CHAPTER II
LITERATURE REVIEW

2.1. Theory of Computer Simulation

Computer simulations are used widely across many real-world subjects such as automotive, social sciences, biology, physics, and many more. Computer simulation itself has many varieties of implementations and models, used to determine the most suitable system for simulating a certain phenomenon.

2.1.1. Definition of a computer simulation

Computer simulation has a variety of definition. There are two points of view regarding the definition of computer simulation (Winsberg, 2013).

1. Narrow definition

A computer simulation is a certain program which runs from a computer using a step-by-step method to determine behavior of a certain model. The models are usually real-world system which are hard or take lots of time if simulated in reality. A computer simulation runs by using a certain algorithm which requires input or variables on the system state at time *t*. The simulation will process all inputs to produce a result, which will be used to produce the next result at $t+1$. The simulation system also produce a visualization of the evolution of the system depending on the result of the simulation, whether it's in form of graphics, charts, 2D or 3D visualization (Winsberg, 2013).

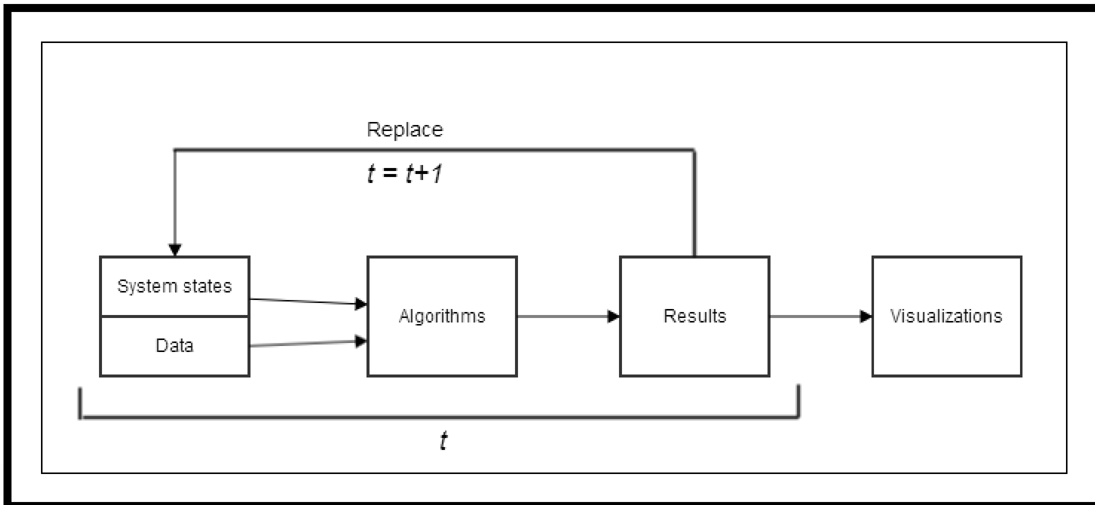

Figure 2.1. Computer simulation in narrow definition

A computer simulation is used to perform a calculation which contains a discrete equation that represents a changing value over time. This equation is simulated in the program's environment because there are limited analytical methods to analyze the equation. The simulation continues to change the value of the equation over time so that it will provide an approximate result in a certain degree of limitations.

2. Broad Definition

Computer simulation in a broader definition can be seen as a study of a certain system. The study of the system includes its model, chosen algorithm suitable for the system, implementation, and visualization of the data. Unlike simple computations, a computer simulation's result is not definite and is being affected by many factors, thus making the reliability of the result is hard to decide (Winsberg, 2013).

Nowadays, engineers, programmers, and analysts use computer simulation to model a certain real-world phenomenon and "what-if" scenarios. The systems used for simulation are systems that are costly and hard to implement in real world (Wolski, 2012). Computer simulations generate results by solving the algorithm with certain data over time. The “What-If” scenarios are controlled variables used to give different scenarios of the simulated system, giving the user the freedom in choosing the best solution in the system with different scenarios.

2.1.2. The use of Computer Simulation

Computer simulation is widely used in many areas, especially in fields which are complicated to analyze, hard and costly to implement, and takes a long time to achieve the desired result. With an increasing power and efficiency of computing power and performance, many complex and time-consuming real-world systems are simulated in computer environment. The simulations itself are able to provide a predictive result which can be analyzed by statistics within a short period of time.

Computer simulation is suitable for certain situations in real-world which can be seen in the table below.

Table 2.1. Situations for computer simulation (McHaney, 2009)

General Situations	**Examples**
Real system does not exist yet. Building the system itself requires high cost and time-consuming	Aircraft, Production system, Nuclear Reactor
System is impossible to build	National economy, biological system
Experiments on the system is expensive and hazardous	Military units, transportation systems, airport baggage handling system
Forecasting, time-consuming system	Population growth, Forest fire spread, Pandemic flu spread
Mathematical modeling	Stochastic problems, Non-linear differential equations

Computer simulations produce many benefits in experiments which are time-consuming, expensive, and hard to implement. Its result also affects many decision making processes done in certain areas. Besides the advantage of time and cost, several advantages given by computer simulations are (McHaney, 2009).

1. No disruption to the existing systems

   Because the simulated system is done in the virtual environment in the computer, we can compare the result between the simulation and the real world system without any impact on the real system.
2. Simulation can be tested

   The concepts and state of the simulation can be tested using different sets of data and scenarios.
3. Faster analysis

   With the simulation done in the computer environment, engineers are capable to speed up the process in a certain virtual time, giving a similar, predictive result rather than the real world system.

2.1.3. Limitations of Computer Simulation

Although speed and cost are the advantage of computer simulation, it still has some limitations (McHaney, 2009).

1. Approximate results

   The result of a computer simulation often relies in the use of random number generators. This produces some uncertainty in the simulation itself and requires further statistical skills and analysis.
2. Difficult to validate

   Especially when the system itself does not exist, the validation of the simulation relies heavily on experts and intuition. Elements such as human behavior are very difficult to model and predict.

2.2. Multi-Agent System

2.2.1. Definition of Multi-Agent System

Multi-agent system is one of the types of computer simulation which consists of multiple agents scattered in a virtual environment. These agents have criteria such as (Davidsson, 2013),

1. Communication language
   Agents in a simulated environment are able to communicate with other agents in a certain form of messaging and are implemented through program code.
2. Autonomous
   Agents are not controlled by the users. They are completely autonomous objects which have the ability to process information based on their knowledge. This knowledge is a set of data and state and the ability to process the information is implemented in a certain Artificial Intelligence (AI) algorithm.
3. Mobility
   Agents range between the ones that are stationary or the ones which can travel through a spatial configuration in the simulated environment.
4. Adaptive
   An agent in a multi-agent simulation system is a learning agent, which could percept the environment around them.

Multi-agent oriented approach in building complex systems is developed from object-oriented principles. Agents in a multi-agent environment are defined as objects with additional features, such as the state and behavior of the objects (Batista, 2011). The difference between agents and objects is their autonomous behavior. Agents have a certain degree of freedom in changing their behavior and pattern which makes it an active objects in a simulated environment, whereas an object is a passive object.

Multi-agent systems are not the opposite of the object-oriented methods. A multi-agent system use both agents and object-oriented approach in the programming implementation. They serve as active and passive objects in a simulated environment. For example, a multi-agent system of a traffic system consists of vehicle and intersection agents, but also consists of passive objects such as roads and environment (Bazghandi, 2012). Passive objects can also serve as the limitations of the system so that the simulation can focus on the analysis of the agent's behavior.

According to Shoham, which is also citated by Batista (2011) in his paper titled "Principles of Agent-Oriented Programming", Agents and its difference from passive objects are described with the table below (Batista, 2011).

Table 2.2. Relation between OOP and AOP (Batista, 2011, Shoham, 1993)

Framework	OOP	AOP
Basic Unit	Object	Agent
States of basic unit	Unconstrained	Beliefs, commitment, capabilities, choices
Process of computation	Message passing and response method	Message passing and response method
Constraints of methods	None	Consistency, honesty

2.2.2. Multi-Agent System in Social Sciences

Computer simulation, especially multi-agent system has been used to describe and model social phenomenon which happens in a certain environment or field in social sciences. Agent-based modeling is a way for exploring the social behaviors between the agents, environment, and the social network among them (weADAPT, 2011). Social science itself is a discipline which study human behavior in cultural and social aspects (Britannica, 2013). One of the methodological approaches which could be implemented in a computer simulation is the approach of statistical modeling. Each social actor such as people, groups, parts of environment, etc. are represented as agents and could interact with each other. The agents themselves have their own characteristic and a set of inter-related rules. Statistical analysis in the simulation takes place in the interactions between the agents themselves which could result to a complex simulation result that is challenging to validate (Edmonds, 2011).

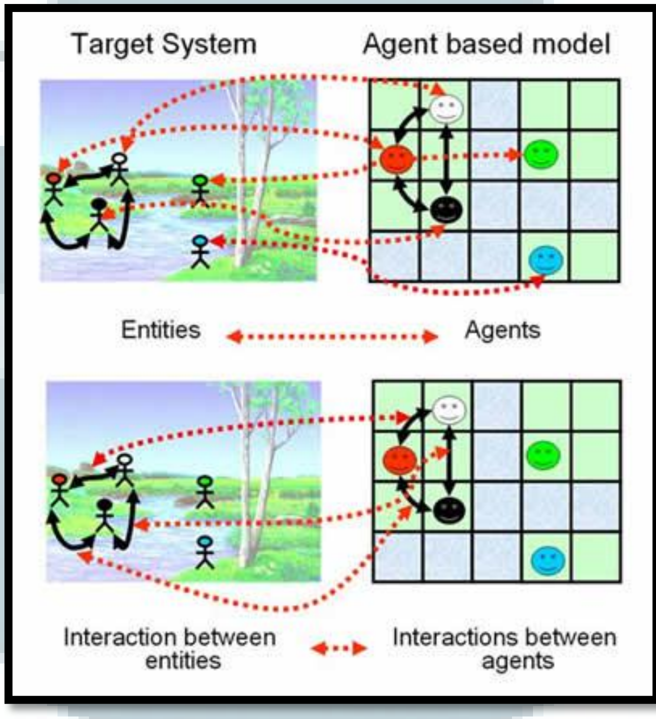

Figure 2.2. Simple overview of multi-agent simulation in social science (Edmonds, 2011).

The complexity of simulation for social science comes from the agents' itself. Even when agents are given simple rules, the behavior can result to be very complex to analyze. Each linear relationship between variables of the agents in the simulated environment could result in non-linear systems which is difficult to define the characteristics of the system (Gilbert & Troitzsch, 2005). The validation of whether a simulation best describe a social phenomenon depends on three things (Marks, 2012),

1. Historical data. Historical data can come from various literature studies or any related data.

2. The ability to transform the data into a simplified model or a set of rules in the simulation.
3. The skill used to derive implications and result and examine it with the original data.

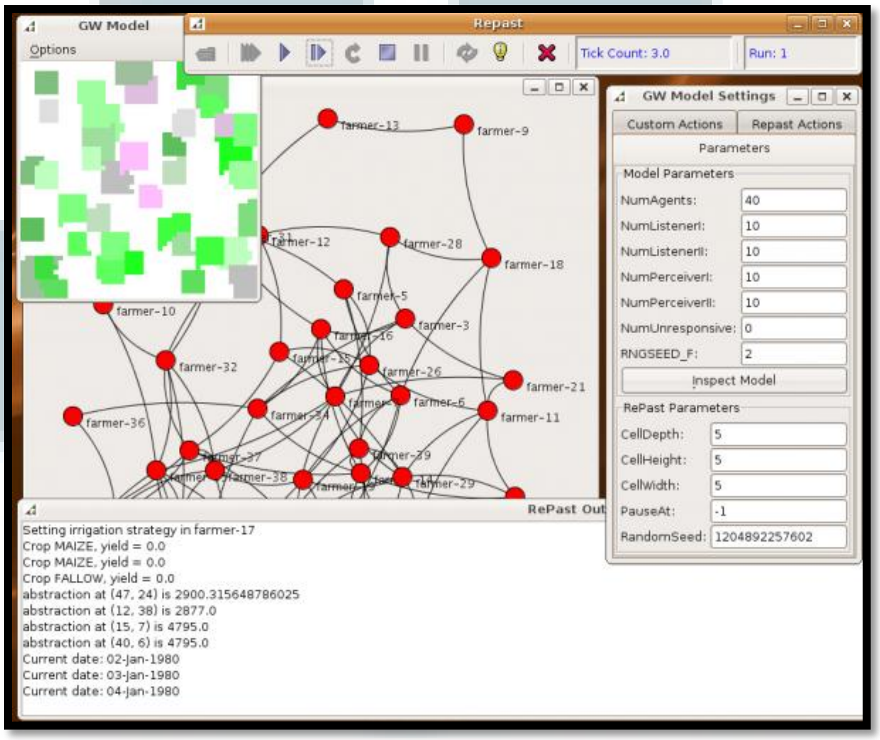

Figure 2.3. Example of agent-based simulation of rice crop system (weADAPT, 2011)

2.2.3. Implementing Rules and Knowledge in Multi-Agent System

In building multi-agent simulation for social science, social theories need to be applied to the system itself. Social theories or social norms are the models that give agents a basic form of rules to do actions in the simulated environment and also used to reduce complexity of the system and conflict (Hollander & Wu, 2011, Hexmoor, 2006). Social norms integrate social and individual factors to each agent. The norms are also used for coordination, cooperation between agents and as the behavior constraint for the agents (Hollander & Wu, 2011).

Social theories or social norms also define how social agents learn and perceive the surrounding environment. One of the proposed models for the agent's learning and decision making strategy model is to choose the action based on the maximum utility value (Tudor, Andrei & Mihai, 2011). The process of calculating the value can be described by the following equations.

$$a = argmax_a utility_{role}(a_i) \text{..........................(1)}$$

where $(argmax)_a$ contains a set of possible actions the agent can do based on its role, which is defined by its knowledge, and is represented by variable $utility_{role}$. The maximum utility value can be calculated based on.

1. Interaction between agents
2. The knowledge of the agent
3. Additional randomly generated value

From the equation (1), the agent is able to determine which actions it should take in the simulated environment. Behind the interaction of each agents, the learning phase of the agent also happens. This occurs by updating the utility role which can be represented with the equation below.

$$utility_{role}^{t+1}(a_i) = utility_{role}^{t}(a_i) + payoff(a_i) \text{..................(2)}$$

Equation (2) states that $utility_{role}^{t+1}$ value is updated by a certain payoff function that the agent gives as the result of the interaction between agents. $payoff(a_i)$ represents the utility value of the action taken.

In other cases, knowledge plays an important role combined with the rules associated. Knowledge is the agent's basic set of data which the agent relies with. Agent's knowledge is closely related to all aspect of the agent, such as decision-making process, translating the rules, and defining interaction between other agents (Teahan, 2010). When the agent is acting, it uses knowledge bases to choose its actions and therefore, updating the knowledge bases itself. The process runs in an infinite loop until the agent's live time is depleted. The term "long-term memory" is used to describe the knowledge to decide the next acts (Poole & Mackworth, 2010).

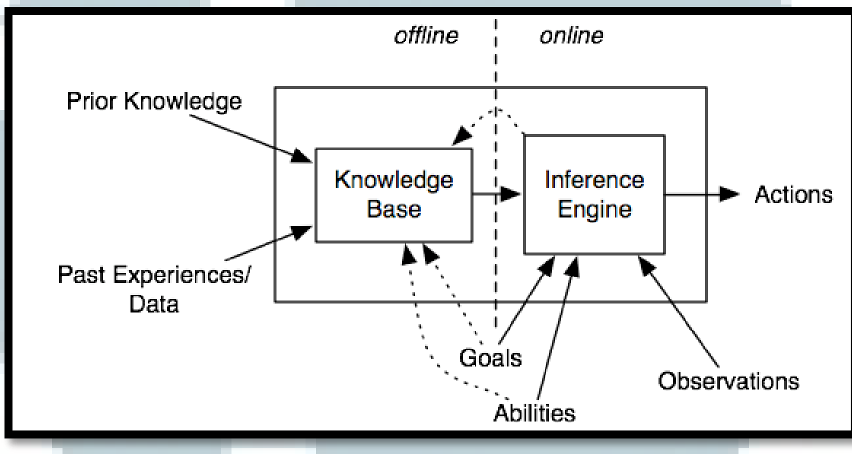

Figure 2.4. Decomposition of Knowledge-based agent (Poole & Mackworth, 2010).

Implementation of the knowledge base on agents can be done in many ways. The simplest way is the "if-then" rules, but it only serves as the basic operation in the simulation. The inference engine is needed to describe the knowledge and rules implemented into "if-then" operations (Sanjeev & Singh, 2012). A string-based rule system can be used in implementing knowledge-based agents. Each rules and knowledge are defined in a string containing logical, defined operations, which will be indexed and parsed by the inference engine and become a simple "if-then" rule (Leondes, 2010).

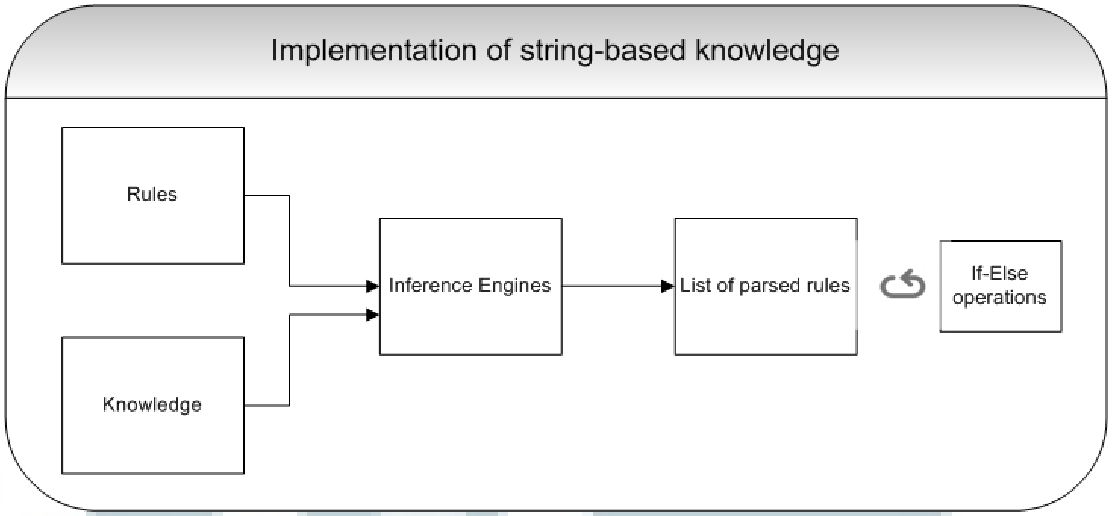

Figure 2.5. Implementation of string-based knowledge base

2.2.4. Validating Multi-Agent Simulation

Validation of a multi-agent simulation is required, so that the simulation itself is able to validly represent a certain world phenomenon. Verifying and validating a complex simulation such as social science becomes a difficult task because of the amount of independent intelligence agent, large number of concurrent interactions, and learning pace of the agents (Pullum & Cui, 2012). Validating a mathematical or scientific simulation model is easier because it is a structured system with clear rules (Xiang, 2005). In validating and ensuring that the simulation could represent the real-world phenomenon as accurate as possible, there are certain factors which should be considered (Ali, 2006),

1. Verification and validation provide an evidence to support the simulation models which is capable of simulating the real phenomenon.
2. Difficulty of the validation depends on the complexity of the system.

3. There is no perfect and accurate simulation in complex social systems. It can only simulate an approximation of the real-world system. Simulation itself is the abstraction and simplification of the reality.
4. Simulation is developed for a certain objective.

Although validation and verification of a multi-agent system is crucial, there are still no definite processes or models designed to validate a certain agent-based simulation. Agent-based systems are known to heavily dependent on informal, subjective, qualitative or no validation at all (Bharathy & Silverman, 2010). This is because agent-based simulation, especially simulating real world social phenomenon, are significantly abstract, which means they do not produce the exact same output as the real world phenomenon. In most cases, validation of such models is carried out in abstract levels such as patterns and macro-level correspondence tests (Bharathy & Silverman, 2010).

There are several verification techniques which are commonly used to verify and validate certain simulation systems (Mitre, 2012).

1. Comparison to other models. This technique compares the result of the simulation output to other valid models. This technique often compares a simple test case between simulation tools.
2. Face Validation. This technique is performed with expert analysts of the phenomenon simulated by asking whether the system or the model and its behavior are reasonable. This technique also verifies whether the logic implemented and input-output relationships are correct and acceptable (Mitre, 2012). Although limited to subjectivity of the experts and lack of statistical analysis, this technique provides a very descriptive analysis of the models itself (Bharathy & Silverman, 2010).
3. Historical Data Validation. This technique compares the output of the simulation to historical data if exists. This is used to test whether the simulation behaves as accurate as the real system does.

4. Parameter Variability. This technique is conducted by varies the parameter of the input by a certain range. The relation changes in simulation models should also occurs in real world system.
5. Cause-Effect Graphing. This technique represents the cause and effects identified in the systems themselves. Cause represents the input of the simulation and effect represents the output of the simulation. CEG is constructed in logic network of nodes. The networks are then transformed to a decision table, which will be converted to use cases and test cases (Mogyorodi, 2010; Jagli 2012).

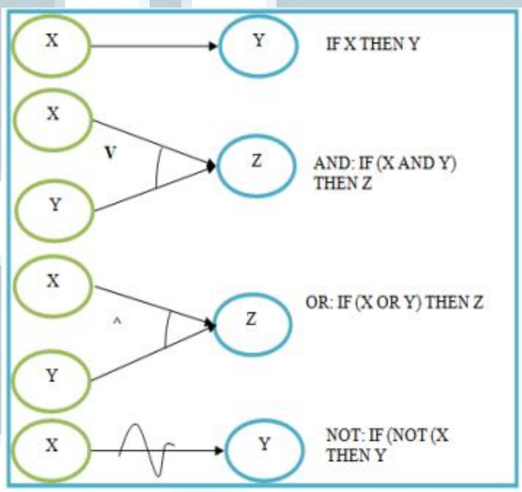

Figure 2.6. Basic Notation of Cause-Effect Graphing (Jagli, 2012)

2.3. Artificial Intelligence in Multi-Agent System

As described above, in multi-agent social simulation, the agents are dependent entities which are autonomous and able to decide their course of actions from a list of possible actions it could do. The agents themselves should be able to describe real world phenomenon as realistic as possible. With the requirements above, there is a need of a certain Artificial Intelligence (AI) method or algorithm which is suitable for the agents.

2.3.1. Agent's Behavior

When talking about AI in simulation environment, the behavior of the agents is closely related to the field of AI. Behavior is the way an agent acts given a certain situation and a set of knowledge. The actions and reactions of the agent depend on the behavior itself (Teahan, 2010).

From its behavior, agents may differ into many categories. "Reactive" agents are passive agents which could react when interacting with other agents. In social simulation, "Cognitive" agents are intelligent agents which represents the active entity of the real world phenomenon. Differences between the two types of agents are shown below.

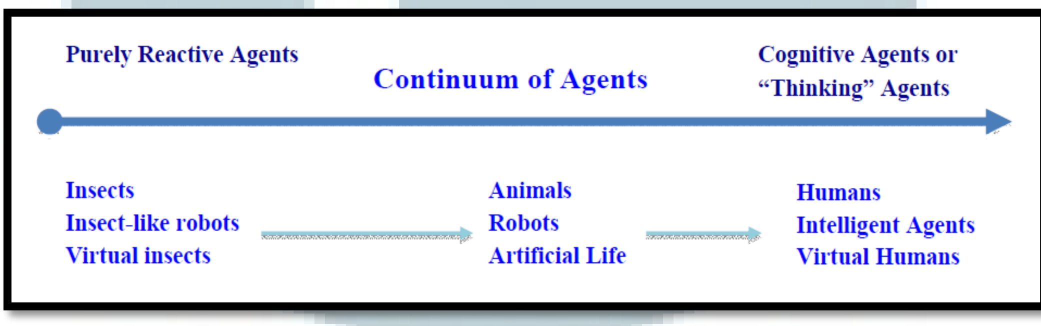

Figure 2.7. Continuum of Agents (Teahan, 2010)

AI is used mainly to simulate the behavior of the agents. Intelligence is hereby defined as the capability of the agent to perceive and act on its environment in a way that maximizes its chances of success (Briegel & De Las Cuevas, 2012). The result of the interactions between agents and environment are based from the AI methods and algorithms used.

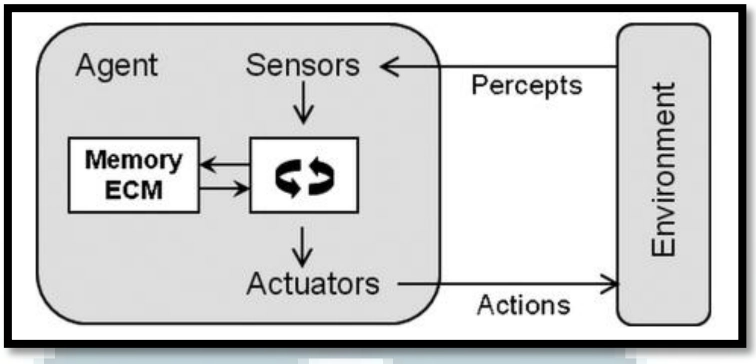

Figure 2.8. Model of an Intelligent Agent (Briegel & De Las Cuevas, 2012)

Behavior models are representations and implementations of the AI algorithm used for simulating the behavior. For example, NetLogo Model Library has a flocking behavior which mimics the flocking behavior of the birds. It implements three rules: Alignment, Separation, and Cohesion to each agent (Wilensky, 1998).

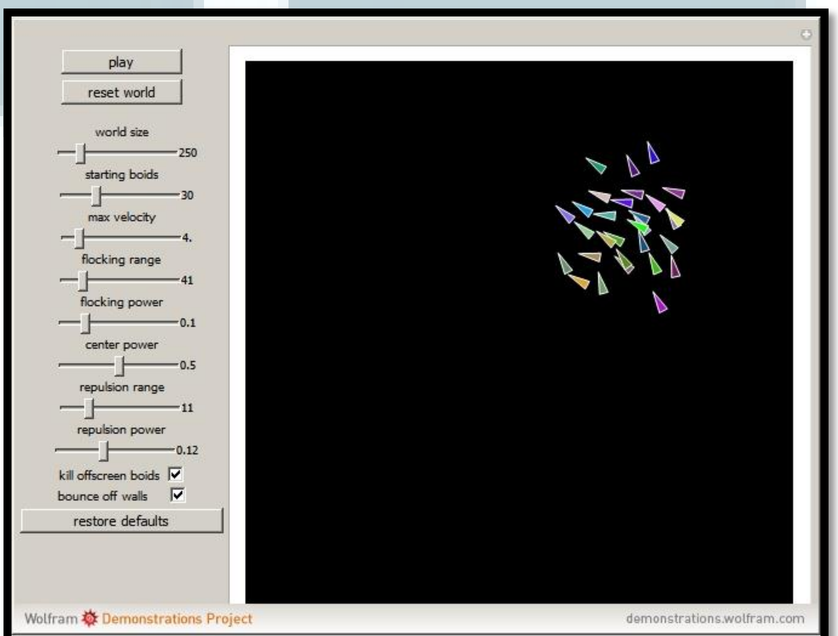

Figure 2.9. Wolfram demonstration of flocking behavior (Wang, 2012)

2.3.2. Decision Making

For autonomous entities such as intelligent agents, decision making plays an important part of the entities. Agents should be able to determine the next possible actions according to its internal and external information. It can also be said that decision making system is the “mind” of the agents themselves (Jaafar & McKenzie, 2011). There are multiple conflicting objectives with certain constraints that are involved in decision-making process. The action done from the decision-making process should serve the real purpose of the agents with the given internal state and its perception ability (Jaafar & McKenzie, 2011).

In decision making process of autonomous agents, agents are viewed as robots that are as realistic as possible into what humans do every day. Rens (2010) uses the term “automated decision-making” to explain the in-depth process of the agents making decision.

1. The agent is an individual system with available options.
2. The actions can be controlled variable parameters that can be tuned.
3. The output of the systems is called *behavior*.
4. The agent should be able to decide which actions it should take to achieve expected behavior.

The theory of decision-making process in autonomous agents is able to be modeled mathematically. Suppose an agent in a simulated environment receive information about certain attributes from the environment. Based on its knowledge base and combining the attributes, the agent could come up with a real number containing its preference (Rens, 2010). Combined with probability theory, the agent could determine the actions based on its preference.

In mathematical model, the list of utilities or attributes that the agent percept in its current state are represented by X_a. Function used to derive the number from the utilities is $U(X)$. Consider two sets of action $U(X_a) = 0.7$ and $U(X_b) = 0.3$ which the agents have decided. By the decision theory, the agent will choose $X_a$which gives the best utility value. After doing the actions, the utility values change based on the change occurs in its knowledge base and environmental changes, thus changing the agent's next course of actions. Rens (2010) formulates the decision theory as a combination of utilities and probability theory.

$$Decision\ theory = prob.theory + utility\ theory \qquad (3)$$

AI also plays a role in decision making process. AI algorithm in multi-agent system decides the best possible actions the agents can take. With the variation of real-world phenomenon, there is a need for a certain AI that could satisfies the realistic decision making process of the agents. To achieve this, the simulation requires a decision making algorithm which is flexible enough to calculate varying amounts of information (Kaufman, 2010). The aim of the AI algorithm in decision making process is to make the best rational choice within their constraints, list of preferences, and the input data (Ilieva, 2011).

Decision making process of agents in multi-agent system differs by the type of the agents.

1. Reactive agents make decision by processing and calculating its internal behavior with its interaction data and with environments or other agents. The model of decision making process can be seen by the figure below.

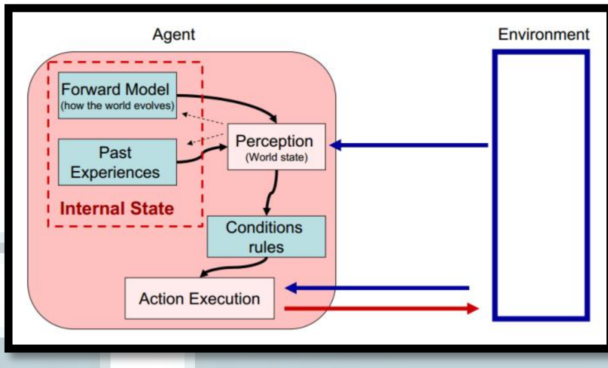

Figure 2.10. Reactive agents' decision making scheme (Oubatti, 2012)

2. Proactive agents are agents which are given task in the simulated environment. The action and decision making itself are influenced by its goal, internal states of the agent, and the perception.

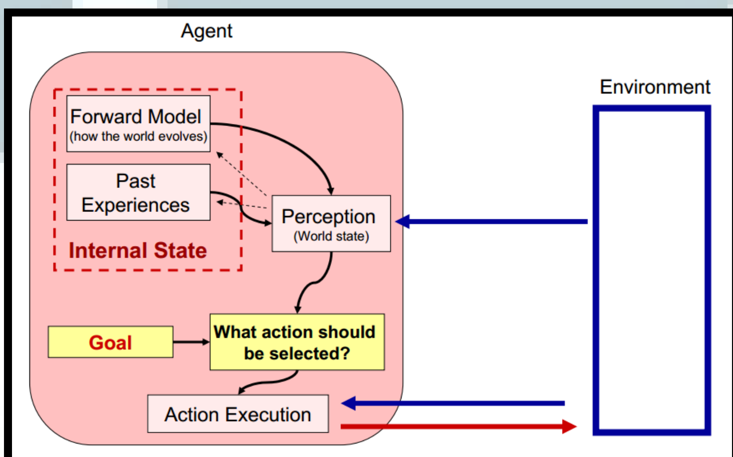

Figure 2.11. Proactive agents' decision making scheme (Oubatti, 2012)

In implementing decision making for smart and autonomous agents, there many approaches that can be taken, depending on the phenomenon simulated. With every approach, there are also tradeoffs between them. Such approaches introduced in this field are:

1. Belief, Desire, Intention (BDI) Model

This model is one of the popular approaches in developing multi-agent system. BDI system comes from a fact that rational agents should not pursue goals that is impossible to achieve (Shapiro, 2012). BDI models are developed from a traditional plan-based approach. Plan-based agents are agents which generate their own set of actions to reach its goal. When the plan becomes invalid, the agents must remake the plan from zero. Invalid plans can come from the changes occurred in simulation environment (Rens, 2010). BDI agents take different approach in practical reasoning. BDI agents can adapt to changes occurred in environment by focusing at the most appropriate goals, and can also makes rational decisions if current plans become invalid.

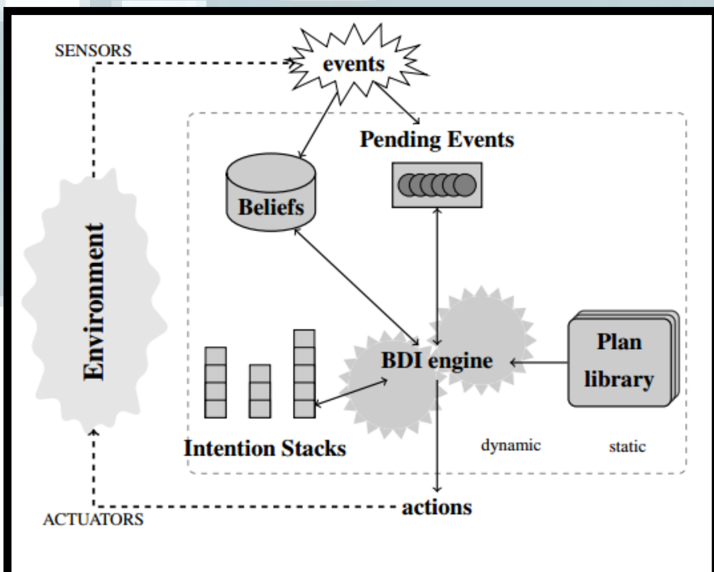

Figure 2.12. BDI-based agent's architecture (Shapiro, 2012)

Typically, a BDI-based agent contains a set of parameters that represents belief, desire, and intentions of the agent (Guerra-Hernandez, 2005).

1) Beliefs. Represents the information about the world in first-order logics. Beliefs can be derived from the agent itself, or the environment.

2) Desires. Desires are goals which the agents are created to. It is consistent among agents and actions are logically based on it.
3) Event Queue. The perceptions and historical actions of the agent are stored in a queue. Event queue is also included in agent's reasoning.
4) Plan. Plan is a set of predefined actions that the agent can take. Each plan can be triggered by a certain reaction from the environment. A Plan can be triggered if it satisfies the requirements. Suppose a "Retreat Plan" in a battlefield simulation can be triggered if a number of enemies are larger and they have more modern artilleries. The "Retreat Plan" itself contains a set of actions including picking arms, running, and hiding.
5) Intentions. Intentions are the actions that the agent has committed to carry on.

2. Markov Decision Process (MDP)

MDP is one of the decision-making methods which makes use of the world's state and rewards given in the interaction between the agents and the environment. MDP is used for sequential decision-making process to choose actions from a large, infinite list of actions where the result of the actions affects the world in a long term or short term (Hasselt, 2011).

MDP is also used to determine the best possible actions with tradeoffs between immediate rewards and long-term goals, which is similar to what people do in real life every day (Cassandra, 2009). There are four components of an MDP model.

1) States are the current parameter and variables of the world currently have. Any actions done with agents affect or change the state of the world.
2) Actions are the set of possible action the agents can do. MDP serves as a purpose to choose the best action.

3) Transitions are the way the action changes the current state of the world. The effects itself can be probabilistic, which means an action done in state n can result in $n+1$ or $n+2$ and so on.
4) Rewards are the value passed in transitions that determine the actions' cost. It is used to automate the decision making process.

Together, MDP forms an n-tuple $\{ S, A, T, R, \gamma \}$. S and A represent the set of states and actions in the current world. T is a transition function which changes the state S based on action A taken. R represents rewards obtained by the agents for doing action A in state S. γ is a discount factor ranging from 0 to 1 representing tradeoffs between long and short term goal (Ross, 2011).

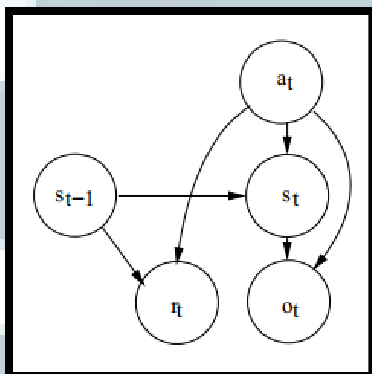

Figure 2.13. MDP model (Doshi-Velez, 2010).

2.3.3. Pathfinding

Pathfinding is a variety of methods used to determine walkable paths between one point to another in an environment. Pathfinding in multi-agent system depends on the real world phenomenon being simulated in it. For a social simulation, pathfinding becomes very important, especially when the agents are proactive and mobile agents which can move freely in a virtual environment.

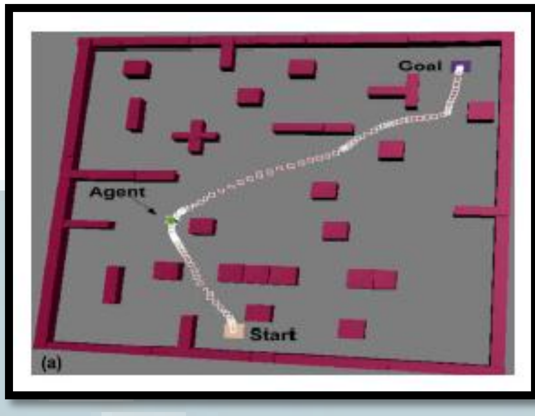

Figure 2.14. Autonomous agent navigation (Jaafar & McKenzie, 2011).

In a crowd simulation developed by Foudil, Noureddine, Sanza, and Duthen (2009), pathfinding is used to determine the movements of each agents in a crowd. It is closely related with collision avoidance where agent's movement involves motion planning and should be able to control itself (Foudil, 2009). Pathfinding in multi-agent system involving social phenomenon is implemented in the mobile agents where geographical factors aren't the only factor. Agent's knowledge base, personality traits, and target also affect the pathfinding behavior (Munchow, 2012).

When discussing about pathfinding algorithms, A* is the most popular choice for pathfinding algorithm (Patel, 2013). A* pathfinding algorithm is one of graph-searching algorithm which utilizes Dijkstra's shortest path. It is usable for huge maps because it searches the best possible and low-cost nodes in the surrounding area (Patel, 2013). A* works by maintaining a pair of lists. The first list contains a series of nodes that haven't been traveled yet and another one contains a series of explored nodes. It searches surrounding nodes for available path with the lowest cost that leads to the goal and using heuristic functions to determine the next steps (Glass, 2013). The pseudo code for A* algorithm is as follow:

```
// A*
initialize the open list
initialize the closed list
put the starting node on the open list (you can leave its f at zero)

while the open list is not empty
    find the node with the least f on the open list, call it "q"
    pop q off the open list
    generate q's 8 successors and set their parents to q
    for each successor
        if successor is the goal, stop the search
        successor.g = q.g + distance between successor and q
        successor.h = distance from goal to successor
        successor.f = successor.g + successor.h

        if a node with the same position as successor is in the OPEN list \
            which has a lower f than successor, skip this successor
        if a node with the same position as successor is in the CLOSED list \
            which has a lower f than successor, skip this successor
        otherwise, add the node to the open list
    end
    push q on the closed list
end
```

Figure 2.15. A* pathfinding pseudo code (Eranki, 2002)

A* pathfinding uses heuristic function to determine the minimum cost of the next step in finding path. The path with the cost higher than the lowest cost will be discarded. There are several heuristic functions that can be used.

1) Euclidean Distance.

Euclidean distance is a distance between two points which is measured by a straight line. It is simply calculated by using Pythagorean formula.

$$D(p,q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \quad \text{.........................(4)}$$

$D(p,q)$ represents the distance between point p and q. The distance is calculated by the sum of quadratic value of p and q in dimension i.

2) Manhattan Distance

Manhattan Distance is calculated by the sum of the difference between two nodes' x and y position. The formula used in Manhattan distance is as follow:

$$D(n_i, n_j) = \left(abs(x_i - x_j) + abs\,(y_i - y_j)\right) * d \quad \text{...........(5)}$$

$D(n_i, n_j)$ is the Manhattan distance between node n_i and n_j. It is calculated by the difference of x-position $abs(x_i - x_j)$ and y-position $abs\ (y_i - y_j)$, then multiplied by the d. In most cases, the value of d is set to 1.

2.3.4. Fuzzy Logic

Fuzzy Logic is one of the AI algorithm based on uncertainty of the situation. In a classic logic, an assertion is either true or false, zero or one. Fuzzy Logic extends the value of those logics by allowing certain value range between true and false, zero and one (Jantzen, 2008). Quoting from Lotfi A. Zadeh, the inventor of Fuzzy Logic, Fuzzy logic is a way for computing with words instead of a simple “if-then” rule.

The real world example which could explain the use of fuzzy logic is humidity. Humans are not only accustomed with the term “humid” and “not humid”. One can also explain the atmosphere as “pretty humid”, “moderately humid”, “a little humid”, and so on. Furthermore, it can be presented by a particular real number h which represents the percentage of humidity. Notice that “50%” of humidity can quickly changes value from true to false by a relatively small number ε (50% ± ε%) (Belohlavek & Klir, 2011). The principal idea in Fuzzy Logic is to allow a partial, intermediate value between zero and one. Scales such as 0.1 and 0.9 is interpreted as the degree of which the proposition or certain value is considered true or false (Belohlavek & Klir, 2011).

A standard fuzzy set which differs from classical “true-false” set is defined below

$$A: U \rightarrow [0,1] \qquad (6)$$

This set assigns each element x in U to a number or degree ranging between zero to one. A fuzzy set can be called Fuzzy membership functions if it satisfies a certain value. For example, a man with the height of 176 cm is considered as “tall”. In classical set, a man with the height of 175.5 cm is considered false in the term “tall”, while in real world, there are many people with the height of (176 cm ± *h*) where *h* is a threshold of how "tall" are the people (Jantzen, 2008). This is where fuzzy logic comes in. In a fuzzy set, the *h* variable is defined in a range of fuzzy value from zero to one. A man with the height of 176 cm is considered as "average tall" or 0.5 tall, while a man with the height of 170 cm is considered as "quite tall" or 0.36 tall. The whole fuzzy sets can be described in the diagram below.

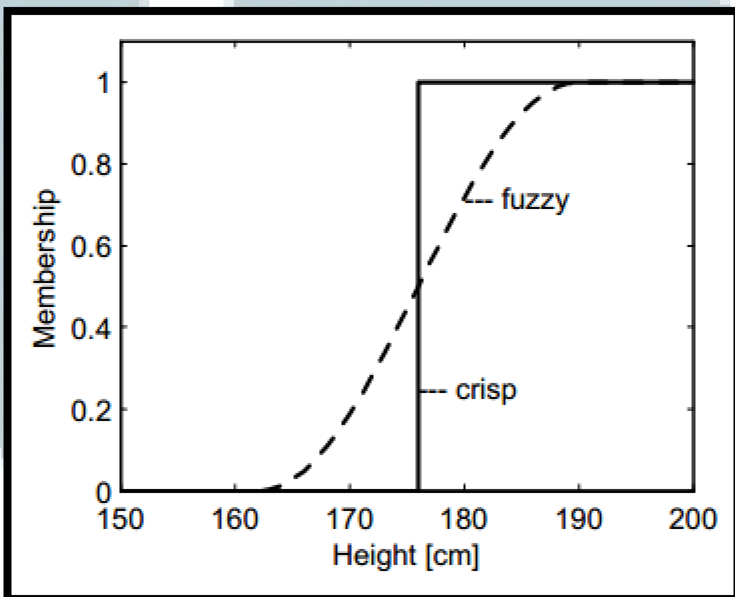

Figure 2.16. 2 definitions of "tall" man (Jantzen, 2008)

Notice that "crisp" set represents a classical set of what defines "tall". Fuzzy sets contains an approximate value which can define "tall" in a more fuzzy and acceptable value.

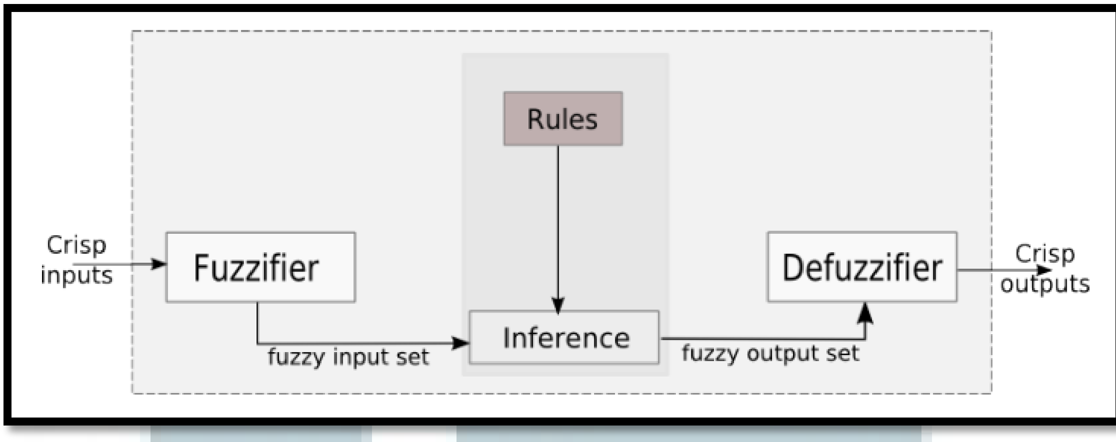

Figure 2.17. Fuzzy Logic System overview (Bilkent University, 2010)

Fuzzy logic has been implemented in many control systems such as air conditioners, robotics, Sendai subway system, crowd systems, and other systems. Implementing fuzzy logic into a system requires three steps (Aziz & Parthiban, 2013).

1. Fuzzification. This step use membership functions to turn the fuzzy set into a fuzzy diagram. For example, an ice cream seller describes a crowd buying ice cream as "crowded" and "unfilled" (Fulton School of Engineering, 2004). The fuzzy sets are described as an equation below.

$$\mu_{crowded}(x) = \begin{cases} 0\,; if\ x < 25 \\ \frac{x-25}{25}\,; if\ 25 < x < 50 \\ 1\,; if\ x > 50 \end{cases} \tag{7}$$

$$\mu_{unfilled}(x) = \begin{cases} 1\,; if\ x < 25 \\ -\frac{x+50}{25}\,; if\ 25 < x < 50 \\ 0\,; if\ x > 50 \end{cases} \tag{8}$$

Equation 2.5. Sample fuzzy membership rules of crowds.

From the equation (6) and (7), we can derive the graphical representation of the fuzzy membership values as follow

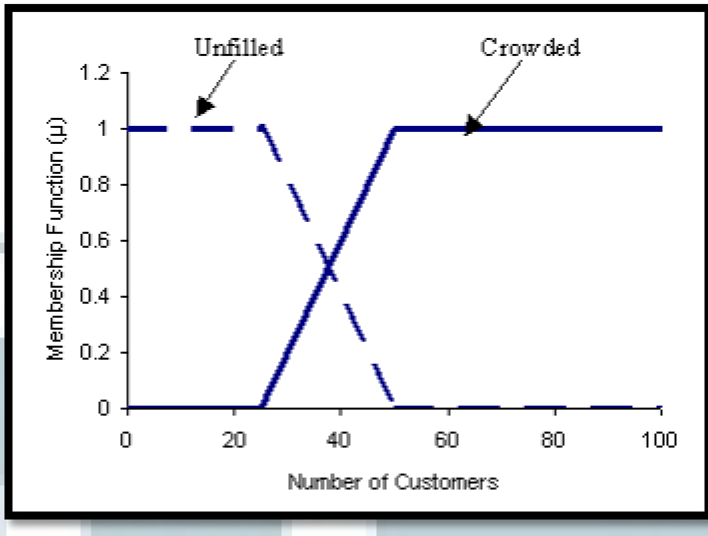

Figure 2.18. Fuzzy Diagram of crowds (Fulton School of Engineering, 2004)

2. Fuzzy Rule Evaluation. This step applies the rules into the early fuzzy sets to derive an actual fuzzy result. Fuzzy rules are implemented in “if-then” statements which contain some fuzzy sets and its implication. For example, fuzzy rules can be expressed in terms “if the room is HOT and number of people is CROWDED, then spin fan blade FAST”. HOT, CROWDED and FAST are imprecise term and they are both fuzzy terms. These rules add human-like reasoning capabilities to artificial intelligences (Pulo, 2000).

   To evaluate fuzzy rules with the given fuzzy membership functions, it is necessary to calculate the fuzzy values related to the fuzzy terms expressed in fuzzy values. Using the example of ice cream parlor, consider an additional fuzzy membership function of a temperature.

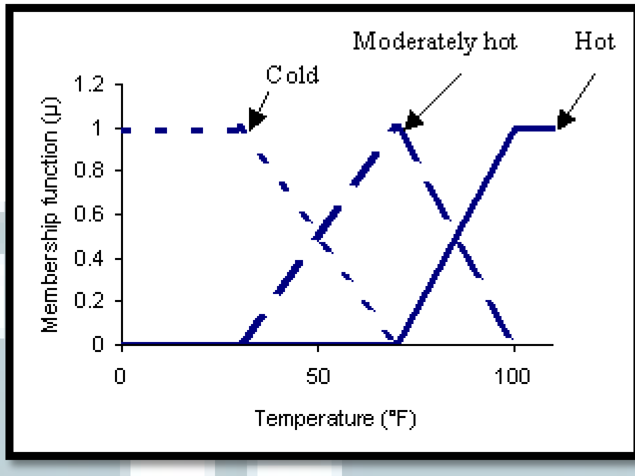

Figure 2.19. Fuzzy Diagram of temperature (Fulton School of Engineering, 2004)

Suppose a fuzzy rule is as follow

"If the temperature is HOT OR the temperature is MODERATELY HOT,

Then ice cream parlor is CROWDED"

And the current temperature is $75\,^{o}F$. By using the implication rule, we can replace the OR operator to $\max(union)$ method.

$$\mu_{temperature}(75) = \mu_{hot} \cup \mu_{moderate-hot}$$

$$= \max(\mu_{hot} \cup \mu_{moderate-hot})$$

$$= \max(0.167, 0.833)$$

$$= 0.833$$

From the result of the equation, the temperature with value $75\,^{o}F$ is considered a 0.833 fuzzy value of "moderate-hot" (Fulton School of Engineering, 2004).

3. Defuzzification. Defuzzification is the method to convert a fuzzy value obtained from the rule evaluation to a single scalar or crisp value, which will be used by the fuzzy control system. Defuzzification can be described as the reversed process from fuzzification (Ross, 2010). The defuzzification process can be a union of the fuzzy sets or the combination of fuzzy values to a fuzzy membership functions.

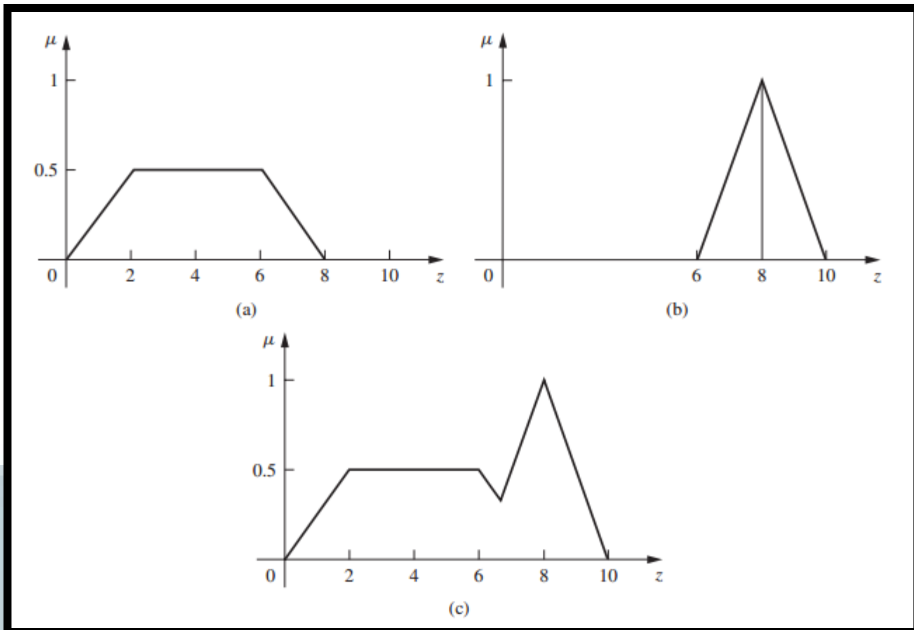

Figure 2.20. A union of fuzzy membership functions (Ross, 2010)

When deriving crisp values from a fuzzy membership functions, there are several defuzzification methods which can be used.

1) Max Membership principle. This method extracts the crisp value from the highest peak of the diagram (Ross, 2010). This method is also known as *height* method and can be described in algebraic expression:

$$\mu_C(Z) \geq \mu_C(z) \quad (9)$$

Z is the crisp value obtained from the highest value of set z in the fuzzy membership function.

2) Centroid or center-of-gravity method obtains the crisp value by computing the model's center of gravity (Fulton School of Engineering, 2004). The center of gravity is obtained from the accumulation of fuzzy values. It is

the most commonly used technique and the most appealing method for defuzzification.

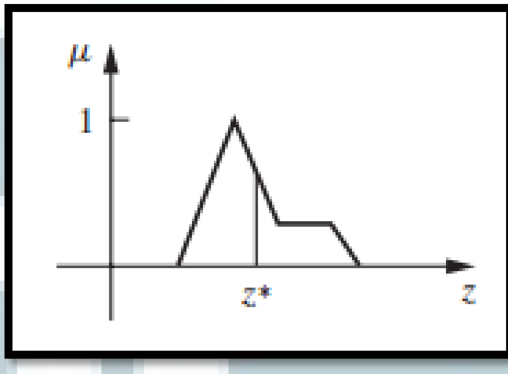

Figure 2.21. Centroid method (Fulton School of Engineering, 2004).

Centroid method can be expressed as

$$Z = \frac{\int \mu_C(z)\,.z\,dz}{\int \mu_C(z)\,dz} \qquad (10)$$

3) Weighted Average method. This method calculates crisp value by the sum of the average of each fuzzy set. It is computationally more efficient than the centroid method, but gives fairly accurate result (Ross, 2010, Fulton School of Engineering, 2004). Weighted average method is described in the figure below.

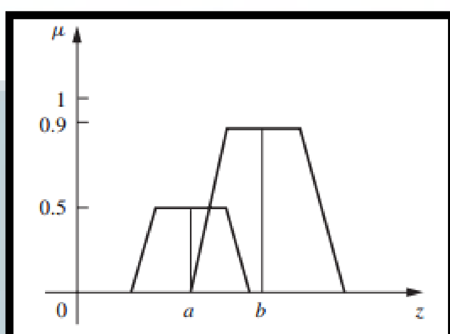

Figure 2.22. Weighted Average of a fuzzy set (Fulton School of Engineering, 2004)

$$Z = \frac{\sum \mu_C(z).z}{\sum \mu_C(z)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(11)$$

From the equation (11) and figure 2.22, the crisp value obtained is

$$Z = \frac{a*0.5+b*0.9}{0.5+0.9} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(12)$$

where a and b is the means of the shape.

4) Mean Max membership or *middle-of-maxima*. This method is similar to the *height* method, except that the locations of the highest peak can be more than a single point (Ross, 2010). Mean Max method can be described with the figure below.

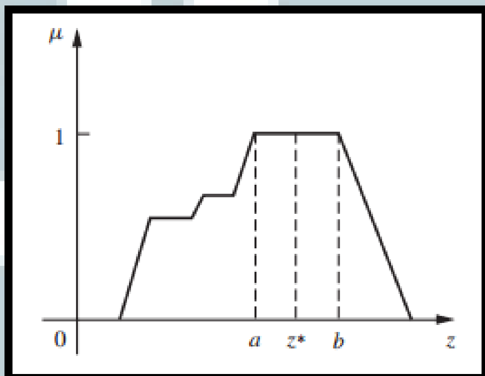

Figure 2.23. *Middle-of-maxima*
*(*Fulton School of Engineering, 2004*)*

The crisp value (z*) is obtained by the mean of the two highest left and right peak.

$$Z = \frac{a+b}{2} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(13)$$

2.4. Shopping Center

Nowadays, shopping centers become an integral part of society lifestyles. According to International Council of Shopping Centers (ICSC, 2013), shopping center is group of retail stores that is planned, developed, and managed as a single property. A shopping mall is a cluster of independent shops planned by one entity, which becomes an alternative to traditional shopping activities (Bajo, 2006). With the growth of shopping centers around the world, shopping centers can be viewed as an effective factor which measures the growth of certain town or city (Sadeghi & Bijandi, 2011). Many researches have also been done involving shopping centers. Research areas such as human behaviors, environment, and social science are done in a shopping center environment. The result of the research itself has been helping the growth of shopping centers around the world.

Figure 2.24. Overview of Plaza Senayan

A multi-agent simulation in shopping centers revolves around how to turn the behavior of real humans in real world shopping center into simulated agents in a virtual environment. Agent-based models are able to integrate individually different variations of consumer’s behavior such as impulse buying, interactions to shopping environment, and other behavior. Consumers are also treated as an intelligent agent with its own shopping preferences (Rauh, Schenk, & Schrodl, 2012).

The field of AI plays a great role in simulating agents in a virtual shopping center. AI gives human-reasoning to intelligent agents, enabling them to perceive their environments and taking decision. Aside AI, visualization also plays an important role. A visual aspect of the simulation can be in a form of 2D or 3D space or based on map. A 3D simulation is a powerful tool to assess and analyze performance such as mobility and user behavior (Dijkstra, 2002). The agents in a simulated shopping center environment are mobile and can move freely. Visualization is important, especially when modeling complex systems, such as shopping centers (Chan, Son, & Macal, 2010).

Figure 2.25. A 3D multi-agent simulation for shopping centers (Dijkstra, 2002)

According to Ali & Moulin (2008), simulation of human behaviors in virtual environment is related to:

1. Believability. This factor is related to the agent's cognitive ability, such as making decision, perception, and other abilities.
2. Usability. Simulations aren't limited to displays or one way interaction between simulation and the user. They have to be equipped with a means of visualization, simulation control tool, and output data generation (Ali & Moulin, 2008).

There are many aspects of human behavior in shopping center which can be simulated. Several behaviors which is commonly simulated are:

1. Shop purchasing decision. Agents in a simulated shopping center environment are able to take decision on what action it should take in the virtual shopping center. The decision itself must be as similar as possible to the real world, whether the agents are buying something or just window shopping. To simulate this behavior, the agents have to generate its own preferences toward a certain shop attributes (Rauh, Schenk, & Schrodl, 2012).

   Mall visitors as agents include many of its own characteristics which is also applied to the real world. Characteristics such as gender, age, financial status, nationality, and other characteristics affect consumption and buying behaviors. However, in a simulated environment, it is necessary to implement only characteristics that affect the most. (Rauh, Schenk, & Schrodl, 2012).
2. Navigation. Autonomous agent navigation can be viewed as the agent's ability to move freely in a simulated environment in order to achieve their goal. Navigation includes agent's decision making in path planning, path selection, and collision avoidance among other agents (Jaafar & McKenzie, 2011). Because of this, each agent must has a coordinated movements with each other in order to simulate a good navigation in a virtual shopping center.