved.

Bibliography

[1] Bar-Yam, Y., *General Features of Complex Systems*, New England Complex Systems Institute, Cambridge, MA, USA

[2] Chan, S., *Complex adaptive systems*, ESD.83 Research Seminar in Engineering Systems, Massachusetts Institute of Technology, 2001

[3] Holland, J., Studying Complex Adaptive Systems, Journal of Systems Sciences and Complexity, Vol. 19, 1-8, 2006

[4] Complex Adaptive Systems Group Wiki, wiki.cas-group.net/, accessed April 2013

[5] Gell-Mann, M., *The Quark and the Jaguar, adventures in the Simple and the Complex*, MacMillan, St Martin's Press, 1995

[6] Macal, C., *Complex Adaptive Systems*, Argonne National Laboratory, http://www.dis.anl.gov, accessed May 2013

[7] North, M., Collier, N., Ozik, J., Macal, C., *Complex adaptive systems modeling*, 1:3, Springer Open Journal, 2013

[8] Bonabeau, E*., Agent-based modeling: Methods and techniques for simulating human systems*, Proceedings of the National Academy of Sciences, Vol. 99(3). Washington, D.C. USA, National Academy of Sciences Press, 7280–7287, 2002

[9] Jennings, N., Wooldridge, M., *Intelligent agents: Theory and practice*, Knowledge Engineering Review, vol. 10, 1995

[10] Narzisi, G., *Agent-based modelling in Repast*, Courant Institute of Mathematical Sciences, 2009

[11] Weidlich, A., Veit, D., *A critical survey of agent-based wholesale electricity market models*, Energy Economics, 1728-1759, 2008

[12] Zhou, Z., Chan, V., Chow, J., *Agent-based simulation of electricity markets,* Artificial Intelligence Review, 305-342, 2009

[13] Pepermans, G., et al, *Distributed Generation: Definition, Benefits and Issues*, Energy Policy, Vol. 33, No. 6, 787-798, 2005

[14] *Distributed Generation and Emerging Technologies*, Center for Climate and Energy Solutions and the University of Texas, Energy Management and Innovation Center, October 2012

[15] Jeremi, M., *Distributed vs. centralized electricity generation: Are we witnessing a change of paradigm?,* HEC Paris, May 2005

[16] Hoff, T. et al, *Distributed Generation and Micro-Grids*, National Renewable Energy Laboratory, Pacific Energy Group, 2007

[17] StarLogo homepage, http://education.mit.edu/starlogo, accessed May 2013

[18] NetLogo homepage, http://ccl.northwestern.edu/netlogo, accessed May 2013

[19] Multi-Agent Simulation Toolkit (MASON) website, http://cs.gmu.edu/~eclab/projects/mason, accessed May 2013

[20] Luke, S., *Multiagent Simulation and the MASON Library*, George Mason University, Department of Computer Science, May 2013

[21] Minar, N., Burkhart, R., Langton, C., *The Swarm Simulation System: A toolkit for building multi-agent simulations*, 1996, http://alumni.media.mit.edu/~nelson/research/swarm, accessed May 2013

[22] North, M., Collier, N., Vos, J., *Experiences Creating Three Implementations of the Repast Agent Modeling Toolkit*, ACM Transactions on Modeling and Computer Simulation, Vol. 16, No. 1, January 2006

[23] North, M. et al, *The Repast Simphony Runtime System*, Proceedings of the Conference on Generative Social Processes, Models, and Mechanisms, 2005

[24] Repast Simphony homepage, http://repast.sourceforge.net, accessed April 2013

[25] Hamilton, B., *Understanding the Smart Grid Benefits*, Smart Grid Implementation Strategy, National Energy Technology Laboratory, United States Department of Energy, 2010

[26] Fang, X., Misra, S., Xue, G., Yang, D., *Smart grid: The New and Improved Power Grid*, Communications Surveys & Tutorials, IEEE, vol.14, 2012

[27] Hicks, C., *Smart Grid: Where We Are Today and What the Future Holds*, ERB Renewable Energy Scholar, ERB Institute, University of Michigan, 2012

[28] Frye, W., *Smart Grid - Transforming the Electricity System to Meet Future Demand and Reduce Greenhouse Gas Emissions*, Cisco Internet Business Solutions Group, 2008

[29] North, M. et al, *E-Laboratories: Agent-based modeling of Electricity Markets,* Center for Energy, Environmental, and Economic Systems Analysis (CEEESA), Argonne National Laboratory, 2002

[30] Conzelmann, G. et al, *Simulating Strategic Market Behavior Using an Agent-Based Modeling Approach: Results of a Power Market Analysis for the Midwestern United States*, Proceedings of the 6th IAEE European Energy Conference on Modeling in Energy Economics and Policy, 2004

[31] Batten, D., Grozev, G., *NEMSIM: Finding Ways to Reduce Greenhouse Gas Emissions Using Multi-Agent Electricity Modelling*, Complex Science for a Complex World, ANU Press, 2006

[32] Deconinck, G., Maenhoudt, M., *Agent-Based Modelling as a Tool for Testing Electric Power Market Designs*, University of Leuven, Department of Electrical Engineering, 2010

[33] Ventosa, M., et al, *Electricity market modeling trends*, Energy Policy, 33:897-913, 2005

[34] Block, C., Collins, J., Ketter W., Weinhardt, C., *A Multi-Agent Energy Trading Competition*, Technical Report, no. ERS–2009-054-LIS, Erasmus Research Institute of Management, Erasmus University, 2009

[35] Weber, A. et al, *OPTIMATE – a Modeling Breakthrough for Market Design Analysis to Test Massive Intermittent Generation Integration in Markets*, Proceedings of the 9th International Conference on the European Energy Market, 2012

[36] Tesfatsion, L., et al, *The AMES Wholesale Power Market Test Bed: A Computational Laboratory for Research, Teaching and Training*, Proceedings of the IEEE Power and Energy Society General Meeting, 2009

[37] Ketter, W., Collins, J., Reddy, P., de Weerdt, C., *The 2013 Power Trading Agent Competition*, ERS-2013-006-LIS, 2013

[38] Repast Development Team, *Repast Simphony Frequently Asked Questions,* http://repast.sourceforge.net/docs/RepastSimphonyFAQ.pdf, accessed April 2013

Sažetak

Svrha ovog diplomskog rada bila je implementirati inteligentnog programskog agenta koristeći alat Repast. Motivacija je proizašla iz potrebe da se korisnicima, koji su upoznati s modeliranjem koristeći alat Repast, omogući implementacija različitih agenata i njihova integracija u jednog agenta koji će sudjelovati u natjecanju PowerTAC. Opisani su složeni prilagodljivi sustavi i njihove karakteristike, kao i metode za njihovo modeliranje temeljene na agentima. Poseban naglasak stavljen je na na modeliranje tržišta električne energije. Energetske mreže trenutno prolaze kroz proces modernizacije i deregulacije te evoluiraju u napredne mreže. Nadalje, opisano je natjecanje PowerTAC. Naposljetku, opisani su razlozi zbog kojih nije bilo moguće postići zadani cilj, kao i problemi koji su se pojavili prilikom pokušaja implementacije agenta koristeći alat Repast.

Ključne riječi: inteligentni programski agenti, natjecanje Power Trading Agent Competition, složeni prilagodljivi sustavi, napredne energetske mreže, modeliranje temeljeno na agentima, alat Repast

Summary

The purpose of this thesis was to implement an intelligent agent using the Repast toolkit. The motivation came from the need to enable users familiar with modeling in Repast toolkit to implement their agents and integrate them into a broker that participates in the PowerTAC competition. An overview of complex adaptive systems and agent-based modeling used for their simulation is provided. A special special emphasis was put on modeling electric power markets. Electrical grids are currently going through processes of modernization and deregulation and evolving into smart grids. Furthermore, a description of the PowerTAC competition is given. Finally, the reasons why the goal of the thesis could not be achieved as well as the problems encountered while implementing the broker in Repast are described.

Keywords: intelligent agents, Power Trading Agent Competition, complex adaptive systems, smart grids, agent-based modeling, Repast toolkit